



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

SISTEMA DE DESARROLLO DE SOFTWARE
DE AUTO-CIRCULACIÓN, VIGILANCIA Y
PRODUCTIVIDAD EN CATALOGACIÓN
PARA BIBLIOTECAS UNAM

REPORTE DE TRABAJO PROFESIONAL

QUE PARA OBTENER EL GRADO DE:
LICENCIADO EN
CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A :
MARÍA ALEJANDRA
GUTIÉRREZ AHUMADA

DIRECTOR DE TESIS:
M. EN I. KARLA RAMÍREZ PULIDO

CDMX

2018





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

FACULTAD DE CIENCIAS
Secretaría General
División de Estudios Profesionales

Votos Aprobatorios

LIC. IVONNE RAMÍREZ WENCE
Directora General
Dirección General de Administración Escolar
Presente

Por este medio hacemos de su conocimiento que hemos revisado el trabajo escrito titulado:

**SISTEMAS DE DESARROLLO DE SOFTWARE PARA VIGILANCIA,
AUTO-CIRCULACIÓN Y PRODUCTIVIDAD EN CATALOGACIÓN PARA
BIBLIOTECAS**

realizado por **María Alejandra Gutiérrez Ahumada** con número de cuenta **094299463** quien ha decidido titularse mediante la opción de **trabajo profesional** en la licenciatura en **Ciencias de la Computación**. Dicho trabajo cuenta con nuestro voto aprobatorio.

Propietaria M. en C. María Guadalupe Elena Ibarguengoitia González

Propietaria Dra. Selene Marisol Martínez Ramírez

Propietaria M. en I. Karla Ramírez Pulido
Tutora

Suplente M. en I. Yasmine Macedo Reza

Suplente M. en I. Beatriz Peralta Cortés

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"
CIUDAD UNIVERSITARIA, CD. MX., A 20 DE JUNIO DE 2018

JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

ACT. MAURICIO AGUILAR GONZÁLEZ

Señor sinodal: antes de firmar este documento, solicite al estudiante que le muestre la versión digital de su trabajo y verifique que la misma incluya todas las observaciones y correcciones que usted hizo sobre el mismo.

El alma sin ciencia no es buena...

Proverbios 19:2

Agradecimientos

- ♥ En primer lugar agradezco a Dios padre, Hijo (Jesucristo) y Espíritu Santo, por la vida misma.
- ♥ A la familia, por su constante cuidado y apoyo, principalmente a mi madre.
- ♥ Al amor, Cid Reyes por la inspiración.
- ♥ A la amistad, por la complicidad.
- ♥ A mis profesores, por sus enseñanzas.
- ♥ A la Universidad Nacional Autónoma de México y la Facultad de Ciencias, por la preparación académica.
- ♥ Al Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas y la DGB, por la oportunidad de desarrollarme profesionalmente.

Índice general

Objetivos	XIII
Introducción	XV
MARCO CONTEXTUAL Y CONCEPTUAL	1
1. Dirección General de Bibliotecas (DGB)	3
1.1. Estructura del SIBIUNAM	5
1.2. Colecciones	6
1.3. Servicios	6
2. Sistema de gestión de bibliotecas <i>Aleph</i>	9
2.1. Arquitectura del sistema <i>Aleph</i>	10
2.2. Capas que conforman el sistema <i>Aleph</i>	11
2.3. Módulos del sistema <i>Aleph</i>	12
2.3.1. Módulo de Catalogación	13
2.3.2. Módulo de Adquisiciones/Seriadas	14

2.3.3.	Modulo de Circulación	14
2.3.4.	Modulo de Préstamo Interbibliotecario (PIB)	15
2.4.	Catálogo en línea (Web OPAC)	16
2.5.	<i>Aleph</i> en Oracle	17
3.	Tecnologías utilizadas	21
3.1.	Lenguaje Unificado de Modelado (UML)	21
3.1.1.	Diagrama de caso de uso	23
3.1.2.	Diagramas de actividad	23
3.1.3.	Diagramas de clases	23
3.1.4.	Diagramas de secuencia	24
3.2.	Lenguaje de programación (Java)	25
3.2.1.	Características de Java	25
3.2.2.	Máquina Virtual de Java	26
3.2.3.	Ediciones de Java	27
3.2.4.	JRE y JDK	27
3.2.5.	Entornos de desarrollo para Java	28
3.3.	Entorno de desarrollo integrado (NetBeans)	28
3.4.	Sistema manejador de bases de datos (Oracle)	32
	SISTEMAS PARA BIBLIOTECAS UNAM	35
4.	Consideraciones generales	37

4.1. Dependencias y bibliotecas	38
4.2. Base de datos	39
4.3. Implementación e instalación	43
4.3.1. Casos de uso	44
4.3.2. Configuración	44
4.3.3. Conexión	47
4.3.4. Paquetes	48
4.3.5. Consultas y transacciones	49
4.3.6. Interfaz	51
4.3.7. Requerimientos de hardware	52
4.3.8. Requerimientos de software	53
4.3.9. Instalación	54
5. Vigilancia	57
5.1. Análisis de requerimientos	57
5.2. Diagrama E-R	59
5.3. Casos de uso	60
5.3.1. Validar préstamo	60
5.3.2. Localizar préstamo	62
5.3.3. Localizar usuario	63
5.3.4. Localizar ejemplar	64

5.4. Interfaz	65
5.5. Situación actual	67
6. <i>Auto-Circulación</i>	71
6.1. Análisis de requerimientos	71
6.1.1. Validaciones y acciones	73
6.2. Diagrama E-R	76
6.3. Casos de uso	78
6.3.1. Autenticar usuario	79
6.3.2. Prestar un ejemplar	81
6.3.3. Renovar un préstamo	81
6.3.4. Devolver un ejemplar	82
6.3.5. Validar préstamo	83
6.3.6. Registrar préstamo	86
6.3.7. Validar renovación	88
6.3.8. Registrar renovación	91
6.3.9. Validar devolución	93
6.3.10. Registrar devolución	94
6.3.11. Imprimir Recibo	96
6.4. Interfaz	97
6.5. Situación actual	99

7. Productividad en Catalogación	103
7.1. Análisis de requerimientos	103
7.2. Diagrama E-R	104
7.3. Casos de uso	106
7.3.1. Generar reporte	106
7.3.2. Generar resumen	108
7.4. Interfaz	109
7.5. Situación actual	110
Conclusiones y trabajo futuro	110
APENDICES	114
A. Subsistemas del SIBIUNAM	I
B. Notación UML	V
C. Relaciones entre BD del Sistema <i>Aleph</i>	XI
D. Diagramas E-R del sistema <i>Aleph</i>	XVII
E. Tablas <i>Aleph</i>	XXI
E.1. Tab16 (Due dates, fines & limits)	XXI
E.2. Tab15 (Item Statuses)	XXII
E.3. Tab31 (Patron Statuses)	XXIII

Índice de figuras

1.1. Estructura orgánica de la DGB [5].	4
1.2. SIBIUNAM [9].	5
2.1. Arquitectura multicapa del sistema <i>Aleph</i> [14].	11
2.2. Cliente GUI de <i>Aleph</i> , Módulo de Circulación.	13
2.3. Catalogo en línea de <i>Aleph</i> (Web OPAC) [21].	16
2.4. Estructura de directorios y base de datos de <i>Aleph</i>	18
3.1. Tipos y clasificación de diagramas UML [26].	22
3.2. Estereotipos del UML para representar tipos de clases [22].	24
3.3. Proceso de compilación y ejecución para aplicaciones Java [27].	26
3.4. Aspecto de NetBeans.	29
3.5. Edición de código en NetBeans [36].	30
3.6. Vistas de NetBeans para la administración de los proyectos [36].	31
3.7. Diseño de GUIs en NetBeans.	32
3.8. Representación gráfica de una tabla.	33

4.1.	Estructura del sistema <i>Aleph</i> cuando existen dos bibliotecas.	39
4.2.	Tablas utilizadas en los sistemas de este trabajo.	40
4.3.	Libros en la base de datos <i>Aleph</i> (Títulos-Ejemplares).	42
4.4.	Usuarios en la base de datos <i>Aleph</i>	43
4.5.	Diagrama de paquetes para los sistemas de las bibliotecas UNAM.	49
5.1.	Diagrama de actividades para validar préstamo.	59
5.2.	Diagrama E-R para Vigilancia.	60
5.3.	Casos de uso del sistema Vigilancia.	61
5.4.	Interfaz del sistema Vigilancia (Biblioteca Central).	66
5.5.	Imágenes y colores para identificar el estado del préstamo.	66
5.6.	Interfaz del sistema Vigilancia multiple	67
5.7.	Estadísticas del sistema <i>Vigilancia</i>	69
6.1.	Diagrama de actividades de circulación.	73
6.2.	Diagrama E-R del sistema <i>Auto-Circulación</i>	77
6.3.	Casos de uso del sistema Auto-Circulación.	79
6.4.	Interfaz del sistema <i>Auto-Circulación</i> en Biblioteca Central.	98
6.5.	Interfaz del sistema <i>Auto-Circulación</i> modificadas.	98
6.6.	Estadísticas del sistema <i>Auto-Circulación</i>	101
7.1.	Diagrama de actividades.	105
7.2.	Diagrama E-R para <i>Productividad en Catalogación</i>	105

7.3.	Casos de uso del sistema <i>Productividad en Catalogación</i>	106
7.4.	Interfaz del sistema <i>Productividad en Catalogación</i>	109
B.1.	Notación UML para representar diagramas de caso de uso.	V
B.2.	Notación UML para representar diagramas de actividad.	VI
B.3.	Notación UML para representar diagramas de clases.	VIII
B.4.	Notación UML para representar diagramas de paquetes.	IX
B.5.	Notación UML para representar diagramas de secuencia.	X
C.1.	Relación entre bases y tablas ALEPH [19].	XII
C.2.	Relación entre tablas de la base administrativa [19].	XIII
C.3.	Relación entre tablas para ejemplares [19].	XIV
C.4.	Relación entre tablas de la base bibliográfica [19].	XV
D.1.	Diagrama E-R: Usuarios [18]	XVIII
D.2.	Diagrama E-R: Circulación-Ejemplares [18]	XIX

Índice de cuadros

4.1. Características mínimas del CPU.	52
4.2. Dispositivos externos adicionales (hardware).	53
4.3. Requerimientos mínimos de software.	54

Objetivos

Objetivo general

Desarrollar, extender y mantener los sistemas *Auto-Circulación*, *Vigilancia* y *Productividad en Catalogación* de la Dirección General de Bibliotecas.

Objetivos secundarios

- Presentar el sistema *Aleph* mediante el que están administradas las bibliotecas de la UNAM, principalmente la base de datos consultada por las aplicaciones *Vigilancia*, *Auto-Circulación* y *Productividad en Catalogación*.
- Mostrar el sistema *Vigilancia* para el resguardo del acervo de las bibliotecas donde es utilizado.
- Dar a conocer el sistema *Auto-Circulación* en el que de forma automatizada se realizan préstamos, renovaciones y en casos específicos devoluciones.
- Explicar el sistema *Productividad en Catalogación*, utilizado para medir la productividad de un área.
- Mencionar sugerencias que facilitarían el desarrollo de aplicaciones que utilizan la base de datos del sistema *Aleph*.

Introducción

La Dirección General de Bibliotecas (DGB) es la encargada de coordinar más de un centenar de bibliotecas que pertenecen a la Universidad Nacional Autónoma de México (UNAM) en los diferentes subsistemas bibliotecarios: bachillerato, licenciatura y posgrado, investigación científica, investigación en humanidades, y por último extensión y administración.

Cada una de las bibliotecas cuenta con al menos una colección bibliográfica, dicha colección puede ser general, histórica o especializada, dependiendo de la naturaleza de la dependencia universitaria de la cual se trate. Sin embargo de manera indistinta las bibliotecas tienen entre sus obligaciones el compartir y resguardar su acervo.

Entre las actividades que la DGB presta al conjunto de bibliotecas a su cargo a través de la Subdirección de Informática y el Departamento de Análisis (DADMS), Desarrollo y Mantenimiento de Sistemas, están el automatizar algunos de sus servicios y apoyar el desempeño de otros haciendo uso de sistemas computacionales, los cuales deberán cubrir las necesidades y características específicas de cada dependencia.

Cabe mencionar que desde el año 2002 el sistema bibliotecario de la UNAM, es administrado por un software comercial cuyo propósito es administrar bibliotecas y/o unidades de información, fue diseñado y desarrollado fuera de nuestro país (Israel) y lleva por nombre ***Aleph*** (*Automated Library Expandable Program*).

Es en particular, la base de datos de ***Aleph*** la que utilizan los sistemas descritos a continuación, por lo que los desarrollos inician con un análisis de

las tablas involucradas y termina con las validaciones necesarias para que no se altere, afecte, pierda o se corrompa la información de esta base de datos.

En resumen, el objetivo principal del Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas y de este trabajo es proporcionar herramientas que automaticen, apoyen y/o simplifiquen la administración del sistema bibliotecario, de manera simple, clara y específica, reduciendo costos de mantenimiento y licencias de uso de un software propietario.

A continuación, se da una breve descripción del contenido de cada uno de los capítulos que integran este reporte, comenzando con la teoría y finalizando con las aplicaciones.

El Capítulo 1 brinda un panorama general sobre la Dirección General de Bibliotecas (DGB), donde son utilizados los sistemas abordados en este trabajo. En el Capítulo 2 se describen los aspectos generales de *Aleph*, su arquitectura, características, estructura de directorios y los módulos que lo integran (siendo el *Módulo de Circulación* el de mayor interés para fines de este trabajo).

En el Capítulo 3 se mencionan las tecnologías relacionadas y utilizadas en este trabajo, por un lado el Lenguaje Unificado de Modelado (UML¹) que es un lenguaje gráfico para visualizar, especificar, construir y documentar sistemas de software (Sección 3.1). Por otro lado se presentan aquellas tecnologías utilizadas en el desarrollo de los sistemas, el lenguaje de programación utilizado es *Java* (Sección 3.2) y el entorno de desarrollo integrado es *Net-Beans* (Sección 3.3), de ellos se mencionan algunas de sus características más importantes (razón por la cual fueron seleccionados). Además se incluye una breve base teórica acerca de bases de datos y el sistema manejador de bases de datos *Oracle* (Sección 3.4).

A partir del Capítulo 5 se presentan los sistemas desarrollados y documentados en este trabajo que son:

El sistema encargado de validar los préstamos llamado *Vigilancia* tiene el objetivo de resguardar la colección del recinto y se logra al simplificar la información necesaria para presentarla de manera concreta y clara al per-

¹ Por sus siglas en inglés UML (*Unified Modeling Language*) [22].

sonal de vigilancia ubicado en la salida de las bibliotecas. El flujo de salida de usuario en ciertos horarios es numeroso, por lo que, el sistema debe ser eficiente (Capítulo 5).

El sistema de ***Auto-Circulación*** automatiza por completo los préstamos a domicilio, las renovaciones y devoluciones (a criterio de la biblioteca), servicios de mayor demanda y cotidianidad que requieren ser registrados. En otras palabras, sin la intervención de un bibliotecario un usuario puede registrar su préstamo o renovación. ***Auto-Circulación*** reproduce las mismas acciones y efectos sobre la base de datos que realiza el sistema ***Aleph*** cuando ejecuta dichos procesos, a través de su propia interfaz gráfica denominada Cliente GUI², con el objetivo de garantizar que la información sea consistente y compatible entre ambos sistemas. A su vez ***Auto-Circulación*** fue creado para disminuir la carga de trabajo del *Módulo de Circulación* de ***Aleph*** y así reducir el tiempo de espera de los usuarios (Capítulo 6).

El sistema de ***Productividad en Catalogación*** proporciona al usuario³ reportes que indican la cantidad de ejemplares catalogados por cada empleado en un periodo de tiempo determinado, permitiendo valorar el desempeño de los trabajadores que realizan esta actividad. La aplicación genera un archivo, el cual cuenta con el detalle de cada una de las catalogaciones y un resumen que contiene los nombres de los catalogadores⁴ y su productividad en otro archivo. Ambos pueden ser visualizados en la interfaz del programa al momento de ser generados, los cuales son almacenados en el equipo para su futura consulta (Capítulo 7).

Finalmente se expondrán las conclusiones y recomendaciones para los sistemas contenidos en este trabajo.

² Por sus siglas en inglés GUI (*Graphic User Interface*) [14].

³ Para fines de este trabajo es el jefe del departamento de procesos técnicos de la DGB o bien al responsable de la biblioteca.

⁴ Catalogador, ra. (1) Que cataloga. (2) Persona que forma catálogos [1].



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

M A R C O

C O N T E X T U A L

Y

C O N C E P T U A L



Capítulo 1

Dirección General de Bibliotecas (DGB)

La Dirección General de Bibliotecas (DGB) ocupa el tercer nivel dentro de la estructura universitaria, después de la Rectoría y de la Secretaría de Desarrollo Institucional, de la cual forma parte. Está conformada por una secretaría académica, cinco subdirecciones, tres secretarías técnicas, veintitrés departamentos, una unidad administrativa, una secretaría particular y una auxiliar, todos ellos coordinados por la Dirección General¹ [3] (Ver la Figura 1.1).

La DGB es la dependencia encargada de coordinar el Sistema Bibliotecario y de Información de la UNAM (SIBIUNAM) conforme al reglamento del sistema bibliotecario [11] que establece el Consejo del Sistema Bibliotecario, determinando las medidas que relacionan y desarrollan a las bibliotecas.

¹ La información que se presenta a continuación y su actualización puede ser consultada en su mayoría en la página de Internet de la Dirección General de Bibliotecas (DGB) en sus diferentes apartados, misma en la que está basado este capítulo.

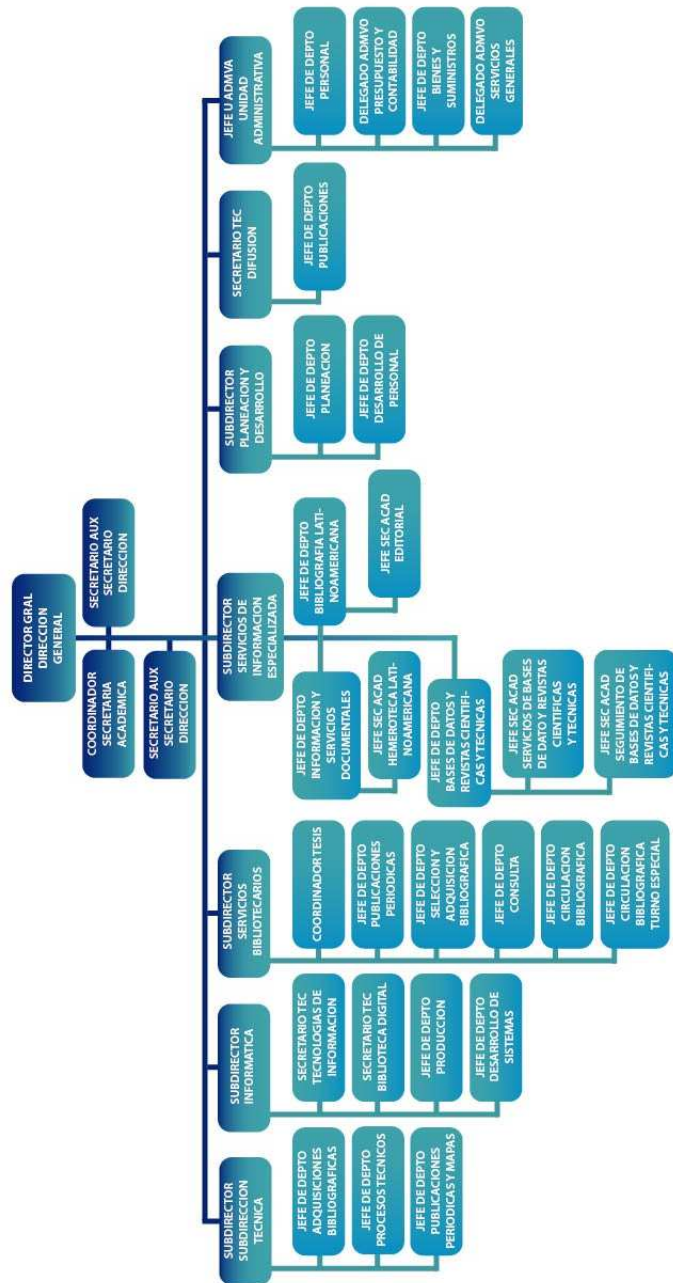


Figura 1.1: Estructura orgánica de la DGB [5].

1.1. Estructura del SIBIUNAM

En el 2017 el SIBIUNAM tuvo una población de 396,960 universitarios [6], entre los cuales se incluye estudiantes de todos los niveles educativos (iniciación universitaria, técnico, técnico profesional, propedéutico, bachillerato, licenciatura y posgrado) del sistema escolarizado y del sistema de universidad abierta, así como al personal académico de la Universidad.

En ese mismo año se contaba con 134 las bibliotecas, consideradas la base del Sistema Bibliotecario de la UNAM, las cuales están organizadas en cinco subsistemas: bachillerato (16 bibliotecas), licenciatura y posgrado (48 bibliotecas), investigación científica (31 bibliotecas), investigación en humanidades (20 bibliotecas) y el de dependencias de extensión y administración universitaria (19 bibliotecas) [7](Ver la Figura 1.2). El nombre de las dependencias que integran cada uno de dichos subsistemas puede ser consultado en el Apéndice A [8]. Están distribuidas geográficamente de la siguiente manera, 72 en el Campus de Ciudad Universitaria, 34 en la zona metropolitana del Distrito Federal y del Estado de México, 29 en diferentes entidades federativas [7].



Figura 1.2: SIBIUNAM [9].

El alcance máximo que pueden tener los sistemas presentados en este trabajo es que todos sean utilizados en cada una de las dependencias del SIBIUNAM, sin embargo, el alcance esperado es que al menos uno de los sistemas lo haga.

1.2. Colecciones

La UNAM cuenta con catálogos colectivos que te permiten consultar los registros de todos los acervos del sistema bibliotecario de la UNAM: libros (LIBRUNAM), revistas (SERIUNAM), tesis (TESIUNAM), mapas (MAPA-MEX), así como otras colecciones especializadas.

La colección mas importante es LIBRUNAM, pues a partir de ella se generan las bases locales de cada dependencia del SIBIUNAM. "En 2017 se contabilizaron referencias bibliográficas de 1'767,898 títulos y 6'994,141 volúmenes, de los cuales alrededor de 426,864 son libros electrónicos" [10].

1.3. Servicios

Para que los usuarios hagan uso de las colecciones que se resguardan en las bibliotecas, se proporcionan diversos servicios bibliotecarios como préstamo interno, a domicilio e interbibliotecario; consulta de colecciones en sala; consulta de bases de datos y servicios de documentación; así como los servicios de búsqueda de información, obtención de documentos y referencia por vía telefónica, correo electrónico o de manera personalizada, entre otros.

El préstamo de libros es fundamental para apoyar y complementar la formación académica de los usuarios, así como las tareas de investigación y docencia, dichos servicios se deben proporcionar de una forma ágil y oportuna para que se satisfagan los requerimientos de los usuarios y así lograr que los materiales se aprovechen al máximo.

Gracias a las tecnologías de cómputo y telecomunicaciones, los universitarios tienen acceso a las colecciones electrónicas adquiridas por la DGB, así como a las de acceso libre o las generadas por la propia UNAM. Entre este tipo de recursos documentales se destacan las bases de datos de carácter internacional, las revistas, tesis y libros electrónicos que están a disposición de todos los usuarios de la comunidad universitaria; aún para aquellos que se encuentran fuera de las instalaciones universitarias, en cuyo caso, dichos recursos tienen mayor importancia.

”Durante 2017 se registró la visita de más de 13 millones de personas que utilizaron la infraestructura tecnológica y los espacios ofrecidos en el sistema, además de consultar en sala 10.7 millones de libros y otro tipo de documentos impresos y de llevar a su domicilio 2.3 millones de volúmenes”[10].

Cabe resaltar que el préstamo externo es el segundo en demanda, requiere de cierta administración, además es objeto de estudio de este reporte de trabajo profesional para dos de los sistemas.

En este capítulo se explica la función de la DGB y la estructura del SIBIUNAM con la intención de mostrar el alcance en bibliotecas, servicios y acervo que pueden llegar a tener las aplicaciones en este trabajo.

Capítulo 2

Sistema de gestión de bibliotecas *ALEPH (Automated Library Expandable Program)*

Aleph (*Automated Library Expandable Program*) es un software diseñado y desarrollado para la administración de bibliotecas y/o unidades de información. Fue desarrollado en la Universidad Hebrea de Jerusalén por un equipo de programadores, analistas y bibliotecarios [12].

Está basado en una arquitectura Cliente/Servidor ¹ con un fuerte soporte del sistema manejador de bases de datos relacionales (RDBMS²) *Oracle*.

¹ Permite que la carga del procesamiento de una aplicación sea distribuido entre dos máquinas. El *cliente* puede asumir la mayor parte de la carga del procesamiento (ejecución de aplicaciones), mientras que el *servidor* donde se encuentra la base de datos, se dedica a resolver las consultas.

² Por sus siglas en inglés RDBMS (*Relational DataBase Management System*) [39].

2.1. Arquitectura del sistema *Aleph*

El sistema *Aleph* proporciona y automatiza los servicios de las bibliotecas, suministrando servicios de aplicación a sus clientes mediante sus *API*³. Su arquitectura está basada en multicapas, siguiendo al modelo Cliente/Servidor, en donde la comunicación esta basada en un sistema de transacción independiente, sin embargo, los servidores *Aleph* mantienen conexiones continuas (con interrupciones) a la base de datos, para asegurar su alto rendimiento.

Tiene un diseño de bases de datos flexible y se encuentra compuesto por módulos separados pero interrelacionados, éstos son: módulo de Catalogación, módulo de Circulación, módulo de Adquisiciones/Seriadas, módulo de Administración, módulo de Préstamo Interbibliotecario y el catálogo en línea (Web OPAC⁴) que no está dentro de la interfaz gráfica.

Basándose sobre un esquema determinado, distribuido en un sistema lógico sobre un diseño orientado a objetos, las dos características más importantes de la arquitectura son [14]:

1. Multicapa, modelo Cliente/Servidor - El sistema *Aleph* está dividido en segmentos lógicos con una interfaz basada en procesamiento paralelo (Ver la Figura 2.1).
2. Modular - El punto clave de un diseño de distribución lógica como la base del sistema *Aleph* es la modularidad; tanto para el caso vertical (entre capas) como el horizontal (dentro de las capas), lo que asegura la facilidad del sistema para su mantenimiento y expansión.

El sistema *Aleph* puede ser instalado en las siguientes plataformas: SUN SOLARIS, LINUX REDHAT, IBM/AIX y ALPHA UNIX [14].

³ Por sus siglas en inglés API (*Application Program Interface*) [27].

⁴ Por las siglas en inglés OPAC (*Online Public Access Catalog*) [14].

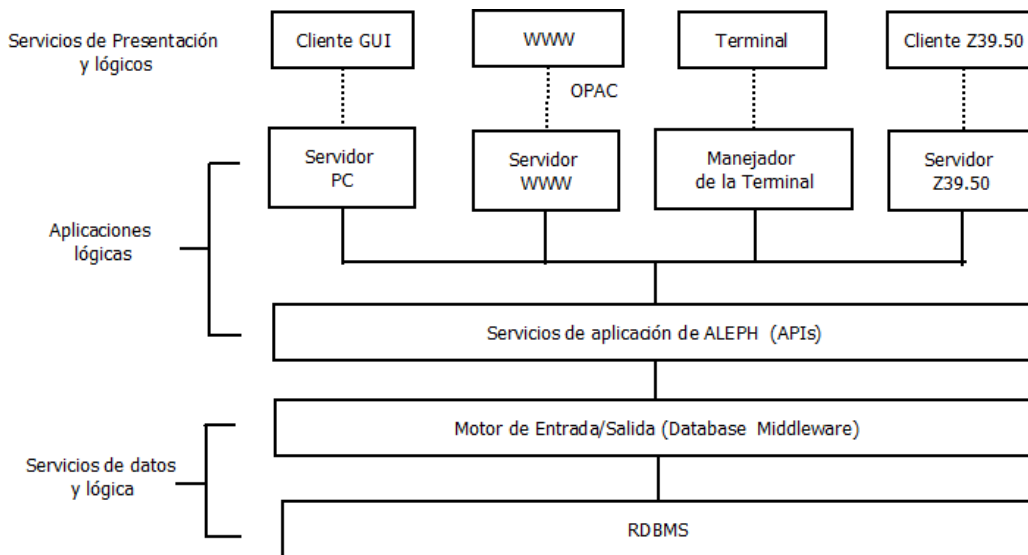


Figura 2.1: Arquitectura multicapa del sistema *Aleph* [14].

2.2. Capas que conforman el sistema *Aleph*

A continuación se describen las capas del sistema *Aleph* que se observan en la Figura 2.1.

La capa de servicios de presentación y lógicos, está integrada por las interfaces mediante las cuales interactúa el usuario con el sistema. A la capa de aplicaciones lógicas la componen una capa frontal llamada capa de servidores de aplicación compuesta de servidores dedicados para cada interfaz y la capa de servicios de aplicación. En la capa de servidores de aplicación, cada servidor de aplicación recibe del usuario una petición, la traduce a un formato uniforme (sin importar su origen) y lo dirige a la capa de servicios de aplicación.

La capa de servicios de aplicación (API) es considerada el corazón de *Aleph*, ya que provee los servicios a la biblioteca para los diferentes clientes. Por ejemplo, el *FIND API* proporciona servicios de búsqueda a todos los clientes después de que una consulta es invocada en uno de los clientes [14]. Como parte de su arquitectura como sistema abierto, *Aleph* incluye un mecanismo para integrar nuevos API así como ampliar los servicios de

aplicación a otros clientes o aplicaciones, lo cual proporciona un considerable potencial de expansión.

La capa de servicios de datos y lógica, consta del motor de Entrada/-Salida (*Database Middleware*) y la base de datos. Éste motor es la capa de administración de datos de alto nivel, donde un servidor lógico proporciona servicios de datos a los objetos de los servicios de aplicación. Contiene un grupo de objetos intermedios entre las aplicaciones y la base de datos, es decir, traduce la petición de una aplicación a una secuencia de órdenes en la base de datos. Teniendo un nivel intermedio del motor de Entrada/Salida entre las aplicaciones y el sistema manejador de bases de datos, se asegura la máxima flexibilidad del diseño lógico y físico de este último.

La base de datos de *Aleph* se ejecuta bajo el sistema manejador de bases de datos relacionales Oracle, el cual se detalla en la Sección 2.5.

2.3. Módulos del sistema *Aleph*

Los módulos funcionales del sistema *Aleph* se trabajan mediante el cliente GUI⁵, éste es un ambiente en el cual el usuario puede interactuar desde su computadora personal con el servidor de *Aleph*, es un software gráfico fácil de usar, que incluye botones, íconos, ventanas y menús (la Figura 2.2 muestra la interfaz gráfica de *Aleph* para el *Módulo de Circulación*). Éste se encuentra almacenado sobre la computadora personal de trabajo, con dicho software el usuario podrá interactuar con las bases de datos.

Los módulos que integran el cliente gráfico de *Aleph* son:

- Catalogación.
- Adquisiciones/Seriadas.
- Circulación.

⁵ El cliente GUI puede ser configurado con base en los requerimientos específicos de un grupo de usuarios y ser distribuido por el administrador del sistema bibliotecario a todos los usuarios a través del servidor de *Aleph*.

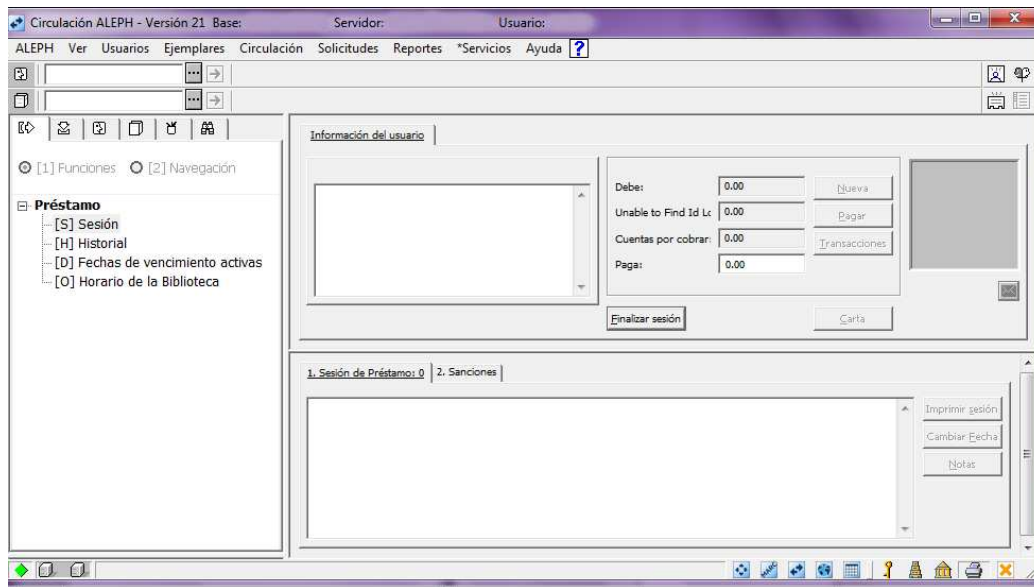


Figura 2.2: Cliente GUI de *Aleph*, Módulo de Circulación.

- Préstamo interbibliotecario (PIB).
- Administración.

2.3.1. Módulo de Catalogación

El *Módulo de Catalogación* integra funciones de catalogación con todos los demás aspectos del sistema, permite crear, editar y borrar registros bibliográficos, así como añadir, cambiar y eliminar ejemplares asociados.

La introducción de los datos se puede hacer de forma libre o por vía de formularios prefijados y debido a lo variable de la descripción bibliográfica y a los diferentes tipos de documentos (libros, monografías, revistas, mapas, etcetera), el sistema presenta una gran flexibilidad en la introducción de la información.

Interactúa con bibliotecas bibliográficas almacenando los datos en *Unico-*

*de*⁶ y el sistema soporta *MARC*⁷ ofreciendo a los usuarios una sola base de datos y al personal que utiliza el módulo la posibilidad de definir libremente campos locales. La adopción de estos estándares de catalogación asegura compatibilidad de la información con otros sistemas.

2.3.2. Módulo de Adquisiciones/Seriadas

La funcionalidad del *Módulo de Adquisiciones/Seriadas* se encarga de la gestión de las compras, canjes y donaciones de material en la biblioteca, mediante procesos para ordenar, facturar, reclamar y recibir el material. Permite administrar monedas, proveedores, órdenes y facturas, así como el material recibido.

Ofrece a su vez acceso a los datos bibliográficos y de acervos, así como a la información de órdenes, fondos (presupuestos) y proveedores. El acceso a la información puede realizarse al utilizar una variedad de puntos de indexación definidos por la biblioteca, por ejemplo, ISSN⁸, palabras clave y títulos, finalmente permite el uso de protocolos de intercambio de datos electrónicos *EDI*⁹ para agilizar las funciones de adquisición.

2.3.3. Modulo de Circulación

El *Módulo de Circulación* permite registrar y verificar las diferentes actividades de circulación de la biblioteca (préstamo, resello, reservas, devoluciones, solicitudes de fotocopias). El sistema soporta diferentes periodos de circulación basados en los tipos de ejemplar, usuario, y de los periodos

⁶ *Unicode* asigna un identificador numérico único para un carácter.

⁷ Del inglés (*MAchine Readable Cataloging*), es un estándar para representar y compartir información bibliográfica [2].

⁸ Por sus siglas en inglés ISSN (*International Standard Serial Number*). Número Internacional Normalizado para Publicaciones Periódicas, es un código numérico reconocido internacionalmente, el cual brinda un medio seguro e inequívoco para obtener, administrar y transferir información referente a las publicaciones periódicas a nivel nacional e internacional, evitando posibles errores de identificación del título o la información hemerográfica [2].

⁹ Por sus siglas en inglés EDI (*Electronic Data Interchange*) [2].

de préstamo (fórmulas para el cálculo de fechas de vencimiento). Estos pueden definirse por la biblioteca para satisfacer sus políticas¹⁰ de circulación y pueden cambiar de acuerdo a sus necesidades.

Cuando un libro es prestado, el sistema verifica tanto el usuario como el libro, y si existiera alguna restricción, entonces el bibliotecólogo es notificado. Cuando un libro es devuelto, el sistema verifica si el libro fue regresado a tiempo. El módulo de circulación es un sistema integrado para el manejo de dinero, despliega las multas y otras sanciones de efectivo definidas por la biblioteca, permitiendo concluir las o pagarlas rápidamente.

La Figura 2.2 es la captura de la pantalla del *Módulo de Circulación* del sistema *Aleph*, donde se puede observar que los datos de entrada son el identificador de un usuario o un ejemplar, a partir del cual se pueden realizar las actividades antes señaladas desplegando la información relacionada.

2.3.4. Modulo de Préstamo Interbibliotecario (PIB)

El *Módulo de Préstamo Interbibliotecario* controla y gestiona las transacciones de préstamo entre bibliotecas clientes y proveedoras, para atender solicitudes de sus usuarios que no pueden ser resueltas con los acervos propios. Este proceso supone una relación entre la biblioteca solicitante y la biblioteca proveedora.

En el sistema *Aleph* el módulo PIB es compatible con el protocolo *ISO 10160/10161* y está integrado con el *Módulo de Circulación* y el *OPAC*. Los usuarios pueden usar el *OPAC* para hacer sus solicitudes que pueden ser visualizadas a través de la función de información para los usuarios del sistema. Cabe mencionar que todas las funciones de circulación están completamente integradas con este módulo.

¹⁰ Son las condiciones para la concesión de material a algún usuario.

2.4. Catálogo en línea (Web OPAC)

El catálogo en línea nombrado en *Aleph* como *OPAC* funciona como la entrada principal del usuario de la biblioteca al sistema *Aleph*, le permite consultar cuáles títulos integran la colección de la biblioteca. En la Figura 2.3 se muestra su apariencia.

Como el sistema está basado en páginas *HTML*, puede ser personalizado para adecuarse a las especificaciones requeridas por la biblioteca. Solo requiere un navegador estándar para la Web en la estación de trabajo del usuario, la cual permite el acceso a la información (catálogos bibliográficos) de la biblioteca en cualquier momento y lugar, aún durante los procedimientos de respaldo del sistema y generación de reportes.

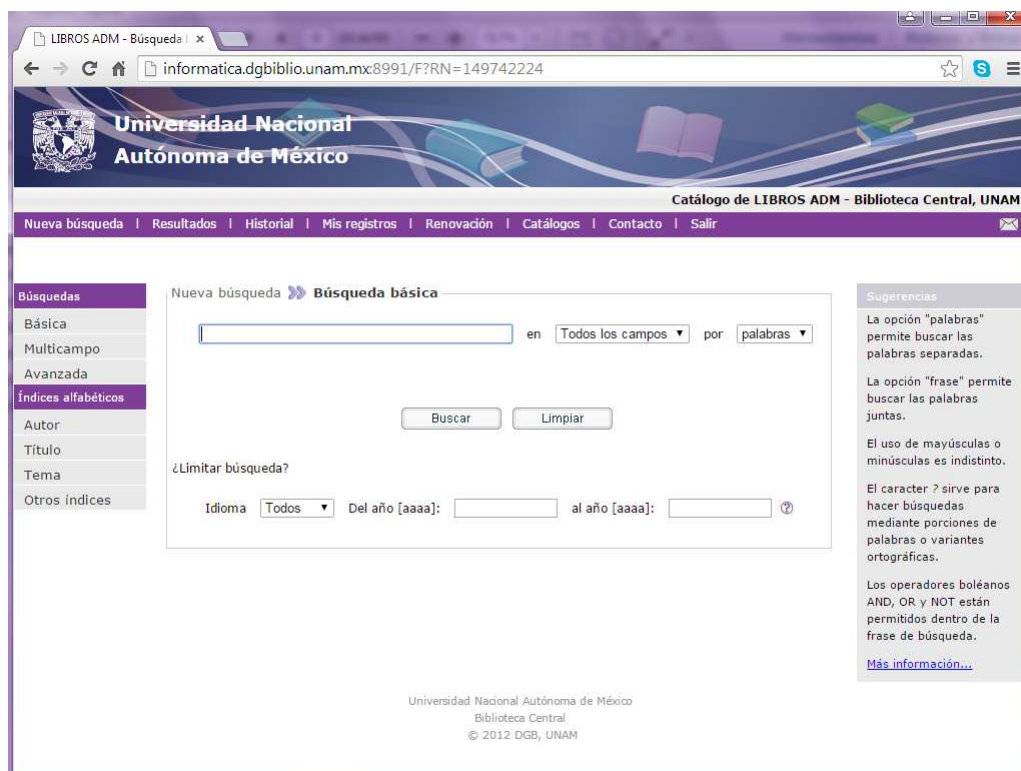


Figura 2.3: Catálogo en línea de *Aleph* (Web OPAC) [21].

2.5. *Aleph* en Oracle

Como ya se mencionó *Aleph* está basado en Oracle, donde una instalación típica incluye varias "bibliotecas"¹¹ de *Aleph*. Cada una tiene su propio conjunto de definiciones (tablas de configuración) y datos (tablas en Oracle).

Cada biblioteca *Aleph* está implementada por un usuario Oracle por separado con un conjunto de tablas que contienen los datos de la biblioteca y una estructura de directorios por separado.

La estructura de directorios inicia con el subdirectorio raíz de la biblioteca *Aleph*, el cual contiene tablas de configuración, archivos temporales, archivos de impresión, etcétera.

Generalmente, una biblioteca real¹² tiene por lo menos tres bibliotecas *Aleph* (bibliográfica, administrativa y autoridades). Sin embargo, algunas pueden compartir una biblioteca bibliográfica y viceversa, una bibliográfica puede abastecer varias administrativas.

Existen varios tipos de bibliotecas *Aleph* las cuales se identifican por un código, que consta de tres caracteres seguidos de dos dígitos que indican el tipo de biblioteca (siguiendo el esquema de dígitos de ExLibris¹³ que es una convención de nombres y no un requerimiento del sistema).

"Bibliotecas Bibliográficas (BIB) - Contienen registros bibliográficos, que se identifican por un número entre 01-09 (por ejemplo USM01)" [13].

"Bibliotecas Administrativas (ADM) - Contienen la información de adquisiciones, circulación, empleados y usuarios, éstos se identifican por un número entre 50-59 (por ejemplo USM50)" [13].

"Bibliotecas de Autoridades (AUT) - Contienen registros de autoridad de formularios de encabezados, relaciones entre encabezados, identificadas por un número entre 10-19 (por ejemplo USM11)" [13].

¹¹ Diferentes bibliotecas representan así mismo diferentes tipos de información.

¹² Se le llama biblioteca real, a una biblioteca física, es decir, el edificio, material (básicamente libros) y usuarios.

¹³ Empresa encargada de la comercialización del sistema *Aleph*.

”Bibliotecas de Préstamo Interbibliotecario (ILL) - Contienen copias de los registros bibliográficos que son solicitados por ILL que se identifican por un número entre 40-49 (por ejemplo USM40)” [13].

La DGB identifica las bases de datos de *Aleph* con la primera letra según el tipo de material de los acervos, es decir, ”L” para libros, ”P” para las publicaciones (revistas), ”M” mapas, etcétera. Dos caracteres (alfanuméricos) que representan la dependencia del SIBIUNAM y los dos últimos son dígitos según el esquema de ExLibris antes mencionado.

La Figura 2.4 muestra la estructura de directorios y base de datos tomando como prefijo de la biblioteca ”lxx”.

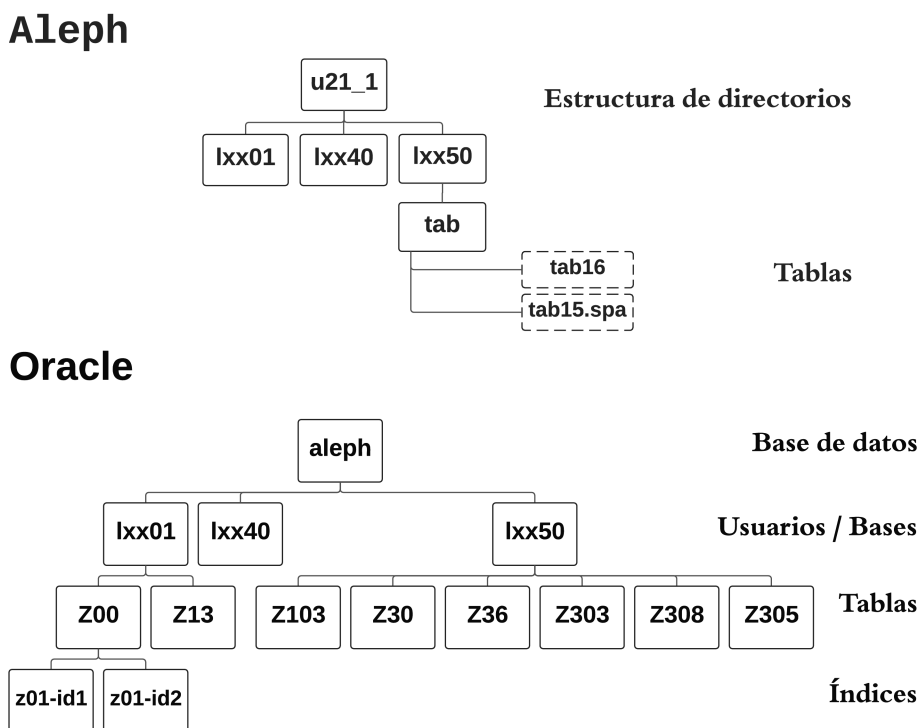


Figura 2.4: Estructura de directorios y base de datos de *Aleph*.

La base de datos de *Aleph* está constituida por más de un centenar de tablas que no serán mencionadas en su totalidad (Debido a que excede el

alcance de este trabajo), se enlistarán en las secciones posteriores aquellas de importancia para el sistema en cuestión.

Para el análisis de cada sistema es necesario estudiar los diagramas E-R de la base de datos (*Entity relationship diagrams* [18]) del sistema ***Aleph***, la relación entre las tablas (*Oracle table relationships* [19]) y el diccionario de datos (*Collected Oracle tables* [17]).

El diagrama E-R está organizado según su funcionalidad, es decir, que se incluyen las tablas de la base de datos que están relacionadas con una función en común, de los cuales para los sistemas contenidos en este trabajo se utilizaron únicamente *Circulación y Ejemplares* y *Usuarios*, mismos que se encuentra en el Apéndice D. Las relaciones que se analizaron son *USM01*, *USM50 (ADM/Item)* y *USM50 (Circ)* incluidas en el Apéndice C.

Cabe destacar que los nombres de las tablas y sus atributos no serán modificados, es decir, serán mencionados tal cual como están identificados por el sistema ***Aleph***, esto con la finalidad de ser congruentes con los fragmentos de código incluidos en este trabajo y las partes de la documentación utilizada en los apéndices.

En este capítulo se presenta la arquitectura del sistema ***Aleph***, examinando los detalles de la base de datos y la nomenclatura sugerida para identificar los tipos de bibliotecas. Se mencionaron los módulos a través de los cuales se administran los servicios de una biblioteca que integran a dicho sistema y se señaló la importancia del modulo *Circulación* para este trabajo.

Capítulo 3

Tecnologías utilizadas

A continuación se mencionan las tecnologías relacionadas y utilizadas en este trabajo, por un lado se menciona y describe el Lenguaje Unificado de Modelado (UML), utilizado para documentar los sistemas de software en este documento. Por otro lado se presentan aquellas tecnologías utilizadas en el desarrollo de los sistemas, Java como el lenguaje de programación y NetBeans como el IDE seleccionado. La base de datos del sistema *Aleph* consultada por los sistemas está en Oracle, por lo tanto, se menciona de forma breve.

3.1. Lenguaje Unificado de Modelado (UML)

En 1994 el Lenguaje Unificado de Modelado (UML¹), fue el resultado de la unión de las técnicas más populares de análisis y diseño orientadas a objetos, la *Técnica de Modelado de Objetos* (OMT²) de Rumbaugh, el método *Grady Booch* y la metodología *Objectory* de Jacobson [22].

Cuando se publicó una versión preliminar de su trabajo, no se había desarrollado una nueva tecnología sino apenas una notación para representar un sistema de información orientado a objetos.

¹ Por sus siglas en inglés UML (*Unified Modeling Language*) [22].

² Por sus siglas en inglés OMT (*Object Modeling Technique*) [22].

La versión 1.0 de UML, tomó por asalto al mundo de los sistemas de información en 1997 y muy pronto se utilizó en todo el mundo. OMG (*Object Management Group*) organizó un estándar internacional para UML que actualmente es la notación estándar internacional indiscutible para representar el comportamiento de dichos sistemas [22].

El proceso unificado³ es una técnica para elaborar diagramas UML que representan uno o más aspectos de algún sistema de información que se desea desarrollar. Es una metodología iterativa y por incrementos, es decir, los pasos se repiten hasta obtener el modelo UML exacto del sistema que se desea desarrollar.

La razón para utilizar una representación gráfica es que permite comunicar de manera más rápida y precisa la estructura o el comportamiento de un sistema, en vez de utilizar solo descripciones verbales, por lo que UML es la herramienta que se utiliza para representar (modelar) un sistema de información de esta manera.

Existen muchos tipos de diagramas UML con usos específicos, la mayoría de ellos sirven para ir diseñando un sistema, la Figura 3.1 los enlista y señala la categoría a la que pertenecen; a continuación se mencionan los más utilizados para describir sistemas.

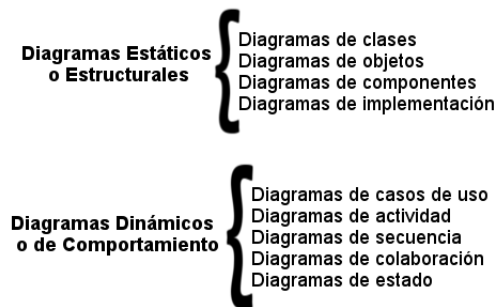


Figura 3.1: Tipos y clasificación de diagramas UML [26].

³ Es una metodología completa de análisis y diseño orientado a objetos, que en sus inicios se llamó proceso unificado de desarrollo de software **USDP** (*Unified Software Development Process*) [22]

3.1.1. Diagrama de caso de uso

Los diagramas de caso de uso modelan la integración entre el sistema y los usuarios de éste (actores⁴), es decir, son las funciones o servicios que provee el sistema a los usuarios [22].

La notación utilizada puede ser consultada en el Apéndice B.

3.1.2. Diagramas de actividad

Se utilizan para describir un sistema entero, un caso de uso o una actividad dentro de un caso de uso. Para mostrar cómo se construyen los diferentes flujos de trabajo o los procesos dentro de un sistema, cómo se inician, los caminos alternativos que se pueden tomar desde el principio hasta el fin [23].

También es capaz de ilustrar en que momento ocurrirá un procesamiento en paralelo durante la ejecución de algunas actividades, donde no importa en qué orden sean invocadas, es decir, pueden ser ejecutadas simultáneamente o una detrás de otra.

Los diagramas de actividad pueden incluir particiones o calles, donde se muestra quién hace qué. La notación utilizada puede ser consultada en el Apéndice B.

3.1.3. Diagramas de clases

Los diagramas de clases tiene como finalidad describir la estructura del sistema mostrando los elementos que existen, por ejemplo las clases, atributos y relaciones. Existen tres tipos de clases [22]:

Clase entidad: es un modelo de la información perdurable, elementos

⁴ Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.[22]

normalmente almacenados en el negocio por un largo periodo, por ejemplo: *Libro*, *Usuario*.

Clase borde: modela la interacción entre el sistema y sus actores. Las clases borde, por lo general, se asocian con la entrada y la salida de información, por ejemplo, si el sistema debe imprimir informes, sugiere que la clase borde *Informes* debe existir.

Los sistemas deben estar lo más aislados posible de cambios en otros sistemas. Generalmente se asigna una clase borde a cada actor (sistema externo o actor humano), también son conocidas como clases límite.

Clase control: es un modelo para los cálculos y algoritmos que en general son fáciles de extraer, cada cálculo no trivial se modela mediante una clase de control.

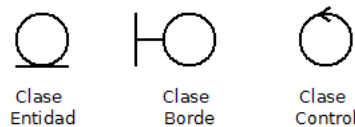


Figura 3.2: Estereotipos del UML para representar tipos de clases [22].

Los estereotipos de la Figura 3.2 son utilizados en un diagrama de clases para representar las clases, su tipo y como están relacionadas, de manera que sean comprensibles para cualquier persona.

Otros diagramas de clases (más técnicos) incluye detalles de la implementación, muestran los atributos, métodos y las relaciones entre las clases.

Las clases se pueden agrupar en unidades lógicas o paquetes de elementos relacionados entre sí. La notación utilizada puede ser consultada en el Apéndice B.

3.1.4. Diagramas de secuencia

Los diagramas de secuencia representan la realización de un caso de uso, muestra la interacción entre los usuarios, pantallas, objetos y entidades dentro del sistema; describen cómo el usuario interactúa con el sistema y que

pasa internamente para realizar el trabajo, es un mapa secuencial con mensajes que pasan entre objetos a través del tiempo. La notación utilizada puede ser consultada en el Apéndice B.

3.2. Lenguaje de programación (Java)

Java se ha convertido en el lenguaje más utilizado por la comunidad mundial de programadores y por la mayoría de los grandes fabricantes de software. Es el lenguaje que más impacto ha tenido en los últimos años, especialmente en el desarrollo para sitios Web, a su vez, se ha impuesto en la programación para dispositivos móviles [29].

3.2.1. Características de Java

Como lenguaje de programación Java no se diferencia mucho del resto de los lenguajes de programación Orientados a Objetos, sin embargo, las siguientes características lo hacen especial.

”Disponibilidad de un amplio conjunto de bibliotecas: la programación de aplicaciones con Java está basada no sólo en el empleo de instrucciones que componen el lenguaje, sino en la posibilidad de utilizar el gran número de clases disponibles para realizar casi cualquier tipo de aplicación. Por ejemplo clases para interfaces gráficas, gestión de redes, multitarea, acceso a datos, etcetera” [27].

”Aplicaciones multiplataforma: el código puede ser ejecutado en diferentes sistemas operativos sin necesidad de realizar cambios al código fuente y sin volver a compilar el programa” [27].

”Ejecución segura de aplicaciones: la seguridad de las aplicaciones Java se presenta en varios aspectos, por ejemplo, el lenguaje carece de instrucciones que realicen accesos descontrolados a la memoria (apuntadores en C o C++), es decir la máquina virtual impone ciertas restricciones para garantizar una ejecución segura” [27].

”Soporte de fabricantes de software: se encuentra una amplia variedad de software de diferentes fabricantes que da soporte a Java, como es el caso de los entornos de desarrollo (Sección 3.2.5) o los servidores de aplicaciones” [27].

3.2.2. Máquina Virtual de Java

La Máquina Virtual de Java (JVM⁵) es un entorno de ejecución para las aplicaciones Java, cuya principal finalidad es la de adaptar los programas compilados a las características del sistema operativo donde se va a ejecutar.

Un programa en Java esta organizado en clases que se codifican en archivos de texto con extensión *.java*, cuando son compilados se generan archivos *.class* de código binario denominados *bytecodes*, los cuales son independientes de la arquitectura. Es durante la fase de la ejecución que la máquina virtual de Java traduce los *bytecodes* a código ejecutable por el sistema operativo [27]. La Figura 3.3 muestra el esquema del proceso de compilación y ejecución de aplicaciones Java.

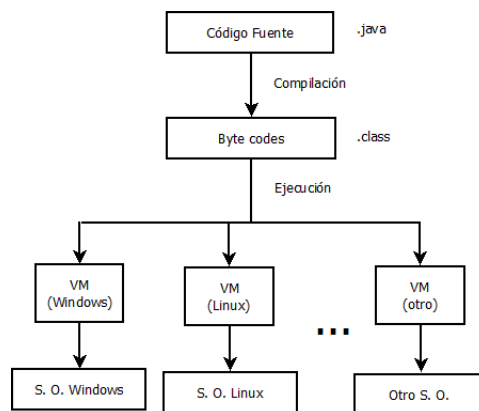


Figura 3.3: Proceso de compilación y ejecución para aplicaciones Java [27].

⁵ Por sus siglas en inglés JVM (*Java Virtual Machine*) [27]

3.2.3. Ediciones de Java

El amplio conjunto de bibliotecas antes mencionado, está organizado en tres grupos conocidos como ediciones de Java, las cuales son [27]:

”Java Standard Edition (Java SE): integrado por las clases de uso general, como son el manejo de cadenas, colecciones y acceso a datos. También incluye paquetes de clases para crear entornos gráficos y aplicaciones para navegadores de Internet.

Java SE permite desarrollar aplicaciones Java para computadoras de escritorio y servidores, ofrece interfaces de usuario, desempeño, versatilidad, potabilidad y seguridad que las aplicaciones actuales necesitan” [30].

”Java Enterprise Edition (Java EE): proporciona los paquetes necesarios para crear aplicaciones empresariales multicapa, entre ellas, las aplicaciones que se van a ejecutar en algún entorno Web” [30].

”Java Micro Edition (Java ME): incluye paquetes y especificaciones para crear aplicaciones para dispositivos electrónicos de capacidades limitadas como los dispositivos móviles” [30].

3.2.4. JRE y JDK

Existen principalmente dos productos de software en la familia de Java SE, los cuales son, Java Runtime Environment (JRE) y Java Development Kit (JDK) [32].

Java Runtime Environment (JRE): proporciona únicamente el entorno de ejecución de las aplicaciones, como son, la Máquina Virtual de Java y las bibliotecas [27].

Java Development Kit (JDK): proporciona el conjunto de herramientas básico para el desarrollo de las aplicaciones Java, como son, compiladores y depuradores, además incluye el JRE [27].

3.2.5. Entornos de desarrollo para Java

Cuando un programa contiene un elevado número de líneas y/o clases resulta práctico utilizar un entorno de desarrollo integrado (IDE⁶). Un IDE proporciona elementos indispensables para la codificación, compilación, depuración y ejecución de programas dentro de un entorno gráfico amigable y fácil de utilizar.

”Los IDE para Java utilizan internamente las herramientas del JDK donde el programador en lugar de hacer uso de la consola para ejecutar comandos, lo hace a través de menús o iconos de la barras de herramientas [27]. La Figura 3.4 muestra el aspecto del IDE NetBeans” [27].

”Cuentan con un editor de texto que resalta las palabras reservadas del lenguaje para distinguirlas del resto del código, algunos permiten la auto-escritura de instrucciones” [27].

”Basan el concepto de proyecto como conjunto de clases que forman una aplicación y cuando se crea un proyecto brindan la posibilidad de elegir entre plantillas o tipos de proyectos” [27].

Existen numerosos IDE para desarrollar aplicaciones Java, por ejemplo, BlueJ, DrJava, Eclipse, Greenfoot, IntelliJ, JDeveloper, jGRASP, NetBeans [27], este último es utilizado en los proyectos de este trabajo.

3.3. Entorno de desarrollo integrado (NetBeans)

NetBeans es un entorno de desarrollo integrado que permite de forma fácil y rápida el desarrollo de aplicaciones Java para escritorio, Web y móviles.

Es *open source*⁷, es decir, de código abierto y con una comunidad numerosa de usuarios y programadores alrededor de todo el mundo que pueden

⁶ Por sus siglas en inglés IDE (*Integrated Development Environment*).

⁷ En español código abierto.

aportar mejoras a las ya desarrolladas. Netbeans es el IDE oficial de Java y algunas de las características más importantes son [36]:

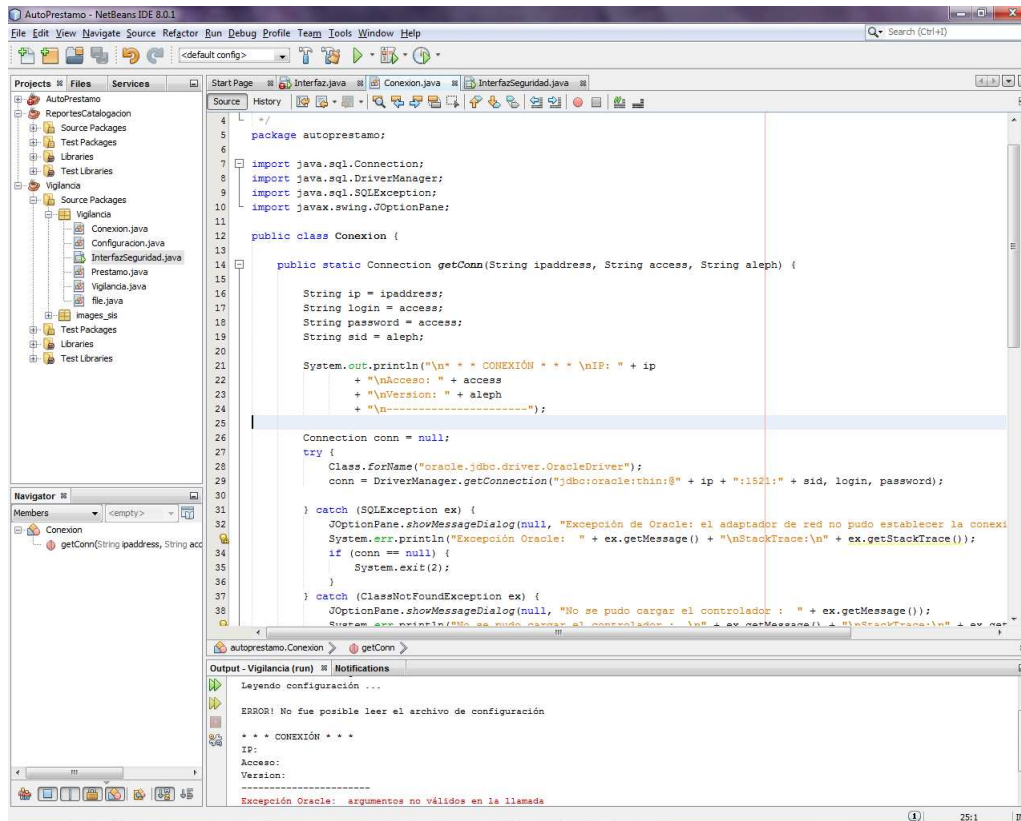


Figura 3.4: Aspecto de NetBeans.

Apoyo para actualización a la última versión de Java: sus editores, analizadores de código y convertidores puede rápidamente actualizar las aplicaciones con el objetivo de utilizar las nuevas construcciones de la última versión del lenguaje Java.

Edición de código rápido e inteligente: NetBeans justifica líneas de código, empareja paréntesis y corchetes, destaca código fuente tanto sintáctica como semánticamente, además de auto-completar las instrucciones. También proporciona plantillas de código, consejos de codificación y herramientas de refactorización. La Figura 3.5 muestra la

auto-escritura de instrucciones y algunos de los colores con los que se destacan las palabras reservadas y otras secciones del código.

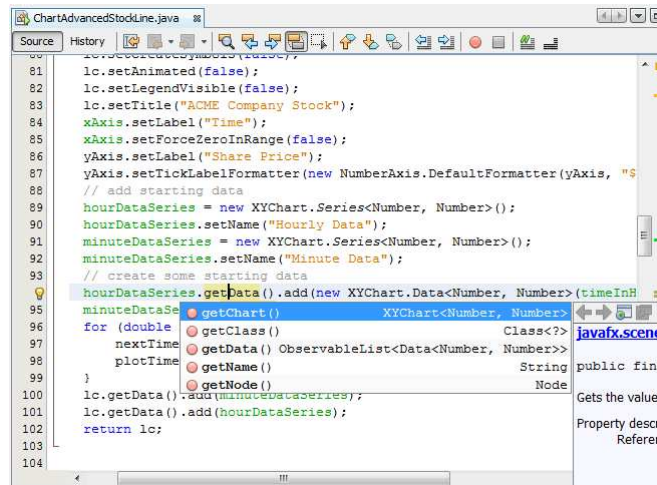


Figura 3.5: Edición de código en NetBeans [36].

Administración de proyectos fácil y eficiente: NetBeans proporciona diferentes vistas de los datos desde múltiples ventanas que incluyen útiles herramientas para crear y manejar las aplicaciones de manera eficiente. Esto permite el acceso a los datos de forma rápida y sencilla. La Figura 3.6 muestra las vistas de proyectos, archivos y favoritos.

Rápido diseño de interfaces de usuario: diseñar interfaces gráficas para usuarios (GUI) de aplicaciones utilizando editores y herramientas *drag-and-drop*⁸. Para las aplicaciones Java SE el constructor se encarga del espaciado correcto y alineación. Es tan fácil de usar e intuitivo que se ha utilizado para crear prototipos de interfaces gráficas de usuarios durante presentaciones en vivo con clientes.

Esta es la característica determinante por la cual se eligió NetBeans como el IDE para el desarrollo de los sistemas en este trabajo, es decir,

⁸ En español arrastrar-y-soltar, una vez que se selecciona el elemento se arrastra hasta soltarlo en la posición deseada observando el resultado en conjunto.

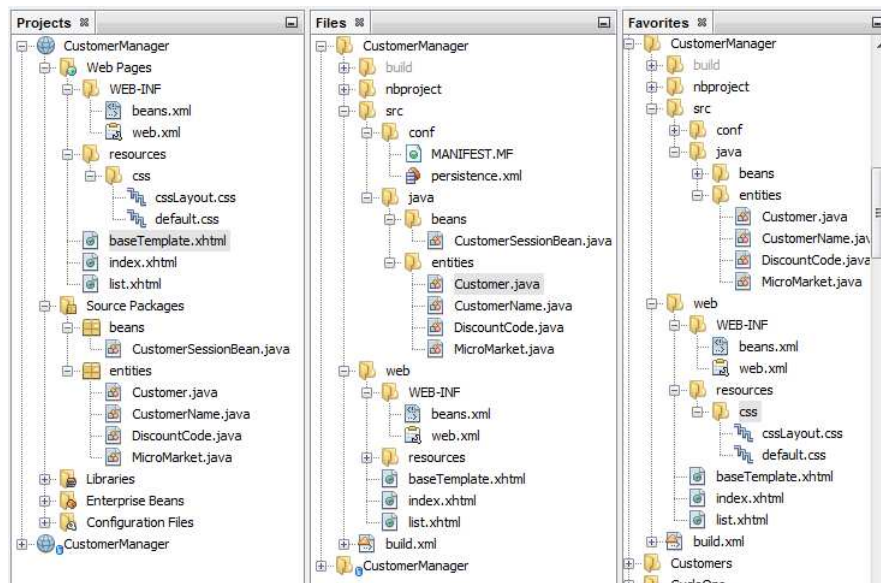


Figura 3.6: Vistas de NetBeans para la administración de los proyectos [36].

las interfaces fueron desarrolladas con él. En la Figura 3.7 se observan algunos de los elementos que proporciona NetBeans para el diseño de interfaces y el diseño de la interfaz del sistema *Vigilancia*.

Soporte para diversos lenguajes: ofrece apoyo a los programadores de C/C++ y PHP, consta de editores y herramientas compatibles con sus tecnologías y marcos de trabajo⁹, también incluye editores y herramientas para XML, HTML, Groovy, Javadoc, JavaScript y JSP.

Multiplataforma: NetBeans puede ser instalado en cualquier sistema operativo que soporte Java como una aplicación propia, por ejemplo Windows, Linux y Mac OS X, esto debido a que está escrito también en Java.

Un amplio conjunto de extensiones: como NetBeans es un IDE extensible y tiene API bien documentadas la comunidad crea extensiones¹⁰ para mejorar este IDE.

⁹ En inglés *frameworks*.

¹⁰ En inglés *plug-in*.

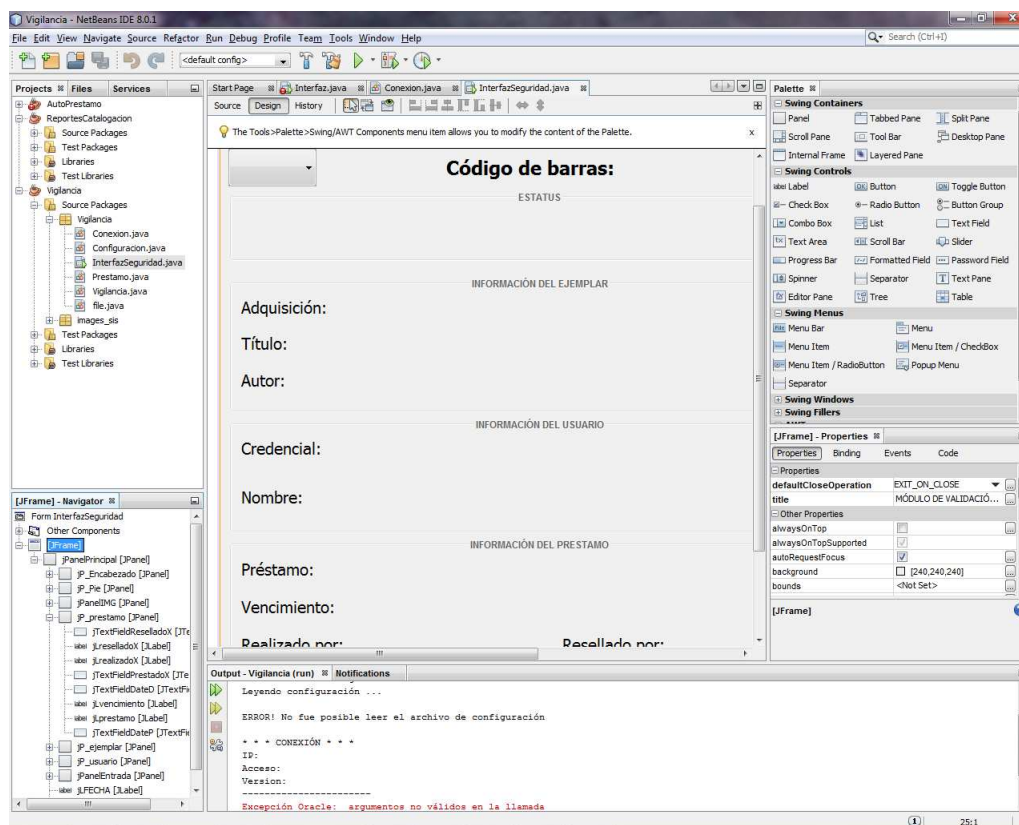


Figura 3.7: Diseño de GUIs en NetBeans.

3.4. Sistema manejador de bases de datos (Oracle)

”Un sistema manejador de bases de datos (SGBD¹¹) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos” [39], en otras palabras, facilita definir, construir, manipular y compartir datos, cuyo objetivo principal es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera tanto práctica como eficiente.

Actualmente las aplicaciones de dichos sistemas son numerosas, por men-

¹¹ En inglés DBMS (*Database Management Systems*).

cionar algunas están, la banca, líneas aéreas, universidades, telecomunicaciones, finanzas, comercio, producción, recursos humanos y bibliotecas.

Algunas de las ventajas de los SGBD son la independencia con respecto a los datos y su administración, así mismo, la integridad, seguridad y el acceso eficiente a éstos. Además el acceso concurrente, la recuperación en caso de fallo y la reducción de tiempo en el desarrollo de aplicaciones.

Los datos de una base de datos se almacenan en tablas, éstas están definidas por columnas que poseen un nombre¹². Los datos se almacenan como filas de la tabla. La Figura 3.8 muestra la estructura de una tabla.

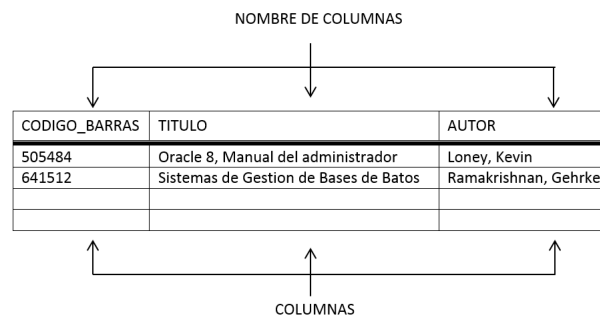


Figura 3.8: Representación gráfica de una tabla.

Los SGBD proporcionan un lenguaje de bases de datos conformado de un lenguaje de definición de datos (LDD¹³) para especificar el esquema de la base de datos y un lenguaje de manipulación de datos (LMD¹⁴) para expresar las consultas y las modificaciones a la base de datos. Un ejemplo de lenguaje de bases de datos es SQL¹⁵, el cual se ha establecido como lenguaje estándar para las bases de datos relacionales [39].

El LMD se encarga de la recuperación de la información almacenada en la base de datos, la inserción, el borrado y la modificación de la información,

¹² En términos de bases de datos son conocidos como atributos o campos.

¹³ En inglés DDL (*Data Definition Language*).

¹⁴ En inglés DML (*Data Manipulation Language*).

¹⁵ Por sus siglas en inglés SQL (*Structured Query Language*).

mientras que el LDD es aquel con el que son definidos los esquemas de la base de datos, se especifican las propiedades de los datos y restricciones de seguridad.

Una base de datos es un conjunto de datos y Oracle ofrece la posibilidad de almacenarlos y acceder a ellos de una forma coherente con un modelo conocido y definido como Modelo Relacional¹⁶, debido a esto, Oracle se conoce como un sistema de gestión de bases de datos relacionales (SGBDR¹⁷) [37].

Es considerado entre los mejores y más utilizados manejadores de base de datos por su robustez, seguridad, manejo de distintos niveles de seguridad, flexibilidad y estabilidad (resistencia), además de su gran capacidad de almacenamiento y amplio soporte [39].

Además de las herramientas relacionadas con la gestión y desarrollo de bases de datos Oracle ofrece algunas herramientas de inteligencia de negocios, productos de minería de datos, servidor de aplicaciones entre otros [39].

Lo anterior nos brinda un breve panorama de Oracle como manejador de bases de datos, pues los sistemas objeto de estudio de este trabajo no involucran ningún diseño de base de datos, debido a que solo se realizan consultas utilizando el LMD de SQL sobre la base de datos que pertenece al sistema *Aleph*.

En este capítulo se mostraron las herramientas utilizadas en el desarrollo de las aplicaciones de este trabajo; Oracle al ser el manejador de bases de datos del sistema *Aleph*, el lenguaje de programación Java seleccionado debido a su característica multiplataforma, y por la facilidad en el diseño y desarrollo de las interfaces se usó Netbeans. Por otro lado UML por ser el más utilizado en el análisis del desarrollo de software y la documentación de sistemas y procesos.

¹⁶ Conjunto de tablas, donde cada tabla es una entidad con columnas denominadas atributos y filas como registros.

¹⁷ En inglés RDBMS (*Relational Database Management System*).



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

S I S T E M A S

P A R A

B I B L I O T E C A S

U N A M



Capítulo 4

Consideraciones generales

Entre las actividades del Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas (DADMS) las relacionadas con los sistemas de este trabajo se encuentran, configurar¹ el sistema requerido para cada dependencia/biblioteca del SIBIUNAM que lo solicita, actualizar cada uno de los sistemas para las nuevas versiones del sistema operativo de la máquina² donde pueda ser instalado y lo más importante, para las nuevas versiones del sistema *Aleph*³ en el servidor. Por otro lado los responsables de las bibliotecas en ocasiones solicitan características particulares que pueden ser visuales o funcionales.

El sistema *Aleph* es un sistema comercial que implica el uso de licencias con costo, por lo tanto, cabe mencionar que los sistemas en este trabajo resuelven sus tareas realizando dicho ahorro económico y en el caso del sistema *Productividad en Catalogación* se brinda un servicio que no es proporcionado.

En general los sistemas en este trabajo son configurables a partir de un archivo de texto fuera del código fuente, se ha reorganizado el código basándose

¹ Esto se hacía en un principio dentro del código fuente.

² Windows (XP Profesional, Vista, 8 y 10) y Ubuntu (10.10,11.04,16.04).

³ Las versiones del sistema *Aleph* en las que se han implementado los sistemas de este trabajo son la versión 16, 21 y próximamente versión 23.

en el paradigma de programación Modelo-Vista-Controlador (MVC)⁴ y aunque se tienen más archivos y líneas de código, dicha organización simplifica su actualización y mantenimiento.

A continuación se mencionan las consideraciones que son comunes a los tres sistemas en este trabajo derivadas a partir de su relación con el sistema **Aleph**, y posteriormente en los capítulos correspondientes a cada uno de los sistemas se mencionan sus propias características.

4.1. Dependencias y bibliotecas

Existen varias configuraciones para el sistema **Aleph** dependiendo de la dependencia universitaria que se trate, la más sencilla y común es aquella en donde tiene una sola biblioteca física, sin embargo, existen casos donde se tienen más de una. Por ejemplo la Facultad de Ingeniería tiene dos bibliotecas una para la Facultad de Ingeniería⁵ y otra para la División de Estudios Profesionales (Anexo)⁶.

En el primero de los casos solo se tiene una biblioteca **Aleph** administrativa (*lxx50*) y una bibliográfica (*lxx01*), en cambio, cuando se tienen dos bibliotecas, cada una de ellas tiene sus propias bibliotecas **Aleph**. Cuando esto sucede se define una de éstas como principal y en su biblioteca administrativa (*lxx50*) se coloca la información global de los usuarios que será común en todas (Sección 2.5). En la Figura 4.1 se ilustra esta situación considerando la biblioteca *A* como principal⁷.

Cada que una dependencia solicita alguno de los sistemas, es primordial analizar la configuración del sistema **Aleph** que tienen en base a las bibliotecas y de ser necesario entonces adaptar la aplicación, más adelante se mencionan algunos casos donde se realiza esta configuración.

⁴ Por sus siglas en inglés MVC (*Model-view-controller*).

⁵ Biblioteca Ing. Antonio Dovalí Jaime.

⁶ Biblioteca Mtro. Enrique Rivero Borrell.

⁷ Las tablas *lxy50.z308* y *lxy50.z303* no son utilizadas.

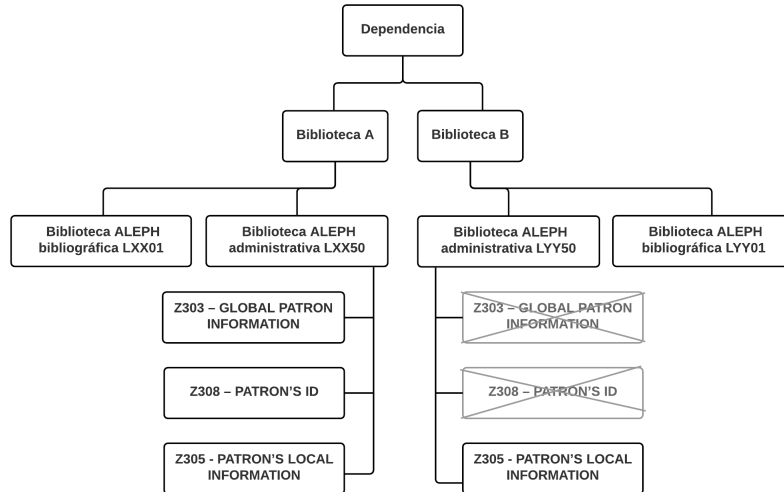


Figura 4.1: Estructura del sistema *Aleph* cuando existen dos bibliotecas.

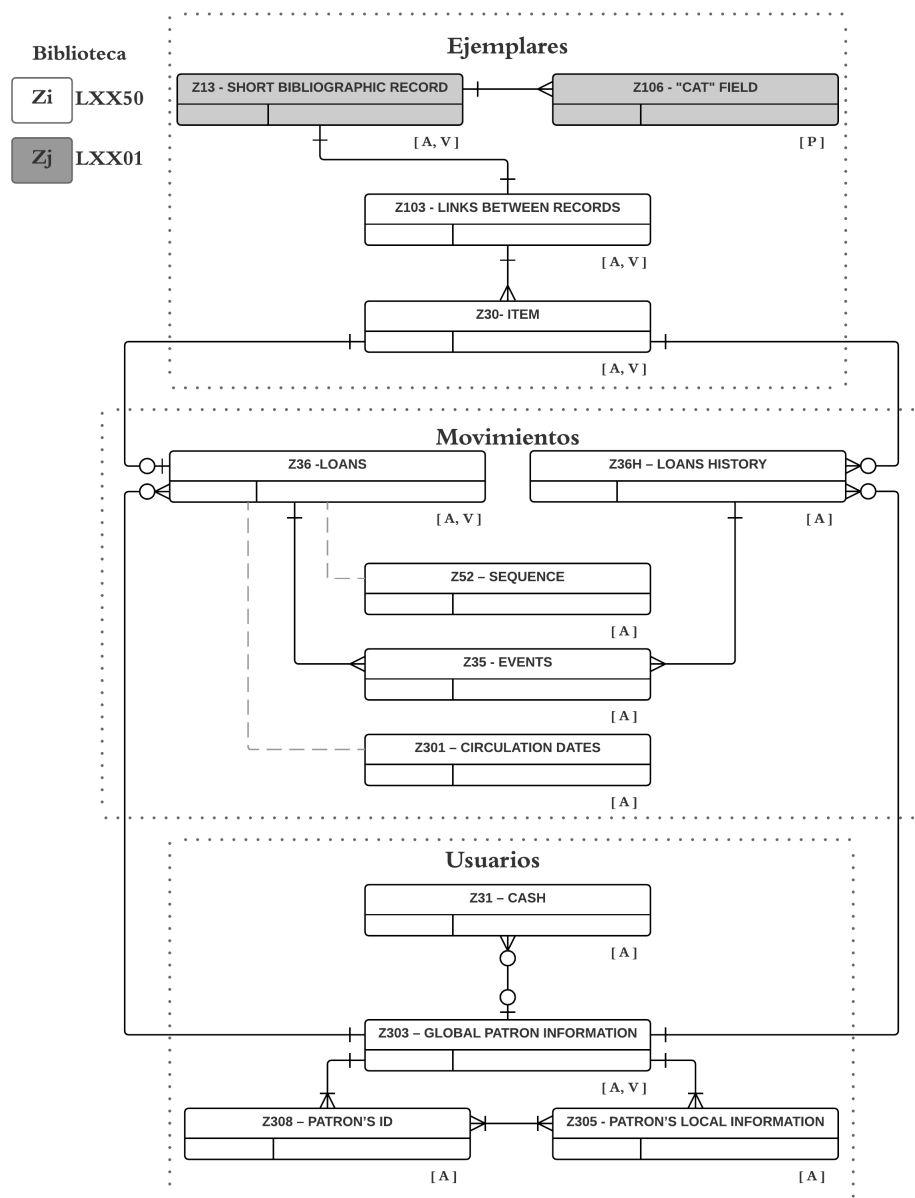
4.2. Base de datos

En la Sección 2.5 se hace mención a la estructura de la base de datos del sistema ALEPH, en este capítulo se muestra el diagrama E-R solo de las tablas relacionadas con los sistemas estudiados en este trabajo. Cabe mencionar que los nombres de las tablas y atributos no han sido modificados para ser congruentes con la documentación colocada en los apéndices a los que se hace referencia.

Los nombres de las tablas están formados por una “Z” y un número, cuando es agregada una letra “H” al final se trata del histórico de la tabla del mismo nombre. El nombre de los atributos tiene como prefijo el nombre de la tabla, lo cual les permite ser asociados de inmediato con la tabla a la que pertenecen, esto es práctico debido a que algunos nombres de los atributos existen en más de una tabla⁸.

A partir de los diagramas que muestran la relación entre tablas [19] (Ver Apéndice C) y los diagramas E-R de la documentación de *Aleph* [18] (Ver Apéndice D) se extrae el diagrama de la Figura 4.2 que muestra las tablas

⁸ Por ejemplo *xxxx_rec_key*.



Sistemas [A] Auto-Circulación [V] Valida-Préstamo [P] Productividad en Catalogación

Figura 4.2: Tablas utilizadas en los sistemas de este trabajo.

utilizadas y la relación entre ellas, señala si pertenecen a la biblioteca **Aleph** administrativa (*lxx50* en color blanco) o bibliográfica (*lxx01* en color gris). El identificador entre corchetes establece el sistema que utiliza dicha tabla, este puede ser “A” para **Auto-Circulación**, “V” para **Vigilancia** y “P” para **Productividad en Catalogación**. Además están agrupadas según la información contenida, si corresponde a los libros, usuarios o movimientos.

Los libros tienen asociados dos conceptos en el contexto de las bibliotecas de la UNAM para su administración, el de *título*⁹ y el de *ejemplar*¹⁰. Los títulos están almacenados en la base de datos bibliográfica en la tabla *Z13 (SHORT BIBLIOGRAPHIC RECORD)*, la cual tienen un identificador único (*z13_rec_key*) conocido como *Matriz* o *Número de sistema*, éste consta de una cadena de nueve números, existen tantas matrices como títulos. Esta tabla también contiene el título (*z13_title*) y el autor (*z13_author*) (Ver Figura 4.3).

Los libros de las bibliotecas se encuentran etiquetados con un código de barras (*z30_barcode*) que los identifica, este número está almacenado en la base de datos administrativa en la tabla *Z30 (ITEMS)*. Pueden existir varios ejemplares del mismo título donde el identificador de cada ejemplar (*z30_rec_key*) es único y se conforma del número de sistema y otra cadena conocida como *Secuencia*, integrada por seis números, múltiplos de 10 completados con ceros a la izquierda (Ver Figura 4.3).

La forma en que se relaciona la biblioteca administrativa con la bibliográfica (relación entre títulos y ejemplares) no es directa, desafortunadamente los números de sistema en la base de datos administrativa no siempre corresponden con el mismo número en la base de datos bibliográfica, a pesar de que existan en ambas tablas y estén definidos de la misma forma, es la tabla *Z103 (LINKS BETWEEN RECORDS)* la que contiene dicha relación. El número de sistema (*z13_rec_key*) en la biblioteca bibliográfica se obtiene a partir de la liga (*z103_link_doc_number*) registrada.

Para la tabla *Z106 (“CAT” FIELD)* el (*z106_rec_key*) si corresponde con (*z13_rec_key*). En la Figura 4.3 se observa como aparece almacenado un libro/ejemplar en cada una de las tablas. También se muestra como en algunos casos (*zxx-doc-number*) forma parte de otros campos o es nombrado con un

⁹ Les corresponde un número de sistema/matriz.

¹⁰ Referenciados como número de adquisición/código de barras.

nombre diferente en la tabla.

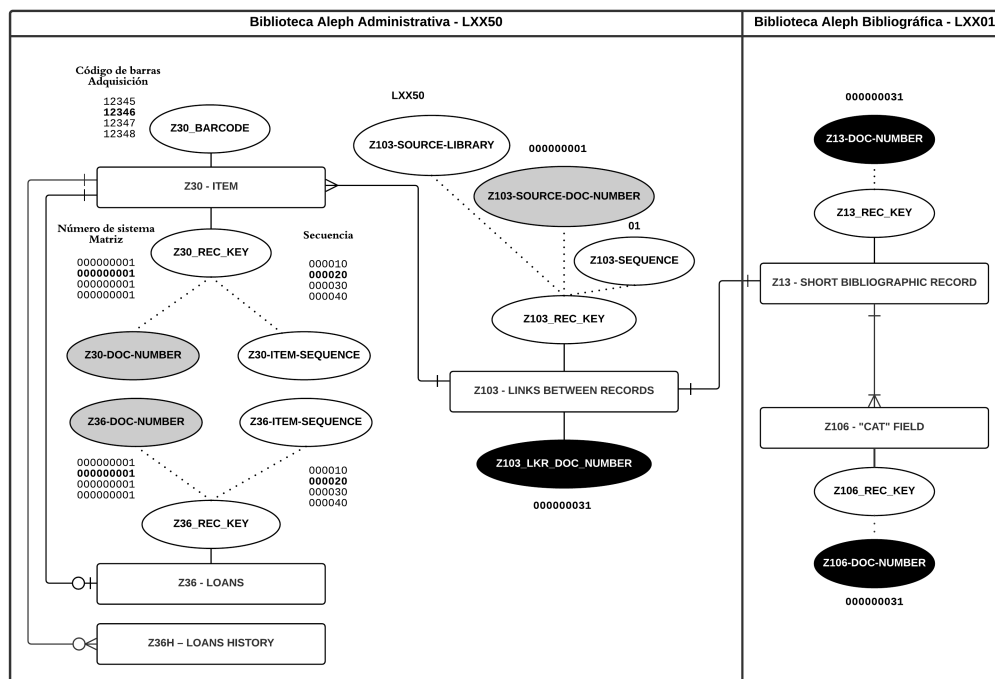


Figura 4.3: Libros en la base de datos *Aleph* (Títulos-Ejemplares).

Las tablas relacionadas con los usuarios que son consultadas son *Z303* (*GLOBAL PATRON INFORMATION*) que es el registro del usuario, es decir contiene los datos personales, *Z308* (*PATRON'S ID*) almacena información de acceso al sistema *Aleph* para las actividades que están disponibles para el usuario normalmente en el OPAC. Por ejemplo, para la renovación de préstamo vía web. *Z305* (*PATRON'S LOCAL INFORMATION*) es donde se registran los privilegios de un usuario como son la vigencia, permiso de préstamo y renovación, además de las multas o sanciones. En la Figura 4.4 se observa como las tres tablas están relacionadas a través del *xxxx.id*.

Los movimientos pueden ser préstamos, renovaciones o devoluciones. La información sobre los primeros dos se encuentra almacenada en la tabla *Z36*(*LOANS*), mientras que las devoluciones son registradas en *Z36H* (*LOANS HISTORY*). Además son registrados en la tabla *Z35* (*EVENTS*) que es una bitácora donde el campo *z35_event_type* identifica al préstamo con el valor 50,

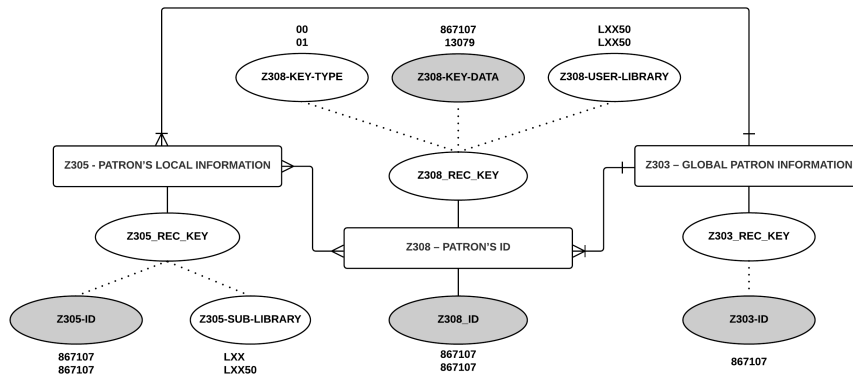


Figura 4.4: Usuarios en la base de datos *Aleph*.

63 para una renovación en PC y 61 si se trata de una devolución. Existen otros estados pero solo estos son considerados en el sistema *Auto-Circulación*.

Las tablas *Z52 (SEQUENCE)* y *Z301 (CIRCULATION DATES)* son utilizadas para generar datos necesarios al realizar movimiento, *Z52 (SEQUENCE)* controla el identificador de cada préstamo y *Z301 (CIRCULATION DATES)* para calcular la fecha de devolución.

Hasta el momento se han mencionado solo las tablas y la relación que existe entre ellas, más adelante en el capítulo que corresponde a cada sistema se detalla el diagrama E-R incluyendo sus atributos utilizados según su propio análisis de requerimientos. Cabe señalar que el sistema *Auto-Circulación* es el único que modifica, actualiza y elimina información en la base de datos, los demás solo realizan consultas.

4.3. Implementación e instalación

Los sistemas fueron programados en el lenguaje de programación Java (Sección 3.2) para que puedan ser instalado en equipos con sistema operativo Windows o Ubuntu de forma indistinta.

4.3.1. Casos de uso

La descripción de los casos de uso incluye la información de quien utiliza el caso de uso (*actores*), que soluciona el caso de uso (*objetivo*), que se necesita para efectuar el caso de uso (*precondiciones*), se indican las actividades que se realizan durante la ejecución normal del programa (*secuencia normal*), situaciones que no siguen el flujo establecido (*excepciones*), la condiciones después de efectuarse el caso de uso (*postcondiciones*) y en ocasiones se incluyen sentencias SQL¹¹.

4.3.2. Configuración

Todas las aplicaciones tienen como parámetros configurables la información de la dependencia/biblioteca y su servidor, estos datos están almacenados en un archivo llamado *dependencia.properties*¹².

Los parámetros utilizados en los tres sistemas son, la dirección IP del servidor de la dependencia/biblioteca, la base administrativa, las bases bibliográficas (generalmente solo es una), el nombre de la dependencia y la versión del sistema *Aleph*.

A continuación se muestra un ejemplo del archivo *dependencia.properties* que corresponde al sistema *Productividad en Catalogación*.

```
■ ip = 132.248.XXX.XXX
  version = 21
  baseAdm = LXX50
  basesBib = LXX
  nombre = Biblioteca Central
```

El sistema *Productividad en Catalogación* no necesita ningún otro

¹¹ Consultas a la base de datos, aquello que se indica entre los símbolos < y > especifica que es lo que debe de ir ahí, aunque en realidad se trata de una variable o el parámetro de algún objeto.

¹² Es una utilidad de configuración de Java para iniciar la aplicación, *Properties* son valores de configuración en pares Llave/Valor donde ambos elementos son cadenas (*String*). [35]

parámetro, sin embargo **Vigilancia** y **Auto-Circulación** utilizan un parámetro adicional para indicar al sistema si se deben eliminar los ceros a la izquierda del código de barras leído, esto debido a que algunas dependencias etiquetan los libros con ceros a la izquierda que no existen en la base de datos, este parámetro es *cb0i*¹³ y su valor puede ser *Y/N*. Donde *Y* indica que los códigos de barras si contienen ceros a la izquierda, mientras que *N* señala que no, siendo esta ultima la más común. Con la siguiente línea se completa el archivo de configuración para el sistema **Vigilancia**.

- `Cb0i = N`

Para habilitar los servicios que estarán disponibles en el sistema **Auto-Circulación** se utilizan *hab_pre* (para préstamo), *hab_ren* (para renovación) y *hab_dev* (para devolución), con el valor *Y* el servicio estará disponible mientras que *N* lo oculta de la interfaz. Además para identificar en el sistema **Aleph** que un movimiento fue realizado a través del sistema **Auto-Circulación** se le coloca el parámetro *catalogador=SAJi*¹⁴, completando el archivo de configuración para el sistema **Auto-Circulación** de la siguiente forma.

- `hab_pre = Y`
`hab_ren = Y`
`hab_dev = N`
`catalogador = SAJ`

El siguiente código pertenece a la clase *Configuracion* para el sistema **Productividad en Catalogación**, en las líneas 3 a 6 se incluyen las bibliotecas de Java para la manipulación de archivos y la utilidad para configuración *properties*. Como su nombre lo indica configura el sistema con la información contenida en el archivo *dependencia.properties* localizado a través de la función *localizarProperties* (líneas 12 a 28) que tiene definidas las ruta donde debe encontrarse si el sistema operativo es Windows¹⁵ o Ubuntu¹⁶.

¹³ *cb0i* por códigos de barras con ceros a la izquierda.

¹⁴ *SAJ* por **S**istema **A**uto-**C**irculación en **J**ava, e *i* que es un número consecutivo por equipo en que se encuentre instalado, si se desea puede ser omitido.

¹⁵ La ruta para Windows es *C:/DGB-UNAM-<Sistema>/dependencia.properties*.

¹⁶ La ruta para Ubuntu es */home/DGB-UNAM-<Sistema>/dependencia.properties*.

En caso de que el archivo sea encontrado como parte de la función *leerArchivoConf* los valores son guardados como atributos del objeto de esta clase con la instrucción *props.getProperty("<parametro>")*; (líneas 37 a 41) para después ser proporcionados a solicitud con el método *get<Atributo>* (líneas 48 a 50).

```

1 package unam.dgb.dadms.<sistema>.configuracion;
2
3 import java.util.Properties;
4 import java.io.File;
5 import java.io.FileInputStream;
6 import java.io.IOException;
7
8 public class Configuracion {
9
10     String ip = "", version = "", baseAdm = "", basesBib = "", nombre =
        "";
11
12     public String localizarProperties() {
13
14         String properties = "";
15         File f = null;
16
17         f = new File("/home/DGB-UNAM_Vigilancia/dependencia.properties");
18         if (f.exists()) {
19             properties = "/home/DGB-UNAM_Vigilancia/dependencia.properties";
20         } else {
21             f = new File("C:/DGB-UNAM_Vigilancia/dependencia.properties");
22             if (f.exists()) {
23                 properties = "C:/DGB-UNAM_Vigilancia/dependencia.properties";
24             }
25         }
26
27         return properties;
28     }
29
30     public void leerArchivoConfig() {
31         try {
32             String archivoConfig = localizarProperties();
33             Properties prop = new Properties();
34             FileInputStream arch = new FileInputStream(archivoConfig);
35             prop.load(arch);
36
37             ip = prop.getProperty("ip");
38             version = prop.getProperty("version");
39             baseAdm = prop.getProperty("baseAdm");
40             basesBib = prop.getProperty("basesBib");
41             nombre = prop.getProperty("nombre");
42         } catch (IOException e) {
43             System.out.print("No fue posible leer el archivo de
                configuracion");
44             System.exit(0);
45         }
46     }

```



```
47     //Existe un metodo get para cada atributo.
48     public String get<Atributo>() {
49         return <atributo>;
50     }
51 }
```

De esta forma al ser utilizados los sistemas en distintas bibliotecas o ser solicitados posteriormente por una dependencia/biblioteca distinta solo es necesario modificar el archivo de configuración.

4.3.3. Conexión

Java permite la conexión con la base de datos a través del JDBC ¹⁷ contenido en el paquete *java.sql* definido para cada manejador. Las librerías utilizadas en esta clase son *Connection*, *DriverManager* y *SQLException* (líneas 3 a 5).

Es en la clase *Conexion* donde se substituyen los datos de la configuración (línea 16) obtenidos con la clase del mismo nombre (Sección 4.3.2) y que recibe como parámetros (línea 10).

A través de su único método *getConn* se obtiene la conexión a la base de datos del sistema **Aleph** utilizando un controlador de Oracle¹⁸ que fue necesario descargar y probar; esto porque existen varios dependiendo de las versiones de Oracle, incluso puede existir más de uno por versión.

El código de la clase es el siguiente:

```
1 package unam.dgb.dadms.<sistema>.configuracion;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6 import javax.swing.JOptionPane;
7
8 public class Conexion {
9
```

¹⁷ Del inglés *Java Database Connectivity*.

¹⁸ En este caso se utiliza la librería *ojdbc14.jar*.

```

10  public static Connection getConn(String ipaddress, String access,
11      String aleph) {
12      String ip = ipaddress, login = access, password = access, version
          = aleph;
13      Connection conn = null;
14      try {
15          Class.forName("oracle.jdbc.driver.OracleDriver");
16          conn = DriverManager.getConnection("jdbc:oracle:thin:@" + ip +
          ":1521:" + version, login, password);
17      } catch (SQLException ex) {
18          JOptionPane.showMessageDialog(null, "No se pudo establecer la
          conexion.\n" + ex.getMessage());
19          if (conn == null) {
20              System.exit(0);
21          }
22      } catch (ClassNotFoundException ex) {
23          JOptionPane.showMessageDialog(null, "No se pudo cargar el
          controlador.\n" + ex.getMessage());
24      }
25      return conn;
26  }
27  }

```

4.3.4. Paquetes

Los sistemas que se mencionan en este trabajo se han ido transformando de manera que tengan la misma estructura, actualmente el código se encuentra organizado en paquetes. Los primeros están anidados de forma que en conjunto constituyen una ruta de identidad y propiedad del código, la cual es la siguiente *mx.unam.dgb.dadms.<nombre del sistema>*, el resto están contenidos dentro paquete que tiene por nombre el nombre del sistema (Ver Figura 4.5).

El paquete *configuración* tiene al menos las clases *Configuración* y *Conexión* que se explicaron en las Secciones 4.3.2 y 4.3.3 respectivamente.

El paquete *interfaz* contiene la clase del mismo nombre (*Interfaz*) donde se define la apariencia y comportamiento de la única pantalla con la que interactúa el usuario.

El paquete *objetos* contiene las clases de todo aquello que se ha identificado como un objeto utilizando la definición de su(s) constructor(es) y variables con sus métodos *.get()* y *.set()*.

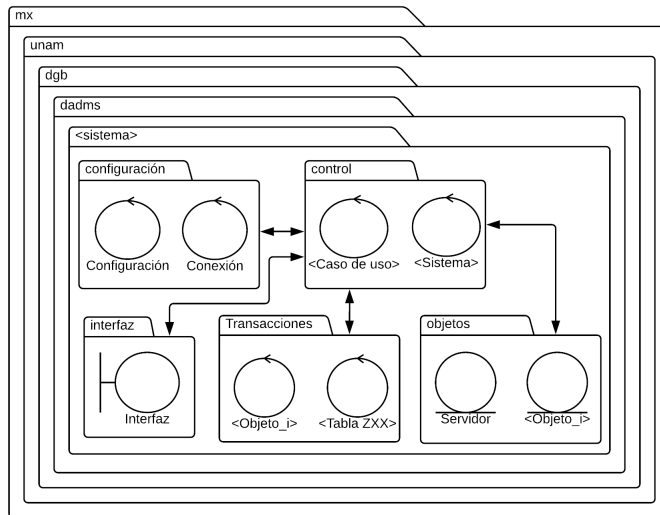


Figura 4.5: Diagrama de paquetes para los sistemas de las bibliotecas UNAM.

El paquete *transacciones* contiene las clases donde se realizan las consultas a la base de datos, agrupadas y nombradas según el objeto con el que se relacionan o la tabla que consultan.

El paquete *control* contiene las clases que coordinan la relación entre los paquetes interfaz, transacciones y objetos, pueden tener por nombre el nombre del sistema o por el caso de uso que resuelven.

4.3.5. Consultas y transacciones

Los sistemas solo tienen relación con la biblioteca **Aleph** bibliográfica y administrativa. Algunos nombres de tablas existen en ambas, por lo tanto en las consultas empleadas en los sistemas para cada campo se indica la tabla a la que pertenece y de está se hace referencia a la biblioteca **Aleph** correspondiente (*lxx50.z52.z52_rec_key*¹⁹). Una vez que se establece la conexión con el servidor se realicen todas las consultas requeridas, con la finalidad de reducir el tiempo de respuesta, haciendo más eficiente el uso del sistema.

¹⁹ <biblioteca aleph>.<tabla>.<campo>

El sistema *Auto-Circulación* que es el único que modifica la base de datos, por lo que las consultas SQL se realizan en una sola transacción, pues si ocurriera algún error se pueden revertir todos los cambios (*rollback*) evitando modificaciones aisladas que comprometan la base de datos afectando incluso la funcionalidad del sistema *Aleph*, y en caso de que no existiera ningún error se realizan todas las actualizaciones necesarias (*commit*).

4.3.5.1. Transacción en Java

A continuación se muestra un fragmento del código para registrar un préstamo, en el que se ilustra cómo se establece una transacción en el lenguaje de programación Java, entre las instrucciones `conn.setAutoCommit(false);` y `conn.commit();` (líneas 18 y 37) dentro de un bloque `try(){... }` se declaran, definen y ejecutan todas las consultas SQL necesarias y de ocurrir algún error en cualquiera de ellas, la instrucción `conn.rollback();` (línea 43) dentro del bloque `catch(){... }` `finally{... }` revierte todos los cambios. Dentro de el o los bloques `catch(){... }` se realiza el manejo de excepciones y envío de mensajes de error.

```

1 public void actualizarTablas(movimiento) throws SQLException {
2     String txt_insert_LXX50_Z36 = "",
3         txt_update_LXX50_Z52 = "",
4         txt_update_LXX50_Z30 = "",
5         txt_insert_LXX50_Z35 = "";
6     PreparedStatement stmt_insert_LXX50_Z36 = null;
7     PreparedStatement stmt_update_LXX50_Z52 = null;
8     PreparedStatement stmt_update_LXX50_Z30 = null;
9     PreparedStatement stmt_insert_LXX50_Z35 = null;
10
11     //Definicion de cada consulta txt_..._LXX50_Z... = ... Ejemplo:
12     txt_update_LXX50_Z52 = "UPDATE_LXX50.Z52_"
13         + "SET_LXX50.Z52.Z52_SEQUENCE_=?_"
14         + "WHERE_LXX50.Z52.Z52_REC_KEY_="
15         + "'last-loan-number'";
16     ...
17     try {
18         conn.setAutoCommit( false );
19
20         stmt_insert_LXX50_Z36
21             = conn.prepareStatement(txt_insert_LXX50_Z36);
22         ...
23         stmt_update_LXX50_Z52
24             = conn.prepareStatement(txt_update_LXX50_Z52);
25         stmt_update_LXX50_Z52.setInt(1, Integer.parseInt(z52_contador));
26         stmt_update_LXX50_Z30

```

```

27         = conn.prepareStatement(txt_update_LXX50_Z30);
28         ...
29         stmt_insert_LXX50_Z35
30         = conn.prepareStatement(txt_insert_LXX50_Z35);
31         ...
32         stmt_insert_LXX50_Z36.execute();
33         stmt_update_LXX50_Z52.execute();
34         stmt_update_LXX50_Z30.execute();
35         stmt_insert_LXX50_Z35.execute();
36
37         conn.commit();
38         valor_insercion = 1;
39     } catch (SQLException e) {
40
41         if (conn != null) {
42             try {
43                 conn.rollback();
44             } catch (SQLException excep) {
45                 System.err.println("Error: " + e.getMessage(excep));
46             }
47         }
48
49         System.err.println("Exception: " + e.getMessage());
50         JOptionPane.showMessageDialog(null, "Fallo al conectar con la
51         base de datos.\nNo se realizo el prestamo", "JDBC_Error!",
52         JOptionPane.ERROR_MESSAGE);
53     } finally {
54         if (stmt_insert_LXX50_Z36 != null) {
55             stmt_insert_LXX50_Z36.close();
56         }
57         if (stmt_update_LXX50_Z52 != null) {
58             stmt_update_LXX50_Z52.close();
59         }
60         if (stmt_update_LXX50_Z30 != null) {
61             stmt_update_LXX50_Z30.close();
62         }
63         if (stmt_insert_LXX50_Z35 != null) {
64             stmt_insert_LXX50_Z35.close();
65         }
66         conn.setAutoCommit(true);
67     }

```

4.3.6. Interfaz

La interfaz consta de una sola pantalla donde se realiza la interacción entre el usuario, el cual introduce los datos²⁰ necesarios y el sistema que

²⁰ Estos datos dependen del sistema, por ejemplo para *Auto-Circulación* son el número de credencial/cuenta/empleado, contraseña y código de barras, para vigilancia solo este

genera una respuesta y entrega la información solicitada. Están diseñadas y desarrolladas utilizando el IDE NetBeans (Sección 3.3) por facilitar su composición arrastrando los elementos seleccionados mientras permite observar el resultado final.

La interfaz está identificada con el nombre del sistema y de la dependencia/biblioteca, el logotipo de la UNAM y de la DGB, los colores utilizados son los colores institucionales (azul y oro) en diversos tonos, blanco, gris y negro.

4.3.7. Requerimientos de hardware

Todos los sistemas son aplicaciones de escritorio utilizadas en una computadora personal estándar, las características que se muestran en el Cuadro 4.1 corresponden al equipo en que se ejecutaban en sus primeras versiones, cualquier otro equipo con especificaciones superiores garantiza que las aplicaciones funcionarán.

Procesador	Intel Core 2 Dúo
Memoria RAM	2 Gb
Capacidad en Disco Duro	250 Gb
Cantidad de puertos USB	5 (Cinco)
Tarjeta de red	128 Kbps

Cuadro 4.1: Características mínimas del CPU.

En el caso de los sistemas *Vigilancia* y *Auto-Circulación* el equipo es exclusivo de esa aplicación por ser aplicaciones prácticamente automatizadas y requieren del apoyo de dispositivos externos²¹ adicionales. El sistema *Vigilancia* necesita de un lector de código de barras y un teclado numérico y en el caso del sistema *Auto-Circulación* es indispensable un monitor táctil, una impresora térmica, un lector de código de barras y un mueble especial (Ver Cuadro 4.2).

último. Para *Productividad en Catalogación* un periodo de tiempo y la base bibliográfica.

²¹ Con esto se hace referencia al hardware.

<i>Vigilancia</i>	<i>Auto-Circulación</i>
	Lector de código de barras
Teclado numérico	Impresora térmica
	Monitor táctil
	Mueble especial

Cuadro 4.2: Dispositivos externos adicionales (hardware).

4.3.8. Requerimientos de software

La principal razón por la que la implementación de los sistemas se ha realizado en Java es porque los sistemas *Auto-Circulación* y *Vigilancia* requieren ser instalados en equipos con sistema operativo Windows o Ubuntu y aunque la sugerencia es que sea en Ubuntu por cuestiones de seguridad y costo, actualmente se encuentran operando en ambos, la última versión en que se encuentran funcionando es 16.04 para Ubuntu y Windows 10. Para poder ejecutar alguna de las aplicaciones debe contar con la instalación de la Máquina Virtual de Java (JVM) versión 6 (*jre-6u23*) o posterior.

Cada que surge una nueva versión del sistema *Aleph* alguno de los sistemas operativos o la Máquina Virtual de Java, se prueban las aplicaciones y de ser necesario se realizan las modificaciones pertinentes.

Auto-Circulación utiliza tablas de configuración²² y *Vigilancia* usa las fotografías de los usuarios que descargan del servidor antes de arrancar la aplicación, para esto deben tener instalado SSH (*Secure Shell - Client*) versión 3.2.9 o posterior y haber generado una llave para que no solicite la contraseña.

Los servidores del SIBIUNAM tienen implementada la seguridad a través de un *firewall*²³ e *ipfilter*²⁴ para el control de las peticiones de entrada y

²² Estas tablas de configuración contienen las políticas de préstamo, por ejemplo tiempo de préstamo, libros permitidos, etcétera. Todo esto dependiendo del estatus del usuario.

²³ Es un sistema que monitorea y controla las entradas y salidas en la red basado en la definición de reglas, es una especie de barrera entre una red de confianza normalmente local y otra, por ejemplo Internet.

²⁴ Es un programa integrado en algunos sistemas operativo Unix en el caso de los servidores de las bibliotecas UNAM, Solaris. Proporciona servicios de firewall y reasignación

salida solicitadas, por lo tanto los equipos de cómputo que tengan instalado alguno de los sistemas deben tener asignada una dirección IP con el mismo segmento de red que la dirección IP del servidor o de lo contrario la aplicación no funcionará. El Cuadro 4.3 resume los requerimientos mínimos de software mencionados anteriormente.

Sistema operativo	Windows XP Ubuntu 10.10
JRE	Versión 6
SSH Secure Shell – Client openssh-client	Versión 3.2.9

Cuadro 4.3: Requerimientos mínimos de software.

4.3.9. Instalación

La instalación de cualquiera de las aplicaciones en este trabajo consiste en:

- Copiar la carpeta que contiene la estructura de directorios necesaria por el sistema, el archivo ejecutable²⁵, las imágenes, el archivo de configuración y los *scripts* que se conectan al servidor para descargar los archivos y ejecutar la aplicación. Para el sistema operativo Windows la carpeta está ubicada en *C:* y para Ubuntu en *\home*.
- Generar una llave utilizando *SSH (Secure Shell-Client)* para que el servidor no solicite la contraseña por cada ocasión que se ejecuta la aplicación al descargar los documentos requeridos.
- Configurar el IPv4²⁶ con los valores permitidos por el servidor para la dirección IP, la máscara de subred, la puerta de enlace predeterminada

de direcciones de red (NAT *Network Address Translation*).

²⁵ Java genera un archivo comprimido *.jar* capaz de ser ejecutado por la máquina virtual de Java.

²⁶ El Protocolo de Internet versión 4 (IPv4 *Internet Protocol Version 4*) es la cuarta revisión del protocolo de Internet (IP) extensamente utilizado en la comunicación sobre diferentes tipos de redes.

y las direcciones del servidor DNS²⁷.

- Instalar y configurar dispositivos.
- Probar la aplicación.

En este capítulo se mencionan las consideraciones que son más relevantes para desarrollar los sistemas expuestos en este trabajo; entre las que se encuentran las tablas en las que se localiza la información de los usuarios, ejemplares y préstamos, así como la biblioteca a la que corresponden, ésta puede ser administrativa o bibliográfica. También se exponen detalles de la implementación para las interfaces y transacciones. Por último, se señalan los requerimientos de hardware, software y el procedimiento de instalación.

²⁷ Por sus siglas en inglés DNS (*Domain Name System*), el Sistema de Nombres de Dominios asocia información, por ejemplo direcciones IP con nombres de dominio.

Capítulo 5

Sistema para validación de préstamos: *Vigilancia*

El sistema *Vigilancia* informa si un libro puede salir de la biblioteca, o no, es decir, si el ejemplar se encuentra prestado y permite verificar si el usuario que tiene el libro físicamente es quien es el mismo que lo tiene registrado en el sistema *Aleph*.

La operación del sistema sólo requiere introducir el identificador del ejemplar a través de un lector de código de barras o un teclado numérico para que la información del préstamo sea visualizada en la pantalla.

5.1. Análisis de requerimientos

El sistema llamado *Vigilancia* es responsable de validar si un libro se encuentra prestado al verificar si existe el registro correspondiente y determina el estado del préstamo que puede ser *vigente*, *vencido* o *renovar*, entonces muestra en pantalla la información del movimiento, el ejemplar y el usuario para que se permita o niegue la salida del material de la biblioteca. Si no se encuentra el registro significa que no está prestado.

El sistema *Vigilancia* determina el estado de un préstamo de la siguiente

manera:

Vigente: en el cual la fecha de devolución es mayor a la fecha actual.

Renovar: donde la fecha de devolución es igual a la fecha actual.

Vencido: en el que la fecha de devolución es menor a la fecha actual.

Con base en el estado proporcionado por el sistema, el vigilante de la biblioteca debe retener los libros cuando es *vencido*, señalar que debe renovar el préstamo, advirtiéndole que podría generar una multa o sanción¹ si es *renovar* y autorizar la salida del ejemplar para *vigente* siempre y cuando el usuario registrado sea el mismo que lo extrae de la biblioteca.

La información que el sistema debe presentar al operador para ser verificada y permitir que el ejemplar salga de la biblioteca es:

Información del préstamo: estado del préstamo, fecha de préstamo, fecha de vencimiento, identificador de quién² realizó el préstamo y la renovación.

Información del usuario: nombre, número de cuenta o credencial y fotografía.

Información del ejemplar: código de barras, título y autor.

La Figura 5.1 presenta un diagrama de actividades que se realizan para validar un préstamo sin la ayuda del sistema *Vigilancia* se distribuyeron las actividades dependiendo del actor que las realiza. Por un lado se tiene un usuario de la biblioteca (quien solicita libros en préstamo de la biblioteca) y del otro el vigilante (usuario del sistema, empleado de la biblioteca). El usuario llega a la salida de la biblioteca y entrega el libro al vigilante, entonces este último revisa la papeleta colocada en el final del libro y verifica la fecha de entrega del libro, si faltan unos días más, autoriza la salida del ejemplar; si es la misma en que se está revisando, le indica que debe renovar el préstamo, y si ya pasó la fecha entonces se retiene el libro.

¹ Esto depende de las políticas de la biblioteca.

² Se puede encontrar al sistema *Auto-Circulación* como el responsable de realizar el movimiento (*SAJ*).

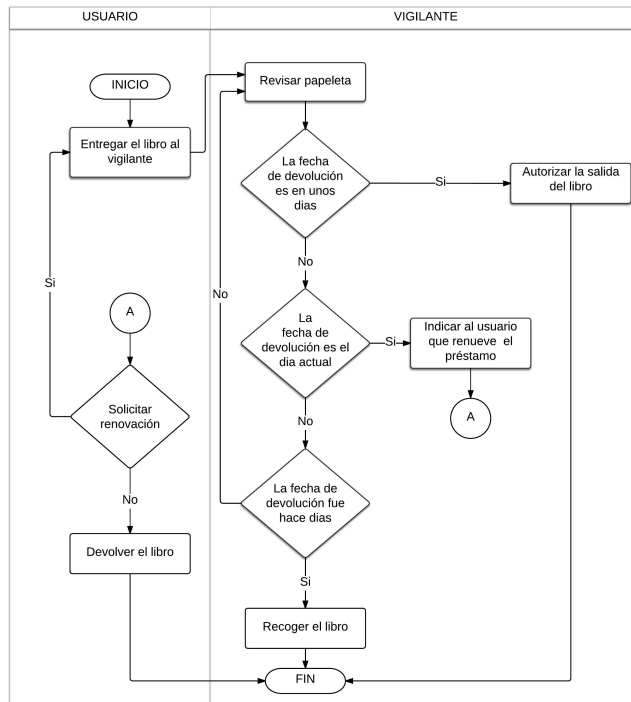


Figura 5.1: Diagrama de actividades para validar préstamo.

5.2. Diagrama E-R

La Figura 5.2 contiene solo aquellas tablas del diagrama E-R de la Figura 4.2 que utiliza el sistema *Vigilancia* incluyendo los campos que son necesarios consultar de cada una y también se puede observar aquellos por los que están relacionadas.

A continuación se menciona la información que se debe recuperar de la base de datos del sistema *Aleph*. Se obtiene a partir del código de barras (*z30_barcode*) impreso en el libro la fecha en que se realizó el préstamo (*z36_loan_date*), la fecha de devolución (*z36_due_date*), el identificador de quien realizó el préstamo (*z36_loan_cataloger_name*), el identificador de quien renovo el préstamo (*z36_renew_cataloger_name*), el identificador (*z36_id*) y nombre del usuario (*z303_name*), el título (*z13_title*) y autor (*z13_author*) del libro.

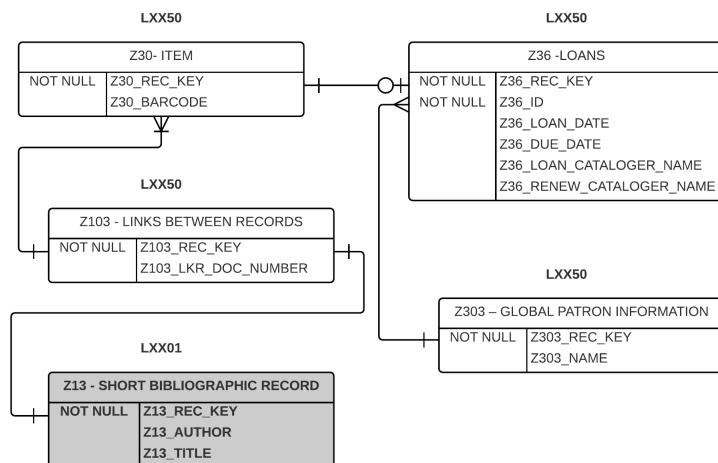


Figura 5.2: Diagrama E-R para Vigilancia.

5.3. Casos de uso

A continuación se describen los casos de uso del sistema según lo señalado en la Sección 4.3.1.

Para el sistema *Vigilancia* se tiene un solo caso de uso **Validar préstamo** (5.3.1), sin embargo incluye los casos de uso **Localizar préstamo** (5.3.2), **Localizar usuario** (5.3.3) y **Localizar ejemplar** (5.3.4) (Ver Figura 5.3); todos estos son explicados a continuación.

5.3.1. Validar préstamo

Identificador

CU No1

Actores

Personal de la biblioteca ubicado en la(s) salida(s) (Vigilante).

Descripción

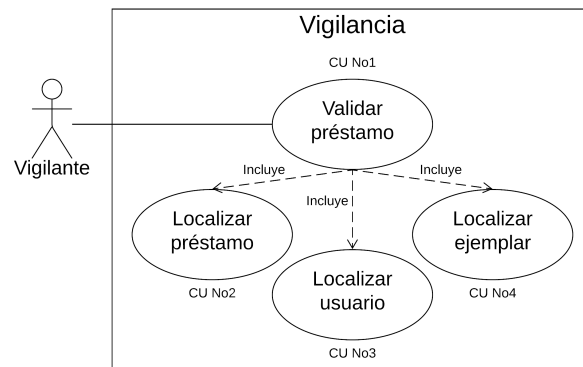


Figura 5.3: Casos de uso del sistema Vigilancia.

El sistema determina el estado de un préstamo, éste puede ser "Vigente", "Renovar" o "Vencido" a partir del código de barras de un libro y muestra la información del usuario, libro y préstamo a través de la interfaz.

Precondiciones

El ejemplar debe estar debidamente etiquetado para que el escáner pueda leerlo, o en su defecto introducir el código de barras mediante un teclado numérico.

Secuencia normal

1. Introducir el código de barras del libro utilizando un escáner o teclado numérico.
2. Localizar préstamo (CU No2).
3. Si existe el registro del préstamo se determina su estado realizando una comparación entre la fecha actual y la fecha de devolución registrada.
4. Muestra el estado y la información del préstamo, el libro y el usuario en la interfaz.

Postcondiciones

Muestra el estado del préstamo del libro y la información relacionada.

Excepciones

Si no existe el préstamo, la interfaz muestra "LIBRO NO PRESTADO" como estado.

5.3.2. Localizar préstamo**Identificador**

CU No2

Actores

El sistema.

Descripción

Verifica en la base de datos si existe un préstamo registrado para el libro con el código de barras introducido.

Precondiciones

Se cuenta con un código de barras.

Secuencia normal

1. A partir del código de barras (*z30_barcode*) se obtiene de la tabla (*z30*) el identificador del ejemplar (*z30_rec_key*).
2. A partir del *z30_rec_key* se busca en la tabla (*z36*) el registro de un préstamo para dicho ejemplar.
3. Si existe el préstamo se obtiene la fecha del préstamo (*z36_loan_date*), fecha de vencimiento (*z36_due_date*), el identificador del libro (*z36_rec_key*), usuario (*z36_id*), catalogador de préstamo (*z36_loan_cataloger_name*) y catalogador de renovación (*z36_renew_cataloger_name*).
4. Localizar usuario (CU No3).

5. Localizar ejemplar (CU No4).

Postcondiciones

Se almacena la fecha de préstamo, la fecha de vencimiento, el identificador del libro, usuario, catalogador de préstamo y de renovación.

Excepciones

No existe el registro del préstamo.

Consultas a la base de datos

```
1. SELECT SUBSTR(LXX50.Z30.Z30_BARCODE,1,10),
      SUBSTR(TRIM(LXX50.Z36.Z36_ID),1,9),
      TO_DATE(LXX50.Z36.Z36_DUE_DATE,'yyyymmdd'),
      TO_DATE(LXX50.Z36.Z36_LOAN_DATE,'yyyymmdd'),
      SUBSTR(LXX50.Z36.Z36_LOAN_CATALOGER_NAME,1,20),
      SUBSTR(LXX50.Z36.Z36_RENEW_CATALOGER_NAME,1,20),
      SUBSTR(LXX50.Z30.Z30_REC_KEY,1,9),
FROM LXX50.Z36
  INNER JOIN LXX50.Z30
      ON (LXX50.Z36.Z36_REC_KEY = LXX50.Z30.Z30_REC_KEY)
WHERE LXX50.Z30.Z30_BARCODE = '<Código de barras>'
```

5.3.3. Localizar usuario

Identificador

CU No3

Actores

El sistema.

Descripción

Obtiene de la base de datos el nombre a partir del identificador del usuario registrado en el préstamo.

Precondiciones

Se cuenta con el identificador de un usuario este puede ser número de cuenta, trabajador o credencial (*z36.id*).

Secuencia normal

1. A partir del identificador del usuario (*z36_id*) se obtiene el nombre del usuario (*z303_name*).

Postcondiciones

Se almacena el nombre del usuario.

Excepciones

Ninguna.

Consultas a la base de datos

1.

```
SELECT SUBSTR(LXX50.Z303.Z303_NAME,1,32)
FROM LXX50.Z303
WHERE LXX50.Z303.Z303_REC_KEY = '<LXX50.Z36.Z36_ID>'
```

5.3.4. Localizar ejemplar

Identificador

CU No4

Actores

El sistema.

Descripción

Obtiene de la base de datos el título y autor del libro.

Precondiciones

Se cuenta con el identificador del libro (*z36_rec_key*) en la base de datos administrativa.

Secuencia normal

1. A partir del identificador del libro en la base administrativa (*z36_rec_key*) se obtiene la liga con la base bibliográfica (*z103_lkr_doc_number*).
2. Se extrae de la base bibliográfica el título (*z13_title*) y autor (*z13_author*) según la liga antes obtenida.

Postcondiciones

Se almacena el título y autor objeto (Clase) *Prestamo*.

Excepciones

Ninguna.

Consultas a la base de datos

1.

```
SELECT LPAD(LXX50.Z103.Z103_LKR_DOC_NUMBER,9,'0')
FROM LXX50.Z103
WHERE LXX50.Z103.Z103_REC_KEY
= '<base_administrativa_+LXX50.Z30.Z30_REC_KEY_+>'
AND LXX50.Z103.Z103_LKR_LIBRARY = '<base_bibliográfica>'
```
2.

```
SELECT SUBSTR(LXX01.Z13.Z13_TITLE,1,40),
SUBSTR(LXX01.Z13.Z13_AUTHOR,1,40)
FROM LXX01.Z13
WHERE LXX01.Z13.Z13_REC_KEY
= '<LPAD(LXX50.Z103.Z103_LKR_DOC_NUMBER,9,'0')>'
```

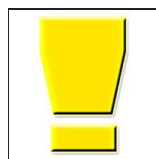
5.4. Interfaz

La interfaz solicita el número de adquisición que es introducido por un lector de código de barras y si se encuentra el registro del préstamo, devuelve la información señalada en la Sección 5.1 agrupándola en diferentes secciones relacionadas con el ejemplar, el usuario y el préstamo, la fotografía del usuario y el estado del préstamo que incluye una imagen (Ver Figura 5.4). Al iniciar la aplicación se muestra el logotipo o alguna imagen de la dependencia y cuando un usuario no cuenta foto el sistema coloca una imagen por defecto;

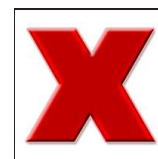
Figura 5.4: Interfaz del sistema Vigilancia (Biblioteca Central).



(a) Vigente / Verde



(b) Renovar / Amarillo



(c) Vencido / Rojo

Figura 5.5: Imágenes y colores para identificar el estado del préstamo.

cuando no se encuentra el número en préstamo no devuelve ninguna otra información solo el mensaje de "LIBRO NO PRESTADO".

Se asignó un color y una imagen a cada estado de préstamo para facilitar al usuario del sistema la identificación del mismo. El estado **vigente** tiene el color verde y una palomita, para **renovar** el color amarillo y un signo de admiración que cierra y para **vencido** rojo y el símbolo "X". Cuando un ejemplar no se encuentra prestado o no se tiene conexión con el servidor utiliza el color y la imagen del estado **vencido** (Ver Figura 5.5).

Para las dependencias que quieren verificar varias bibliotecas de forma simultánea estos elementos se repiten tantas veces como bibliotecas, un ejemplo es la Facultad de Ingeniería (Ver Figura 5.6a) y la Facultad de Medicina (Ver Figura 5.6b).



(a) Facultad de Ingeniería

(b) Facultad de Medicina

Figura 5.6: Interfaz del sistema Vigilancia multiple

5.5. Situación actual

El sistema *Vigilancia* originalmente contaba con un solo archivo que incluía la implementación de la interfaz, consultas a la base de datos y el control del flujo de la aplicación, actualmente el código se encuentra organizado en paquetes (Sección 4.3.4) como se muestra a continuación.

PAQUETE	CLASE(S)
configuración	Configuración Conexión
transacciones	Vigilancia
interfaz	Interfaz
objetos	Prestamo

A pesar de que son más líneas de código, el hacerlo configurable evita que se deba modificar el código solo cambiar los datos de la dependencia. Con esta organización de paquetes y clases, tanto la actualización, como el mantenimiento resultan más sencillas, además de que la programación se apega al Paradigma de Orientación Objetos.

Una de las modificaciones necesarias para este sistema fue el hacer configurable un parámetro llamado *cbdi* porque en el etiquetado de los ejemplares

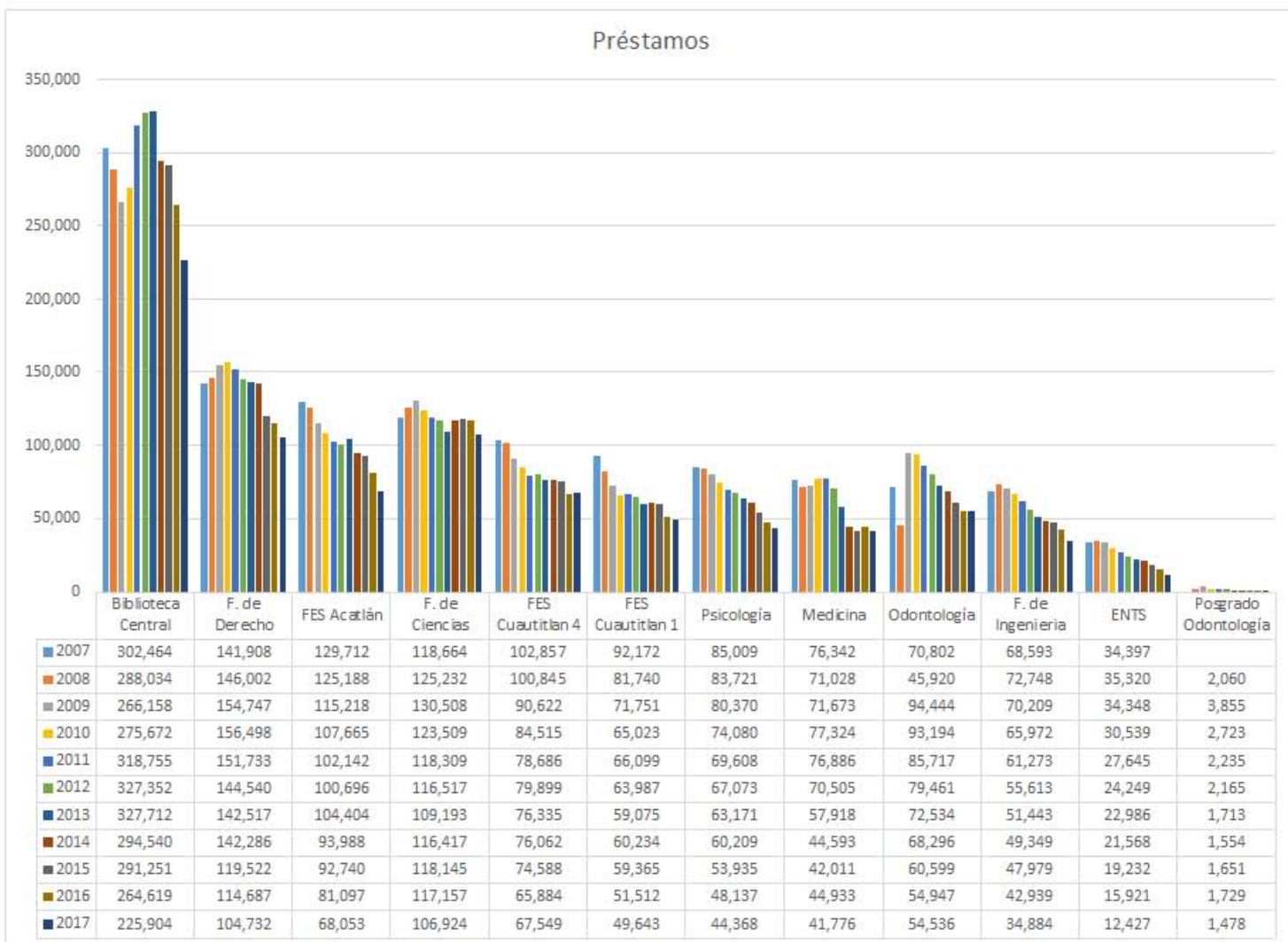
se incluyeron ceros a la izquierda y dependiendo del lector del código de barras no se eliminaban los ceros a la izquierda, por lo que el ejemplar no era encontrado en la base de datos pues estaba registrado sin los ceros del lado izquierdo.

El sistema ha sido instalado en la Biblioteca Central, Bibliotecas Conjuntas de las Ciencias de la Tierra (BCCT), Escuela Nacional de Trabajo Social, Facultades de Ciencias, Derecho, Ingeniería, Medicina, Odontología y Psicología; las Facultad de Estudios Superiores (FES) Acatlán, Aragón y Cuautitlán campo 1 y 4; el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) y la División de Estudios de Posgrado de la Facultad de Odontología; sumando un total de quince dependencias.

Este sistema es utilizado cada que un ejemplar sale de la biblioteca, es decir, cada que se realizó un préstamo a domicilio por lo tanto para tener una idea del uso del sistema en la figura 5.7 se puede observar la cantidad de préstamos en las dependencias en las que está instalado el sistema *Vigilancia*. En general se puede observar que la cantidad de préstamos ha decrecido en todas las dependencias a través de los años.

En este capítulo se mencionan los requerimientos que impulsaron el desarrollo de esta aplicación, los casos de uso identificados y desarrollados para resolverlo, el diagrama E-R concreto de la base de datos del sistema *Aleph* que es consultado y la interfaz de *Vigilancia*; además de exponer la situación actual.

Figura 5.7: Estadísticas del sistema *Vigilancia*.



Capítulo 6

Sistema automatizado de circulación: *Auto-Circulación*

Uno de los principales servicios en las bibliotecas de la UNAM es el préstamo, renovación y devolución de libros, en la actualidad estos procesos se realizan utilizando el módulo de *Circulación* del Cliente GUI ALEPH 500, como ya se mencionó con anterioridad el uso de éste requiere de licencias que generan un costo, por lo tanto tener una aplicación propia que realice estas actividades no solo implica un costo menor al presupuesto destinado, sino también disminuye la carga de trabajo de los módulos donde ésta se realiza.

6.1. Análisis de requerimientos

El objetivo del sistema de *Auto-Circulación* es automatizar por completo el préstamo a domicilio, la renovación y la devolución¹ de libros, por lo tanto su operación depende de hardware adicional que se mencionó en la Sección 4.3.7.

Para poder utilizar el sistema un usuario previamente registrado en la

¹ La devolución se habilita solo para las dependencias que lo requieran ya que en general se prefiere asegurar el estado del ejemplar al momento de ser devuelto.

biblioteca debe identificarse utilizando su *usuario*² y *contraseña*.

El sistema debe registrar en la base de datos el movimiento³ de la misma forma que lo hace el sistema *Aleph* para que al ser consultados desde este último no exista pérdida de información y viceversa, es decir, si un ejemplar fue prestado en el sistema *Aleph* debe ser reconocido en *Auto-Circulación* para realizar ahí la renovación o devolución. Por lo anterior se debe tener especial cuidado en todas las transacciones con la base de datos, pues el sistema realiza actualización y borrado de registros.

Para realizar cualquiera de los movimientos se deben efectuar las validaciones necesarias en cada uno de los casos, hacer el registro en la base de datos e imprimir un comprobante que sustituye el sello con la fecha de vencimiento en la papeleta del ejemplar en la forma tradicional.

Los usuarios del sistema son estudiantes, investigadores, profesores y trabajadores registrados en la biblioteca que solicitan prestado algún libro, en general todos los usuarios tienen la capacidad de utilizar un equipo de cómputo.

El diagrama de actividades de la Figura 6.1 muestra el proceso para realizar un movimiento a través del sistema *Aleph*, como se puede observar, el usuario de la biblioteca se acerca al mostrador con el ejemplar donde el personal utilizando el sistema *Aleph* selecciona la opción necesaria, ingresa el identificador del usuario y del ejemplar, si es autorizado se registra la fecha de devolución del libro en la papeleta en caso de préstamo y renovación.

Es importante mencionar que existen diferentes tipos de usuario y de colecciones⁴, a lo que se le denomina estatus, cada uno con sus respectivos permisos. El sistema *Aleph* almacena esta información en archivos de texto denominadas tablas ALEPH, las cuales son diferentes a las tablas de la base de datos (Ver Figura 2.4). Un ejemplo es la tabla *tab16* donde se definen

² El usuario es para cada persona el número de cuenta o empleado de la UNAM, también puede ser un número de credencial si la biblioteca la proporciona.

³ Movimiento se utiliza para referirnos a préstamo, renovación y devolución de forma indistinta.

⁴ Los ejemplares están organizados en colecciones, por ejemplo préstamo regular, préstamo en sala, estantería cerrada, reserva, etcétera y algunas de estas colecciones no pueden ser prestadas a domicilio.

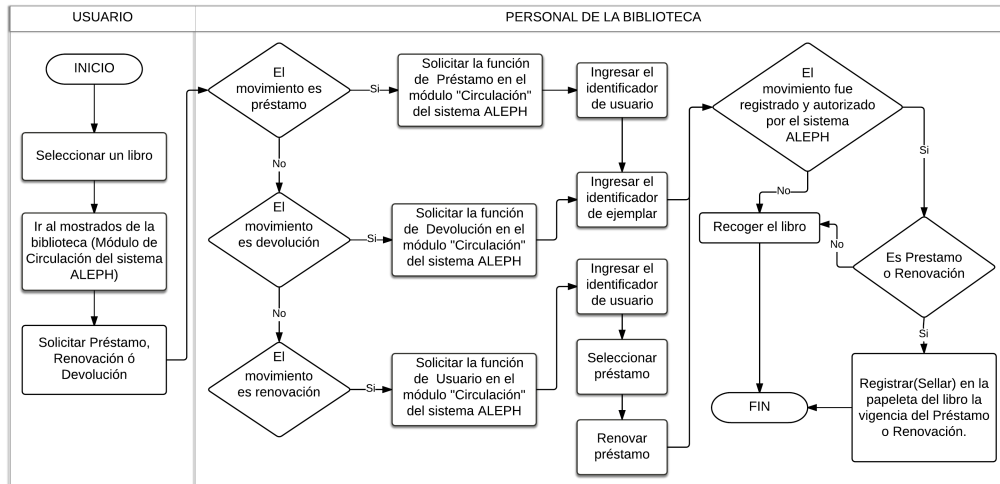


Figura 6.1: Diagrama de actividades de circulación.

las políticas de préstamo entre estatus de usuario y estatus de ejemplar que consulta el sistema *Auto-Circulación* (Ver Apéndice E.1).

Es indispensable que se registren los movimientos realizados en el sistema para ser monitoreados y registrados, por lo tanto se debe contar con una bitácora.

6.1.1. Validaciones y acciones

Para poder duplicar los movimientos tal y como los registra el sistema *Aleph* se detectaron las validaciones y acciones descritas en esta sección.

6.1.1.1. Validaciones

Las validaciones necesarias para realizar el préstamo sobre el usuario de la biblioteca son:

- Es un usuario de la biblioteca, es decir, debe estar registrado en el

sistema.

- El usuario debe estar vigente.
- El usuario no debe tener sanciones.
- El usuario no debe estar bloqueado.
- El usuario no debe tener multas.
- La cantidad de préstamos no debe exceder el límite de préstamos permitidos especificado en la *tab16* para el estatus de usuario solicitante.
- El usuario no debe tener ningún préstamo vencido.

Y sobre el ejemplar están:

- El ejemplar debe estar registrado en el sistema.
- El ejemplar debe pertenecer a una colección que puede salir de la biblioteca.
- El ejemplar no se encuentra prestado.
- El ejemplar no ha sido devuelto el mismo día que se solicita en préstamo por el mismo usuario.

Para renovar un préstamo lo que se necesita validar es que el préstamo no está vencido al momento de hacer la renovación y que tenga renovaciones disponibles según el número definido para el estatus de usuario en la *tab16*.

En el caso de las devoluciones es necesario validar que efectivamente se encuentre el ejemplar prestado y que el préstamo no haya vencido.

6.1.1.2. Acciones

A continuación se mencionan las acciones necesarias para cada movimiento después de haber realizado las validaciones. En la mayoría de los casos cada una de estas acciones son implementadas como funciones en la programación del sistema.

Préstamo

- Establecer la fecha de devolución.
- Verificar que la fecha de devolución no sea posterior a la fecha de vigencia del usuario.
- Asignar el identificador del préstamo.
- Incrementar el número de préstamos del ejemplar.
- Incrementar el contador de préstamos.
- Registrar el préstamo como usuario SAJ.
- Registrar el préstamo en la bitácora.
- Imprimir comprobante.

Renovación

- Establecer la fecha de devolución.
- Verificar que la fecha de devolución no sea posterior a la fecha de vigencia del usuario.
- Registrar la renovación como usuario SAJ.
- Registrar la renovación en la bitácora.
- Incrementar el número de renovaciones.
- Imprimir comprobante.

Devolución

- Mover el préstamo al registro histórico.
- Registrar la devolución en la bitácora.
- Imprimir comprobante.

6.2. Diagrama E-R

La Figura 6.2 muestra el diagrama E-R del sistema *Auto-Circulación* incluyendo los campos que son necesarios consultar, modificar o eliminar de cada tabla y la relación entre tablas. El detalle de la relación entre las tablas exclusivas de ejemplares (Ver Figura 4.3) y usuarios (Ver Figura 4.4) se explicaron en la Sección 4.2.

Las líneas punteadas significan que no están relacionadas propiamente; sin embargo a partir de ellas se genera algún valor necesario en algún calculo o table, por ejemplo, la fecha de devolución de un ejemplar se genera a partir de la tabla *Z301-CURCULATION DATES* y para cada préstamo su identificador *z36_number* se genera incrementando en uno el valor contenido en *z52_secuence* donde *z52_rec_key* es igual a *last-loan-number*, en otras palabras la tabla *Z52-SEQUENCE* es donde se almacenan algunos contadores, en este caso en particular el de los préstamos.

Todo movimiento inicia como un préstamo que es registrado en la tabla *z36-loans* con la fecha de préstamo (*z36_loan_date*) y la fecha de devolución (*z36_original_due_date*). Cuando se hace la renovación se registra la fecha (*z36_last_renew_date*) y se modifica la fecha de vencimiento (*z36_due_date*), además se registra el número de renovaciones (*z36_no_renewal*) y porque medio se efectuó (*z36_renew_mode*⁵); el número de renovaciones permitidas está definido en la tabla de *Aleph tab16*. Al realizar la devolución se registra

⁵ Si la renovación se realiza a través del modulo Circulación del sistema *Aleph* el valor es GUI si se renovó en el OPAC el valor es WEB y para identificar al sistema *Auto-Circulación* se asignó SAJ.

la fecha (*z36h_returned_date*) y el registro del préstamo se elimina de la tabla *z36*, y pasa a la *z36h(loans history)*.

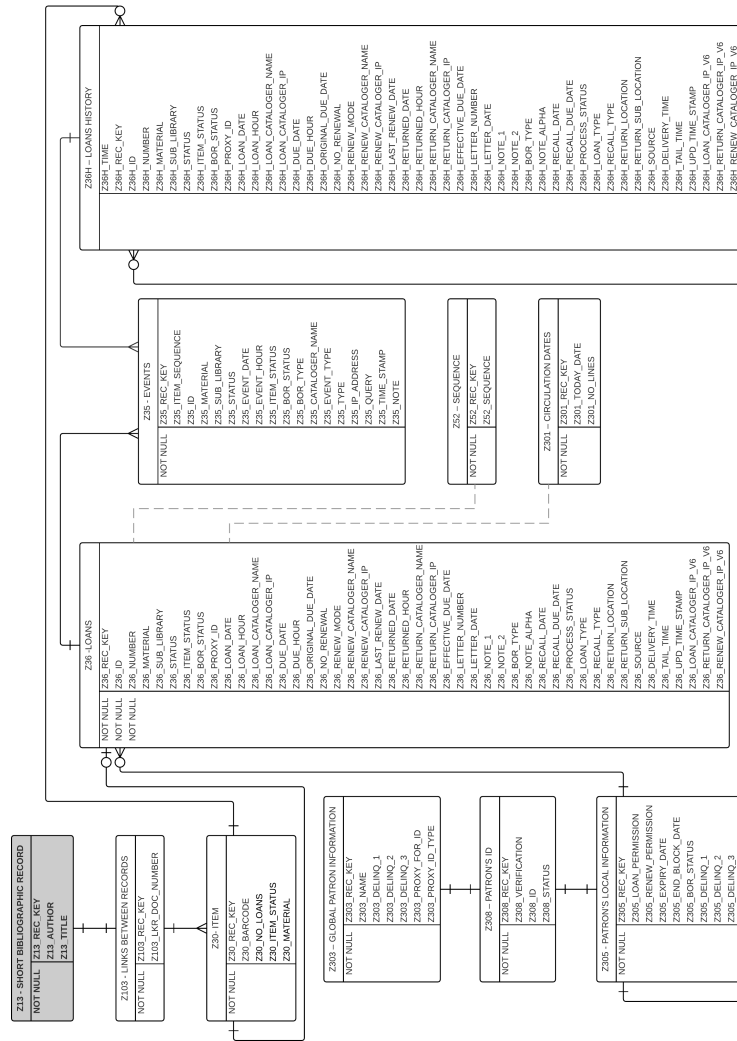


Figura 6.2: Diagrama E-R del sistema *Auto-Circulación*.

En la Figura 6.2 se incluyen todos los campos porque cuando un préstamo es devuelto todo el registro se mueve de la tabla *Z36* a la *Z36H*, aunque existan campos que estén vacíos.

La tabla *Z31 (CASH)* contiene el registro de las multas y es consultada

durante las validaciones, antes de realizar cualquier movimiento y solo se revisa si existe para el usuario en cuestión (*z31_rec_key*) alguna multa activa (*z31_status*).

Las consultas a la base de datos no representan ningún riesgo para la integridad de la información, sin embargo este sistema realiza actualizaciones y crea y elimina registros haciéndolo más complejo y delicado. En la descripción de los casos de uso (Sección 6.3) se precisa los campos utilizados de cada una de las tabla.

6.3. Casos de uso

Los casos de uso del sistema *Auto-Circulación* son (Ver Figura 6.3)

- Prestar ejemplar (6.3.2)
- Renovar préstamo (6.3.3)
- Devolver ejemplar (6.3.4)
- Autenticar usuario (6.3.1)
- Validar préstamo (6.3.5)
- Validar renovación (6.3.7)
- Validar devolución (6.3.9)
- Registrar préstamo (6.3.6)
- Registrar renovación (6.3.8)
- Registrar devolución (6.3.10)
- Imprimir recibo (6.3.11)

A continuación serán descritos cada uno de los casos de uso, el orden y su relación.

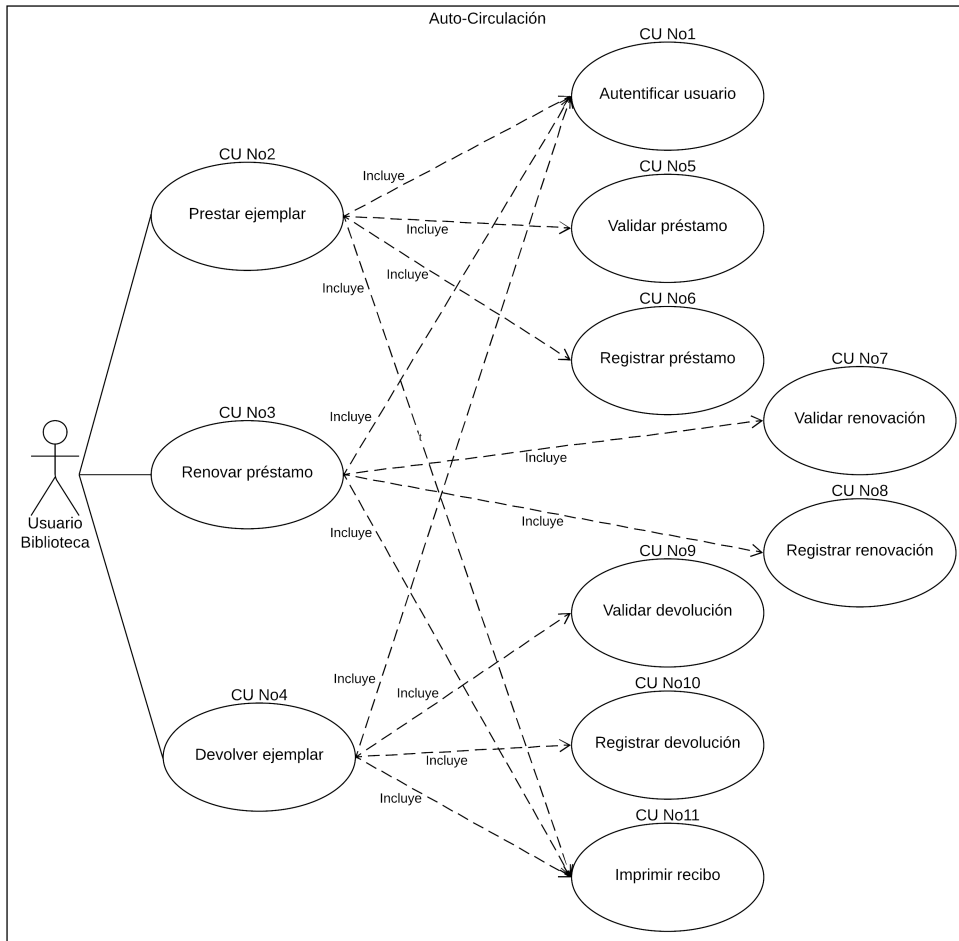


Figura 6.3: Casos de uso del sistema Auto-Circulación.

6.3.1. Autenticar usuario

Identificador

CU No1

Actores

Usuario de la biblioteca.

Descripción

Verificar la existencia del usuario en el sistema y la autenticidad del mismo a través de su número de credencial, número de cuenta o número de empleado y contraseña.

Precondiciones

Permite hacer uso del sistema, es decir, realizando un préstamo, renovación o devolución.

Secuencia normal

1. Seleccionar movimiento.
2. Introducir número de credencial, el número de cuenta o el número de empleado.
3. Introducir contraseña.
4. Introducir código de barras.
5. Valida que el número de credencial, el número de cuenta o el número de empleado este registrado y que la contraseña sea correcta.

Postcondiciones

Según el movimiento seleccionado se realiza alguno de los siguientes casos de uso:

- Prestar ejemplar (CU No2).
- Renovar préstamo (CU No3).
- Devolver ejemplar (CU No4).

Excepciones

Si el usuario no está registrado o la contraseña que se introdujo no es válida se informa que los datos son incorrectos.

6.3.2. Prestar un ejemplar

Identificador

CU No2

Actores

Sistema.

Descripción

Realiza el préstamo a domicilio de un ejemplar a un usuario.

Precondiciones

El usuario se autenticó exitosamente.

Secuencia normal

1. Validar préstamo (CU No5).
2. Registrar préstamo (CU No6).
3. Imprimir recibo (CU No11).

Postcondiciones

El ejemplar es prestado al usuario, realizando el registro en la base de datos.

Excepciones

Si ocurre algún error se informa al usuario y no es prestado el ejemplar.

6.3.3. Renovar un préstamo

Identificador

CU No3

Actores

Sistema.

Descripción

Se extiende la vigencia de un préstamo.

Precondiciones

El usuario se autenticó exitosamente y el ejemplar se encuentra prestado.

Secuencia normal

1. Validar renovación (CU No7).
2. Registrar renovación (CU No8).
3. Imprimir recibo (CU No11).

Postcondiciones

El préstamo se extiende.

Excepciones

Si ocurre algún error se informa al usuario y no es renovado el préstamo.

6.3.4. Devolver un ejemplar

Identificador

CU No4

Actores

Sistema.

Descripción

Devuelve un ejemplar que se encuentra en préstamo.

Precondiciones

El usuario se autenticó exitosamente y el préstamo aun es vigente.

Secuencia normal

1. Validar devolución (CU No9).
2. Registrar devolución (CU No10).
3. Imprimir recibo (CU No11).

Postcondiciones

El ejemplar es devuelto.

Excepciones

Si ocurre algún error se informa al usuario y no es devuelto el ejemplar.

6.3.5. Validar préstamo**Identificador**

CU No5

Actores

Sistema.

Descripción

Realiza todas las validaciones necesarias para realizar el préstamo.

Precondiciones

El usuario es válido y el movimiento seleccionado fue préstamo.

Secuencia normal

1. Validar que el usuario se encuentre vigente (*z305_expiry_date*).

2. Revisa que el usuario tenga permisos de préstamo (*z305_loan_permission*) y renovación (*z305_renew_permission*) en función de su estatus (*z305_bor_status*).
3. Verificar que el usuario no este bloqueado (*z305_end_block_date*).
4. Validar que el usuario no tenga sanciones locales (*z303_delinq_1*, *z303_delinq_2* y *z303_delinq_3*) y generales (*z305_delinq_1*, *z305_delinq_2* y *z305_delinq_3*).
5. Verificar que el usuario no tenga multas (*z31*).
6. Validar que el usuario no tenga préstamos vencidos y que no haya alcanzado el número de préstamos permitidos (*z36*).
7. Verifica cuantos préstamos y renovaciones tiene derecho el usuario en base su estatus⁶ (*z305_bor_status*) y las políticas de préstamo ⁷ de la biblioteca.
8. Validar que el ejemplar/libro exista (*z30*).
9. Verifica si el libro pertenece a una colección⁸ que se presta a domicilio en función de su estatus (*z30_item_status*) y el tipo de material (*z30_material*).
10. Verificar que el libro no este prestado (*z36*).

Postcondiciones

Si las validaciones se cumplen se almacena la información en la base de datos (Registrar préstamo (CU No6)).

Excepciones

Si no se cumple con alguna de las condiciones de préstamo manda un mensaje de error al usuario y no se realiza el préstamo.

⁶ Los estatus de los usuarios están definido en un archivo de texto llamado *tab31.spa* (E.3).

⁷ Las políticas de préstamo están definidas en un archivo de texto llamado *tab16*(E.1).

⁸ Los estatus de los ejemplares están definido en un archivo de texto llamado *tab15.spa* E.2.

Consultas a la base de datos

1.

```
SELECT TO_DATE(LXX50.Z305.Z305_EXPIRY_DATE, 'yyyymmdd'),
LXX50.Z305.Z305_LOAN_PERMISSION, LXX50.Z305.Z305_RENEW_PERMISSION
FROM LXX50.Z305
WHERE SUBSTR(LXX50.Z305.Z305_REC_KEY,1,12)
= RPAD('<número_de_cedencial>',12,'_')
AND SUBSTR(LXX50.Z305.Z305_REC_KEY,13,17) = 'LXX_...'
```
2.

```
SELECT LXX50.Z305.Z305_END_BLOCK_DATE, LXX50.Z305.Z305_BOR_STATUS
FROM LXX50.Z305
WHERE (LXX50.Z305.Z305_REC_KEY
= RPAD('<número_de_cedencial>',12,'_') || 'LXX50')
```
3.

```
SELECT LXX50.Z305.Z305_END_BLOCK_DATE
FROM LXX50.Z305
WHERE (LXX50.Z305.Z305_REC_KEY
= RPAD('<número_de_cedencial>',12,'_') || 'LXX_...')
```
4.

```
SELECT LXX50.Z303.Z303_DELINQ_1, LXX50.Z303.Z303_DELINQ_2,
LXX50.Z303.Z303_DELINQ_3
FROM LXX50.Z303
WHERE SUBSTR(LXX50.Z303.Z303_REC_KEY,1,12)
= RPAD('<número_de_cedencial>',12,'_')
```
5.

```
SELECT LXX50.Z305.Z305_DELINQ_1, LXX50.Z305.Z305_DELINQ_2,
LXX50.Z305.Z305_DELINQ_3
FROM LXX50.Z305
WHERE SUBSTR(LXX50.Z305.Z305_REC_KEY,1,12)
= RPAD('<número_de_cedencial>',12,'_')
AND (LXX50.Z305.Z305_REC_KEY
= RPAD('<número_de_cedencial>',12,'_') || 'LXX50')
```
6.

```
SELECT count(*)
FROM LXX50.Z31
WHERE SUBSTR(LXX50.Z31.Z31_REC_KEY,1,12)
= RPAD('<número_de_cedencial>',12,'_')
AND LXX50.Z31.Z31_STATUS = 'O'
```
7.

```
SELECT count(*)
FROM LXX50.Z36
WHERE LXX50.Z36.Z36_ID = '<número_de_cedencial>'
AND LXX50.Z36.Z36_DUE_DATE < TO_NUMBER(TO_CHAR(SYSDATE, 'YYYYMMDD'))
```
8.

```
SELECT LXX50.Z36.Z36_REC_KEY
FROM LXX50.Z36
LEFT OUTER JOIN LXX50.Z30
ON LXX50.Z30.Z30_REC_KEY = LXX50.Z36.Z36_REC_KEY
WHERE LXX50.Z30.Z30_BARCODE = '<código_de_barras>'
```

9.

```
SELECT LXX50.Z30.Z30_REC_KEY, LXX50.Z30.Z30_ITEM_STATUS,  
LXX50.Z30.Z30_MATERIAL  
FROM LXX50.Z30  
WHERE LXX50.Z30.Z30_BARCODE = '<código_de_barras>'
```
10.

```
SELECT count(*)  
FROM LXX50.Z36  
WHERE LXX50.Z36.Z36_ID = '<número_de_cedencial>'
```

6.3.6. Registrar préstamo

Identificador

CU No6

Actores

Sistema.

Descripción

Se insertan y actualizan los registros en las tablas de la base de datos necesarias para registrar el préstamo.

Precondiciones

Se cumplieron todas las validaciones necesarias para realizar el préstamo (Validar préstamo (CU No5)).

Secuencia normal

1. Obtener la fecha de devolución (*z301*) según el estatus del usuario y ejemplar.
2. Comparar la fecha de devolución con la fecha de la vigencia del usuario y toma la menor.
3. Obtener el contador de préstamos e incrementarlo (*z52*).
4. Transacción de registro se préstamo
 - a) Inserción del registro correspondiente de préstamo (*z36*).

- b) Actualizar el contador (*z52*).
- c) Actualizar la cantidad de préstamos del ejemplar (*z30*).
- d) Inserción del registro del evento (*z35*).

Postcondiciones

El préstamo es registrado de forma adecuada al insertar y actualizar todos los registros y tablas de la base de datos correctamente.

Excepciones

Si alguna de las sentencias SQL contenidas en la transacción falla se hace un *ROLLBACK*, se notifica al usuario que hubo un error y no se realiza el préstamo.

Consultas a la base de datos

1.

```
SELECT LXX50.Z301.Z301_NO_LINES , LXX50.Z301.Z301_LINE
FROM LXX50.Z301
WHERE TRIM(LXX50.Z301.Z301_REC_KEY) = 'LXX'
```
2.

```
SELECT LPAD(TO_CHAR(LXX50.Z52.Z52_SEQUENCE + 1),9,'0')
FROM LXX50.Z52
WHERE LXX50.Z52.Z52_REC_KEY = 'last-loan-number'
```
3.

```
INSERT INTO LXX50.Z36
(LXX50.Z36.Z36_REC_KEY, LXX50.Z36.Z36_ID, LXX50.Z36.Z36_NUMBER,
LXX50.Z36.Z36_MATERIAL, LXX50.Z36.Z36_SUB_LIBRARY,
LXX50.Z36.Z36_STATUS, LXX50.Z36.Z36_LOAN_DATE,
LXX50.Z36.Z36_LOAN_HOUR, LXX50.Z36.Z36_DUE_DATE,
LXX50.Z36.Z36_DUE_HOUR, LXX50.Z36.Z36_ITEM_STATUS,
LXX50.Z36.Z36_BOR_STATUS, LXX50.Z36.Z36_LOAN_CATALOGER_NAME,
LXX50.Z36.Z36_ORIGINAL_DUE_DATE, LXX50.Z36.Z36_PROXY_ID)
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)
```
4.

```
UPDATE LXX50.Z52
SET LXX50.Z52.Z52_SEQUENCE = <z52_sequence_ant + 1>
WHERE LXX50.Z52.Z52_REC_KEY = 'last-loan-number'
```
5.

```
UPDATE LXX50.Z30
SET LXX50.Z30.Z30_NO_LOANS = <z30_no_loans + 1>
WHERE LXX50.Z30.Z30_REC_KEY = <número_de_matriz>
```

```

6. INSERT INTO LXX50.Z35
(LXX50.Z35.Z35_REC_KEY, LXX50.Z35.Z35_ITEM_SEQUENCE,
LXX50.Z35.Z35_ID,
LXX50.Z35.Z35_MATERIAL, LXX50.Z35.Z35_SUB_LIBRARY,
LXX50.Z35.Z35_STATUS, LXX50.Z35.Z35_EVENT_DATE,
LXX50.Z35.Z35_EVENT_HOUR, LXX50.Z35.Z35_ITEM_STATUS,
LXX50.Z35.Z35_BOR_STATUS, LXX50.Z35.Z35_BOR_TYPE,
LXX50.Z35.Z35_CATALOGER_NAME, LXX50.Z35.Z35_EVENT_TYPE,
LXX50.Z35.Z35_TYPE, LXX50.Z35.Z35_IP_ADDRESS, LXX50.Z35.Z35_QUERY,
LXX50.Z35.Z35_TIME_STAMP, LXX50.Z35.Z35_NOTE)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, NULL,
TO_CHAR(CURRENT_TIMESTAMP, 'YYYYMMDDHH24MIFF4') || '000000', NULL)

```

6.3.7. Validar renovación

Identificador

CU No7

Actores

Sistema.

Descripción

Realiza todas las validaciones necesarias para extender la vigencia del préstamo, es decir, renovar el préstamo.

Precondiciones

El usuario se autenticó exitosamente y el préstamo esta vigente.

Secuencia normal

1. Validar que el usuario este vigente (*z305_expiry_date*).
2. Verificar que el usuario no este bloqueado (*z305_end_block_date*).
3. Revisa que el usuario tenga permisos de préstamo (*z305_loan_permission*) y renovación (*z305_renew_permission*) en función de su estatus (*z305_bor_status*).
4. Validar que el usuario no tenga sanciones locales (*z303_delinq_1*, *z303_delinq_2* y *z303_delinq_3*) y generales (*z305_delinq_1*, *z305_delinq_2* y *z305_delinq_3*).

5. Verificar que el usuario no tenga multas (*z31*).
6. Validar que el usuario no tenga préstamos vencidos y que no haya alcanzado el número de préstamos permitidos (*z36*).
7. Verificar cuantos préstamos y renovaciones tiene derecho el usuario en base su estatus (*z305_bor_status*) y las políticas de préstamo de la biblioteca.
8. Validar que el ejemplar/libro exista (*z30*).
9. Verificar que el libro este prestado (*z36*).
10. Revisar que el préstamo no este vencido (*z36_due_date*).

Postcondiciones

Si las validaciones se cumplen se almacena la información en la base de datos (Realizar el CU No.8)

Excepciones

Si no se cumple con alguna de las condiciones de renovación manda un mensaje de error al usuario y no se realiza la renovación.

Consultas a la base de datos

1.

```
SELECT TO_DATE(LXX50.Z35.Z305_EXPIRY_DATE, 'yyyymmdd'),
LXX50.Z35.Z305_LOAN_PERMISSION, LXX50.Z35.Z305_RENEW_PERMISSION
FROM LXX50.Z35
WHERE SUBSTR(LXX50.Z35.Z305_REC_KEY,1,12)
= RPAD('<número_de_cedencial>',12,'_')
AND SUBSTR(LXX50.Z35.Z305_REC_KEY,13,17) = 'L10_..'
```
2.

```
SELECT LXX50.Z35.Z305_END_BLOCK_DATE, LXX50.Z35.Z305_BOR_STATUS
FROM LXX50.Z35
WHERE (LXX50.Z35.Z305_REC_KEY
= RPAD('<número_de_cedencial>',12,'_') || 'L1050')
```
3.

```
SELECT LXX50.Z305.Z305_END_BLOCK_DATE
FROM LXX50.Z305
WHERE (LXX50.Z305.Z305_REC_KEY
= RPAD('<número_de_cedencial>',12,'_') || 'L10_..')
```

4.

```
SELECT LXX50.Z303.Z303_DELINQ_1, LXX50.Z303.Z303_DELINQ_2,
LXX50.Z303.Z303_DELINQ_3
FROM LXX50.Z303
WHERE SUBSTR(LXX50.Z303.Z303_REC_KEY,1,12)
= RPAD('<número_de_cedencial>',12,'_')
```
5.

```
SELECT LXX50.Z305.Z305_DELINQ_1, LXX50.Z305.Z305_DELINQ_2,
LXX50.Z305.Z305_DELINQ_3
FROM LXX50.Z305
WHERE SUBSTR(LXX50.Z305.Z305_REC_KEY,1,12)
= RPAD('<número_de_cedencial>',12,'_')
AND (LXX50.Z305.Z305_REC_KEY
= RPAD('<número_de_cedencial>',12,'_') || 'LXX50')
```
6.

```
SELECT count(*)
FROM LXX50.Z31
WHERE SUBSTR(LXX50.Z31.Z31_REC_KEY,1,12)
= RPAD('<número_de_cedencial>',12,'_')
AND LXX50.Z31.Z31_STATUS = 'O'
```
7.

```
SELECT count(*)
FROM LXX50.Z36
WHERE LXX50.Z36.Z36_ID = '<número_de_cedencial>'
AND LXX50.Z36.Z36_DUE_DATE < TO_NUMBER(TO_CHAR(SYSDATE, 'YYYYMMDD'))
```
8.

```
SELECT LXX50.Z36.Z36_REC_KEY
FROM LXX50.Z36
LEFT OUTER JOIN LXX50.Z30
ON LXX50.Z30.Z30_REC_KEY = LXX50.Z36.Z36_REC_KEY
WHERE LXX50.Z30.Z30_BARCODE = '<código_de_barras>'
```
9.

```
SELECT LXX50.Z30.Z30_REC_KEY, LXX50.Z30.Z30_ITEM_STATUS,
LXX50.Z30.Z30_MATERIAL
FROM LXX50.Z30
WHERE LXX50.Z30.Z30_BARCODE = '<código_de_barras>'
```
10.

```
SELECT count(*)
FROM LXX50.Z36
WHERE LXX50.Z36.Z36_ID = '<número_de_cedencial>'
```
11.

```
SELECT LXX50.Z36.Z36_REC_KEY,
TO_CHAR(SYSTIMESTAMP, 'YYYYMMDDHH24MIF3'), LXX50.Z36.Z36_ID,
LXX50.Z36.Z36_NUMBER, LXX50.Z36.Z36_MATERIAL,
LXX50.Z36.Z36_SUB_LIBRARY, LXX50.Z36.Z36_STATUS,
LXX50.Z36.Z36_LOAN_DATE, LXX50.Z36.Z36_LOAN_HOUR,
LXX50.Z36.Z36_DUE_DATE, LXX50.Z36.Z36_DUE_HOUR,
LXX50.Z36.Z36_RETURNED_DATE, LXX50.Z36.Z36_RETURNED_HOUR,
LXX50.Z36.Z36_ITEM_STATUS, LXX50.Z36.Z36_BOR_STATUS,
LXX50.Z36.Z36_LETTER_NUMBER, LXX50.Z36.Z36_LETTER_DATE,
CASE WHEN LXX50.Z36.Z36_NO_RENEWAL IS NULL
THEN 0 ELSE LXX50.Z36.Z36_NO_RENEWAL END,
```

```

LXX50.Z36.Z36_NOTE_1, LXX50.Z36.Z36_NOTE_2,
LXX50.Z36.Z36_LOAN_CATALOGER_NAME,
    LXX50.Z36.Z36_RETURN_CATALOGER_NAME,
LXX50.Z36.Z36_RENEW_CATALOGER_NAME, LXX50.Z36.Z36_RENEW_MODE,
LXX50.Z36.Z36_BOR_TYPE, LXX50.Z36.Z36_NOTE_ALPHA,
    LXX50.Z36.Z36_RECALL_DATE,
LXX50.Z36.Z36_RECALL_DUE_DATE, LXX50.Z36.Z36_LAST_RENEW_DATE,
LXX50.Z36.Z36_ORIGINAL_DUE_DATE, LXX50.Z36.Z36_PROCESS_STATUS,
LXX50.Z36.Z36_LOAN_TYPE, LXX50.Z36.Z36_PROXY_ID,
    LXX50.Z36.Z36_RECALL_TYPE,
LXX50.Z36.Z30_ITEM_STATUS, LXX50.Z36.Z30_REC_KEY
FROM LXX50.Z36
LEFT OUTER JOIN LXX50.Z30
ON (LXX50.Z30.Z30_REC_KEY = LXX50.Z36.Z36_REC_KEY)
WHERE LXX50.Z30.Z30_BARCODE = '<código_de_barras>'
AND LXX50.Z36.Z36_ID = '<número_de_credencial>'

```

6.3.8. Registrar renovación

Identificador

CU No8

Actores

Sistema.

Descripción

Se insertan y actualizan los registros en las tablas de la base de datos necesarias para extender el préstamo, es decir, realizar la renovación.

Precondiciones

Se cumplieron todas las validaciones necesarias para realizar la renovación (Validar renovación (CU No7)).

Secuencia normal

1. Obtener la fecha de devolución (*z301*) según el estatus del usuario y ejemplar.
2. Comparar la fecha de devolución con la fecha de la vigencia del usuario y toma la menor.
3. Transacción de registro de renovación.

- Actualización de la fecha de devolución (*z36_due_date*), registra la fecha de la renovación (*z36_last_renew_date*), registra el catalogador que lo hizo (*z36_renew_cataloger_name*), incrementa el número de renovaciones (*z36_no_renewal*) y registra el modo en que se renovó el préstamo (*z36_renew_mode*) del registro correspondiente a ese préstamo.
- Inserción del registro del evento (*z35*).

Postcondiciones

La renovación es registrada de forma adecuada al insertar y actualizar todos los registros y tablas de la base de datos correctamente.

Excepciones

Si alguna de las sentencias SQL contenidas en la transacción falla se hace un *ROLLBACK*, se notifica al usuario que hubo un error y no se realiza la renovación.

Consultas a la base de datos

1.

```
SELECT LXX50.Z301.Z301_NO_LINES, LXX50.Z301.Z301_LINE
FROM LXX50.Z301
WHERE TRIM(LXX50.Z301.Z301_REC_KEY) = 'LXX'
```
2.

```
UPDATE LXX50.Z36
SET LXX50.Z36.Z36_DUE_DATE = ?,
LXX50.Z36.Z36_NO_RENEWAL =
(CASE WHEN LXX50.Z36.Z36_NO_RENEWAL IS NULL
THEN 0 ELSE LXX50.Z36.Z36_NO_RENEWAL END) + 1,
LXX50.Z36.Z36_RENEW_CATALOGER_NAME = ?,
LXX50.Z36.Z36_RENEW_MODE = ?,
LXX50.Z36.Z36_LAST_RENEW_DATE = ?
WHERE LXX50.Z36.Z36_NUMBER= ?
```
3.

```
INSERT INTO LXX50.Z35
(LXX50.Z35.Z35_REC_KEY, LXX50.Z35.Z35_ITEM_SEQUENCE,
LXX50.Z35.Z35_ID, LXX50.Z35.Z35_MATERIAL, LXX50.Z35.Z35_SUB_LIBRARY,
LXX50.Z35.Z35_STATUS, LXX50.Z35.Z35_EVENT_DATE,
LXX50.Z35.Z35_EVENT_HOUR, LXX50.Z35.Z35_ITEM_STATUS,
LXX50.Z35.Z35_BOR_STATUS, LXX50.Z35.Z35_BOR_TYPE,
LXX50.Z35.Z35_CATALOGER_NAME, LXX50.Z35.Z35_EVENT_TYPE,
LXX50.Z35.Z35_TYPE, LXX50.Z35.Z35_IP_ADDRESS, LXX50.Z35.Z35_QUERY,
LXX50.Z35.Z35_TIME_STAMP, LXX50.Z35.Z35_NOTE)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, NULL,
TO_CHAR(CURRENT_TIMESTAMP, 'YYYYMMDDHH24MIFF4') || '000000', NULL)
```

6.3.9. Validar devolución

Identificador

CU No9

Actores

Sistema.

Descripción

Realizar todas las validaciones necesarias para realizar la devolución de un ejemplar.

Precondiciones

El usuario se autenticó exitosamente y el préstamo esta vigente.

Secuencia normal

1. Validar que el ejemplar se encuentre prestado.
2. Verificar que la devolución sea en tiempo, es decir que la fecha de vencimiento (*z36_due_date*) sea mayor o igual a la actual.

Postcondiciones

Si las validaciones se cumplen se almacena la información en la base de datos (Realizar el CU No.10)

Excepciones

Si no se cumple con alguna de las condiciones para realizar la devolución manda un mensaje de error al usuario y no se realiza la devolución.

Consultas a la base de datos

1.

```
SELECT LXX50.Z36.Z36_REC_KEY,
TO_CHAR(SYSTIMESTAMP , 'YYYYMMDDHH24MIF3'),
LXX50.Z36.Z36_ID , LXX50.Z36.Z36_NUMBER, LXX50.Z36.Z36_MATERIAL,
LXX50.Z36.Z36_SUB_LIBRARY, LXX50.Z36.Z36_STATUS,
LXX50.Z36.Z36_LOAN_DATE, LXX50.Z36.Z36_LOAN_HOUR,
LXX50.Z36.Z36_DUE_DATE, LXX50.Z36.Z36_DUE_HOUR,
```

```

LXX50.Z36.Z36_RETURNED_DATE, LXX50.Z36.Z36_RETURNED_HOUR,
LXX50.Z36.Z36_ITEM_STATUS, LXX50.Z36.Z36_BOR_STATUS,
LXX50.Z36.Z36_LETTER_NUMBER, LXX50.Z36.Z36_LETTER_DATE,
CASE WHEN LXX50.Z36.Z36_NO_RENEWAL IS NULL
THEN 0 ELSE LXX50.Z36.Z36_NO_RENEWAL END,
LXX50.Z36.Z36_NOTE_1, LXX50.Z36.Z36_NOTE_2,
LXX50.Z36.Z36_LOAN_CATALOGER_NAME,
LXX50.Z36.Z36_RETURN_CATALOGER_NAME,
LXX50.Z36.Z36_RENEW_CATALOGER_NAME, LXX50.Z36.Z36_RENEW_MODE,
LXX50.Z36.Z36_BOR_TYPE, LXX50.Z36.Z36_NOTE_ALPHA,
LXX50.Z36.Z36_RECALL_DATE, LXX50.Z36.Z36_RECALL_DUE_DATE,
LXX50.Z36.Z36_LAST_RENEW_DATE, LXX50.Z36.Z36_ORIGINAL_DUE_DATE,
LXX50.Z36.Z36_PROCESS_STATUS, LXX50.Z36.Z36_LOAN_TYPE,
LXX50.Z36.Z36_PROXY_ID, LXX50.Z36.Z36_RECALL_TYPE,
LXX50.Z30.Z30_ITEM_STATUS, LXX50.Z30.Z30_REC_KEY
FROM LXX50.Z36
LEFT OUTER JOIN LXX50.Z30
ON (LXX50.Z30.Z30_REC_KEY = LXX50.Z36.Z36_REC_KEY)
WHERE LXX50.Z30.Z30_BARCODE = '<codigo_de_barras>'
AND LXX50.Z36.Z36_ID = '<número_de_credencial>'

```

6.3.10. Registrar devolución

Identificador

CU No10

Actores

Sistema.

Descripción

Se insertan y actualizan los registros en las tablas de la base de datos necesarias para registrar la devolución.

Precondiciones

Se cumplieron todas las validaciones necesarias para realizar la devolución (Validar devolución(CU No9)).

Secuencia normal

1. Transacción de devolución.
 - Registra el préstamo en el historial de préstamos (*z36h*).

- Elimina el préstamo (*z36*).
- Inserción del registro del evento (*z35*).

Postcondiciones

La devolución es registrada de forma adecuada al insertar y actualizar todos los registros y tablas de la base de datos correctamente.

Excepciones

Si alguna de las sentencias SQL contenidas en la transacción falla se hace un *ROLLBACK*, se notifica al usuario que hubo un error y no se realiza la renovación.

Consultas a la base de datos

1. **INSERT INTO** LXX50.Z36H
 (LXX50.Z36H.Z36H.TIME, LXX50.Z36H.Z36H.REC_KEY, LXX50.Z36H.Z36H.ID,
 LXX50.Z36H.Z36H.NUMBER, LXX50.Z36H.Z36H.MATERIAL,
 LXX50.Z36H.Z36H.SUB_LIBRARY, LXX50.Z36H.Z36H.STATUS,
 LXX50.Z36H.Z36H.LOAN_DATE, LXX50.Z36H.Z36H.LOAN_HOUR,
 LXX50.Z36H.Z36H.DUE_DATE, LXX50.Z36H.Z36H.DUE_HOUR,
 LXX50.Z36H.Z36H.RETURNED_DATE, LXX50.Z36H.Z36H.RETURNED_HOUR,
 LXX50.Z36H.Z36H.ITEM_STATUS, LXX50.Z36H.Z36H.BOR_STATUS,
 LXX50.Z36H.Z36H.LETTER_NUMBER, LXX50.Z36H.Z36H.LETTER_DATE,
 LXX50.Z36H.Z36H.NO_RENEWAL, LXX50.Z36H.Z36H.NOTE_1,
 LXX50.Z36H.Z36H.NOTE_2, LXX50.Z36H.Z36H.LOAN_CATALOGER_NAME,
 LXX50.Z36H.Z36H.RETURN_CATALOGER_NAME,
 LXX50.Z36H.Z36H.RENEW_CATALOGER_NAME, LXX50.Z36H.Z36H.RENEW_MODE,
 LXX50.Z36H.Z36H.BOR_TYPE, LXX50.Z36H.Z36H.NOTE_ALPHA,
 LXX50.Z36H.Z36H.RECALL_DATE, LXX50.Z36H.Z36H.RECALL_DUE_DATE,
 LXX50.Z36H.Z36H.LAST_RENEW_DATE, LXX50.Z36H.Z36H.ORIGINAL_DUE_DATE,
 LXX50.Z36H.Z36H.PROCESS_STATUS, LXX50.Z36H.Z36H.LOAN_TYPE,
 LXX50.Z36H.Z36H.PROXY_ID, LXX50.Z36H.Z36H.RECALL_TYPE)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
 TO_NUMBER(TO_CHAR(SYSDATE, 'YYYYMMDD')),
 TO_NUMBER(TO_CHAR(SYSDATE, 'HH24MI')),
 ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
2. **DELETE FROM** LXX50.Z36
WHERE LXX50.Z36.Z36.REC_KEY = '<número_de_préstamo>'
3. **INSERT INTO** LXX50.Z35
 (LXX50.Z35.Z35.REC_KEY, LXX50.Z35.Z35.ITEM_SEQUENCE,
 LXX50.Z35.Z35.ID, LXX50.Z35.Z35.MATERIAL, LXX50.Z35.Z35.SUB_LIBRARY,
 LXX50.Z35.Z35.STATUS, LXX50.Z35.Z35.EVENT_DATE,
 LXX50.Z35.Z35.EVENT_HOUR, LXX50.Z35.Z35.ITEM_STATUS,
 LXX50.Z35.Z35.BOR_STATUS, LXX50.Z35.Z35.BOR_TYPE,
 LXX50.Z35.Z35.CATALOGER_NAME, LXX50.Z35.Z35.EVENT_TYPE,

```
LXX50.Z35.Z35_TYPE, LXX50.Z35.Z35_IP_ADDRESS, LXX50.Z35.Z35_QUERY,
LXX50.Z35.Z35_TIME_STAMP, LXX50.Z35.Z35_NOTE)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, NULL,
TO_CHAR(CURRENT_TIMESTAMP, 'YYYYMMDDHH24MIFF4') || '000000', NULL)
```

6.3.11. Imprimir Recibo

Identificador

CU No11

Actores

Sistema.

Descripción

Se imprime el comprobante del movimiento.

Precondiciones

Se finalizó un movimiento de forma exitosa.

Secuencia normal

1. Obtener el identificador del ejemplar (*z36_rec_key*) en la base administrativa a partir del movimiento.
2. Obtener la liga del ejemplar (*z103_lkr_doc_number*) desde la base administrativa a la bibliográfica.
3. Obtener el título (*z13_title*) y autor (*z13_author*) del ejemplar en la base bibliográfica.
4. Definir el comprobante del movimiento con los datos necesarios como son la información del usuario, ejemplar y prestamo.
5. Imprimir el comprobante.

Postcondiciones

Se imprimió el comprobante del movimiento.

Excepciones

Si falla la impresora el movimiento no es afectado, queda registrado en el sistema.

Consultas a la base de datos

1.

```
SELECT SUBSTR(LXX50.Z36.Z36_REC_KEY,1,9)
FROM LXX50.Z36
WHERE LXX50.Z36.Z36_NUMBER='<número_de_préstamo>
```
2.

```
SELECT LPAD(LXX50.Z103.Z103_LKR_DOC_NUMBER,9,'0')
FROM LXX50.Z103
WHERE LXX50.Z103.Z103_REC_KEY = 'LXX50<número_de_ejemplar>01'
AND Z103_LKR_LIBRARIY='LXX01'
```
3.

```
SELECT SUBSTR(LXX01.Z13.Z13_TITLE,1,50),
SUBSTR(LXX01.Z13.Z13_AUTHOR,1,50)
FROM LXX01.Z13
WHERE SUBSTR(LXX01.Z13.Z13_REC_KEY,0,9) = '<liga_de_bib_a_adm>
```

6.4. Interfaz

La interfaz del sistema *Auto-Circulación* consta de una sola pantalla donde se muestran los movimientos que se pueden seleccionar, un teclado numérico para ingresar los datos del usuario (número de credencial⁹ y contraseña) y el código de barras¹⁰. También cuenta con los botones limpiar pantalla y otro que toma por título el movimiento seleccionado que ejecutará el programa solo si se introdujeron todos los datos. Además cuenta con las instrucciones de uso y una barra de progreso que va señalando el avance del proceso. Cualquier error es informado al usuario a través de ventanas emergentes¹¹ (Ver Figura 6.4).

Para la Facultad de Derecho se solicitó una modificación donde para introducir el número de credencial y el número de adquisición sea solo a través

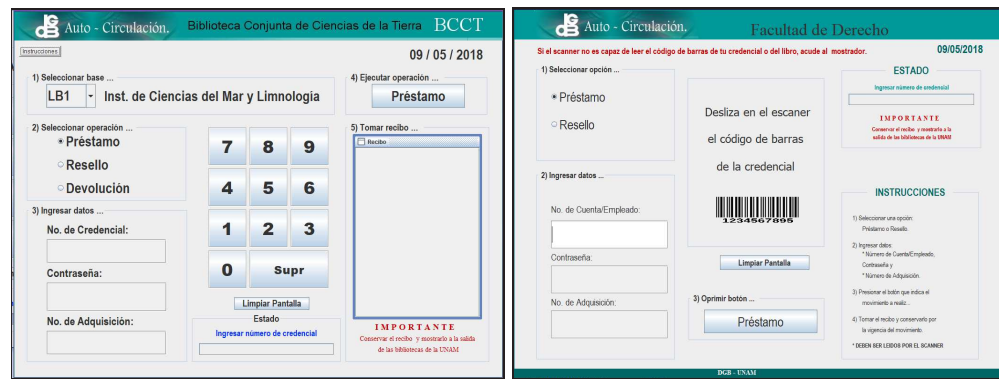
⁹ Esté se puede introducir a través del lector de código de barras, pero, para la Facultad de Derecho es obligatorio hacerlo de esta forma.

¹⁰ Regularmente es introducido a través del lector de código de barras.

¹¹ En Java *JOptionPanel*.



Figura 6.4: Interfaz del sistema *Auto-Circulación* en Biblioteca Central.



(a) BCCT.

(b) Facultad de Derecho.

Figura 6.5: Interfaz del sistema *Auto-Circulación* modificadas.

del lector de código de barras, esto debido a un mal uso del sistema con la interfaz original (Ver Figura 6.5b).

Para la Biblioteca Conjunta de Ciencias de la Tierra (BCCT) se incluye un combo de selección donde se especifica en cuál de las bibliotecas¹² se va a realizar el movimiento y la visualización del recibo que será impreso (Ver Figura 6.5a).

¹² Centro de Ciencias de la Atmósfera, Instituto de Geología, Instituto de Geofísica, Instituto de Ciencias del Mar y Limnología o Colección de tesis BCCT.

6.5. Situación actual

El sistema *Auto-Circulación* fue migrado de Power Builder a Java, las clases se reestructuraron agregando la clases *Configuracion* y *Conexion* para hacer el sistema configurable, también se incluyó la clase *Movimiento* para concentrar toda la información según se va obteniendo. Además se organizaron las clases en paquetes (Sección 4.3.4) como se muestra a continuación.

Paquete	Clase(s)
configuración	Configuracion Conexion
control	ImpresorRecibo
	ValidadorPrestamo
	ValidadorRenovacion
	ValidadorDevolucion
transacciones	Validador
	EjecutorPrestamo
	EjecutorRenovacion
	EjecutorDevolucion
interfaz	Interfaz
objetos	Movimiento

Para hacer configurable el sistema fueron necesarias más líneas de código, sin embargo, ya no se requiere modificar el código fuente cada que se va a instalar en una biblioteca distinta, solo hay que cambiar los datos de la dependencia en el archivo de configuración. Con esta organización de paquetes y clases la actualización y mantenimiento es más simple, además de que la programación se apega a una de las características más relevantes del Paradigma de Orientación Objetos que es el diseñar los programas de forma modular.

Además de las diferencias en la interfaz señaladas para la Facultad de Derecho se requirió una verificación adicional porque los usuarios tenían registrada como contraseña su mismo número de credencial, por lo tanto si esto ocurría debían actualizar su contraseña o no podían hacer uso del sistema.

El sistema ha sido instalado en la Biblioteca Central [2010], Biblioteca Conjunta de la Ciencias de la Tierra (BCCT) [2008], Centro de Investigaciones Interdisciplinarias en Ciencias y Humanidades (CEIICH) [2011], Instituto de Investigaciones Bibliotecológicas y de la Información (IIBI) [2008], Facultad de Derecho [2012], y la Facultad de estudios Superiores (FES) Aragón [2017]; sumando un total de seis dependencias.

La Figura 6.6 muestra el uso del sistema *Auto-Circulación* durante el periodo del 2013 al 2017 donde se puede observar que el mayor uso fue registrado en el año 2013 al solicitar solamente el 30 % de los préstamos.

En este capítulo se mencionaron la necesidad de crear este sistema y los requerimientos, se definieron los casos de uso, se señaló el diagrama E-R concreto de la base de datos del sistema *Aleph* que es consultado y se mostraron las características de las distintas interfaces existentes. Además de exponer la situación actual de este sistema en la UNAM.

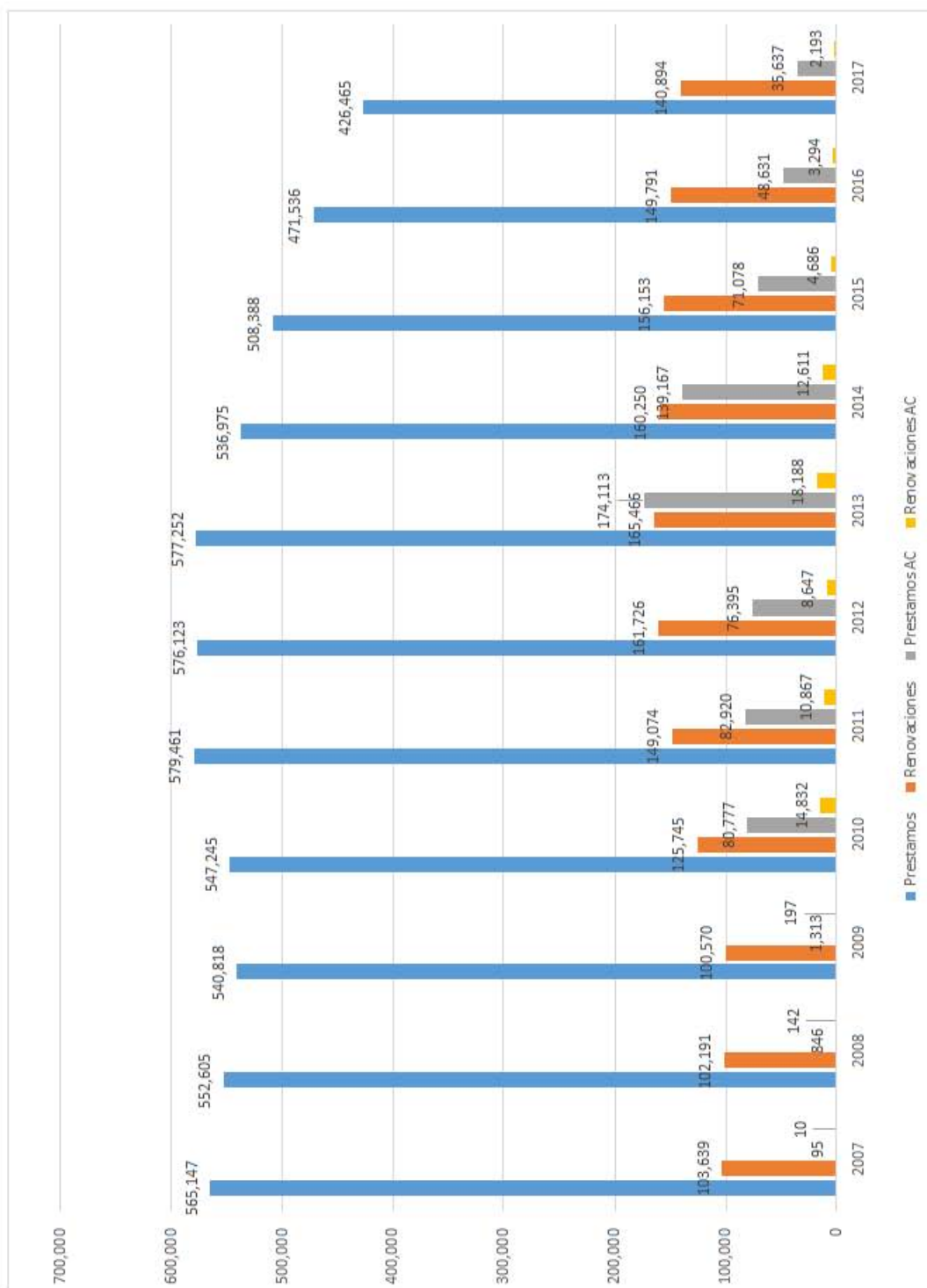


Figura 6.6: Estadísticas del sistema *Auto-Circulación*.

Capítulo 7

Sistema de reportes de productividad en catalogación: *Productividad en Catalogación*

Los reportes son parte esencial de la administración de un departamento, área incluyendo las empresas o instituciones de educación. Este sistema fue solicitado por la Facultad de Música quien requería un mecanismo que le permitiera conocer la productividad del personal en el área de catalogación, ya que no existe un informe que proporcione dicha información en el sistema *Aleph* se desarrolló el sistema *Productividad en Catalogación*.

7.1. Análisis de requerimientos

Para agregar un nuevo título en el acervo de la biblioteca es necesario que sea registrado y catalogado, al mismo tiempo en el sistema *Aleph* se registra quién y cuándo lo hizo. A partir de esta información es posible generar reportes que reflejen la productividad de los catalogadores y la cantidad de ejemplares procesados en determinado periodo de tiempo.

Entre los reportes que genera el sistema *Aleph* no existe uno que cubra

esta necesidad, por lo tanto, se diseñó un sistema que proporcionará esta información, a través de una interfaz simple que fuera capaz de almacenar la información obtenida en archivos para su consulta posterior, incluso fuera de la aplicación.

El sistema debe generar un listado¹ de los ejemplares que fueron registrados en el periodo de tiempo solicitado, el cual pueda manipularse de forma simple; debe incluir el número de matriz, la fecha, hora, la clave (usuario) y nivel del catalogador que registró el título. Además de mostrar por separado la productividad del personal medida por el conteo de registros trabajados desde su creación.

Este tipo de información es de interés para los jefes de las bibliotecas o el Jefe del Departamento de Procesos Técnicos para conocer la productividad de su personal.

La Figura 7.1 muestra las actividades que tiene que seguir el sistema para obtener la información solicitada, se inicia con el usuario que ingresa la fecha inicial y final del periodo a evaluar, entonces el sistema debe extraer todos los registros en el rango señalado, y para cada número de sistema distinto se debe verificar si es una catalogación nueva o una actualización, esto se realiza en base a la fecha y hora.

7.2. Diagrama E-R

Para el sistema Productividad en Catalogación solo es necesario consultar la tabla *Z106 ("CAT" FIELD)* de la base bibliográfica (*lxx01*), ésta contiene la información de las actualizaciones de los registros, la matriz del registro bibliográfico (*z106_rec_key*), fecha (*z106_update_date*) y hora (*z106_time*) de la actualización, el identificador (*z106_cataloger*) y nivel (*z106_level*) del catalogador que lo creó o modificó el registro (Ver Figura 7.2); Se considera el primer registro (según la fecha y hora) para cada matriz el alta del ejemplar.

¹ El listado de catalogaciones o registro de nuevos títulos en el periodo solicitado también es almacenado en un archivo en formato Excel en el directorio de instalación de la aplicación.

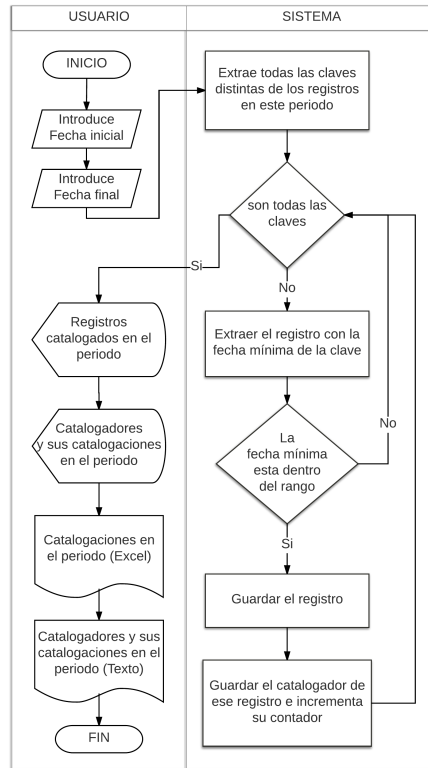


Figura 7.1: Diagrama de actividades.

Z106 - "CAT" FIELD	
NOT NULL	Z106_REC_KEY
	Z106_CATALOGER
	Z106_LEVEL
	Z106_UPDATE_DATE
	Z106_TIME

Figura 7.2: Diagrama E-R para *Productividad en Catalogación*.

7.3. Casos de uso

Para el sistema *Productividad en Catalogación* se definió un único caso de uso llamado **Generar reporte** (7.3.1), sin embargo este incluye a **Generar resumen** (7.3.2) (Ver Figura 7.3), a continuación se describen éstos.

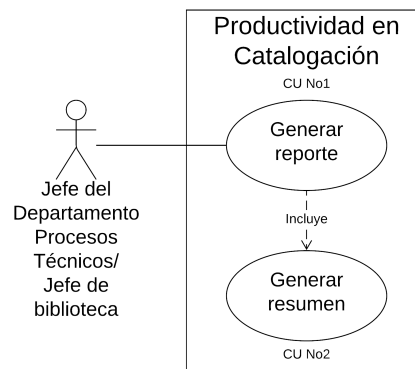


Figura 7.3: Casos de uso del sistema *Productividad en Catalogación*.

7.3.1. Generar reporte

Identificador

CU No1

Actores

Jefe del Departamento de Procesos Técnico de la DGB/Responsable de la biblioteca.

Descripción

Genera el reporte con todas las catalogaciones realizadas en el periodo de tiempo seleccionado, indicando para cada registro, número de matriz, fecha y hora en que se catalogó y la clave y nivel del catalogador.

Precondiciones

El sistema tiene conexión con el servidor.

Secuencia normal

1. Se introduce el periodo de tiempo del que se desea generar el reporte.
2. Se extraen todos los registros de los números de matriz (*z106_rec_key*) que fueron actualizados en el periodo señalado ordenados por su número de matriz, fecha de modificación (*z106_update_date*) y hora (*z106_time*).
3. Si la fecha del primer registro de cada número de matriz, pertenece al periodo solicitado es almacenado para formar parte del reporte.
4. Se busca el catalogador (*z106_cataloger*) en la lista de los mismos, Si existe se incrementa el contador de catalogaciones de este. Si no existe se registra en la lista de catalogadores y se inicializa el contador de este catalogador en uno.
5. Se despliega los registros catalogados en el periodo solicitado en la interfaz y se guarda la información en un archivo Excel con el nombre compuesto por la base y periodo consultado.
6. Si existen catalogaciones se realiza el resumen de la productividad (Generar resumen CU No2).

Postcondiciones

Se muestran los registros catalogados en el periodo solicitado, la información contenida es el número de matriz, la fecha, hora, catalogador y nivel del catalogador.

Excepciones

Cuando no hay ninguna catalogación en el periodo señalado muestra el aviso “NO EXISTE NINGUNA CATALOGACIÓN EN ESTE PERIODO”

Consultas a la base de datos

1.

```
SELECT LXX01.Z106.Z106_REC_KEY, LXX01.Z106.Z106_UPDATE_DATE,
       LXX01.Z106.Z106_TIME, LXX01.Z106.Z106_CATALOGER,
       LXX01.Z106.Z106_LEVEL
FROM LXX01.Z106
WHERE LXX01.Z106.Z106_REC_KEY IN
```

```
(SELECT DISTINCT(LXX01.Z106.Z106_REC_KEY) FROM LXX01.Z106
WHERE LXX01.Z106.Z106_UPDATE_DATE
BETWEEN '<fecha_inicial_del_periodo>' AND
        '<fecha_final_del_periodo>'
AND LXX01.Z106.Z106_CATALOGER IS NOT NULL)

ORDER BY LXX01.Z106.Z106_REC_KEY, LXX01.Z106.Z106_UPDATE_DATE,
LXX01.Z106.Z106_TIME
```

7.3.2. Generar resumen

Identificador

CU No2

Actores

El sistema.

Descripción

Genera el reporte de catalogaciones por catalogador.

Precondiciones

Tener almacenada la información obtenida mientras se generó el reporte de registros catalogados en un periodo solicitado.

Secuencia normal

1. Se vacía el contenido almacenado de los catalogadores y su contador para ser presentado en la interfaz.
2. Se guarda la información antes mencionada en un reporte titulado resumen.

Postcondiciones

Se muestran el listado de catalogadores y conteo de los registros realizados durante el periodo.

Excepciones

Ninguna.

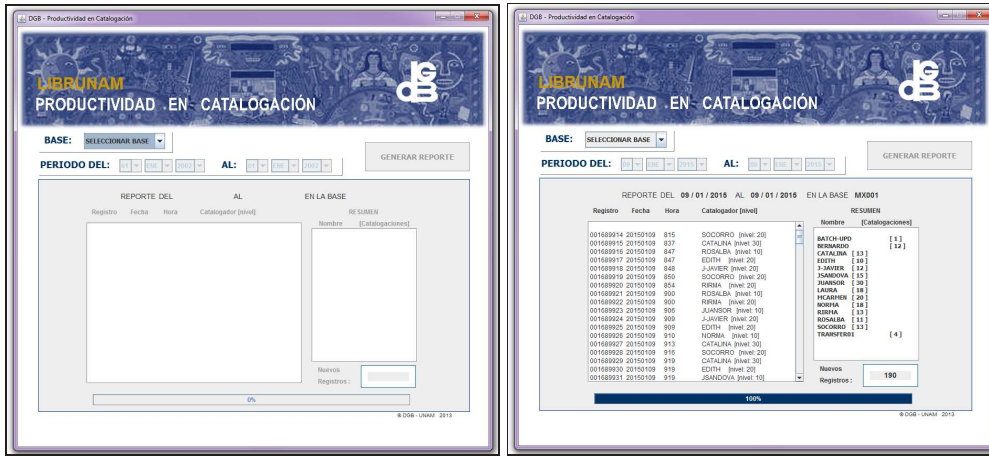


Figura 7.4: Interfaz del sistema *Productividad en Catalogación*.

7.4. Interfaz

La interfaz del sistema *Productividad en Catalogación* es una sola pantalla dividida en varias secciones, el extremo superior es el encabezado de la aplicación. A continuación se tienen los campos de entrada de datos donde se introduce la fecha inicial y final del periodo a evaluar y en algunos casos se debe seleccionar la base de datos *Aleph* porque como se explicó en la Sección 4.1 pueden existir varias de estas para una misma dependencia/biblioteca.

La parte inferior muestra los resultados, del lado izquierdo se muestran todas las catalogaciones con la información del número de matriz, la fecha y hora cuando se registró, el identificador y nivel de la persona que realizó cada una. Del lado derecho se despliega el listado de catalogadores y número de sus catalogaciones, así como el total del periodo. En el extremo inferior se cuenta con una barra de progreso para informar el avance, porque en ocasiones puede demorarse la visualización de los resultados dependiendo del tamaño de la base y el periodo seleccionado (Ver Figura 7.4).

7.5. Situación actual

Los paquetes y clases de esta aplicación están distribuidos de la siguiente manera.

Paquete	Clase(s)
configuración	Configuracion Conexión
control	Reporte
transacciones	Z106
interfaz	Interfaz

El sistema ha sido instalado en la Biblioteca Central para el Departamento de Procesos Técnicos, Bibliotecas Conjuntas de la Ciencias de la Tierra (BCCT) y la Facultad de Música quien solicitó el desarrollo de esta aplicación.

En este capítulo se mencionan las razones por la que se implementó este sistema, los requerimientos, los casos de uso identificados y desarrollados para resolverlo, las características de la única tabla que es consultada y detalles de la interfaz.

Conclusiones y trabajo futuro

Los sistemas descritos en este reporte de práctica profesional brindan el apoyo tecnológico para desempeñar las tareas descritas en cada sección de forma eficiente, cumpliendo el objetivo particular de cada uno de ellos.

A partir del análisis de la base de datos del Sistema Integrado de Gestión Bibliotecaria o de Bibliotecas (ILS²) *Aleph*, se determinan las consultas necesarias para obtener la información a través de una interfaz sencilla de utilizar.

El sistema *Vigilancia* muestra el estado en que se encuentra el préstamo de un ejemplar que se desea sacar de la biblioteca, así como la información del usuario que lo tiene asignado. De esta forma el personal de vigilancia puede autorizar la salida del libro si, el préstamo está vigente y la persona corresponde con la información y fotografía desplegada en el sistema; evitando el mal uso del servicio de préstamo como la extracción de material por un individuo distinto.

Se observó que el personal de vigilancia de las bibliotecas en el que se encuentra instalado pueden desempeñar su trabajo más rápido y fácil, ya que es más sencillo leer la información de la pantalla que revisar los sellos con la fecha de devolución en papeleta, pues en varias ocasiones la legibilidad se dificulta (1) por falta o exceso de tinta, (2) se enciman sellos, (3) no se muestran en un orden estricto o (4) falta de espacio, etcétera. Además permite verificar que el usuario que desea retirar el ejemplar es quien lo tiene registrado pues la fotografía y nombre del usuario también aparecen en pantalla.

² Por sus siglas en inglés *Integrated Library System*.

Auto-Circulación permite realizar préstamos, renovaciones y devoluciones según la configuración que se le establezca, lo hace de la misma forma que el cliente GUI de **Aleph** para que se reconozcan indistintamente en ambos sistemas y no exista pérdida de información ya que es la única aplicación que actualiza, agrega y elimina registros de la base de datos.

Para evitar el manejo de multas sólo se pueden realizar devoluciones en tiempo, es decir, que el préstamo se encuentre vigente en ese momento. El uso de este sistema ayuda a los usuarios, disminuyendo el tiempo de espera en horas pico o con la disposición del servicio de préstamo cuando hace falta personal en los mostradores; además de que no se genera el uso y el pago de otra licencia del sistema **Aleph**.

La razón por la que las renovaciones no son tan populares en el sistema **Auto-Circulación** es debido a que el sistema **Aleph** cuenta con la renovación vía Web en el OPAC sin la necesidad de asistir a la biblioteca. La devolución no se habilita en la mayoría de las instalaciones para que el bibliotecario valide la integridad del libro al momento de la devolución.

El sistema **Auto-Circulación** fue el único que reportó un mal uso, una de las razones es que es el único que se utiliza por usuarios de la biblioteca y no por el personal, en el caso particular de la Facultad de Derecho tenían como contraseña registrada el número de credencial y éste era publicado en algunos listados de la Facultad. Se modificó la programación para que si la contraseña era igual al número de cuenta no pudiera utilizar el sistema y se le notificaba al usuario que debía cambiar su contraseña. Más tarde también se solicitó que sólo se pudiera utilizar con la credencial.

El sistema **Productividad en Catalogación** es de uso exclusivo del Departamento de Procesos Técnicos o el Responsable de la biblioteca para verificar la producción en catalogaciones dado un determinado periodo de tiempo. En este sólo se hacen consultas a la base de datos y muestra el reporte del periodo que incluye el detalle de las catalogaciones así como un resumen por catalogador en pantalla, mismos que almacena en archivos para su futura consulta.

A partir de estos sistemas se consideró la parametrización de cualquier otro sistema a desarrollar que deba ser instalado en el SIBIUNAM, algunos

ejemplos son Inventarios y Reportes de Circulación.

Se espera que se extienda el uso de estos sistemas en todo el SIBIUNAM, en particular en aquellas dependencias que ya han solicitado los requerimientos para la implementación de alguno de éstos en sus bibliotecas.

Para cualquiera de los sistemas incluidos en este trabajo es necesaria la revisión del personal del Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas (DADMS) y si se requiere la adaptación a consecuencia de una actualización del sistema *Aleph*, empezando por probar y/o corregir la conexión³ con la base datos y analizar la estructura; ya que cuando se migró del sistema *Aleph* de la versión 16 a la versión 21 se incluyeron nuevas tablas y en algunas de las existentes aparecieron nuevos campos los cuales afectaron al sistema *Auto-Circulación* en particular.

Si surge una nueva versión de los sistemas operativos (Windows y Ubuntu) en los que son instaladas las aplicaciones contenidas en este trabajo y en general de todas las desarrolladas por el DADMS se realizan pruebas de funcionamiento y de ser necesario se tienen que implementar las modificaciones o actualizaciones que se requieran hasta lograr su completo funcionamiento.

Por otra parte, en ocasiones se solicita que alguno de estos sistemas tenga características específicas (distintas a las ya establecidas) para cierta dependencia, éstas pueden ser visuales o funcionales y requiere adaptar los sistemas. Por ejemplo, el sistema *Auto-Circulación* para la Facultad de Derecho fue modificado porque exigían que su uso fuera estrictamente con credencial por lo que fue necesario modificar la interfaz y reprogramar algunos de sus eventos.

La actualización de los sistemas hacia las nuevas versiones del lenguaje de programación Java o migración a otras tecnologías⁴ es una parte muy importante a considerar. Es preciso mencionar que los últimos programas elaborados en el DADMS se han realizado como aplicaciones Web, para evitar la instalación en diferentes sistemas operativos o versiones y que la actualización se refleje de forma inmediata. También es necesario analizar la

³ Cambio de versión del controlador de Java

⁴ Por mencionar algunas tecnologías Web están Java Servlets, JavaServer Faces y Struts

implementación con el apoyo de marcos de trabajo⁵.

De las aplicaciones contenidas en este reporte el DADMS consideró oportuna la implementación del sistema ***Auto-Circulación*** para dispositivos móviles, por lo que se hizo dicha implementación para Android.

También existe la posibilidad de adaptar las aplicaciones en caso de que cambie el ILS ***Aleph*** de uso en el SIBIUNAM, cabe mencionar que ya fue solicitado e implementado el sistema de ***Vigilancia*** para trabajar sobre la base de datos del ILS KOHA, actualmente en la etapa de pruebas.

⁵ En inglés Frameworks.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

A P E N D I C E S



Apéndice A

Subsistemas del SIBIUNAM [4]

1.- Bachillerato

Colegio de Ciencias y Humanidades Plantel Azcapotzalco
Colegio de Ciencias y Humanidades Plantel Naucalpan
Colegio de Ciencias y Humanidades Plantel Oriente
Colegio de Ciencias y Humanidades Plantel Sur
Colegio de Ciencias y Humanidades Plantel Vallejo
Dirección General de la Escuela Nacional Preparatoria
Dirección General del Colegio de Ciencias y Humanidades
Escuela Nacional Preparatoria Plantel 1 "Gabino Barreda"
Escuela Nacional Preparatoria Plantel 2 "Erasmus Castellanos Qunito"
Escuela Nacional Preparatoria Plantel 3 "Justo Sierra"
Escuela Nacional Preparatoria Plantel 4 "Vidal Catañeda y Nájera"
Escuela Nacional Preparatoria Plantel 5 "José Vasconcelos"
Escuela Nacional Preparatoria Plantel 6 "Antonio Caso"
Escuela Nacional Preparatoria Plantel 7 "Ezequiel A. Chávez"
Escuela Nacional Preparatoria Plantel 8 "Miguel E. Schultz"
Escuela Nacional Preparatoria Plantel 9 "Pedro de Alba"

2.- Licenciatura y Posgrado

Escuela Nacional de Enfermería y Obstetricia
Escuela Nacional de Estudios Superiores León
Escuela Nacional de Estudios Superiores Morelia

Facultad de Arquitectura (cuatro bibliotecas)
Facultad de Artes y Diseño (tres bibliotecas)
Facultad de Ciencias Políticas
Facultad de Ciencias
Facultad de Contaduría (dos bibliotecas)
Facultad de Derecho
Facultad de Economía (dos bibliotecas)
Facultad de Estudios Superiores Acatlán
Facultad de Estudios Superiores Aragón
Facultad de Estudios Superiores Cuautitlán (dos bibliotecas)
Facultad de Estudios Superiores Iztacala (dos bibliotecas)
Facultad de Estudios Superiores Zaragoza
Facultad de Filosofía y Letras
Facultad de Ingeniería (cuatro bibliotecas)
Facultad de Medicina (seis bibliotecas)
Facultad de Música
Facultad de Odontología (dos bibliotecas)
Facultad de Psicología (dos bibliotecas)
Facultad de Química (cuatro bibliotecas)
Facultad de Trabajo Social
Facultad de Veterinaria (siete bibliotecas)

3.- Investigación Científica

Biblioteca Conjunta de Ciencias de la Tierra (BCCT)
Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET)
Centro de Ciencias Genómicas (CCG)
Centro de Ciencias Matemáticas (CCM)
Centro de Investigaciones en Ecosistemas (CIECO)
Centro de Investigaciones en Geografía Ambiental (CIGA)
Centro de Nanociencias y Nanotecnología (CNyN)
Instituto de Astronomía (IA) (dos bibliotecas)
Instituto de Biología (IB) (tres bibliotecas)
Instituto de Biotecnología (IBT)
Instituto de Ciencias Físicas (ICF)
Instituto de Ciencias Nucleares (ICN)
Instituto de Ecología
Instituto de Energías Renovables (IER)
Instituto de Física (IF)

Instituto de Fisiología Celular (IFC)
Instituto de Geografía (IGG)
Instituto de Ingeniería (II)
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS)
Instituto de Investigaciones Biomédicas (IIB)
Instituto de Investigaciones en Materiales (IIM)
Instituto de Matemáticas (IM) (dos bibliotecas)
Instituto de Neurobiología (INB)
Instituto de Química (IQ)

4.- Investigación en Humanidades

Centro de Enseñanza de Lenguas Extranjeras (CELE)
Centro de Investigaciones Interdisciplinarias en Ciencias y Humanidades (CEIICH)
Centro de Investigaciones Multidisciplinarias sobre Chiapas y la Frontera Sur
(CIMSUR)
Centro de Investigaciones Sobre América del Norte (CISAN)
Centro de Investigaciones Sobre América Latina y el Caribe (CIALC)
Centro Peninsular en Humanidades y Ciencias Sociales (CEPHICIS)
Centro Regional de Investigaciones Multidisciplinarias (CRIM)
Instituto de Investigaciones Antropológicas (IIA)
Instituto de Investigaciones Bibliotecológicas y de la Información (IIBI)
Instituto de Investigaciones Económicas (IIEc)
Instituto de Investigaciones Estéticas (IIE)
Instituto de Investigaciones Filológicas (IIFl)
Instituto de Investigaciones Filosóficas (IIFs)
Instituto de Investigaciones Históricas (IIH)
Instituto de Investigaciones Jurídicas (IIJ)
Instituto de Investigaciones sobre la Universidad y la Educación (IISUE)
Instituto de Investigaciones Sociales (IIS)
Programa Universitario de Estudios de Género (PUEG)
Programa Universitario de Estudios sobre la Ciudad (PUEC)
Programa Universitario de Estudios de la Diversidad Cultural y la
Interculturalidad(PUIC)
Unidad Académica de Estudios Regionales (UAER)

5.- Extensión y Administración

Centro de Enseñanza para Extranjeros (CEPE)

Centro Cultural Universitario Tlatelolco
Centro Universitario de Estudios Cinematográficos (CUEC)
Centro Universitario de Teatro (CUT)
Coordinación de Estudios de Posgrado
Coordinación de Universidad Abierta y Educación a Distancia (CUAED)
Dirección General de Actividades Cinematográficas
Dirección General de Artes Visuales
Dirección General de Bibliotecas (Biblioteca Central)
Dirección General de Cómputo y de Tecnologías de Información y
Comunicación (DGTIC)
Dirección General de Deporte Universitario
Dirección General de Divulgación de la Ciencia (DGDC)
Dirección General de Estudios de Legislación Universitaria (DGELU)
Dirección General de Servicios Médicos (DGSM)

Apéndice B

Notación para diagramas UML [23]

Los símbolos utilizados para los tipos de diagramas UML descritos a continuación son consistente en las distintas fuentes consultadas

Diagramas de caso de uso

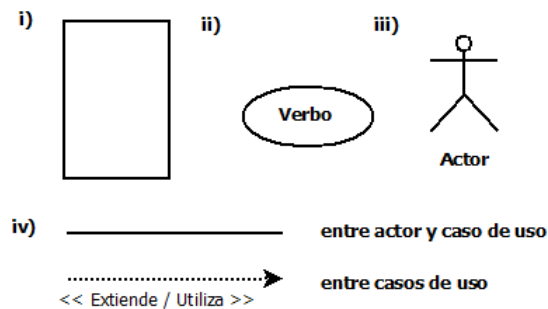


Figura B.1: Notación UML para representar diagramas de caso de uso.

Sistema: Los límites del sistema se dibujan utilizando un rectángulo que contiene los casos de uso y el usuario esta fuera del rectángulo. (Ver Figura B.1, inciso i)).

Casos de uso: Los casos de uso se dibujan utilizando óvalos etiquetados con verbos que representan las funciones del sistema. (Ver Figura B.1, inciso ii)).

Actores: Los actores son los usuarios del sistema y son representados con el estereotipo¹ de *figura de palo*. Cuando un sistema es actor de otro sistema, se etiqueta el estereotipo con el nombre del sistema (Ver Figura B.1, inciso iii)).

Relaciones: Las relaciones se señalan entre el actor y el caso de uso con una línea simple. (Ver Figura B.1, inciso iv)).

Para las relaciones entre casos de uso se utilizan líneas punteadas con flecha y las etiquetas <<utiliza>> o <<extiende>> (Ver Figura B.1, inciso iv)).

Diagramas de actividad

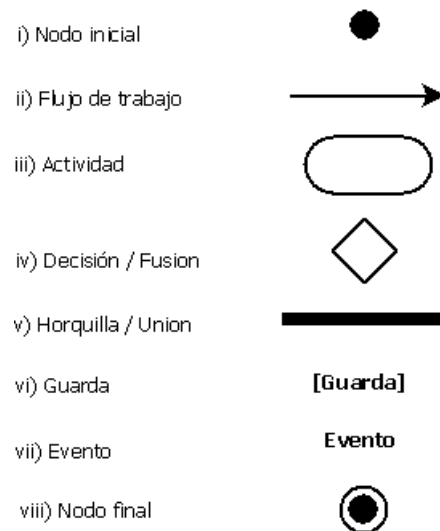


Figura B.2: Notación UML para representar diagramas de actividad.

Nodo inicial: Indica donde comienza el flujo de trabajo (Ver Figura B.2, inciso i)).

¹ Los estereotipos son extensiones de UML. Una de las muchas ventajas de UML es que permite definir constructores adicionales que no son parte de UML, pero que son necesarios para hacer el modelo de un sistema con precisión [22].

Flujo de control: Una flecha indica la dirección del flujo de trabajo (Ver Figura B.2, ii)).

Actividad: Indica un paso en el proceso (Ver Figura B.2, inciso iii)).

Decisión: Un símbolo en forma de rombo, indicando una elección. El flujo de trabajo procederá por el numero de rutas posibles, según la condición (Ver Figura B.2, inciso iv)).

Fusión: Se utiliza este símbolo para apegarse al estándar UML cuando varios flujos alternativos llevan a la misma actividad (Ver Figura B.2, inciso iv)).

Horquillas y Unión: Las barras se utilizan para documentar actividades paralelas que en UML significa que son ejecutadas a la vez o una tras otra, es decir, una horquilla indica el punto a partir del cual distintas actividades pueden comenzar en cualquier orden y una unión indica que el flujo de trabajo solo puede comenzar una vez que las actividades que llegan a éste se han completado (Ver Figura B.2, inciso v)).

Condición: Una condición se muestra entre corchetes, si es verdadera, el flujo de trabajo puede continuar. Normalmente están unidas a flujos de control que salen de un símbolo de decisión (Ver Figura B.2, inciso vi)).

Evento: El evento debe ocurrir para que el flujo continúe, tiene mayor implicación que una condición forzando a que la actividad anterior termine y se indica sin usar corchetes (Ver Figura B.2, inciso vii)).

Nodo final: Indica el final del proceso (Ver Figura B.2, inciso viii)).

Un diagrama de actividad puede estar dividido en *calles* o *particiones*, agrupando elementos en columnas, es decir, rectángulos.

Diagramas de clases

Clase: Se ilustran con un rectángulo dividido en tres secciones. En la primera sección se coloca el nombre de la clase, éste debe ir centrado, en negrilla y con la primera letra mayúscula, la lista de atributos en la segunda sección y las operaciones en la tercera (Ver Figura B.3, inciso i)).

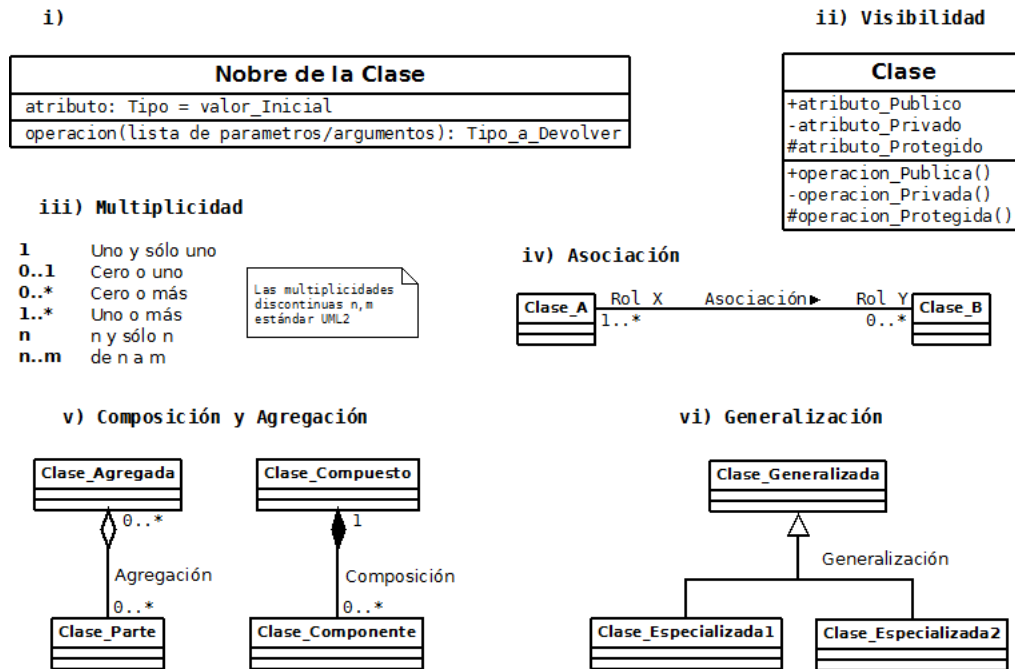


Figura B.3: Notación UML para representar diagramas de clases.

Visibilidad: Se utilizan marcas que significan quien puede tener acceso a la información contenida. *Publico* (+) permite al resto de las clases ver esta información. *Protegido* exclusivo de la clase. *Privado* (-) oculta la información para todos aquellos fuera de la clase. (#) es visible en la clase en la cual se definió como dentro de las subclases de dicha clase (Ver Figura B.3, inciso ii)).

Multiplicidad: Indica el número de instancias de un clase ligada a una instancia de otra clase. Se coloca al final de la asociación (Ver Figura B.3, inciso iii)).

Asociaciones: Representan relaciones estáticas entre clases. El nombre de la asociación puede ser colocado por encima o debajo de la línea de asociación y se utiliza una flecha rellena para indicar la dirección de la relación. Los roles se deben colocar cerca del final de la asociación y éstos representan como se ven las clases entre sí. Es poco común que se nombre a ambas, la asociación y los roles (Ver Figura B.3, inciso iv)).

Agregación: Es el termino del UML para la relación *parte-todo*, y aunque la clase *todo* desempeña un rol más importante, ambas clases son independien-

tes una de la otra. El diamante se coloca del extremo del *todo*, y éste debe ir vacío (Ver Figura B.3, inciso v)).

Composición: Es un tipo especial de agregación que denota una fuerte propiedad entre la clase *todo* y la *parte*, es posible que cada parte pertenezca a un solo *todo*, y si el *todo* se borra las partes también son eliminadas. El diamante se coloca del extremo del *todo*, y éste debe ir relleno (Ver Figura B.3, inciso v)).

Generalización: La herencia es una función requerida de la orientación a objetos, éste es un caso especial de generalización. La notación UML para la generalización es un triángulo vacío. Es una relación entre dos clases una de las clases es una especialización de otra (Ver Figura B.3, inciso vi)).

Notas: Son utilizadas para agregar comentarios a un diagrama UML, se representan por medio de un rectángulo con la esquina superior derecha doblada. Y se unen utilizando una línea punteada desde la nota hasta el elemento al cual hace referencia (Ver Figura B.3, inciso iii)).

Diagramas de paquete

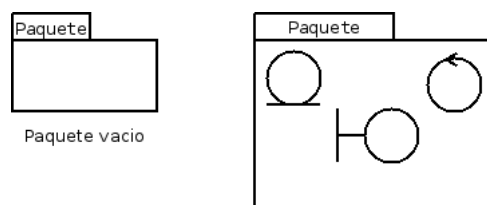


Figura B.4: Notación UML para representar diagramas de paquetes.

La notación para un paquete es un rectángulo con una etiqueta (folder) que contiene las clases que lo integran. (Ver Figura B.4).

Diagramas de secuencia

Usuarios: Se utiliza el estereotipo de *figura de palo* (Ver Figura B.5, inciso i)).

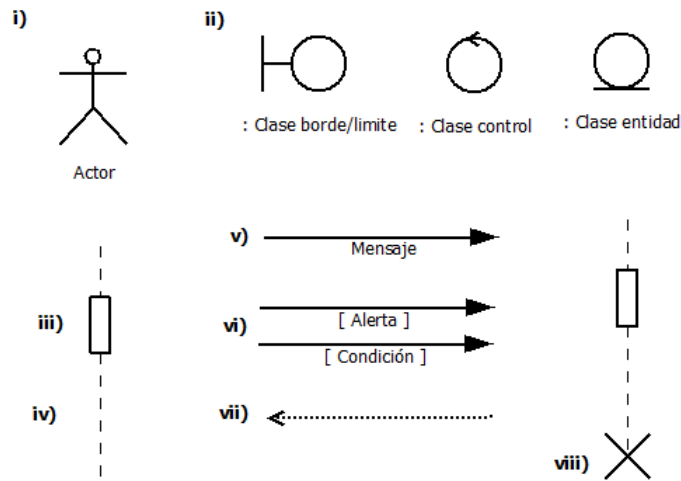


Figura B.5: Notación UML para representar diagramas de secuencia.

Clases: Las clases utilizan los estereotipos de la Figura 3.2 además de indicar el nombre de la clase de la cual se trata, precedida de dos puntos (Ver Figura B.5, inciso ii)).

Activación: Cuando un objeto está activo se denota por un rectángulo (*recuadro de activación*) (Ver Figura B.5, inciso iii)).

Líneas de vida: Son las líneas punteadas verticales que representan la presencia del objeto en el tiempo (Ver Figura B.5, inciso iv)).

Mensajes: Son flechas que representan la comunicación entre objetos (Ver Figura B.5, inciso v)).

Alerta/Condición: Solo si la alerta es verdadera el mensaje es enviado. Se escribe entre corchetes (Ver Figura B.5, inciso vi)).

Respuesta: Son flechas de punta abierta punteada (Ver Figura B.5, inciso vii)).

Destrucción: La destrucción de un objeto se representa con una "X" en negrita. Normalmente ocurre después de dar una respuesta (Ver Figura B.5, inciso viii)).

Apéndice C

Relaciones entre las bases de datos del sistema *Aleph* [19]

Los diagramas muestran las relaciones entre las bibliotecas ALEPH (identificadas con el prefijo USM) y las tablas de cada una de ellas. En algunos casos también se hace referencia al módulo que las utiliza.

Se incluye el diagrama en general de todas las bibliotecas y los que fueron estudiados para este trabajo.

Algunas notas que se deben tener en cuenta al revisar estos diagramas son [19]:

- La línea punteada es utilizada para indicar una relación indirecta.
- Los registros marcados con un asterisco (*) actualmente no existen en la distribución de los archivos USM, son incluidos como ejemplos en este documento.
- Para información detallada de cada campo en cada registro, incluyendo posibles valores hay que consultar la descripción de las tablas en "Collected Oracle tables" [17].

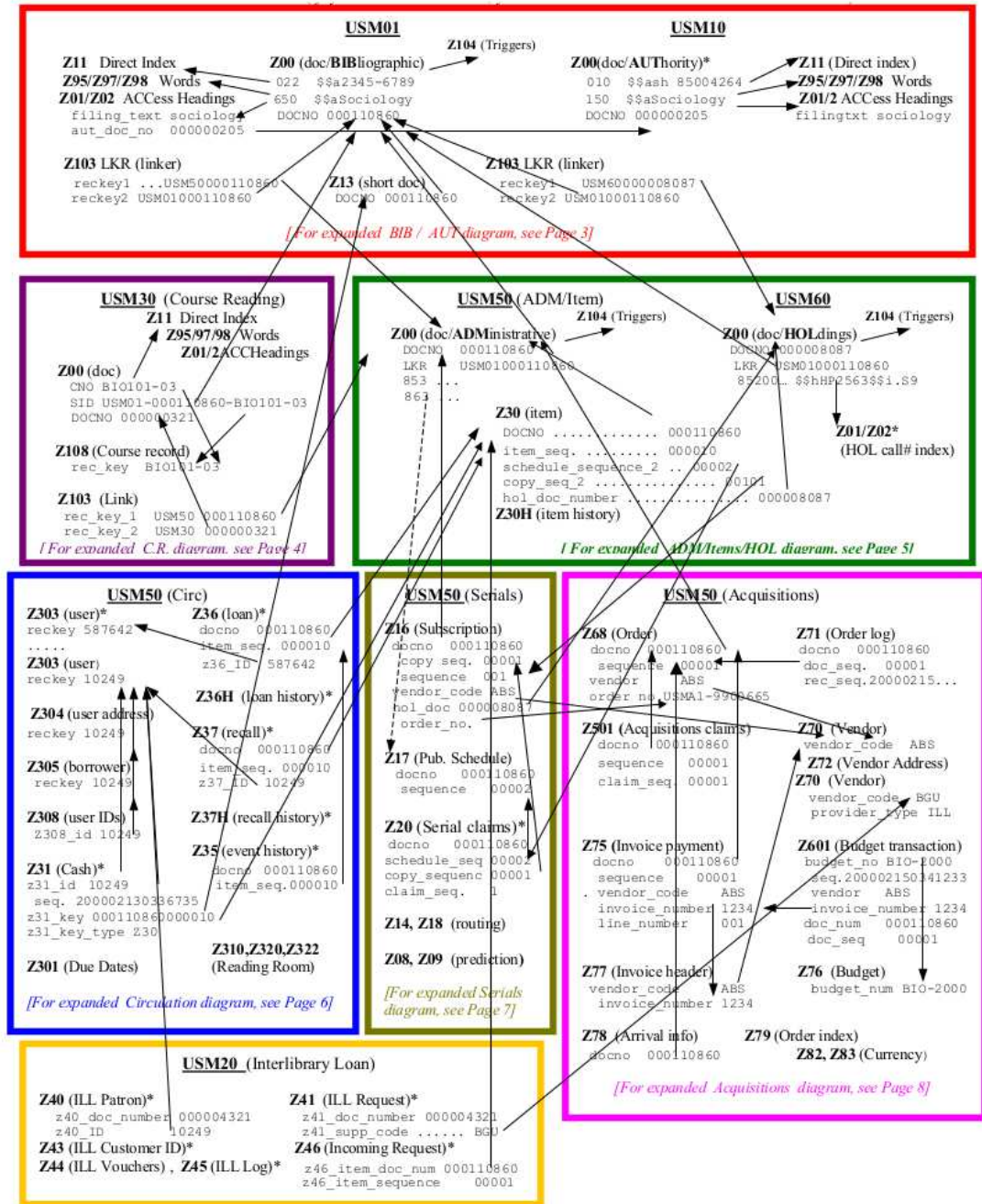


Figura C.1: Relación entre bases y tablas ALEPH [19].

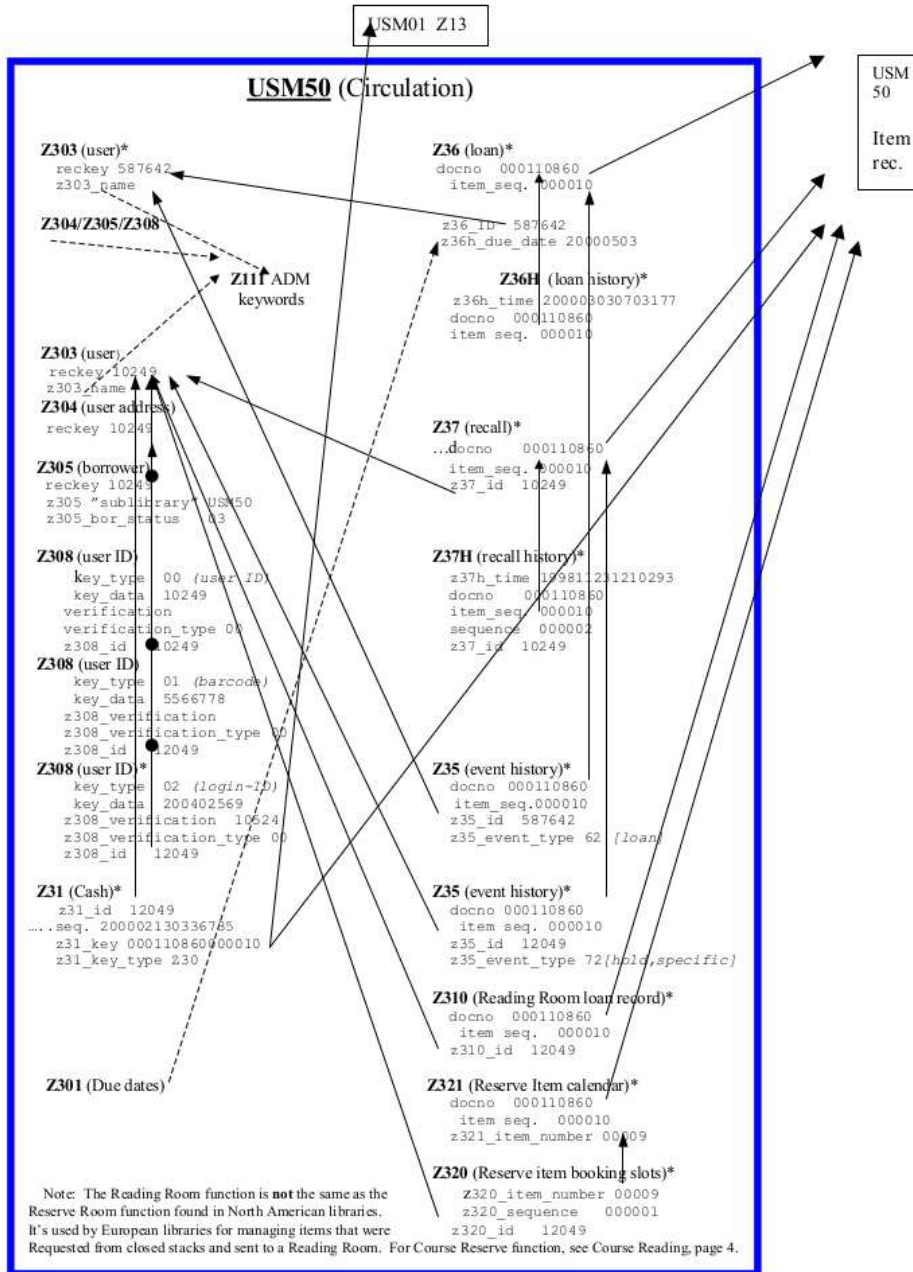


Figura C.2: Relación entre tablas de la base administrativa [19].

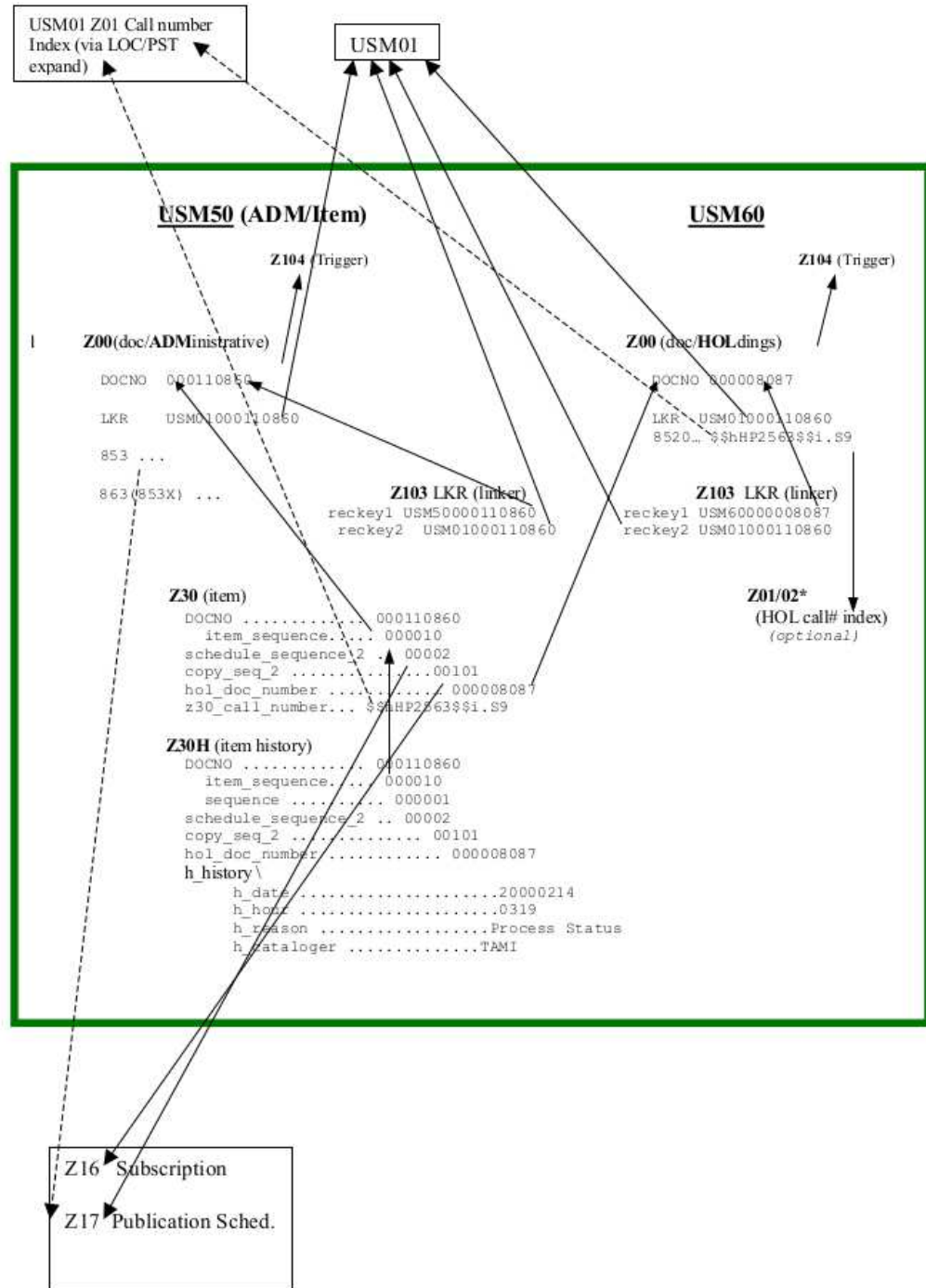


Figura C.3: Relación entre tablas para ejemplares [19].

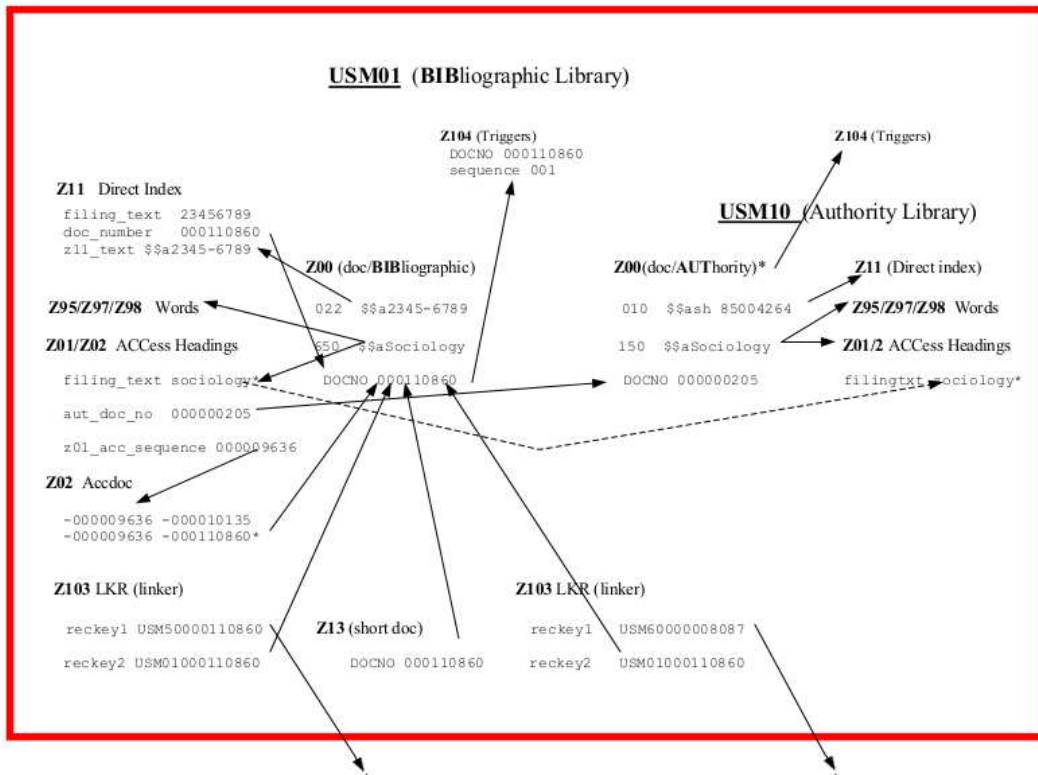


Figura C.4: Relación entre tablas de la base bibliográfica [19].

Apéndice D

Diagramas E-R de la base de datos del sistema *Aleph*[18]

Los diagramas a continuación son un mapa que muestra todas las relaciones que tiene la entidad central.

Las convenciones utilizadas en los diagramas son las siguientes:

- Cada uno de los diagramas tiene una entidad central rodeadas de todas aquellas con las que se relaciona. La entidad central siempre está de color blanco.
- Otras entidades que están almacenadas en la misma base que la entidad central están en color amarillo, si no lo están tienen un color diferente.
- Cada relación es nombrada con los nombres de los campos relacionados utilizando la siguiente convención NombreDeLaTablaOracle1_Campo1_ NombreDeLaTablaOracle2_Campo2
- La mayoría de las relaciones son uno-a-varios. Donde el símbolo parecido a una Y identifica el lado de la conexión “a-varios”. Cuando la relación es uno-a-uno, tres asteriscos (***) son agregados al nombre de la relación.
- Las líneas punteadas simbolizan relaciones opcionales, mientras que la línea continua representa las relaciones obligatorias.

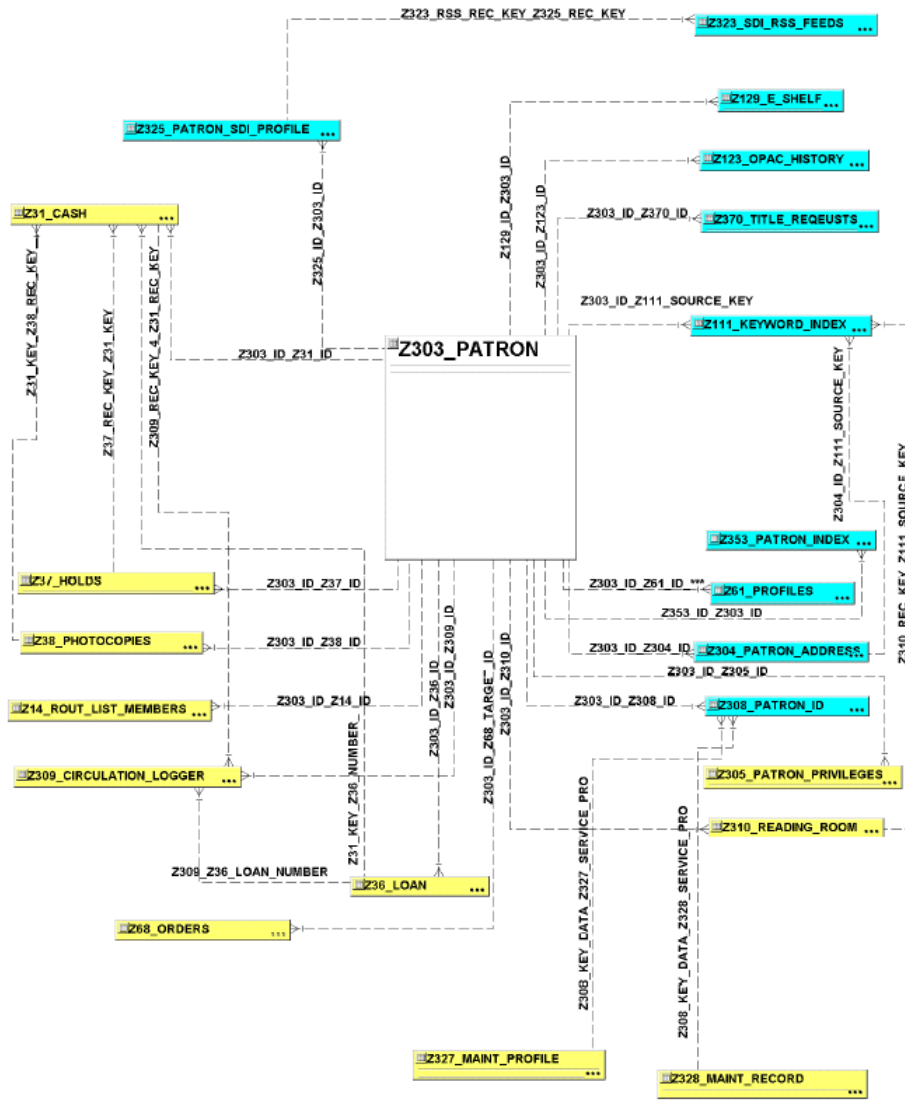


Figura D.1: Diagrama E-R: Usuarios [18]

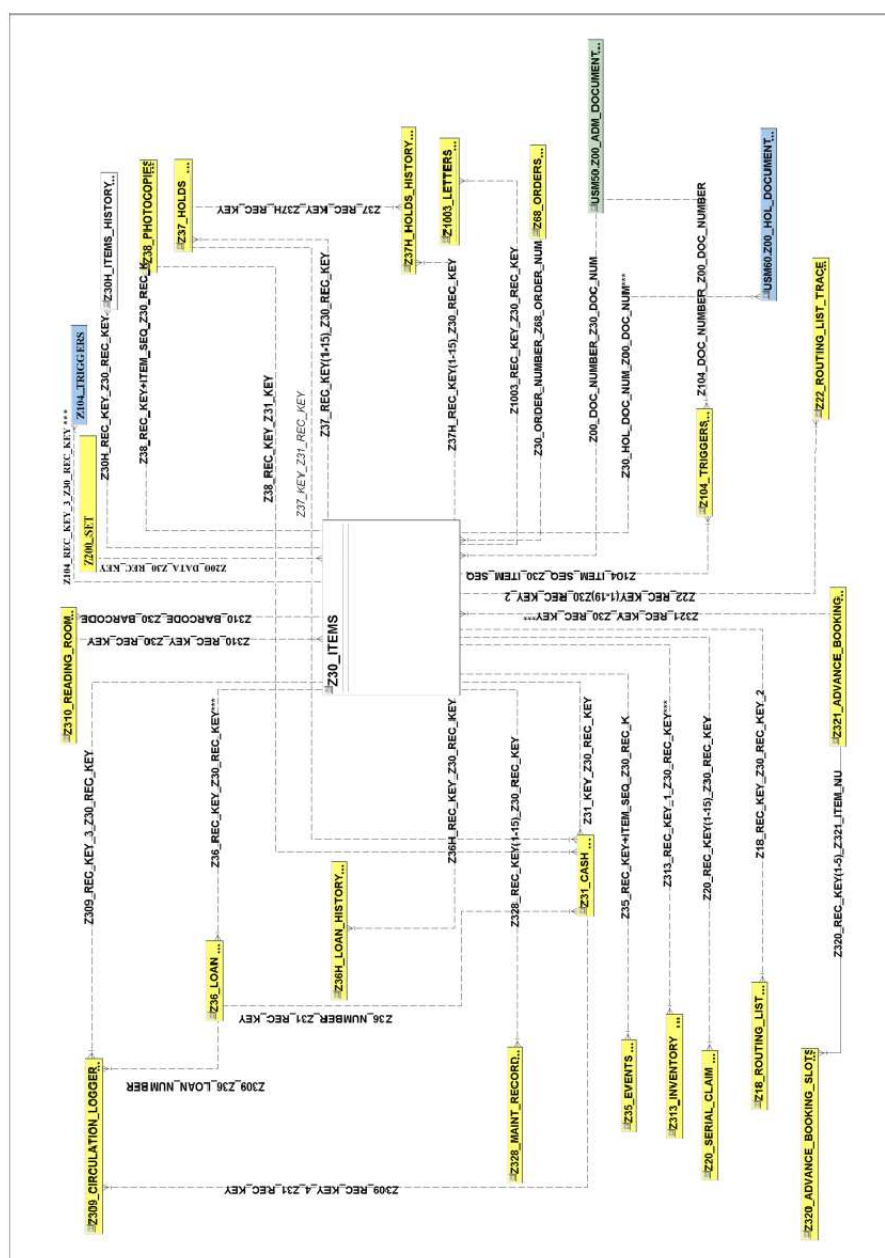


Figura D.2: Diagrama E-R: Circulación-Ejemplares [18]

Apéndice E

Tablas *Aleph*

Las tablas de Aleph son archivos de texto que contienen información de configuración que utiliza el sistema *Aleph* y por lo tanto los sistemas de este trabajo consultan. Las tablas consultadas son tab16, tab15 y tab31.

La documentación de las mismas se encuentra como comentario al inicio de cada uno de los archivos.

E.1. Tab16 (Due dates, fines & limits)

La información consultada es:

COLUMNA	DESCRIPCIÓN
2	Estatus del ejemplar
4	Estatus del usuario
6	Días de préstamo para la combinación ejemplar-usuario
17	Cantidad de préstamos permitidos para la combinación ejemplar-usuario
21	Cantidad de renovaciones permitidas para la combinación ejemplar-usuario

La apariencia de la tabla es la siguiente:

```

! 1  2  3  4  5    6    7  8  9   10  11  12  13  4      21
!!!!-!!-!!-!!-!!-!!!!!!-!!-!!-!!!!-!!!!-!!!!-!!!!-!!-!!-!!-... -!! ...
16A  10 ## 01 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 02 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 03 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 04 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 05 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 06 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 07 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 08 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 09 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 10 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 11 + 00000007 00 A 2300 0000 00000 010 000 4      02
16A  10 ## 12 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 13 + 00000090 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 14 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 15 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 16 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 17 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 18 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  10 ## 99 + 00000007 00 A 2300 0000 00200 003 000 4      02
16A  72 ## 01 + 00000000 00 + 0000 0000 00200 003 000 4
16A  72 ## 02 + 00000000 00 + 0000 0000 00200 003 000 4
16A  72 ## 03 + 00000000 00 + 0000 0000 00200 003 000 4
16A  72 ## 10 + 00000000 00 + 0000 0000 00200 003 000 4
16A  72 ## 11 + 00000000 00 + 0000 0000 00200 010 000 4

```

E.2. Tab15 (Item Statuses)

La información consultada es:

COLUMNA	DESCRIPCIÓN
2	Estatus del ejemplar
5	Nombre o descripción del estatus
6	Permiso de préstamo
7	Permiso de renovación

La apariencia de la tabla es la siguiente:

Apéndice F

Lista de acrónimos

Aleph	Automated Library Expandable Program
API	Application Programming Interface
DADMS	Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas
DBMS	DataBase Management System
DDL	Data Definition Language
DGB	Dirección General de Bibliotecas
DML	Data Manipulation Language
GUI	Graphic User Interface
IDE	Integrated Development Environment
ILS	Integrated Library System
ISSN	International Standard Serial Number
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
MARC	Machine Readable Cataloging
MVC	Model–View–Controller
MVC	Modelo-Vista-Controlador
OMG	Object Management Group
OMT	Object Modeling Technique
OPAC	Online Public Access Catalog
PIB	Prestamo interbibliotecario
SIBIUNAM	Sistema Bibliotecario de la UNAM
SQL	Structured Query Language

RDBMS	Relational DataBase Management System
UML	Unified Modeling Language
UNAM	Universidad Nacional Autónoma de México
USDP	Unified Software Development Process

Bibliografía

- [1] Real Academia Española. *Real Academia Española, Consultas lingüísticas*.
Disponible en: <http://www.rae.es> Consultado: Abril 2017.
- [2] Library of Congress *MARC Standards*.
Disponible en: <http://www.loc.gov/marc/> Consultado: Septiembre 2017.
- [3] Dirección General de Bibliotecas. *Dirección General de Bibliotecas*.
Disponible en: <http://www.dgb.unam.mx/> Consultado: Febrero 2018.
- [4] Dirección General de Bibliotecas. *Subsistemas Bibliotecas SIBIUNAM*.
Disponible en: <http://www.bibliotecas.unam.mx/index.php/quien-es-somos/-bibliotecas-unam> Consultado: Febrero 2018.
- [5] Dirección General de Bibliotecas. *Estructura orgánica*.
Disponible en: <http://www.dgb.unam.mx/index.php/quienes-somos/estructura-organica> Consultado: Febrero 2018.
- [6] Dirección General de Bibliotecas. *Resultados Censo 2017. Usuarios*.
Disponible en: <http://dgb.unam.mx/index.php/sistema-bibliotecario/censo-sibiunam/resultados-censo-2017/364-5-usuarios> Consultado: Febrero 2018.
- [7] Dirección General de Bibliotecas. *Resultados Censo 2017. Bibliotecas*.
Disponible en: <http://dgb.unam.mx/index.php/sistema-bibliotecario/censo-sibiunam/resultados-censo-2017/360-1-bibliotecas> Consultado: Febrero 2018.

- [8] Dirección General de Bibliotecas. *Subsistemas Bibliotecas SIBIUNAM*. Disponible en: <http://www.bibliotecas.unam.mx/index.php/quien-es-somos/-bibliotecas-unam> Consultado: Febrero 2018.
- [9] Dirección General de Bibliotecas. *Acerca de Bibliotecas SIBIUNAM*. Disponible en: <http://www.bibliotecas.unam.mx/index.php/quien-es-somos/acerca-de-bibliotecas-unam> Consultado: Marzo 2018.
- [10] Dirección General de Bibliotecas. *Numeralia DGB*. Disponible en: <http://dgb.unam.mx/index.php/quienes-somos/estadisticas/numeralia-dgb/390-numeralia-dgb-2018> Consultado: Marzo 2018.
- [11] Dirección General de Bibliotecas. *Reglamento General del Sistema Bibliotecario y de Información de la Universidad Nacional Autónoma de México*. Disponible en: <http://www.dgb.unam.mx/index.php/sistema-bibliotecario/reglamento-del-sistema-bibliotecario> Consultado: Marzo 2018.
- [12] Cabrera Heredia, G. & Jimenez Bernal, M. A. (2007). *Migración del módulo de préstamo al sistema Aleph 500 en bibliotecas de la UNAM*. (Tesis de Licenciatura). Universidad Nacional Autónoma de México. Facultad de Ingeniería. Ciudad Universitaria, México. Disponible en: <http://132.248.9.195/pd2008/0622958/Index.html>
- [13] ExLibris Aleph. (2012). *Aleph 21 system administration guide, Versión 21*.
- [14] ExLibris Aleph. (2012). *System administrator's guide – system overview, Versión 21*.
- [15] ExLibris Aleph. (2014). *Requirements for Aleph installation, Versión 21*.
- [16] ExLibris Aleph. (2012). *List of Oracle tables, Versión 21*.
- [17] ExLibris Aleph. (2012). *Collected Oracle tables, Versión 21*.
- [18] ExLibris Aleph. (2012). *Entity relationship diagrams, Versión 21*.
- [19] ExLibris Aleph. (2001). *Oracle table relationships, Releases 12.4-14.2*.

- [20] ExLibris. *Aleph integrated library system*.
Disponible en: <http://www.exlibrisgroup.com/category/Aleph>
Consultado: Septiembre 2017.
- [21] UNAM. *Catálogo de libros - Biblioteca Central - UNAM*.
Disponible en: <http://informatica.dgbiblio.unam.mx:8991> Consultado: Abril 2018.
- [22] Schach, Stephen R. (2004). *Análisis y diseño orientado a objetos con UML y el Proceso Unificado*. 1a ed. Edición, Impreso en México : McGraw-Hill.
- [23] Howard Podeswa. (2010). *UML*. Madrid : Anaya Multimedia.
- [24] Object Management Group. *Unified Modeling LanguageTM (UML®) Resource Page*.
Disponible en: <http://www.uml.org/> Consultado: Julio 2017
- [25] Object Management Group. *Introduction to OMG's Unified Modeling LanguageTM (UML®)*.
Disponible en: <http://www.uml.org/what-is-uml.htm> Consultado: Mayo 2017
- [26] SPARX SYSTEMS. *The Unified Modeling Language (UML)*.
Disponible en: <http://www.sparxsystems.com/uml-tutorial.html>
Consultado: Mayo 2017
- [27] Martín Sierra, Antonio J. (2010). *Programador certificado Java 2, curso práctico*. 3a ed. Edición, Impreso en México : Alfaomega, RA-MA.
- [28] Oracle corporation (US). *Qué es Java?*.
Disponible en: https://www.java.com/es/about/whatis_java.jsp
Consultado: Junio 2016
- [29] Oracle corporation (US). *Conozca más sobre la tecnología Java*.
Disponible en: <https://www.java.com/es/about/> Consultado: Enero 2018
- [30] Oracle. *Java SE at a Glance*.
Disponible en: <http://www.oracle.com/technetwork/java/javase/overview/index.html> Consultado: Diciembre 2017

- [31] Oracle. *Java SE Technologies*.
Disponible en: <http://www.oracle.com/technetwork/java/javase/tech/index.html> Consultado: Diciembre 2017
- [32] Oracle. *Java Documentation, Java™ plataform overview*.
Disponible en: <http://docs.oracle.com/javase/8/docs/technotes/guides/index.html> Consultado: Diciembre 2017
- [33] Oracle. *Java Documentation, Java platform standard edition 8 documentation*.
Disponible en: <http://docs.oracle.com/javase/8/docs/index.html>
Consultado: Diciembre 2017
- [34] Oracle. *Java Documentation, Description of Java conceptual diagram*.
Disponible en: http://docs.oracle.com/javase/8/docs/technotes/guides/desc_jdk_structure.html Consultado: Diciembre 2016
- [35] Oracle Java Documentation. *The Java™ Tutorials*.
Disponible en: <https://docs.oracle.com/javase/tutorial/> Consultado: Enero 2017
- [36] NetBeans. *NetBeans IDE Features*.
Disponible en: <https://netbeans.org/features/index.html> Consultado: Octubre 2016
- [37] Loney, Kevin. (2000). *Oracle 8, Manual del administrador*. Madrid ; México : McGraw-Hill.
- [38] Ramakrishnan, Raghu; Gehrke, Johannes. (2007). *Sistemas de gestión de bases de datos*. Madrid; México: McGraw-Hill.
- [39] Silberschatz, Abraham; Korth, Henry F.; Sudarshan, S. (2006). *Fundamentos de bases de datos*. Madrid: McGraw-Hill.
- [40] <http://docs.oracle.com> *ORACLE® DATABASE, 2 DAY + JAVA DEVELOPER'S GUIDE*.
Disponible en: http://docs.oracle.com/cd/E11882_01/appdev.112/e12137.pdf Consultado: Mayo 2017
- [41] <http://docs.oracle.com> *ORACLE® DATABASE, JDBC DEVELOPER'S GUIDE*.

Disponible en: http://docs.oracle.com/cd/E11882_01/java.112/e16548.pdf Consultado: Octubre 2016