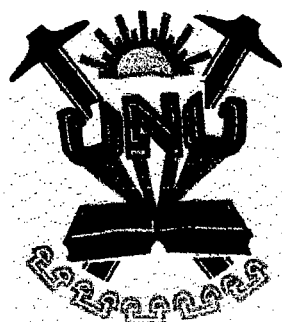


UNIVERSIDAD NACIONAL DE UCAYALI

Facultad de Ingeniería de Sistemas

Escuela de Formación Profesional de Ingeniería de Sistemas



1808T

**SISTEMA WEB CON TECNOLOGÍA J2EE PARA LA
ADMINISTRACIÓN EN LA BIBLIOTECA DE LA UNIVERSIDAD
NACIONAL DE UCAYALI**

Tesis

**Para Optar al Título de:
INGENIERO DE SISTEMAS**

**AUTORES : Bach. LUNA VELASQUEZ, Javier
Bach. TOMINAGA GARCIA, Yumi**

PUCALLPA – PERU

2008

**SISTEMA WEB CON TECNOLOGÍA J2EE PARA LA
ADMINISTRACIÓN EN LA BIBLIOTECA DE LA UNIVERSIDAD
NACIONAL DE UCAYALI**

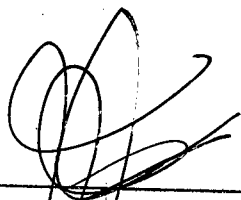
Por:

Bach. Javier Luna Velásquez
Bach. Yumi Tominaga García

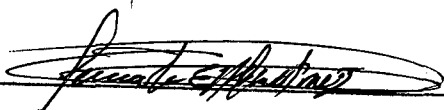
Aprobado



Ing. Romel Pinedo Ríos

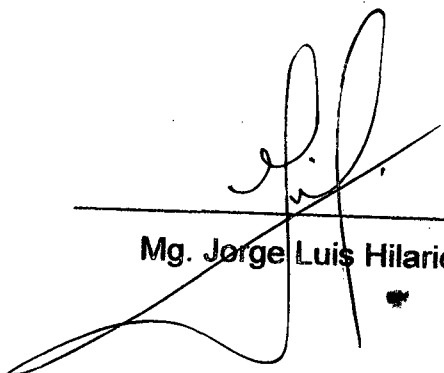


Ing. Nilton César Ayra Apac



Lic. Fernando Alcántara Infantes

Asesor:



Mg. Jorge Luis Hilario Rivas

**A nuestro Señor creador, por
permitirme disfrutar una vida llena de
alegrías e iluminar el sendero por
donde voy.**

**A mis abuelitos Marcos y Regina
que desde el cielo me cuidan,
protegen y guían mis pasos para ser
cada día mejor.**

**A mis queridos padres Raúl y
Luisa, y mi hermana Marita
por darme su apoyo
incondicional en todas las
cosas que he emprendido en mi
vida y por brindarme su amor
en todo momento.**

Javier

A mis padres Carlos Alberto y Elder Gladis, por el apoyo y amor incondicional que me dan para lograr mis objetivos. A ellos les doy mi gratitud y admiración por ser los mejores padres del mundo.

A mis hermanos Tony y Carlos por brindarme su amor, apoyo y fuerzas para seguir adelante.

A Dios por darme fuerza y sabiduría, para sobrepasar los obstáculos del vivir del día a día y ser la luz que ilumina mi vida.

A mi Abuelitos que aun estando ellos en el cielo siempre serán ejemplos y guías en el sendero que aun me queda por recorrer.

Yumi

RECONOCIMIENTO

Al Ing. Wilder López Meléndez por su asesoramiento continuo desde la concepción de éste proyecto y por habernos formado tanto académica como personalmente, nuestro mayor reconocimiento y profundo respeto, le estaremos eternamente agradecidos.

AGRADECIMIENTOS

Al Mg. Jorge Luis Hilario Rivas, por su apoyo, dedicación y asesoramiento académico en el desarrollo de la presente tesis.

A la Lic. Liliana Arévalo La Serna, Jefa de la Biblioteca Central de La Universidad Nacional de Ucayali por su generosa contribución en recolección de toda la información necesaria para la elaboración de la tesis.

RESUMEN

Actualmente en la Universidad Nacional de Ucayali, los alumnos y docentes no cuentan con información actualizada sobre los materiales bibliográficos que se encuentran en la Biblioteca, los procesos dentro de ella se realizan manualmente generando pérdida de tiempo y malestar en los usuarios finales. Por ello se propone el Sistema Web con tecnología J2EE para mejorar la administración en la Biblioteca. Para el desarrollo de este proyecto se utilizó la metodología RUP (Rational Unified Process) en todo el proyecto a través de las fases inicial, elaboración, construcción y transición usando la notación UML (Unified Modeling Language) y Rational Rose 2003 como herramienta Case; en la construcción del software se usaron formularios, paginas dinámicas (Java Server Pages), servlets y clases las cuales se conectan a la base de datos realizada en Microsoft SQL Server 2000. Se aplicó también el patrón modelo vista controlador. Finalmente los resultados de la presente investigación demuestran que la implementación de éste sistema Web con tecnología J2EE permitirá mejorar la administración en la Biblioteca de la Universidad Nacional de Ucayali.

Palabras claves: Biblioteca, Tecnología, Administración.

ABSTRACT

At present time In the Ucayali National University, the students and teachers haven't with the update information about the library's bibliographic materials, the processes inside the library are realized manually generating the loss of time and discomfort in the end users. Hence, it is proposed the Sistem Web with technology J2EE for the better the administration in the library. To develop this project was utilized the RUP (Rational Unified Process) methodology in all the project through the initial phase, elaboration, construction and transition using the notation UML (Unified Modeling Language) and Rational Rose 2003 as tool Case; in the construction of the software was utilized formularys, dynamic pages (Java Server Pages), servlets and class which the connect to the data base made in Microsoft SQL Server 2000. It was also applied the model vista controlador standard. Finally the results of the present investigation demonstrate to us than the implementation of this sistem web with technology J2EE will permit to improve the administration in the Ucayali National University's library.

Key words: Library, Technology, Administration.

TABLA DE CONTENIDO

DEDICATORIA	ii
RECONOCIMIENTO	iii
RESUMEN	iv
ABSTRACT	v
INTRODUCCIÓN	1
CAPITULO I	4
PLANTEAMIENTO DEL PROBLEMA	4
1.1. Formulación del Problema	4
1.2. Antecedentes del Problema	4
1.3. Objetivos de la Investigación	5
1.3.1. General	5
1.3.2. Específicos	6
1.4. Justificación e Importancia	7
1.4.1. Técnica	7
1.4.2. Económica	7
1.4.3. Académica	8
1.5. Hipótesis y Variables	9
1.5.1. Hipótesis	9
1.5.2. Variables	9
1.6. Operacionalización de las Variables	9
CAPITULO II	11
MARCO TEORICO	11
2.1. Antecedentes del Estudio	11
2.2. Planteamientos Teóricos	13

2.2.1. Arquitectura Cliente Servidor	13
2.2.2. Tecnología J2EE	22
2.2.3. Programación Orientada a Objetos (14)	39
2.2.4. El Lenguaje de Modelado Unificado (15)	44
2.2.5. El Proceso Unificado Rational (19)	52
2.2.6. Elementos del Proceso Administrativo	58
2.2.7. La Biblioteca como Organización	66
CAPITULO III	71
METODOLOGÍA	71
3.1. Método de Investigación	71
3.1.1. Tipo de Investigación	71
3.1.2. Nivel de Investigación	71
3.1.3. Diseño de Investigación	71
3.2. Población y Muestra	72
3.2.1. Población	72
3.2.2. Muestra	74
3.3. Técnicas e instrumentos para la recolección de datos.	77
3.3.1. Técnicas e instrumentos	77
3.3.2. Validación de las técnicas e instrumentos de recolección.	77
3.4. Procedimiento de recolección de datos	79
3.5. Tratamiento de datos	79
3.5.1. Mecanismos para el procesamiento y presentación de datos	79
3.5.2. Mecanismos para el análisis e interpretación de resultados	80
CAPITULO IV	81
PROCESO DE DESARROLLO DE SOFTWARE	81
4.1. Fase Inicial	81
4.1.1. Modelamiento de visión del negocio	81

A. Documento Visión del Negocio	81
B. Plan de Desarrollo del Software	90
4.2. Fase Elaboración	107
4.2.1. Requerimientos	107
A) Modelo Use Case	107
B) Diagrama de Actividades	109
C) Especificaciones de los Use-Case	111
4.2.2. Análisis y Diseño	116
A) Diagrama de Colaboración	116
B) Diagrama de Paquetes del Análisis	122
C) Diagrama de Clases	123
D) Diseño de la Arquitectura	124
F) Diseño de la Base de Datos	154
G) Diagrama de Estado	156
H) Diagrama de Base de Datos	157
I) Diagrama Vista Web	158
J) Mapa de Navegación	161
4.3. Fase Construcción	162
4.3.1. Diagrama de Componentes	162
4.3.2. Diagrama de Despliegue Lógico	167
4.3.3. Sentencias Lógicas	167
4.4. Fase Transición	255
4.4.1. Flujo de Trabajo de Prueba	255
CAPITULO V	258
RESULTADOS DE LA INVESTIGACIÓN	258
5.1. Resultados	258
5.1.1. A los usuarios	258
5.1.2. A trabajadores	259
5.2. Prueba de hipótesis	261
5.2.1. Formulación de cuadros de valores de indicadores	261

5.2.2. Supuestos de la prueba de hipótesis	265
5.2.3. Cálculo del valor crítico y la función de prueba	265
CAPITULO VI	269
CONCLUSIONES Y RECOMENDACIONES	269
6.1. CONCLUSIONES	269
6.2. RECOMENDACIONES	271
REFERENCIAS BIBLIOGRÁFICAS	272
ANEXOS	276

INDICE DE CUADROS

Nro.	Descripción	Pág.
Cuadro 1	Indicadores Variable Independiente	9
Cuadro 2	Indicadores Variable Dependiente	10
Cuadro 3	Patron Modelo Vista Controlador-Capa de Presentación	37
Cuadro 4	Patron Modelo Vista Controlador-Capa de Negocios	38
Cuadro 5	Patron Modelo Vista Controlador-Capa de Integración	39
Cuadro 6	Distribución Alumnos Matriculados Semestre 2007-II	73
Cuadro 7	Distribución Docentes Nombrados y Contratados	73
Cuadro 8	Distribución Trabajadores Biblioteca Central	74
Cuadro 9	Distribución Alumnos para la Muestra	75
Cuadro 10	Distribución Docentes para la Muestra	76
Cuadro 11	Distribución Trabajadores para la muestra	76
Cuadro 12	Técnicas e Instrumentos	77
Cuadro 13	Sumario de Stakeholders	84
Cuadro 14	Sumario de Usuarios	85
Cuadro 15	Necesidades de Usuarios	85
Cuadro 16	Entregables del proyecto	92
Cuadro 17	Papeles y Responsabilidades	93
Cuadro 18	Plan de proceso de desarrollo de acuerdo a fases	94
Cuadro 19	Fases del proyecto e hitos principales	95
Cuadro 20	Tareas del proyecto	96
Cuadro 21	Presupuesto del proyecto	97
Cuadro 22	Características de Servidor de Base de Datos	98
Cuadro 23	Características de Servidor Web	99
Cuadro 24	Características de las Estaciones de trabajo	99
Cuadro 25	Características de la tarjeta de red	100
Cuadro 26	Características concentradoras	100
Cuadro 27	Procesos del negocio	106

Cuadro 28	Diccionario de datos	154
Cuadro 29	Prueba de Operatividad	256
Cuadro 30	Prueba de Seguridad	256
Cuadro 31	Prueba de Carga Servidor	257
Cuadro 32	Prueba de Carga Cliente	257
Cuadro 33	Valores Pregunta 1 Post Test aplicada a usuarios	258
Cuadro 34	Valores Pregunta 3 Post Test aplicada a usuarios	258
Cuadro 35	Valores Pregunta 1 Post Test aplicada a especialistas	259
Cuadro 36	Valores Pregunta 5 Post Test aplicada a especialistas	259
Cuadro 37	Comparación de valores Pre y Post	260
Cuadro 38	Comparación de indicadores cualitativos	263
Cuadro 39	Comparación de indicadores cuantitativos	264
Cuadro 40	Resumen de Resultados de Valores	265

INDICE DE GRAFICOS

Nro.	Descripción	Pág.
Gráfico 1	Arquitectura cliente servidor 3 capas	17
Gráfico 2	Arquitectura de un sitio Web básica	22
Gráfico 3	Arquitectura de un sitio Web dinámico	22
Gráfico 4	Compilación y ejecución de programas en Java	23
Gráfico 5	Ciclo de Vida de un Servlets	26
Gráfico 6	El modelo tradicional para las aplicaciones Web comparado con el modelo de AJAX	30
Gráfico 7	La interacción sincrónica de una aplicación comparada con el patrón asincrónico de Ajax	31
Gráfico 8	Modelo Vista Controlador	33
Gráfico 9	Origen UML	46
Gráfico 10	Fases RUP	56
Gráfico 11	Use Case del Negocio	102
Gráfico 12	Modelo de Objetos Gestión Material Bibliográfico	103
Gráfico 13	Modelo de Objetos Servicio biblioteca	104
Gráfico 14	Modelo de Objetos Administración de Usuarios	105
Gráfico 15	Modelo Dominio	105
Gráfico 16	Requerimientos Gestión Material	108
Gráfico 17	Requerimientos Servicio Biblioteca	109
Gráfico 18	Requerimientos Administración de Usuarios	109
Gráfico 19	Actividades Gestión Material	110
Gráfico 20	Actividades Servicio Biblioteca	111
Gráfico 21	Actividades Administración de Usuarios	112
Gráfico 22	Colaboración Consultar material	117
Gráfico 23	Colaboración Mantener autor	118
Gráfico 24	Colaboración Mantener editorial	118
Gráfico 25	Colaboración Mantener materia	119
Gráfico 26	Colaboración Registrar material	119
Gráfico 27	Colaboración Registrar préstamo	120

Gráfico 28	Colaboración Registrar reserva	120
Gráfico 29	Colaboración Registrar Asistencia	121
Gráfico 30	Colaboración Registrar Lectura	121
Gráfico 31	Colaboración Validar Préstamo	121
Gráfico 32	Colaboración Registrar Devolución	122
Gráfico 33	Colaboración Registrar Usuario	122
Gráfico 34	Diagrama de paquetes	123
Gráfico 35	Diagrama de clases	124
Gráfico 36	Diseño Arquitectura	125
Gráfico 37	Interfaz Registrar Libro	126
Gráfico 38	Diagrama De Secuencia Registrar Libro	127
Gráfico 39	Interfaz Registrar Publicación Seriada	128
Gráfico 40	Diagrama de Secuencia Registrar Publicación Seriada	129
Gráfico 41	Interfaz Registrar Tesis	130
Gráfico 42	Diagrama de Secuencia Registrar Tesis	131
Gráfico 43	Interfaz Mantener Autor	132
Gráfico 44	Diagrama de Secuencia Registrar Autor	133
Gráfico 45	Interfaz Mantener Editorial	134
Gráfico 46	Diagrama de Secuencia Mantener Editorial	135
Gráfico 47	Interfaz Mantener Materia	136
Gráfico 48	Diagrama de Secuencia Mantener Materia	137
Gráfico 49	Interfaz Registrar Préstamo	138
Gráfico 50	Diagrama de Secuencia Registrar Préstamo	139
Gráfico 51	Interfaz Registrar Devolución	140
Gráfico 52	Diagrama de Secuencia Registrar Devolución	141
Gráfico 53	Interfaz Consulta-Reserva	142
Gráfico 54	Diagrama de Secuencia Consulta-Reserva	143
Gráfico 55	Interfaz Validar Reserva	144
Gráfico 56	Diagrama de Secuencia Validar Reserva	144
Gráfico 57	Interfaz Registrar Lectura	145
Gráfico 58	Diagrama de Secuencia Registrar Lectura	146
Gráfico 59	Interfaz Registrar Asistencia	147
Gráfico 60	Diagrama de Secuencia Registrar Asistencia	147

Gráfico 61	Interfaz Mantener Usuarios	148
Gráfico 62	Diagrama de Secuencia Mantener Usuario	148
Gráfico 63	Interfaz Reporte de Materiales Según Clasificación	149
Gráfico 64	Diagrama de Secuencia Reporte de Materiales Según Clasificación	149
Gráfico 65	Interfaz Reporte de Prestamos Realizados a Usuarios	150
Gráfico 66	Diagrama de Secuencia Reporte de Préstamos Realizados A Usuarios	150
Gráfico 67	Interfaz Reporte de Prestamos por Fecha	151
Gráfico 68	Diagrama de Secuencia Reporte de Prestamos por Fecha	151
Gráfico 69	Interfaz Reporte Estadístico de Visitas Mensuales por Escuela	152
Gráfico 70	Diagrama de Secuencia Reporte Estadístico de Visitas Mensuales por Escuela	152
Gráfico 71	Interfaz Reporte Estadístico de Variación de Prestamos	153
Gráfico 72	Diagrama de Secuencia Reporte Estadístico de Variación de Prestamos	153
Gráfico 73	Interfaz Reporte Estadístico de Prestamos por Escuelas	154
Gráfico 74	Diagrama de Secuencia Reporte Estadístico de Prestamos por Escuelas	154
Gráfico 75	Diagrama de Estados Material	157
Gráfico 76	Diagrama de Estados Préstamo	158
Gráfico 77	Diagrama de Estados Reserva	158
Gráfico 78	Diagrama de Estados Usuario	158
Gráfico 79	Diagrama de Base de Datos	159
Gráfico 80	Diagrama Vista Web Gestión de Material Bibliográfico	160
Gráfico 81	Diagrama Vista Web Servicio de Biblioteca	161
Gráfico 82	Diagrama Vista Web Administración Usuario	162
Gráfico 83	Diagrama Mapa de Navegación del Sistema	163

Gráfico 84	Diagrama de Componentes	164
Gráfico 85	Componente GesAca.jar	165
Gráfico 86	Componente Escalafon.jar	165
Gráfico 87	Componente GestionMaterial.jar	166
Gráfico 88	Componente ServicioBiblioteca.jar	166
Gráfico 89	Componente Administración-Usuarios.jar	167
Gráfico 90	Componente dBarcode128.jar	167
Gráfico 91	Componente jfreechart-1.0.3.jar	168
Gráfico 92	Componente ajaxtags1.2.jar	168
Gráfico 93	Diagrama de Despliegue Lógico	169
Gráfico 94	Comparación de valores Pre y Post.	244

INTRODUCCIÓN

Es el conocimiento el elemento más importante de una organización, su gestión constituye el proceso que permite que este conocimiento no se pierda y sea conservado en el interior de la organización y a la vez pueda ser compartido, organizado, transformado, utilizado y distribuido entre los implicados de la misma, con el fin de lograr los objetivos previstos y elevar la calidad de los servicios, a través de la Gestión del conocimiento se permite diligenciar los flujos de información y elevar la información concreta a las personas que la necesitan. En la actualidad las nuevas tecnologías de la información y las comunicaciones han contribuido a una mayor facilidad en el uso y la creación del conocimiento.

La Biblioteca de la Universidad Nacional De Ucayali, es un centro de información que facilita a sus miembros toda clase de conocimientos e información. Para ello debe estar administrado de manera eficiente y así poder brindar a sus usuarios los elementos necesarios para su desarrollo académico. Teniendo en cuenta lo anterior, la Biblioteca debe permitir a sus usuarios consultar la información de los materiales bibliográficos en cualquier momento y desde cualquier lugar, hacer reservas en línea y realizar préstamos y devoluciones minimizando la pérdida de tiempo.

Esto nos lleva a pensar que la implementación de un sistema Web con tecnología J2EE, puede ser la base para una mejor administración de la Biblioteca Central de la Universidad Nacional de Ucayali. Actualmente no se cuenta con un sistema de éstas características dentro de la biblioteca

mencionada anteriormente.

La investigación viene estructurada de la siguiente manera:

En el Capítulo I : Planteamiento del problema; en esta parte se describe completamente el plan de investigación realizado en el proyecto, tales como formulación del problema, la hipótesis, variables de la hipótesis, objetivos generales y específicos, justificación técnica, económica, académica.

En el Capítulo II: Marco Teórico; En esta parte se describe el fundamento teórico, donde se hace hincapié en exponer el conocimiento teórico sobre la metodología "Rational Unified Process" y UML, además definir los conceptos de Sistemas Web, tecnología J2EE, además de definir y dar características para su comparación de las herramientas y patrones de diseño, también mostrar la arquitectura de implementación con es Cliente/Servidor en sus diferentes modelos, finalmente nos referimos a los Administración de Procesos.

En el Capítulo III: Metodología de Investigación; se detalla el método de la investigación, la población y la muestra obtenida, los procedimientos, técnicas e instrumentos usados para la recolección y procesamiento de datos.

En el Capítulo IV: Desarrollo de la metodología RUP; se detalla la aplicación del "Rational Unified Process" (RUP), mostrando los resultados del Análisis y Diseño a través de documentos, diagramas con estereotipos UML, además de las características para la construcción del Software; abarcando las fases Inicial, de Elaboración, Construcción y Transición.

En el Capítulo V: Resultados y Discusión; Se describe los resultados de la comprobación de la Hipótesis obtenida a través de las encuestas realizadas en la Biblioteca.

En el Capítulo VI: Conclusiones y Recomendaciones; obtenidas en el desarrollo de la presente tesis.

Finalmente se presenta; las referencias Bibliográficas, y los Anexos necesarios como encuestas y cuadros del Plan de desarrollo de software.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. Formulación del Problema

¿En qué medida el Sistema Web con tecnología J2EE mejora la Administración en la Biblioteca de la Universidad Nacional de Ucayali?

1.2. Antecedentes del Problema

La Universidad Nacional de Ucayali es una institución educativa dedicada a la formación profesional de la población estudiantil de la Región Ucayali, por tanto y con el fin de cumplir éste vital objetivo debe ofrecer al alumnado, docentes y personal administrativo, las herramientas necesarias para un óptimo desarrollo profesional, personal e intelectual. En éste sentido la Universidad Nacional de Ucayali brinda la Biblioteca Central como fuente de consulta bibliográfica a su comunidad universitaria.

Tener una óptima gestión de los procesos es esencial dentro de cualquier organización, ello permitirá un mejor manejo administrativo y una atención eficiente a los usuarios finales. El uso de las computadoras para procesar y estar actualizados con la información que nos compete es hoy algo imprescindible para la mayoría de las personas y organizaciones inmersas en ello.

La Biblioteca Central de la Universidad Nacional de Ucayali no escapa a ésta realidad y necesita mantener una eficiente gestión de sus procesos:

En una biblioteca, la función administrativa descansa primordialmente en el director de la misma; pero deben compartirla, proporcionalmente todos aquellos miembros del personal que a su vez supervisan y guían el trabajo de otros. En una biblioteca cada nuevo asistente debe aprender, tan pronto como sea posible, cuáles son sus y la parte que su departamento y sus tareas particulares representan en la obtención de los objetivos de la biblioteca.

Las tareas que se realizan dentro de la Biblioteca deben ser llevadas a cabo por personal especializado en la materia:

Proceso Técnico: Se encarga de clasificar los materiales bibliográficos para su posterior posicionamiento dentro de la biblioteca según código Dewey y Málaga.

Préstamo: Encargado de Realizar los prestamos y la guía para la obtención de Materiales Bibliográficos según la necesidad del usuario

Actualmente los mencionados procesos se realizan de forma manual lo que genera retraso y pérdida de tiempo a los usuarios.

1.3. Objetivos de la Investigación

1.3.1. General

Conocer como el Sistema Web con tecnología J2EE mejora la administración en la biblioteca de la Universidad Nacional de Ucayali.

1.3.2. Específicos

1. Identificar los procesos de la Biblioteca Central de la Universidad Nacional de Ucayali.
2. Identificar las necesidades de información para la toma de decisiones de la Biblioteca Central de la Universidad Nacional de Ucayali.
3. Modelar los datos que permitan obtener información consolidada y agrupada técnicamente de la Biblioteca Central de la Universidad Nacional de Ucayali.
4. Aplicar la Metodología RUP y el Lenguaje Unificado de Modelado UML para el análisis y diseño del Sistema Web con tecnología J2EE para la administración en la biblioteca de la Universidad Nacional de Ucayali.
5. Aplicar la Tecnología J2EE, en el Sistema Web para la administración de la Biblioteca Central de la Universidad Nacional de Ucayali.
6. Migrar a plataforma con interfaces Web, con el fin de de brindar opciones más flexibles de accesibilidad a la información que se maneja en la Biblioteca de la Universidad Nacional de Ucayali.
7. Determinar el grado de asociación, correlación o causalidad entre el Sistema Web con tecnología J2EE y la administración en la biblioteca de la Universidad Nacional de Ucayali.

8. Proponer el Sistema Web con Tecnología J2EE que mejora la administración en la Biblioteca de la Universidad Nacional de Ucayali.

1.4. Justificación e Importancia

1.4.1. Técnica

Este estudio es un proyecto base abierto a la integración de otros módulos necesarios para automatizar otros procesos, navegabilidad, usabilidad y presentación de datos.

Desde el punto de vista de su construcción, la Base de Datos del sistema propuesto organizará y usará de manera apropiada los datos contenidos en la misma, permitiendo brindar información relevante a los usuarios finales.

1.4.2. Económica

El desarrollo del sistema propuesto no requerirá de equipos de cómputo muy costosos, la Universidad Nacional de Ucayali cuenta con equipos necesarios para el desarrollo e implementación del proyecto que se pretende realizar.

El sistema propuesto podrá manejar con mayor rapidez la información existente en la Biblioteca Central de la Universidad Nacional de Ucayali y mediante una interfaz adecuada mostrará reportes precisos a los requerimientos del usuario, permitiendo tomar decisiones adecuadas y oportunas.

Otros beneficios de naturaleza tangible e intangible como resultado de este proyecto se resumen a continuación:

Beneficios Tangibles:

- Resultados en menor tiempo, operación que no se podía obtener anteriormente; ahora la obtención de consultas y reportes se darán en contados segundos.
- El proceso de interpretación se verá agilizado y apoyado por la herramienta que se pretende construir.

Beneficios Intangibles:

- Facilita la toma de decisiones.
- Facilita el planeamiento y ejecución de actividades dentro de la Biblioteca Central.

1.4.3. Académica

Este Proyecto servirá de referencia para futuros estudios relacionados al mejoramiento de la administración de la Biblioteca de la Universidad Nacional de Ucayali.

El proyecto nos permite cumplir con las normas y directivas de nuestra Universidad Nacional de Ucayali para obtener el Título Profesional de Ingeniero de Sistemas.

1.5. Hipótesis y Variables

1.5.1. Hipótesis

"El Sistema Web con tecnología J2EE mejora la administración en la Biblioteca de la Universidad Nacional de Ucayali."

1.5.2. Variables

a. Variable Independiente

Sistema Web con Tecnología J2EE.

b. Variable Dependiente

Administración de Biblioteca Central.

c. Variable Interviniente

Universidad Nacional de Ucayali

1.6. Operacionalización de las Variables

Variable Independiente: Sistema Web con Tecnología J2EE

Cuadro 1: Indicadores Variable Independiente

Dimensiones	Indicadores
Construcción del sistema	Grado de utilidad manifestado por el usuario (%)
Implementar la base de datos	Tiempo de respuesta de las consultas (segundos)
Análisis y Diseño del sistema	Nivel de Satisfacción del usuario al usar el sistema (cantidad)
Procesamiento de datos	Tiempo de respuesta de ejecución de procesos (segundos)

Variable Dependiente: Administración de Biblioteca Central

Cuadro 2: Indicadores Variable Dependiente

Dimensiones	Indicadores
Planificación	Nº de materiales bibliográficos asignados según especialidad Nº de ingresos de material bibliográfico Tiempo de registro de materiales bibliográficos
Organización	Nº trabajadores asignados según función Nº de tipos de niveles de acceso
Dirección	Nivel de atención a los usuarios para realizar un préstamo Tiempo de consulta de un material bibliográfico Nº de asistencias a la biblioteca Nº de prestamos realizados Nº de usuarios multados Nº de libros leídos Nº de reservas realizadas Nº total de materiales bibliográficos según tipo
Coordinación	Nº de niveles de acceso Nº de usuarios atendidos
Control	Nº de prestamos realizados por cada trabajador Nº de personas que asistieron a la biblioteca en un mes. Nivel de Control de usuarios.

CAPITULO II

MARCO TEORICO

2.1. Antecedentes del Estudio

Dentro de los sistemas de información universitaria el acceso a la información científico, técnico y cultural es esencial para el desarrollo de la formación profesional del estudiante universitario. Estos sistemas deben promover el escenario en el cual las actividades de planificación, gestión, promoción, investigación, educación, atención, etc., sean progresivamente fundadas en el conocimiento científico actualizado.

El acceso ilimitado a la información científico técnica perfecciona los procesos de toma de decisiones en las diversas áreas del conocimiento humano, en diferentes situaciones y niveles sociales, es ahí donde aparece el cambio del modelo de flujo de información, que acontece con la progresiva confluencia de las actividades de los diferentes actores, instancias y eventos relacionados con el flujo de información científico técnica en torno a espacios comunes en Internet. (1)

En los últimos años el uso de los Sistemas Web en las organizaciones dieron muchos resultados favorables para la mejor optimización de los procesos y políticas de negocio dentro de una organización; existen Sistemas que brindan reportes o informes que permiten plantearse diferentes alternativas para toma de decisiones, así también los que aceleran procesos tediosos. Es por ello que ahora las organizaciones como las Bibliotecas que son organizaciones que en los últimos 25 años

han sufrido una serie de cambios con los avances tecnológicos (2); sin ir muy lejos en el Perú las Universidades tantas públicas como privadas vienen haciendo uso de los sistemas Web con el fin de organizar, automatizar y ejecutar los procesos de negocios, logrando de esta manera monitorear los procesos y a la vez brindar una calidad de servicio eficiente a sus usuarios.

Estos sistemas Web pueden ser desarrollados sobre distintas plataformas como J2EE, .NET, PHP, etc. Pero a nivel nacional y regional no existen antecedentes de Sistemas Web desarrollados bajo la plataforma J2EE, por lo que esta investigación se presenta como un punto de inicio para futuras investigaciones.

A nivel mundial podemos mencionar algunos sistemas de biblioteca desarrollados en el lenguaje de programación Java, entre ellos tenemos:

- **Avanti (3)** fue desarrollado en 1998 como un sistema integrado de la biblioteca para aquellas de pequeño y mediano tamaño, pero hubo poca actividad en su desarrollo hasta el 2004. Esto debido a que Peter Schlumpf, su desarrollador, trabajaba solo en el proyecto. El software fue escrito en Java y funcionará sobre cualquier sistema operativo. El código de la fuente y documentación no están disponibles en línea porque el desarrollador no busca la participación de otros. No se precisa la aplicación en alguna organización específica.
- **Pytheas (4)** fue originalmente desarrollado en 1999 por un bibliotecario de la Universidad de Arizona. Después de que él abandonó el proyecto, otro bibliotecario de la Universidad de Windsor mantuvo la programación. Esta desarrollado en el lenguaje de programación Java.

El código fuente está disponible para su descarga y también existe una demo. El producto es del dominio público, la documentación está limitada y el producto es altamente dimensionable. No se precisa la aplicación en alguna organización específica.

2.2. Planteamientos Teóricos

2.2.1. Arquitectura Cliente Servidor

Es una arquitectura descentralizada que permite a los usuarios finales obtener acceso a la información de forma transparente; es decir que al usuario le es indiferente de donde viene la información. Clientes y servidores son entidades lógicas independientes que operan en conjunto a través de una red para realizar una tarea. (5)

Con la proliferación de ordenadores personales de bajo coste en el mercado, los recursos de sistemas de información existentes en cualquier organización se pueden distribuir entre ordenadores de diferentes tipos: ordenadores personales de gama baja, media y alta, estaciones de trabajo, mini ordenadores o incluso grandes ordenadores.

El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos estos recursos de máquina de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe se traspasa a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario. (6)

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de Periféricos compartidos
- Control de accesos concurrentes a la Base de Datos
- Enlaces de comunicaciones con otras redes de área local o extensa

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en

distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

a) Arquitectura Cliente Servidor 2 capas

La arquitectura Cliente Servidor de dos capas ha sido la más utilizada con los lenguajes de cuarta generación. En este tipo de aplicaciones se desarrollan dos capas:

Una capa llamada front-end (la interfaz de usuario), que normalmente está constituida de aplicaciones de escritorio que hacen llamadas a un servidor SQL. La otra capa se conoce como

back-end, normalmente constituida por un servidor de base de datos con interfaz SQL y un sistema operativo multitarea.

Aunque este tipo de arquitectura es fácil de implementar, el hecho de que la lógica del negocio y la presentación se encuentren almacenados en el cliente (front-end), lleva a que cambios en la base de datos o en la lógica de negocios implique hacer actualizaciones en todos los clientes. Otra desventaja es la dificultad de compartir procesos comunes.

b) Arquitectura Cliente Servidor 3 capas

Una arquitectura de tres capas provee adicionalmente una capa explícita para las reglas del negocio, que se sitúa entre el front-end y el back-end. Esta capa intermedia encapsula el modelo de negocio asociado con el sistema y lo separa tanto de la presentación como del manejo de la base de datos. Normalmente las reglas del negocio se encapsulan en componentes, en cada uno de los cuales radican servicios al usuario.

Por regla general, La capa de la presentación es una interfaz gráfica que muestra los datos a los usuarios.

La capa de la lógica de negocios es responsable de procesar los datos recuperados y enviarlos a la capa de presentación.

La capa de datos almacena los datos de la aplicación en un almacén persistente, tal como una base de datos relacional o archivos XML.

- Se puede modificar la lógica del negocio sin hacer cambios a la interfaz del usuario o la base de datos.
- Provee escalabilidad horizontal y vertical. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

c) Arquitectura Cliente Servidor n capas

Los sistemas de n capas están formados por múltiples capas totalmente independientes y que por lo tanto presentan muy poco acoplamiento. Una división típica de un sistema de n-capas será la siguiente:

- **Capa de cliente:** Está formada por los componentes que se ejecutan en el cliente. Los tipos de clientes pueden ser muy variados (aplicaciones de escritorio, navegadores Web, dispositivos portátiles, etc). Según el cliente se distingue entre dos tipos de clientes: "gruesos" y "finos". Los clientes gruesos representan aplicaciones que acceden directamente a la capa de lógica de negocio. Los clientes finos, suelen ser navegadores Web que acceden a la capa de presentación.
- **Capa de presentación:** Se encarga de crear la presentación que renderizará la capa de cliente. Actúa como un puente entre la capa de cliente y la capa de lógica de negocio. En Java, esta

capa se implementa típicamente utilizando componentes Servlets/JSP.

- **Capa de lógica de negocio:** Esta capa contiene a todos nuestros objetos de negocio y servicios de procesamiento de la lógica de negocio. Suelen representar una vista objetual de la información presente en la base de datos. En Java, esta capa se suele implementar con EJBs (Enterprise JavaBeans) aunque pueden utilizarse también otras tecnologías más ligeras como el uso directo de objetos planos. El servidor de aplicaciones se encarga de automatizar gran cantidad de tareas como la gestión de transacciones, seguridad, redundancia, balanceo de carga, cachés, pools, etc.
- **Capa de integración:** En esta capa suelen colocarse clases u objetos que automaticen el acceso a base de datos. No es una capa básica pero si que suele utilizarse mucho. Tradicionalmente está compuesta por alguna implementación del patrón DAO (Data Access Objects), aunque depende del tipo de lógica de negocio que estemos utilizando por ejemplo con CMP no tendría sentido tener esta capa mientras que con objetos planos podríamos utilizar JDO).
- **Capa de datos:** Representa los sistemas de información empresarial de nuestra organización. Suele estar formada por bases de datos, servidores de sockets, sistemas legacy, etc. La capa de integración nos ayuda a acceder a estos sistemas heterogéneos de la forma más transparente posible.

Algunas características generales de los sistemas de n-capas son:

- Tienen una buena fiabilidad y disponibilidad ya que el servidor de aplicaciones ofrece sistemas de clustering que aseguran que el sistema está disponible. No existe un único punto de fallo.
- La lógica de negocio está separada del resto de capas. Un cambio a la lógica de negocio no implica la modificación de todos los clientes. Asimismo la capa de presentación nos permite modificar el interfaz de la aplicación sin tener la necesidad de realizar cambios en los clientes.
- El rendimiento suele ser muy bueno. El servidor de aplicaciones ofrece automáticamente soporte de pool de objetos y conexiones, cachés, balanceo de carga, gestión de múltiples instancias, etc.
- Es un modelo con muy buena escalabilidad, tanto horizontal como vertical.
- La extensibilidad es excelente. Existe poco acoplamiento entre las diferentes capas.
- La seguridad se encuentra centralizada. Por contra la seguridad suele implicar gran cantidad de sistemas y tiene una arquitectura compleja.

d) Arquitectura Web

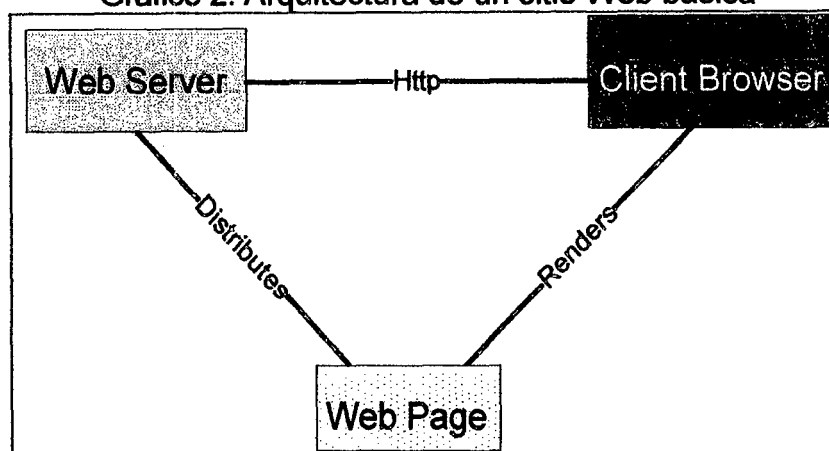
La arquitectura de un sitio Web tiene tres componentes: un servidor Web, una conexión de red y uno o más clientes (Browsers).

El servidor Web distribuye páginas de información formateada a los clientes que la solicitan. Los requerimientos son hechos a través de una conexión de red y para ello se usa el protocolo HTTP.

La arquitectura de las aplicaciones Web está evolucionando hacia modelos más avanzados que permitan cumplir los requisitos de las empresas que invierten en Internet. La aparición de clientes dinámicos y nuevos dispositivos, y el compromiso de nuevos servicios Web, están orientando las tecnologías de desarrollo de aplicaciones para Internet en una nueva dirección.

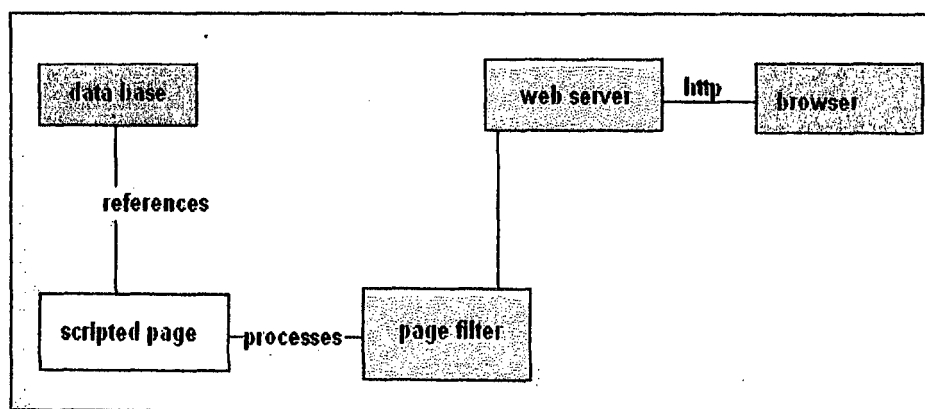
La información mostrada en las páginas esta típicamente almacenada en un archivo. Sin embargo muchas veces esta información esta almacenada en una base de datos y las páginas son creadas dinámicamente. Los sitios Web que usan este esquema son llamados sitios Web dinámicos. (7)

Gráfico 2: Arquitectura de un sitio Web básica



Fuente: Tecnología Cliente Servidor (8)

Gráfico 3: Arquitectura de un sitio Web dinámico



Fuente: Tecnología Cliente Servidor (8)

2.2.2. Tecnología J2EE

a) Lenguaje de Programación Java (9)

La tecnología Java consta de un lenguaje de programación y una plataforma. Java es un lenguaje de programación de alto nivel que tiene las siguientes características:

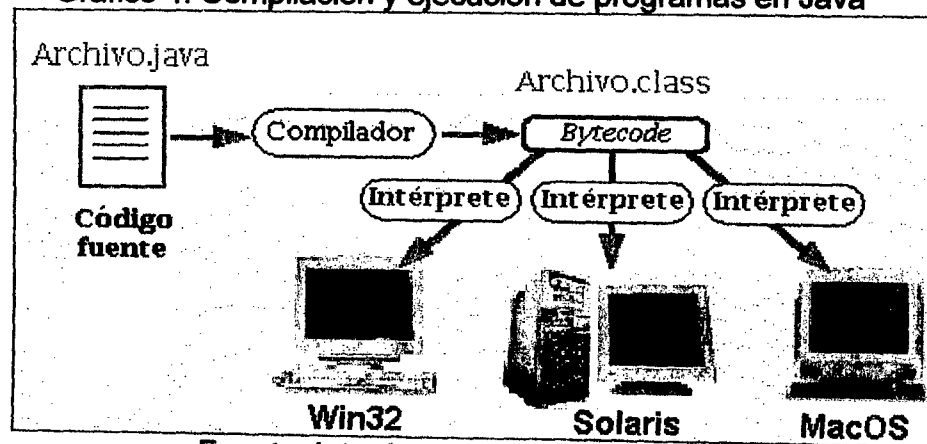
- Orientado a objetos
- Distribuido y dinámico

- Robusto
- Seguro
- Multitarea
- Portable

La mayoría de los lenguajes de programación se caracterizan por ser interpretados o compilados, lo que determina la manera en como serán ejecutados en una computadora.

Java tiene la característica de ser al mismo tiempo compilado e interpretado. El compilador es el encargado de convertir el código fuente de un programa en un código intermedio llamado bytecode que es independiente de la plataforma en que se trabaje y que es ejecutado por el intérprete de Java que forma parte de la Máquina Virtual de Java.

Gráfico 4: Compilación y ejecución de programas en Java



Fuente: Introducción al Lenguaje Java (10)

Entre las características más importantes del lenguaje se encuentra la robustez. Justamente por la forma en que está diseñado, Java

no permite el manejo directo del hardware ni de la memoria (inclusive no permite modificar valores de punteros, por ejemplo). El intérprete siempre tiene el control. El compilador es suficientemente inteligente como para no permitir cosas que podrían traer problemas, como usar variables sin inicializarlas, modificar valores de punteros directamente, acceder a métodos o variables en forma incorrecta, etc. Además, Java implementa mecanismos de seguridad que limitan el acceso a recursos de las máquinas donde se ejecuta, especialmente en el caso de los Applets (que son aplicaciones que se cargan desde un servidor y se ejecutan en el cliente). También está diseñado específicamente para trabajar sobre una red, de modo que incorpora objetos que permiten acceder a archivos en forma remota (vía URL por ejemplo). Además, con el JDK (Java Development Kit) vienen incorporadas muchas herramientas, entre ellas un generador automático de documentación.

b) Java Servlets

Los servlets son aplicaciones Java que se ejecutan en el servidor bajo una arquitectura cliente-servidor-Web, extendiendo las capacidades del servidor Web. El servidor donde se ejecutan los servlets debe contar con una máquina virtual de Java (JVM). Los servlets responden a eventos generados en las estaciones clientes desde requerimientos HTML, y permiten construir una respuesta dinámica a dichos requerimientos. Permiten además retornar una respuesta dinámica a los requerimientos, estos constituyen una

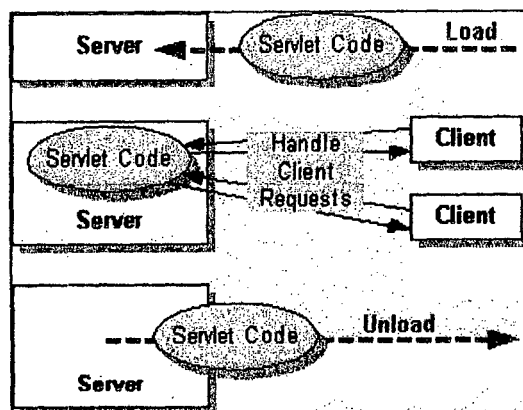
mejor alternativa frente a otras tecnologías como los CGI's, aunque ambos permiten generar respuestas dinámicas para los requerimientos de los clientes, los servlets presentan ventajas sobre los CGI's, como por ejemplo la portabilidad e independencia de la plataforma.

Los servlets al ser componentes escritos en Java, tienen acceso al conjunto completo de API's Java que les permiten interactuar con diferentes tipos de interfaces, como por ejemplo aquellas que acceden a Bases de Datos. Además pueden encontrarse en muchos tipos de servidores diferentes pues el API definido para los servlets, al igual que las demás API's Java, no dependen del ambiente del servidor o de los protocolos utilizados.

Un servlet (11) es cargado una sola vez en el servidor Web e invocado en cada requerimiento. Esto se debe a que los servlets son programas multihilos (multithread). Los servlets pueden entonces mantener recursos del sistema como conexiones a Bases de Datos.

Como los servlets son cargados y ejecutados completamente en el servidor, sin comunicarse directamente con los usuarios Web, no requieren la funcionalidad de una interfaz gráfica, estos implementan una forma simple de interacción entre el cliente y el servidor a través del intercambio de mensajes del tipo request/response (requerimiento/respuesta).

Gráfico 5: Ciclo de Vida de un Servlets



Fuente: Introducción al Lenguaje Java (10)

c) Java Server Pages

JavaServer Pages (JSP), en el campo de la Informática, es una tecnología para crear aplicaciones Web. Es un desarrollo de la compañía Sun Microsystems, y su funcionamiento se basa en scripts, que utilizan una variante del lenguaje java.

JSP, es una tecnología Java que permite a los programadores generar contenido dinámico para Web, en forma de documentos HTML, XML, o de otro tipo. Los JSP's permite al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento Web.

En los JSP, se escribe el texto que va a ser devuelto en la salida (normalmente código HTML) incluyendo código java dentro de él para poder modificar o generar contenido dinámicamente. El código java se incluye dentro de las marcas de etiqueta `<% y %>`, a esto se le denomina scriptlet.

En una posterior especificación, se incluyeron taglib; esto es, la posibilidad de definir etiquetas nuevas que ejecuten código de clases java. La asociación de las etiquetas con las clases java se declara en archivos de configuración en XML.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones Web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento html en el archivo jsp.

Independientemente de la certeza de la aseveración, Java es conocido por ser un lenguaje muy portable (su lema publicitario reza: escríbelo una vez, córralo donde sea), y sumado a las capacidades de JSP se hace una combinación muy atractiva.

Sin embargo JSP no se puede considerar un script al 100% ya que antes de ejecutarse el servidor Web compila el script y genera un servlet, por lo tanto, se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API de Java en su totalidad.

Debido a esto la tecnología JSP, así como Java está teniendo mucho peso en el desarrollo Web profesional (sobre todo en intranets).

d) Java Beans

Un JavaBean o Bean es un componente hecho en software que se puede reutilizar y que puede ser manipulado visualmente por una herramienta de programación en lenguaje Java.

Para ello, se define un interfaz para el momento del diseño (design time) que permite a la herramienta de programación o IDE, interrogar (query) al componente y conocer las propiedades (properties) que define y los tipos de sucesos (events) que puede generar en respuesta a diversas acciones.

Aunque los Beans individuales pueden variar ampliamente en funcionalidad desde los más simples a los más complejos, todos ellos comparten las siguientes características:

- **Introspection:** Permite analizar a la herramienta de programación o IDE como trabaja el bean
- **Customization:** El programador puede alterar la apariencia y la conducta del bean.
- **Events:** Informa al IDE de los sucesos que puede generar en respuesta a las acciones del usuario o del sistema, y también los sucesos que puede manejar.
- **Properties:** Permite cambiar los valores de las propiedades del bean para personalizarlo (customization).

- **Persistence:** Se puede guardar el estado de los beans que han sido personalizados por el programador, cambiando los valores de sus propiedades.

En general, un bean es una clase que obedece ciertas reglas:

- Un bean tiene que tener un constructor por defecto (sin argumentos)
- Un bean tiene que tener persistencia, es decir, implementar el interface *Serializable*.
- Un bean tiene que tener introspección (introspection). Los IDE reconocen ciertas pautas de diseño, nombres de las funciones miembros o métodos y definiciones de las clases, que permiten a la herramienta de programación mirar dentro del bean y conocer sus propiedades y su conducta.

e) Técnica Ajax

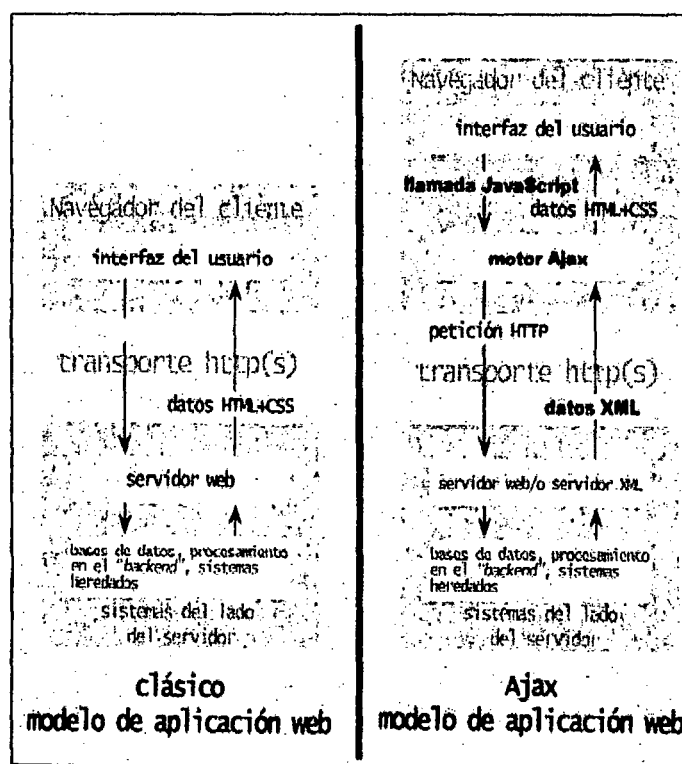
Ajax (Asynchronous JavaScript And XML) no es una tecnología. Es realmente muchas tecnologías, cada una floreciendo por su propio mérito, uniéndose en poderosas nuevas formas. AJAX incorpora:

- presentación basada en estándares usando XHTML y CSS;
- exhibición e interacción dinámicas usando el Document Object Model ;
- Intercambio y manipulación de datos usando XML and XSLT ;
- Recuperación de datos asincrónica usando XMLHttpRequest ;

- y JavaScript poniendo todo junto.

El modelo clásico de aplicaciones Web funciona de esta forma: La mayoría de las acciones del usuario en la interfaz disparan un requerimiento HTTP al servidor Web. El servidor efectúa un proceso (recopila información, procesa números, hablando con varios sistemas propietarios), y le devuelve una página HTML al cliente.

Gráfico 6: El modelo tradicional para las aplicaciones Web comparado con el modelo de AJAX



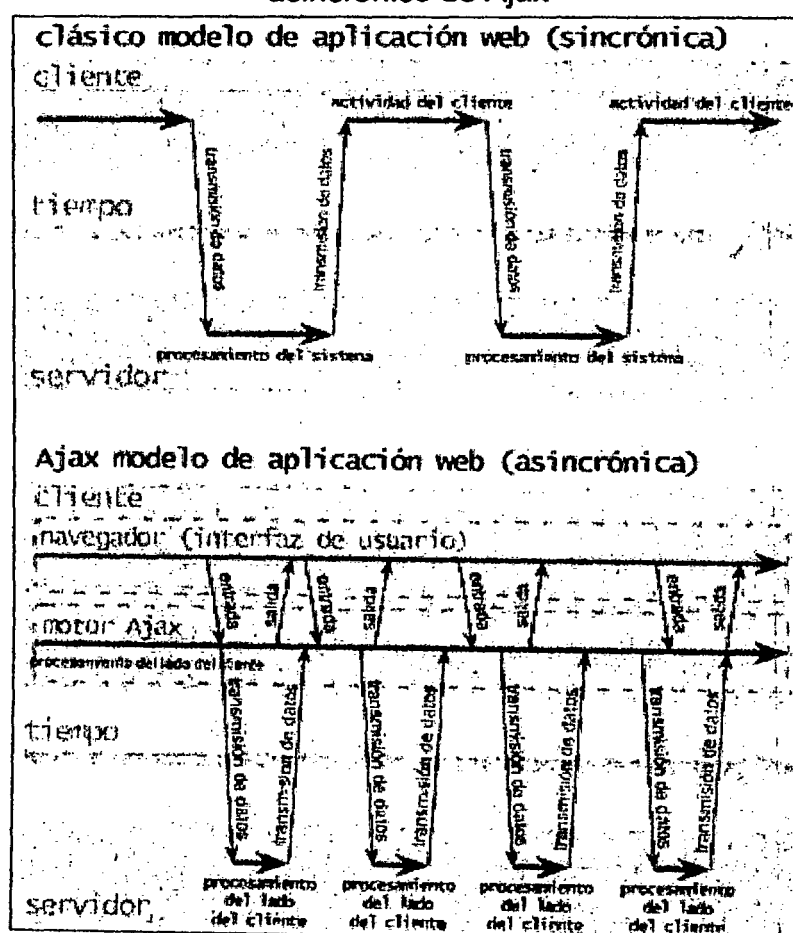
Fuente: Ajax: Una nueva aproximación a las aplicaciones Web (12)

Una aplicación AJAX elimina la naturaleza "arrancar-frenar-arrancar-frenar" de la interacción en la Web introduciendo un intermediario -un motor AJAX- entre el usuario y el servidor. Parecería que sumar una capa a la aplicación la haría menos reactiva, pero la verdad es lo contrario.

En vez de cargar una página Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript y usualmente "sacado" en un frame oculto). Este motor es el responsable por renderizar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario.

El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo.

Gráfico 7: La interacción sincrónica de una aplicación comparada con el patrón asincrónico de Ajax



Fuente: Ajax: Una nueva aproximación a las aplicaciones Web (12)

Cada acción de un usuario que normalmente generaría un requerimiento HTTP toma la forma de un llamado JavaScript al motor AJAX en vez de ese requerimiento. Cualquier respuesta a una acción del usuario que no requiera un viaje de vuelta al servidor (como una simple validación de datos, edición de datos en memoria, incluso algo de navegación) es manejado por su cuenta.

Si el motor necesita algo del servidor para responder (sea enviando datos para procesar, cargar código adicional, o recuperando nuevos datos) hace esos pedidos asincrónicamente, usualmente usando XML, sin frenar la interacción del usuario con la aplicación.

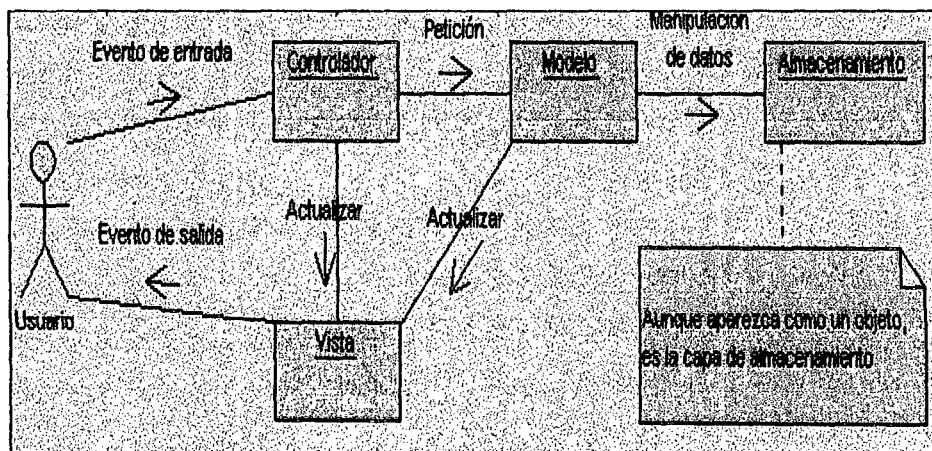
f) Modelo Vista Controlador

La arquitectura Modelo-Vista-Controlador es un patrón de diseño clásico que es muy usado en aplicaciones que requieren la habilidad de mantener múltiples vistas del mismo dato; es decir, se usa para administrar la información y notificar algún cambio en la misma, por lo que su funcionalidad está dirigida a un propósito común. Esta arquitectura es muy similar a la de Cliente/Servidor, la diferencia es que en la primera todos los componentes se encuentran ligados en una sola aplicación; mientras que en ésta el desempeño depende de la completa separación de los objetos en una de las tres categorías que lo componen: el modelo, la vista y el controlador.

La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, un pastiche que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Gráfico 8: Modelo Vista Controlador



Fuente: RedIris (13)

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en

el almacén, consultar el tiempo de entrega estándar del proveedor".

- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción, etc).

El controlador es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas.

Las vistas son responsables de:

- Recibir datos del modelo y la muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

g) Patrones de Diseño J2EE

Un patrón de diseño es una abstracción de una solución en un nivel alto. Los patrones solucionan problemas que existen en muchos niveles de abstracción. Hay patrones que abarcan las distintas etapas del desarrollo; desde el análisis hasta el diseño y desde la arquitectura hasta la implementación.

Muchos diseñadores y arquitectos de software han definido el término de patrón de diseño de varias formas que corresponden al ámbito a la cual se aplican los patrones. Luego, se dividió los patrones en diferentes categorías de acuerdo a su uso.

Los diseñadores de software extendieron la idea de patrones de diseño al proceso de desarrollo de software. Debido a las características que proporcionaron los lenguajes orientados a objetos (como herencia, abstracción y encapsulamiento) les permitieron relacionar entidades de los lenguajes de programación a entidades del mundo real fácilmente, los diseñadores empezaron a aplicar esas características para crear soluciones comunes y reutilizables para problemas frecuentes que exhibían patrones similares.

Fue por los años 1994, que apareció el libro "Design Patterns: Elements of Reusable Object Oriented Software" escrito por los ahora famosos Gang of Four (GoF, que en español es la pandilla de los cuatro) formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Ellos recopilaron y documentaron 23

patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos. Desde luego que ellos no son los inventores ni los únicos involucrados, pero ese fue luego de la publicación de ese libro que empezó a difundirse con más fuerza la idea de patrones de diseño.

El grupo de GoF clasificaron los patrones en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

- **Creacionales:** Patrones creacionales tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.
- **Estructurales:** Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionados pueden ser incluso objetos simples u objetos compuestos.
- **Comportamiento:** Los patrones de comportamiento nos ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

Con la aparición del J2EE, todo un nuevo catálogo de patrones de diseño apareció. Desde que J2EE es una arquitectura por si misma que involucra otras arquitecturas, incluyendo servlets, JavaServer

Pages, Enterprise JavaBeans, y más, merece su propio conjunto de patrones específicos para diferentes aplicaciones empresariales. Veamos el catalogo de patrones en 3 capas: Presentación, Negocios e Integración.

Cuadro 3: Patrón Modelo Vista Controlador-Capa de Presentación

Decorating Filter / Intercepting Filter	Un objeto que está entre el cliente y los componentes Web. Este procesa las peticiones y las respuestas.
Front Controller/ Front Component	Un objeto que acepta todos los requerimientos de un cliente y los direcciona a manejadores apropiados. El patrón Front Controller podría dividir la funcionalidad en 2 diferentes objetos: el Front Controller y el Dispatcher. En ese caso, El Front Controller acepta todos los requerimientos de un cliente y realiza la autenticación, y el Dispatcher direcciona los requerimientos a manejadores apropiada.
View Helper	Un objeto helper que encapsula la lógica de acceso a datos en beneficio de los componentes de la presentación. Por ejemplo, los JavaBeans pueden ser usados como patrón View Helper para las páginas JSP.
Composite view	Un objeto vista que está compuesto de otros objetos vista. Por ejemplo, una página JSP que incluye otras páginas JSP y HTML usando la directiva include o el action include es un patrón Composite View.
Service To Worker	Es como el patrón de diseño MVC con el Controlador actuando como Front Controller pero con una cosa importante: aquí el Dispatcher (el cual es parte del Front Controller) usa View Helpers a gran escala y ayuda en el manejo de la vista.
Dispatcher View	Es como el patrón de diseño MVC con el controlador actuando como Front Controller pero con un asunto importante: aquí el Dispatcher (el cual es parte del Front Controller) no usa View Helpers y realiza muy poco trabajo en el manejo de la vista. El manejo de la vista es manejado por los mismos componentes de la vista

Cuadro 4: Patrón Modelo Vista Controlador-Capa de Negocios

Business Delegate	Un objeto que reside en la capa de presentación y en beneficio de los otros componentes de la capa de presentación llama a métodos remotos en los objetos de la capa de negocios.
Value Object/ Data Transfer Object/ Replicate Object	Un objeto serializable para la transferencia de datos sobre la red.
Session Façade/ Session Entity Façade/ Distributed Façade	El uso de un bean de sesión como una fachada (facade) para encapsular la complejidad de las interacciones entre los objetos de negocio y participantes en un flujo de trabajo. El Session Façade maneja los objetos de negocio y proporciona un servicio de acceso uniforme a los clientes.
Aggregate Entity	Un bean entidad que es construido o es agregado a otros beans de entidad.
Value Object Assembler	Un objeto que reside en la capa de negocios y crea Value Objects cuando es requerido.
Value List Handler/ Page-by-Page Iterator/ Paged List	Es un objeto que maneja la ejecución de consultas SQL, caché y procesamiento del resultado. Usualmente implementado como beans de sesión.
Service Locator	Consiste en utilizar un objeto Service Locator para abstraer toda la utilización JNDI y para ocultar las complejidades de la creación del contexto inicial, de búsqueda de objetos home EJB y recreación de objetos EJB. Varios clientes pueden reutilizar el objeto Service Locator para reducir la complejidad del código, proporcionando un punto de control.

Cuadro 5: Patrón Modelo Vista Controlador-Capa de Integración

Data Access Object Service Activator	Consiste en utilizar un objeto de acceso a datos para abstraer y encapsular todos los accesos a la fuente de datos. El DAO maneja la conexión con la fuente de datos para obtener y almacenar datos.
Service Activator	Se utiliza para recibir peticiones y mensajes asíncronos de los clientes. Cuando se recibe un mensaje, el Service Activator localiza e invoca a los métodos de los componentes de negocio necesarios para cumplir la petición de forma asíncrona.

2.2.3. Programación Orientada a Objetos (14)

Los objetos son entidades que combinan estado, comportamiento e identidad. El estado está compuesto de datos, y el comportamiento por procedimientos o métodos. La identidad es una propiedad de un objeto que lo diferencia del resto. La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

De esta forma, un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, los objetos disponen de mecanismos de interacción llamados métodos que favorecen la comunicación entre ellos. Esta comunicación favorece a su vez el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan ni deben separarse el estado y el comportamiento.

Los métodos y atributos están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. El programador debe pensar indistintamente en ambos conceptos, sin separar ni darle mayor importancia a ninguno de ellos. Hacerlo podría resultar en seguir el hábito erróneo de crear clases contenedoras de información por un lado y clases con métodos que la manejen por el otro. De esta manera se estaría llegando a una programación estructurada camuflada en un lenguaje de programación orientado a objetos.

Esto difiere de la programación estructurada tradicional, en la que los datos y los procedimientos están separados y sin relación, ya que lo único que se busca es el procesamiento de unos datos de entrada para obtener otros de salida. La programación estructurada anima al programador a pensar sobre todo en términos de procedimientos o funciones, y en segundo lugar en las estructuras de datos que esos procedimientos manejan. En la programación estructurada sólo se escriben funciones que procesan datos. Los programadores que emplean éste nuevo paradigma, en cambio, primero definen objetos para luego enviarles mensajes solicitándoles que realicen sus métodos por sí mismos.

a) Orígenes

Los conceptos de la programación orientada a objetos tienen origen en Simula 67, un lenguaje diseñado para hacer simulaciones, creado por Ole-Johan Dahl y Kristen Nygaard del Centro de Cómputo Noruego en

Oslo. Según se informa, la historia es que trabajaban en simulaciones de naves, y fueron confundidas por la explosión combinatoria de cómo las diversas cualidades de diversas naves podían afectar unas a las otras. La idea ocurrió para agrupar los diversos tipos de naves en diversas clases de objetos, siendo responsable cada clase de objetos de definir sus propios datos y comportamiento. Fueron refinados más tarde en Smalltalk, que fue desarrollado en Simula en Xerox PARC (y cuya primera versión fue escrita sobre Basic) pero diseñado para ser un sistema completamente dinámico en el cual los objetos se podrían crear y modificar "en marcha" en lugar de tener un sistema basado en programas estáticos.

La programación orientada a objetos tomó posición como el estilo de programación dominante a mediados de los años ochenta, en gran parte debido a la influencia de C++, una extensión del lenguaje de programación C. Su dominación fue consolidada gracias al auge de las Interfaces gráficas de usuario, para los cuales la programación orientada a objetos está particularmente bien adaptada. En este caso, se habla también de programación dirigida por eventos.

Las características de orientación a objetos fueron agregadas a muchos lenguajes existentes durante ese tiempo, incluyendo Ada, BASIC, Lisp, Pascal, entre otros. La adición de estas características a los lenguajes que no fueron diseñados inicialmente para ellas condujo a menudo a problemas de compatibilidad y a la capacidad de mantenimiento del código. Los lenguajes orientados a objetos "puros", por otra parte, carecían de las características de las cuales muchos

programadores habían venido a depender. Para saltar este obstáculo, se hicieron muchas tentativas para crear nuevos lenguajes basados en métodos orientados a objetos, pero permitiendo algunas características imperativas de maneras "seguras". El Eiffel de Bertrand Meyer fue un temprano y moderadamente acertado lenguaje con esos objetivos pero ahora ha sido esencialmente reemplazado por Java, en gran parte debido a la aparición de Internet, y a la implementación de la máquina virtual de Java en la mayoría de navegadores. PHP en su versión 5 se ha ido modificando y soporta una orientación completa a objetos, cumpliendo todas las características propias de la orientación a objetos.

b) Características

• Abstracción

La abstracción consiste en captar las características esenciales de un objeto, así como su comportamiento. Por ejemplo, en los automóviles, ¿Qué características podemos abstraer de ellos? O lo que es lo mismo ¿Qué características semejantes tienen todos los automóviles? Todos tendrán una marca, un modelo, número de chasis, peso, llantas, puertas, ventanas, etc. Y en cuanto a su comportamiento todos los automóviles podrán acelerar, frenar, retroceder, etc.

En los lenguajes de programación orientada a objetos, el concepto de Clase es la representación y el mecanismo por el cual se gestionan las abstracciones.

- **Encapsulamiento**

El encapsulamiento consiste en unir en la Clase las características y comportamientos, esto es, las variables y métodos. Es tener todo esto en una sola entidad. En los lenguajes estructurados esto era imposible. La utilidad del encapsulamiento va por la facilidad para manejar la complejidad, ya que tendremos a las Clases como cajas negras donde sólo se conoce el comportamiento pero no los detalles internos, y esto es conveniente porque nos interesará saber qué hace la Clase pero no será necesario saber cómo lo hace.

- **Herencia**

La herencia es uno de los conceptos más cruciales en la POO. La herencia básicamente consiste en que una clase puede heredar sus variables y métodos a varias subclases (la clase que hereda es llamada superclase o clase padre). Esto significa que una subclase, aparte de los atributos y métodos propios, tiene incorporados los atributos y métodos heredados de la superclase. De esta manera se crea una jerarquía de herencia.

- **Polimorfismo**

Es la capacidad de que diferentes objetos reaccionen de distinta forma a un mismo mensaje. Es la capacidad de referirse a objetos de clases distintas en una jerarquía utilizando el mismo elemento de programa (método) para realizar la misma operación, pero de manera diferente.

2.2.4. El Lenguaje de Modelado Unificado (15)

En todas las disciplinas de la Ingeniería se hace evidente la importancia de los modelos ya que describen el aspecto y la conducta de "algo". Ese "algo" puede existir, estar en un estado de desarrollo o estar, todavía, en un estado de planeación. Es en este momento cuando los diseñadores del modelo deben investigar los requerimientos del producto terminado y dichos requerimientos pueden incluir áreas tales como funcionalidad, performance y confiabilidad. Además, a menudo, el modelo es dividido en un número de vistas, cada una de las cuales describe un aspecto específico del producto o sistema en construcción.

El modelado sirve no solamente para los grandes sistemas, aun en aplicaciones de pequeño tamaño se obtienen beneficios de modelado, sin embargo es un hecho que entre más grande y más complejo es el sistema, más importante es el papel de que juega el modelado por una simple razón: "El hombre hace modelos de sistemas complejos porque no puede entenderlos en su totalidad".

UML es una técnica para la especificación sistemas en todas sus fases. Nació en 1994 cubriendo los aspectos principales de todos los métodos de diseño antecesores y, precisamente, los padres de UML son Grady Booch, autor del método Booch; James Rumbaugh, autor del método OMT e Ivar Jacobson, autor de los métodos OOSE y Objectory. La versión 1.0 de UML fue liberada en Enero de 1997 y ha sido utilizado con éxito en sistemas construidos para toda clase de

industrias alrededor del mundo: hospitales, bancos, comunicaciones, aeronáutica, finanzas, etc.

UML es un Lenguaje, que proporciona un vocabulario y las reglas para combinar palabras de ese vocabulario con el objetivo de posibilitar la comunicación.

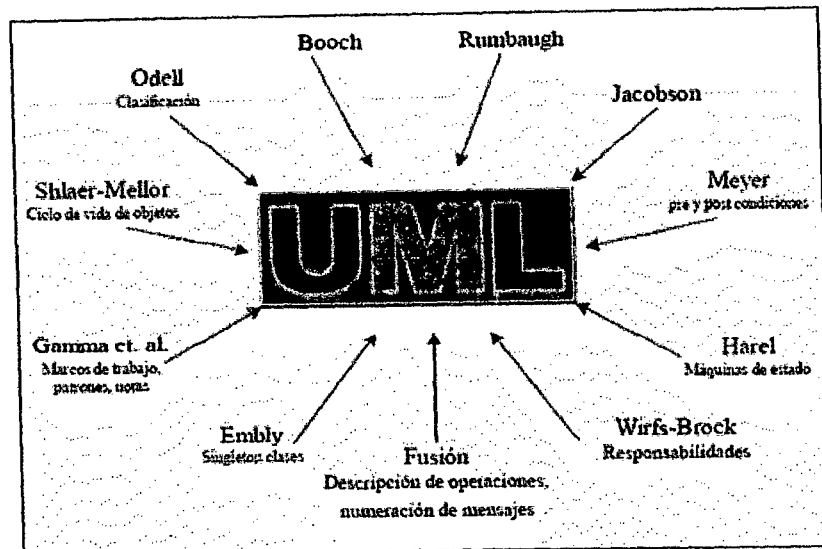
UML es un Lenguaje para Visualizar, es algo más que un simple montón de símbolos gráficos; detrás de cada símbolo en la notación UML hay una semántica bien definida. De esta manera, un desarrollador puede escribir un modelo en UML y otro desarrollador o incluso otra herramienta, puede interpretar este modelo sin ambigüedad.

UML es un Lenguaje para especificar, significa construir modelos precisos y completos.

UML es un Lenguaje para construir, sus modelos pueden conectarse de forma directa a una gran variedad de lenguajes de programación.

UML es un lenguaje para documentar, una organización produce toda clase de artefactos que incluyen requisitos, arquitectura, diseño, código fuente, planificación de proyectos, pruebas, prototipos y versiones.

Gráfico 9: Origen UML



Fuente: Servicios Open Modeling (16)

Los principales beneficios de UML son: (17)

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos 'business'.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

2.2.4.1 Descripción de Estereotipos WAE (18)

- ❖ **Página del Servidor (Clase):** Una página del servidor representa una página Web que tiene scripts que son ejecutan en el servidor. Estos scripts actúan recíprocamente con recursos en el servidor (base de datos, lógica del negocio, sistemas externos). Las operaciones del objeto representan las funciones en el script, y sus atributos representan las variables que son visibles en el

alcance de la página (accesible por todas las funciones en la página).

- ❖ **Página del Cliente (Clase):** Una página del cliente es una página Web estructurada en HTML. Como cualquier página HTML esta página es una mezcla de datos, presentación y lógica de eventos. Las páginas del cliente son dadas por el browser del cliente, y puede contener scripts que son interpretadas por el browser. Las funciones de las páginas clientes es mapear las funciones de las etiquetas de los scripts en la página. Los atributos de la página cliente se mapean en variables declaradas en los scripts y las etiquetas que son accesibles por cualquier función de la páginas. Las páginas del cliente pueden tener asociaciones con otras páginas, cliente o de servidor.
- ❖ **Formulario (Clase):** Una clase formulario es una colección de campos de entrada que son parte de una página cliente. Una clase formulario mapea directamente las etiquetas del formulario a HTML. Sus atributos representan los campos de entrada del formulario de HTML. Un formulario no tiene operaciones, ya que las operaciones no pueden encapsularse en un formulario. Cualquier operación que interactúa con la forma está en la página que contiene el formulario.
- ❖ **Frameset (Clase):** Un frameset es un contenedor múltiples páginas Web. El área es rectangular y es dividida en marcos rectangulares más pequeños. Cada marco puede asociarse con

sólo nombrar un “target”, aunque no necesariamente. El contenido de un marco pueden ser una página Web u otro conjunto de marcos. Una clase frameset estereotipada se mapea directamente a una página Web de frameset y a etiquetas frame de HTML. Ya que un frameset es una página de cliente, puede tener operaciones y atributos, pero éstos sólo son activados por el browser que no tienen frames.

- ❖ **Target (Clase):** Un target es un compartimiento nombrado desde una ventana del browser donde las páginas Web pueden ser mostradas. El nombre de la clase estereotipada es el nombre del target. Típicamente un target es un marco en una ventana definida por un frameset, sin embargo un target podría ser completamente una nueva instancia del browser. Las asociaciones “Target link” especifican targets como el lugar donde una página Web va a ser mostrada.
- ❖ **Objeto Java Script (Clase):** En un browser habilitado para Java Script es posible simular a objetos definidos por el usuario con funciones de JavaScript. Las instancias de los objetos “JavaScript” sólo existen en el contexto de las páginas del cliente.
- ❖ **Link (Asociación):** Un link es un apuntador desde una página del cliente a otra. En un diagrama de clase un link es una asociación entre la página del cliente y cualquier otra página del cliente o una

página del servidor. Una asociación link se mapea directamente a la etiqueta HTML.

- ❖ **Target Link (Asociación):** Similar a un link, donde la página asociada se da con otro target. Esta asociación se mapea directamente a la etiqueta HTML, con el target especificado como atributo de la etiqueta.
- ❖ **Frame Content (Asociación):** Es una asociación de agregación que expresa frames contenidos en otra página o target. Una asociación Frame Content también puede apuntar a otro frameset y puede indicar frames anidados.
- ❖ **Submit (Asociación):** Una asociación "submit" siempre es entre un formulario y una página del servidor. Los formularios envían los valores de sus campos al servidor a través de las páginas del servidor para su procesamiento. El servidor Web procesa la página del servidor la cual acepta y usa la información enviada por el formulario.
- ❖ **Build (Asociación):** La relación "builds" es una relación especial que hace un puente entre el cliente y la página del servidor. Las páginas del servidor solo existen en el servidor. Ellos son utilizados para construir páginas del cliente. La relación builds identifica que página del servidor es responsable para la creación de una página cliente. Está es una relación direccional, ya que la

página del cliente no contiene conocimiento de cómo entro en existencia.

- ❖ **Redirect (Asociación):** Es una asociación unidireccional con otra página Web. Puede dirigirse a los dos, a las páginas del cliente y páginas del servidor. Si la relación se origina de una página de servidor entonces indica que el procesamiento de la petición puede continuar en otra página. Esto no indica que la página de destino participa en la construcción de una página cliente, sólo si pudo. Esta relación particular no es completamente estructural, ya que la invocación real de una operación de redirección debe ser programada en el código de la página que origina la llamada. Si la relación se origina de una página del cliente entonces esto indica que la página del destino será pedida automáticamente por el browser, sin la intervención del usuario. Puede ponerse un valor de retraso de tiempo antes que la segunda página sea pedida.
- ❖ **Página Web (Componente):** Una página componente es una página Web. Puede ser pedida por su nombre por un browser. Una página puede o no contener scripts de cliente o de servidor. Típicamente una página es un archivo texto accesible por el servidor Web. Una página produce un documento estructurado en HTML que se envía en respuesta a la petición de un browser.
- ❖ **ASP Pages (Componente):** Estas páginas Web son implementadas del lado del servidor en ASP. Este componente es

aplicado solo en Microsoft Active Server Pages, basado en aplicaciones del entorno.

❖ **JSP Pages (Componente):** Estas páginas Web son implementadas del lado del servidor en JSP. Este componente es aplicado solo en aplicaciones Web desarrolladas en Java.

❖ **Servlet (Componente):** Un servlet es un componente de Java que corre en el lado del servidor. Este componente es aplicado solo en aplicaciones Web desarrolladas en Java.

2.2.5. El Proceso Unificado Rational (19)

2.2.5.1 Concepto

El Proceso Unificado Rational es un proceso de Ingeniería de software. Este provee un enfoque disciplinado de tareas y responsabilidades dentro de una organización para su desarrollo. Su meta es asegurar la producción de software de una calidad superior que satisfaga las necesidades de sus usuarios finales, dentro de una administración y cronograma establecido.

El Proceso Unificado Rational realiza la productividad del equipo, proporcionándole el acceso fácil a cada miembro del equipo que esta desarrollando el sistema, a una base de conocimientos con las pautas, plantillas y guías de la herramienta para toda actividad crítica de desarrollo. Teniendo todos los miembros del equipo el acceso a la misma base de conocimiento, no importa si se trabaja

con los requerimientos, diseño, prueba, administración del proyecto, o administración de la configuración, se asegura que todos los miembros del equipo comparten un lenguaje común, un proceso común y una visión de cómo se desarrollará el software.

Las actividades del Proceso Unificado Rational crean y mantienen modelos más que enfocarse en cantidades de producción de documentos en general. El Proceso Unificado Rational enfatiza el desarrollo y mantenimiento de modelos, para una representación rica en semántica para el desarrollo del software del sistema.

2.2.5.2 Orígenes

El antecedente más importante lo ubicamos en 1967 con la Metodología Ericsson (Ericsson Approach), ésta es una aproximación de desarrollo basada en componentes, que introdujo el concepto de caso de uso; entre los años de 1987 a 1995 Jacobson funda la compañía "Objectory AB" y lanza el proceso de desarrollo Objectory (abreviación de Object Factory), posteriormente en 1995 "Rational Software Corporation" adquiere "Objectory AB" y es entre 1995 y 1997 que se desarrolla "Rational Objectory Process (ROP)" fruto del encuentro y evolución de Objectory 3.8 y la Metodología Rational (Rational Approach) que adopta por primera vez UML como lenguaje de modelamiento.

A principios de los noventas, la guerra de los métodos hizo evidente la necesidad de unificar criterios, es así como Grady Booch autor

del método Booch y James Rumbaugh (desarrollador para General Electric) se unieron en Rational en 1994, después en 1995 se une Jacobson y gracias al esfuerzo de varias compañías y metodologistas evolucionó UML hasta ser un estándar en 1997, el cual es adoptado en todos los modelos del ROP. Desde ese entonces y a la cabeza de Booch, Jacobson y Rumbaugh, Rational ha desarrollado e incorporado diversos elementos para expandir el ROP, destacándose especialmente el flujo de trabajo conocido como modelamiento del negocio, es así como en junio del 1998 se lanza Rational Unified Process 5.0 evolucionado hasta el momento de elaboración de este documento bajo el nombre de RUP.

2.2.5.3 Características

a) Guiado/Manejado por casos de uso: La razón de ser de un sistema software es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.

b) Centrado en arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como

sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.

c) Iterativo e Incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un miniproyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

d) Desarrollo basado en componentes: La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o que se desarrollen y maduran sus componentes.

e) Utilización de un único lenguaje de modelamiento: UML es adoptado como único lenguaje de modelamiento para el desarrollo de todos los modelos.

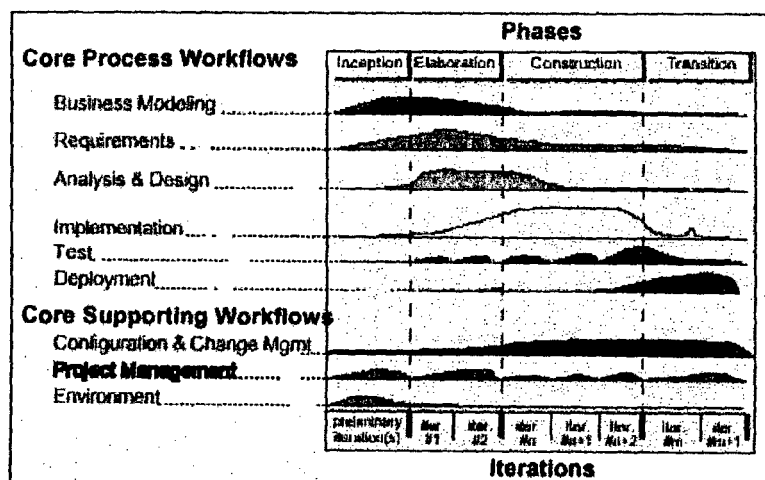
f) Proceso Integrado: Se establece una estructura que abarque los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de

calidad, gestión del proyecto y control de configuración; el proceso unificado establece una estructura que integra todas estas facetas. Además esta estructura cubre a los vendedores y desarrolladores de herramientas para soportar la automatización del proceso, soportar flujos individuales de trabajo, para construir los diferentes modelos e integrar el trabajo a través del ciclo de vida y a través de todos los modelos.

2.2.5.4 Fases en el Ciclo de Desarrollo

Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases que se describirán a continuación, la figura muestra las fases de desarrollo y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica en cual de los flujos se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.

Gráfico 10: Fases RUP



Fuente: Genbiz (20)

A) Fase Inicial

Su objetivo principal es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto.

B) Fase Elaboración

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

C) Fase Construcción

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta.

D) Fase Transición

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos.

2.2.6. Elementos del Proceso Administrativo

Según Henry Fayol: "El acto de administrar se define como: planear, organizar, dirigir, coordinar y controlar" (21)

2.2.6.1 Planeación

La Planeación es decir por adelantado, qué hacer, cómo y cuando hacerlo, y quién ha de hacerlo. La planeación cubre la brecha que va desde donde estamos hasta donde queremos ir. Hace posible que ocurran cosas que de otra manera nunca sucederían; aunque el futuro exacto rara vez puede ser predicho, y los factores fuera de control pueden interferir con los planes mejor trazados, a menos que haya planeación, los hechos son abandonados al azar.

La planeación es un proceso intelectualmente exigente; requiere la determinación de los cursos de acción y la fundamentación de las decisiones, en los fines, conocimientos y estimaciones razonadas. La tarea de la planeación es exactamente: minimización del riesgo y el aprovechamiento de las oportunidades. La naturaleza esencial de la planeación puede ponerse de relieve mediante sus cuatro componentes principales que son:

- 1) Contribución a los objetivos y propósitos.
- 2) Primacía de la planeación.
- 3) Extensión de la administración.
- 4) Eficacia de la planeación.

El propósito de cada plan es facilitar el logro de los objetivos de la empresa. Puesto que las organizaciones empresariales de organización, integración, dirección, liderazgo y control están

encaminadas a apoyar el logro de los objetivos empresariales, la planeación lógicamente precede a la ejecución de todas las funciones.

Para Sánchez Guzmán la planeación es: "Aquella herramienta de la administración que nos permite determinar el curso concreto de acción que debemos seguir, para lograr la realización de los objetivos previstos".

Según Terry (22), la planeación es "Seleccionar información y hacer suposiciones respecto al futuro para formular las actividades necesarias para realizar los objetivos organizacionales, está compuesta de numerosas decisiones orientadas al futuro.- Representa el destinar pensamiento y tiempo ahora para una inversión en el futuro".

2.2.6.2 Organización (21)

Organización es el establecimiento de la estructura necesaria para la sistematización racional de los recursos, mediante la determinación de jerarquías, disposición, correlación y agrupación de actividades, con el fin de poder realizar y simplificar las funciones del grupo social.

Elementos:

Estructura. La organización implica el establecimiento del marco fundamental en el que habrá el grupo social, ya que establece la disposición y la correlación de las funciones, jerarquías y actividades necesarias para lograr los objetivos.

Sistematización. Las actividades y recursos de la empresa, deben de coordinarse racionalmente para facilitar el trabajo y la eficiencia.

Agrupación y asignación de actividades y responsabilidades. En la organización surge la necesidad de agrupar, dividir y asignar funciones a fin de promover la especialización.

Jerárquica. La organización, como estructura, origina la necesidad de establecer niveles de autoridad y responsabilidad dentro de la empresa.

Simplificación de funciones. Uno de los objetivos básicos de la organización es establecer los métodos más sencillos para realizar el trabajo de la mejor manera posible.

Importancia de la Organización.

Los fundamentos básicos que demuestran la importancia de la organización son:

1. Es de carácter continuo; jamás se puede decir que ha terminado, dado que la empresa y sus recursos están sujetos a cambios constantes (expansión, contracción, nuevos productos, etc.), lo que obviamente redundará en la necesidad de efectuar cambios en la organización.
2. Es un medio a través del cual se establece la mejor manera de lograr los objetivos del grupo social.
3. Suministra los métodos para que se puedan desempeñar las actividades eficientemente, con un mínimo de esfuerzo.

4. Evita la lentitud e ineficiencia de las actividades, reduciendo los costos e incrementando la productividad.
5. Reduce o elimina la duplicidad de esfuerzos, al delimitar funciones y responsabilidades.

2.2.6.3 Dirección (23)

Esta etapa del proceso administrativo, llamada también ejecución, comando o liderazgo, es una función de tal trascendencia, que algunos autores consideran que la administración y la dirección son una misma cosa.

Elementos.

- Ejecución de los planes de acuerdo con la estructura organizacional.
- Motivación.
- Guía o conducción de los esfuerzos de los subordinados.
- Comunicación.
- Supervisión.
- Alcanzar las metas de la organización.

Importancia de la Dirección.

La dirección es trascendental porque:

- Pone en marcha todos los lineamientos establecidos durante la planeación y la organización.
- A través de ella se logran las formas de conducta más deseables en los miembros de la estructura organizacional.

- La dirección eficiente es determinante en la moral de los empleados y consecuentemente, en la productividad.
- Su calidad se refleja en el logro de los objetivos, la implementación de métodos de organización, y en la eficacia de los sistemas de control.
- A través de ella se establece la comunicación necesaria para que la organización funcione.

Principios.

- De la armonía del objetivo o coordinación de intereses.
- Impersonalidad de mando.
- De la supervisión directa.
- De la vía jerárquica.
- De la resolución del conflicto.
- Aprovechamiento del conflicto.

De la armonía del objetivo o coordinación de intereses:

La dirección será eficiente en tanto se encamine hacia el logro de los objetivos generales de la empresa.

Así mismo, establece que los objetivos de todos los departamentos y secciones deberán relacionarse armoniosamente para lograr el objetivo general.

Impersonalidad de mando:

Se refiere a que la autoridad y su ejercicio (el mando), surgen como una necesidad de la organización para obtener ciertos resultados.

De la supervisión directa:

Se refiere al apoyo y comunicación que debe proporcionar el dirigente a sus subordinados durante la ejecución de los planes, de tal manera que éstos se realicen con mayor facilidad.

De la vía jerárquica:

Es el camino que siguen, pasando por todos los grados de la jerarquía, las comunicaciones que parten de la autoridad superior o de las que le son dirigidas.

De la resolución del conflicto:

Indica la necesidad de resolver los problemas que surjan durante la gestión administrativa, a partir del momento en que aparezcan.

Aprovechamiento del conflicto:

- Experiencia.
- Experimentación.
- Investigación.
- Aplicar la decisión. Consiste en poner en práctica la decisión elegida, por lo que se debe contar con un plan para el desarrollo de la misma. Dicho plan comprenderá: los recursos, los

procedimientos y los programas necesarios para la implantación de la decisión.

2.2.6.4 Coordinación (24)

Según Fayol: "Coordinar es establecer la armonía entre todos los actos de una empresa de manera de facilitar su funcionamiento y procurar el buen éxito.

Es dar al organismo material y social de cada función las proporciones convenientes para que ésta pueda cumplir su misión en forma segura y económica."

Para que reine la armonía entre las diversas partes del organismo material o social se necesita no solamente un buen programa y una buena organización, sino también una coordinación realizada en todo momento.

En una empresa bien coordinada se verifican los hechos siguientes:

- Cada servicio marcha de acuerdo con los otros.
- En cada servicio las divisiones y subdivisiones son exactamente informadas sobre la parte que les corresponde en la obra común y sobre la ayuda mutua que deben prestarse.
- El programa de acción de los diversos servicios es puesto constantemente en armonía con las circunstancias. Tal resultado exige una dirección inteligente, experimentada y activa.

Hay que admitir que estas tres condiciones no se cumplen siempre, pues pueden observarse en algunas empresas los signos siguientes que revelan una evidente incoordinación:

- Cada servicio ignora o quiere ignorar la existencia de los otros. Marcha como si tuviera en sí mismo su fin y su razón de ser, sin preocuparse de los servicios adyacentes ni del conjunto de la empresa.
- El tabique aislador existe entre las divisiones y las oficinas de un mismo servicio como entre los diferentes servicios. La gran preocupación de cada uno es poner su responsabilidad personal a cubierto detrás de un papel, de una orden o de una circular.
- Nadie piensa en el interés general. La iniciativa y la abnegación se hallan ausentes. Esta disposición del personal (es) de consecuencias desastrosas para la empresa. Es el resultado de una coordinación nula o insuficiente. Un buen personal no tarda en debilitarse.

Uno de los mejores medios de mantener el entusiasmo del personal y facilitarle el cumplimiento de su deber es la conferencia de jefes de servicio.

2.2.6.5 Control (25)

Los gerentes siempre han encontrado conveniente comprobar o vigilar lo que se está haciendo para asegurar que el trabajo de otros está progresando en forma satisfactoria hacia el objetivo predeterminado. Establecer un buen plan, distribuir las actividades componentes requeridas para ese plan y la ejecución exitosa de cada miembro no asegura que la empresa será un éxito. Pueden presentarse discrepancias, malas interpretaciones y obstáculos

inesperados y habrán de ser comunicados con rapidez al gerente para que se emprenda una acción correctiva.

Actividades importantes de control

- Comparar los resultados con los planes generales.
- Evaluar los resultados contra los estándares de desempeño.
- Idear los medios efectivos para medir las operaciones.
- Comunicar cuales son los medios de medición.
- Transferir datos detallados de manera que muestren las comparaciones y las variaciones.
- Sugerir las acciones correctivas cuando sean necesarias.
- Informar a los miembros responsables de las interpretaciones.
- Ajustar el control a la luz de los resultados del control.

2.2.7. La Biblioteca como Organización

2.2.7.1 La Organización Biblioteca (26)

La biblioteca se caracteriza como institución por su relación con el entorno, por lo que para la comprensión de su funcionamiento, debemos estudiar simultáneamente los procesos internos y los condicionantes externos. La biblioteca deberá enfrentar el dinamismo de los procesos de cambio, la constante innovación tecnológica, la ampliación de las redes de relaciones. En definitiva, el cambio en las formas del trabajo que conlleva la sociedad de la información. La biblioteca constituye un sistema complejo, que generalmente es enfocado bien como un sistema de información,

bien como una organización de servicios (27). Desde esta doble óptica sostendremos que su automatización debe abarcar el conjunto de los aspectos informacionales y de servicio.

La automatización afectará tanto a aquellos procesos relacionados con la información depositada en ella y las técnicas sobre ella aplicadas, como aquellos relacionados con su dimensión de organización productora de servicios y por lo tanto en relación con el usuario, puesto que la biblioteca como institución justifica su necesidad social en su voluntad de servicio (28).

Además de los acercamientos sectoriales a las bibliotecas (Bibliotecas públicas, escolares, especializadas, universitarias, etc.) es necesario partir de un modelo genérico de biblioteca para entender la globalidad del proyecto de informatización. Estos modelos teóricos generales tratan de "ser representaciones simplificadas de la realidad" y plantear conclusiones y principios generales (29).

Aunque las bibliotecas proyecten una imagen social reduccionista, apoyada en tópicos repetitivos y superficiales, como proyecto comunitario, poseen una complejidad intrínseca, derivada de su papel de mediadoras en los procesos informativo-documentales. Para comprender de forma general la situación, hay que considerar que la información posee dos clases de complejidades; la complejidad ordenada de los mecanismos técnicos y la complejidad desorganizada del hombre y de los sistemas sociales.

La complejidad que surge de "la intersección de los sistemas tecnológicos con los sistemas sociales origina un nudo de

relaciones de gran complejidad que se manifiesta específicamente por fenómenos relacionados con el desorden, la incertidumbre, la desorganización, la inestabilidad, la entropía, la borrosidad, etc.”
(30)

La biblioteca se mueve en un entorno de complejidad e inestabilidad interna y externa. Estos dos factores influyen en la estructura, el estilo de gestión y la cultura corporativa. De entre las características internas de la biblioteca, aquellas que la definen, hemos seleccionado la información y el servicio. Desde el punto de vista de la información, la biblioteca puede ser tratada como un sistema de información, en concreto como un sistema de información documental. Desde el punto de vista del servicio, la biblioteca será considerada una organización de servicios. Comprenderemos a la biblioteca como una combinación dinámica de procesos informativos que se producen en un contexto de servicio, que justifican la necesidad social de la biblioteca.

Junto a estos dos enfoques, estudiaremos aquellos elementos del entorno que más afecta en su definición, entendiéndola como una entidad abierta, para llegar a la conceptualización de la biblioteca-red, como paradigma para entender la biblioteca en la sociedad de la información.

2.2.7.2 La Biblioteca y el Entorno (31)

En el ámbito teórico de la gestión de información, el ambiente o entorno es uno de los conceptos clave, sobre todo en las organizaciones de servicios. Se trata de atender a una comunidad.

Las bibliotecas/centros de información, como cualquier organización, se vinculan con un conjunto de influencias externas que están fuera de su control. Los elementos que conforman el ambiente de las organizaciones de información ejercen influencias generales o específicas.

Hemos situado la biblioteca en el contexto general de la sociedad de la información, como escenario para el desarrollo de servicios. Los aspectos generales ejercen una influencia indirecta y entre ellos pueden mencionarse el desarrollo tecnológico y la situación socioeconómica, los cuales hemos tratado en los capítulos precedentes. Junto a la necesidad de concebir el entorno con una perspectiva globalizadora, es necesario delimitar con precisión los aspectos concretos que afectan más sensiblemente a la organización.

La gestión de bibliotecas y centros de información se realiza cada vez más aplicando técnicas de gestión estándares. Estos centros no pueden ignorar los cambios psicológicos, técnicos, sociológicos y políticos que ocurren en los entornos externos e interno de sus organizaciones. Se considera necesario situar estratégicamente a la biblioteca en su entorno para asegurar el éxito.

Las relaciones con el entorno son difíciles de sistematizar. Podemos distinguir dos entornos

- Entorno transaccional. Aquellos subsistemas del entorno con los que la organización interactúa directamente.

- Entorno contextual. La parte del entorno no transaccional y que influye o es influida por la organización. Se ejerce nulo control sobre el mismo.

De una manera amplia, los factores externos que condicionan la actividad de la biblioteca son muy amplios: tecnología, valores, legislación, usuarios, niveles educativos, proveedores, presupuestos públicos, mercado de trabajo, tendencias sociales, etc.

CAPITULO III

METODOLOGÍA

3.1. Método de Investigación

3.1.1. Tipo de Investigación

Aplicada, según la clasificación de Mario Tamayo y Tamayo (32), el tipo de investigación es aplicada, porque busca la aplicación o utilización de los conocimientos que se adquieren.

3.1.2. Nivel de Investigación

Correlacional: Causa-Efecto. Es aquel tipo de estudio que persigue medir el grado de relación existente entre dos o más conceptos o variables.

3.1.3. Diseño de Investigación

Se aplicará el método de diseño en sucesión o en línea, conocido también como método Pre Test – Post Test el cual consiste en:

- Medición de la Variable Dependiente antes de aplicar la Variable Independiente (Pre - Test).
- Aplicación de la Variable Independiente.
- Medición de la Variable Dependiente después de aplicar la Variable Independiente (Post- Test).

Se puede representar mediante la siguiente simbología: GE, O1, X, O2.

Donde:

GE : Los usuarios de la Biblioteca Central de la Universidad Nacional de Ucayali (Grupo Experimental)

O₁ : Análisis de los resultados, antes de la instalación del Sistema Web con Tecnología J2EE.

X : Es el Sistema Web con tecnología J2EE

O₂ : Análisis de los resultados, después de la instalación del Sistema Web con Tecnología J2EE.

Al final de la investigación se establecerán diferencias entre O₁ y O₂ para determinar si existe mejoramiento o no en los indicadores expresados en la variable independiente.



3.2. Población y Muestra

3.2.1. Población

Para la presente investigación está dada por:

- Los alumnos matriculados de la Universidad Nacional de Ucayali que suman un total de 2902, distribuidos de la siguiente manera:

Cuadro 6: Distribución Alumnos Matriculados Semestre 2007-II

Facultad	Escuela Profesional	Cantidad	Porcentaje %
Ciencias Administrativas y Contables	Administración	295	10
	Contabilidad	375	13
Ingeniería de sistemas	Ingeniería de Sistemas	297	10
Ciencias Agropecuarias	Agronomía	327	11
	Agronomía (Sede Aguaytia)	88	3
Ciencias de la Salud	Enfermería	273	10
Medicina Humana	Medicina Humana	172	5
Ciencias Forestales	Ingeniería Forestal	299	10
	Ingeniería Ambiental	57	2
Derecho y Ciencias Políticas	Derecho	372	13
Educación y Ciencias Sociales	Educación Primaria	142	5
	Lengua y Literatura	46	2
	Ciencias Sociales	25	1
	Ciencias Naturales y Medio Ambiente	18	1
	Matemática, Física e Informática	54	2
	Estudios Generales	62	2
Total General		2902	100%

- Los docentes nombrados y contratados de la Universidad Nacional de Ucayali que suman un total de 148 y 133 respectivamente, distribuidos de la siguiente manera:

Cuadro 7: Distribución Docentes Nombrados y Contratados

Categoría	Nombrados	Contratados	Cantidad Total	Porcentaje %
Principales	30	4	34	12
Asociados	56	13	69	24
Auxiliares	51	103	154	55
Jefe de Práctica	11	13	24	9
Total	148	133	281	100%

- Los trabajadores de la Biblioteca Central de la Universidad Nacional de Ucayali que suman un total de 10, distribuidos de la siguiente manera:

Cuadro 8: Distribución Trabajadores Biblioteca Central

Áreas	Cantidad
Jefatura	1
Proceso Técnico	2
Secretaría	1
Servicio de Atención al Público	2
Recepción	1
Limpieza	3
Total	10

3.2.2. Muestra

La muestra para el presente estudio de investigación, se estimó siguiendo los criterios que ofrece la estadística, por ello se hizo uso del método Probabilístico, mediante la técnica del Muestreo Proporcional, teniendo en cuenta las siguientes consideraciones:

N = Población

n = Tamaño de la muestra necesaria.

P = Probabilidad de que evento ocurra (50%)

Q = Probabilidad de que el evento no ocurra (50%)

$Z_{\alpha/2} = 1,96$

$\varepsilon = 0,05$

Cuya formula matemática es:

$$n = \frac{Z^2_{\alpha/2} P Q N}{\varepsilon^2 (N-1) + Z^2 P Q}$$

Obteniéndose una muestra de 339 encuestas

$$n = 339.35 \approx 339$$

Cuadro 9: Distribución Alumnos para la Muestra

Facultad	Escuela Profesional	Porcentaje %	Cantidad Muestral
Ciencias Administrativas y Contables	Administración	10	34
	Contabilidad	13	44
Ingeniería de sistemas	Ingeniería de Sistemas	10	34
Ciencias Agropecuarias	Agronomía	11	37
	Agronomía (Sede Aguaytia)	3	10
Ciencias de la Salud	Enfermería	10	34
Medicina Humana	Medicina Humana	5	17
Ciencias Forestales	Ingeniería Forestal	10	34
	Ingeniería Ambiental	2	7
Derecho y Ciencias Políticas	Derecho	13	44
Educación y Ciencias Sociales	Educación Primaria	5	17
	Lengua y Literatura	2	7
	Ciencias Sociales	1	3
	Ciencias Naturales y Medio Ambiente	1	3
	Matemática, Física e Informática	2	7
	Estudios Generales	2	7
Total General		100%	339

Para la muestra de los docentes nombrados y contratados con una población de 281 se obtuvo:

$$n = \frac{281 * 1,96^2 * 0,5 * 0,5}{[0,5^2 * (281 - 1)] - [1,96^2 * 0,5 * 0,5]}$$

$$n = 163$$

Cuadro 10: Distribución Docentes para la Muestra

Categoría	Porcentaje %	Cantidad Muestral
Principales	12	19
Asociados	24	39
Auxiliares	55	90
Jefe de Práctica	9	15
Total	100%	163

- Para la muestra de los trabajadores de la Biblioteca Central de la Universidad Nacional de Ucayali se excluyó al personal de limpieza por no estar relacionados directamente con el sistema.

Cuadro 11: Distribución Trabajadores para la muestra

Áreas	Cantidad
Jefatura	1
Proceso Técnico	2
Secretaría	1
Servicio de Atención al Público	2
Recepción	1
Total	7

3.3. Técnicas e instrumentos para la recolección de datos.

3.3.1. Técnicas e instrumentos

Para llevar a cabo el trabajo de campo de la investigación en referencia las técnicas e instrumentos, se muestra en el Cuadro 12.

Cuadro 12: Técnicas e Instrumentos

TECNICAS	INSTRUMENTOS
a. Sistematización bibliográfica y hemerográfica, videoteca.	<ul style="list-style-type: none"> - Fichas bibliográficas. - Fichas de resumen. - Fichas de transcripción. - Fichas de comentario.
b. Observación.	- Guía de observación.
c. Entrevista.	- Guía de entrevista.
d. Encuesta.	- Cuestionario.

3.3.2. Validación de las técnicas e instrumentos de recolección.

A efectos de validar los instrumentos de medición, se realizó una prueba piloto de cada uno de ellos con personas expertas en la confección del mismo. A continuación mencionamos como fueron desarrollados:

a) Sistematización bibliográfica, hemerográfica, videoteca

Técnica mediante la cual recabamos información actualizada en torno al tema de investigación para armar nuestro marco teórico y conceptual, llevando a cabo análisis de contenidos documentales, así como antecedentes que nos permitieron tener ideas más claras en el entendimiento del tema.

Instrumentos:

- Se ficharon diferentes textos; para un mayor orden y facilidad de recolección de datos sustanciales. Asimismo fichamos revistas y captamos información importante vía Internet, para luego armar fichas de comentarios propios.

b) La observación

Mediante esta técnica se realizó el procesamiento perceptivo del comportamiento de los fenómenos, desenvolvimiento de los hechos y acontecimientos en los escenarios que son objeto de nuestra investigación.

Instrumentos:

- Guía de observación y lista de cotejos, donde se consignaron los aspectos más fundamentales y relevantes de la investigación.

c) La entrevista

Entendida la entrevista como una comunicación interpersonal, entre el investigador y los sujetos que constituyen la muestra de la investigación, con tal razón seleccionamos esta técnica con la finalidad de obtener datos en cuanto a los conocimientos básicos. Se usó la entrevista estructurada.

**Instrumento:**

- Guía de entrevista; consistente en un conjunto de preguntas secuenciales, categorizadas y estructuradas con anticipación.

d) La encuesta

Con la finalidad de recoger evidencias sobre el conocimiento, interés y necesidades.

Instrumentos:

- Los cuestionarios y pruebas de ejecución; hoja que contiene un conjunto de preguntas en su mayor parte cerradas, para tener información de primera mano.

3.4. Procedimiento de recolección de datos

Los instrumentos de recolección de datos fueron aplicados a los alumnos, docentes y trabajadores de la Biblioteca Central de la Universidad Nacional de Ucayali, bajo la supervisión continua del asesor del proyecto.

3.5. Tratamiento de datos**3.5.1. Mecanismos para el procesamiento y presentación de datos**

Los datos han sido presentados en tablas, cuadros, figuras y gráficos, analizados con la aplicación de la estadística descriptiva, seguidamente se procedió a analizarlos. Para el procesamiento de los datos se utilizaron las herramientas informáticas como: Ms Excel y SPSS,

presentándose los resultados en cuadros, teniendo en cuenta las variables de la investigación.

Para realizar el procesamiento y posterior presentación de los datos recolectados, de las fuentes primarias (trabajo de campo) y secundarias, que fueron necesarios para un análisis e interpretación clara y precisa de los mismos, se utilizó la estadística.

3.5.2. Mecanismos para el análisis e interpretación de resultados

En esta parte se usó el análisis lógico como el estadístico, para comprobar nuestra hipótesis y poder plantear las conclusiones y recomendaciones a la que arribamos, teniendo en cuenta siempre la confiabilidad y veracidad de los datos obtenidos y procesados con anterioridad.

En lo referente al análisis lógico se realizó observaciones directas y constantes, para ir verificando si los objetivos planteados se cumplían o no, y en cuanto al análisis estadístico se hizo a través de las herramientas descritas en el numeral anterior.

CAPITULO IV

PROCESO DE DESARROLLO DE SOFTWARE

4.1. Fase Inicial

4.1.1. Modelamiento de visión del negocio

A. Documento Visión del Negocio

A.1 Introducción

A.1.1. Propósito

El propósito de este documento es ofrecer un esquema del funcionamiento del sistema, a nivel de procesos, actores, y diagramas del **“Sistema Web con Tecnología J2EE para la Administración en la Biblioteca de la Universidad Nacional de Ucayali”**

A.1.2. Alcance

En este trabajo se realizará el modelamiento del Sistema Web con Tecnología J2EE para la Administración en la Biblioteca así como su implementación, desarrollado por los bachilleres en mención del proyecto de investigación.

El sistema permitirá a los usuarios lo siguiente:

- ✓ Registrar y actualizar el material bibliográfico.
- ✓ Registrar y actualizar usuarios de la biblioteca
- ✓ Registrar préstamos tanto a los alumnos como a los docentes de la UNU.
- ✓ Gestionar multas y sanciones por realización de préstamos.

- ✓ Consultar material disponible vía Web.
- ✓ Realizar consultas y reservas de materiales vía Web.
- ✓ Registrar asistencia de usuarios a la biblioteca.
- ✓ Registrar las lecturas realizadas.
- ✓ Generar reportes

A.2 Posicionamiento

A.2.1 Oportunidad del negocio

El Sistema Web con Tecnología J2EE para la Administración en la Biblioteca mejorará progresivamente la forma en que se posibilita el encuentro de los usuarios con el conocimiento. La introducción de nuevas tecnologías permite a las organizaciones mejorar la administración general del entorno de la empresa. El avance de la Sociedad del Conocimiento, logros alcanzados así como el progreso de la tecnología obligan hoy a plantearse nuevos retos.

A.2.2 Exposición del Problema

➤ Realidad Problemática

- Deficiencia en el control de préstamos.
- Demoras en el proceso registro de material.
- Perdida de tiempo para la búsqueda de fichas bibliográficas.
- No existe disponibilidad de consulta de libros en tiempo real.

➤ **Análisis del problema**

- El proceso de registro del material bibliográfico de la Biblioteca se realiza en hojas de cálculo, generando inconsistencia de datos y pérdida de información.
- El proceso de consultas se hace a través de fichas manuales antiguas y no actualizadas, ocasionando pérdida de tiempo al buscar un libro específico.
- Para el préstamo de material, se realiza de forma manual, pidiendo el carné del estudiante y anotando en una ficha el libro que está llevando, con la implementación de sistema esto será realizado automatizadamente.

➤ **Afecta a:**

- Universidad Nacional de Ucayali.
- Alumnos.
- Docentes.
- Personal biblioteca.

➤ **Impacto**

- Imagen institucional.
- Proceso lento y mal definido que dificultan el control de la biblioteca.
- Carga de trabajo al personal encargado de la biblioteca.
- Ocasiona pérdida de tiempo al alumno al consultar los libros que necesita.
- Demora en los requerimientos de información.

➤ **Solución exitosa será:**

- Mejorar el registro del material bibliográfico, consulta y préstamo de los materiales bibliográficos, presentándola vía Web.
- Contar con información relevante y oportuna para la eficiente toma de decisiones y brindar un mejor servicio a s usuarios.

A.3 Descripción de Stakeholder y usuarios

A.3.1 Mercado demográfico

La Universidad Nacional de Ucayali actualmente cuenta con más de 3500 personas entre estudiantes y docentes, los cuales no tienen un eficiente Sistema de Biblioteca, generando demoras al momento de registro y préstamo del material existente, esto ocasiona la perdida de interés y motivación por parte del estudiantado.

El Sistema Web con Tecnología J2EE para la Administración en la Biblioteca de la Universidad Nacional de Ucayali será ejecutado y usado en cualquier parte del mundo que tenga acceso a Internet, el sistema estará ubicado en servidor central de la UNU.

A.3.2 Sumario de Stakeholder

Cuadro 13: Sumario de Stakeholders

NOMBRE	REPRESENTANTE	ROL
Jefe de Biblioteca	La Biblioteca de la Universidad Nacional de Ucayali.	Evalúa y monitorea los procesos de la biblioteca.
Jefe de Almacén	El Área de Almacén de la Universidad Nacional de Ucayali.	Monitorea los inventarios de la institución.

Encargado de Proceso Técnico	Realiza la clasificación del material, según estándares internacionales.	Representa así mismo.
------------------------------	--	-----------------------

A.3.3 Sumario de Usuarios

Cuadro 14: Sumario de Usuarios

NOMBRE	DESCRIPCIÓN	STAKEHOLDERS
Secretaria de Biblioteca	Realiza el registro del material bibliográfico.	Representa así mismo.
Encargado de Distribución	Realiza los préstamos de los libros existentes.	Representa así mismo.
Alumno	Consulta la disponibilidad de libros y realiza reservas.	Representa así mismo.
Docentes	Consulta la disponibilidad de libros y realiza reservas.	Representa así mismo.

A.3.4 Ambiente de Usuarios

Secretaria de Biblioteca: Tendrá acceso al sistema para registrar los materiales bibliográficos.

Encargado de Distribución: Accederá al sistema para realizar los prestamos del material.

Alumno: Podrá acceder desde cualquier lugar para realizar las consultas de libros y verificar si puede realizar un préstamo o no.

Docente: Podrá realizar préstamos previa verificación del sistema.

A.3.5 Necesidades de los Usuarios

Cuadro 15: Necesidades de Usuarios

NECESIDAD	PRIORIDAD	CONCERNIENTE	SOLUCION ACTUAL	SOLUCION PROPUESTA
Registrar Material	Alta	Tiempo de registro lento.	Se realiza a través de una hoja de	El sistema automatizará en su

Bibliográfico			calculo	totalidad
Realizar Prestamos	Alta	Tiempo de respuesta lento e ineficiente	El sistema es totalmente manual	El sistema Automatizará este proceso.
Realizar reservas de libros	Alta	Tiempo de respuesta y carga del trabajo	No se pude realizar actualmente	El sistema Permitirá que este proceso sea rápido y seguro
Consultar libros	Alta	Tiempo de respuesta lento	Se realiza a través de fichas manuales	Implementar un módulo para que se pueda consultar los libros desde cualquier lugar
Consultar los usuarios que hicieron préstamo de material bibliográfico	Alta	Tiempo de respuesta	Se realiza de manera manual anotando en un libro de préstamo.	El sistema Permitirá que este proceso sea rápido y seguro
Consultar la asistencia diaria y mensual de los usuarios a Biblioteca	Alta	Tiempo de respuesta	Se realiza de manera manual anotando en un libro de asistencia	El sistema Permitirá que este proceso sea rápido y seguro
Consultar Materiales leídos	Alta	Tiempo de respuesta	No se pude realizar actualmente	El sistema Permitirá que este proceso sea rápido y seguro

A.3.6 Alternativas

El desarrollo del Proyecto de implementación "El Sistema Web con Tecnología J2EE para la Administración en la Biblioteca de la Universidad Nacional de Ucayali" presenta las siguientes alternativas:

- ✓ Formar un equipo de trabajo dentro de la Escuela Profesional de Ingeniería de Sistemas de la UNU para el desarrollo del Sistema.
- ✓ Solicitar los servicios de una consultoría para que realice el proyecto de implementación de un Sistema de Biblioteca.
- ✓ Solicitar los servicios de una empresa desarrolladora de software para que realice el Sistema de Información.
- ✓ Solicitar Tesitas o estudiantes interesados en desarrollar el proyecto de implementación del Sistema Web con Tecnología J2EE para la Administración en la Biblioteca de la Universidad Nacional de Ucayali.

A.4 Objetivos de Modelamiento del Negocio

Proceso de Gestión de Material Bibliográfico

Registrar material bibliográfico

Consultar material bibliográfico

Registrar autor

Registrar editorial

Registrar materia

Proceso de Servicio de Biblioteca

Registrar préstamo.

Validar Préstamo.

Registrar devolución

Registrar Usuarios

Registrar Lecturas.

Registrar Asistencia

Proceso de Administración de Usuarios

Registrar Usuarios

Dar de Baja y Alta de Usuarios a Biblioteca.

A.5 Rangos de Calidad

Disponibilidad: El Sistema Web con Tecnología J2EE para la Administración en la Biblioteca deberá estar disponible las 24 horas del día en un Servidor Web para la atención de los usuarios.

Uso: El Sistema Web con Tecnología J2EE para la Administración en la Biblioteca, deberá ser fácil de manejar con interfaces sencillas y entendibles para el usuario.

A.6 Panorama del Producto

Perspectiva: El Sistema Web con Tecnología J2EE para la Administración en la Biblioteca se implementará en el Servidor de la Dirección General de Sistemas e Informática de la Universidad Nacional de Ucayali – Pucallpa. Cada usuario tendrá acceso al sistema mediante una interfaz que les permitirá acceder a la información que requiera en el momento necesario, conectándose con su nombre de usuario y contraseña.

Restricciones: El Sistema Web con Tecnología J2EE para la Administración en la Biblioteca no contará con:

- El sistema no incluye control de almacén de equipos de biblioteca.
- El sistema no incluye registro de datos para dar baja a los libros, solo cambiamos el estado.

- El sistema no contempla la exoneración de multas por huelga, paros u otras interrupciones para la contabilización de la multa
- El sistema no incluye la clasificación automática de los libros.

A.7 Requerimientos

A.7.1 Funcionales

- ✓ Registrar Materiales Bibliográficos.
- ✓ Consultar Materiales Bibliográficos.
- ✓ Registrar Préstamos.
- ✓ Registrar Lecturas
- ✓ Registrar Asistencias
- ✓ Generar multas.
- ✓ Reportar Lista de deudores
- ✓ Registrar Clasificación.

A.7.2 No Funcionales

- ✓ El sistema se desarrollará utilizando herramientas Web: como el lenguaje de programación Java, las tecnologías JSP y Java Script; el manejador de Base de Datos SQL Server 2000 y un Servidor Web TomCat.
- ✓ Definir políticas de seguridad.
- ✓ Registrar niveles de seguridad.
- ✓ El tiempo de respuesta por consulta se realice en tiempo real.
- ✓ Contar con máquina para el Servidor, con requerimientos mínimos como Tecnología Pentium y una estructura de red.
- ✓ Instalar el Sistema operativo Windows Server 2003 Enterprise Edition

- ✓ Características del servidor: Servidor PIV 3.6 Ghz HD 180 GB.
Ram 2 Gb.
- ✓ Contar con una Impresora de Inyección de Tinta o Láser que actuará como dispositivo de salida para las terminales en la impresión de reportes del sistema y consultas requeridas.

B. Plan de Desarrollo del Software

B.1. Introducción

B.1.1 Propósito

El objetivo de este Plan de Desarrollo de Software es definir las actividades realizadas durante el desarrollo de las fases e iteraciones requeridas para llevar a cabo el Sistema Propuesto.

B.1.2 Alcance

Este Plan de Desarrollo de Software describe el plan global a ser usado por los encargados del proyecto para desarrollar la implementación de un *Sistema Web con Tecnología J2EE para la Administración en la Biblioteca*, se describirán los detalles de los procesamientos individuales del plan. Los planes que se dan en éste documento se basan en los requisitos del producto como está especificado en el documento visión.

B.1.3 Referencias

Las referencias aplicables son: La visión para la implementación del Sistema Web con Tecnología J2EE para la Administración en la Biblioteca.

B.1.4 Apreciación Global

Este Plan de Desarrollo de Software contiene la información siguiente:

- **Proyecto de Apreciación Global:** Proporciona la descripción del propósito del proyecto, alcance y objetivos. También determina el entregable que se espera en el proyecto en determinados periodos.
- **El Proceso de Dirección:** Explica el costo estimado y lo fija, define las fases mayores e hitos para el proyecto, y describe el modo de supervisión para el proyecto.
- **Los Planes del Proceso Técnicos:** Proporciona un panorama global del proceso de desarrollo de Software, incluso los métodos, herramientas y técnicas para ser seguido.

B.2. La Apreciación Global del Proyecto

B.2.1 Propósito del Proyecto, Alcance y Objetivos

El propósito, alcance y objetivo de este Plan de Desarrollo de Software es definir las actividades realizadas durante el desarrollo de las fases e iteraciones requeridas para implementar el Sistema a realizar.

B.2.2 Entregables del Proyecto.

Los entregables siguientes se desarrollaran durante el proyecto.

Cuadro 16: Entregables del proyecto.

FASES	WORKFLOWS	ARTEFACTOS
INICIAL	MODELO DEL NEGOCIO	<ul style="list-style-type: none"> • Documentos de visión. • Plan de desarrollo de Software. • Modelo de Use Case del Negocio. • Modelo de Dominio del Problema.
ELABORACIÓN	REQUERIMIENTOS	<ul style="list-style-type: none"> • Modelo de Use Case. • Especificación de los Use Case.
	ANÁLISIS Y DISEÑO	<ul style="list-style-type: none"> • Diagrama de Colaboración. • Prototipo Arquitectónico. • Diagrama de Clases. • Diagrama de Secuencia. • Paquetes del Diseño.
CONSTRUCCIÓN	ANALISIS DE DISEÑO	<ul style="list-style-type: none"> • Diseño de Base de Datos. • Prototipo Inicial.
	IMPLEMENTACIÓN	<ul style="list-style-type: none"> • Diagrama de Componentes. • Diagrama de Despliegue. • Prototipo del Software Final
TRANSICIÓN	PRUEBA	<ul style="list-style-type: none"> • Prueba por Use Case

B.2.3 Evolución del Plan de desarrollo de Software.

El Plan de Desarrollo de Software se revisará anterior a la salida de cada proceso de iteración.

B.3. La Organización del Proyecto

B.3.1 Estructura Orgánica

El equipo de trabajo comprende de dos tesisistas Bachiller en Ingeniería de Sistemas de la Universidad Nacional de Ucayali, un Asesor Ingeniero Industrial y un Co-Asesor Ingeniero de Sistemas, para el desarrollo del proyecto.

B.3.2 Interfaces Externas

Los tesisistas del proyecto también actuarán recíprocamente con otro stakeholder para solicitar las entradas y revisión de los artefactos.

B.3.3 Papeles y Responsabilidades

La tabla siguiente identifica las unidades orgánicas que serán responsables para cada uno de los flujos de trabajo, y el detalle de los procesos de apoyo.

Cuadro 17: Papeles y Responsabilidades

PAPEL	RESPONSABILIDAD
Bach. Javier Luna Velásquez Bach. Yumi Tominaga García Tesisistas	<ul style="list-style-type: none"> • Responsabilidad del manejo del flujo de producto de dirección del proyecto global. • Responsable principal de manejar el modelado comercial y el flujo de trabajo de los requisitos, proporciona el apoyo y entradas de Workflow de Dirección de Proyecto. • Responsable principal para el análisis y diseño, aplicación, configuración, capacitación y flujos de trabajo de ambiente. • Proporciona el apoyo al Workflow de dirección de proyección. • Es el responsable para manejar la prueba, y Workflows del

	Despliegue.
Asesores del Proyecto	<ul style="list-style-type: none"> Realiza la asesoría, seguimiento Y correcciones de entregables de proyectos.

B.4. El Proceso de Dirección

B.4.1 Estimación del Proyecto

Las estimaciones del proyecto son basadas en el Estudio de Factibilidad aplicado al proyecto. El tiempo y el esfuerzo estimado en este informe es la base del presupuesto del proyecto y horario.

B.4.2 Plan de Proyecto

a) Plan de la Fase:

El Sistema Web con Tecnología J2EE para la Administración en la Biblioteca será desarrollado usando un acercamiento escalonado a la cual utilizará cuatro fases de interacción. Las fases y el horario relativo se muestran en la tabla siguiente:

Cuadro 18: Plan de proceso de desarrollo de acuerdo a fases

FASE	EMPIEZA	TERMINA
Fase de Inicio (10%)	01/10/07	30/10/07
Fase de la Elaboración (25%)	30/10/07	20/12/07
Fase de la Construcción (60%)	20/12/07	23/02/08
Fase de la Transición (5%)	23/02/08	08/03/08

Cuadro 19: Fases del proyecto e hitos principales

FASE	DESCRIPCIÓN	HITO
INICIO	En esta Etapa se define el modelo del negocio, los requerimientos del producto, se elabora el plan de desarrollo de Software.	Determina la Factibilidad del proyecto desde un punto de vista del negocio. Se definen los requerimientos, características, claves y principales restricciones. Estima los recursos (Tiempo, costos del ambiente de desarrollo)
ELABORACIÓN	La fase de Elaboración analizará los requisitos y se desarrollará el prototipo arquitectónico. En la realización de la fase de la Elaboración todos los Use Case seleccionados para una primera versión 1.0 habrán completado el análisis y el plan. Además se habrán analizado los Use Case de alto riesgo que para una Versión 2.0 ya se habrán diseñado. El prototipo arquitectónico probará la viabilidad y actuación de la arquitectura que se requiere para Versión 1.0.	El hito del Prototipo Arquitectónico marca el término de la Fase de la Elaboración.
CONSTRUCCIÓN	Durante la Fase de la Construcción se analizan los Use Case restantes y se diseñarán estos. La	Actualización con todos los elementos necesarios para dar soporte a la implantación de la

	versión beta para la Versión 1.0 se desarrollará y se distribuirá para la evaluación.	persistencia (ejemplo tablas, index, mapeo de modelo de datos orientado a objetos a relacionar).
TRANSICIÓN	En esta fase se empaquetará, distribuirá e instalará el producto.	Completa en la concordancia con los requerimientos del producto definidas en el documento de Visión del Negocio. El producto final debe estar disponible para los usuarios.

Cada fase es dividida en las interacciones del desarrollo. Se espera que la duración del proyecto sea de 4 meses, el cálculo de esta estimación de tiempo se muestra en el Anexo 1.

b) Horario del Proyecto

El horario del proyecto que contiene el nombre de las Labores, las fechas de inicio y fin se muestran a continuación.

Cuadro 20: Tareas del proyecto

FASE	EMPIEZA	TERMINA
Modelamiento del Negocio	01/10/07	30/10/07
Requerimientos	30/10/07	15/11/07
Análisis y Diseño	15/11/07	20/12/07
Implementación	20/12/07	23/02/08
Prueba	23/02/08	29/02/08
Despliegue	01/03/08	08/03/08

B.5 Recursos para el Proyecto

B.5.1 Plan de Adquisición de Recursos

La UNU ha proyectado asignar a personal especializado para lograr el objetivo.

B.5.2 Entrenamiento que se planean

Se entrenará al equipo del proyecto en las siguientes habilidades, al comienzo de las actividades del plan:

- ✓ Análisis y Diseño Orientado a Objeto
- ✓ Proceso Unificado Rational
- ✓ NetBeans 5.5
- ✓ SQL Server 2000

B.6 Presupuesto

El siguiente presupuesto se basa en estimaciones iniciales (Ver Anexo 2):

Cuadro 21: Presupuesto del proyecto.

Sistema Web con Tecnología J2EE para la Administración de la Biblioteca		
Actividades	Esfuerzo	Costo S/.
Desarrollo del Sistema de Biblioteca	1398.8 horas/ hombres * Use Case	8000.00
	Total Trabajo del personal	8000.00
Gastos de Aprovisionamiento		
	Viáticos y Asignaciones	1080.00
	Telefonía Celular	100.00
	Materiales de Escritorio	200.00
	Equipos y Bienes Duraderos	7500.00
	Total Gastos de Aprovisionamiento:	8900.00
	Total del Presupuesto	16900.00

B.7. Entorno de Trabajo

B.7.1 Elección de Equipos y Accesorios de la Red LAN

B.7.1.1 Elección del Servidor

Para elegir el tipo de servidor se ha tenido en consideración los equipos existentes dentro de la Oficina de Computo e Informática de la Universidad Nacional de Ucayali, el Sistema Operativo instalado es Microsoft Windows Server 2003 Enterprise Edition con un servidor Apache Tomcat 1.5.

Cuadro 22: Características servidor de Base de Datos

CARACTERÍSTICAS	OPCIONES
Procesador	Intel Xeon IV 3.60 GHz.
Memoria	Caché Interna 1GB
Memoria	RAM 2GB
Bus entrada/Salida	PCI/EISA
Puerto	1 Paralelo, 2 Seriales
Puerto	3 USB
Disco Duro	5 discos de 180 GB SCSI
Unidad CD-ROM	Lectora 48x
Unidad de Diskettes	3 ½ 1.44MB
Tarjeta de RED	Dual Gigabit Ethernet 10/100 Base T
Tarjeta Video	ATI RADEON 7000-M 16MB SDRAM
Monitor :	LG 17"
Mouse:	Sleek 2 botones
Teclado	Genius PS/2

Cuadro 23: Características de Servidor Web

CARACTERÍSTICAS	OPCIONES
Procesador	Intel Xeon IV 3.40 GHz.
Memoria	Caché Interna 512MB
Memoria	RAM 1GB
Bus entrada/Salida	PCI/EISA
Puerto	1 Paralelo, 2 Seriales
Puerto	4 USB
Disco Duro	1 disco de 73.4 GB SCSI
Unidad CD-ROM	Lectora 48x
Unidad de Diskettes	3 ½ 1.44MB
Tarjeta de RED	Gigabit Ethernet 10/100/1000 Base T
Tarjeta Video	PCI 7000-M 16MB SDRAM
Monitor	LG 15"
Mouse	Genius 2 botones
Teclado	Genius PS/2

B.7.1.2 Elección de Estaciones de Trabajo.

Actualmente la Biblioteca cuenta con 49 Pentium IV 2.8 Ghz, las cuales estarán conectadas al servidor principal ubicado en el departamento de informática obteniendo así una infraestructura de red adecuada que soporte los ingresos de datos que realizarán los usuarios.

Cuadro 24: Características de las Estaciones de trabajo

CARACTERÍSTICAS	OPCIONES
Procesador	Intel Pentium IV 2.8 Ghz
Memoria Caché interna	512 Kb
Memoria RAM	256 Mb
Disco Duro	40 GB

Tarjeta de RED.	Dlink Ethernet 10/100
Teclado y Mouse	Genius
Monitor	15" Digital LG

B.7.1.3 Elección de las tarjetas de RED para las Estaciones de Trabajo

La elección de las tarjetas de red han sido tomadas en consideración a la tecnología de red a usar (Ethernet) y al tipo de estaciones de trabajo.

Cuadro 25: Características de la tarjeta de red

CARACTERÍSTICAS	OPCIONES
Marca	D-Link
Tecnología	Ethernet
Tipo de Bus (Plug and Play)	ISA/PCI
Full duplex	Si
Conectores permitidos	RJ-45
Leds Indicadores	Power, TX/Rx
Soporte Boot Room	No

B.7.1.4 Elección del Switch (Concentrador)

La elección del Switch ha sido tomada en consideración a la tecnología de red (Ethernet) y a la cantidad de estaciones de trabajo.

Cuadro 26: Características concentradoras

CARACTERÍSTICAS	OPCIONES
Marca	DLink

Tecnología	Ethernet
Tipo de Switch	Activo
Numero de Puertos	24UTP/STP-RJ45)
Administración	SNMP, RMON
Soporte de otras tecnologías	Si
Fuente de Alimentación redundante	Si

B.7.1.5 Accesorios de Red.

Se ha tomado en consideración el estándar de instalaciones comerciales de red ANSI/EIA/TIA 568-A.

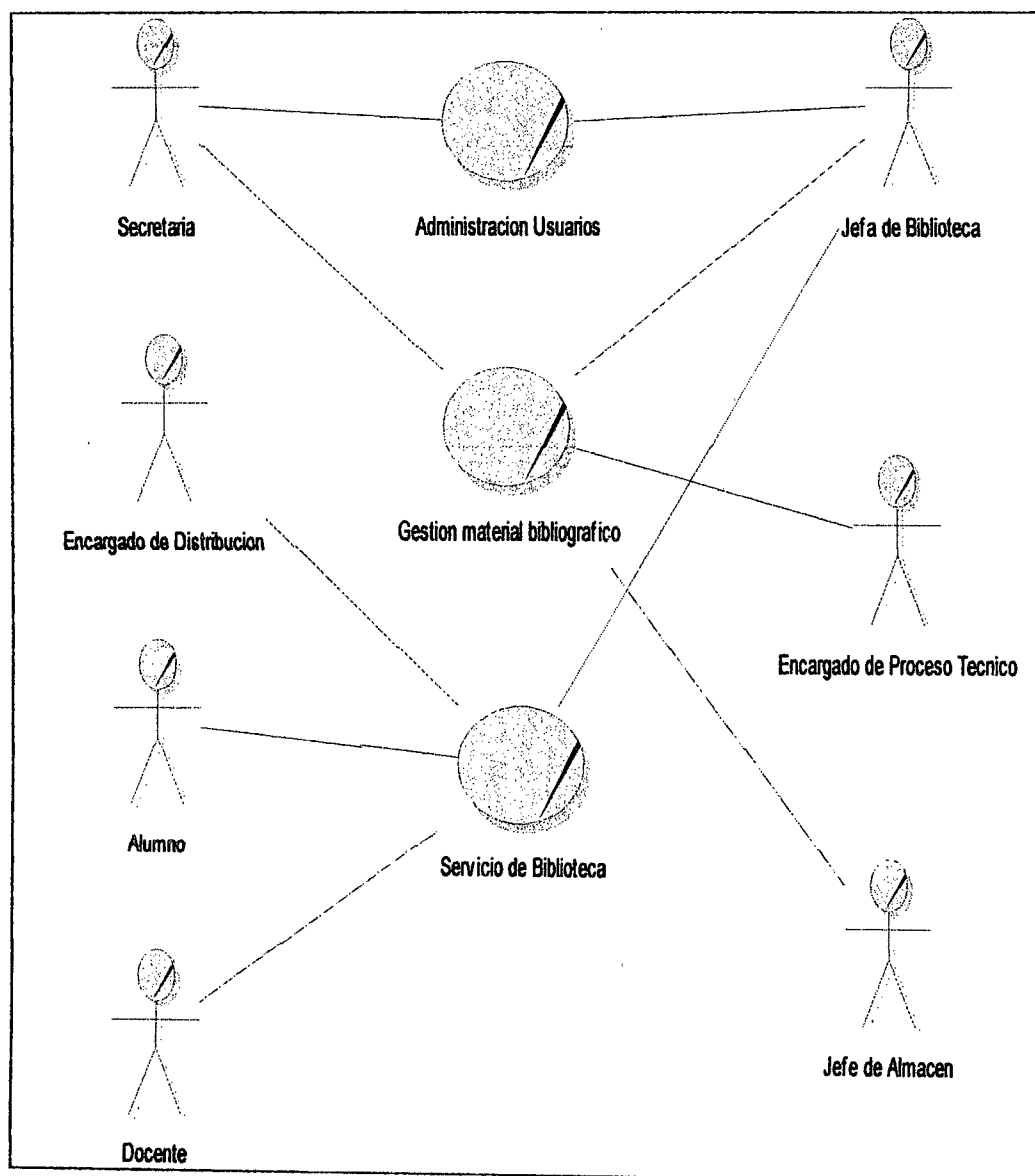
Tipos de Cable a Usar:

- Cable par trenzado (UTP – RJ-45 DE 100 W).
- El cable de par trenzado UTP, tiene 4 partes de hilos trenzados juntos a seis vueltas por pulgada para producir protección de interferencias eléctricas más impedancia consistente, o resistencia eléctrica.
- El Cable par trenzado UTP es barato, fácil de instalar y puede funcionar en red. En la actualidad es muy usado en redes locales Ethernet (UTP) de 8 hilos.
- Conectores RJ-45 Categorías 5, soportan 4 pares de cables UTP categoría 5.
- Caja toma de datos (Rosetas) RJ-45 Categoría 5.
- Roseta simple (1 jack) por cada estación de trabajo.
- Montaje con tornillo para facilitar la instalación.

B.8 Vistas de Use Case

B.8.1 Modelo use case del Negocio

Gráfico 11: Use Case del Negocio



B.8.1 Modelo de Objeto del Negocio

Gráfico 12: Modelo de Objetos Gestión Material Bibliográfico

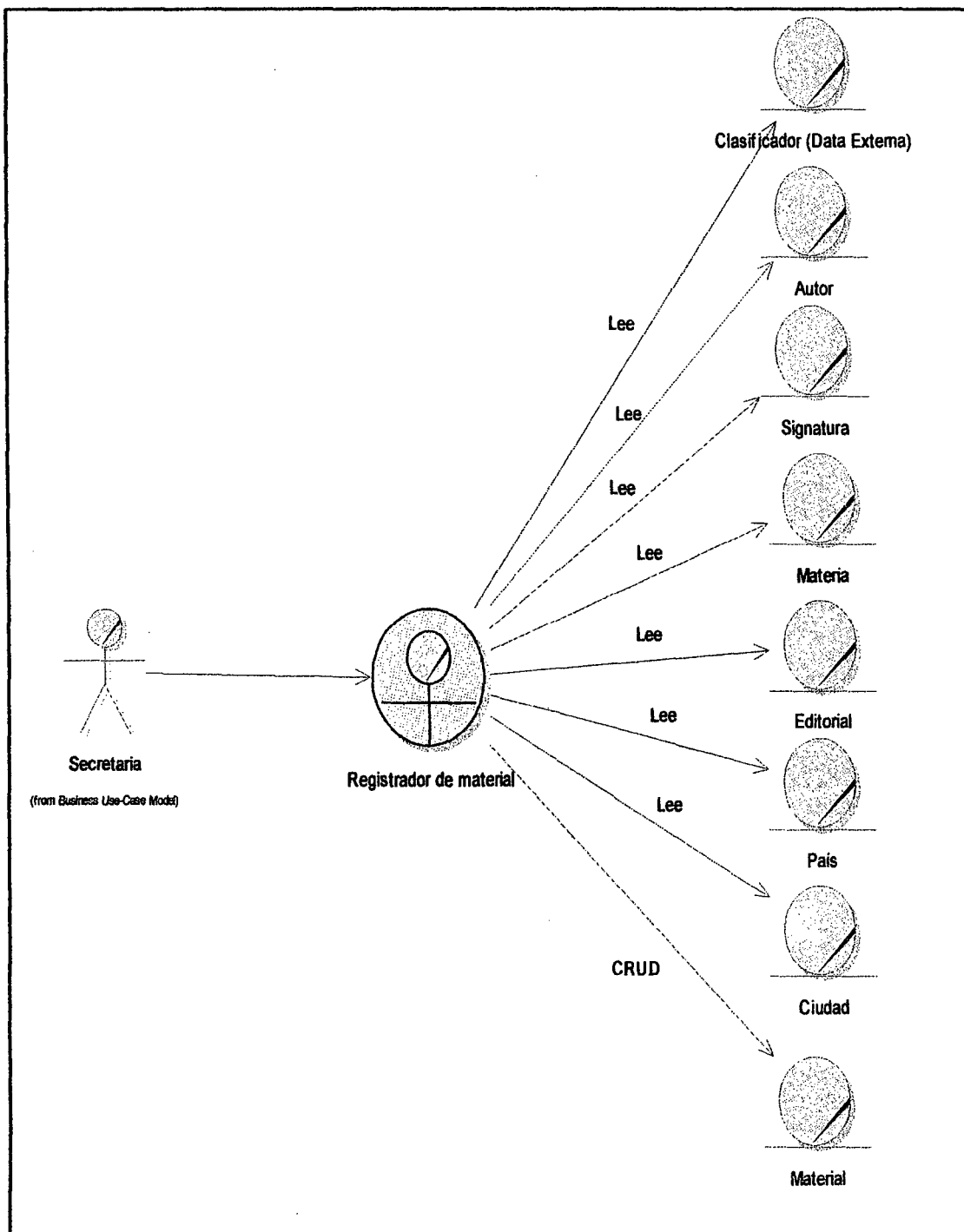


Gráfico 13: Modelo de Objetos Servicio biblioteca

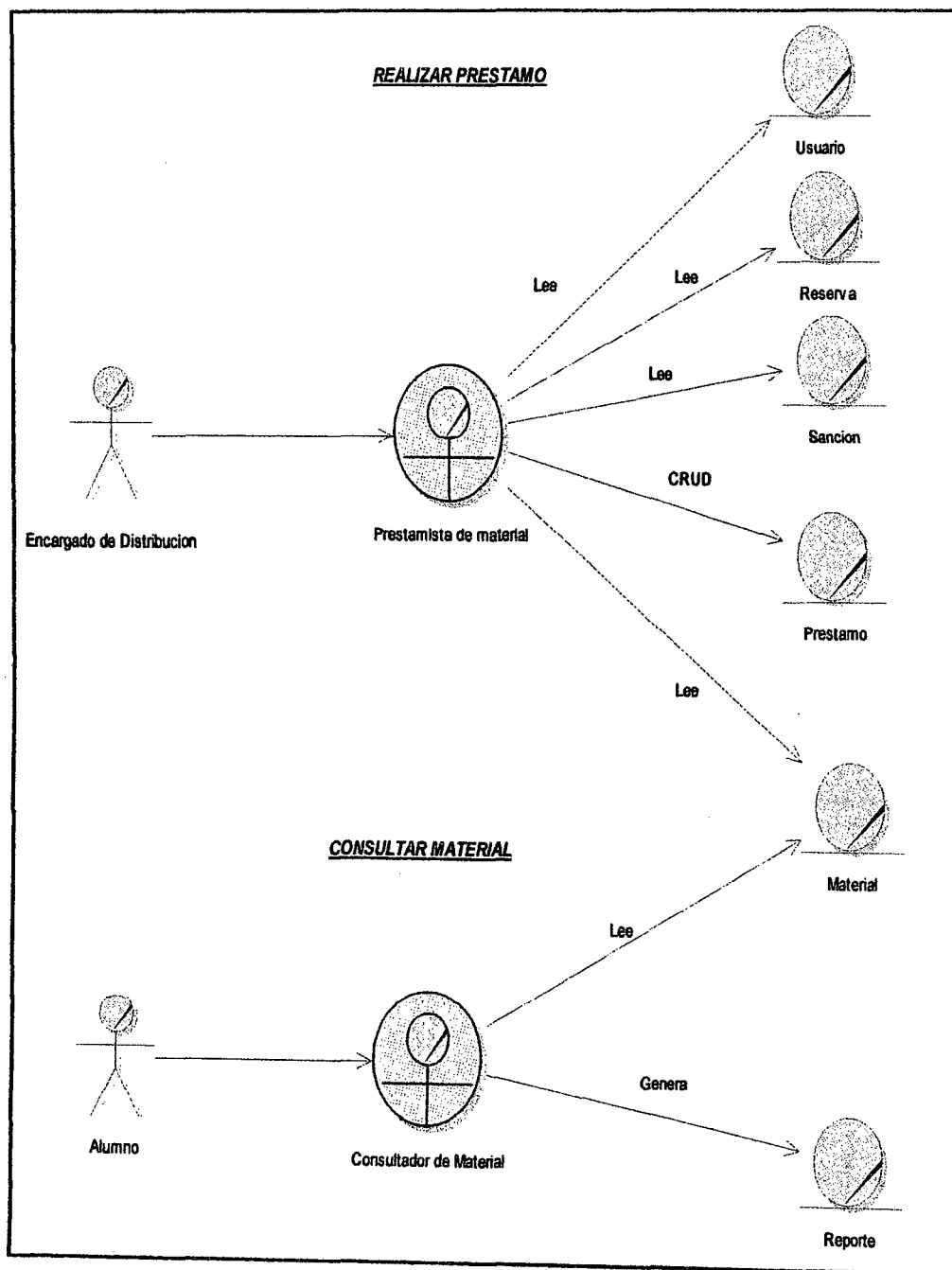
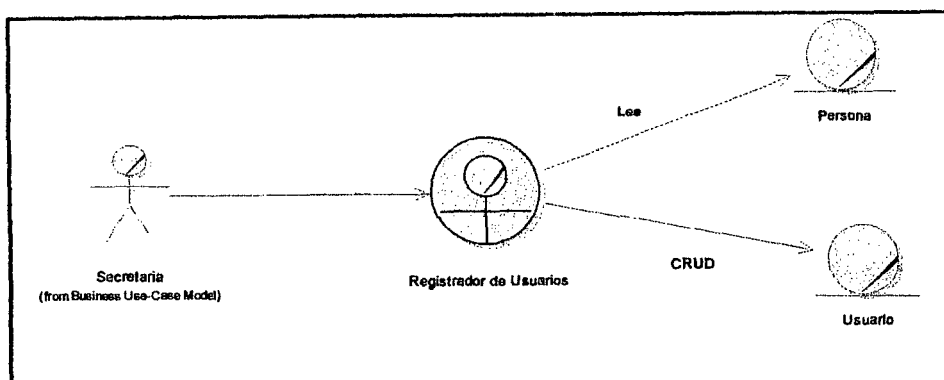
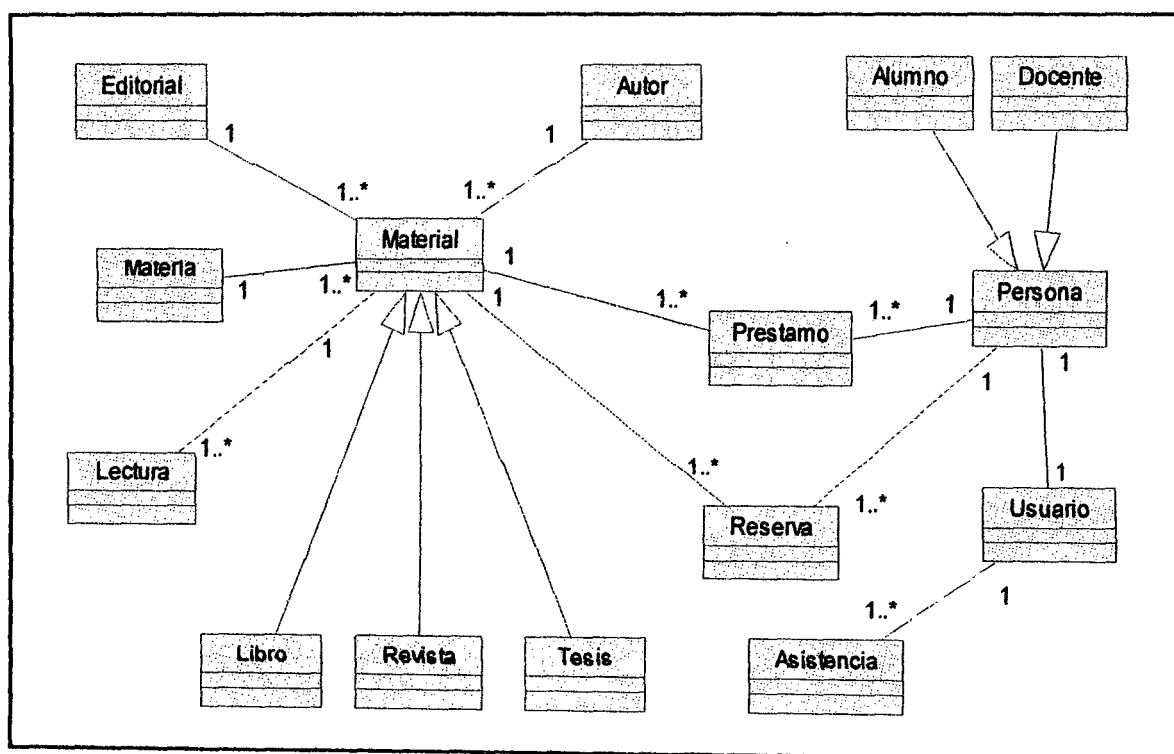


Gráfico 14: Modelo de Objetos Administración de Usuarios



B.8.2 Modelo de Dominio

Gráfico 15: Modelo Dominio



B.9 Descripción de Procesos del Negocio**Cuadro 27: Procesos del negocio**

ESTEREOTIPO	DESCRIPCIÓN
GESTION DE MATERIAL	Se registran los materiales nuevos, se clasifican según estándares internacionales de clasificación, registrar y mantener autor, editorial y materias según lo requiera.
SERVICIO DE BIBLIOTECA	Se realizan los prestamos del material tanto a los usuarios, generando multas y lista de deudores además permitirá al alumno consultar y reservar los materiales disponibles, se valida la reserva realizada externamente por los usuarios.
ADMINISTRACION DE USUARIOS	Se registran a los usuarios para que puedan acceder a la biblioteca, además se da la baja en caso de cometer alguna falta.

4.2. Fase Elaboración

4.2.1. Requerimientos

A) Modelo Use Case

Gráfico 16: Requerimientos Gestión Material

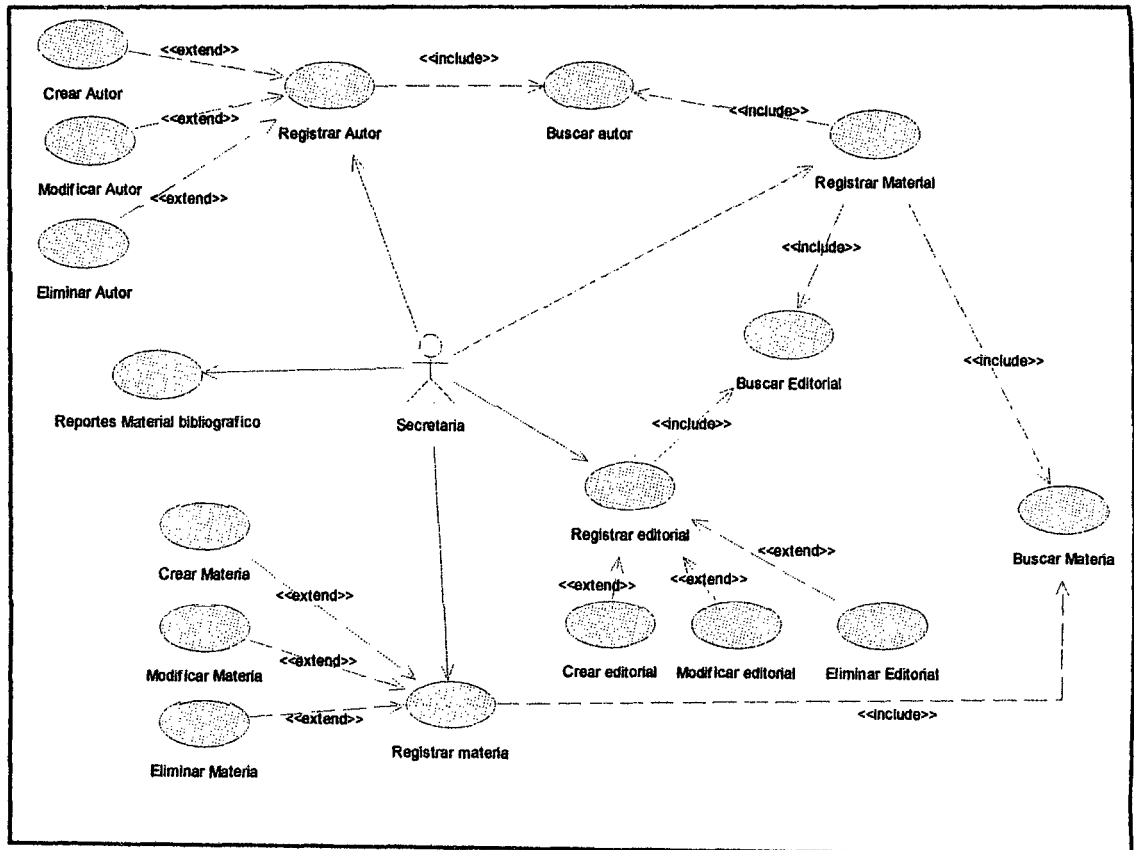


Gráfico 17: Requerimientos Servicio Biblioteca

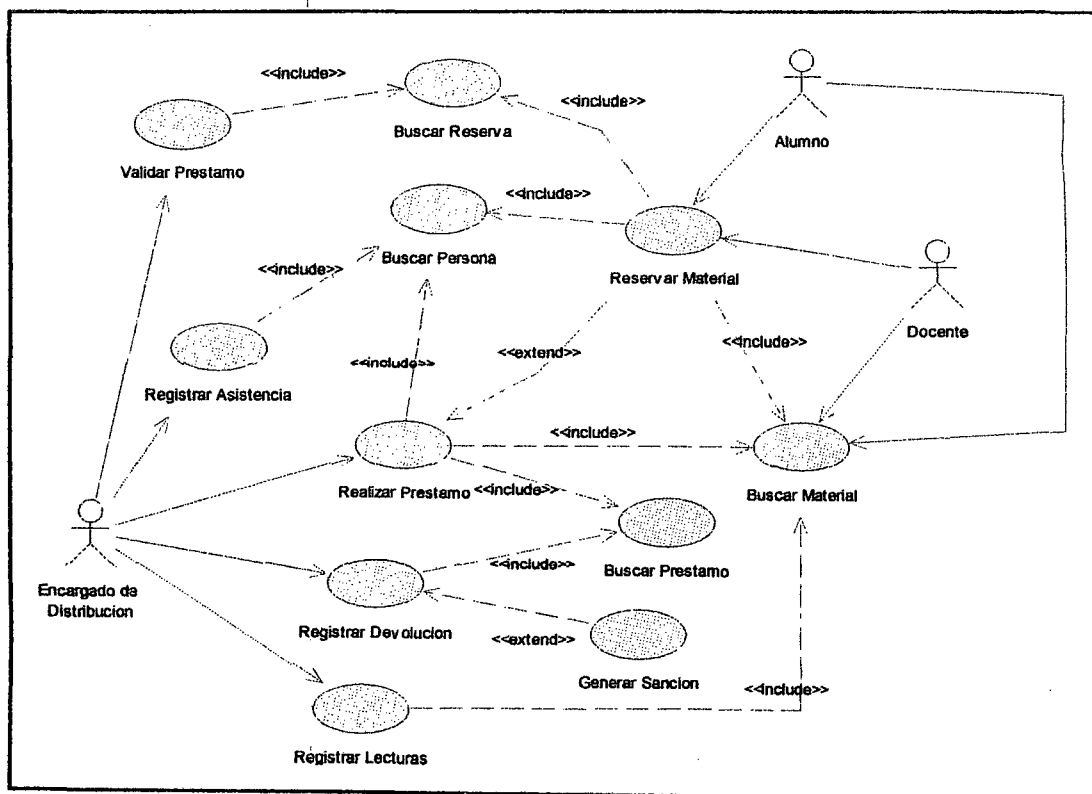
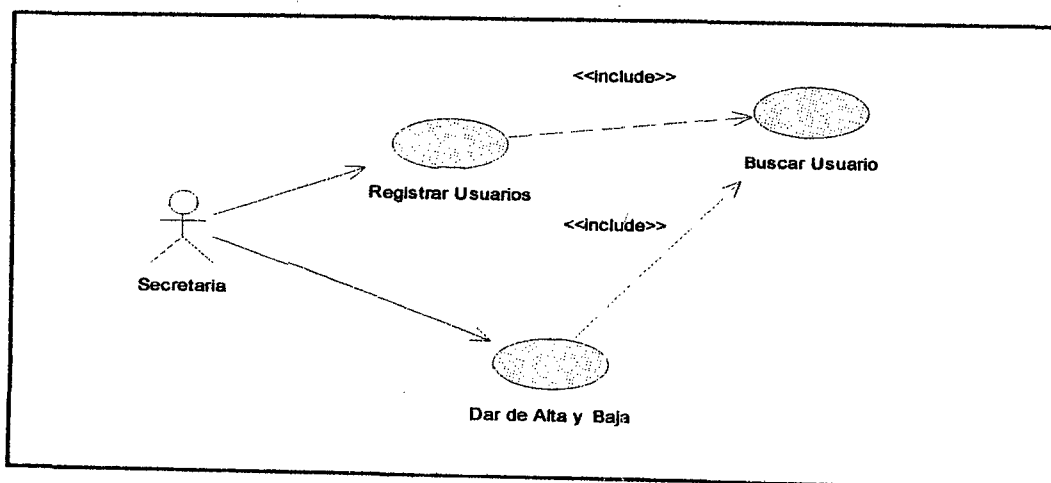


Gráfico 18: Requerimientos Servicio Biblioteca



B) Diagrama de Actividades

Gráfico 19: Actividades Gestión Material

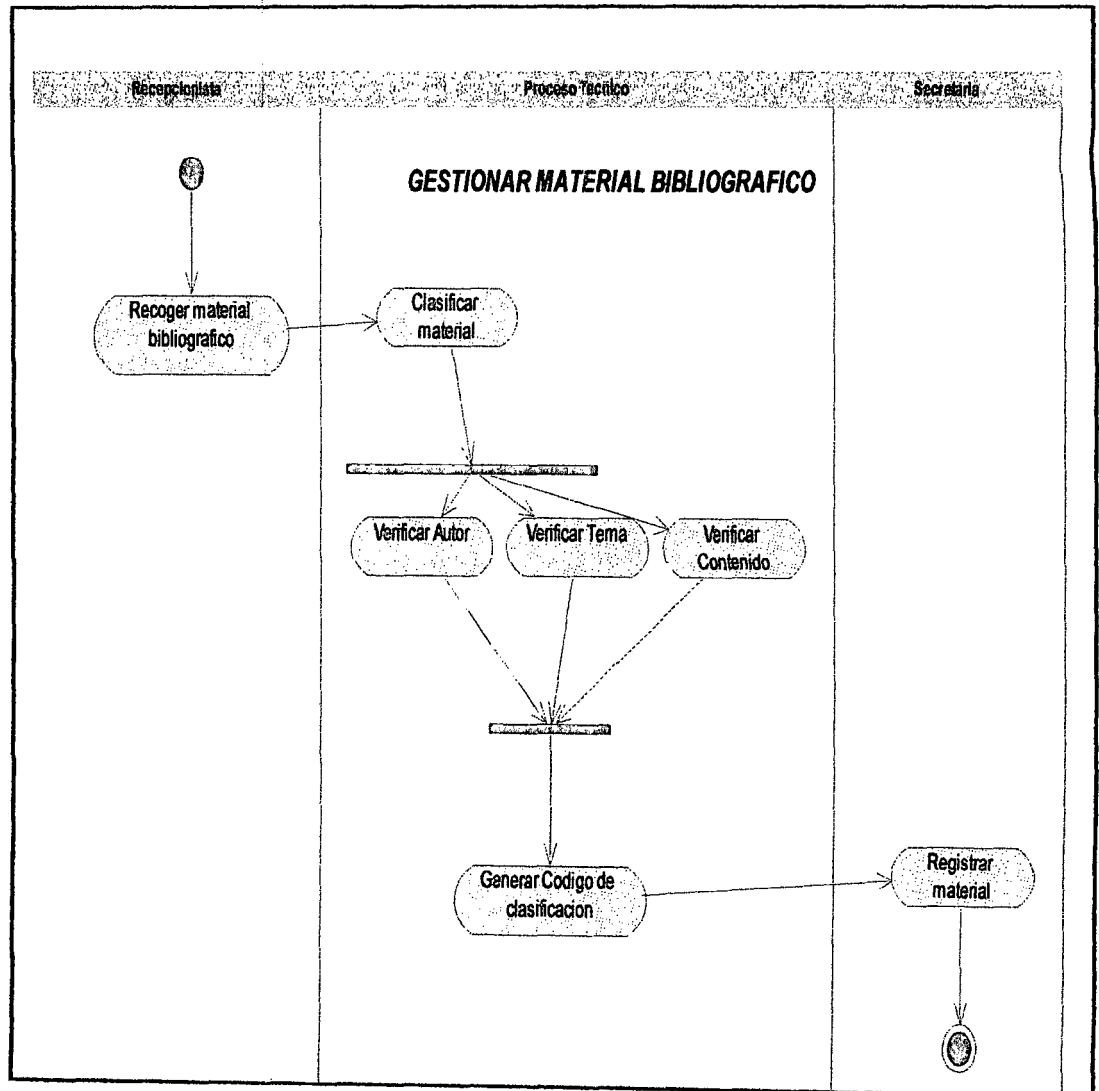


Gráfico 20: Actividades Servicio Biblioteca

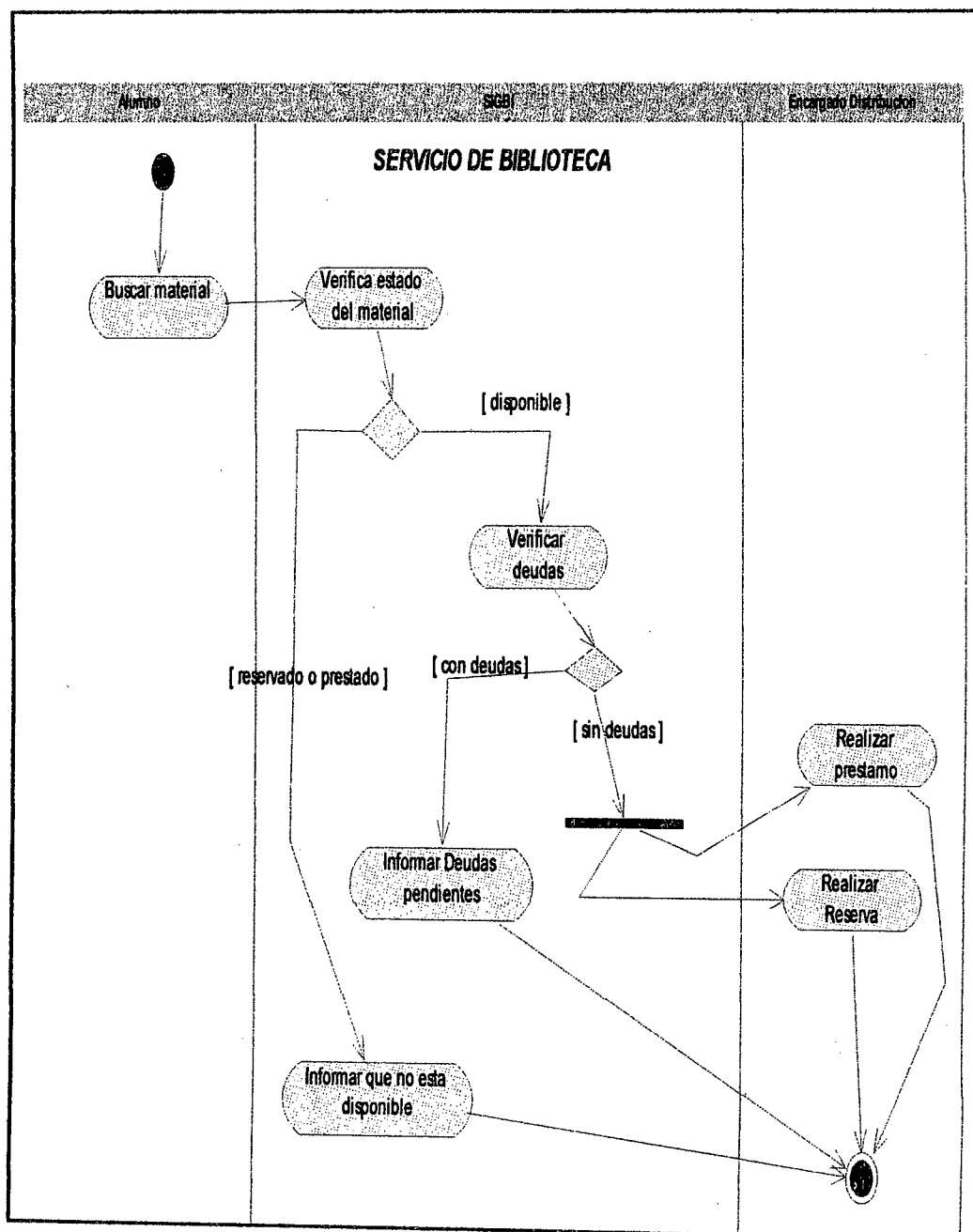
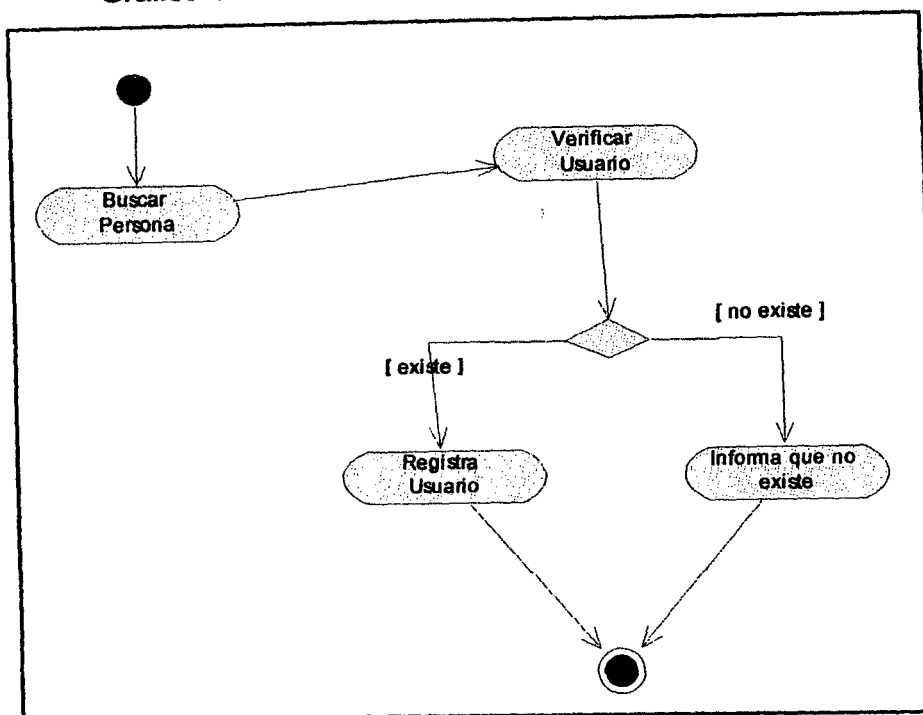


Grafico 21: Actividades Administración de Usuarios



C) Especificaciones de los Use-Case

C.1. Descripción del modelo Use-Case: Gestión de Material

Bibliográfico

Nombre del Use-Case: Registrar Material Bibliográfico	
Actor:	Secretaria
Definición :	Permite registrar el material bibliográfico que ingresa a la Biblioteca
Pasos:	
1.	Se extraen los datos pertinentes de cada material, incluyendo el código generado por Proceso Técnico
2.	Buscamos y seleccionamos el autor
3.	Buscamos y seleccionamos el editorial
4.	Buscamos y seleccionamos el material
5.	Se registra el material bibliográfico
6.	El sistema genera el código de barras del material

Nombre :	Registrar Autor
Actor:	Secretaria
Definición :	Permite registrar los autores de los materiales
Pasos:	
1.	Se busca si el autor solicitado no se encuentra en la base de datos
2.	Se ingresan los datos del autor
3.	Se registra el autor

Nombre :	Registrar Editorial
Actor:	Secretaria
Definición :	Permite registrar las editoriales de los materiales
Pasos:	
1.	Se busca si la editorial solicitada no se encuentra en la base de datos
2.	Se ingresan los datos de la editorial
3.	Se registra la editorial

Nombre :	Registrar Materia
Actor:	Secretaria
Definición :	Permite registrar la materia de los materiales
Pasos:	
1.	Se busca si la materia solicitada no se encuentra en la base de datos
2.	Se ingresan los datos de la materia
3.	Se registra la materia

Nombre :	Mantener Material Bibliográfico
Actor:	Secretaria
Definición :	Permite mantener material bibliográfico.
Pasos:	
1.	Se busca el material bibliográfico solicitado mostrándose una lista según los parámetros de búsqueda.
2.	Si selecciona en nuevo, redirecciona al formulario de registrar material bibliográfico.
3.	Si selecciona en modificar, se cambia la información y se guarda los cambios.
4.	Si selecciona en Eliminar, se elimina el registro seleccionado en el caso de no estar asignado a un material bibliográfico.

Nombre :	Mantener Autor
Actor:	Secretaria
Definición :	Permite mantener autor.
Pasos:	
1.	Se busca el autor solicitado mostrándose una lista según los parámetros de búsqueda.
2.	Si selecciona en nuevo, redirecciona al formulario de registrar autor.
3.	Si selecciona en modificar, se cambia la información y se guarda los cambios.
4.	Si selecciona en Eliminar, se elimina el registro seleccionado en el caso de no estar asignado a un material bibliográfico.

Nombre :	Mantener Editorial
Actor:	Secretaria
Definición :	Permite mantener editorial.
Pasos:	
1.	Se busca el editorial solicitado mostrándose una lista según los parámetros de búsqueda.
2.	Si selecciona en nuevo, redirecciona al formulario de registrar editorial.
3.	Si selecciona en modificar, se cambia la información y se guarda los cambios.
4.	Si selecciona en Eliminar, se elimina el registro seleccionado en el caso de no estar asignado a un material bibliográfico.

Nombre :	Mantener Materia
Actor:	Secretaria
Definición :	Permite mantener la materia
Pasos:	
1.	Se busca la materia solicitada mostrándose una lista según los parámetros de búsqueda.
2.	Si selecciona en nuevo, redirecciona al formulario de registrar de materia.
3.	Si selecciona en modificar, se cambia la información y se guarda los cambios.
4.	Si selecciona en Eliminar, se elimina el registro seleccionado en el caso de no estar asignado a un material bibliográfico.

C.2 Descripción del modelo Use-Case: Servicio de Biblioteca

Nombre :	Registrar Préstamo
Actor:	Dirección de Distribución
Definición :	Permite a los usuarios realizar prestamos de material bibliográfico
Pasos:	
1.	Los usuarios seleccionan el material que van a prestar.
2.	El sistema verifica la disponibilidad del material
3.	El sistema verifica las deudas del usuario
4.	El sistema verifica el número de préstamos
5.	Si no existe restricciones se registra el préstamo y se genera la papeleta de salida

Nombre :	Validar Préstamo
Actor:	Dirección de Distribución
Definición :	Permite validar las reservas de material bibliográfico
Pasos:	
1.	El usuario ingresa el código del quien se hizo la reserva
2.	El sistema verifica si realizó la reserva además de la disponibilidad y deudas del usuario
3.	Se cancela la reserva y se registra el préstamo del material.

Nombre	Registrar reservas
Actor:	Alumno
Definición :	Permite realizar reservas de material bibliográfico
Pasos:	
1.	El usuario selecciona el material que desea reservar
2.	El sistema verifica la disponibilidad del material y las deudas del usuario
3.	Se registra la reserva del material.

Nombre	Registrar devolución
Actor:	Dirección de Distribución
Definición :	Permite devolver un material prestado
Pasos:	
1.	Se busca el préstamo realizado
2.	El sistema registra la devolución del material, generando la deuda a pagar si se sobrepaso el tiempo límite

C.3 Descripción del modelo Use-Case: Administración de Usuarios

Nombre	Mantener usuarios
Actor:	Secretaria
Definición :	Permite crear, modificar y eliminar a los usuarios del sistema
Pasos:	
1.	Se ingresa código del usuario
2.	El sistema lo busca en caso de no estar como usuario del sistema se procede a llenar sus datos como usuario y contraseña procediendo luego a guardar.
3.	El sistema lo busca en caso de ser usuario del sistema y seleccione la opción modificar se procede a cambiar la información para luego guardar.
4.	El sistema lo busca en caso de ser usuario del sistema y seleccione la opción eliminar se procede a eliminar el registro.

Actor:	Secretaria
Definición :	Permite dar de baja y alta a los usuarios del sistema.
Pasos:	
1.	Se ingresa código del usuario
2.	El sistema lo busca y muestra la lista según los parámetros de búsqueda
3.	Al seleccionar el usuario, determina si le da baja o alta según sea el caso.

4.2.2. Análisis y Diseño

A) Diagrama de Colaboración

Gráfico 22: Colaboración Consultar material

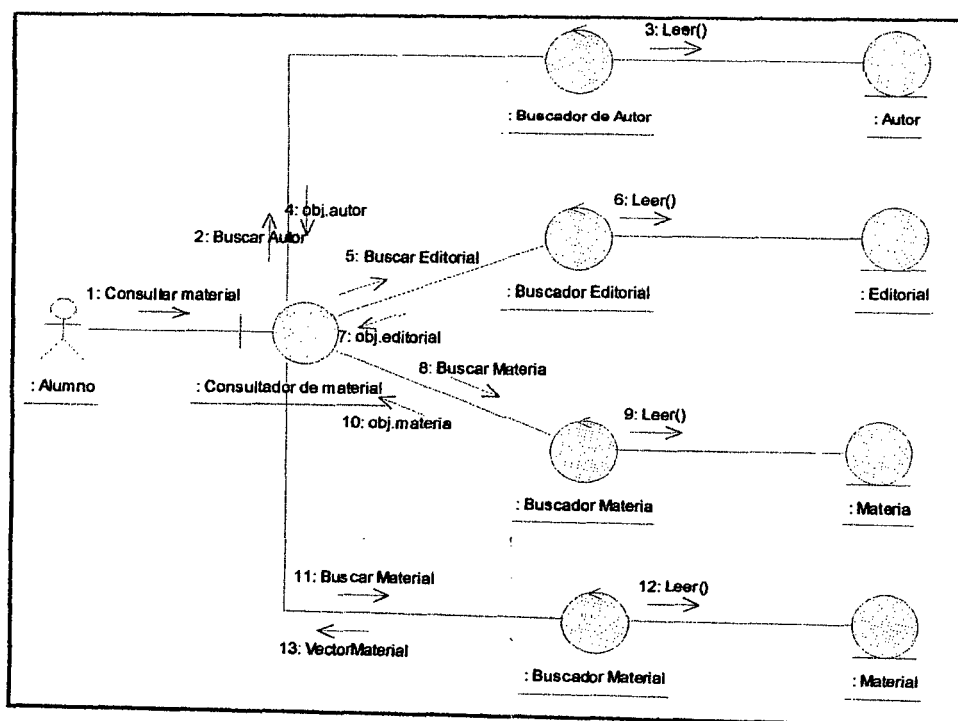


Gráfico 23: Colaboración Mantener autor

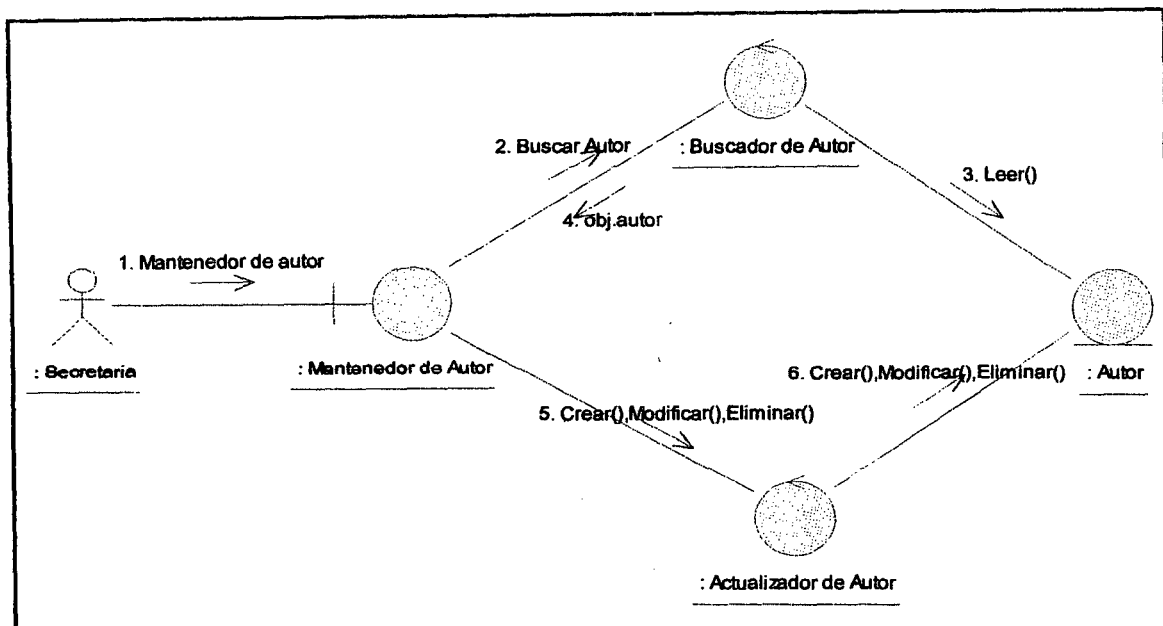


Gráfico 24: Colaboración Mantener editorial

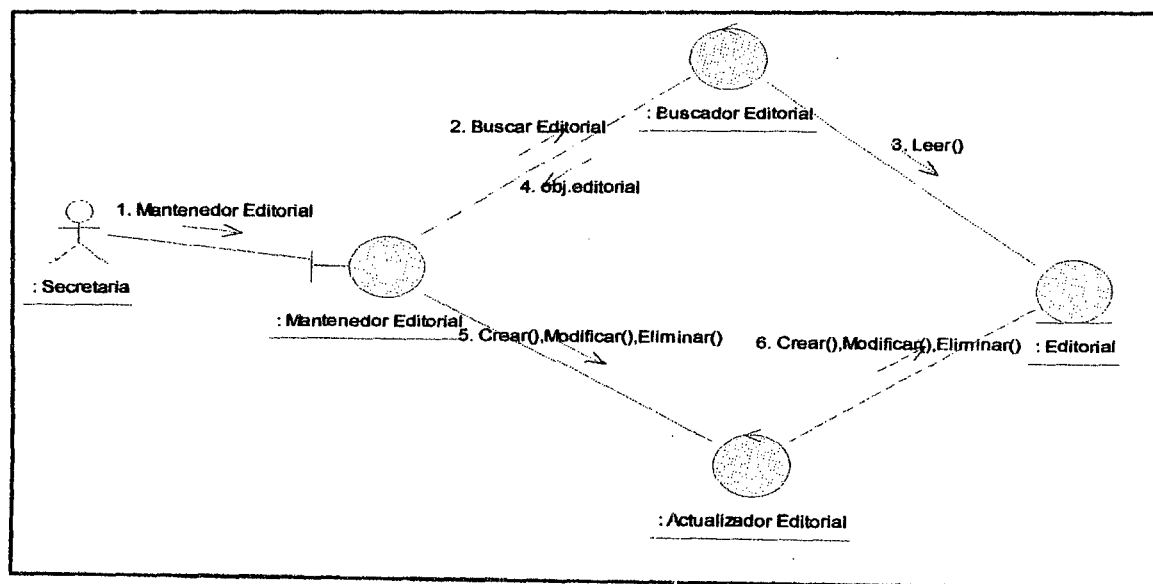


Gráfico 25: Colaboración Mantener materia

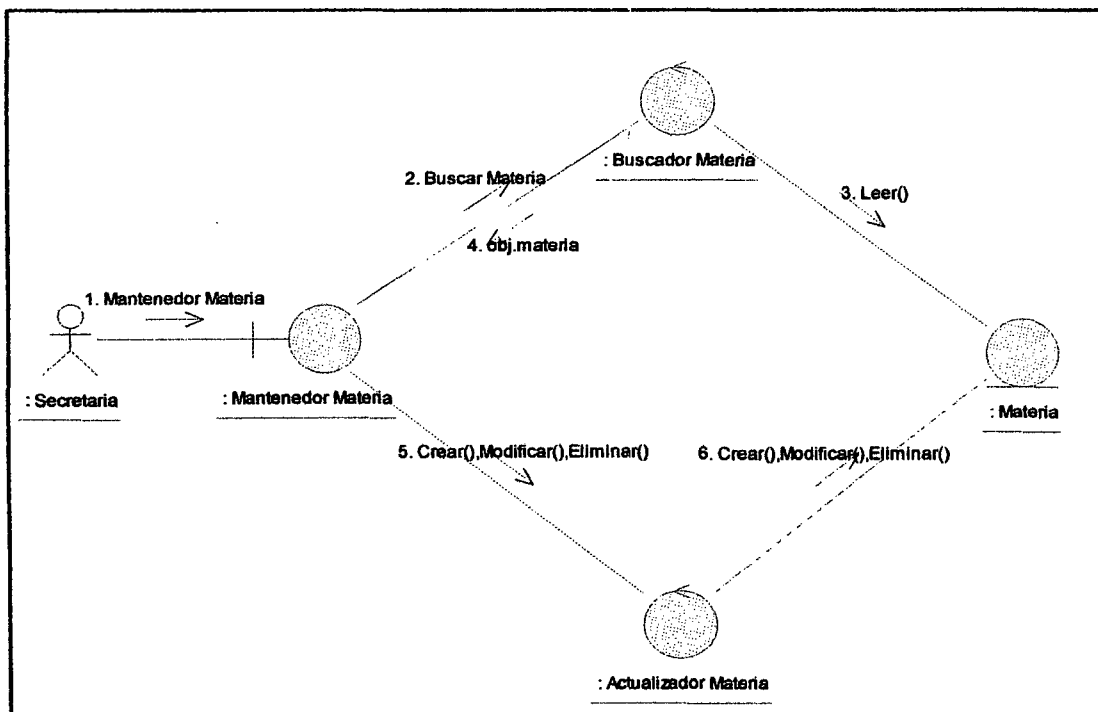


Gráfico 26: Colaboración Registrar material

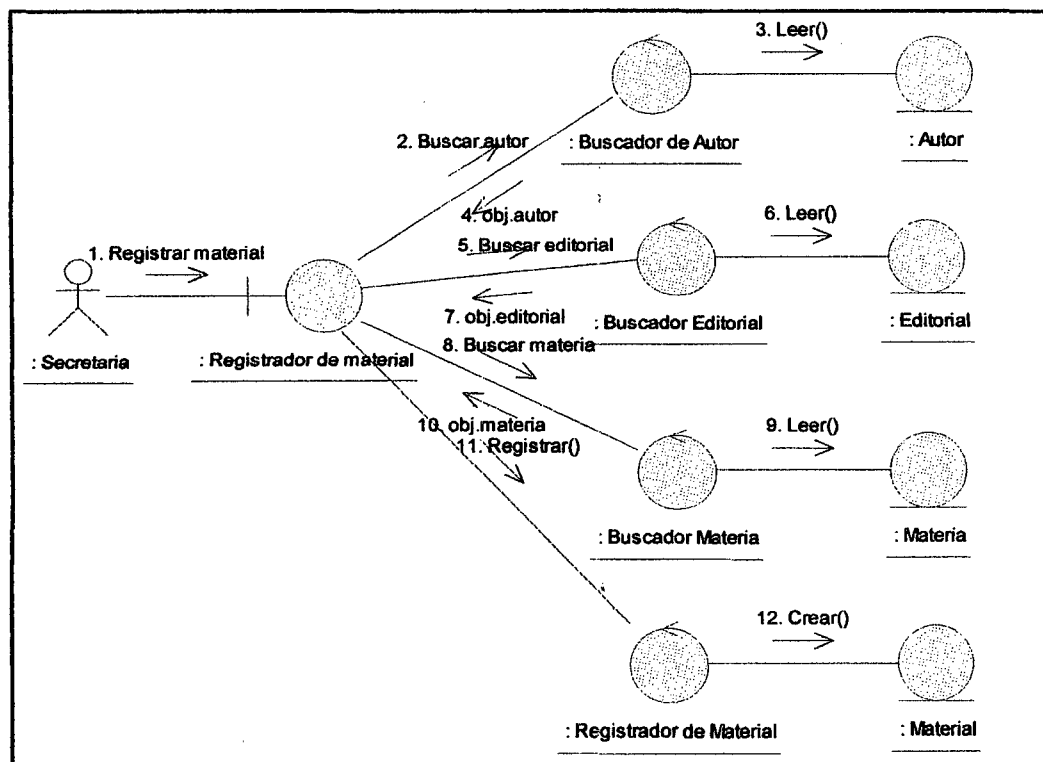


Gráfico 27: Colaboración Registrar préstamo

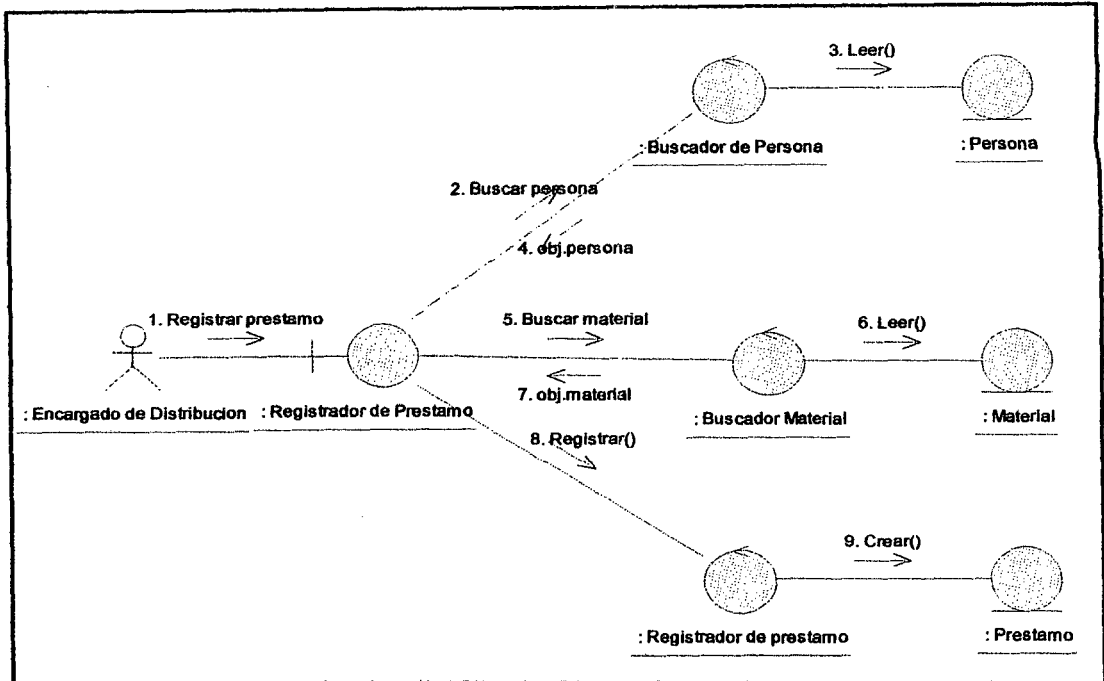


Gráfico 28: Colaboración Registrar reserva

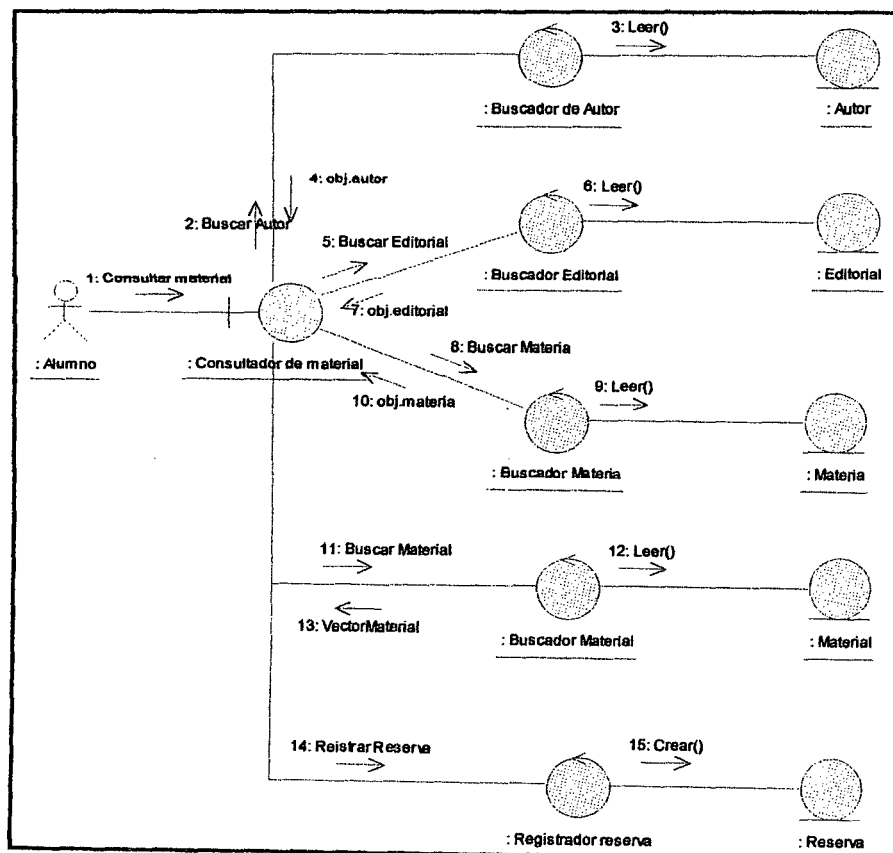


Gráfico 29: Colaboración Registrar Asistencia

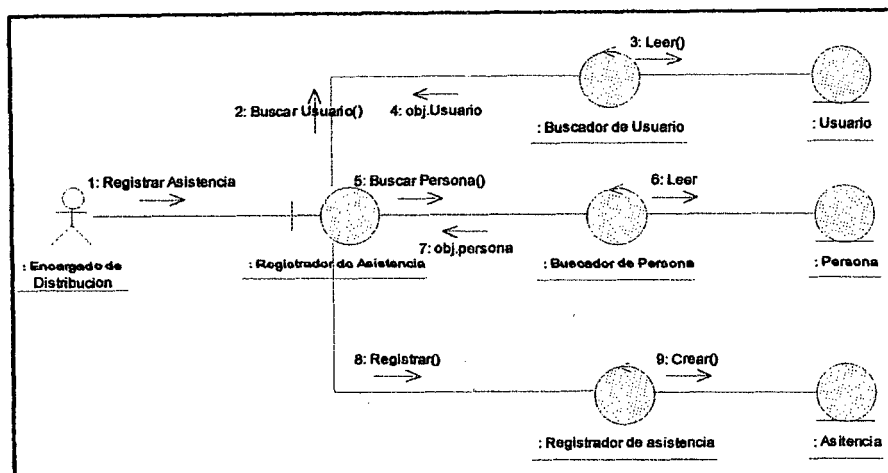


Gráfico 30: Colaboración Registrar Lectura

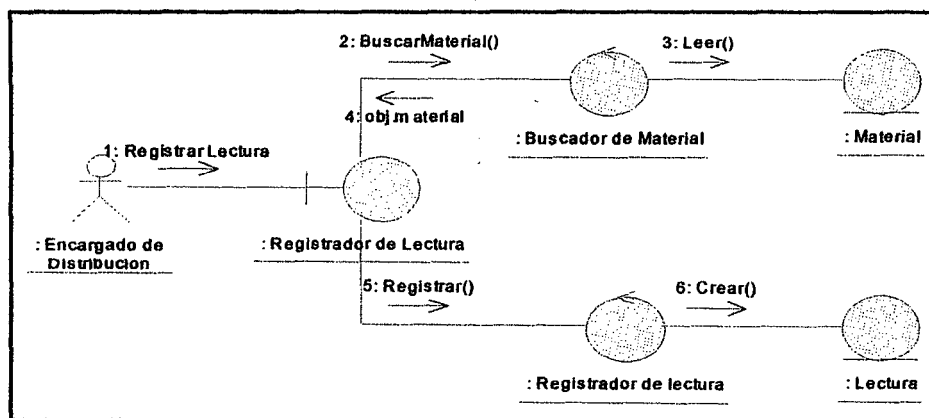


Gráfico 31: Colaboración Validar Préstamo

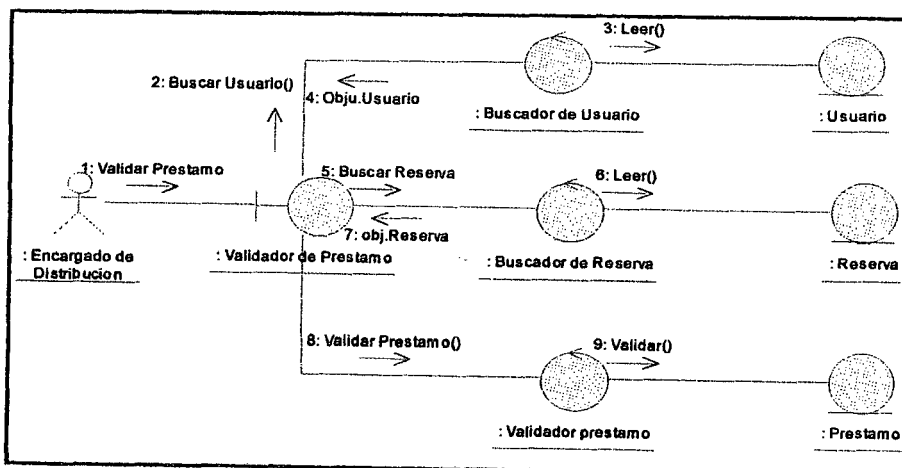


Gráfico 32: Colaboración Registrar Devolución

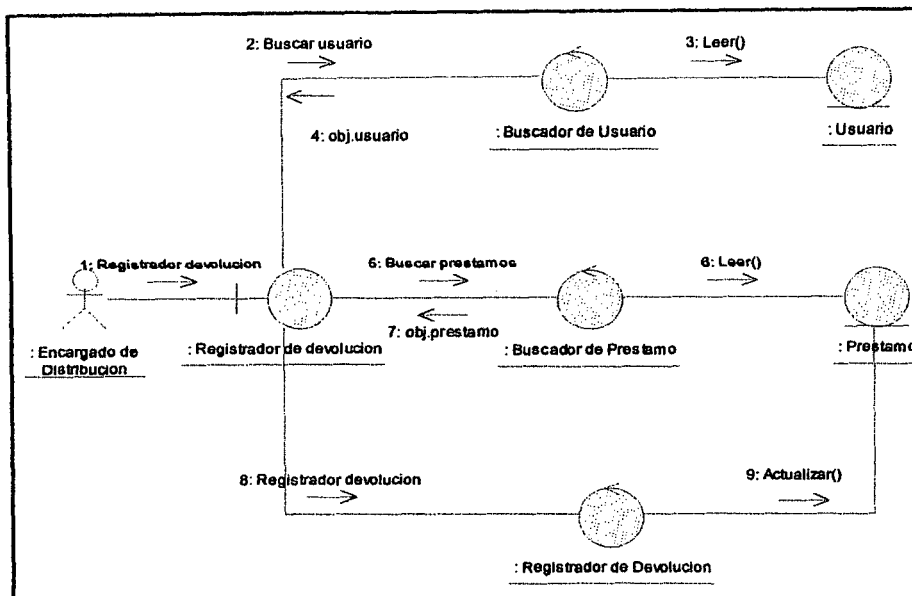
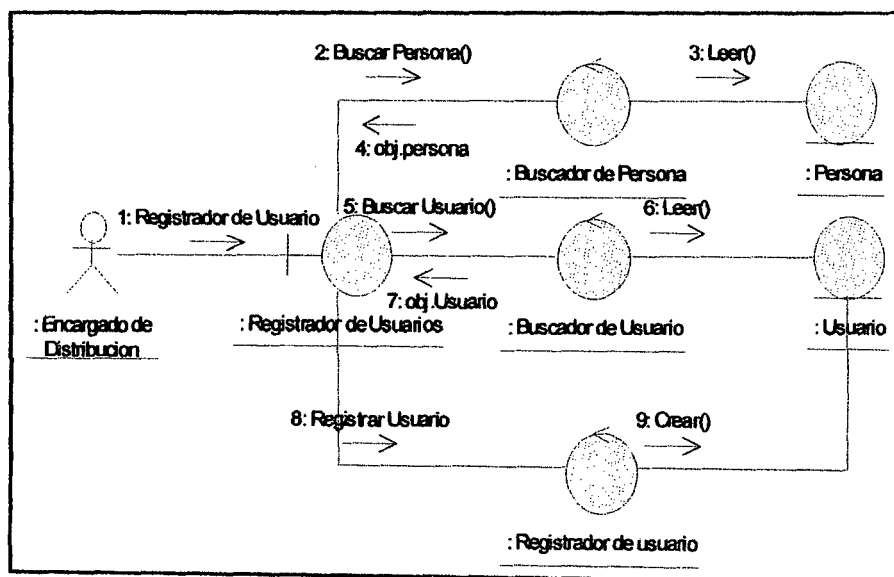
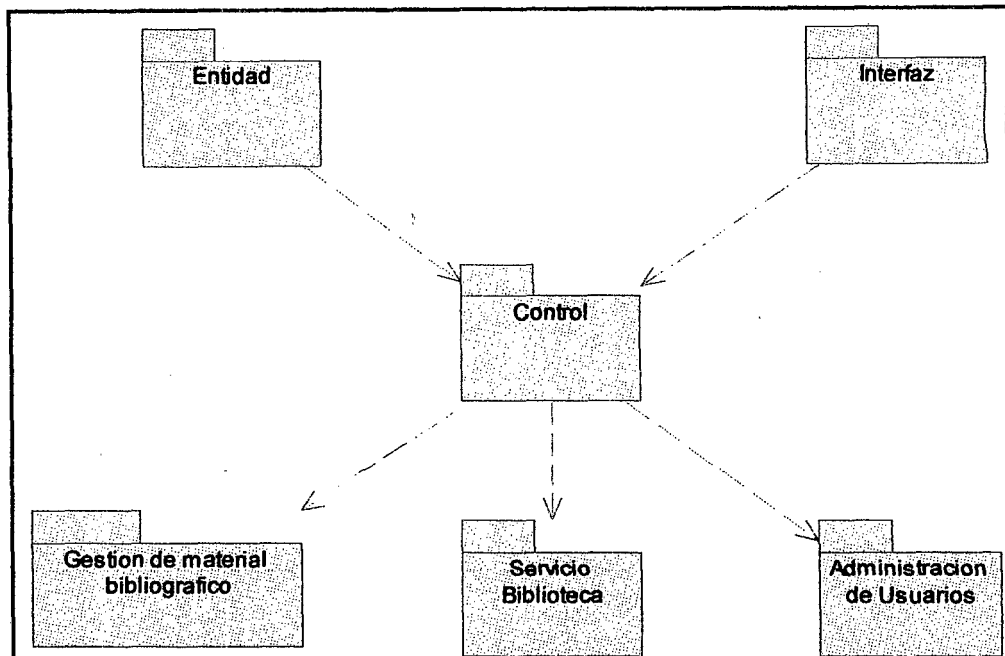


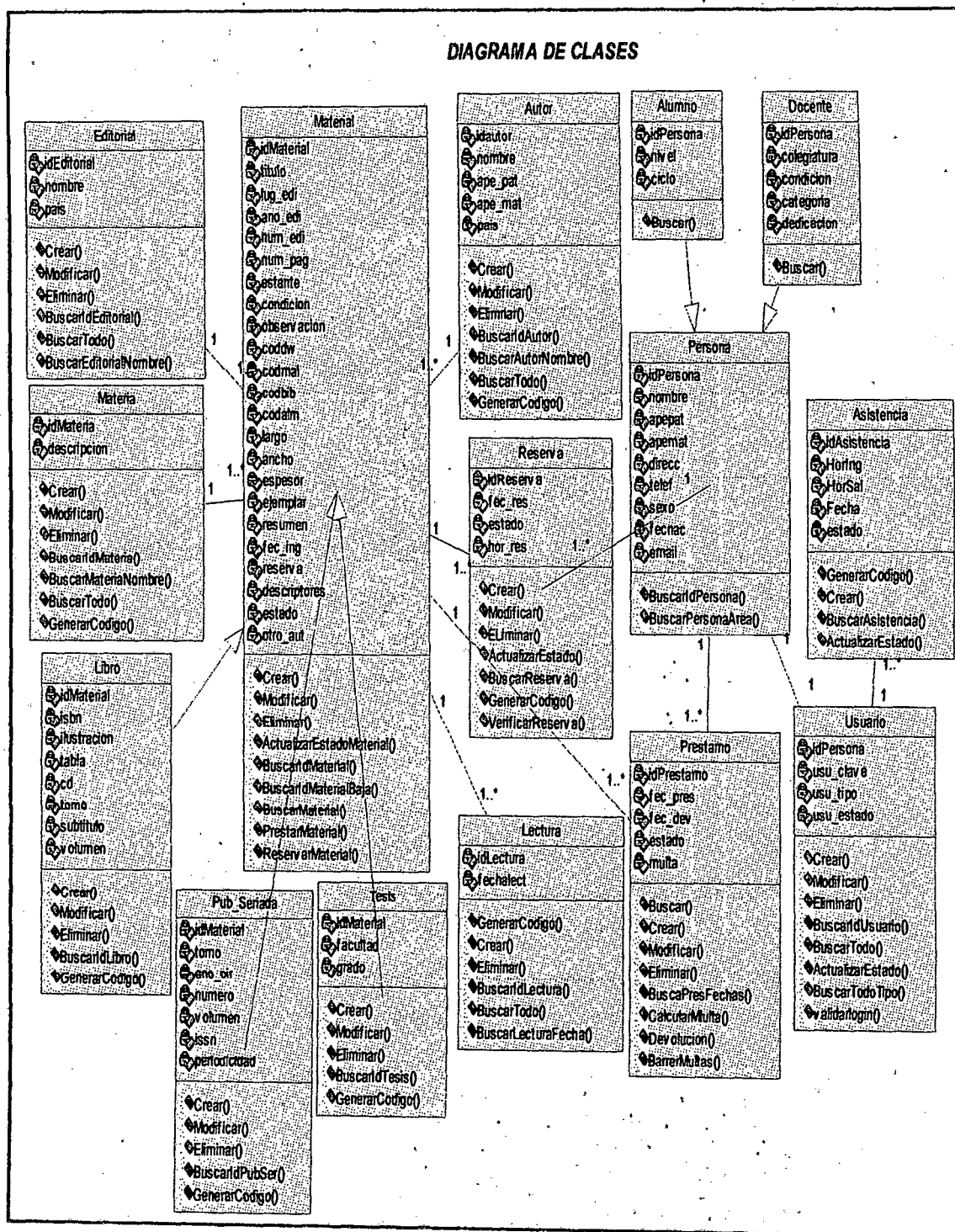
Gráfico 33: Colaboración Registrar Usuario



B) Diagrama de Paquetes del Análisis**Gráfico 34: Diagrama de paquetes**

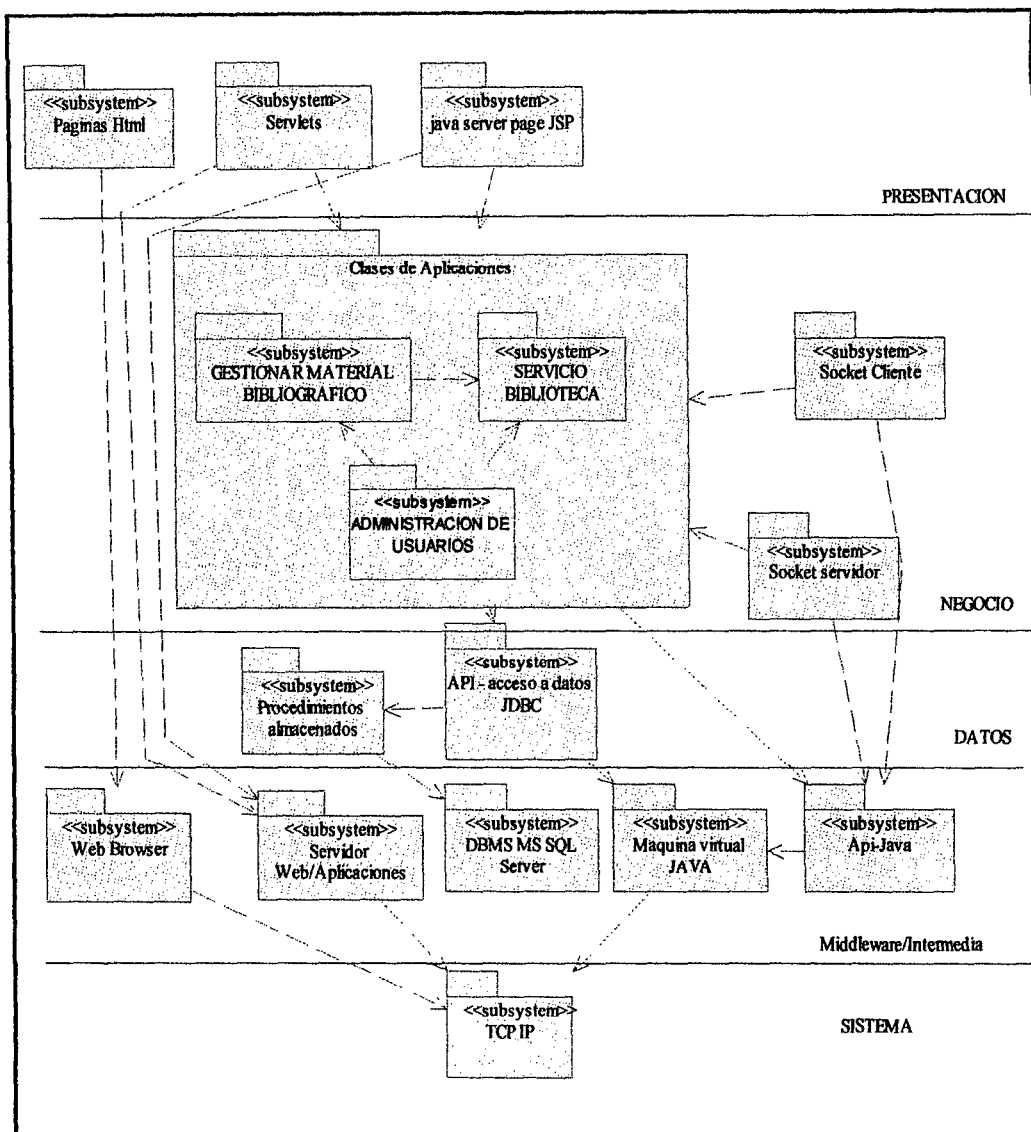
C) Diagrama de Clases

Gráfico 35: Diagrama de clases



D) Diseño de la Arquitectura

Gráfico 36: Diseño Arquitectura



E) Interfaz Vs. Secuencia.

Gráfico 37: Interfaz Registrar Libro

BIBLIOTECAS VIRTUALES DE LA UNO Microsoft Internet Explorer

SISTEMA DE BIBLIOTECA SISBIB

El Compromiso a tu Alcance

Gestión de Material · Servicio de Biblioteca · Administración de Usuarios · Reportes · Estadísticas · Cerrar Sesión

Usuarios **ROGER LAVI CHAVEZ**

REGISTRAR LIBRO

Autogenerado

ve

P. FEDOSEEV
 VICTOR VASQUEZ VILLARUEJA
 ROBERTO B. AVILA ACOSTA
 MIGUEL RODRIGUEZ RIVAS
 ANGEL R. VELASQUEZ FERRAZ
 ALVARO GARRO R.
 PEDRO ISASSI VINUELA
 HARVEY M. DETTEL
 OSCAR VALDIVIA PONCE
 CARMEN VALLEGAS HERNANDEZ
 VICTOR A. MELCHOR ESPINOZA
 JOSE DEL CARMEN CANCHUCAJA VILCHEZ

SISTEMA DE BIBLIOTECA SISBIB

El Compromiso a tu Alcance

Gestión de Material · Servicio de Biblioteca · Administración de Usuarios · Reportes · Estadísticas · Cerrar Sesión

Usuarios **JAVIER LUNA**


00013854

352.44 c
P84b ed.1
o.2



Gráfico No 39: Interfaz Registrar Publicación Seriada

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Biblioteca - Servicio de Biblioteca - Administración de Usuarios - Reportes - Estadísticas - Cerrar Sesión

Usuario: JAVIER LUNA VELASQUEZ

REGISTRAR PUBLICACION SERIADA

Autogenerado

1	MARIA TERESA ANGUEIRA	Nuevo
2	JOSE MARIA DESANTES QUANTER	UI
3	HECTOR DANIEL LERMA GONZALEZ	
4	JOSE MANUEL PRELLEZO	
5	MAURO RODRIGUEZ ESTRADA	
6	BENJAMIN MARTICORENA CASTILLO	
7	ROMAN MARTINEZ	
8	MARK DIX	
9	MARCO ANTONIO FLORES ROSA	
10	MARIO DANTAS	Nuevo
11	MARIAMITA SALAS	
12	GERMAN BECERRA JIMENEZ	
13	VERONICA ANGULO GARCIA	

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Biblioteca - Servicio de Biblioteca - Administración de Usuarios - Reportes - Estadísticas - Cerrar Sesión

Usuario: JAVIER LUNA VELASQUEZ

0001354L

As65

IA666

6.1

50465

T.S

ed.1111

A.1999

n.5

50465

T.S

ed.1111

A.1999

n.5

Gráfico 41: Interfaz Registrar Tesis

[illegible]

SISTEMA DE BIBLIOTECA		SISBIB	
El Conculmiano en el Airance			
Gestión de Material Servicio de Biblioteca Administración de Usuarios Reportes Estadísticas Correo Electrónico	Descargas		
JAVIER LUNA VELASCO			
 00013554L		S6455 T.S	
As65 IA666		ed.1111	
e.1		A.1999 n.5	
			

Gráfico 42: Diagrama De Secuencia Registrar Tesis

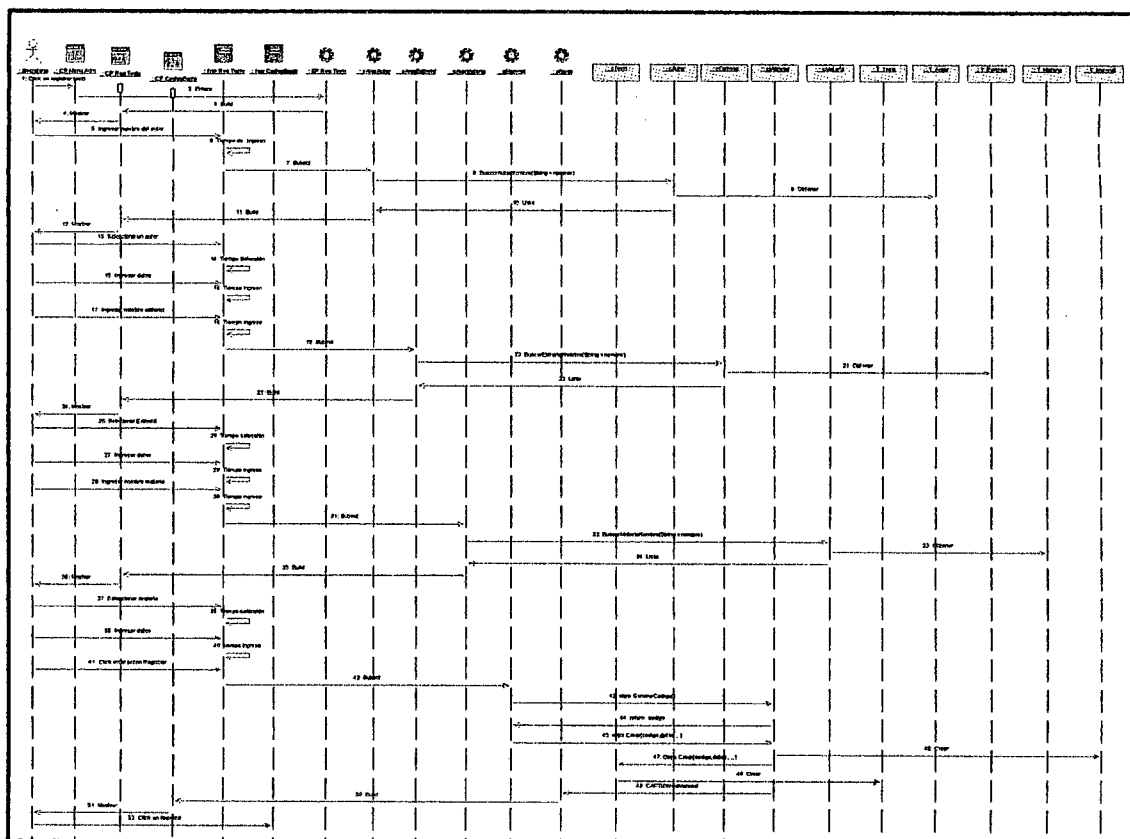


Gráfico 43: Interfaz Mantener Autor

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento es la Alcaza

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Centro de Ayuda

Usuario: JAVIER LUNA VELASQUEZ

MANTENER AUTORES

Buscar | Modificar | Nuevo

Código Autor	Nombre	Apellido	País	Modificar
00002869	W.	TRINIDAD	ESPAÑA	X
00002252		BIODIDACTICA	ALBANIA	X
00000705		VON HUMBOLDT	ALBANIA	X
00000484	A	STADNICHENKO	RUSIA	X
00001016	A	CHENAPPAPOV	RUSIA	X
00001043	A	SANTILLANA	ESPAÑA	X
00001624	A	SANTILLANA	ESPAÑA	X
00001818	A	KERTY	ESPAÑA	X
00002032	A	WOOD	ESPAÑA	X
00002040	A	PLATA BEDIAN	ESPAÑA	X
00003006	A	PERANI	MEXICO	X
00003063	A	MATOR	ESPAÑA	X
00003447	A	D PA	PERU	X
00005002	A	AMO VIESER	ESPAÑA	X
00005956	A	SULLON	MADRID	X

Nuevo

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento es la Alcaza

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Centro de Ayuda

Usuario: ROGER LAY CHAVEZ

NUEVO-AUTOR

Nombre:

Apellido:

País:

Modificar

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento es la Alcaza

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Centro de Ayuda

Usuario: JAVIER LUNA

MODIFICAR-AUTOR

Nombre:

Apellido:

País:

Eliminar

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento es la Alcaza

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Centro de Ayuda

Usuario: JAVIER LUNA VELASQUEZ

ELIMINAR AUTORES

Buscar | Modificar | Nuevo

Código Autor	Nombre	Apellido	País	Modificar
00002869	W.	TRINIDAD	ESPAÑA	X
00002252		BIODIDACTICA	ALBANIA	X
00000705		VON HUMBOLDT	ALBANIA	X
00000484	A	STADNICHENKO	RUSIA	X
00001016	A	CHENAPPAPOV	RUSIA	X
00001043	A	SANTILLANA	ESPAÑA	X
00001624	A	SANTILLANA	ESPAÑA	X
00001818	A	KERTY	ESPAÑA	X
00002032	A	WOOD	ESPAÑA	X
00002040	A	PLATA BEDIAN	ESPAÑA	X
00003006	A	PERANI	MEXICO	X
00003063	A	MATOR	ESPAÑA	X
00003447	A	D PA	PERU	X
00005002	A	AMO VIESER	ESPAÑA	X
00005956	A	SULLON	MADRID	X

Microsoft Internet Explorer

Realmente desea eliminar el registro?

Gráfico 44: Diagrama De Secuencia Registrar Autor

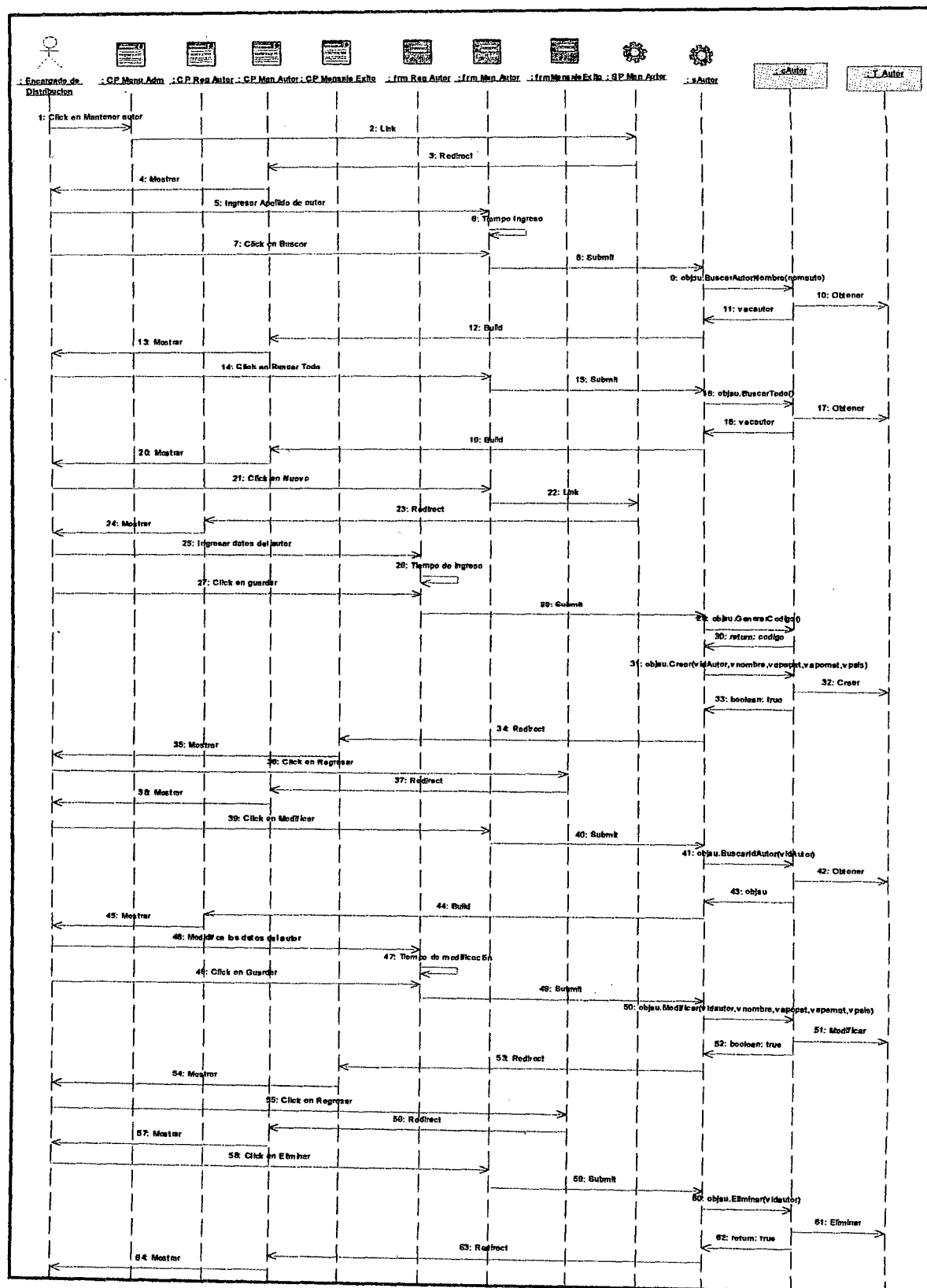


Gráfico 45: Interfaz Mantener Editorial

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Control de Material Servicio de Biblioteca Administración de Usuarios Reportes Estadísticas Control de Stock Usuario: JAVIER LUNA VELAZQUEZ

Editorial

Código Editorial	Nombre	País	Seleccionar
00001	CATEDRA S.A.	ESPAÑA	X
00002	AREOGARANDINA	PERU	X
00003	AMAZONAS	PERU	X
00004	OMEGA S.A.	ESPAÑA	X
00005	IMPRESA UNION	PERU	X
00006	ESTUDIOS Y EDICIONES S.A.	PERU	X
00012	EDITSA S.A.	ESPAÑA	X
00013	EDICORAF	PERU	X
00015	COFAS S.A.	ESPAÑA	X
00016	ENTREIS S.A.	ESPAÑA	X
00017	INTERNACIONAL THOMSON EDITORES S.A.	MEXICO	X
00018	ENVI S.A.	MEXICO	X
00019	ECOS	COLOMBIA	X
00020	ARCO LIBROS S.L.	ESPAÑA	X
00021	MONTANO	PERU	X

Nuevo

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Control de Material Servicio de Biblioteca Administración de Usuarios Reportes Estadísticas Control de Stock Usuario: JAVIER

NUEVO EDITORIAL

Autogenerado:

Modificar

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Control de Material Servicio de Biblioteca Administración de Usuarios Reportes Estadísticas Control de Stock Usuario: JAVIER

MODIFICAR EDITORIAL

Autogenerado:

OMEGA S.A.

ESPAÑA

Eliminar

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Control de Material Servicio de Biblioteca Administración de Usuarios Reportes Estadísticas Control de Stock Usuario: JAVIER LUNA VELAZQUEZ

Editorial

Código Editorial	Nombre	País	Seleccionar
00001	CATEDRA S.A.	ESPAÑA	X
00002	AREOGARANDINA	PERU	X
00003	AMAZONAS	PERU	X
00004	OMEGA S.A.	ESPAÑA	X
00005	IMPRESA UNION	PERU	X
00006	ESTUDIOS Y EDICIONES S.A.	PERU	X
00012	EDITSA S.A.	ESPAÑA	X
00013	EDICORAF	PERU	X
00015	COFAS S.A.	ESPAÑA	X
00016	ENTREIS S.A.	ESPAÑA	X
00017	INTERNACIONAL THOMSON EDITORES S.A.	MEXICO	X
00018	ENVI S.A.	MEXICO	X
00019	ECOS	COLOMBIA	X
00020	ARCO LIBROS S.L.	ESPAÑA	X
00021	MONTANO	PERU	X

Gráfico 46: Diagrama De Secuencia Mantener Editorial

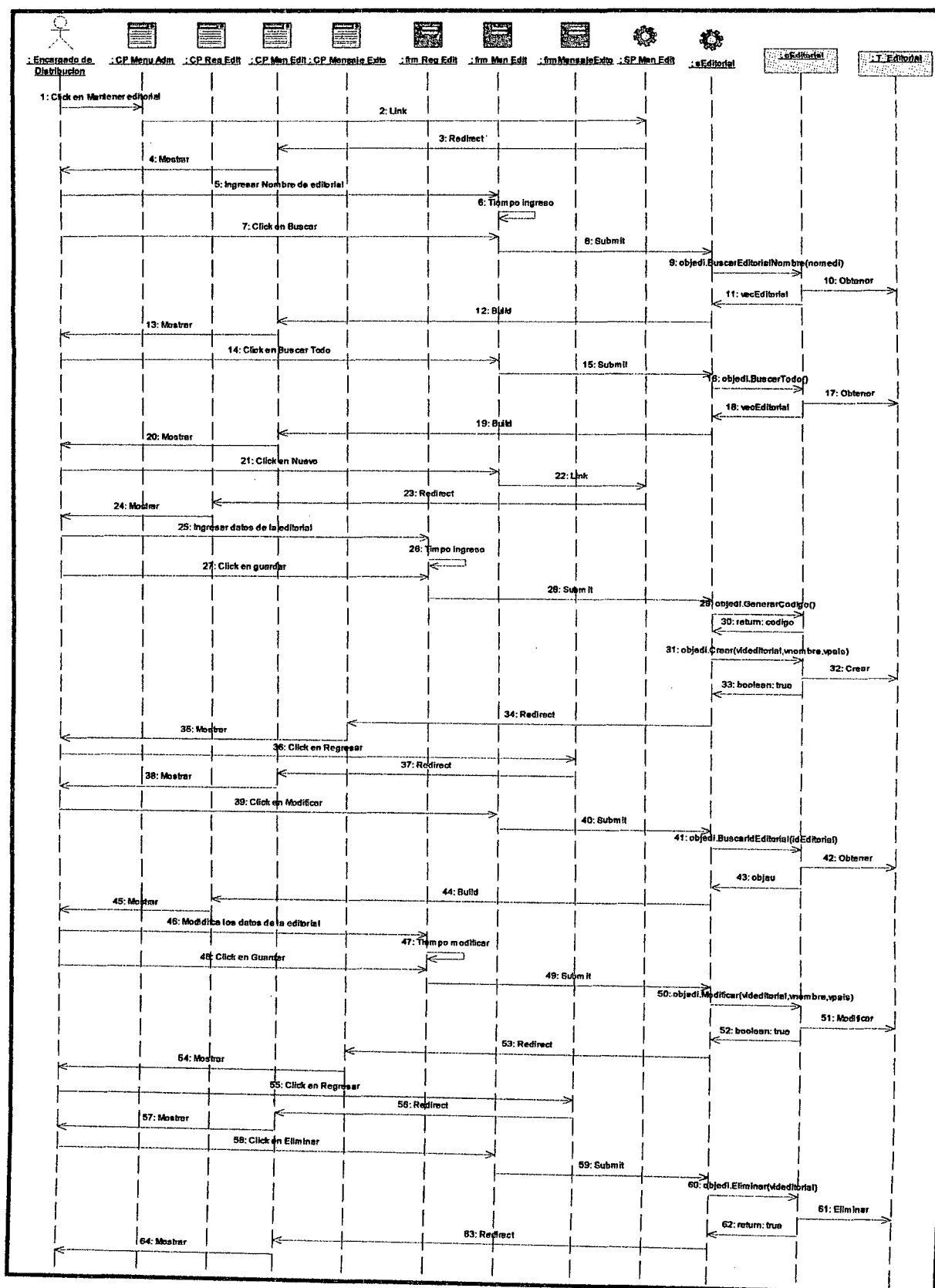


Gráfico 47: Interfaz Mantener Materia

SISTEMA DE BIBLIOTECA SISBIB

El Compromiso a tu Alcance

Gestión de Materiales · Servicio de Biblioteca · Administración de Usuarios · Reportes · Estadísticas · Correo Electrónico

Usuario: JAVIER LUNA VELASQUEZ

Buscar · Mostrar Todo · Nuevo

Código material	Nombre	Eliminar
00001	MATEMÁTICAS	X
00002	CIENCIAS HUMANAS	X
00003	METODOLOGÍA DE LA OBSERVACIÓN	X
00004	INVESTIGACIÓN CIENTÍFICA	X
00007	REDACCIÓN DE DOCUMENTOS	X
00010	PRESUPUESTOS	X
00017	PLANIFICACIÓN	X
00018	TECNICAS CUANTITATIVAS	X
00022	EDUCACIÓN	X
00023	CREATIVIDAD	X
00025	GUIA DE TESIS	X
00027	RECOLECCIÓN DE DATOS	X
00029	MANEJO DE DATOS	X
00029	HIPOTESIS CIENTÍFICA	X
00031	DISEÑOS EXPERIMENTALES	X

1 2 3 4 5 6 7 8 9 10 11 12

Nuevo

SISTEMA DE BIBLIOTECA SISBIB

El Compromiso a tu Alcance

Gestión de Materiales · Servicio de Biblioteca · Administración de Usuarios · Reportes · Estadísticas · Correo Electrónico

Usuario: JAVIER LUNA VELASQUEZ

NUEVO MATERIA

Material:

Nombre:

Guardar Cancelar

Modificar

SISTEMA DE BIBLIOTECA SISBIB

El Compromiso a tu Alcance

Gestión de Materiales · Servicio de Biblioteca · Administración de Usuarios · Reportes · Estadísticas · Correo Electrónico

Usuario: JAVIER LUNA VELASQUEZ

MODIFICAR MATERIA

Material:

Nombre:

Guardar Cancelar

Eliminar

SISTEMA DE BIBLIOTECA SISBIB

El Compromiso a tu Alcance

Gestión de Materiales · Servicio de Biblioteca · Administración de Usuarios · Reportes · Estadísticas · Correo Electrónico

Usuario: JAVIER LUNA VELASQUEZ

Buscar · Mostrar Todo · Nuevo

Código material	Nombre	Eliminar
00001	MATEMÁTICAS	X
00002	CIENCIAS HUMANAS	X
00003	METODOLOGÍA DE LA OBSERVACIÓN	X
00004	INVESTIGACIÓN CIENTÍFICA	X
00007	REDACCIÓN DE DOCUMENTOS	X
00010	PRESUPUESTOS	X
00017	PLANIFICACIÓN	X
00018	TECNICAS CUANTITATIVAS	X
00022	EDUCACIÓN	X
00023	CREATIVIDAD	X
00025	GUIA DE TESIS	X
00027	RECOLECCIÓN DE DATOS	X
00029	MANEJO DE DATOS	X
00029	HIPOTESIS CIENTÍFICA	X
00031	DISEÑOS EXPERIMENTALES	X

1 2 3 4 5 6 7 8 9 10 11 12

Gráfico 48: Diagrama De Secuencia Mantener Materia

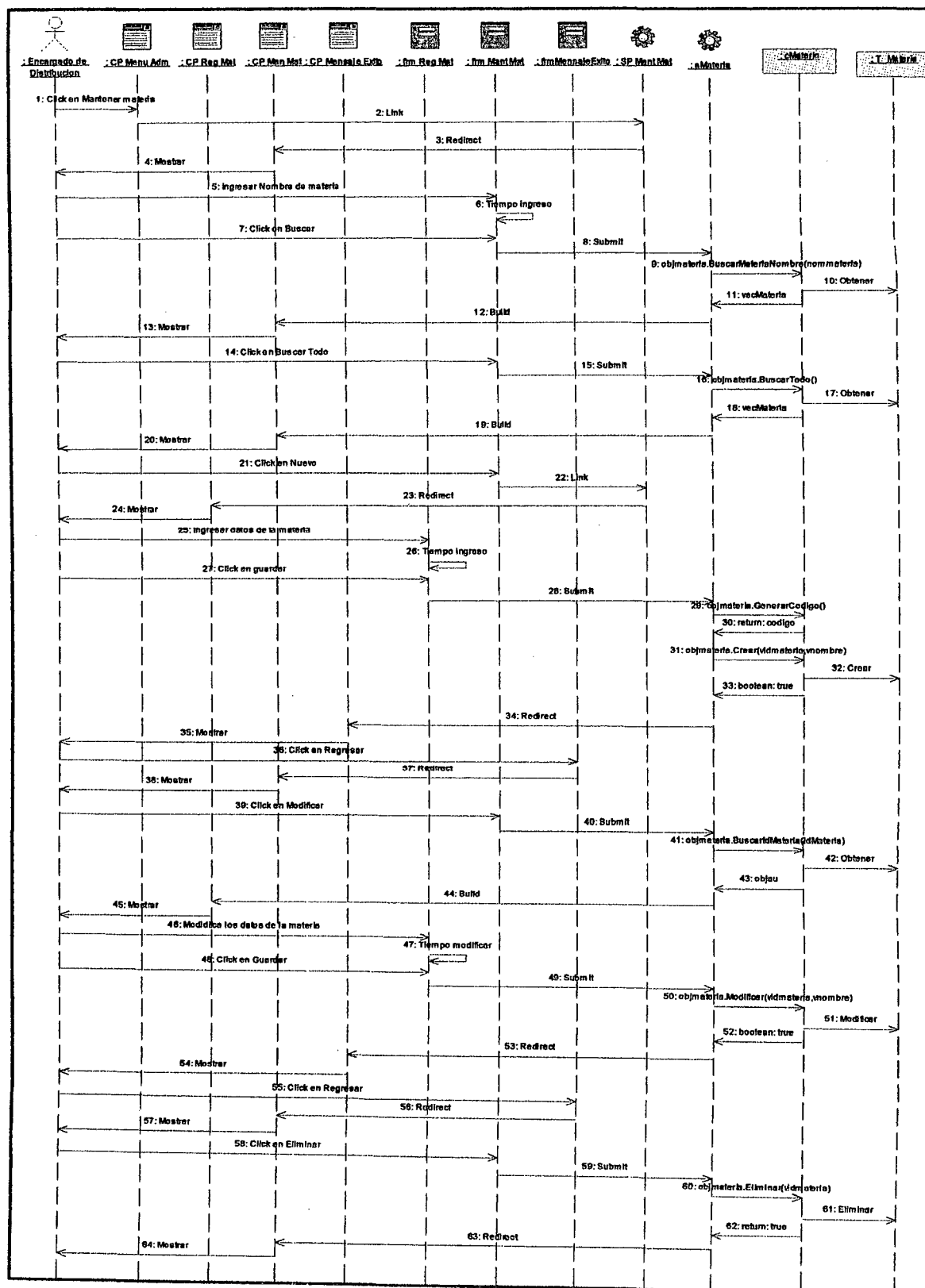


Gráfico 49: Interfaz Registrar Préstamo

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material | Servicio de Biblioteca | Reportes | Estadísticas | Cerrar Sesión

Usuario: ROGER LAVI CHAVEZ

REGISTRAR PRESTAMO

COPIAS: 0002020142

000000003L

Consultar

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material | Servicio de Biblioteca | Reportes | Estadísticas | Cerrar Sesión

Usuario: ROGER LAVI CHAVEZ

REGISTRAR PRESTAMO

COPIAS: 0002020142

ROGER

LAVI CHAVEZ

000000003L

EL SER HUMANO Y SUS R

HEBREO

PONTIFICIA UNIVERSIDAD

ANATOMIA

Registrar

Autorización de Salida

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Cerrar Sesión

Usuario: JAVIER LUNA VELAZO

BIBLIOTECA CENTRAL

AUTORIZACION DE SALIDA DE MATERIAL BIBLIOTECARIO

USUARIO: CONTRERAS NARRO ERIKA MERCEDES -
Alumno

FACULTAD: Ing. de Sistemas

CODIGO DEL LIBRO: 000000011L No. CLASIFICACION: 311 W84

AUTOR: MOTHES . J.

TITULO: ESTADISTICA APLICADA A LA INGENIERIA

Duración: 29 de Octubre del 2007

LUNA VELAZO JAVIER
BIBLIOTECARIO:

RECIBI CONT ORDEN
USUARIO:

Gráfico 51: Interfaz Registrar Devolución

BIBLIOTECA VIRTUAL DE LA UHU - Microsoft Internet Explorer

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material - Servicio de Biblioteca - Administración de Usuarios - Reportes - Estadísticas - Cerrar Sesión

Usuario: ROGER LAMI CHAVEZ

CONSULTA DE DEVOLUCIÓN

LIBRO	0002020146	
USUARIO	000000006L	Consultar

Retornar

BIBLIOTECA VIRTUAL DE LA UHU - Microsoft Internet Explorer

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material - Servicio de Biblioteca - Administración de Usuarios - Reportes - Estadísticas - Cerrar Sesión

Usuario: ROGER LAMI CHAVEZ

REGISTRAR DEVOLUCIÓN

LIBRO: LUNA VELASQUEZ, JAVIER

Código de Material	Código de Usuario	Nombre de Usuario	Nombre de Material	Fecha de Devolución	Estado	Acciones
1403070001	000000006L	JUAN RULFO P.	PEDRO PARRAMO EL LLANO EN LLAMAS	14/03/2007	11	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/> <input type="button" value="X"/>

Retornar

Gráfico 52: Diagrama De Secuencia "Registrar Devolución"

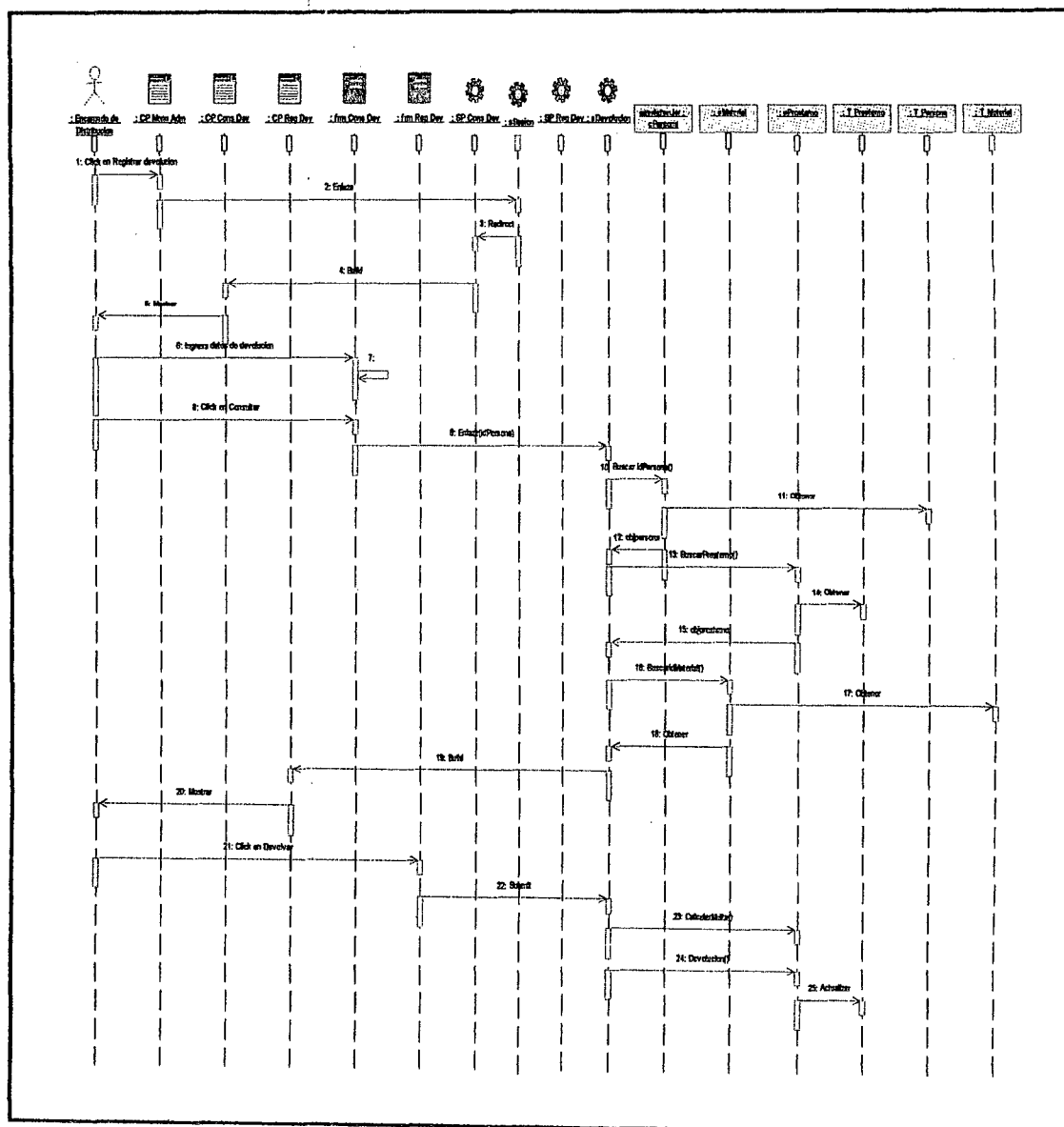


Gráfico 53: Interfaz Consulta-Reserva

Archivo Edición Ver Favoritos Herramientas Ayuda

http://localhost:8084/sisbib/com_material.jsp

SISTEMA DE BIBLIOTECA SISBIB

Consulta de Libros en Línea

Libro Titulo

CODIGO	TITULO	AUTOR	OPCIONES
005.13 A21A	APLICANDO JAVA 2	RAMIREZ CORRALES LUIS EDUARDO	<input type="button" value="X"/>
SUBTITULO:	MACRO		
EDITORIAL:	PERU, 2002		
EDICION:			
RESUMEN:			
NO PAGINAS:	305		
No de Ejemplares:	1		
005.1 K41H	NETSCAPE JAVASCRIPT 1.2	KENT, PETER	<input type="button" value="X"/>
SUBTITULO:	TODAS LAS PLATAFORMAS		
EDITORIAL:	MUNDI PRENSA		
EDICION:	MADRID, 1997		
AUTORES:	JOHN KENT		
RESUMEN:			
NO PAGINAS:	201		

CONSULTAS INTERNET - Microsoft Internet Explorer

http://localhost:8084/sisbib/ReservarCODDEW-005.16&CODNAL=K41H&TXTPARA=C

RESERVA DE LIBROS

DATOS PERSONALES

FECHA DE RESERVA FECHA DE DEVOLUCION

FECHA DE RESERVA FECHA DE DEVOLUCION

EJEMPLARES

CODIGO	TITULO	DE RESERVA
000001423L	NETSCAPE JAVASCRIPT 1.2	☐
000001423L	NETSCAPE JAVASCRIPT 1.2	☐

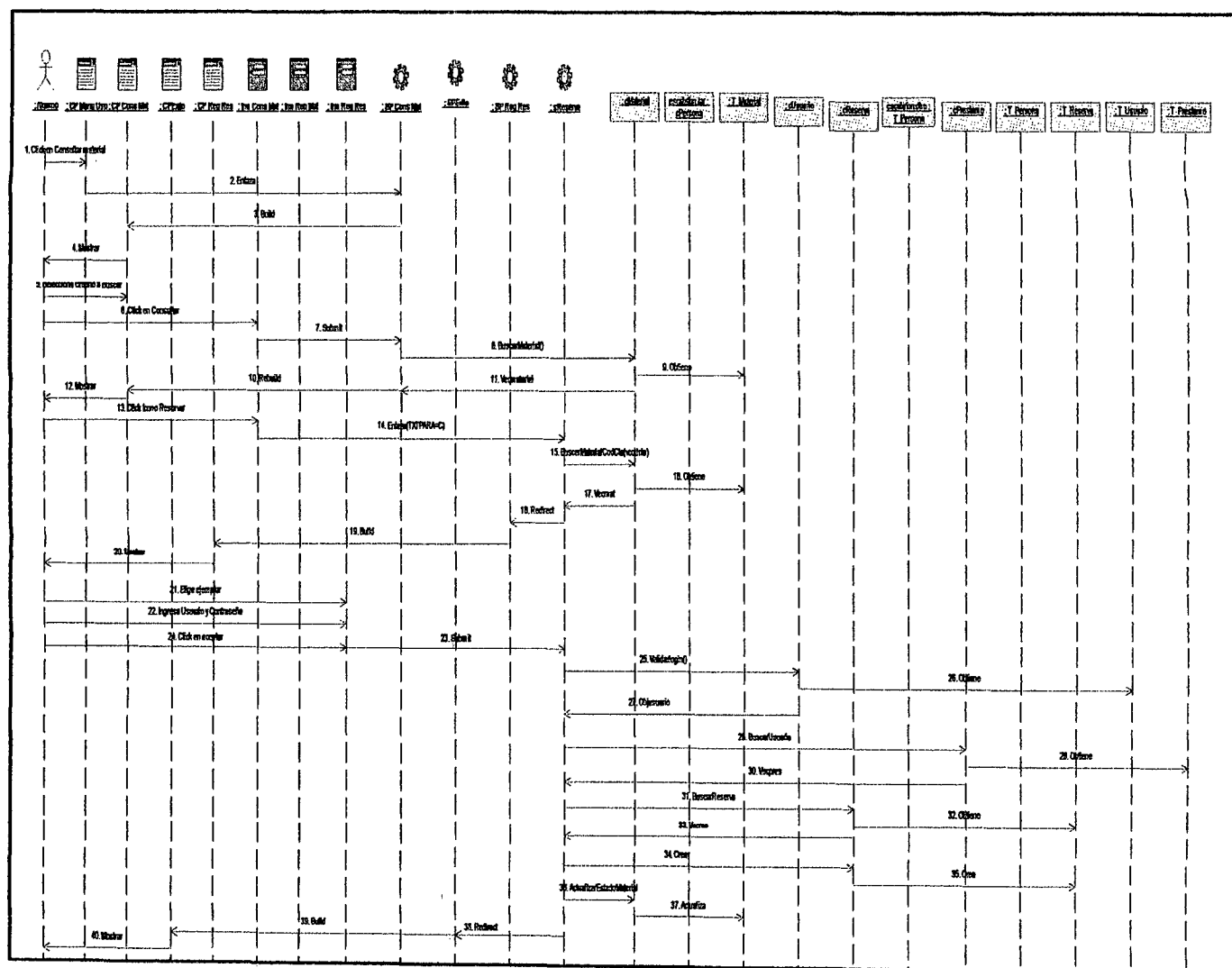


Gráfico 55: Interfaz Validar Reserva

SISTEMA DE BIBLIOTECA SISBIB

El Consentimiento a la Albarca

Gestión de Material · Servicio de Biblioteca · Administración de Usuarios · Reportes · Estadísticas · Cerrar Sesión

Usuario: **ROGER LAMI CHAVEZ**

VALIDAR PRESTAMO

0002020134

CODIGO	TITULO	AUTOR	OPCIONES
094.4 R94p	PEDRO PARRAMO. EL LLANO EN LLANAS	RULFO P. JUAN	 
SUBTITULO: EDITORIAL: SEIX BARRAL EDICION: BOGOTA, 1983 AUTORES: RESUMEN: Nº PAGINAS: 222 Hora de Reserva: 12:12:09			

Gráfico 56: Diagrama de Secuencia Validar Reserva

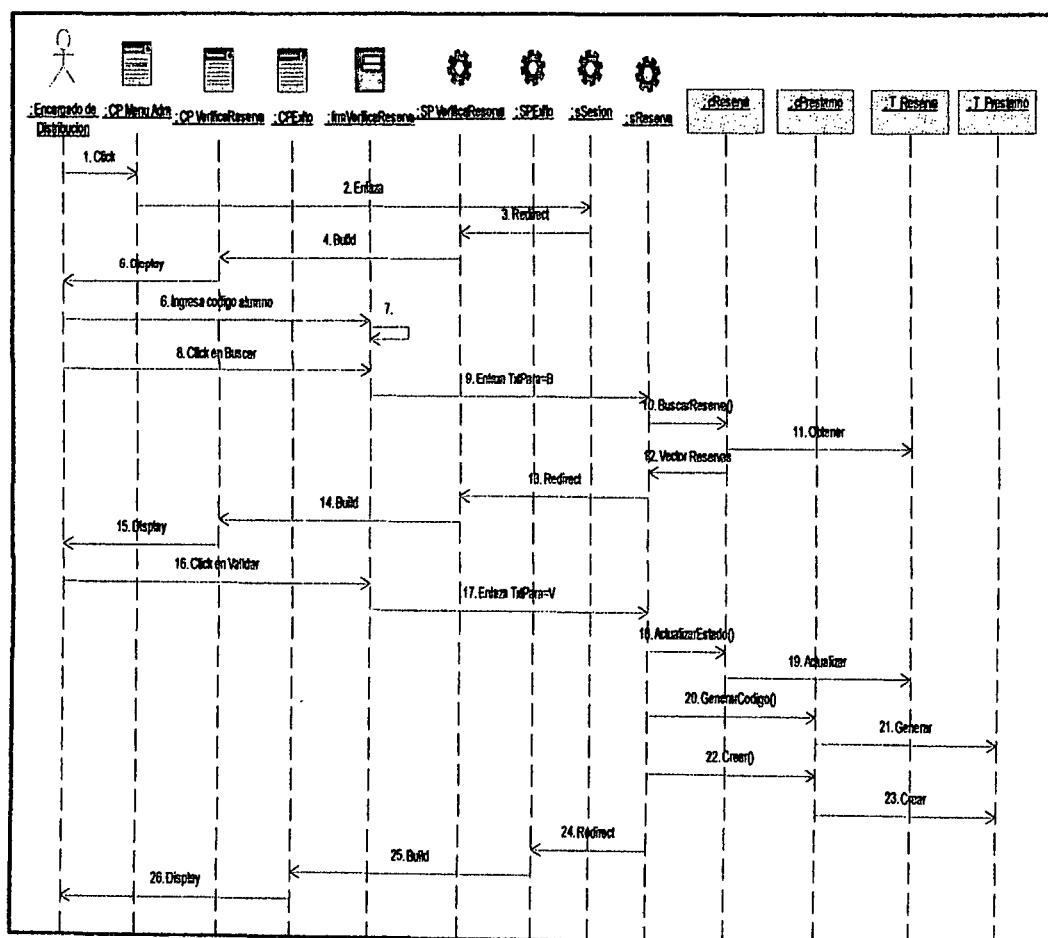


Gráfico 57: Interfaz Registrar Lectura

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Cerrar Sesión | Usuarios | JAVIER LUNA VELASQUEZ

REGISTRAR LECTURA

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Cerrar Sesión | Usuarios | JAVIER LUNA VELASQUEZ

LA TRANSACCION SE REALIZO CORRECTAMENTE

EXITO

Guardando los datos, por favor espere...

Gráfico 58: Diagrama de Secuencia Registrar Lectura

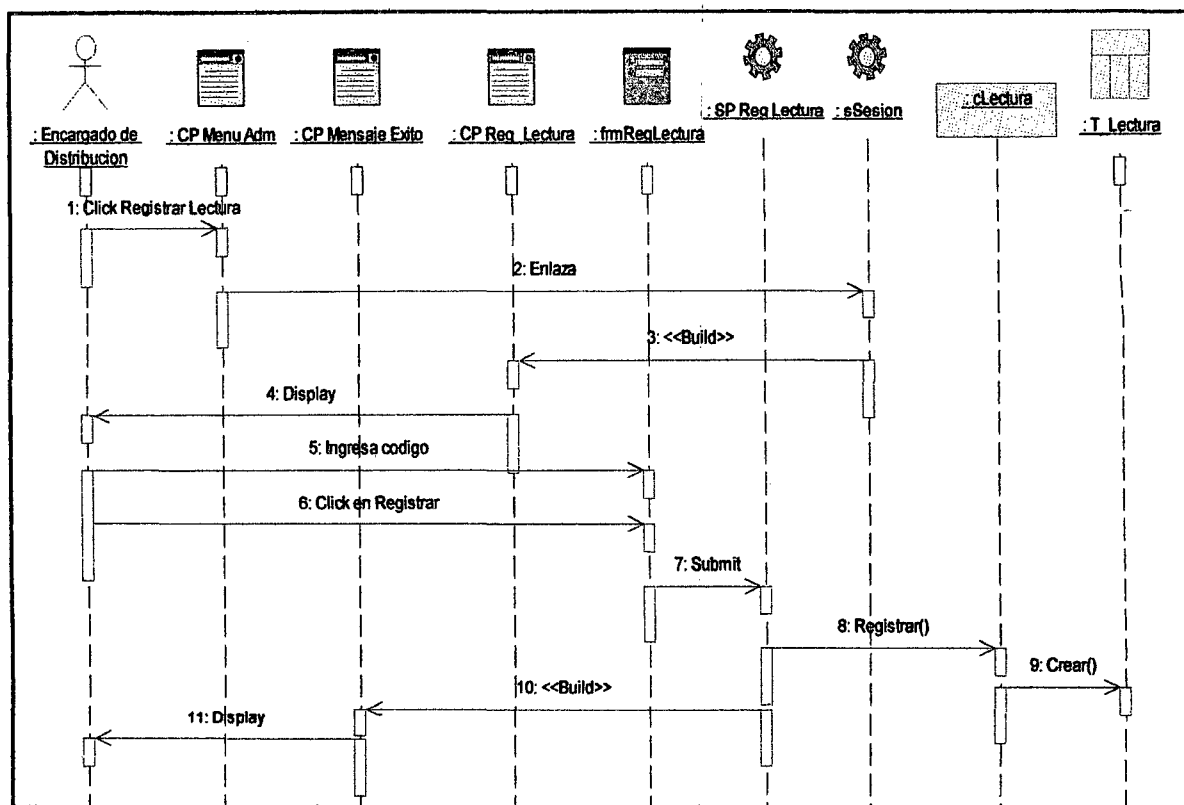


Gráfico 59: Interfaz Registrar Asistencia

SISTEMA DE BIBLIOTECA SISBIB

El Compromiso a su Alcance

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Correr Sesión

Usuario: JAVIER LUIS VELASQUEZ

REGISTRAR ASISTENCIA

Buscar Interrogatorio por Nombre

Identificación	0002040100	
Nombre	ERKA MERCEDES	
Apellido	CONTRERAS NARRO	
Identificación	ALUMNO	
Identificación	INGENIERIA DE SISTEMAS	
Identificación	INGENIERIA DE SISTEMAS	
Identificación	Usuario Apto para el ingreso a Biblioteca	
Identificación	ENTRADA	25/03/2008 - 17:32:57

Gráfico 60: Diagrama de Secuencia Registrar Asistencia

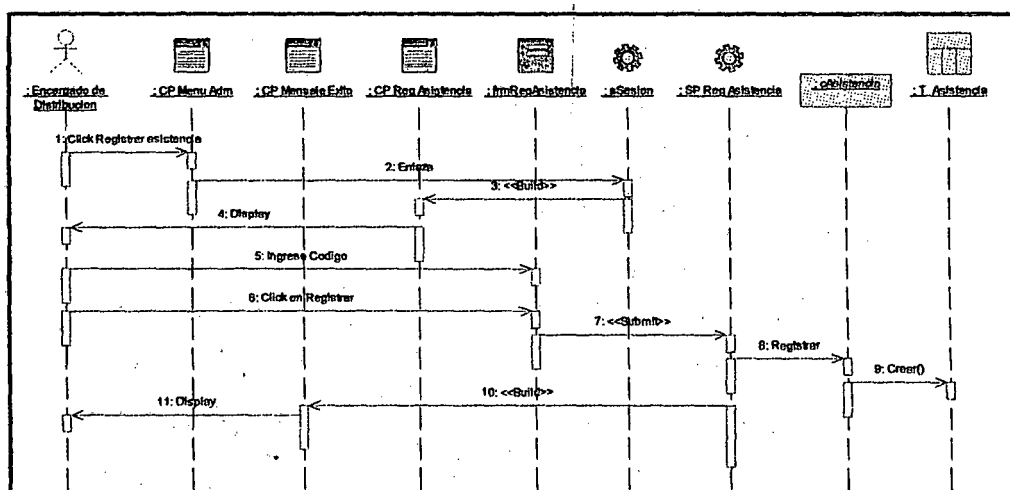


Gráfico 61: Interfaz Mantener Usuarios

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Cerrar Sesión

Usuario: JAVIER LUIS VELASQUEZ

Código Usuario: 0002040100 | Nombre: ERIKA MERCEDES | Apellidos: CONTRERAS NARRO | Tipo: Alumno | Estado: Alta | Mantener

0002040100 | [Buscar] | [Mostrar Todo] | Ordenar Usuario del tipo: --

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestión de Material | Servicio de Biblioteca | Administración de Usuarios | Reportes | Estadísticas | Cerrar Sesión

Usuario: JAVIER LUIS VELASQUEZ

MODIFICAR USUARIO

1. Estradice

2. ERIKA MERCEDES

3. CONTRERAS

4. ALUMNO

5. *****

6. *****

7. Alumno

[Guardar]

[Cancelar]

Gráfico 62: Diagrama de Secuencia Mantener Usuario

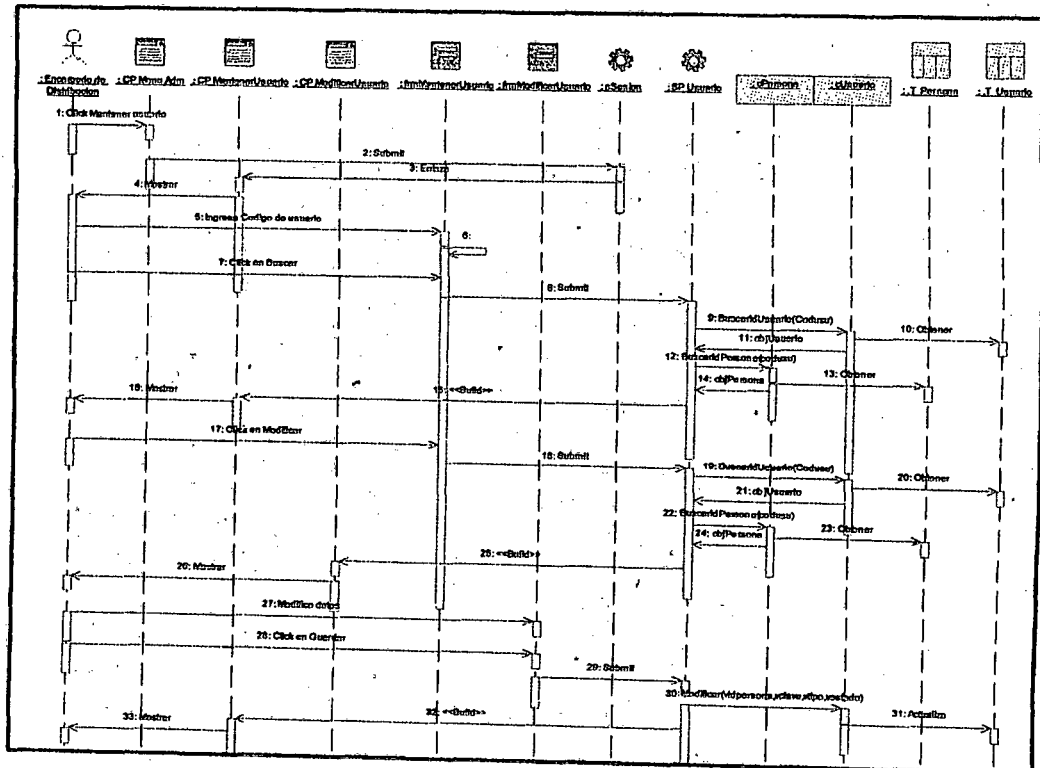


Gráfico 63: Interfaz Reporte de Materiales Según Clasificación

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Cesión de Material Servicio de Biblioteca Administración de Usuarios Reportes Estadísticas Cerrar Sesión Usuario: JAVIER LUNA VELASQUEZ

UNIVERSIDAD NACIONAL DE UCAYALI

REPORTE DE MATERIALES SEGUN CLASIFICACION

Seleccione tipo de material: **SISTEMAS**

No	Codigo Material	Título	Año Edición	No ejemplar
1	000001546P	SISTEMICA: JOURNAL OF THE INTER AND TRANSDISCIPLINARY MANAGEMENT OF COMPLEXITY	1990	1
2	000001547D	SISTEMICA: JOURNAL OF THE INTER AND TRANSDISCIPLINARY MANAGEMENT OF COMPLEXITY	1990	2
3	000001548P	SISTEMICA: JOURNAL OF THE INTER AND TRANSDISCIPLINARY MANAGEMENT OF COMPLEXITY	1990	1
4	000001549P	SISTEMICA: JOURNAL OF THE INTER AND TRANSDISCIPLINARY MANAGEMENT OF COMPLEXITY	1990	2
5	000014321L	LASISTEMICA LOS SISTEMAS BLANDOS Y LOS SISTEMAS DE INFORMACION	1990	1
6	000001990L	INTRODUCCION E HISTORIA DE LA CIBERNETICA	1968	1
7	000002000L	INTRODUCCION E HISTORIA DE LA CIBERNETICA	1968	2
8	000001913L	INTRODUCCION A LA INFORMATICA	1991	1
9	000001914L	INTRODUCCION A LA INFORMATICA	1991	2
10	000001924L	INFORMATICA CONTABLE	1992	1
11	000001925L	INFORMATICA CONTABLE	1992	2
12	000001934L	EXTENSION Y PROGRAMACION FORTRAN	1981	0

Internet 100%

Gráfico 64: Diagrama de Secuencia Reporte de Materiales Según Clasificación

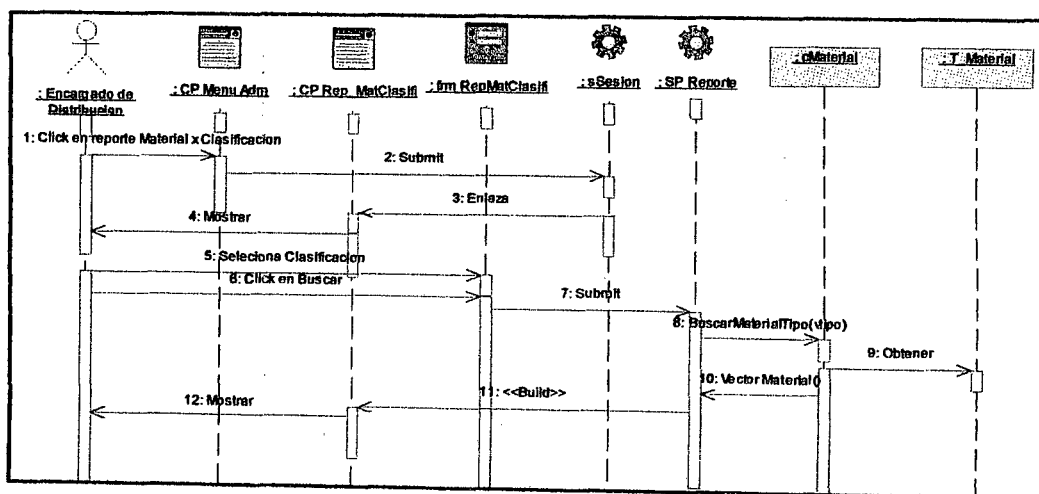


Gráfico 65: Interfaz Reporte de Prestamos Realizados A Usuarios

SISTEMA DE BIBLIOTECA SISBIB

BIBLIOTECA PÚBLICA DEL ALCAZAR

Gestión de Material · Servicio de Prestamos · Administración de Usuarios · Reportes · Estadísticas · Cerrar Sesión

Usuario: JAVIER LUIS VELASQUEZ

UNIVERSIDAD NACIONAL DE UCAVALI

PRESTAMOS REALIZADOS POR LOS USUARIOS

DESDE: 01/03/2007 HASTA: 31/05/2007

Cod. Prestamo	Cod. Material	Título	Usuario	Prestamista	Fecha Prest.	Fecha Devoluc.	Estado
0805070001	000004318L	EL ALGODON	PANDURO BASILONIA CECILIA	LOPEZ HELENDEZ WILDER	08/05/2007	11/05/2007	Devuelto
0405070008	000000003L	PEDRO PARRAMO EL LLANO EN LLAMAS	FASABI HIDALGO FARLEY MARCELO	ALBORNOZ JIMENEZ GABY LUCIA	04/05/2007	12/05/2007	Devuelto
0405070007	000005328L	CONTABILIDAD GENERAL LA MANERA MAS FACIL DE APRENDER CONTABILIDAD	CRIDILLO HIDALGO VIRGINIA	LAVI CHAVEZ ROGER	04/05/2007	11/05/2007	Devuelto
0405070004	000004348L	LOS SUELOS Y SU FERTILIDAD	REYES VALERA DINA GIANINA	LAVI CHAVEZ ROGER	04/05/2007	12/05/2007	Devuelto
0405070006	000003363L	WILLIAMS OBSTETRICIA	AMBICHO DA CRUZ MONICA BRIGGETH	LAVI CHAVEZ ROGER	04/05/2007	12/05/2007	Devuelto
0405070001	000010738L	GUIA DE PRACTICAS DE METEOROLOGIA GENERAL	GAMARRA BOCANEGRA EVER JAVIER	LAVI CHAVEZ ROGER	04/05/2007	11/05/2007	Devuelto
0405070003	000010940L	FISIOLOGIA DE LAS PLANTAS	ALVA HIDALGO ABDIAS	LAVI CHAVEZ ROGER	04/05/2007	08/05/2007	Devuelto
0405070003	000010993L	FISIOLOGIA DE LAS PLANTAS	POVIS CUZMAN YERALDIN	LAVI CHAVEZ ROGER	04/05/2007	08/05/2007	Devuelto

Leto: 100%

Gráfico 66: Diagrama de Secuencia "Reporte de Préstamos Realizados a Usuarios"

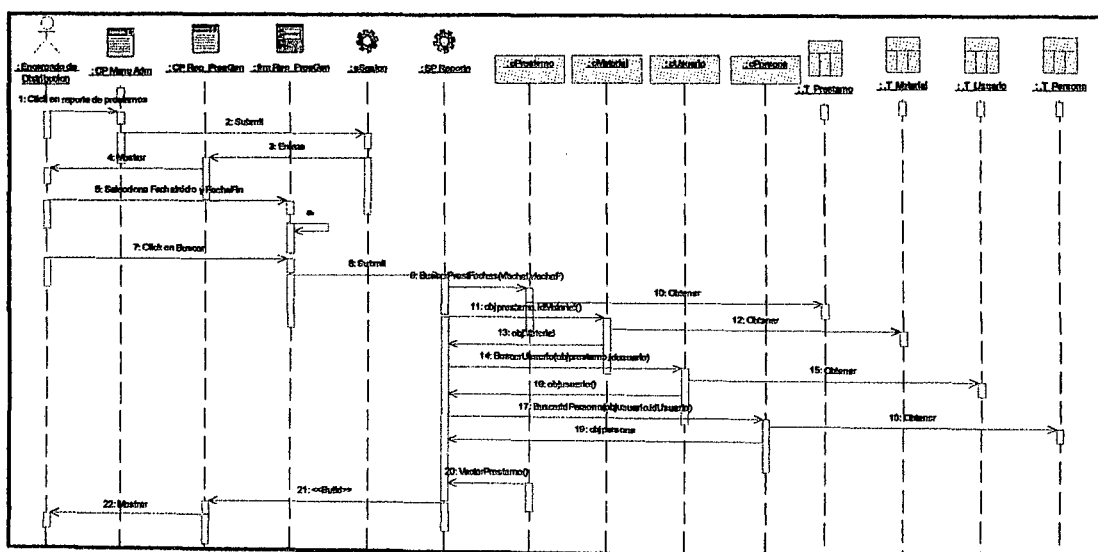


Gráfico 67: Interfaz Reporte de Prestamos por Fecha

SISTEMA DE BIBLIOTECA SISBIB

El Compromiso por el Aprendizaje

Gestión de Material Servicio de Biblioteca Administración de Usuarios Reportes Estadísticas Cerrar Sesión

Usuario: JAVIER LUNA VELASQUEZ

REPORTES DE BIBLIOTECA

PRESTAMOS POR FECHA

Fecha: 10-mar-2008
Hora: 2:20:25

UNIVERSIDAD NACIONAL DE UCAYALI

DESDE: 01/03/2007 HASTA: 30/11/2007

Fecha:	Nro de Prestamos
04/05/2007	8
07/05/2007	3
08/05/2007	23
09/05/2007	12
10/05/2007	13
11/05/2007	19
12/05/2007	8
14/05/2007	5
15/05/2007	29
16/05/2007	8
17/05/2007	18
18/05/2007	21
19/05/2007	9
21/05/2007	20
22/05/2007	21
23/05/2007	16
29/05/2007	25
TOTAL GENERAL	258

Libro Internet 100%

Gráfico 68: Diagrama de Secuencia Reporte de Prestamos por Fecha

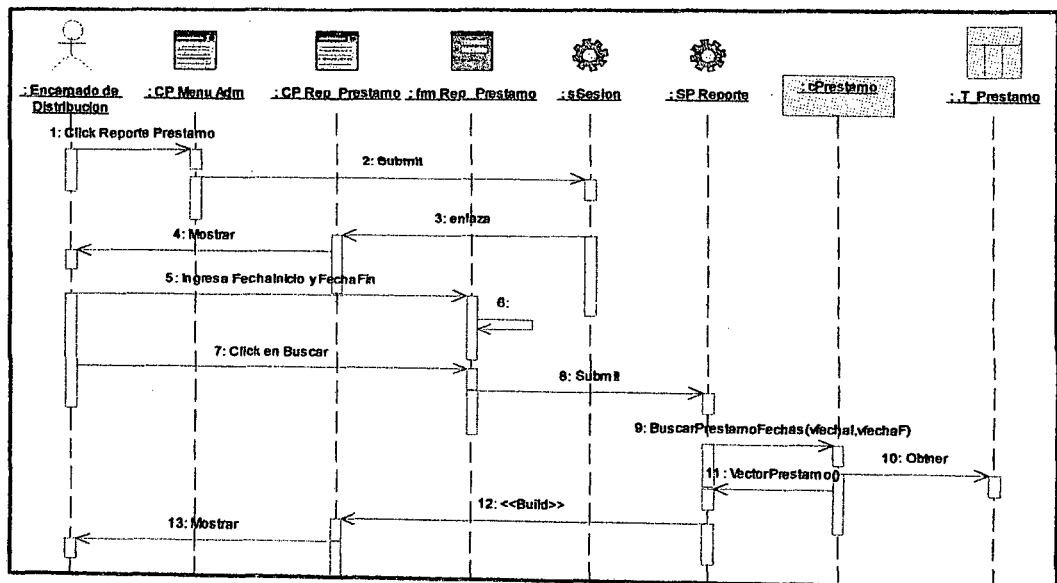


Gráfico 69: Interfaz Reporte Estadístico de Visitas Mensuales por Escuela

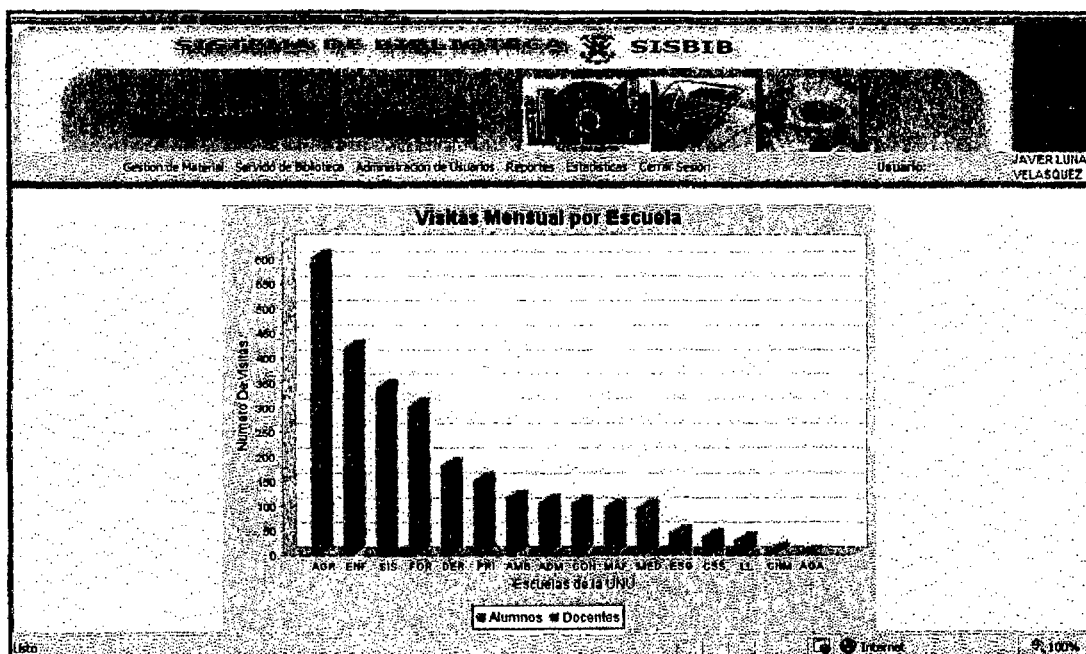


Gráfico 70: Diagrama de Secuencia Reporte Estadístico de Visitas Mensuales por Escuela

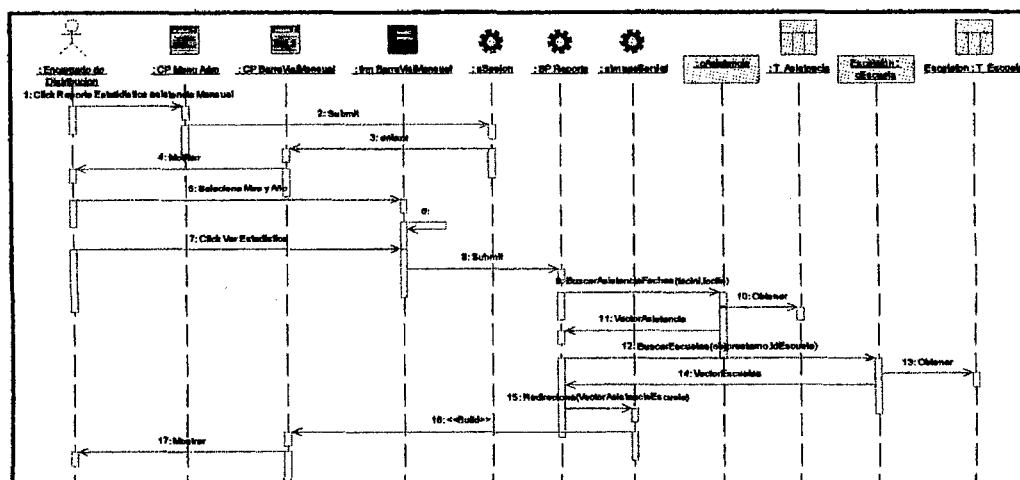


Gráfico 71: Interfaz Reporte Estadístico de Variación De Prestamos

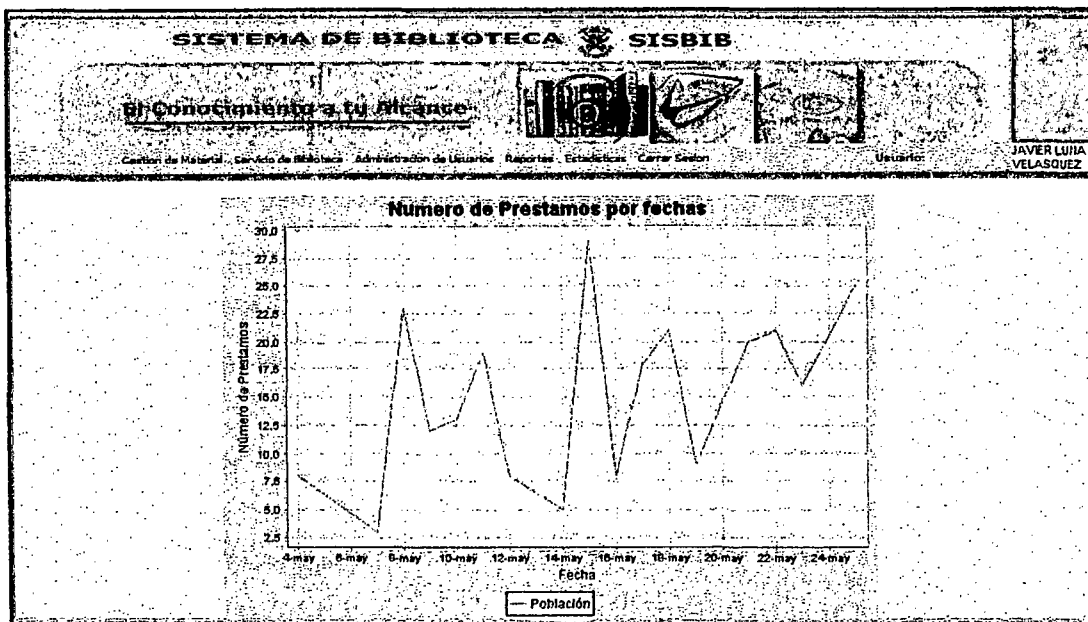


Gráfico 72: Diagrama de Secuencia Reporte Estadístico de Variación de Prestamos

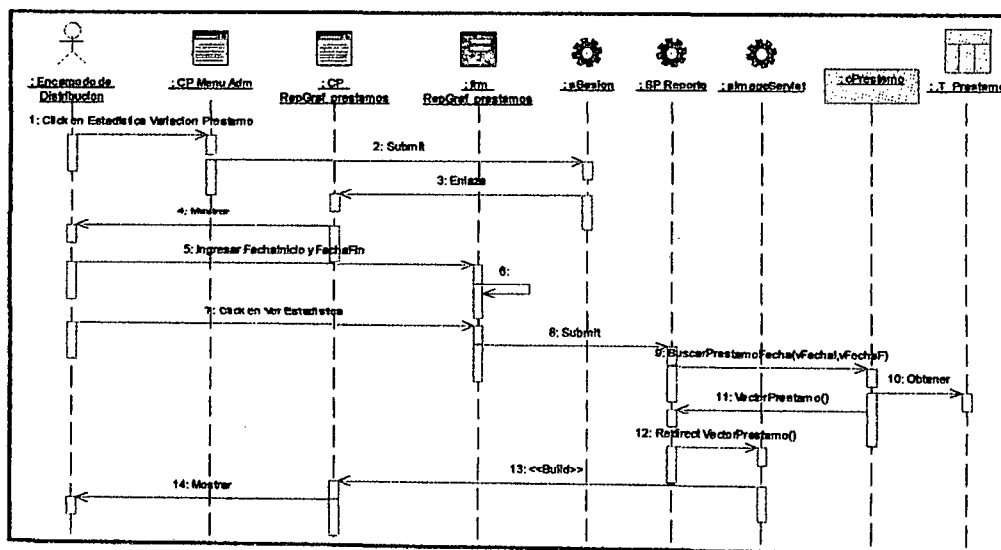


Gráfico 73: Interfaz Reporte Estadístico de Prestamos por Escuelas

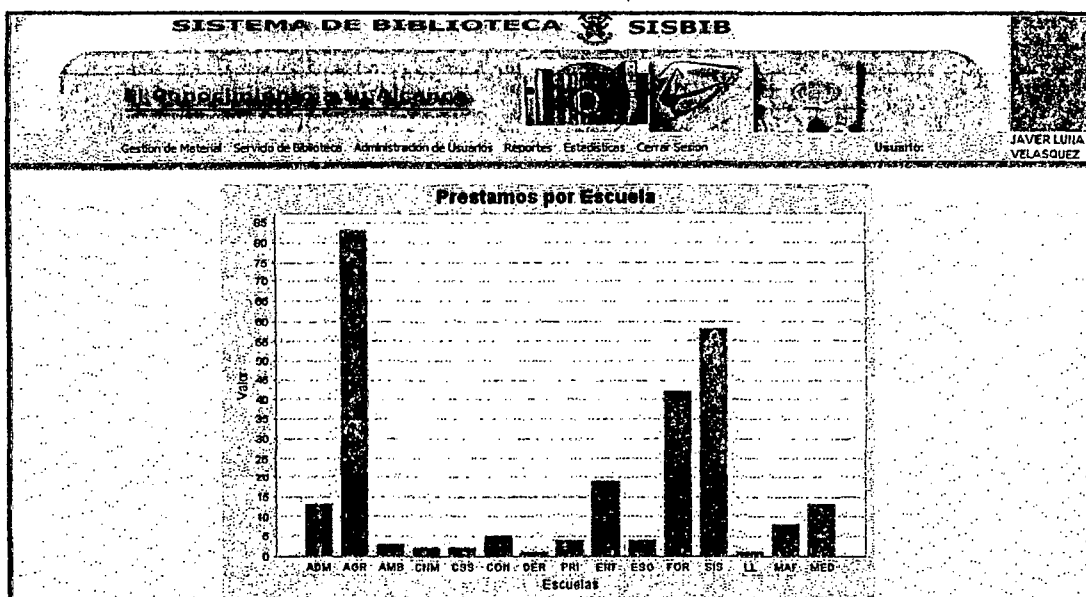
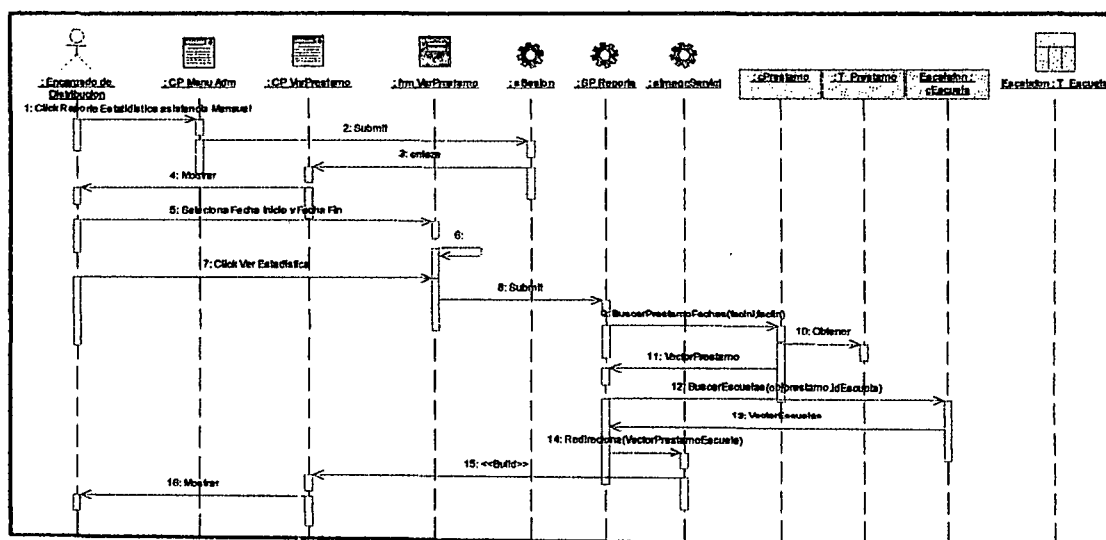


Gráfico 74: Diagrama de Secuencia Reporte Estadístico de Prestamos Por Escuelas



F) Diseño de la Base de Datos

F.1) Diccionario de Datos

Cuadro 28: Diccionario de datos

CLASE	DESCRIPCION
MATERIAL	<ul style="list-style-type: none"> • Sirve para el modelado de los datos de un material bibliográfico existente en la biblioteca. • Sus atributos son: idmaterial, titulo, lug_edi, ano_edi, num_pag, condicion, cod_cla, resumen, fec_ing, descriptores, estado • El campo idmaterial es la clave primaria y será un numero correlativo generado por cada registro de material • El campo lug_edi almacena el lugar donde fue editado el material bibliográfico • El campo ano_edi almacena el año en que fue editado el material • El campo num_pag almacena el numero de paginas del material • El campo cod_cla almacena el código de clasificación según las normas internacionales para cada material • El campo fec_ing almacena la fecha de ingreso del material • El campo condición almacena como fue adquirido el material: COMPRA, DONACION, TRASLADO • El campo estado indica el estado en que se encuentra el material: D = Disponible, R = Reserva, P = Prestado.
PRESTAMO	<ul style="list-style-type: none"> • Sirve para el modelado de los datos de la realización de un préstamo de material. • Sus atributos son: idPrestamo, fec_pres, fec_dev • El campo idPrestamo es la clave primaria • El campo fec_pres almacena la fecha de préstamo. • El campo fec_dev almacena la fecha de devolución
RESERVA	<ul style="list-style-type: none"> • Sirve para el modelado de los datos para la realización de una reserva de un material bibliográfico desde otro lugar fuera de la biblioteca.

	<ul style="list-style-type: none"> • Sus atributos son: idReserva, fec_res, hor_res, estado. • El campo idReserva es la clave primaria • El campo fec_res almacena la fecha de reserva. • El campo hor_res almacena la hora de reserva. • El campo estado indica el estado de la reserva: A= Activo, I=Inactivo.
AUTOR	<ul style="list-style-type: none"> • Sirve para el modelado de los datos de un autor de material. • Sus atributos son: idautor, nombre, ape_pat, ape_mat, pais . • El campo idautor es la clave primaria • El campo ape_pat almacena el apellido paterno. • El campo ape_mat almacena el apellido materno.
EDITORIAL	<ul style="list-style-type: none"> • Sirve para el modelado los datos de una editorial • Sus atributos son: ideditorial, nombre pais • El campo ideditorial es la clave primaria • El campo nombre almacena el nombre de la editorial
MATERIA	<ul style="list-style-type: none"> • Sirve para el modelado los datos de una materia • Sus atributos son: idmateria, descripcion • El campo idmateria es la clave primaria • El campo descripcion almacena la descripción de la materia.
LECTURA	<ul style="list-style-type: none"> • Sirve para el modelado los datos de la lectura de los materiales bibliográficos • Sus atributos son: idlectura, fechalect • El campo idlectura es la clave primaria • El campo idPersona almacena el usuario que registro la lectura del material bibliográfico. • El campo fechalect almacena la fecha de la lectura del material bibliográfico.

G) Diagrama de Estado

Gráfico 75: Diagrama de Estados de Material

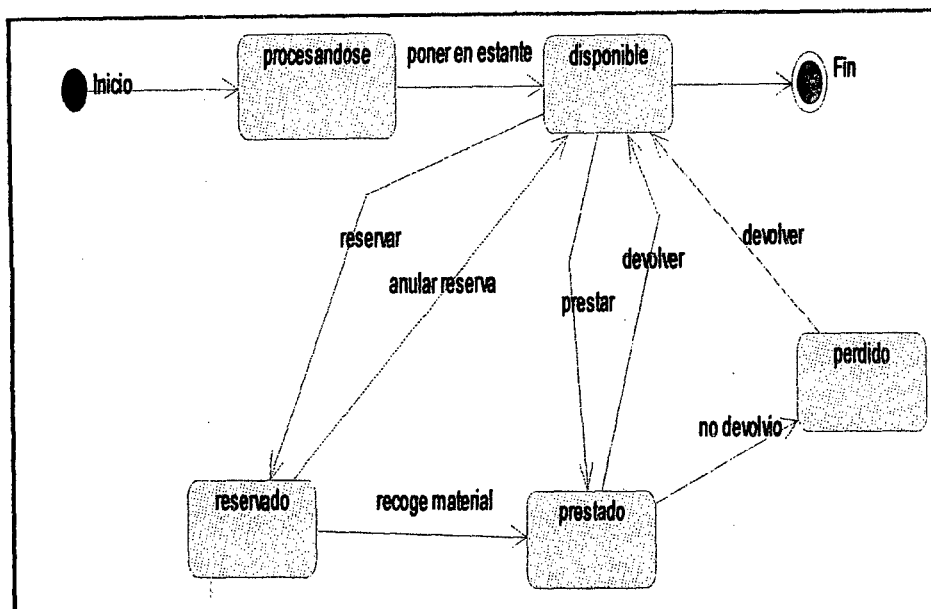


Gráfico 76: Diagrama de Estados de Préstamo

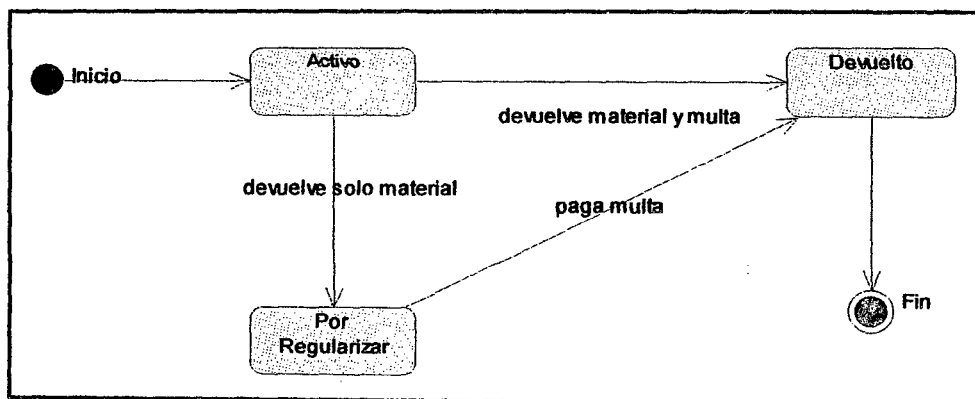


Gráfico 77: Diagrama de Estados Reserva

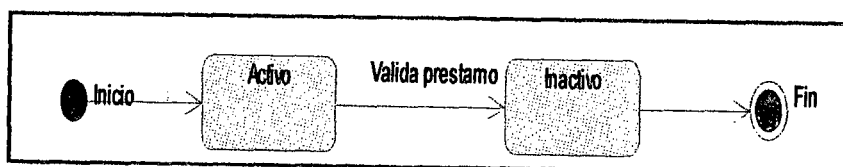
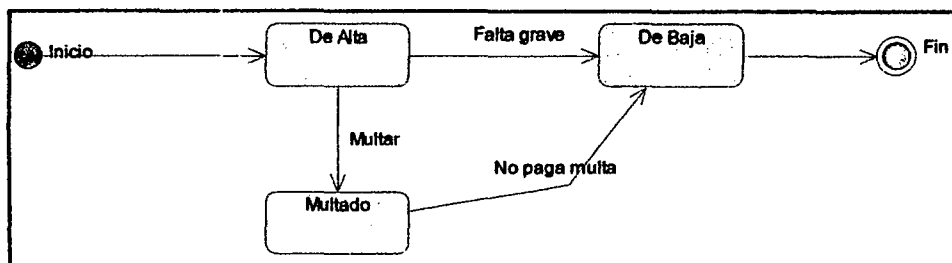
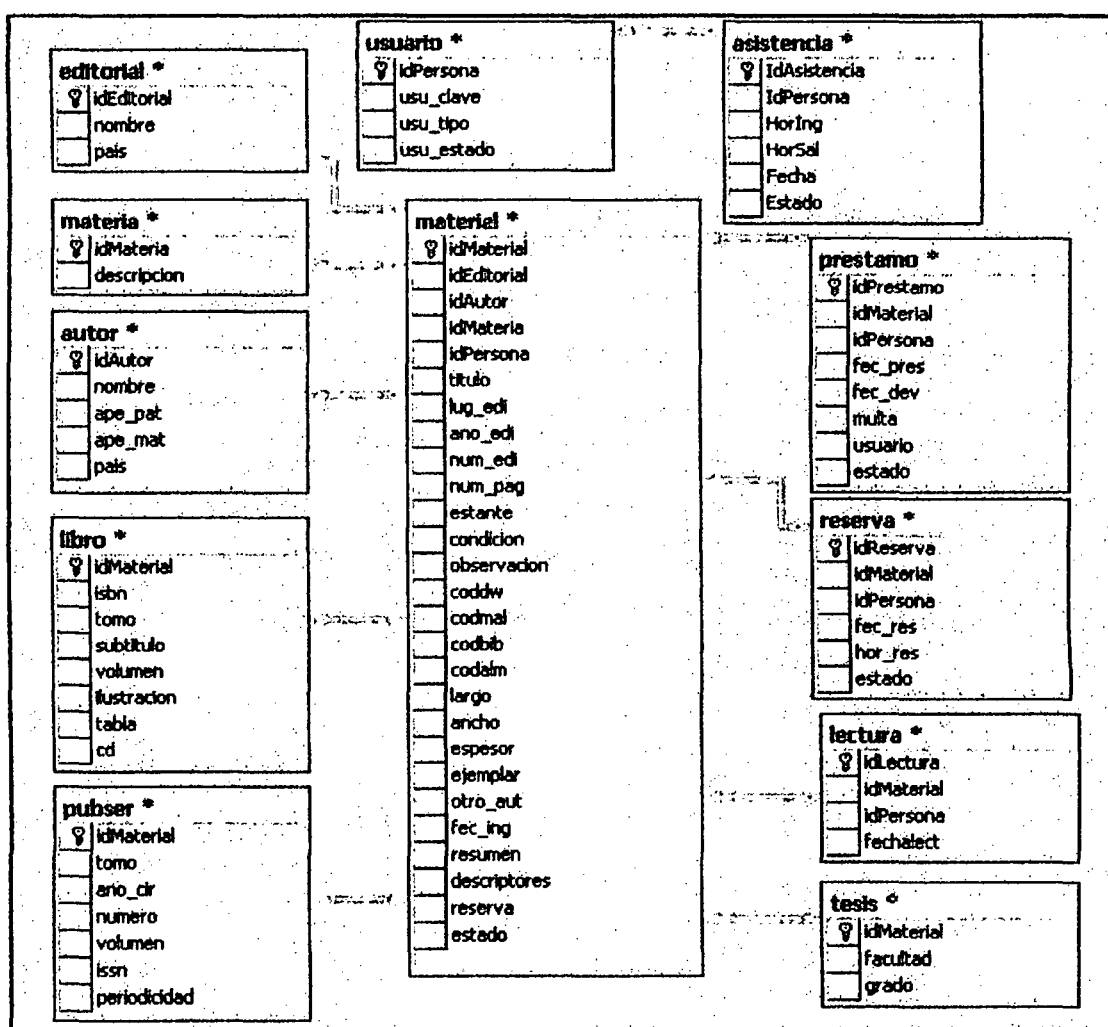


Gráfico 78: Diagrama de Estados Usuario



H) Diagrama de Base de Datos

Gráfico 79: Diagrama de Base de Datos



I) Diagrama Vista Web

Gráfico 80: Diagrama Vista Web Gestión de Material Bibliográfico

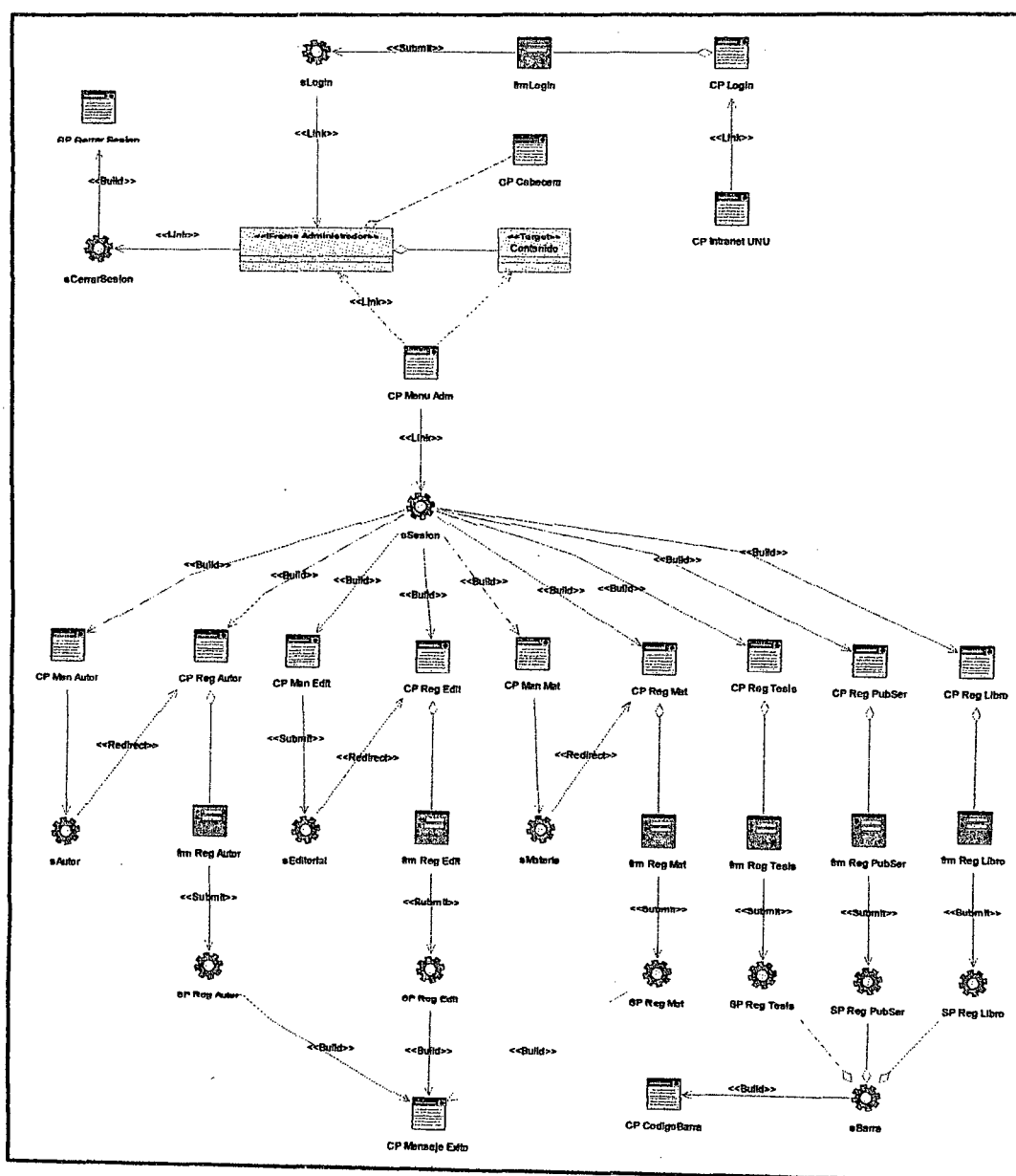


Gráfico 81: Diagrama Vista Web Servicio de Biblioteca

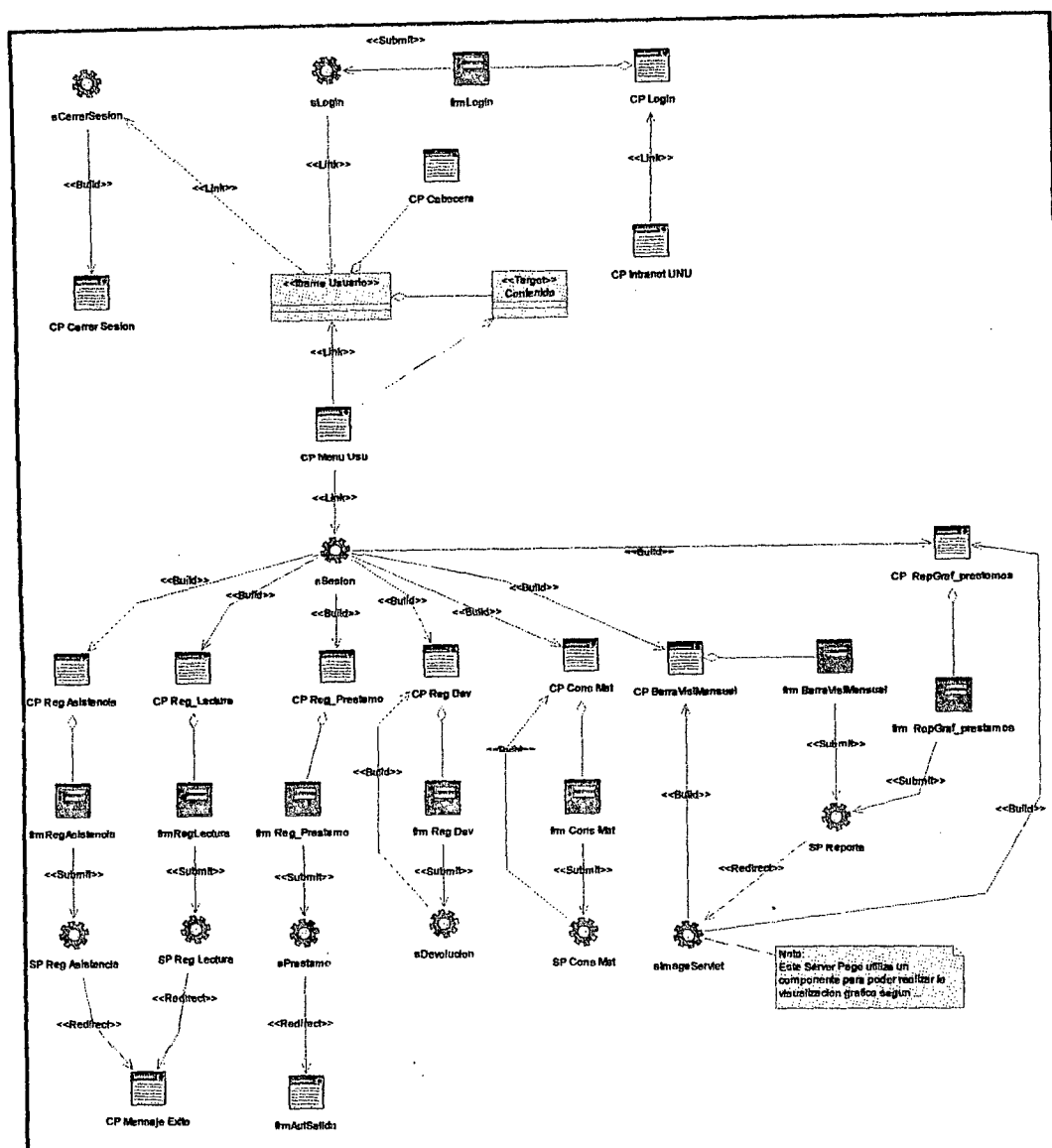
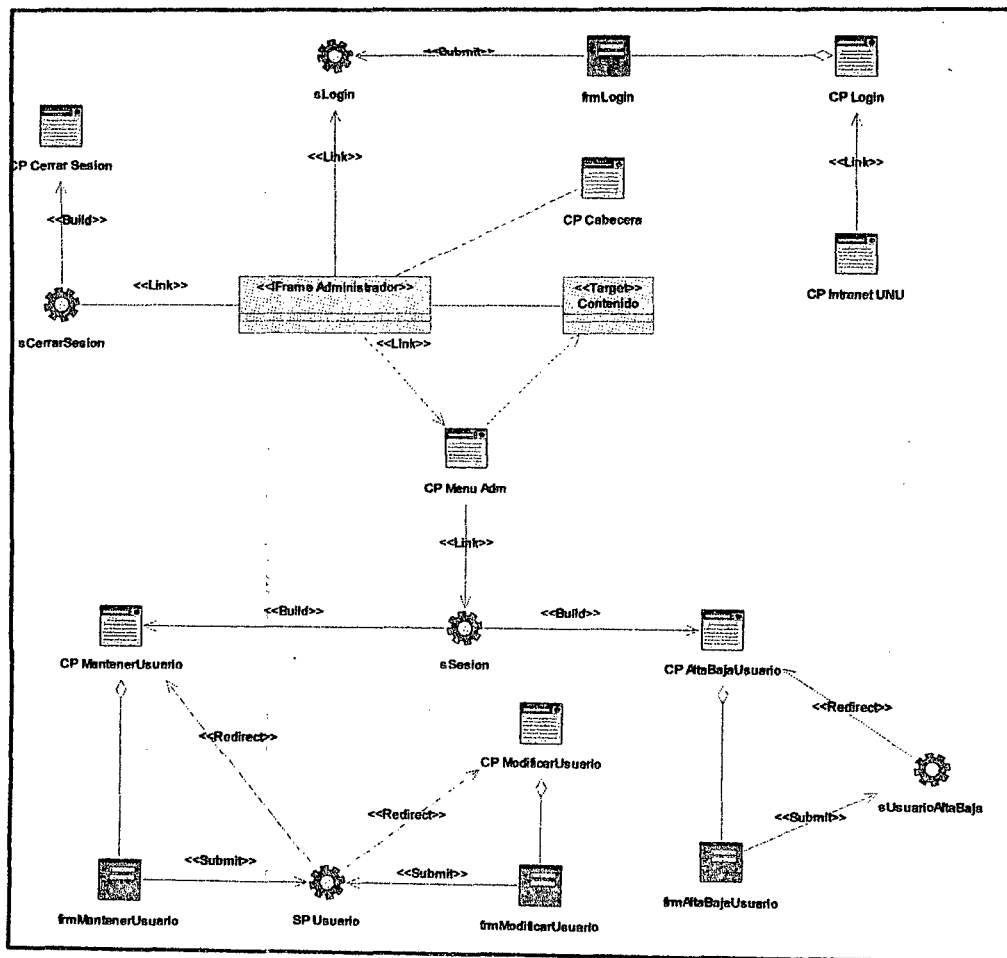
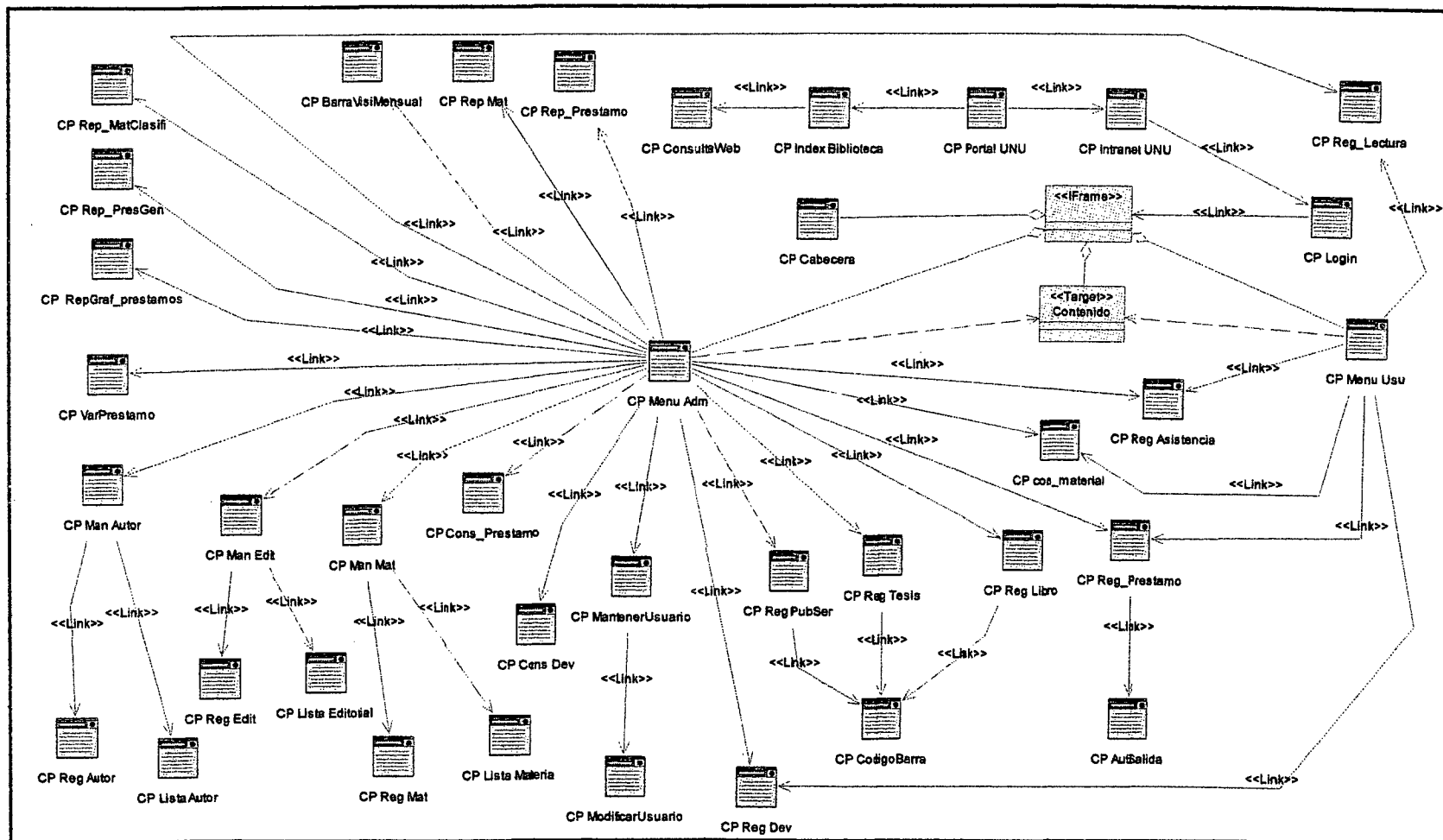


Gráfico 82: Diagrama Vista Web Administración Usuario



J) Mapa de Navegación

Gráfico 83: Diagrama Mapa de Navegación del Sistema



4.3. Fase Construcción

4.3.1. Diagrama de Componentes

Gráfico 84: Diagrama de Componentes

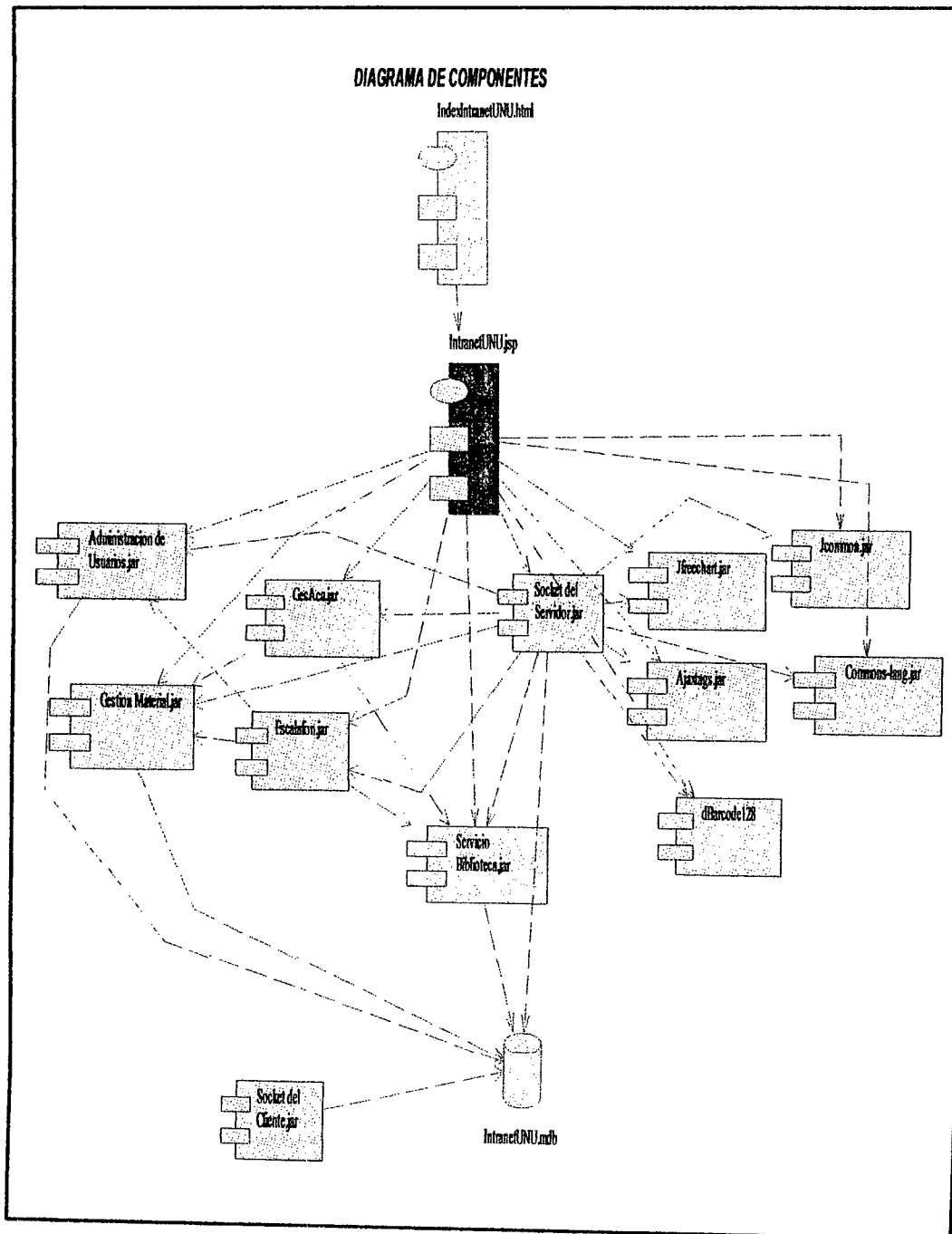


Gráfico 85: Componente GesAca.jar

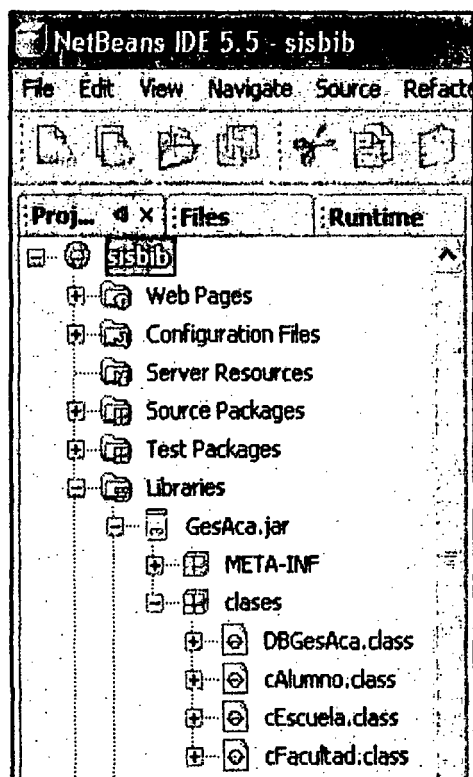


Gráfico 86: Componente Escalafon.jar

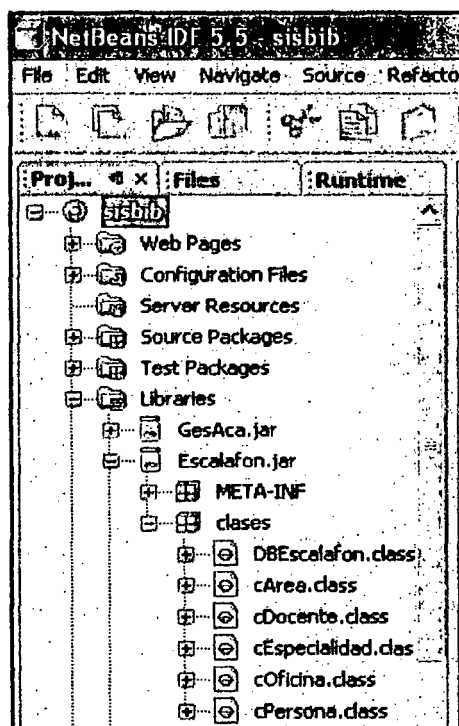


Gráfico 87: Componente GestionMaterial.jar

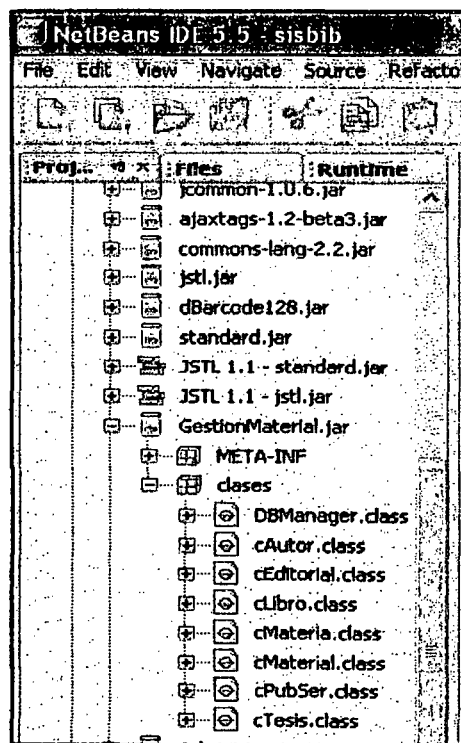


Gráfico 88: Componente ServicioBiblioteca.jar

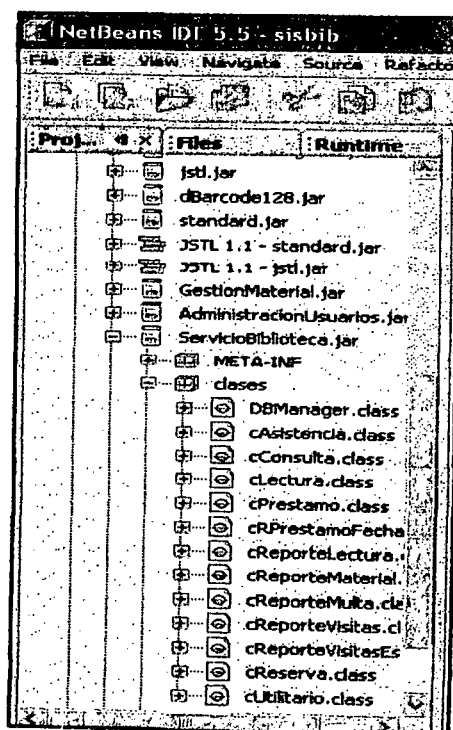


Gráfico 89: Componente Administración-Usuarios.jar

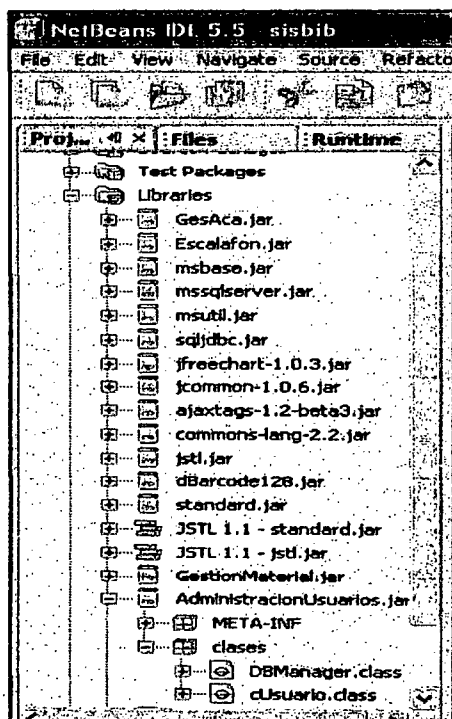


Gráfico 90: Componente dBarcode128.jar

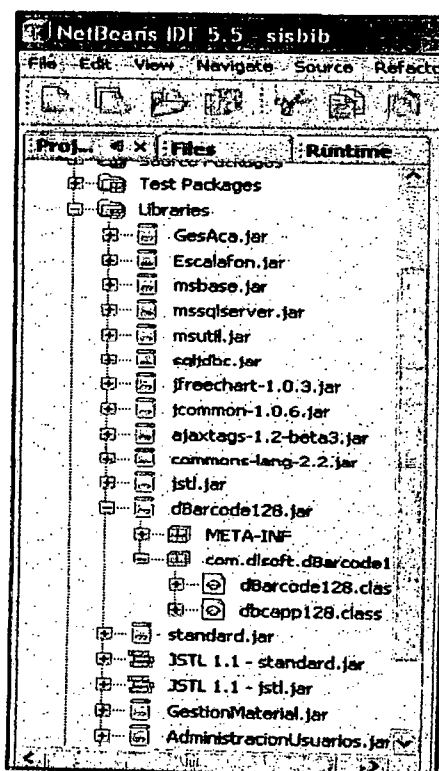


Gráfico 91: Componente jfreechart-1.0.3.jar

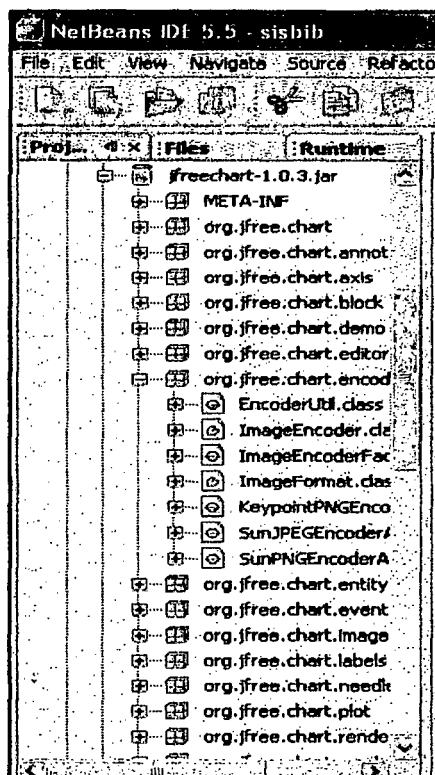
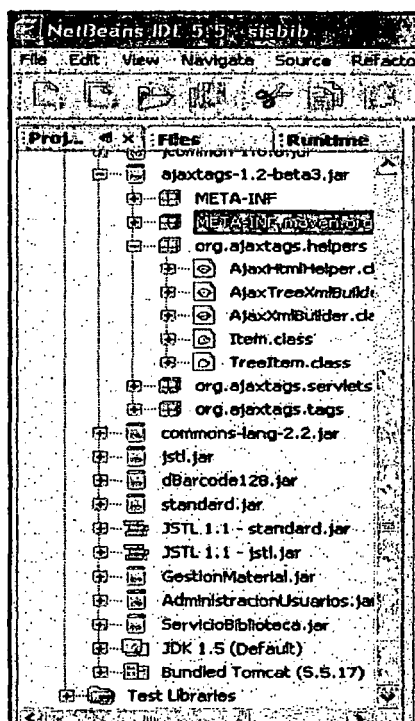
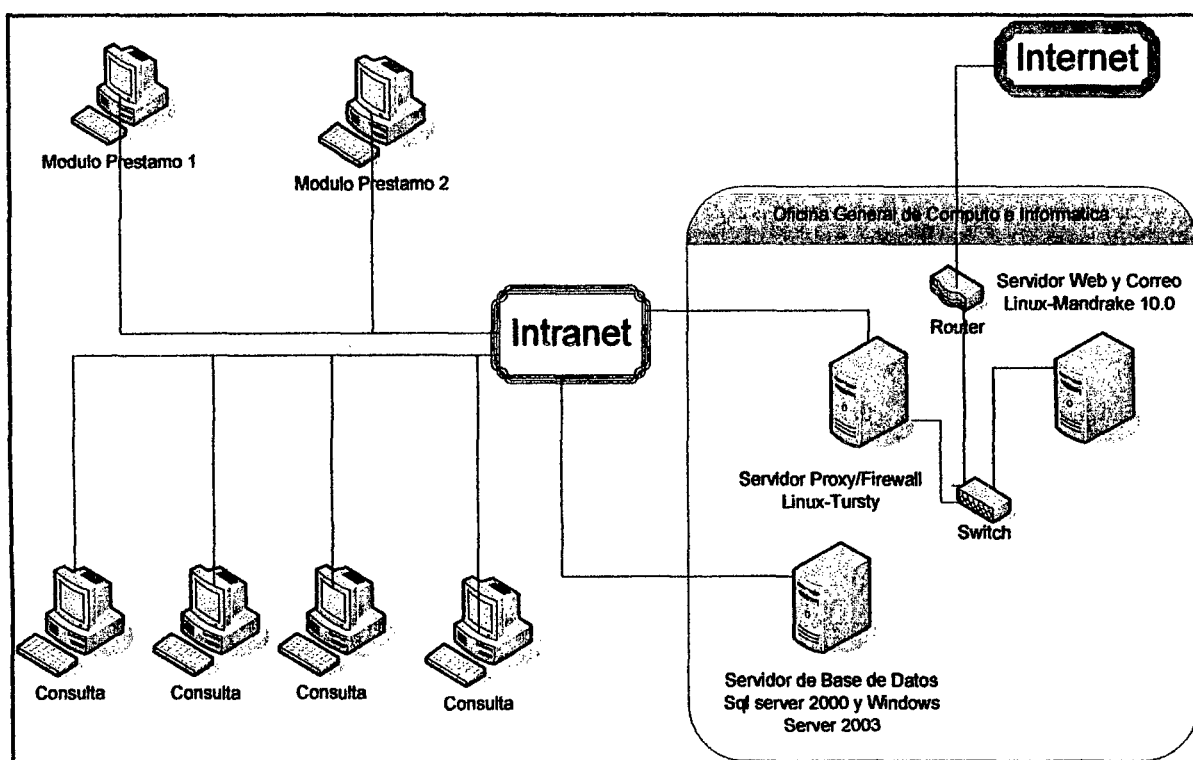


Gráfico 92: Componente ajaxtags1.2.jar



4.3.2. Diagrama de Despliegue Lógico

Gráfico 93: Diagrama de Despliegue Lógico



4.3.3. Sentencias Lógicas

Clase cAutor:

// Importando librerías

```
package clases;
```

```
import java.sql.*;
```

```
import java.util.*;
```

```
// @author SISBIB
```

// Atributos

```
public class cAutor {
```

```
    protected String idAutor;
```

```
    protected String nombre;
```

```
    protected String ape_pat;
```

```
    protected String ape_mat;
```

```
    protected String pais;
```

```
protected static String error;
```

// Constructores de la clase autor

// Constructor Vacio

```
public cAutor() {  
}
```

// Constructor Lleno

```
public cAutor(String idAutor,  
              String nombre,  
              String ape_pat,  
              String ape_mat,  
              String pais)
```

```
{  
this.setIdAutor(idAutor);  
this.setNombre(nombre);  
this.setApe_pat(ape_pat);  
this.setApe_mat(ape_mat);  
this.setPais(pais);  
}
```

// Lógica del Negocio

```
public String GenerarCodigo(){  
    int vcorre =1;  
    String sql,vCeros="";  
    try{  
        DBManager dbm = new DBManager();  
        Connection con = dbm.getConnection();  
        if (con==null)  
            throw new NullPointerException(dbm.geterror());  
        sql ="SELECT idAutor FROM autor order by idAutor";  
        PreparedStatement st = con.prepareStatement(sql,1005,1007);  
        ResultSet rs = st.executeQuery();  
        rs.afterLast();  
        if ( rs.previous() ){  
            vcorre = Integer.parseInt(rs.getString("idAutor"));  
            vcorre++;  
            //System.out.println(vCeros+vcorre);  
        }  
        for(int i=1;i<9-String.valueOf(vcorre).length();i++)  
            vCeros = vCeros + "0";  
    }  
}
```

```

        rs.close();
        st.close();
        con.close();
        System.out.println(vCeros+vcorre);
    }catch(Exception e){e.getMessage();System.out.println(e.getMessage());}
    return (vCeros + vcorre);
}

public boolean Crear(String vidAutor,String vnombre,String vapepat,String vapemat,String
vpais) {
    try{
        error = null;
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spAutor_A(?,?,?,?,?)");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.VARCHAR);
        st.registerOutParameter(3,Types.VARCHAR);
        st.registerOutParameter(4,Types.VARCHAR);
        st.registerOutParameter(5,Types.VARCHAR);
        st.setString(1,vidAutor);
        st.setString(2,vnombre);
        st.setString(3,vapepat);
        st.setString(4,vapemat);
        st.setString(5,vpais);

        if ( st.execute() ) //devuelve verdadero si fallo
            error = "No se puede registrar el autor";
        st.close();
        con.close();
        dbm.closeConnection( con );
    }catch(Exception e){error=e.toString(); System.out.println(e.getMessage()); }
    return (error == null);
}

public boolean Modificar(String vidAutor,String vnombre,String vapepat,String vapemat,String
vpais) {
    try{
        error = null;
        DBManager dbm = new DBManager();

```

```

Connection con = dbm.getConnection();
if ( con == null )
    throw new NullPointerException( dbm.getError() );
CallableStatement st = con.prepareCall("{CALL spAutor_M(?,?,?,?)}");
st.registerOutParameter(1,Types.CHAR);
st.registerOutParameter(2,Types.VARCHAR);
st.registerOutParameter(3,Types.VARCHAR);
st.registerOutParameter(4,Types.VARCHAR);
st.registerOutParameter(5,Types.VARCHAR);

st.setString(1,vidAutor);
st.setString(2,vnombre);
st.setString(3,vapepat);
st.setString(4,vapemat);
st.setString(5,vpais);

if ( st.execute() ) //devuelve verdadero si fallo
    error = "No se puede modificar el autor";
st.close();
con.close();
}catch(Exception e){error=e.toString(); System.out.println(e.getMessage());}
return (error == null);
}

```

```

public boolean Eliminar(String vidAutor){
    try{
        error = null;
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.getError() );
        CallableStatement st = con.prepareCall("{CALL spAutor_E(?) }");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidAutor);
        if ( st.execute() ) //devuelve verdadero si fallo
            error = "No se puede eliminar el autor";
        st.close();
        con.close();
        dbm.closeConnection( con );
    }catch(Exception e){error=e.toString(); }
}

```



```

    return (error == null);
}

public boolean BuscarIdAutor(String vidautor){
    boolean res=false;
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spAutor_TC(?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidautor);
        ResultSet rs = st.executeQuery();
        if ( rs.next() ){
            this.setIdAutor(rs.getString("idAutor"));
            this.setNombre(rs.getString("nombre"));
            this.setApe_pat(rs.getString("ape_pat"));
            this.setApe_mat(rs.getString("ape_mat"));
            this.setPais(rs.getString("pais"));
            res=true;
        }
        rs.close();
        st.close();
        dbm.closeConnection( con );
    }catch(Exception e){e.getMessage();res=false;System.out.println(e.getMessage());}
    return res;
}

public static Vector BuscarTodo(){
    Vector autor = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spAutor_TT}");
        ResultSet rs = st.executeQuery();
        autor = new Vector();
    }
}

```

[illegible]

```

        rs.getString("pais")
    )
    );

    cont++;
} while ( rs.next() && (cont <= numRegPagina));
}

rs.close();
st.close();
dbm.closeConnection( con );
return resultadoautores;
} catch ( Exception e ) { error = e.getMessage(); }
return null;
}

public static Vector BuscarAutorNombrePagina(String vnombre,int numPagina,int
numRegPagina){
    Vector resultadoautores = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spAutor_BN('"+ vnombre
        +"')}",ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
        st.setFetchSize(numRegPagina);
        ResultSet rs = st.executeQuery();

        int fila = numRegPagina * (numPagina - 1)+1 ;
        int cont = 1;

        if (rs.absolute(fila) && numRegPagina > 0) {
            resultadoautores= new Vector();
            do { //System.out.println("DENTRO DEL FOR"+" "+rs.getString("nombre"));
                resultadoautores.add( new cAutor( rs.getString("idAutor"),
                    rs.getString("nombre"),
                    rs.getString("ape_pat"),
                    rs.getString("ape_mat"),

```

```

        rs.getString("pais")
    )
    );

    cont++;
} while ( rs.next() && (cont <= numRegPagina));
}
rs.close();
st.close();
dbm.closeConnection( con );
return resultadoautores;
} catch ( Exception e ) { error = e.getMessage(); }
return null;
}

public static Vector BuscarAutorNombre(String vnombre){
    Vector lista = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.getError() );

        CallableStatement st = con.prepareCall("{CALL spAutor_BN('"+ vnombre +"')}");
        ResultSet rs = st.executeQuery();
        lista = new Vector();
        while ( rs.next() )
            lista.add( new cAutor(rs.getString("idAutor"),
                                   rs.getString("nombre"),
                                   rs.getString("ape_pat"),
                                   rs.getString("ape_mat"),
                                   rs.getString("pais")
                                )
                       );
        rs.close();
        st.close();
        dbm.closeConnection( con );
        return lista;
    } catch ( Exception e ) { error = e.getMessage(); }
    return null;
}

```

```

}
// Encapsulamiento de los atributos
public String getIdAutor() {
    return idAutor;
}
public void setIdAutor(String idAutor) {
    this.idAutor = idAutor;
}
public String getNombre() {
    return nombre;
}
public void setNombre(String nombre) {
    this.nombre = nombre;
}
public String getApe_pat() {
    return ape_pat;
}
public void setApe_pat(String ape_pat) {
    this.ape_pat = ape_pat;
}
public String getApe_mat() {
    return ape_mat;
}
public void setApe_mat(String ape_mat) {
    this.ape_mat = ape_mat;
}
public String getPais() {
    return pais;
}
public void setPais(String pais) {
    this.pais = pais;
}
}
// Fin de la clase Autor

```

Clase cEditorial:

// Importando librerías

```

package clases;
import java.sql.*;

```

```

import java.util.*;
// @author SISBIB
// Atributos
public class cEditorial {
    protected String idEditorial;
    protected String nombre;
    protected String pais;

    protected static String error;
// Constructores de la clase editorial
// Constructor Vacío
    public cEditorial() {
    }
// Constructor lleno
    public cEditorial(String idEditorial,
                      String nombre,
                      String pais) {
        this.setIdEditorial(idEditorial);
        this.setNombre(nombre);
        this.setPais(pais);
    }
// Logica del Negocio
    public String GenerarCodigo(){
        int vcorre =1;
        String sql,vCeros="";
        try{
            DBManager dbm = new DBManager();
            Connection con = dbm.getConnection();
            if (con==null)
                throw new NullPointerException(dbm.geterror());
            sql ="SELECT ideditorial FROM editorial order by ideditorial";
            PreparedStatement st = con.prepareStatement(sql,1005,1007);
            ResultSet rs = st.executeQuery();
            rs.afterLast();
            if ( rs.previous() ){
                vcorre = Integer.parseInt(rs.getString("ideditorial"));
                vcorre++;
            }
            for(int i=1;i<6-String.valueOf(vcorre).length();i++)
                vCeros = vCeros + "0";
        }
    }
}

```

```

        rs.close();
        st.close();
        con.close();
    }catch(Exception e){e.getMessage();System.out.println(e.getMessage());}
    return (vCeros + vcorre);
}

public boolean Crear(String vidEditorial,String vnombre,String vpais) {
    try{
        error = null;
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spEditorial_A(?,?,?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.VARCHAR);
        st.registerOutParameter(3,Types.VARCHAR);

        st.setString(1,vidEditorial);
        st.setString(2,vnombre);
        st.setString(3,vpais);

        if ( st.execute() ) //devuelve verdadero si fallo
            error = "No se puede registrar la editorial";
        st.close();
        con.close();
        dbm.closeConnection( con );
    }catch(Exception e){error=e.toString(); System.out.println(e.getMessage()); }
    return (error == null);
}

public boolean Modificar(String vidEditorial,String vnombre,String vpais) {
    try{
        error = null;
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spEditorial_M(?,?,?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.VARCHAR);

```

```

        st.registerOutParameter(3,Types.VARCHAR);

        st.setString(1,vidEditorial);
        st.setString(2,vnombre);
        st.setString(3,vpais);

        if ( st.execute() ) //devuelve verdadero si fallo
            error = "No se puede modificar la editorial";
        st.close();
        con.close();
    }catch(Exception e){error=e.toString(); System.out.println(e.getMessage());}
    return (error == null);
}

```

```

public boolean Eliminar(String vidEditorial){
    try{
        error = null;
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spEditorial_E(?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidEditorial);
        if ( st.execute() ) //devuelve verdadero si fallo
            error = "No se puede eliminar la editorial";
        st.close();
        con.close();
        dbm.closeConnection( con );
    }catch(Exception e){error=e.toString(); }
    return (error == null);
}

```

```

public boolean BuscarIdEditorial(String videditorial){
    boolean res=false;
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

```



```

        CallableStatement st = con.prepareCall("{CALL spEditorial_TC(?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,videditorial);
        ResultSet rs = st.executeQuery();
        if ( rs.next() ){
            this.setIdEditorial(rs.getString("idEditorial"));
            this.setNombre(rs.getString("nombre"));
            this.setPais(rs.getString("pais"));
            res=true;
        }
        rs.close();
        st.close();
        dbm.closeConnection( con );
    }catch(Exception e){e.getMessage();res=false;System.out.println(e.getMessage());}
    return res;
}

```

```

public static Vector BuscarEditorialNombre(String vnombre){
    Vector lista = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spEditorial_BN('"+ vnombre +"')}");
        ResultSet rs = st.executeQuery();
        lista = new Vector();
        while ( rs.next() )
            lista.add( new cEditorial(rs.getString("idEditorial"),
                                     rs.getString("nombre"),
                                     rs.getString("pais")
                                   )
                      );
        rs.close();
        st.close();
        dbm.closeConnection( con );
    }
    return lista;
}

```

```

    } catch ( Exception e ) { error = e.getMessage(); }
    return null;
}

public static Vector BuscarTodo(){
    Vector editorial = null;
    editorial = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spEditorial_TT}");
        ResultSet rs = st.executeQuery();
        editorial = new Vector();
        while ( rs.next() )
            editorial.add( new cEditorial( rs.getString("idEditorial"),
                                           rs.getString("nombre"),
                                           rs.getString("pais")
                                         )
                          );
        rs.close();
        st.close();
        dbm.closeConnection( con );
        return editorial;
    } catch ( Exception e ) { error = e.getMessage(); }
    return null;
}

public static Vector BuscarTodopaginas(int numPagina,int numRegPagina){
    Vector resultadoeditoriales = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
    }
}

```

```

        CallableStatement st = con.prepareCall("{CALL
spEditorial_TT}",ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
        st.setFetchSize(numRegPagina);
        ResultSet rs = st.executeQuery();

        int fila = numRegPagina * (numPagina - 1)+1 ;
        int cont = 1;

        if (rs.absolute(fila) && numRegPagina > 0) {
            resultadoeditoriales= new Vector();
            do { //System.out.println("DENTRO DEL FOR"+" "+rs.getString("nombre")+" "+fila);
                resultadoeditoriales.add( new cEditorial( rs.getString("idEditorial"),
                                                            rs.getString("nombre"),
                                                            rs.getString("pais")
                                                            )
                );
                cont++;
            } while ( rs.next() && (cont <= numRegPagina));
        }
        rs.close();
        st.close();
        dbm.closeConnection( con );
        return resultadoeditoriales;
    } catch ( Exception e ) { error = e.getMessage(); }
    return null;
}

```

```

public static Vector BuscarEditorialNombrePagina(String vnombre,int numPagina,int
numRegPagina){

```

```

    Vector resultadoeditoriales = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
    }

```

```

        CallableStatement st = con.prepareCall("{CALL spEditorial_BN('"+ vnombre
+"')}",ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);

```

```

    st.setFetchSize(numRegPagina);
    ResultSet rs = st.executeQuery();

    int fila = numRegPagina * (numPagina - 1)+1 ;
    int cont = 1;

    if (rs.absolute(fila) && numRegPagina > 0) {
        resultadoeditoriales= new Vector();
        do { //System.out.println("DENTRO DEL FOR"+" "+rs.getString("nombre"));
            resultadoeditoriales.add( new cEditorial( rs.getString("idEditorial"),
                rs.getString("nombre"),
                rs.getString("pais")
            )
        );
        cont++;
    } while ( rs.next() && (cont <= numRegPagina));
    }
    rs.close();
    st.close();
    dbm.closeConnection( con );
    return resultadoeditoriales;
} catch ( Exception e ) { error = e.getMessage(); }
return null;
}

// Encapsulamiento de atributos
public String getIdEditorial() {
    return idEditorial;
}

public void setIdEditorial(String idEditorial) {
    this.idEditorial = idEditorial;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getPais() {
    return pais;
}

```

```

    }
    public void setPais(String pais) {
        this.pais = pais;
    }
}
// Fin de la clase Editorial

```

Clase cMateria:

// Importando librerías

```

package clases;
import java.sql.*;
import java.util.*;
// @author SISBIB

```

// Atributos

```

public class cMateria {
    protected String idMateria;
    protected String descripcion;

```

```

    protected static String error;

```

// Constructores de la clase materia

// Constructor Vacío

```

public cMateria() {
}

```

// Constructor lleno

```

public cMateria( String idMateria,
                String descripcion)
{
    this.setIdMateria(idMateria);
    this.setDescripcion(descripcion);
}

```

// Logica del Negocio

```

public String GenerarCodigo(){
    int vcorre =1;
    String sql,vCeros="";
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if (con==null)
            throw new NullPointerException(dbm.geterror());
        sql ="SELECT idMateria FROM materia order by idMateria";

```

```

        PreparedStatement st = con.prepareStatement(sql,1005,1007);
        ResultSet rs = st.executeQuery();
        rs.afterLast();
        if ( rs.previous() ){
            vcorre = Integer.parseInt(rs.getString("idMateria"));
            vcorre++;
        }
        for(int i=1;i<6-String.valueOf(vcorre).length();i++)
            vCeros = vCeros + "0";
        rs.close();
        st.close();
        con.close();
        System.out.println(vCeros+vcorre);
    }catch(Exception e){e.getMessage();System.out.println(e.getMessage());}
    return (vCeros + vcorre);
}

public boolean Crear(String vidMateria,String vdescripcion) {
    try{
        setError(null);
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.getError() );
        CallableStatement st = con.prepareCall("{CALL spMateria_A(?,?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.VARCHAR);

        st.setString(1,vidMateria);
        st.setString(2,vdescripcion);

        if ( st.execute() ) //devuelve verdadero si fallo
            setError("No se puede registrar la Materia");
        st.close();
        con.close();
        dbm.closeConnection( con );
    }catch(Exception e){setError(e.toString()); System.out.println(e.getMessage()); }
    return (getError() == null);
}

```

```

public boolean Modificar(String vidMateria,String vdescripcion) {
    try{
        setError(null);
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spMateria_M(?,?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.VARCHAR);

        st.setString(1,vidMateria);
        st.setString(2,vdescripcion);

        if ( st.execute() ) //devuelve verdadero si fallo
            setError("No se puede modificar la Materia");
        st.close();
        con.close();
    }catch(Exception e){setError(e.toString()); System.out.println(e.getMessage());}
    return (getError() == null);
}

public boolean Eliminar(String vidMateria){
    try{
        setError(null);
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spMateria_E(?) }");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidMateria);
        if ( st.execute() ) //devuelve verdadero si fallo
            setError("No se puede eliminar la Materia");
        st.close();
        con.close();
        dbm.closeConnection( con );
    }catch(Exception e){setError(e.toString()); }
    return (getError() == null);
}

```

```

}
public boolean BuscarIdMateria(String vidmateria){
    boolean res=false;
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spMateria_TC(?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidmateria);
        ResultSet rs = st.executeQuery();
        if ( rs.next() ){
            this.setIdMateria(rs.getString("idMateria"));
            this.setDescripcion(rs.getString("descripcion"));

            res=true;
        }
        rs.close();
        st.close();
        dbm.closeConnection( con );
    }catch(Exception e){e.getMessage();res=false;System.out.println(e.getMessage());}
    return res;
}

public static Vector BuscarMateriaNombre(String vnombre){    //Buscar por nombre de
Oficina
    Vector lista = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spMateria_BN('"+ vnombre +"')}");
        ResultSet rs = st.executeQuery();
        lista = new Vector();
        while ( rs.next() )
            lista.add( new cMateria(rs.getString("idMateria"),

```



```

        rs.getString("descripcion")
    )
    );
    rs.close();
    st.close();
    dbm.closeConnection( con );
    return lista;
} catch ( Exception e ) { error = e.getMessage(); }
return null;
}

public static Vector BuscarTodo(){
    Vector materia = null;
    setError(null);
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spMateria_TT}");

        //CallableStatement st =
con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_O
NLY);
        ResultSet rs = st.executeQuery();
        materia = new Vector();
        while ( rs.next() )
            materia.add( new cMateria( rs.getString("idMateria"),
                rs.getString("descripcion")
            )
        );
        rs.close();
        st.close();
        dbm.closeConnection( con );
        return materia;
    } catch ( Exception e ) { setError(e.getMessage()); }
    return null;
}

public static Vector BuscarTodopaginas(int numPagina,int numRegPagina){

```

```

Vector resultadomaterias = null;
error = null;
try {
    DBManager dbm = new DBManager();
    Connection con = dbm.getConnection();

    if ( con == null )
        throw new NullPointerException( dbm.getError() );

    CallableStatement st = con.prepareCall("{CALL
spMateria_TT}",ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
    st.setFetchSize(numRegPagina);
    ResultSet rs = st.executeQuery();

    int fila = numRegPagina * (numPagina - 1)+1 ;
    int cont = 1;

    if (rs.absolute(fila) && numRegPagina > 0) {
        resultadomaterias= new Vector();
        do { //System.out.println("DENTRO DEL FOR"+" "+rs.getString("nombre")+" "+fila);
            resultadomaterias.add( new cMateria( rs.getString("idMateria"),
                rs.getString("descripcion")
            )
        );
        cont++;
    } while ( rs.next() && (cont <= numRegPagina));
}
rs.close();
st.close();
dbm.closeConnection( con );
return resultadomaterias;
} catch ( Exception e ) { error = e.getMessage(); }
return null;
}

public static Vector BuscarMateriaNombrePagina(String vnombre,int numPagina,int
numRegPagina){
    Vector resultadomaterias = null;
    error = null;
    try {
        DBManager dbm = new DBManager();

```

```

Connection con = dbm.getConnection();

if ( con == null )
    throw new NullPointerException( dbm.getError() );

CallableStatement st = con.prepareCall("{CALL spMateria_BN('"+ vnombre
+ "')}",ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
st.setFetchSize(numRegPagina);
ResultSet rs = st.executeQuery();

int fila = numRegPagina * (numPagina - 1)+1 ;
int cont = 1;

if (rs.absolute(fila) && numRegPagina > 0) {
    resultadomaterias= new Vector();
    do { //System.out.println("DENTRO DEL FOR"+" "+rs.getString("nombre"));
        resultadomaterias.add( new cMateria( rs.getString("idMateria"),
                                             rs.getString("descripcion")
                                             )
                                );
        cont++;
    } while ( rs.next() && (cont <= numRegPagina));
}
rs.close();
st.close();
dbm.closeConnection( con );
return resultadomaterias;
} catch ( Exception e ) { error = e.getMessage(); }
return null;
}

// Encapsulamiento de atributos
public String getIdMateria() {
    return idMateria;
}

public void setIdMateria(String idMateria) {
    this.idMateria = idMateria;
}

public String getDescripcion() {
    return descripcion;
}

```

```

public void setDescription(String descripcion) {
    this.descripcion = descripcion;
}
public static String getError() {
    return error;
}
public static void setError(String aError) {
    error = aError;
}
}
// Fin de la clase materia

```

Clase cMaterial.

```

package clases;
import java.sql.*;
import java.util.*;
// @author SISBIB
// Atributos
public class cMaterial {
    private String idMaterial;
    private String idEditorial;
    private String idAutor;
    private String idMateria;
    private String idPersona;
    private String titulo;
    private String lug_edi;
    private String ano_edi;
    private int num_edi;
    private int num_pag;
    private int estante;
    private String condicion;
    private String observacion;
    private String coddw;
    private String codmal;
    private String codbib;
    private String codalm;
    private int largo;
    private int ancho;
    private int espesor;

```

```

private int ejemplar;
private String otro_aut;
private String fec_ing;
private String resumen;
private String descriptores;
private String reserva;
private String estado;
private static String error;
// Constructores de la clase materia
// Constructor Vacío
public cMaterial() {
}
// Constructor lleno
public cMaterial(String idMaterial,String idEditorial,String idAutor,String idMateria,String
idPersona,String titulo,String lug_edi,String ano_edi,int num_edi,int num_pag,int estante,String
condicion,String observacion,String coddw, String codmal,String codbib,String codalm,int
largo,int ancho, int espesor,int ejemplar,String otro_aut,String fec_ing,String resumen,
String descriptores,String reserva,String estado){
    this.setIdMaterial(idMaterial);
    this.setIdEditorial(idEditorial);
    this.setIdAutor(idAutor);
    this.setIdMateria(idMateria);
    this.setIdPersona(idPersona);
    this.setTitulo(titulo);
    this.setLug_edi(lug_edi);
    this.setAno_edi(ano_edi);
    this.setNum_edi(num_edi);
    this.setNum_pag(num_pag);
    this.setEstante(estante);
    this.setCondicion(condicion);
    this.setObservacion(observacion);
    this.setCoddw(coddw);
    this.setCodmal(codmal);
    this.setCodbib(codbib);
    this.setCodalm(codalm);
    this.setLargo(largo);
    this.setAncho(ancho);
    this.setEspesor(espesor);
    this.setEjemplar(ejemplar);
    this.setOtro_aut(otro_aut);

```

```

this.setFec_ing(fec_ing);
this.setResumen(resumen);
this.setDescriptores(descriptores);
this.setReserva(reserva);
this.setEstado(estado);
}

// Logica del Negocio

public boolean Crear(String idMaterial,String idEditorial,String idAutor,String idMateria,String
idPersona,
    String titulo,String lug_edi,String ano_edi,int num_edi,int num_pag,
    int estante,String condicion,String observacion,String coddw,
    String codmal,String codbib,String codalm,int largo,int ancho,
    int espesor,int ejemplar,String otro_aut,String fec_ing,String resumen,
    String descriptores,String reserva,String estado) {
try{
    setError(null);
    DBManager dbm = new DBManager();
    Connection con = dbm.getConnection();
    if ( con == null )
        throw new NullPointerException( dbm.geterror() );
    CallableStatement st = con.prepareCall("{CALL
spMaterial_A(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}");
    st.registerOutParameter(1,Types.CHAR);
    st.registerOutParameter(2,Types.CHAR);
    st.registerOutParameter(3,Types.CHAR);
    st.registerOutParameter(4,Types.CHAR);
    st.registerOutParameter(5,Types.VARCHAR);
    st.registerOutParameter(6,Types.VARCHAR);
    st.registerOutParameter(7,Types.VARCHAR);
    st.registerOutParameter(8,Types.CHAR);
    st.registerOutParameter(9,Types.INTEGER);
    st.registerOutParameter(10,Types.INTEGER);
    st.registerOutParameter(11,Types.INTEGER);
    st.registerOutParameter(12,Types.CHAR);
    st.registerOutParameter(13,Types.VARCHAR);
    st.registerOutParameter(14,Types.VARCHAR);
    st.registerOutParameter(15,Types.VARCHAR);
    st.registerOutParameter(16,Types.VARCHAR);
    st.registerOutParameter(17,Types.VARCHAR);
    st.registerOutParameter(18,Types.INTEGER);

```

```

st.registerOutParameter(19,Types.INTEGER);
st.registerOutParameter(20,Types.INTEGER);
st.registerOutParameter(21,Types.INTEGER);
st.registerOutParameter(22,Types.VARCHAR);
st.registerOutParameter(23,Types.CHAR);
st.registerOutParameter(24,Types.VARCHAR);
st.registerOutParameter(25,Types.VARCHAR);
st.registerOutParameter(26,Types.CHAR);
st.registerOutParameter(27,Types.CHAR);

```

```

st.setString(1,idMaterial);
st.setString(2,idEditorial);
st.setString(3,idAutor);
st.setString(4,idMateria);
st.setString(5,idPersona);
st.setString(6,titulo);
st.setString(7,lug_edi);
st.setString(8,ano_edi);
st.setInt(9,num_edi);
st.setInt(10,num_pag);
st.setInt(11,estante);
st.setString(12,condicion);
st.setString(13,observacion);
st.setString(14,coddw);
st.setString(15,codmal);
st.setString(16,codbib);
st.setString(17,codalm);
st.setInt(18,largo);
st.setInt(19,ancho);
st.setInt(20,espesor);
st.setInt(21,ejemplar);
st.setString(22,otro_aut);
st.setString(23,fec_ing);
st.setString(24,resumen);
st.setString(25,descriptores);
st.setString(26,reserva);
st.setString(27,estado);

```

```

if ( st.execute() ) //devuelve verdadero si fallo
    setError("No se puede registrar");

```

```

        st.close();
        con.close();
        dbm.closeConnection( con );
    }catch(Exception e){setError(e.toString()); System.out.println(e.getMessage()); }
    return (getError() == null);
}

public boolean Modificar(String idMaterial,String idEditorial,String idAutor,String idMateria,
    String titulo,String lug_edi,String ano_edi,int num_edi,int num_pag,
    int estante,String condicion,String observacion,String coddw,
    String codmal,String codbib,String codalm,int largo,int ancho,
    int espesor,int ejemplar,String otro_aut,String fec_ing,String resumen,
    String descriptors,String reserva,String estado) {
    try{
        setError(null);
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL
spMaterial_M(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.CHAR);
        st.registerOutParameter(3,Types.CHAR);
        st.registerOutParameter(4,Types.CHAR);
        st.registerOutParameter(5,Types.VARCHAR);
        st.registerOutParameter(6,Types.VARCHAR);
        st.registerOutParameter(7,Types.CHAR);
        st.registerOutParameter(8,Types.INTEGER);
        st.registerOutParameter(9,Types.INTEGER);
        st.registerOutParameter(10,Types.INTEGER);
        st.registerOutParameter(11,Types.CHAR);
        st.registerOutParameter(12,Types.VARCHAR);
        st.registerOutParameter(13,Types.VARCHAR);
        st.registerOutParameter(14,Types.VARCHAR);
        st.registerOutParameter(15,Types.VARCHAR);
        st.registerOutParameter(16,Types.VARCHAR);
        st.registerOutParameter(17,Types.INTEGER);
        st.registerOutParameter(18,Types.INTEGER);
        st.registerOutParameter(19,Types.INTEGER);
    }
}

```



```

st.registerOutParameter(20,Types.INTEGER);
st.registerOutParameter(21,Types.VARCHAR);
st.registerOutParameter(22,Types.CHAR);
st.registerOutParameter(23,Types.VARCHAR);
st.registerOutParameter(24,Types.VARCHAR);
st.registerOutParameter(25,Types.CHAR);
st.registerOutParameter(26,Types.CHAR);

```

```

st.setString(1,idMaterial);
st.setString(2,idEditorial);
st.setString(3,idAutor);
st.setString(4,idMateria);
st.setString(5,titulo);
st.setString(6,lug_edi);
st.setString(7,ano_edi);
st.setInt(8,num_edi);
st.setInt(9,num_pag);
st.setInt(10,estante);
st.setString(11,condicion);
st.setString(12,observacion);
st.setString(13,coddw);
st.setString(14,codmal);
st.setString(15,codbib);
st.setString(16,codalm);
st.setInt(17,largo);
st.setInt(18,ancho);
st.setInt(19,espesor);
st.setInt(20,ejemplar);
st.setString(21,otro_aut);
st.setString(22,fec_ing);
st.setString(23,resumen);
st.setString(24,descriptores);
st.setString(25,reserva);
st.setString(26,estado);

```

```

if ( st.execute() ) //devuelve verdadero si fallo
    setError("No se puede modificar");
st.close();
con.close();
dbm.closeConnection( con );

```



```

        rs.getString("idAutor"),
        rs.getString("idMateria"),
        rs.getString("idPersona"),
        rs.getString("titulo"),
        rs.getString("lug_edi"),
        rs.getString("ano_edi"),
        rs.getInt("num_edi"),
        rs.getInt("num_pag"),
        rs.getInt("estante"),
        rs.getString("condicion"),
        rs.getString("observacion"),
        rs.getString("coddw"),
        rs.getString("codmal"),
        rs.getString("codbib"),
        rs.getString("codalm"),
        rs.getInt("largo"),
        rs.getInt("ancho"),
        rs.getInt("espesor"),
        rs.getInt("ejemplar"),
        rs.getString("otro_aut"),
        rs.getString("fec_ing"),
        rs.getString("resumen"),
        rs.getString("descriptores"),
        rs.getString("reserva"),
        rs.getString("estado") )
    );
    rs.close();
    st.close();
    dbm.closeConnection( con );
    return materiales;
} catch ( Exception e ) { setError(e.getMessage()); System.out.println(getError());}
return null;
}

public static Vector BuscarMaterialCodCla(String vcoddw,String vcodmal){
    Vector materiales = null;
    setError(null);
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )

```



```

        rs.close();
        st.close();
        dbm.closeConnection( con );
        return materiales;
    } catch ( Exception e ) { setError(e.getMessage()); System.out.println(getError());}
    return null;
}

```

```

public static Vector BuscarMaterialTipo(String vtipo){
    Vector materiales = null;
    setError(null);
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException(dbm.geterror());

        CallableStatement st = con.prepareCall("{CALL spMaterial_ "+vtipo+"}");

        ResultSet rs = st.executeQuery();
        materiales = new Vector();
        while ( rs.next() )
            materiales.add( new cMaterial(rs.getString("idMaterial"),
                rs.getString("idEditorial"),
                rs.getString("idAutor"),
                rs.getString("idMateria"),
                rs.getString("idPersona"),
                rs.getString("titulo"),
                rs.getString("lug_edi"),
                rs.getString("ano_edi"),
                rs.getInt("num_edi"),
                rs.getInt("num_pag"),
                rs.getInt("estante"),
                rs.getString("condicion"),
                rs.getString("observacion"),
                rs.getString("coddw"),
                rs.getString("codmal"),
                rs.getString("codbib"),
                rs.getString("codalm"),
                rs.getInt("largo"),

```

```

        rs.getInt("ancho"),
        rs.getInt("espesor"),
        rs.getInt("ejemplar"),
        rs.getString("otro_aut"),
        rs.getString("fec_ing"),
        rs.getString("resumen"),
        rs.getString("descriptores"),
        rs.getString("reserva"),
        rs.getString("estado") )

    );

    rs.close();
    st.close();
    dbm.closeConnection( con );
    return materiales;
} catch ( Exception e ) { setError(e.getMessage()); System.out.println(getError());}
return null;
}

```

```

public boolean BuscarIdMaterial(String vidmaterial){
    boolean res=false;
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spMaterial_TC(?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidmaterial);
        ResultSet rs = st.executeQuery();
        if ( rs.next() ){

            this.setIdMaterial(rs.getString("idMaterial"));
            this.setIdEditorial(rs.getString("idEditorial"));
            this.setIdAutor(rs.getString("idAutor"));
            this.setIdMateria(rs.getString("idMateria"));
            this.setIdPersona(rs.getString("idPersona"));
            this.setTitulo(rs.getString("titulo"));
            this.setLug_edi(rs.getString("lug_edi"));
            this.setAno_edi(rs.getString("ano_edi"));
            this.setNum_edi(rs.getInt("num_edi"));

```

```

        this.setNum_pag(rs.getInt("num_pag"));
        this.setEstante(rs.getInt("estante"));
        this.setCondicion(rs.getString("condicion"));
        this.setObservacion(rs.getString("observacion"));
        this.setCoddw(rs.getString("coddw"));
        this.setCodmal(rs.getString("codmal"));
        this.setCodbib(rs.getString("codbib"));
        this.setCodalm(rs.getString("codalm"));
        this.setLargo(rs.getInt("largo"));
        this.setAncho(rs.getInt("ancho"));
        this.setEspesor(rs.getInt("espesor"));
        this.setEjemplar(rs.getInt("ejemplar"));
        this.setOtro_aut(rs.getString("otro_aut"));
        this.setFec_ing(rs.getString("fec_ing"));
        this.setResumen(rs.getString("resumen"));
        this.setDescriptores(rs.getString("descriptores"));
        this.setReserva(rs.getString("reserva"));
        this.setEstado(rs.getString("estado"));
        res=true;
    }
    rs.close();
    st.close();
    dbm.closeConnection( con );
} catch (Exception e){e.getMessage();res=false;System.out.println(e.getMessage());}
return res;
}

```

```

public boolean ReservarMaterial(String vidmaterial,String vestado){
    boolean res=false;
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spMaterial_ME(?,?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.CHAR);
        st.setString(1,vidmaterial);
        st.setString(2,vestado);
    }
}

```

```

        ResultSet rs = st.executeQuery();
        rs.close();
        st.close();
        dbm.closeConnection( con );
    }catch(Exception e){e.getMessage();res=false;System.out.println(e.getMessage());}
    return res;
}

```

```

public boolean PrestarMaterial(String vidmaterial,String vestido){ //metodo de luna
try{

```

```

    setError(null);
    DBManager dbm = new DBManager();
    Connection con = dbm.getConnection();
    if ( con == null )
        throw new NullPointerException( dbm.geterror() );
    CallableStatement st = con.prepareCall("{CALL spMaterial_EP(?,?)}");
    st.registerOutParameter(1,Types.CHAR);
    st.registerOutParameter(2,Types.CHAR);
    st.setString(1,vidmaterial);
    st.setString(2,vestado);

    if ( st.execute() ) //devuelve verdadero si fallo
        setError("No se puede Actualizar el material");
    st.close();
    con.close();
}catch(Exception e){setError(e.toString()); System.out.println(e.getMessage());}
return (getError() == null);
}

```

```

public boolean ActualizarEstadoMaterial(String vidmaterial,String vestido){

```

```

try{
    setError(null);
    DBManager dbm = new DBManager();
    Connection con = dbm.getConnection();
    if ( con == null )
        throw new NullPointerException( dbm.geterror() );
    CallableStatement st = con.prepareCall("{CALL spMaterial_ME(?,?)}");
    st.registerOutParameter(1,Types.CHAR);
    st.registerOutParameter(2,Types.CHAR);
    st.setString(1,vidmaterial);

```



```

        st.setString(2,vestado);
        if ( st.execute() ) //devuelve verdadero si fallo
            setError("No se puede Actualizar El material");

        st.close();
        dbm.closeConnection( con );
    }catch(Exception
e){e.getMessage();setError(e.getMessage());System.out.println(e.getMessage());}
        return (getError()==null);
    }

    public static Vector BuscarMaterialAutor(String vidAutor){
        Vector autores = null;
        setError(null);
        try {
            DBManager dbm = new DBManager();
            Connection con = dbm.getConnection();

            if ( con == null )
                throw new NullPointerException( dbm.geterror() );

            CallableStatement st = con.prepareCall("{CALL spMaterial_TCA(?)}");
            st.registerOutParameter(1,Types.CHAR);
            st.setString(1,vidAutor);
            ResultSet rs = st.executeQuery();
            autores = new Vector();
            while ( rs.next() )
                autores.add( new cMaterial(rs.getString("idMaterial"),
                    rs.getString("idEditorial"),
                    rs.getString("idAutor"),
                    rs.getString("idMateria"),
                    rs.getString("idPersona"),
                    rs.getString("titulo"),
                    rs.getString("lug_edi"),
                    rs.getString("ano_edi"),
                    rs.getInt("num_edi"),
                    rs.getInt("num_pag"),
                    rs.getInt("estante"),
                    rs.getString("condicion"),
                    rs.getString("observacion"),
                    rs.getString("coddw"),

```

```

        rs.getString("codmal"),
        rs.getString("codbib"),
        rs.getString("codalm"),
        rs.getInt("largo"),
        rs.getInt("ancho"),
        rs.getInt("espesor"),
        rs.getInt("ejemplar"),
        rs.getString("otro_aut"),
        rs.getString("fec_ing"),
        rs.getString("resumen"),
        rs.getString("descriptores"),
        rs.getString("reserva"),
        rs.getString("estado") )
    );

    rs.close();
    st.close();
    dbm.closeConnection( con );
    return autores;
} catch ( Exception e ) { setError(e.getMessage()); }
return null;
}

public static Vector BuscarMaterialAutorAltas(String vbuscapor,String vsegun,String viautor){
    Vector materiales = null;
    setError(null);
    //String vtipo=vbuscapor+vsegun;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror());

        CallableStatement st = con.prepareCall("{CALL
spMaterial_T"+vbuscapor+"C"+vsegun+"(?)}");
        st.registerOutParameter(1,Types.VARCHAR);
        st.setString(1,viautor);
        ResultSet rs = st.executeQuery();
        materiales = new Vector();
        while ( rs.next() )
            materiales.add( new cMaterial(rs.getString("idMaterial"),
                rs.getString("idEditorial"),

```

```

        rs.getString("idAutor"),
        rs.getString("idMateria"),
        rs.getString("idPersona"),
        rs.getString("titulo"),
        rs.getString("lug_edi"),
        rs.getString("ano_edi"),
        rs.getInt("num_edi"),
        rs.getInt("num_pag"),
        rs.getInt("estante"),
        rs.getString("condicion"),
        rs.getString("observacion"),
        rs.getString("coddw"),
        rs.getString("codmal"),
        rs.getString("codbib"),
        rs.getString("codalm"),
        rs.getInt("largo"),
        rs.getInt("ancho"),
        rs.getInt("espesor"),
        rs.getInt("ejemplar"),
        rs.getString("otro_aut"),
        rs.getString("fec_ing"),
        rs.getString("resumen"),
        rs.getString("descriptores"),
        rs.getString("reserva"),
        rs.getString("estado") )

    );

    rs.close();
    st.close();
    dbm.closeConnection( con );
    return materiales;
} catch ( Exception e ) { setError(e.getMessage()); System.out.println(getError());}
return null;
}

public static Vector BuscarMaterialCodClaAlta(String vbuscapor,String vsegun,String
vcoddw,String vcodmal){
    Vector materiales = null;
    setError(null);
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

```



```

        rs.getString("estado" )

    );

    rs.close();
    st.close();
    dbm.closeConnection( con );
    return materiales;
} catch ( Exception e ) { setError(e.getMessage()); System.out.println(getError());}
return null;
}

public boolean BuscarIdMaterialBaja(String vidmaterial){
    boolean res=false;
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spMaterial_TBC(?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidmaterial);
        ResultSet rs = st.executeQuery();
        if ( rs.next() ){

            this.setIdMaterial(rs.getString("idMaterial"));
            this.setIdEditorial(rs.getString("idEditorial"));
            this.setIdAutor(rs.getString("idAutor"));
            this.setIdMateria(rs.getString("idMateria"));
            this.setIdPersona(rs.getString("idPersona"));
            this.setTitulo(rs.getString("titulo"));
            this.setLug_edi(rs.getString("lug_edi"));
            this.setAno_edi(rs.getString("ano_edi"));
            this.setNum_edi(rs.getInt("num_edi"));
            this.setNum_pag(rs.getInt("num_pag"));
            this.setEstante(rs.getInt("estante"));
            this.setCondicion(rs.getString("condicion"));
            this.setObservacion(rs.getString("observacion"));
            this.setCoddw(rs.getString("coddw"));
            this.setCodmal(rs.getString("codmal"));
            this.setCodbib(rs.getString("codbib"));
            this.setCodalm(rs.getString("codalm"));
            this.setLargo(rs.getInt("largo"));

```



```

        rs.getString("ano_edi"),
        rs.getInt("num_edi"),
        rs.getInt("num_pag"),
        rs.getInt("estante"),
        rs.getString("condicion"),
        rs.getString("observacion"),
        rs.getString("coddw"),
        rs.getString("codmal"),
        rs.getString("codbib"),
        rs.getString("codalm"),
        rs.getInt("largo"),
        rs.getInt("ancho"),
        rs.getInt("espesor"),
        rs.getInt("ejemplar"),
        rs.getString("otro_aut"),
        rs.getString("fec_ing"),
        rs.getString("resumen"),
        rs.getString("descriptores"),
        rs.getString("reserva"),
        rs.getString("estado") )

    );

    rs.close();
    st.close();
    dbm.closeConnection( con );
    return materias;
} catch ( Exception e ) { setError(e.getMessage()); }
return null;
}

public static Vector BuscarMaterialEditorial(String vidEditorial){
    Vector editoriales = null;
    setError(null);
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spMaterial_TCE(?)}");
        st.registerOutParameter(1, Types.CHAR);

```

```

st.setString(1,vidEditorial);
ResultSet rs = st.executeQuery();
editoriales = new Vector();
while ( rs.next() )
    editoriales.add( new cMaterial(rs.getString("idMaterial"),
        rs.getString("idEditorial"),
        rs.getString("idAutor"),
        rs.getString("idMateria"),
        rs.getString("idPersona"),
        rs.getString("titulo"),
        rs.getString("lug_edi"),
        rs.getString("ano_edi"),
        rs.getInt("num_edi"),
        rs.getInt("num_pag"),
        rs.getInt("estante"),
        rs.getString("condicion"),
        rs.getString("observacion"),
        rs.getString("coddw"),
        rs.getString("codmal"),
        rs.getString("codbib"),
        rs.getString("codalm"),
        rs.getInt("largo"),
        rs.getInt("ancho"),
        rs.getInt("espesor"),
        rs.getInt("ejemplar"),
        rs.getString("otro_aut"),
        rs.getString("fec_ing"),
        rs.getString("resumen"),
        rs.getString("descriptores"),
        rs.getString("reserva"),
        rs.getString("estado") )
        );
rs.close();
st.close();
dbm.closeConnection( con );
return editoriales;
} catch ( Exception e ) { setError(e.getMessage()); }
return null;
}

```

// Encapsulamiento de atributos


```

public String getIdMaterial() {
    return idMaterial;
}
public void setIdMaterial(String idMaterial) {
    this.idMaterial = idMaterial;
}
public String getIdEditorial() {
    return idEditorial;
}
public void setIdEditorial(String idEditorial) {
    this.idEditorial = idEditorial;
}
public String getIdAutor() {
    return idAutor;
}
public void setIdAutor(String idAutor) {
    this.idAutor = idAutor;
}
public String getIdMateria() {
    return idMateria;
}
public void setIdMateria(String idMateria) {
    this.idMateria = idMateria;
}
public String getTitulo() {
    return titulo;
}
public void setTitulo(String titulo) {
    this.titulo = titulo;
}
public String getLug_edi() {
    return lug_edi;
}
public void setLug_edi(String lug_edi) {
    this.lug_edi = lug_edi;
}
public String getAno_edi() {
    return ano_edi;
}
public void setAno_edi(String ano_edi) {

```

```
        this.ano_edi = ano_edi;
    }
    public int getNum_edi() {
        return num_edi;
    }
    public void setNum_edi(int num_edi) {
        this.num_edi = num_edi;
    }
    public int getNum_pag() {
        return num_pag;
    }
    public void setNum_pag(int num_pag) {
        this.num_pag = num_pag;
    }
    public int getEstante() {
        return estante;
    }
    public void setEstante(int estante) {
        this.estante = estante;
    }
    public String getCondicion() {
        return condicion;
    }
    public void setCondicion(String condicion) {
        this.condicion = condicion;
    }
    public String getObservacion() {
        return observacion;
    }
    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }
    public String getCoddw() {
        return coddw;
    }
    public void setCoddw(String coddw) {
        this.coddw = coddw;
    }
    public String getCodmal() {
        return codmal;
    }
```

```
}  
public void setCodmal(String codmal) {  
    this.codmal = codmal;  
}  
public String getCodbib() {  
    return codbib;  
}  
public void setCodbib(String codbib) {  
    this.codbib = codbib;  
}  
public String getCodalm() {  
    return codalm;  
}  
public void setCodalm(String codalm) {  
    this.codalm = codalm;  
}  
public int getLargo() {  
    return largo;  
}  
public void setLargo(int largo) {  
    this.largo = largo;  
}  
public int getAncho() {  
    return ancho;  
}  
public void setAncho(int ancho) {  
    this.ancho = ancho;  
}  
public int getEspesor() {  
    return espesor;  
}  
public void setEspesor(int espesor) {  
    this.espesor = espesor;  
}  
public int getEjemplar() {  
    return ejemplar;  
}  
public void setEjemplar(int ejemplar) {  
    this.ejemplar = ejemplar;  
}
```

```
public String getOtro_aut() {  
    return otro_aut;  
}  
public void setOtro_aut(String otro_aut) {  
    this.otro_aut = otro_aut;  
}  
public String getFec_ing() {  
    return fec_ing;  
}  
public void setFec_ing(String fec_ing) {  
    this.fec_ing = fec_ing;  
}  
public String getResumen() {  
    return resumen;  
}  
public void setResumen(String resumen) {  
    this.resumen = resumen;  
}  
public String getDescriptores() {  
    return descriptores;  
}  
public void setDescriptores(String descriptores) {  
    this.descriptores = descriptores;  
}  
public String getReserva() {  
    return reserva;  
}  
public void setReserva(String reserva) {  
    this.reserva = reserva;  
}  
public String getEstado() {  
    return estado;  
}  
public void setEstado(String estado) {  
    this.estado = estado;  
}  
public static String getError() {  
    return error;  
}  
public static void setError(String aError) {
```

```

        error = aError;
    }
    public String getIdPersona() {
        return idPersona;
    }
    public void setIdPersona(String idPersona) {
        this.idPersona = idPersona;
    }
}

```

// Fin de la clase Material

Clase cPrestamo.

```

package clases;
import java.sql.*;
import java.util.*;
import java.util.Date;
import java.text.*;
// @author SISBIB
// Atributos
public class cPrestamo {
    private String idPrestamo;
    private String idMaterial;
    private String idPersona;
    private String fec_pres;
    private String fec_dev;
    private int multa;
    private String usuario;
    private String estado;
    public static String error;
// Constructores de la clase materia
// Constructor Vacío
    public cPrestamo() {
    }
// Constructor lleno
    public cPrestamo(String idPrestamo,String idMaterial,String idPersona,String fec_pres,String
fec_dev, int multa,String usuario,String estado) {
        this.setIdPrestamo(idPrestamo);
        this.setIdMaterial(idMaterial);
        this.setIdPersona(idPersona);
        this.setFec_pres(fec_pres);
    }
}

```

```

    this.setFec_dev(fec_dev);
    this.setMulta(multa);
    this.setUsuario(usuario);
    this.setEstado(estado);
}
// Logica del Negocio
public boolean BuscarIdPrestamo(String vidprestamo){
    boolean res=false;
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spPrestamo_TC(?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidprestamo);
        ResultSet rs = st.executeQuery();
        if ( rs.next() ){
            this.setIdPrestamo(rs.getString("idPrestamo"));
            this.setIdMaterial(rs.getString("idMaterial"));
            this.setIdPersona(rs.getString("idPersona"));
            this.setFec_pres(rs.getString("fec_pres"));
            this.setFec_dev(rs.getString("fec_dev"));
            this.setMulta(rs.getInt("multa"));
            this.setUsuario(rs.getString("usuario"));
            this.setEstado( rs.getString("estado"));

        }
        rs.close();
        st.close();
        dbm.closeConnection( con );
    }catch(Exception e){e.getMessage();res=false;System.out.println(e.getMessage());}
    return res;
}

public boolean BuscarPrestamo(String vidprestamo,String vidmaterial){
    boolean res=false;
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

```

```

if ( con == null )
    throw new NullPointerException( dbm.geterror() );
CallableStatement st = con.prepareCall("{CALL spPrestamo_TCM(?,?)}");
st.registerOutParameter(1,Types.CHAR);
st.registerOutParameter(2,Types.CHAR);
st.setString(1,vidprestamo);
st.setString(2,vidmaterial);
ResultSet rs = st.executeQuery();
if ( rs.next() ){
    this.setIdPrestamo(rs.getString("idPrestamo"));
    this.setIdMaterial(rs.getString("idMaterial"));
    this.setIdPersona(rs.getString("idPersona"));
    this.setFec_pres(rs.getString("fec_pres"));
    this.setFec_dev(rs.getString("fec_dev"));
    this.setMulta(rs.getInt("multa"));
    this.setUsuario(rs.getString("usuario"));
    this.setEstado( rs.getString("estado"));
    res=true;
}
rs.close();
st.close();
dbm.closeConnection( con );
}catch(Exception e){e.getMessage();res=false;System.out.println(e.getMessage());}
return res;
}

```

```

public boolean Crear(String vidPrestamo,String vidMaterial,String vidPersona,String
vfec_pres,String vfec_dev, int vmulta,String vusuario,String vestado){

```

```

try{
    error = null;
    DBManager dbm = new DBManager();
    Connection con = dbm.getConnection();
    if ( con == null )
        throw new NullPointerException( dbm.geterror() );
    CallableStatement st = con.prepareCall("{CALL spPrestamo_A(?,?,?,?,?,?,?)}");
    st.registerOutParameter(1,Types.CHAR);
    st.registerOutParameter(2,Types.CHAR);
    st.registerOutParameter(3,Types.CHAR);
    st.registerOutParameter(4,Types.CHAR);
    st.registerOutParameter(5,Types.CHAR);
    st.registerOutParameter(6,Types.INTEGER);

```

```

st.registerOutParameter(7,Types.CHAR);
st.registerOutParameter(8,Types.CHAR);
st.setString(1,vidPrestamo);
st.setString(2,vidMaterial);
st.setString(3,vidPersona);
st.setString(4,vfec_pres);
st.setString(5,vfec_dev);
st.setInt(6,vmulta);
st.setString(7,vusuario);
st.setString(8,vestado);

```

```

if ( st.execute() ) //devuelve verdadero si fallo
error = "No se puede registrar el prestamo";

```

```

st.close();
con.close();
dbm.closeConnection( con );
}catch(Exception e){ error=e.toString(); }
return (error == null);
}

```

```

public String GenerarCodigo(String vfecha){
    int vcorre =1;
    String sql,vceros="";
    String dia=vfecha.substring(0,2);
    String mes=vfecha.substring(3,5);
    String ano=vfecha.substring(8);
    try{
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if (con==null)
            throw new NullPointerException(dbm.getError());
        sql ="SELECT IdPrestamo FROM prestamo where fec_pres=? order by IdPrestamo";
        PreparedStatement st = con.prepareStatement(sql,1005,1007);
        long date=new SimpleDateFormat("dd/MM/yyyy").parse(vfecha,new
ParsePosition(0)).getTime();
        java.sql.Date dbDate=new java.sql.Date(date);
        st.setDate(1, (java.sql.Date) dbDate);
        ResultSet rs = st.executeQuery();
        rs.afterLast();
    }
}

```



```

        if ( rs.previous() ){
            vcorre = Integer.parseInt(rs.getString("IdPrestamo").substring(6));
            vcorre++;
        }

        for(int i=1;i<5-String.valueOf(vcorre).length();i++)
            vceros = vceros + "0";

        rs.close();
        st.close();
        con.close();
    }catch(Exception e){e.getMessage();System.out.println(e.getMessage());}

    return ( dia+mes+ano+vceros+vcorre);
}

```

```
public Vector BuscarUsuario(String vcodusu,String vestado){
```

Vector prestamos = null;

error = null;

try {

DBManager dbm = new DBManager();

Connection con = dbm.getConnection();

if (con == null)

throw new NullPointerException(dbm.geterror());

CallableStatement st = con.prepareCall("{CALL spPrestamo_TU(? ,?)}");

st.registerOutParameter(1,Types.CHAR);

st.registerOutParameter(2,Types.CHAR);

st.setString(1,vcodusu);

st.setString(2,vestado);

ResultSet rs = st.executeQuery();

prestamos = new Vector();

while (rs.next())

prestamos.add(new cPrestamo(rs.getString("idPrestamo"),

rs.getString("idMaterial"),

rs.getString("idPersona"),

rs.getString("fec_pres"),

rs.getString("fec_dev"),

rs.getInt("multa"),

rs.getString("usuario"),

rs.getString("estado")

```

        )
    );
    rs.close();
    st.close();
    dbm.closeConnection( con );
    return prestamos;
} catch ( Exception e ) { error = e.getMessage(); }
return null;
}

```

```

public Vector BuscarPrestamo(String vidPersona){    //Buscar por codigo de Persona
    Vector prestamos = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spPrestamo_TP(?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.setString(1,vidPersona);
        ResultSet rs = st.executeQuery();
        prestamos = new Vector();
        while ( rs.next() )
            prestamos.add( new cPrestamo( rs.getString("idPrestamo"),
                                           rs.getString("idMaterial"),
                                           rs.getString("idPersona"),
                                           rs.getString("fec_pres"),
                                           rs.getString("fec_dev"),
                                           rs.getInt("multa"),
                                           rs.getString("usuario"),
                                           rs.getString("estado")
                                           )
            );
        rs.close();
        st.close();
        dbm.closeConnection( con );
        return prestamos;
    }
}

```

```

    } catch ( Exception e ) { error = e.getMessage(); }
    return null;
}

```

**public Vector BuscarPrestFechas(String vidPersona,String vfectini,String vfectfin){ //Buscar
por codigo de Persona y entre fechas**

```
Vector prestamos = null;
```

```
error = null;
```

```
try {
```

```
    DBManager dbm = new DBManager();
```

```
    Connection con = dbm.getConnection();
```

```
    if ( con == null )
```

```
        throw new NullPointerException( dbm.geterror() );
```

```
    CallableStatement st = con.prepareCall("{CALL spPrestamo_TUF(?,?,?)}");
```

```
    st.registerOutParameter(1,Types.CHAR);
```

```
    st.registerOutParameter(2,Types.CHAR);
```

```
    st.registerOutParameter(3,Types.CHAR);
```

```
    st.setString(1,vidPersona);
```

```
    st.setString(2,vfectini);
```

```
    st.setString(3,vfectfin);
```

```
    ResultSet rs = st.executeQuery();
```

```
    prestamos = new Vector();
```

```
    while ( rs.next() )
```

```
        prestamos.add( new cPrestamo( rs.getString("idPrestamo"),
```

```
                                     rs.getString("idMaterial"),
```

```
                                     rs.getString("idPersona"),
```

```
                                     rs.getString("fec_pres"),
```

```
                                     rs.getString("fec_dev"),
```

```
                                     rs.getInt("multa"),
```

```
                                     rs.getString("usuario"),
```

```
                                     rs.getString("estado")
```

```
                                )
```

```
        );
```

```
    rs.close();
```

```
    st.close();
```

```
    dbm.closeConnection( con );
```

```
    return prestamos;
```

```

} catch ( Exception e ) { error = e.getMessage(); }

```

```

    return null;
}

public Vector BuscarPrestUsu(String vidusuario,String vfectini,String vfectfin){ //Buscar por
codigo de Persona y entre fechas
    Vector prestamos = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spPrestamo_TPUS(?,?,?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.CHAR);
        st.registerOutParameter(3,Types.CHAR);
        st.setString(1,vidusuario);
        st.setString(2,vfectini);
        st.setString(3,vfectfin);
        ResultSet rs = st.executeQuery();
        prestamos = new Vector();
        while ( rs.next() )
            prestamos.add( new cPrestamo( rs.getString("idPrestamo"),
                                           rs.getString("idMaterial"),
                                           rs.getString("idPersona"),
                                           rs.getString("fec_pres"),
                                           rs.getString("fec_dev"),
                                           rs.getInt("multa"),
                                           rs.getString("usuario"),
                                           rs.getString("estado")
                                           )
            );
        rs.close();
        st.close();
        dbm.closeConnection( con );
        return prestamos;
    } catch ( Exception e ) { error = e.getMessage(); }
    return null;
}

```

```

}

public Vector BuscarPrestFechas(String vfechini,String vfechfin){    //Busca entre fechas
    Vector prestamos = null;
    error = null;
    try {
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();

        if ( con == null )
            throw new NullPointerException( dbm.geterror() );

        CallableStatement st = con.prepareCall("{CALL spPrestamo_TPFE(?,?)");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.CHAR);

        st.setString(1,vfechini);
        st.setString(2,vfechfin);
        ResultSet rs = st.executeQuery();
        prestamos = new Vector();
        while ( rs.next() )
            prestamos.add( new cPrestamo(  rs.getString("idPrestamo"),
                                            rs.getString("idMaterial"),
                                            rs.getString("idPersona"),
                                            rs.getString("fec_pres"),
                                            rs.getString("fec_dev"),
                                            rs.getInt("multa"),
                                            rs.getString("usuario"),
                                            rs.getString("estado")
                                            )
                            );
        rs.close();
        st.close();
        dbm.closeConnection( con );
        return prestamos;
    } catch ( Exception e ) { error = e.getMessage(); }
    return null;
}

public int CalcularMulta(String vfech_pres,int numdias){

```

```

int dp, mp, ap, dd, md, ad, dias=0;
int sumaP=0;
int sumaD=0;
int sumaA=0;
int multa=0;
int nroDomin=0;
Date hoy = new Date();
cPrestamo objp=new cPrestamo();
// fecha devolucion
dd=hoy.getDate();
md=hoy.getMonth()+1;
ad=hoy.getYear()+1900;

// fecha prestamo
dp = Integer.parseInt(vfec_pres.substring(0,2));
mp = Integer.parseInt(vfec_pres.substring(3,5));
ap = Integer.parseInt(vfec_pres.substring(6,10));

//calculamos dias por año
if(ap==ad)
{
    for(int i= mp; i<md; i++)
    {
        sumaP+=CalcularDiasMes(i,ap);
    }
}
else
{
    for(int j=ap+1; j<=ad-1; j++)
    {
        if ((j % 4 == 0) && ((j % 100 != 0) || (j % 400 == 0)))
        { sumaA+=364; }
        else
        { sumaA+=366; }
    }

    for(int i= mp; i<=12; i++)
    {
        sumaP+=CalcularDiasMes(i,ap);
    }
}

```

```

        for(int i= 1; i<md; i++)
        {
            sumaD = sumaD + CalcularDiasMes(i,ad);
        }
    }

    nrodomin=objp.CalculaDomingos(vfec_pres);
    dias = sumaA + sumaP - dp + sumaD + dd - nrodomin;

    if(dias>numdias)
    {
        dias -=numdias;
        multa=dias*1;
    }
    else
    {
        multa=0;
    }
    return multa;
}

public int CalcularDias(String vfec_pres){
    int dp, mp, ap, dd, md, ad, dias=0;
    int sumaP=0;
    int sumaD=0;
    int sumaA=0;
    int multa=0;
    Date hoy = new Date();

    // fecha devolucion
    dd=hoy.getDate();
    md=hoy.getMonth()+1;
    ad=hoy.getYear()+1900;

    // fecha prestamo
    dp = Integer.parseInt(vfec_pres.substring(0,2));
    mp = Integer.parseInt(vfec_pres.substring(3,5));
    ap = Integer.parseInt(vfec_pres.substring(6,10));

    //calculamos dias por año

```

```

if(ap==ad)
{
    for(int i= mp; i<md; i++)
    {
        sumaP+=CalcularDiasMes(i,ap);
    }
}
else
{
    for(int j=ap+1; j<=ad-1; j++)
    {
        if ((j % 4 == 0) && ((j % 100 != 0) || (j % 400 == 0)))
            { sumaA+=364; }
        else
            { sumaA+=366; }
    }

    for(int i= mp; i<=12; i++)
    {
        sumaP+=CalcularDiasMes(i,ap);
    }

    for(int i= 1; i<md; i++)
    {
        sumaD = sumaD + CalcularDiasMes(i,ad);
    }
}

dias = sumaA + sumaP - dp + sumaD + dd;

return dias;
}

public int CalculaDomingos(String vfecpres){
    cPrestamo objp=new cPrestamo();
    int numdias=objp.CalcularDias(vfecpres);

    int dp,mp,ap,contdomin=0;
    int nrodia=0;
    Calendar objcal = Calendar.getInstance();

```



```

//sacando datos a la fecha
dp = Integer.parseInt(vfecpres.substring(0,2)); //dia
mp = Integer.parseInt(vfecpres.substring(3,5)); //mes
ap = Integer.parseInt(vfecpres.substring(6,10)); //año
objcal.set(ap,(mp-1),dp); //instanciamos la nueva fecha con el aa//mm-1//dd

for (int i=1;i<=numdias;i++){
    objcal.add(Calendar.DATE,i); //agregamos los dias a la fecha
    nrodia=objcal.get(Calendar.DAY_OF_WEEK);
    if (nrodia==1){
        contdomin++; //contamos los domingos
    }
    objcal.set(ap,(mp-1),dp);
}

return contdomin;
}

public int CalcularDiasMes(int mes, int anio)
{
    int dias=0;

    switch (mes)
    {
        case 1: dias=31; break;
        case 2:
            if ((anio % 4 == 0) && ((anio % 100 != 0) || (anio % 400 == 0)))
            { dias = 27; }
            else
            { dias = 28; }
            break;
        case 3: dias=31; break;
        case 4: dias=30; break;
        case 5: dias=31; break;
        case 6: dias=30; break;
        case 7: dias=31; break;
        case 8: dias=31; break;
        case 9: dias=30; break;
        case 10: dias=31; break;
        case 11: dias=30; break;
    }
}

```

```

        case 12: dias=31; break;
    }

```

```

    return dias;
}

```

```

public boolean Devolucion(String vidPrestamo, String vidPersona, String vfecha, int
vmulta,String vestado) {

```

```

    try{
        error = null;
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )
            throw new NullPointerException( dbm.geterror() );
        CallableStatement st = con.prepareCall("{CALL spPrestamo_D(?,?,?,?,?)}");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.CHAR);
        st.registerOutParameter(3,Types.CHAR);
        st.registerOutParameter(4,Types.INTEGER);
        st.registerOutParameter(5,Types.CHAR);

        st.setString(1,vidPrestamo);
        st.setString(2,vidPersona);
        st.setString(3,vfecha);
        st.setInt(4,vmulta);
        st.setString(5,vestado);
        if ( st.execute() ) //devuelve verdadero si fallo
            error = "No se puede registrar la devolución";
        st.close();
        con.close();
    }catch(Exception e){error=e.toString(); System.out.println(e.getMessage());}
    return (error == null);
}

```

```

public boolean ActualizarMultas(String vidPrestamo, int vmulta) {

```

```

    try{
        error = null;
        DBManager dbm = new DBManager();
        Connection con = dbm.getConnection();
        if ( con == null )

```

```

        throw new NullPointerException( dbm.getError() );
        CallableStatement st = con.prepareCall("{CALL spPrestamo_AM(?,?)");
        st.registerOutParameter(1,Types.CHAR);
        st.registerOutParameter(2,Types.INTEGER);
        st.setString(1,vidPrestamo);
        st.setInt(2,vmulta);
        if ( st.execute() ) //devuelve verdadero si fallo
            error = "No se puede registrar la devolucion";
        st.close();
        con.close();
    }catch(Exception e){error=e.toString(); System.out.println(e.getMessage());}
    return (error == null);
}

```

```

public void BarrerMultas(){
    cPrestamo objpres=new cPrestamo();
    cPrestamo objpr=new cPrestamo();
    cUsuario objusu=new cUsuario();
    int numdias=0;
    Vector vec=new Vector();
    String fec="";
    int multa=0;
    vec= objpr.BuscarPrestamos("P");
    for (int i=0;i<vec.size();i++){
        objpres=(cPrestamo) vec.elementAt(i);
        fec=objpres.getFec_pres();
        fec = fec.substring(8,10)+"/"+fec.substring(5,7)+"/"+fec.substring(0,4);
        objusu.BuscarIdUsuario(objpres.getIdPersona());
        if (objusu.getUsu_tipo().trim().equals("A")) numdias=2;
        if (objusu.getUsu_tipo().trim().equals("T")) numdias=2;
        if (objusu.getUsu_tipo().trim().equals("D")) numdias=5;
        multa=objpr.CalcularMulta(fec,numdias);
        objpr.ActualizarMultas(objpres.getIdPrestamo(),multa);
    }
}

```

```

public Vector BuscarPrestamos(String vestado){
    Vector prestamos = null;
    error = null;
    try {

```

```

DBManager dbm = new DBManager();
Connection con = dbm.getConnection();

if ( con == null )
    throw new NullPointerException( dbm.getError() );

CallableStatement st = con.prepareCall("{CALL spPrestamo_TE(?)}}");
st.registerOutParameter(1,Types.CHAR);
st.setString(1,vestado);
ResultSet rs = st.executeQuery();
prestamos = new Vector();
while ( rs.next() )
    prestamos.add( new cPrestamo(  rs.getString("idPrestamo"),
                                   rs.getString("idMaterial"),
                                   rs.getString("idPersona"),
                                   rs.getString("fec_pres"),
                                   rs.getString("fec_dev"),
                                   rs.getInt("multa"),
                                   rs.getString("usuario"),
                                   rs.getString("estado")
                                )
    );
rs.close();
st.close();
dbm.closeConnection( con );
return prestamos;
} catch ( Exception e ) { error = e.getMessage(); }
return null;
}

```

```

public int BuscarMulta(String codusu){

```

```

    int multa=0;

```

```

    Vector vec1=new Vector();

```

```

    Vector vec2=new Vector();

```

```

    cPrestamo objpres=new cPrestamo();

```

```

    objpres.BarrerMultas();

```

```

    //verificamos sus prestamos pendientes

```

```

    vec1=objpres.BuscarUsuario(codusu,"P");

```

```

    System.out.println("Vector 1"+ vec1.size());

```

```

    vec2=objpres.BuscarUsuario(codusu,"F"); //verificamos sus prestamos q falta

```

regularizar

```

        System.out.println("Vector 2"+ vec2.size());
        for (int i=0;i<vec1.size();i++){ //sacamos la multa de prestamos pendientes
            objpres=(cPrestamo) vec1.elementAt(i);
            multa=multa+objpres.getMulta();
        }

        for (int i=0;i<vec2.size();i++){ //sacamos la multa de prestamos por regularizar
            objpres=(cPrestamo) vec2.elementAt(i);
            multa=multa+objpres.getMulta();
        }
        return multa;
    }

// Encapsulamiento de atributos
    public String getIdPrestamo() {
        return idPrestamo;
    }
    public void setIdPrestamo(String idPrestamo) {
        this.idPrestamo = idPrestamo;
    }
    public String getIdMaterial() {
        return idMaterial;
    }
    public void setIdMaterial(String idMaterial) {
        this.idMaterial = idMaterial;
    }
    public String getIdPersona() {
        return idPersona;
    }
    public void setIdPersona(String idPersona) {
        this.idPersona = idPersona;
    }
    public String getFec_pres() {
        return fec_pres;
    }

    public void setFec_pres(String fec_pres) {
        this.fec_pres = fec_pres;
    }
}

```

```

public String getFec_dev() {
    return fec_dev;
}
public void setFec_dev(String fec_dev) {
    this.fec_dev = fec_dev;
}
public int getMulta() {
    return multa;
}
public void setMulta(int multa) {
    this.multa = multa;
}
public String getEstado() {
    return estado;
}
public void setEstado(String estado) {
    this.estado = estado;
}
public String getUsuario() {
    return usuario;
}
public void setUsuario(String usuario) {
    this.usuario = usuario;
}
}
// Fin de la clase prestamo

```

Clase DBManager.

```

package clases;
// Importando librerías
import java.sql.*;

public class DBManager {
    private final String url="jdbc:microsoft:sqlserver://localhost:1433;databaseName=biblioteca";
    private final String driver="com.microsoft.jdbc.sqlserver.SQLServerDriver";
    private final String user="sa";
    private final String pass="sa";

    private String error;
    public String geterror(){
        return error;
    }
}

```

```

public Connection getConnection(){
    error = null;
    try{
        Class.forName(driver);
        return (DriverManager.getConnection(url, user, pass));
    }catch (Exception e){error = e.toString();}
    return null;
}

public void closeConnection(Connection con){
    error = null;
    try{
        con.close();
    }catch (Exception e){error=e.toString();}
}
}

```

// Fin de la clase DBManager

Servlet sAutor

```

package servlets;

// Importando librerías
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
import clases.*;

// @author SISBIB

public class sAutor extends HttpServlet {
    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code>
methods.
    * @param request servlet request
    * @param response servlet response
    */

    protected void processRequest(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        RequestDispatcher rd=null;

        String vbusca="";

```

```

String vform="";
String vidAutor="";
String nomusu="";
int numPag=0;
int limitepag=0;
String vnombre,vapepat,vapemat,vpais;
vnombre=vapepat=vapemat=vpais="";

//recojo el txtpara
vbusca = (String) req.getParameter("txtpara");
if (vbusca==null) vbusca="";///si en nulo

//recojo el FORM
vform = (String) req.getParameter("FORM");
if (vform==null) vform="";///si en nulo
///recojo el TXTCON
nomusu=(String) req.getParameter("TXTCON");
if (nomusu==null) nomusu="";///si en nulo

//para hacer disable a mis botones de mi jsp
String borraradelante="";
String borraratras="";

cAutor objau= new cAutor();
cMaterial objmat= new cMaterial();

if(vform.equals("M")){
    numPag = Integer.parseInt((String) req.getParameter("numpag"));
    if (String.valueOf(numPag)==null) numPag=0;///si en nulo
    if(vbusca.equals("BN")){
        Vector vecautor=new Vector();
        vecautor = objau.BuscarAutorNombre(nomusu);
        int tamcons=vecautor.size();// estraee el tamaño de mi vector
        if(tamcons>0){
            int numReg=15;//para mostrar de 15 en 15
            limitepag=(tamcons/numReg) + 1;//saca el numero de paginas de la paginacion
            if((tamcons%numReg)==0){limitepag=limitepag-1;}
            if((numPag==limitepag) || (tamcons<=numReg)){borraradelante="True";}
            if((numPag==1)){borraratras="True";}
            vecautor=objau.BuscarAutorNombrePagina(nomusu,numPag,numReg);

```



```

        req.setAttribute("vector", vecautor);
    }else
    {vecautor = objau.BuscarAutorNombre(nomusu);
    req.setAttribute("vector", vecautor);
    }

    rd =
getServletContext().getRequestDispatcher("/manAutor.jsp?mostrar=MV&nomusu="+nomusu+"
&numpag="+numPag+"&borraradelante="+borraradelante+"&limite="+limitepag+"&borraratras=
"+borraratras+"&vbusca=BN");
    }

    if(vbusca.equals("BT")){
        Vector vecautor=new Vector();
        vecautor = objau.BuscarTodo();
        int tamcons=vecautor.size();
        if(tamcons>0){
            int numReg=15;//para mostrar de 15 en 15
            limitepag=(tamcons/numReg) + 1;
            if((tamcons%numReg)==0){limitepag=limitepag-1;}
            if((numPag==limitepag) || (tamcons<=numReg)){borraradelante="True";}
            if(numPag==1){borraratras="True";}
            vecautor=objau.BuscarTodopaginas(numPag,numReg);
            req.setAttribute("vector", vecautor);
        }else
        {vecautor = objau.BuscarTodo();
        req.setAttribute("vector", vecautor);
        }

        rd =
getServletContext().getRequestDispatcher("/manAutor.jsp?mostrar=MV&nomusu="+nomusu+"
&numpag="+numPag+"&borraradelante="+borraradelante+"&limite="+limitepag+"&borraratras=
"+borraratras+"&vbusca=BT");
    }
    rd.forward(req,res);
}

if(vform.equals("MM")){
    if(vbusca.equals("M")){
        String CODUSU=(String) req.getParameter("CODUSU");

```

```

        if(objau.BuscarIdAutor(CODUSU.trim())){
            req.setAttribute("obj",objau);
            rd =
getServletContext().getRequestDispatcher("/reg_autor.jsp?txtpara=M&FORM=M");
        }
    }
    if(vbusca.equals("N")){System.out.println("Estoy psts form nuevo");
        rd = getServletContext().getRequestDispatcher("/reg_autor.jsp?txtpara=N&FORM=M");
    }
    rd.forward(req,res);
}
if (vform.equals("MR")){
    String vidautor=(String) req.getParameter("txtcodusu2");
    vnombre=(String) req.getParameter("txtNombre");
    vapepat=(String) req.getParameter("txtApepat");
    vapemat=(String) req.getParameter("txtApemat");
    vpais=(String) req.getParameter("txtPais");

    if(vbusca.equals("R")){
        vidAutor=objau.GenerarCodigo();
        if(objau.Crear(vidAutor,vnombre,vapepat,vapemat,vpais)){
            rd =
getServletContext().getRequestDispatcher("/exito.jsp?url=sAutor?txtpara=BT&FORM=M&nump
ag=1&opcion=M");
        }
        else{
            rd = getServletContext().getRequestDispatcher("/error.jsp?url=manAutor&razon=No se
puede guardar los datos");
        }
    }

    if(vbusca.equals("M")){
        if(objau.Modificar(vidautor,vnombre,vapepat,vapemat,vpais)){
            rd =
getServletContext().getRequestDispatcher("/exito.jsp?url=sAutor?txtpara=BT&FORM=M&nump
ag=1&opcion=M");
        }
        else{
            rd = getServletContext().getRequestDispatcher("/error.jsp?url=manAutor&razon=No se
puede guardar los datos");
        }
    }
}

```

```

    }
}
if(vbusca.equals("E")){
    Vector autores = new Vector();
    autores = objmat.BuscarMaterialAutor(vidautor);
    System.out.println("dentro de E"+" "+vidautor);
    if(autores.size()==0){
        if(objau.Eliminar(vidautor)){
            rd =
getServletContext().getRequestDispatcher("/exito.jsp?url=sAutor?btpara=BT&FORM=M&nump
ag=1&opcion=M");
        }
        else{
            rd =
getServletContext().getRequestDispatcher("/error.jsp?url=manAutor&razon=No se puede
eliminar al Autor, esta registrado en un material");
        }
    }
    else{
        rd = getServletContext().getRequestDispatcher("/error.jsp?url=manAutor&razon=No
se puede eliminar al Autor, esta registrado en un material");
    }

}
rd.forward(req,res);
}
out.close();
}

```

// Fin del Servlet sAutor

Servlet sMaterial.

```

package servlets;
// Importamos librerias
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
import clases.*;
import java.text.*;

```

```

// @author SISBIB
public class sMaterial extends HttpServlet {

    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code>
    methods.
    * @param request servlet request
    * @param response servlet response
    */
    protected void processRequest(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {
        res.setContentType("text/html;charset=UTF-8");
        PrintWriter out = res.getWriter();

        //invocando a la sesion
        cUsuario objusuario= new cUsuario();
        HttpSession session = req.getSession();
        objusuario = (cUsuario) session.getAttribute("user_name");
        String codusu = objusuario.getIdPersona();

        String tipo_material = (String) req.getParameter("TIPO");
        String para = (String) req.getParameter("TXTPARA");
        String vbuscarpor = "";
        cMaterial objm=new cMaterial();
        cTesis objt = new cTesis();
        cLibro objl = new cLibro();
        cPubSer objp = new cPubSer();
        cEditorial objedi = new cEditorial();
        cAutor objautor = new cAutor();
        cMateria objmat = new cMateria();

        ///busca los materiales de baja
        if(para.equals("BO")){
            String codlib= (String) req.getParameter("CODLIB");
            codlib=codlib.toUpperCase();
            vbuscarpor=codlib.substring(9,10);

            if(vbuscarpor.equals("L")){
                if (objm.BuscarIdMaterialBaja(codlib)){
                    if(objl.BuscarIdLibro(objm.getIdMaterial())){
                        objedi.BuscarIdEditorial(objl.getIdEditorial());

```

```

        objautor.BuscarIdAutor(objl.getIdAutor());
        objmat.BuscarIdMateria(objl.getIdMateria());
        vbuscarpor=objm.getIdMaterial().substring(9,10);
        req.setAttribute("objmaterial",objm);
        req.setAttribute("objl",objl);
        req.setAttribute("objedi",objedi);
        req.setAttribute("objautor",objautor);
        req.setAttribute("objmat",objmat);

req.getRequestDispatcher("/cons_materialbaja.jsp?txtpara=MO&valor="+vbuscarpor).forward(r
eq,res);

        } else{
            RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialbaja");
            rd.forward(req,res);
        }
    } else{
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialbaja");
        rd.forward(req,res);
    }
}
if(vbuscarpor.equals("T")){
    if(objm.BuscarIdMaterialBaja(codlib)){
        if(objt.BuscarIdTesis(objm.getIdMaterial())){
            objedi.BuscarIdEditorial(objt.getIdEditorial());
            objautor.BuscarIdAutor(objt.getIdAutor());
            objmat.BuscarIdMateria(objt.getIdMateria());
            vbuscarpor=objm.getIdMaterial().substring(9,10);

            req.setAttribute("objmaterial",objm);
            req.setAttribute("objt",objt);
            req.setAttribute("objedi",objedi);
            req.setAttribute("objautor",objautor);
            req.setAttribute("objmat",objmat);

```

```
req.getRequestDispatcher("/cons_materialbaja.jsp?txtpara=MO&valor="+vbuscarpor).forward(r
eq,res);
```

```
    } else{
```

```
        RequestDispatcher rd =
```

```
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialbaja");
```

```
        rd.forward(req,res);
```

```
    }
```

```
    } else{
```

```
        RequestDispatcher rd =
```

```
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialbaja");
```

```
        rd.forward(req,res);
```

```
    }
```

```
}
```

```
if(vbuscarpor.equals("P")){
```

```
    if (objm.BuscarIdMaterialBaja(codlib)){
```

```
        if(objp.BuscarIdPubser(objm.getIdMaterial())){
```

```
            objedi.BuscarIdEditorial(objp.getIdEditorial());
```

```
            objautor.BuscarIdAutor(objp.getIdAutor());
```

```
            objmat.BuscarIdMateria(objp.getIdMateria());
```

```
            vbuscarpor=objm.getIdMaterial().substring(9,10);
```

```
            req.setAttribute("objmaterial",objm);
```

```
            req.setAttribute("objp",objp);
```

```
            req.setAttribute("objedi",objedi);
```

```
            req.setAttribute("objautor",objautor);
```

```
            req.setAttribute("objmat",objmat);
```

```
req.getRequestDispatcher("/cons_materialbaja.jsp?txtpara=MO&valor="+vbuscarpor).forward(r
eq,res);
```

```
    } else{
```

```
        RequestDispatcher rd =
```

```
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialbaja");
```

```
        rd.forward(req,res);
```

```
    }
```

```
    } else{
```

```

        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialbaja");
        rd.forward(req,res);
    }
}

}

///busca los materiales por codigo de barras
if(para.equals("BCB")){
    String codlib= (String) req.getParameter("CODLIB");
    codlib=codlib.toUpperCase();
    vbuscarpor=codlib.substring(9,10);
    System.out.println("cod material" +codlib);
    System.out.println("buscar por" +vbuscarpor);

    if(vbuscarpor.equals("L")){
        if(objl.BuscarIdLibro(codlib)){
            objedi.BuscarIdEditorial(objl.getIdEditorial());
            objautor.BuscarIdAutor(objl.getIdAutor());
            objmat.BuscarIdMateria(objl.getIdMateria());
            vbuscarpor=objl.getIdMaterial().substring(9,10);
            req.setAttribute("objmaterial",objl);
            req.setAttribute("objl",objl);
            req.setAttribute("objedi",objedi);
            req.setAttribute("objautor",objautor);
            req.setAttribute("objmat",objmat);

            req.getRequestDispatcher("/cons_materialcod.jsp?txtpara=MO&valor="+vbuscarpor).forward(re
q,res);
        } else{
            RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialcod");
            rd.forward(req,res);
        }
    }
    if(vbuscarpor.equals("T")){

```

```

if(objt.BuscarIdTesis(codlib)){
    objedi.BuscarIdEditorial(objt.getIdEditorial());
    objautor.BuscarIdAutor(objt.getIdAutor());
    objmat.BuscarIdMateria(objt.getIdMateria());
    vbuscarpor=objt.getIdMaterial().substring(9,10);

```

```

    req.setAttribute("objmaterial",objt);
    req.setAttribute("objt",objt);
    req.setAttribute("objedi",objedi);
    req.setAttribute("objautor",objautor);
    req.setAttribute("objmat",objmat);

```

```

req.getRequestDispatcher("/cons_materialcod.jsp?txtpara=MO&valor="+vbuscarpor).forward(re
q,res);

```

```

    } else{

```

```

        RequestDispatcher rd =

```

```

getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialcod");

```

```

        rd.forward(req,res);

```

```

    }

```

```

}

```

```

if(vbuscarpor.equals("P")){

```

```

    if(objp.BuscarIdPubser(codlib)){

```

```

        objedi.BuscarIdEditorial(objp.getIdEditorial());
        objautor.BuscarIdAutor(objp.getIdAutor());
        objmat.BuscarIdMateria(objp.getIdMateria());
        vbuscarpor=objp.getIdMaterial().substring(9,10);
        req.setAttribute("objmaterial",objp);
        req.setAttribute("objp",objp);
        req.setAttribute("objedi",objedi);
        req.setAttribute("objautor",objautor);
        req.setAttribute("objmat",objmat);

```

```

req.getRequestDispatcher("/cons_materialcod.jsp?txtpara=MO&valor="+vbuscarpor).forward(re
q,res);

```

```

    } else{

```



```

        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialcod");
        rd.forward(req,res);
    }
}
else{
    RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_materialcod");
    rd.forward(req,res);
}
}

```

```

if(tipo_material.equals("B")){///para buscar el material
    String codaut = (String) req.getParameter("CODAUTOR");
    String coddw = (String) req.getParameter("COWDW");
    String codmal = (String) req.getParameter("CODMAL");
    String clase = (String) req.getParameter("CLASE");
    vbuscarpor = (String) req.getParameter("CBOMAT");
    if(vbuscarpor==null) vbuscarpor="";
    String vsegun = (String) req.getParameter("CBOSEGUN");

```

```

/*si llega null*/
if(codaut==null){codaut="";}
if(coddw==null){coddw="";}
if(codmal==null){codmal="";}

```

```

if(clase.equals("L")){///busca en libro
    System.out.println("entro L");
    String segun = (String) req.getParameter("SEGUN");
    ///buscar para mostrar vector
    if(para.equals("BV")){
        Vector libro=new Vector();
        Vector tesis=new Vector();
        Vector pubs=new Vector();
        Vector veclibro=new Vector();
        Vector veceditorial=new Vector();
        Vector vecmateria=new Vector();
        Vector vectesis=NET Vector();

```

```

Vector vecpubser=new Vector();
vbuscarpor=vbuscarpor.trim();
vsegun=vsegun.trim();
if(vbuscarpor.equals("L")){
    //preguntamos si es buscado por autot o por codigo dewey y mala

if(segun.equals("A")){libro=objm.BuscarMaterialAutorAltas(vbuscarpor,vsegun,codaut);}

if(segun.equals("C")){libro=objm.BuscarMaterialCodClaAlta(vbuscarpor,vsegun,coddw,codmal);
}

    if(libro.size(>0){
        for(int i=0;i<libro.size();i++){
            objm=(cMaterial)libro.elementAt(i);
            if(objl.BuscarIdLibro(objm.getIdMaterial()))//para q saque el subtítulo
            {veclibro.addElement(objl.getSubtitulo());} else{veclibro.addElement("");}
            if(objjedi.BuscarIdEditorial(objl.getIdEditorial()))//para sacar el nombre
editorial
                {veceditorial.addElement(objjedi.getNombre());}
            else{veceditorial.addElement("");}
            if(objmat.BuscarIdMateria(objl.getIdMateria()))//para sacar el nombre
materia
                {vecmateria.addElement(objmat.getDescripcion());}
            else{vecmateria.addElement("");}
        }
        objautor.BuscarIdAutor(objl.getIdAutor());//para el autor

        req.setAttribute("vector",libro);
        req.setAttribute("objlibro",veclibro);
        req.setAttribute("objjedi",veceditorial);
        req.setAttribute("objautor",objautor);
        req.setAttribute("objmat",vecmateria);

        req.getRequestDispatcher("/cons_material.jsp?txtpara=MV").forward(req,res);

    } else{
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_material");
        rd.forward(req,res);
    }

```

```

    }
    if(vbuscarpor.equals("T")){
        //preguntamos si es buscado por autot o por codigo dewey y mala

if(segun.equals("A")){tesis=objm.BuscarMaterialAutorAltas(vbuscarpor,vsegun,codaut);}

if(segun.equals("C")){tesis=objm.BuscarMaterialCodClaAlta(vbuscarpor,vsegun,coddw,codmal);
}

    if(tesis.size()>0){
        for(int i=0;i<tesis.size();i++){
            objm=(cMaterial)tesis.elementAt(i);
            if(objt.BuscarIdTesis(objm.getIdMaterial()))//para q saque el subtitulo
            {vectesis.addElement(objt.getFacultad());} else{vectesis.addElement("");}
            if(objjedi.BuscarIdEditorial(objt.getIdEditorial()))//para sacar el nombre
editorial
                {veceditorial.addElement(objjedi.getNombre());}
            else{veceditorial.addElement("");}
            if(objmat.BuscarIdMateria(objt.getIdMateria()))//para sacar el nombre
materia
                {vecmateria.addElement(objmat.getDescripcion());}
            else{vecmateria.addElement("");}
        }
        objautor.BuscarIdAutor(objt.getIdAutor());//para el autor

        req.setAttribute("vector",tesis);
        req.setAttribute("objtes",vectesis);
        req.setAttribute("objjedi",veceditorial);
        req.setAttribute("objautor",objautor);
        req.setAttribute("objmat",vecmateria);

        req.getRequestDispatcher("/cons_material.jsp?btppara=MV").forward(req,res);

    } else{
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_material");
        rd.forward(req,res);
    }
}

```

```

        if(vbuscarpor.equals("P")){
            //preguntamos si es buscado por autot o por codigo dewey y mala

            if(según.equals("A")){pubs=objm.BuscarMaterialAutorAltas(vbuscarpor,vsegún,codaut);}

            if(según.equals("C")){pubs=objm.BuscarMaterialCodClaAlta(vbuscarpor,vsegún,coddw,codmal)
            ;}

            System.out.println("ACA en P"+" "+vbuscarpor+" "+tamano+pubs.size());
            if(pubs.size(>0){
                for(int i=0;i<pubs.size();i++){
                    objm=(cMaterial)pubs.elementAt(i);
                    if(objp.BuscarIdPubser(objm.getIdMaterial()))//para q saque el subtítulo
                    {vecpubser.addElement(objp.getIssn());} else{vecpubser.addElement("");}
                    if(objedi.BuscarIdEditorial(objp.getIdEditorial()))//para sacar el nombre
editorial
                        {veceditorial.addElement(objedi.getNombre());}
                    else{veceditorial.addElement("");}
                    if(objmat.BuscarIdMateria(objp.getIdMateria()))//para sacar el nombre
materia
                        {vecmateria.addElement(objmat.getDescripcion());}
                    else{vecmateria.addElement("");}
                }
                objautor.BuscarIdAutor(objp.getIdAutor());//para el autor

                req.setAttribute("vector",pubs);
                req.setAttribute("objpub",vecpubser);
                req.setAttribute("objedi",veceditorial);
                req.setAttribute("objautor",objautor);
                req.setAttribute("objmat",vecmateria);

                req.getRequestDispatcher("/cons_material.jsp?txtpara=MV").forward(req,res);

            } else{
                RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el
registro&url=cons_material");
                rd.forward(req,res);
            }
        }
    }
}

```

```
}
```

```
///buscar para modificar
```

```
if(para.equals("BM")){
```

```
    String codlib= (String) req.getParameter("CODLIB");
```

```
    vbuscarpor=codlib.substring(9,10);
```

```
    if(vbuscarpor.equals("L")){
```

```
        objm.BuscarIdMaterial(codlib);
```

```
        if(objl.BuscarIdLibro(objm.getIdMaterial())){
```

```
            objedi.BuscarIdEditorial(objm.getIdEditorial());
```

```
            objautor.BuscarIdAutor(objm.getIdAutor());
```

```
            objmat.BuscarIdMateria(objm.getIdMateria());
```

```
            req.setAttribute("objl",objl);
```

```
            req.setAttribute("objedi",objedi);
```

```
            req.setAttribute("objautor",objautor);
```

```
            req.setAttribute("objmat",objmat);
```

```
req.getRequestDispatcher("/man_libro.jsp?valor=E&segun="+segun).forward(req,res);
```

```
    } else{
```

```
        RequestDispatcher rd =
```

```
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el  
registro&url=man_libro");
```

```
        rd.forward(req,res);
```

```
    }
```

```
}
```

```
if(vbuscarpor.equals("T")){
```

```
    objm.BuscarIdMaterial(codlib);
```

```
    if(objt.BuscarIdTesis(objm.getIdMaterial())){
```

```
        objedi.BuscarIdEditorial(objm.getIdEditorial());
```

```
        objautor.BuscarIdAutor(objm.getIdAutor());
```

```
        objmat.BuscarIdMateria(objm.getIdMateria());
```

```
        req.setAttribute("objt",objt);
```

```
        req.setAttribute("objedi",objedi);
```

```
        req.setAttribute("objautor",objautor);
```

```
req.setAttribute("objmat",objmat);
```

```
req.getRequestDispatcher("/man_tesis.jsp?valor=E&segun="+segun).forward(req,res);
```

```
    } else{
```

```
        RequestDispatcher rd =
```

```
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el  
registro&url=man_tesis");
```

```
        rd.forward(req,res);
```

```
    }
```

```
}
```

```
if(vbuscarpor.equals("P")){
```

```
    objm.BuscarIdMaterial(codlib);
```

```
    if(objp.BuscarIdPubser(objm.getIdMaterial())){
```

```
        objedi.BuscarIdEditorial(objm.getIdEditorial());
```

```
        objautor.BuscarIdAutor(objm.getIdAutor());
```

```
        objmat.BuscarIdMateria(objm.getIdMateria());
```

```
        req.setAttribute("objp",objp);
```

```
        req.setAttribute("objedi",objedi);
```

```
        req.setAttribute("objautor",objautor);
```

```
        req.setAttribute("objmat",objmat);
```

```
req.getRequestDispatcher("/man_pubser.jsp?valor=E&segun="+segun).forward(req,res);
```

```
    } else{
```

```
        RequestDispatcher rd =
```

```
getServletContext().getRequestDispatcher("/error.jsp?razon=No se encontro el  
registro&url=man_pubser");
```

```
        rd.forward(req,res);
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
}
```

```
//fin del if buscar
```

```

else{
    //REGOGIENDO PARAMETROS
    String idEditorial = (String) req.getParameter("CODEEDITORIAL");
    String idAutor = (String) req.getParameter("CODAUTOR");
    String idMateria = (String) req.getParameter("CODMATERIA");
    String titulo = (String) req.getParameter("NOMTITULO");
    String lug_edi = (String) req.getParameter("LUGEDICION");
    String ano_edi = (String) req.getParameter("ANIOEDICION");
    String num_edi = (String) req.getParameter("EDICIONNUM");
    String num_pag = (String) req.getParameter("NUMPAG");
    String estante = (String) req.getParameter("ESTANTE");
    String condicion = (String) req.getParameter("CONDICION");
    String observacion = (String) req.getParameter("OBS");
    if (observacion==null) observacion="";
    String otro_aut = (String) req.getParameter("OTROSAUT");
    if (otro_aut==null) otro_aut="";
    String fec_ing = "";
    String resumen = (String) req.getParameter("RESUMEN");
    String descriptores = (String) req.getParameter("DESCRIPTORES");
    String reserva = (String) req.getParameter("RESERVA");
    String coddw = (String) req.getParameter("CODIGODW");
    String codml = (String) req.getParameter("CODIGOML");
    String codbib = (String) req.getParameter("CODIGOBIB");
    if (codbib==null) codbib="";
    String codalm = (String) req.getParameter("CODIGOALM");
    String largo = (String) req.getParameter("LARGO");
    String ancho = (String) req.getParameter("ANCHO");
    String espesor = (String) req.getParameter("ESPESOR");
    String ejemplar = (String) req.getParameter("EJEMLAR");
    String estado = "D";
    String idMaterial = "";
    //fecha
    String d,m,a;
    Date hoy=new Date();
    d=String.valueOf(hoy.getDate());
    m=String.valueOf(hoy.getMonth()+1);
    a=String.valueOf(hoy.getYear()+1900);
    if (d.length()==1) d = "0"+d;
    if (m.length()==1) m = "0"+m;
    fec_ing=d+"/"+m+"/"+a;

```

```
String error = "";
```

```
//preguntamos que tipo de material es
```

```
if(tipo_material.equals("L")){//MATERIAL = LIBRO
```

```
    //cLibro objl = new cLibro();
```

```
    String ilustracion = (String) req.getParameter("ILUSTRACION");
```

```
    String tabla = (String) req.getParameter("TABLAS");
```

```
    String cd = (String) req.getParameter("CD");
```

```
    String tomo = (String) req.getParameter("TOMO");
```

```
    String subtítulo = (String) req.getParameter("NOMSUBTITULO");
```

```
    String volumen = (String) req.getParameter("VOLUMEN");
```

```
    String isbn = (String) req.getParameter("ISBN");
```

```
    if (tomo==null) tomo="";
```

```
    if (subtítulo==null) subtítulo="";
```

```
    if (volumen==null) volumen="";
```

```
    if (isbn==null) isbn="";
```

```
    if(!para.equals("E")){//ENTRA SI ES NUEVO O MODIFICAR
```

```
        if(para.equals("N")){
```

```
            idMaterial=objl.GenerarCodigo();
```

```
if(objl.Crear(idMaterial,idEditorial,idAutor,idMateria,codusu,titulo,lug_edi,ano_edi,Integer.parseInt(num_edi),Integer.parseInt(num_pag),Integer.parseInt(estante),condicion,observacion,coddw,codml,codbib,codalm,Integer.parseInt(largo),Integer.parseInt(ancho),Integer.parseInt(espesor),Integer.parseInt(ejemplar),otro_aut,fec_ing,resumen,descriptores,reserva,estado,isbn,tomo,subtitulo,volumen,ilustracion,tabla,cd)){
```

```
    RequestDispatcher rd =
```

```
getServletContext().getRequestDispatcher("/sBarra?CAPTION="+idMaterial);
```

```
    rd.forward(req,res);
```

```
    }else{
```

```
        error="NO SE PUEDE CREAR EL REGISTRO...";
```

```
        RequestDispatcher rd =
```

```
getServletContext().getRequestDispatcher("/error.jsp?razon="+error);
```

```
        rd.forward(req,res);
```

```
    }
```

```
    } else {
```

```
        idMaterial = (String) req.getParameter("txtIDMATERIAL");
```

```
        estado=(String) req.getParameter("txtESTADO");
```

```
        if(estado.equals("B")){estado="D";}//para dar de alta a los materiales en este
```

```
caso el libro
```



```

if(objt.Modificar(idMaterial,idEditorial,idAutor,idMateria,titulo,lug_edi,ano_edi,Integer.parseInt(num_edi),Integer.parseInt(num_pag),Integer.parseInt(estante),condicion,observacion,coddw,codml,codbib,codalm,Integer.parseInt(largo),Integer.parseInt(ancho),Integer.parseInt(espesor),Integer.parseInt(ejemplar),otro_aut,fec_ing,resumen,descriptores,reserva,"B",facultad,grado)){
    RequestDispatcher rd =
getServletContext().getRequestDispatcher("/exitos.jsp?url=cons_material");
    rd.forward(req,res);
}
else{
    error="NO SE PUDE ELIMINAR EL REGISTRO...";
    RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon="+error);
    rd.forward(req,res);
}

}

}
}
//FIN DEL ELSE IF TESIS
else if(tipo_material.equals("P")){//MATERIAL = PUBLICACION
    //cPubSer objp = new cPubSer();
    String periodicidad = (String) req.getParameter("PERIODICIDAD");
    String tomo = (String) req.getParameter("TOMO");
    String numero = (String) req.getParameter("NUMERO");
    String aniocir = (String) req.getParameter("ANIOCIR");
    String volumen = (String) req.getParameter("VOLUMEN");
    String issn = (String) req.getParameter("ISSN");
    if (tomo==null) tomo="";
    if (numero==null) numero="";
    if (aniocir==null) aniocir="";
    if (volumen==null) volumen="";
    if (issn==null) issn="";
    if(!para.equals("E")){//ENTRA SI ES NUEVO O MODIFICAR

        if(para.equals("N")){
            idMaterial=objp.GenerarCodigo();

            if(objp.Crear(idMaterial,idEditorial,idAutor,idMateria,codusu,titulo,lug_edi,ano_edi,Integer.parseInt(num_edi),Integer.parseInt(num_pag),Integer.parseInt(estante),condicion,observacion,coddw,codml,codbib,codalm,Integer.parseInt(largo),Integer.parseInt(ancho),Integer.parseInt(espesor),Integer.parseInt(ejemplar),otro_aut,fec_ing,resumen,descriptores,reserva,estado,tomo,aniocir,Integer.parseInt(numero),volumen,issn,periodicidad)){

```

```

        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/sBarra?CAPTION="+idMaterial);
        rd.forward(req,res);
    }else{
        error="NO SE PUDE CREAR EL REGISTRO...";
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon="+error);
        rd.forward(req,res);
    }
} else {
    idMaterial = (String) req.getParameter("txtIDMATERIAL");
    estado=(String) req.getParameter("txtESTADO");

    if(objp.Modificar(idMaterial,idEditorial,idAutor,idMateria,titulo,lug_edi,ano_edi,Integer.parseInt(num_edi),Integer.parseInt(num_pag),Integer.parseInt(estante),condicion,observacion,coddw,codml,codbib,codalm,Integer.parseInt(largo),Integer.parseInt(ancho),Integer.parseInt(espesor),Integer.parseInt(ejemplar),otro_aut,fec_ing,resumen,descriptores,reserva,estado,tomo,aniocir,Integer.parseInt(numero),volumen,issn,periodicidad)){
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/sBarra?CAPTION="+idMaterial);
        rd.forward(req,res);
    }else{
        error="NO SE PUDE MODIFICAR EL REGISTRO...";
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon="+error);
        rd.forward(req,res);
    }
}
} else{//ENTRA SI ES ELIMINAR:DAR DE BAJA A UN MATERIAL
    idMaterial = (String) req.getParameter("txtIDMATERIAL");

    if(objp.Modificar(idMaterial,idEditorial,idAutor,idMateria,titulo,lug_edi,ano_edi,Integer.parseInt(num_edi),Integer.parseInt(num_pag),Integer.parseInt(estante),condicion,observacion,coddw,codml,codbib,codalm,Integer.parseInt(largo),Integer.parseInt(ancho),Integer.parseInt(espesor),Integer.parseInt(ejemplar),otro_aut,fec_ing,resumen,descriptores,reserva,"B",tomo,aniocir,Integer.parseInt(numero),volumen,issn,periodicidad)){
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/exito.jsp?url=cons_material");
        rd.forward(req,res);
    }else{

```

```

        error="NO SE PUDE ELIMINAR EL REGISTRO...";
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon="+error);
        rd.forward(req,res);
    }

}
} //FIN DE PUBLICACION
else{
    RequestDispatcher rd =
getServletContext().getRequestDispatcher("/error.jsp?razon="+error);
    rd.forward(req,res);
}
}
out.close();
}

```

4.4. Fase Transición

4.4.1. Flujo de Trabajo de Prueba

El Proceso Unificado de Rational propone que las pruebas se dimensionan en tres niveles: Calidad, Estado y Pruebas de Tipo. Para este proyecto se ha optado por realizar las siguientes pruebas:

4.4.1.1 A Nivel de Calidad

A) Funcionalidad

A.1) Requerimientos del Sistema

Por cada requerimiento se realizó un cuadro con datos válidos y otro con datos no válidos respectivamente. (Ver Anexo 4).

A.2) Operatividad

Para demostrar la operatividad del Sistema se realizaron las pruebas necesarias a través de la validación de la entrada de

datos, cuyos valores pueden ser numéricos, alfabéticos y alfanuméricos; como se muestra a continuación:

Cuadro 29: Prueba de Operatividad

Tipo	Caso de Prueba	Resultado
Numérico	1.- Numero Valido	Aceptado
	2.- Numero Positivo	Aceptado
	3.- Numero Negativo	Rechazado
	4.- No Numérico	Rechazado
Alfabético	1.- Alfabético Valido	Aceptado
	2.- No Alfabético	Rechazado
Alfanumérico	1.- Alfanumérico Valido	Aceptado

4.4.1.2 A Nivel de Estado

A) Sistema

A.1) Seguridad

Los usuarios acceden al sistema mediante su nombre de usuario y contraseña; y de acuerdo al tipo de usuario se ha validado para que sea restringido y así ingresen a donde les corresponde.

Cuadro 30: Prueba de Seguridad

Nombre de Usuario	Tipo de Usuario	Dirección URL
0001970004	Biblioteca Admin	index.jsp
0001970289	Biblioteca Gestión	indexges
0001890102	Biblioteca Ingreso	indexing.jsp
0001970466	Biblioteca Préstamo	indexbib.jsp

4.4.1.3 A Nivel de Tipo

A) Prueba de Carga

➤ En el Servidor

El sistema Web con Tecnología J2EE para la Administración de la Biblioteca se ha instalado en el servidor junto con la base de datos SQL Server, que existe en la Universidad Nacional de Ucayali.

Cuadro 31: Prueba de Carga Servidor

<u>Aplicación</u>	<u>Resultado</u>
Base de Datos biblioteca	Levantó
Aplicación sisbib en Tomcat	Levantó

➤ En el Cliente

En el caso de los clientes accedemos al sistema a través del navegador sin necesidad de instalar otros aplicativos.

Cuadro 32: Prueba de Carga Cliente

<u>Aplicación</u>	<u>Resultado</u>
Acceso a través del navegador	Levantó
Acceso BD	Levantó

CAPITULO V

RESULTADOS DE LA INVESTIGACIÓN

5.1. Resultados

5.1.1. A los usuarios

Según las encuestas realizadas a los usuarios (alumnos y docentes), se verifica que existe una mejora en cuanto a los valores de los indicadores luego de la implantación del sistema. (Ver Anexos 13 y 14).

- **A la pregunta:** El tiempo que demora en consultar un material bibliográfico con el sistema, le parece:

Cuadro 33: Valores Pregunta 1 Post Test aplicada a usuarios

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	200	39,8	39,8	39,8
	BUENO	242	48,2	48,2	88,0
	REGULAR	60	12,0	12,0	100,0
	Total	502	100,0	100,0	

- **A la pregunta:** La atención dentro de la Biblioteca Central de la Universidad Nacional de Ucayali con el sistema, le parece:

Cuadro 34: Valores Pregunta 3 Post Test aplicada a usuarios

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	200	39,8	39,8	39,8
	BUENO	251	50,0	50,0	89,8
	REGULAR	31	6,2	6,2	96,0
	MALO	20	4,0	4,0	100,0
	Total	502	100,0	100,0	

5.1.2. A trabajadores

Según los resultados obtenidos los trabajadores de la Biblioteca indicaron una mejora en cuanto a la ejecución de sus procesos y al manejo de información. Ver Anexos 13 y 14.

- **A la pregunta:** ¿Qué le parece la nueva forma en que se maneja la información en la Biblioteca Central?

Cuadro 35: Valores Pregunta 1 Post Test aplicada a especialistas

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	BUENO	6	85,7	85,7	85,7
	REGULAR	1	14,3	14,3	100,0
	Total	7	100,0	100,0	

- **A la pregunta:** El tiempo de demora y su facilidad en el procesamiento de información de los materiales bibliográficos con el sistema le parece:

Cuadro 36: Valores Pregunta 5 Post Test aplicada a especialistas

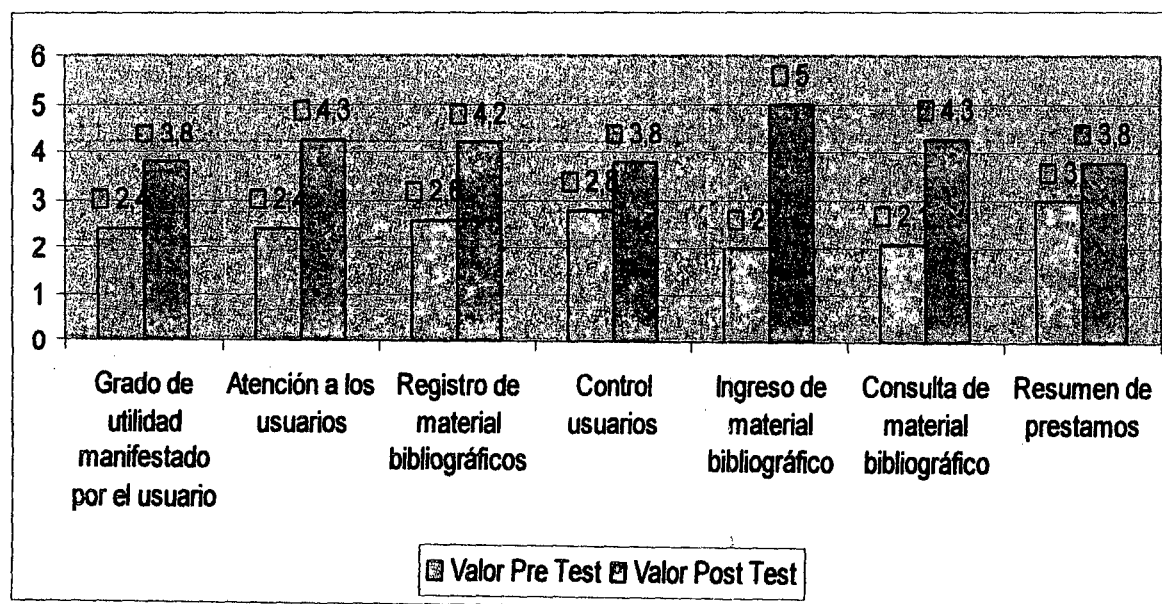
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	2	28,6	28,6	28,6
	BUENO	5	71,4	71,4	100,0
	Total	7	100,0	100,0	

A continuación, un cuadro comparativo que muestra en resumen todos los indicadores tomados con sus respectivos valores Pre y Post Test.

Cuadro 37: Comparación de valores Pre y Post

No	Indicador	Valor Pre Test	Valor Post Test
1	Grado de utilidad manifestado por el usuario	2.4	3.8
2	Atención a los usuarios	2.4	4.3
3	Registro de material bibliográficos	2.6	4.2
4	Control usuarios	2.8	3.8
5	Ingreso de material bibliográfico	2	5
6	Consulta de material bibliográfico	2.1	4.3
7	Resumen de prestamos	3	3.8

Gráfico 94: Comparación de valores Pre y Post



Los valores ponderados de los resultados de las encuestas antes de implantar el sistema (pre test) y luego de la implantación del mismo (post test), muestran una diferencia significativa en cada uno de los indicadores,

es así que el ingreso de material bibliográfico (5), tienen una diferencia de mayor relevancia luego de implantar el sistema, seguido de atención a los usuarios y consulta de material bibliográfico, ambos con 4,3.

Cada encuesta, tanto pre y post test han sido tabuladas, obteniéndose de ésta forma los resultados esperados. Ver Anexos 12, 13 y 14.

5.2. Prueba de hipótesis

5.2.1. Formulación de cuadros de valores de indicadores

La contrastación de la hipótesis se realizará de acuerdo al diseño de investigación mostrado en el capítulo III, el cual es conocido también como pre-test y post-test, que se representa mediante la siguiente simbología:

GE O1 X O2

Donde:

GE: Los usuarios de la Biblioteca Central de la Universidad Nacional de Ucayali

O1: Análisis de los resultados, antes de la instalación del Sistema Web con Tecnología J2EE.

X: Es el Sistema Web con tecnología J2EE

O2: Análisis de los resultados, después de la instalación del Sistema Web con Tecnología J2EE.

El procedimiento consiste en determinar en primer lugar una tabla de rango de valores, la cual nos permite ubicar valores cuantitativos de los indicadores, por medio de valores cualitativos expresados en este rango.

Posteriormente realizamos la comparación de valores entre indicadores de acuerdo al diseño de contrastación; esta comparación nos permite finalmente aceptar o rechazar la hipótesis de acuerdo a los estándares científicos de la Estadística.

Para variables cuantitativas:

RANGO SATISFACCIÓN

[0 – 1.5] Nada Satisfactorio

[1.6 – 3.2] Parcialmente Satisfactorio

[3.3 – 5] Satisfactorio

Para variables cualitativos:

RANGO SATISFACCIÓN

[0 – 1.5] Nada Ventajoso

[1.6 – 3.2] Poco Ventajoso

[3.3 – 5] Altamente Ventajoso

Los cuadros siguientes muestran la comparación de indicadores de la siguiente página se obtiene mediante la recolección de datos al realizar una encuesta pre test y post test según formato especificado en los anexos 5, 6, 7, 8, 9, 10 y 11.

Se emplea las siguientes abreviaturas:

VL: Valor cualitativo

VC: Valor cuantitativo

VI: Valor Inicial

VF: Valor Final

Cuadro 38: Comparación de indicadores cualitativos

No	Indicador	Procedimiento Actual (O ₁)	Valor Cualitativo (VL)	VI	Procedimiento con Estímulo (O ₂)	Valor Cualitativo (VL)	VF
01	Grado de utilidad manifestado por el usuario	Los procesos se realizan manualmente.	Deficiencia en el desarrollo de los procesos	2.4	Los procesos se automatizan mediante el uso del sistema	Optimización de los procesos	3.8
02	Atención a los usuarios	Los usuarios son atendidos lentamente debido a la falta de automatización	Insatisfacción por parte de los usuarios	2.4	La automatización permite dar mejor atención a los usuarios	Satisfacción de los usuarios	4.3
03	Registro de material bibliográficos	Se registran los materiales en general en hojas de cálculo.	Deficiencia en la consultas por material	2.6	Se realizan según tipo de material bibliográfico	Rapidez en las consultas por material	4.2
04	Control usuarios	No se realiza un control de ingreso y salida de usuarios	Deficiencia en el registro de usuarios	2.8	Se registra en el sistema las entradas y salidas de los usuarios a la biblioteca	Optimización para el registro de usuarios	3.8
05	Ingreso de material bibliográfico	Planeación por lo general en base a información desactualizada, sesgada y poco confiable.	Deficiencia en la planeación	2	Planeación basada con información en tiempo real, integra y altamente confiable.	Optimización en la planeación	5

Cuadro 39: Comparación de indicadores cuantitativos

No	Indicador	Procedimiento Actual (O ₁)	Valor Cuantitativo (VL)	VI	Procedimiento con Estímulo (O ₂)	Valor Cuantitativo (VL)	VF
01	Consulta de material bibliográfico	Se realiza en fichas bibliográficas	5 min.	2.1	Se realizan directamente en el sistema	2 seg.	4.3
02	Resumen de prestamos	Control manual de los préstamos realizados en la biblioteca	10 min.	3	Generación automática de los prestamos realizados	2 seg.	3.8

5.2.2. Supuestos de la prueba de hipótesis

1. Los datos muestrales se seleccionaron de manera no probabilística y por conveniencia, a una muestra de 7, correspondiente al personal administrativo de la biblioteca, que esta en contacto con el sistema, así mismo evaluamos los 7 principales indicadores relevantes para la prueba de hipótesis.
2. La hipótesis nula H_0 es la negación de la relación existente entre la variable independiente y la variable dependiente y la hipótesis H_a es la afirmación correspondiente.
3. Se utilizó el 95% del nivel de confiabilidad y 5% del nivel de significancia
4. Se acepta la hipótesis nula si el Valor Calculado T_c es menor al valor en tabla T_t , caso contrario se rechaza la H_0 y se acepta la hipótesis alterna H_a .

5.2.3. Cálculo del valor crítico y la función de prueba

Después de analizar las diferencias entre los indicadores en Pre-test (O_1) y post-test (O_2), se puede concluir que para todos los indicadores hay diferencias significativas y una mejora después de utilizar el Sistema Web con tecnología J2EE.

Cuadro 40: Resumen de Resultados de Valores

No	Indicador	VI	VF	D
01	Grado de utilidad manifestado por el usuario	2.4	3.8	1,4
02	Atención a los usuarios	2.4	4.3	1,9
03	Registro de material bibliográficos	2.6	4.2	1,6
04	Control usuarios	2.8	3.8	1
05	Ingreso de material bibliográfico	2	5	3
06	Consulta de material bibliográfico	2.1	4.3	2,2
07	Resumen de prestamos	3	3.8	0,8
TOTAL				11.9

Cálculo de la diferencia promedio (D)

$$\bar{D} = \frac{\sum \bar{D}_i}{n} = \frac{11,9}{7} = 1,7$$

Cálculo de la desviación estándar (S)

$$S = \sqrt{\frac{\sum (D_i - \bar{D})^2}{n - 1}}$$

$$S = \sqrt{\frac{(1,4-1,7)^2 + (1,9-1,7)^2 + (1,6-1,7)^2 + (1-1,7)^2 + (3-1,7)^2 + (2,2-1,7)^2 + (0,8-1,7)^2}{6}}$$

$$S = 0.75$$

Hipótesis Estática

$$H_0 = \mu_1 - \mu_2 \geq 0$$

$$H_a = \mu_2 - \mu_1 > 0$$

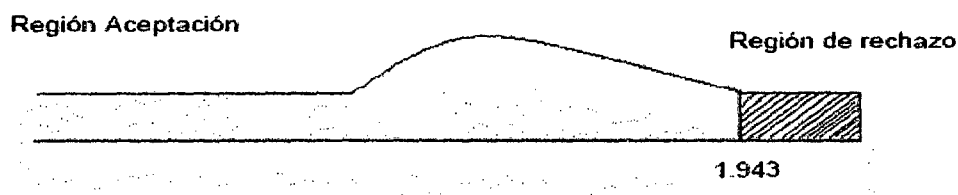
Cálculo del valor crítico o valor en tabla de la "T" Student (Tt)

$$T_t (1 - \alpha)_{(n-1)}$$

$$T_t (1 - 0.05)_{(7-1)}$$

$$T_t (0.95)_{(6)} = 1.943$$

$$T_t = 1,943$$

Regiones de Aceptación y Rechazo**Cálculo del valor calculado o Función de Prueba (Tc).**

$$T_c = \frac{\bar{D}}{\frac{S}{\sqrt{n}}} = \frac{1,7}{\frac{0,75}{\sqrt{7}}}$$

$$T_c = 5.99$$

Por lo tanto:

Como $T_c = 5,99$ es mayor que $T_t = 1,943$; entonces se rechazan H_0 y se acepta $H_a = \mu_2 - \mu_1 > 0$

Finalmente se concluye que la hipótesis planteada es aceptada.

En el desarrollo de la presente investigación a través de sus diferentes etapas, queda demostrado que el uso del Sistema Web con Tecnología J2EE mejora la administración de la Biblioteca en la Universidad Nacional de Ucayali, con lo cual constituye una alternativa de solución al problema planteado, con un nivel de confianza del 95%.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Según el análisis realizado y sobre la base de los objetivos de la investigación se establecieron las siguientes conclusiones:

1. Se ha demostrado que la Tecnología J2EE nos ha permitido solucionar el problema sobre la administración de la biblioteca de la UNU por las características propias, que ésta Tecnología brinda.
2. El modelado de los datos de manera ordenada, consolidada y agrupada técnicamente permiten obtener de manera eficiente información necesaria y precisa en menor tiempo.
3. La metodología RUP se ha adaptado fácilmente a la naturaleza de este proyecto de investigación permitiéndonos realizar un desarrollo organizado del mismo. Además la integración con herramientas Rational nos ha permitido aprovechar potencialmente sus virtudes.
4. El uso de la notación UML para representar los requerimientos funcionales del sistema de administración bibliotecaria ha sido adecuada y flexible al permitirnos disponer de una variedad de diagramas para plasmar la representación requerida y ser interpretados posteriormente por los desarrolladores o usuarios.

5. Los Resultados de la encuesta post-test realizada a los usuarios de la Biblioteca demuestran que el uso de esta nueva tecnología contribuirá notablemente a mejorar la administración de la biblioteca de la UNU.
6. Las interfaces Web permiten brindar aplicaciones más flexibles para el manejo y la accesibilidad de la información así como se demostró en este proyecto de investigación.
7. Se demostró que existe un grado de correlación entre el Sistema Web con tecnología J2EE y la administración de la Biblioteca de la Universidad Nacional de Ucayali.
8. Es factible desarrollar un Sistema Web con tecnología J2EE para la administración de Bibliotecas, con características de horizontalidad.

6.2. RECOMENDACIONES

De acuerdo a las conclusiones de la investigación realizada se recomienda lo siguiente:

1. Aplicar Rational Unified Process y UML para la construcción de Sistemas Web con integración de la Tecnología J2EE.
2. Realizar las pruebas necesarias en lo que respecta al entorno de trabajo para que este tipo de sistemas funcionen en su totalidad.
3. Implantar el Sistema Web con Tecnología J2EE para la administración en la biblioteca, por los beneficios que se obtendrían al respecto.
4. Promover trabajos de investigación utilizando la tecnología J2EE y la Ingeniería de software que permitan solucionar problemas sociales y culturales.
5. Promover trabajos de investigación para tratar de llevar a la práctica los estudios económicos para el desarrollo de Ingeniería de Software, con la finalidad de establecer un estándar.
6. La adquisición de hardware, software y accesorios de red deben tener un adecuado nivel de tecnología sincronizada que permitan satisfacer las necesidades de procesamiento que el Sistema Web con Tecnología J2EE para la administración en la biblioteca requiera para garantizar un óptimo desempeño.

REFERENCIAS BIBLIOGRÁFICAS

1. Morales B, Callizaya N. La Virtualización de la Información en la Red de Bibliotecas Universitarias de San Andrés. 2003; [5 páginas]. Disponible en: <http://eprints.rclis.org/archive/00001091/01/lapaz17.pdf>. Consultado Noviembre 15, 2007.
2. Universidad de Colima. Suplemento Interfaces [En Línea]. 2005; [1 página]. Disponible en: <http://www.ucol.mx/publicaciones/interfaces/>. Consultado Diciembre 11, 2007.
3. About Avanti Library Systems [En Línea]. 2004; [1 página]. Disponible en: <http://www.avantilibrarysystems.com/about.html>. Consultado Diciembre 11, 2007.
4. Pytheas [En Línea]. 2004. [1 página]. Disponible en: <http://web2.uwindsor.ca/library/leddy.people/art/pytheas/>. Consultado en Diciembre 11, 2007.
5. El Modelo Cliente Servidor [En Línea]. 2006. [3 páginas]. Disponible en: <http://agamenon.uniandes.edu.co/~revista/articulos/cliser.html>. Consultado en Enero 8, 2008.
6. Cliente/Servidor [En Línea]. 2006. [10 páginas]. Disponible en: <http://www.csi.map.es/csi/silice/Global71.html>. Consultado en Enero 8, 2008.

7. Weitzenfeld A, Ingeniería de Software Orientada a Objetos con UML, Java e Internet. 1era ed. Madrid España: Cengage Learning Editores; 2005.
8. Tecnología Cliente Servidor [En Línea]. 2004. [15 páginas]. Disponible en:
<http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/arqui.HTM>.
Consultado en Enero 11, 2008.
9. Falkner J. Fundamentos Desarrollo Web con JSP. 1era ed. Madrid España: Anaya Multimedia; 2002.
10. Introducción al Lenguaje Java [En Línea]. 2004. [23 páginas]. Disponible en: <http://www.unav.es/cti/manuales/Java/indice.html>.
Consultado en Diciembre 19, 2007
11. Mukhar K, Fundamentos de Base de Datos con Java. 1era ed. Madrid España: Anaya Multimedia; 2001.
12. Ajax: Una nueva aproximación a las aplicaciones Web [En Línea]. 2005. [3 páginas]. Disponible en: <http://www.deltablog.com/2005/03/31/ajax-una-nueva-aproximacion-a-las-aplicaciones-web/>. Consultado en Enero 30, 2008
13. Rediris [En Línea]. 2007. [3 páginas]. Disponible en: <http://www.rediris.es/rediris/boletin/58-59/ponencia13.html>. Consultado en Diciembre 19, 2007.
14. Brad J, Programación Orientada a Objetos: Un enfoque evolutivo. 1era ed. Manassas EEUU: Addison-Wesley; 1999.

15. Booch G, Rumbaugh J, Jacobson Ivar, El Lenguaje Unificado de Modelado. Manassas EEUU: Addison Wesley; 2000.
16. Servicios Open Modeling [En Línea]. 2005. [3 páginas]. Disponible en: <http://www.vico.org/ser.htm> . Consultado en Enero 19, 2008.
17. Ericsson H, Pender M, UML Toolkit. San Francisco EEUU: Wiley Computer Publishing; 1999.
18. Boggs M, Boggs W, UML with Rational Rose 2002. 1era ed. EEUU: Sybex Inc.; 2002
19. Tondreau C, Proceso de desarrollo de Aplicaciones Web basado en metodología RUP. Santiago de Chile: Pontificia Universidad Católica de Chile; 2002.
20. Genbiz [En Línea]. 2005. [4páginas]. Disponible en: <http://www.puntogen.cl/implementacion.html> . Consultado en Febrero 11, 2008.
21. Las principales Teorías administrativas y sus principales enfoques [En Línea]. 2003. [5 páginas]. Disponible en: <http://www.ilustrados.com/publicaciones/EpVAAyyFAulkJzDPoX.php> . Consultado en Febrero 8, 2008.
22. George T, Principios de Administración. México: CECSA; 1999.
23. Charles W, Gareth R, Administración Estratégica: Un enfoque Integrado. EEUU: McGraw Hill; 1996.

24. Coordinación [En Línea]. 2005. [2 páginas]. Disponible en:
<http://www.managershelp.com/coordinacion.htm> . Consultado en:
Febrero 10, 2008.
25. Reyes P, Administración Moderna. México: Editorial Limusa; 1999.
26. Tomás S, Modelo Conceptual para la automatización de bibliotecas en el contexto digital. España: Universidad de Murcia; 2002.
27. Bryson J. Técnicas de gestión para bibliotecas y centros de información. Madrid España: Pirámide; 1999.
28. Crawford W, Gorman M, Future Libraries: dreams, madness and reality. Chicago: American Library Association; 1999
29. Brophy , P. The library in the twenty-first century: new services for the information age. London: Library Association Publishing; 2001.
30. Rodríguez M, Información, tecnología y complejidad. Scire; 1999.
31. Ponjuán D, Gestión de información en las organizaciones: principios, conceptos y aplicaciones. Chile: Centro de Capacitación en Información, Universidad de Chile; 1999.
32. Tamayo M, El Proceso de la Investigación Científica. Editorial Limusa; 2001.

ANEXOS

ANEXO 1: CÁLCULO DE ESFUERZO Y COSTO PARA EL PRESUPUESTO DEL PROYECTO

1. Peso de los actores

Primero empezamos considerando los actores de nuestro sistema y determinamos para cada Actor si estos son simples, promedio o complejos; para esto nos guiamos de la siguiente tabla:

Tipo de Actor	Descripción	Factor
Simple	Interfaz del programa	1
Promedio	Interactivo, o manejador de interfaz con protocolo	2
Complejo	Interfaz gráfica	3

Asignamos a cada actor su tipo:

- Secretaria – Promedio
- Encargado de Distribución – Promedio
- Alumno – Simple
- Docente - Simple

Por tanto:

$$2 \text{ Simple} * 1 = 2$$

$$2 \text{ Promedio} * 2 = 4$$

$$0 \text{ Complejo} * 3 = 0$$

$$\text{Total de peso de actores} = 2 + 4 + 0 = 6$$

2. Peso de los Use Case

Ahora hacemos algo similar para la lista de Use Case; con la diferencia que esto basado en el Número de transacciones que realiza cada Use Case. Determinando si estos son simples, promedios o complejos.

Tipo de Actor	Descripción	Factor
Simple	3 o menos Transacciones	5
Promedio	4 a 7 Transacciones	10
Complejo	7 Transacciones	15

Asignamos a cada use case su tipo:

1.-	Registrar material bibliográfico	Simple
2.-	Registrar Prestamos	Promedio
3.-	Registrar autor	Simple
4.-	Registrar editorial	Simple
5.-	Registrar materia	Simple
6.-	Registrar devolución	Simple
7.-	Consultar material bibliográfico	Simple
8.-	Registrar Reservas	Simple

Entonces:

$$7 \text{ simple} * 5 = 35$$

$$1 \text{ promedio} * 10 = 10$$

$$0 \text{ complejo} * 15 = 0$$

$$\text{Total de peso de use case} = 35 + 10 + 0 = 45$$

3. Calculando UUCP

Para encontrar el Ajuste de Puntos para el Use Case (UUCP); el cual refleja la complejidad del proyecto y la experiencia de las personas en el proyecto, para estos utilizamos los pesos de los actores y de los use case:

$$4 + 45 = 49 \text{ UUCP}$$

4. Calculando el TCF

Ahora necesitamos calcular la complejidad técnica para este proyecto, a esto se le llama Factor Técnico de Complejidad (TFC).

Para calcular el TFC lo hacemos a través de la siguiente tabla, que llenamos con factores de 0 a 5 un puntaje de 0 significa que el factor es irrelevante, un puntaje de 5 significa que el factor es significativo para este proyecto:

Numero de factor	Descripción del Factor	Peso de Factor	Valor Asignados	Valor Total
T1	Sistema Distribuido	2	5	10
T2	Respuesta o Rendimiento de los objetivos cumplidos	1	4	4
T3	Eficiencia de los Usuarios Finales (en Línea)	1	5	5
T4	Procesamiento interno complejo	1	3	3
T5	Código debe ser reusable	1	5	5
T6	Fácil de instalar	0.5	5	2.5
T7	Fácil de usar	0.5	5	2.5
T8	Portable	2	5	10
T9	Fácil de cambiar	1	5	5
T10	Concurrente	1	2	2
T11	Incluye características especiales de seguridad	1	4	4
T12	Provee acceso directo para terceros	1	1	1
T13	Capacitación especial	1	3	3

$TFactor = \text{Sumatoria (Peso del Factor)} * (T\text{Valores Asignados})$

$TFactor = 57$

$TFC = 0.6 + (0.01 * Factor)$

$TFC = 0.6 + (0.01 * 57) = 1.17$

5. Calculando el EF

En este punto calcularemos el nivel de experiencia de las personas del proyecto, a esto se llama el Factor Environment. Para calcular esto lo hacemos a través de la siguiente tabla; Teniendo en consideración los siguientes puntos:

- De F1 a F4; 0 es no experiencia, 3 es mas o menos y 5 es experto.
- F5; 0 no motivado, 3 más o menos y 5 muy motivado.
- F6; 0 requerimientos inestables, 3 más o menos y 5 requerimientos estables.
- F7; 0 no hay staff de medio tiempo, 3 más o menos y 5 todos trabajan medio tiempo.
- F8; 0 fácil uso de la programación, 3 más o menos y 5 mucha dificultad para la programación.

Número del Factor	Descripción del Factor	Peso	Valor Asignado	Valor Total
F1	Manejo de Procesos Unificados	1.5	5	7.5
F2	Experiencia en Aplicaciones	0.5	4	2
F3	Experiencia en la Orientación a Objetos.	1	5	5
F4	Capacidad de Análisis y Liderazgo	0.5	5	2.5
F5	Motivación	1	5	5
F6	Requerimientos estables	2	4	8
F7	Trabajadores a medio tiempo	-1	1	-1
F8	Dificultad en el lenguaje de Programación	-1	0	0

$E_{Factor} = \text{Sumatoria (Valor Asignado * Peso del Factor)}$

$E_{Factor} = 29$

$EF = 0.35 + (0.03 * E_{Factor})$

$EF = 0.35 + (0.03 * 29) = 1.22$

6. Calculando el UCP

Finalmente para calcular los puntos de Use Case;

$UCP = UUCP * TCF * EF$

$UCP = 49 * 1.17 * 1.22 = 69.94$

7. Para elegir el factor hombre / horas

Para esto examinamos los datos en los EF y contamos del F1 a F6 los factores que son menores a tres y contamos de F7 a F8 son a partir de tres. Si el total es 2 o menos utilizamos 20 hombres/horas por UCP, si son mayores a tres usamos 28 hombres/horas por UCP.

En nuestro caso utilizaremos 20 hombres/horas, por lo que multiplicaremos; $20 \text{ hombres/horas} * 69.94 \text{ UCP} = 1398.8$, que consideramos que es el esfuerzo que vamos a necesitar para el proyecto.

Con esto también podemos calcular el tiempo aproximado que necesitaremos para el Proyecto; considerando que la semana tiene 40 horas (5 días * 8) entonces:

$1398.8 / 40 = 34.97$ Semanas, entre dos personas que desarrollarán éste proyecto, el tiempo calculado es de 17 semanas.

8. El costo del Proyecto se calculó; en base a un sueldo mensual para el integrante del Equipo (S/. 950 c/u) que multiplicado por el tiempo estimado para dicho proyecto (4.25 meses) Hacen un total de S/. 8000.00 aproximadamente por los 2 programadores; a este costo se le suma los gastos de aprovisionamiento que hace un total de S/.8900, llegando así a un **Costo Total Estimado de S/.16900.00** por todo el Proyecto.

ANEXO 2: PRESUPUESTO O COSTO DEL PROYECTO

Cantidad	Unidad	Descripción	Costo	
			Unitario S/.	Total S/.
6	meses	Movilidad Local	180.00	1,080.00
		PARTIDA 20	SubTotal	1,080.00
6	global	Telefonía Celular	20.00	120.00
		PARTIDA 36	SubTotal	120.00
5	meses	Honorarios Programadores(2)	1,600.00	8,000.00
		PARTIDA 39	SubTotal	8,000.00
5	millares	Papel Bond A4	20.00	100.00
20	millares	Papel Bulky	5.00	100.00
		PARTIDA 49	Subtotal	200.00
1	und	Licencia de MSSQLSERVER 2000	4,500.00	4,500.00
1	und	Licencia WINSERVER 2003	3,000.00	3,000.00
1	und	Licencia NetBeans	0.00	0.00
1	und	Licencia Apache Tomcat	0.00	0.00
		PARTIDA 51	Subtotal	7,500.00
			TOTAL S/.	16,900.00

Resumen del monto de inversión

Grupo de Gastos	Descripción	Monto
20.00	Viáticos y Asignaciones	1,080.00
36.00	Telefonía Celular	120.00
39.00	Otros Servicios de Terceros	8,000.00
49.00	Materiales de Escritorio	200.00
51.00	Equipos y Bienes Duraderos	7,500.00
	TOTAL S/.	16,900.00

ANEXO 3: FUENTES DE FINANCIAMIENTO

Grupo de Gastos	Descripción	Fuente		TOTAL
		Recursos Propios	UNU	
20.00	Viáticos y Asignaciones	1,080.00	-	1,080.00
36.00	Alimentos para personas	120.00	-	120.00
39.00	Honorarios Programadores	1,200.00	6,800.00	8,000.00
49.00	Materiales de Escritorio	200.00	-	200.00
51.00	Equipos y Bienes Duraderos		7,500.00	7,500.00
TOTAL SI.		2,600.00	14,300.00	16,900.00
En Porcentaje		15.38%	84.62%	100%

ANEXO 4: PRUEBAS DEL SISTEMA

Registrar Préstamo

<u>Item</u>	<u>Datos</u>	<u>Tipo de Datos</u>	<u>Respuesta</u>
1	Ingresar Código Usuario	(&5-/%\$	No se encontró en la Base de Datos
2	Ingresar Código Material	% "7/"* -+	No se encontró en la Base de Datos

Registrar Asistencia

<u>Item</u>	<u>Datos</u>	<u>Tipo de Datos</u>	<u>Respuesta</u>
1	Ingresar Código Usuario	(%-=?^"+	No se pudo encontrar al usuario
2	Tipo Asistencia	"_"	Falta que seleccione el tipo

ANEXO 5: ENCUESTA SOBRE LA INTERFAZ APLICADA A USUARIOS DE LA BIBLIOTECA CENTRAL DE LA UNIVERSIDAD NACIONAL DE UCAYALI

Código de encuesta: EI-0001

Código del encuestador: YTG3003

Nro de Encuesta: _____

Fecha: _____

Encuesta para la Interfaz

Indicaciones Previas:

Buen día estimado Sr.(a), a continuación le presentamos una serie de preguntas que pretenden evaluar el grado de satisfacción respecto a la interfaz del Sistema Web implantado en la Biblioteca Central de la Universidad Nacional de Ucayali, por favor sírvase contestarlas marcando sólo una de las respuestas:

P1.- ¿Le parece difícil el acceso al sistema?

- 1) Si 2) No

P2.- ¿Es entendible el menú de opciones?

- 1) Si 2) No

P3.- ¿Es sencillo el ingreso de datos?

- 1) Si 2) No

P4.- ¿Las respuestas de las búsquedas son entendibles?

- 1) Si 2) No

P5.- ¿Es fácil la navegación en el sistema?

- 1) Si 2) No

P6.- ¿Los colores que se utilizan en la interfaz le irritan la vista?

- 1) Si 2) No

P7.- ¿Está de acuerdo con la distribución del menú?

- 1) Si 2) No

P8.- ¿Los reportes son fáciles de comprender?

- 1) Si 2) No

¡La Universidad Nacional de Ucayali le agradece su valiosa colaboración!

ANEXO 6: ENCUESTA PRE – TEST APLICADA A LOS ESPECIALISTAS DE LA BIBLIOTECA CENTRAL DE LA UNIVERSIDAD NACIONAL DE UCAYALI

Código de encuesta: EE0001

Código del encuestador: EMN0911

Nro de Encuesta: _____

Fecha: _____

Encuesta Pre – Test Aplicada a especialistas de la Biblioteca.

Indicaciones Previas

Buen día Sr. (a), a continuación le presentamos unas preguntas que pretenden medir como se maneja la información dentro de la Biblioteca Central actualmente, por favor sírvase contestar las siguientes preguntas, marcando con un círculo donde crea conveniente:

P1.- ¿Qué le parece la forma en que se maneja actualmente la información en la Biblioteca Central?

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P2.- ¿Qué le parece el nivel de atención que se brinda a los usuarios en la Biblioteca Central?

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P3.- El registro y almacenamiento de los materiales bibliográficos le parecen:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P4.- El tiempo que demora en consultar un material bibliográfico con información dispersa le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P5.- El tiempo de demora y su facilidad en el procesamiento de información de los materiales bibliográficos actualmente le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P6.- El tiempo de demora y su facilidad en elaborar consultas de asistencia de alumnos consideran que es:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P7.- El tiempo de demora y su facilidad en elaborar consultas de préstamos de materiales bibliográficos consideran que es:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P8.- El tiempo de demora y su facilidad en elaborar consultas de lecturas de materiales bibliográficos le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P9.- La forma de controlar el ingreso de usuarios a la Biblioteca Central le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

¡La Universidad Nacional de Ucayali le agradece su valiosa colaboración!

ANEXO 7: ENCUESTA POST – TEST APLICADA A LOS ESPECIALISTAS DE LA BIBLIOTECA CENTRAL DE LA UNIVERSIDAD NACIONAL DE UCAYALI

Código de encuesta: EE0002

Código del encuestador: EMN0911

Nro de Encuesta: _____

Fecha: _____

Encuesta Post – Test Aplicada a especialistas de la Biblioteca.

La Universidad Nacional de Ucayali está desarrollando un Sistema Web para la Biblioteca Central que permitirá tener toda la información de los materiales bibliográficos, préstamos realizados, asistencias de los usuarios y reserva en línea en una base de datos centralizada a la que podrá tener acceso para realizar consultas y procesar datos. Éste Sistema pretende mejorar la administración de la Biblioteca Central de la Universidad Nacional de Ucayali.

Por favor sírvase contestar las siguientes preguntas, marcando con un círculo donde crea conveniente:

P1.- ¿Qué le parece la nueva forma en que se maneja la información en la Biblioteca Central?

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P2.- ¿Qué le parece el nivel de atención que se brinda actualmente a los usuarios en la Biblioteca Central?

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P3.- El registro y almacenamiento de los materiales bibliográficos en una base de datos centralizada le parecen:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P4.- El tiempo que demora en consultar un material bibliográfico con el sistema le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P5.- El tiempo de demora y su facilidad en el procesamiento de información de los materiales bibliográficos con el sistema le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P6.- El tiempo de demora y su facilidad en elaborar consultas de asistencia de alumnos con el sistema le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P7.- El tiempo de demora y su facilidad en elaborar consultas de préstamos de materiales bibliográficos con el sistema le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P8.- El tiempo de demora y su facilidad en elaborar consultas de lecturas de materiales bibliográficos con el sistema le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P9.- La forma de controlar el ingreso de usuarios a la Biblioteca Central con el sistema le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

¡La Universidad Nacional de Ucayali le agradece su valiosa colaboración!

ANEXO 8: ENCUESTA PRE – TEST APLICADA A LOS USUARIOS DE LA BIBLIOTECA CENTRAL DE LA UNIVERSIDAD NACIONAL DE UCAYALI

Código de encuesta: EU-0001

Código del encuestador: DML2108

Nro de Encuesta: _____

Fecha: _____

Encuesta Pre – Test Aplicada a los usuarios de la Biblioteca.

Indicaciones Previas

Buen día Sr. (a), a continuación le presentamos unas preguntas que pretenden medir como se manejan los procesos dentro de la Biblioteca Central **actualmente**, por favor sírvase contestar las siguientes preguntas, marcando con un círculo donde crea conveniente:

P1.- El tiempo que demora en consultar un material bibliográfico actualmente, le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P2.- El tiempo de demora y su facilidad para realizar el préstamo de un material bibliográfico, le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P3.- La atención dentro de la Biblioteca Central de la Universidad Nacional de Ucayali, le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P4.- El registro de asistencia a Biblioteca Central de la Universidad Nacional de Ucayali, actualmente, le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P5.- La reserva de material bibliográfico en la Biblioteca Central de Ucayali actualmente le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

¡La Universidad Nacional de Ucayali le agradece su valiosa colaboración!

ANEXO 9: ENCUESTA POST – TEST APLICADA A LOS USUARIOS DE LA BIBLIOTECA CENTRAL DE LA UNIVERSIDAD NACIONAL DE UCAYALI

Código de encuesta: EU-0002

Código del encuestador: DML2108

Nro de Encuesta: _____

Fecha: _____

Encuesta Post – Test Aplicada a los usuarios de la Biblioteca.

La Universidad Nacional de Ucayali está desarrollando un **Sistema Web para la Biblioteca** que permitirá controlar los préstamos, asistencias, consulta y reserva en línea de material bibliográfico realizados por los usuarios. Éste Sistema pretende mejorar la administración dentro de la Biblioteca Central de la Universidad Nacional de Ucayali.

Por favor sírvase contestar las siguientes preguntas, marcando con un círculo donde crea conveniente:

P1.- El tiempo que demora en consultar un material bibliográfico **con el sistema**, le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P2.- El tiempo de demora y su facilidad para realizar el préstamo de un material bibliográfico **con el sistema**, le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P3.- La atención dentro de la Biblioteca Central de la Universidad Nacional de Ucayali **con el sistema**, le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P4.- El registro de asistencia a Biblioteca Central de la Universidad Nacional de Ucayali **con el sistema**, le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P5.- La reserva de material bibliográfico en la Biblioteca Central de Ucayali **con el sistema** le parece:

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

¡La Universidad Nacional de Ucayali le agradece su valiosa colaboración!

ANEXO 10: ENCUESTA PRE – TEST APLICADA A LOS ADMINISTRADORES DE LA BIBLIOTECA CENTRAL DE LA UNIVERSIDAD NACIONAL DE UCAYALI

Código de encuesta: EA0001

Código del encuestador: JLV1408

Nro de Encuesta: _____

Fecha: _____

Encuesta Pre – Test Aplicada a los administradores de la Biblioteca.

Indicaciones Previas

Buen día Sr. (a), a continuación le presentamos unas preguntas que pretenden recolectar la información necesaria para realizar el diagnóstico de la administración en la Biblioteca Central así como para encontrar soluciones que conduzcan a su mejoramiento.

Para diligenciar el formulario, usted debe en cada valor, realizar lo siguiente:

Ubicar (calificar) con base en su **actual** percepción de la realidad y entre un rango de excelente a pésimo, cada una de las diversas formas de manifestación del valor allí propuestas.

Para ello encontrará, al final de cada manifestación, unas alternativas para consignar (marcando con una X) una calificación cualitativa.

I. PLANEACION.

P1.- Profundo conocimiento de la realidad organizacional y de las causas que la determinan

P1.- Desconocimiento frecuente de la realidad organizacional (Fortalezas, Debilidades, Amenazas y Oportunidades) y de las causas internas y externas que la determinan

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

P2.- Planeación permanente, de todos los días, debido a la necesidad de mantener a la organización siempre un paso delante

P2.- Planeación por lo general periódica (cada año, cada dos años, etc.)

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

P3.- Planeación basada con información en tiempo real, integra y altamente confiable.

P3.- Planeación por lo general en base a información desactualizada, sesgada y poco confiable.

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

II. ORGANIZACIÓN.

P4.- Se establecen y definen niveles de autoridad y responsabilidad dentro de la organización.

P4.- Los niveles de autoridad y responsabilidad en la organización no están bien definidos o no se establecen.

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

P5.- Las actividades y recursos de la organización se coordinan racionalmente para facilitar el trabajo y la eficiencia.

P5.- No se utilizan adecuadamente los recursos para facilitar el trabajo y la eficiencia de la organización.

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

III. DIRECCION

P6.- Los objetivos de todos los departamentos y secciones se relacionan armoniosamente para lograr el objetivo general de la organización.

P6.- Cada departamento trabaja por sus objetivos propios sin importarle los objetivos generales de la organización.

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

P7.- El dirigente proporciona apoyo y comunicación a sus subordinados durante la ejecución de los planes, de tal manera que éstos se realicen con mayor facilidad.

P7.- El dirigente deja a sus subordinados a la deriva durante la ejecución de los planes, complicando la ejecución de éstos.

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

IV. COORDINACION

P8.- Cada departamento conoce de la existencia de los otros. Marcha según los fines y objetivos comunes, preocupándose de los departamentos adyacentes y de la organización en conjunto.

P8.- Cada departamento ignora o quiere ignorar la existencia de otros. Marcha como si tuviera en sí su mismo fin y su razón de ser, sin preocuparse de los departamentos adyacentes ni del conjunto de la organización.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P9.- Todos piensan en el interés general. Teniendo siempre presente la iniciativa y la abnegación.

P9.- Nadie piensa en el interés general. La iniciativa y la abnegación se hallan ausentes.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

V. CONTROL

P10.- Se utilizan medios efectivos para medir las operaciones de la organización.

P10.- Las operaciones de la organización no se miden.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P11.- En base a los resultados del control se toman ajustes o medidas necesarias para mejorar los procesos de la organización.

P11.- Los resultados del control no influyen en lo más mínimo para tomar medidas o ajustes en los procesos de la organización.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

¡La Universidad Nacional de Ucayali le agradece su valiosa colaboración!

**ANEXO 11: ENCUESTA POST – TEST APLICADA A LOS
ADMINISTRADORES DE LA BIBLIOTECA CENTRAL DE LA UNIVERSIDAD
NACIONAL DE UCAYALI**

Código de encuesta: EA0002

Código del encuestador: JLV1408

Nro de Encuesta: _____

Fecha: _____

Encuesta Post – Test Aplicada a los administradores de la Biblioteca.

Indicaciones Previas

Buen día Sr. (a), a continuación le presentamos unas preguntas que pretenden recolectar la información para realizar el diagnóstico de la administración en la Biblioteca Central, luego de la instalación del Sistema Web con Tecnología J2EE.

Para diligenciar el formulario, usted debe en cada valor, realizar lo siguiente:

Ubicar (calificar) con base en su **actual** percepción de la realidad y entre un rango de excelente a pésimo, cada una de las diversas formas de manifestación del valor allí propuestas.

Para ello encontrará, al final de cada manifestación, unas alternativas para consignar (marcando con una X) una calificación cualitativa.

I. PLANEACION.

P1.- Profundo conocimiento de la realidad organizacional y de las causas que la determinan

P1.- Desconocimiento frecuente de la realidad organizacional (Fortalezas, Debilidades, Amenazas y Oportunidades) y de las causas internas y externas que la determinan

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

P2.- Planeación permanente, de todos los días, debido a la necesidad de mantener a la organización siempre un paso adelante

P2.- Planeación por lo general periódica (cada año, cada dos años, etc.)

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

P3.- Planeación basada con información en tiempo real, integra y altamente confiable.

P3.- Planeación por lo general en base a información desactualizada, sesgada y poco confiable.

1) Muy Bueno

2) Bueno

3) Regular

4) Malo

5) Muy Malo

II. ORGANIZACIÓN.

P4.- Se establecen y definen niveles de autoridad y responsabilidad dentro de la organización.

P4.- Los niveles de autoridad y responsabilidad en la organización no están bien definidos o no se establecen.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P5.- Las actividades y recursos de la organización se coordinan racionalmente para facilitar el trabajo y la eficiencia.

P5.- No se utilizan adecuadamente los recursos para facilitar el trabajo y la eficiencia de la organización.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

III. DIRECCION

P6.- Los objetivos de todos los departamentos y secciones se relacionan armoniosamente para lograr el objetivo general de la organización.

P6.- Cada departamento trabaja por sus objetivos propios sin importarle los objetivos generales de la organización.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P7.- El dirigente proporciona apoyo y comunicación a sus subordinados durante la ejecución de los planes, de tal manera que éstos se realicen con mayor facilidad.

P7.- El dirigente deja a sus subordinados a la deriva durante la ejecución de los planes, complicando la ejecución de éstos.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

IV. COORDINACION

P8.- Cada departamento conoce de la existencia de los otros. Marcha según los fines y objetivos comunes, preocupándose de los departamentos adyacentes y de la organización en conjunto.

P8.- Cada departamento ignora o quiere ignorar la existencia de otros. Marcha como si tuviera en sí su mismo fin y su razón de ser, sin preocuparse de los departamentos adyacentes ni del conjunto de la organización.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P9.- Todos piensan en el interés general. Teniendo siempre presente la iniciativa y la abnegación.

P9.- Nadie piensa en el interés general. La iniciativa y la abnegación se hallan ausentes.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

V. CONTROL

P10.- Se utilizan medios efectivos para medir las operaciones de la organización.

P10.- Las operaciones de la organización no se miden.

1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

P11.- En base a los resultados del control se toman ajustes o medidas necesarias para mejorar los procesos de la organización.

P11.- Los resultados del control no influyen en lo más mínimo para tomar medidas o ajustes en los procesos de la organización.

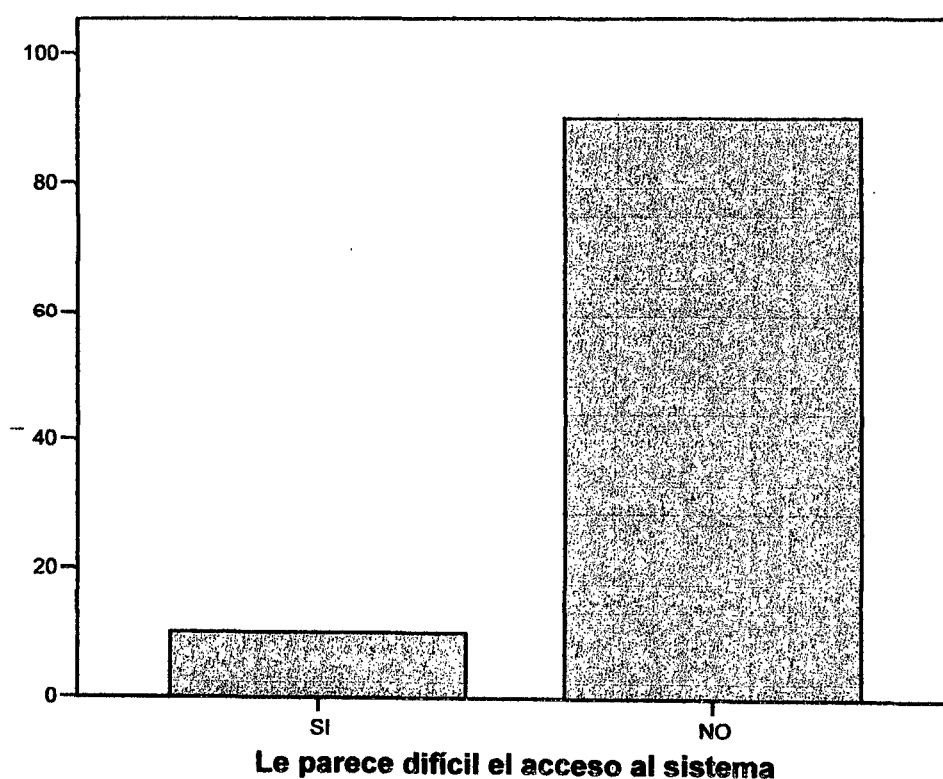
1) Muy Bueno 2) Bueno 3) Regular 4) Malo 5) Muy Malo

¡La Universidad Nacional de Ucayali le agradece su valiosa colaboración!

**ANEXO 12: RESULTADOS DE LA ENCUESTA SOBRE LA INTERFAZ
APLICADA A USUARIOS DE LA BIBLIOTECA CENTRAL DE LA
UNIVERSIDAD NACIONAL DE UCAYALI**

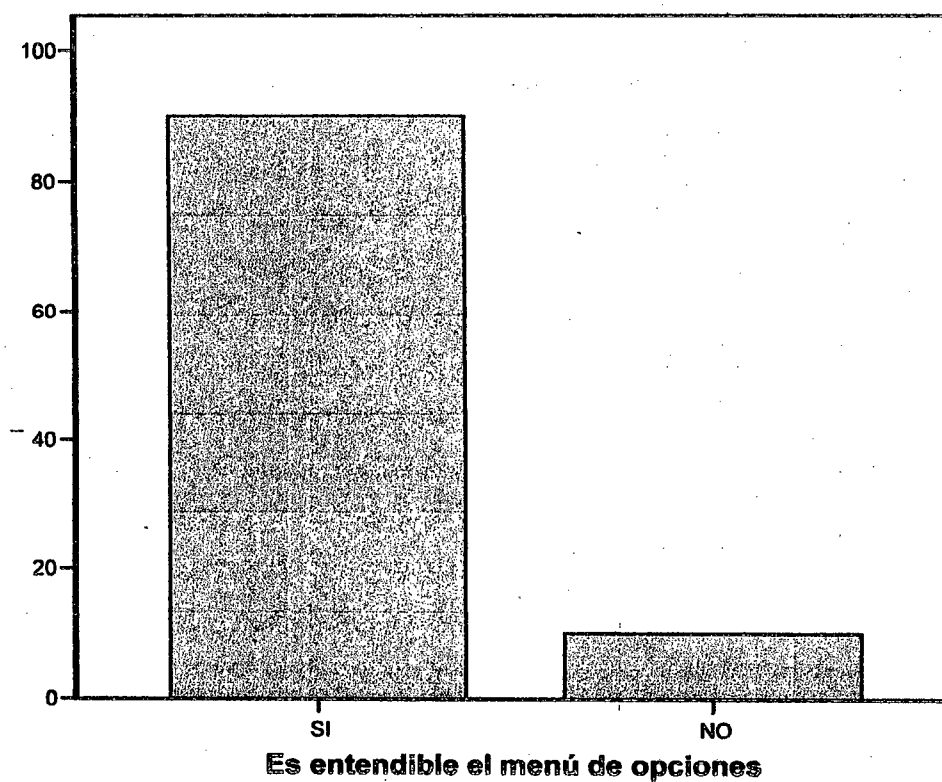
➤ **A la pregunta:** ¿Le parece difícil el acceso al sistema?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	SI	1	14,3	14,3	14,3
	NO	6	85,7	85,7	100,0
	Total	7	100,0	100,0	



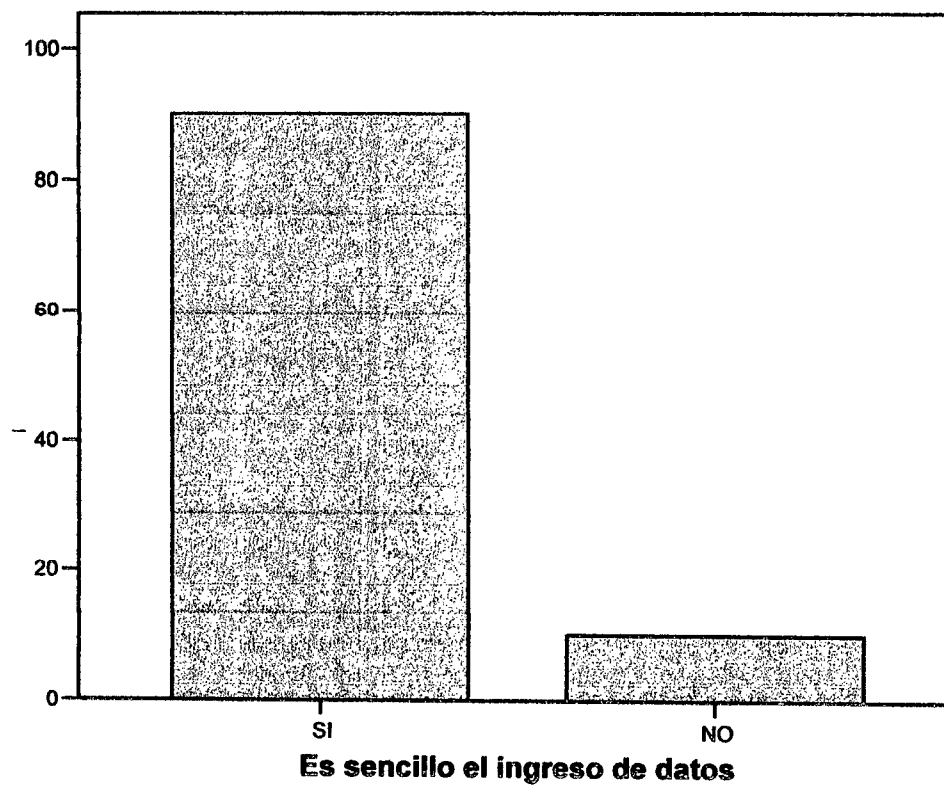
➤ A la pregunta: ¿Es entendible el menú de opciones?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	SI	6	85,7	85,7	85,7
	NO	1	14,3	14,3	100,0
	Total	7	100,0	100,0	



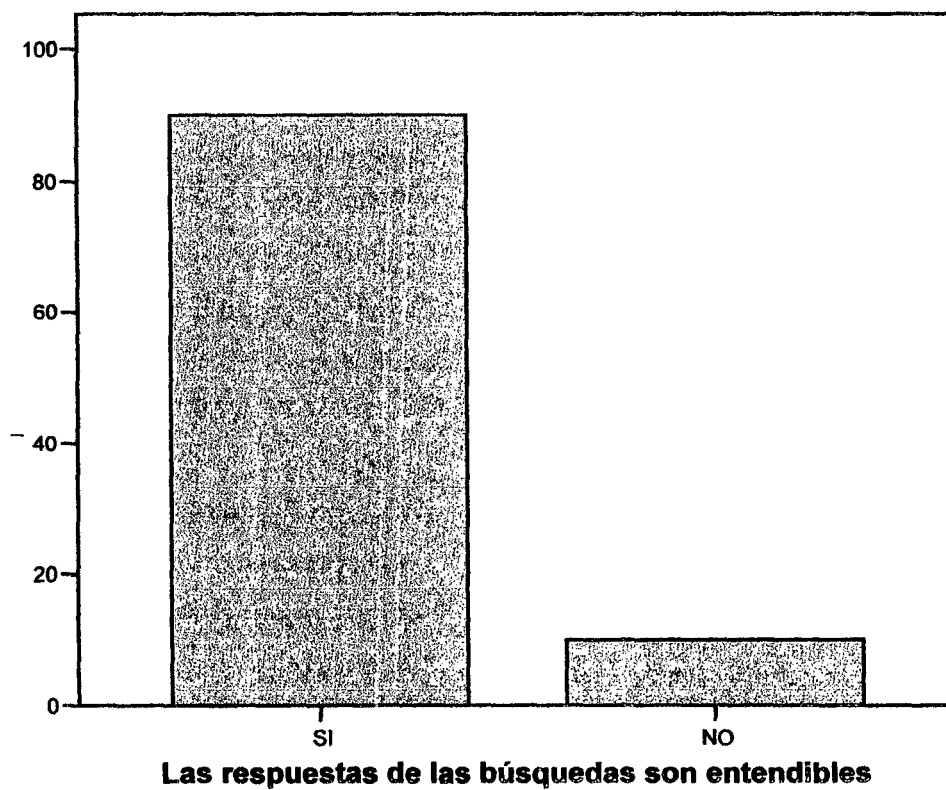
➤ **A la pregunta: ¿Es sencillo el ingreso de datos?**

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	SI	6	85,7	85,7	85,7
	NO	1	14,3	14,3	100,0
	Total	7	100,0	100,0	



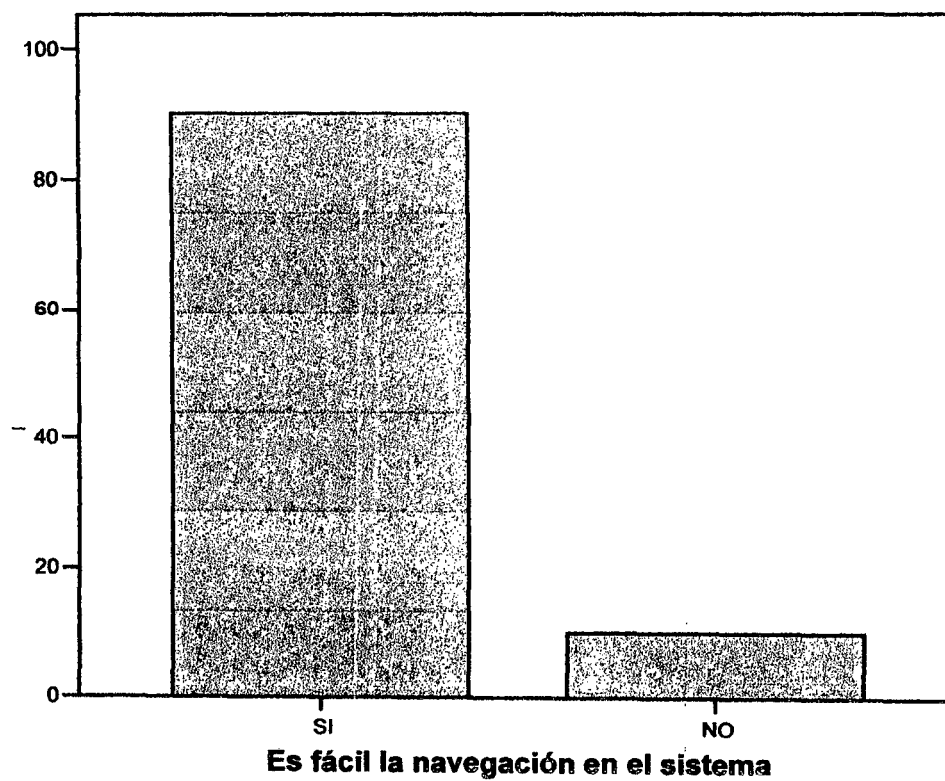
➤ **A la pregunta:** ¿Las respuestas de las búsquedas son entendibles?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	SI	6	85,7	85,7	85,7
	NO	1	14,3	14,3	100,0
	Total	7	100,0	100,0	



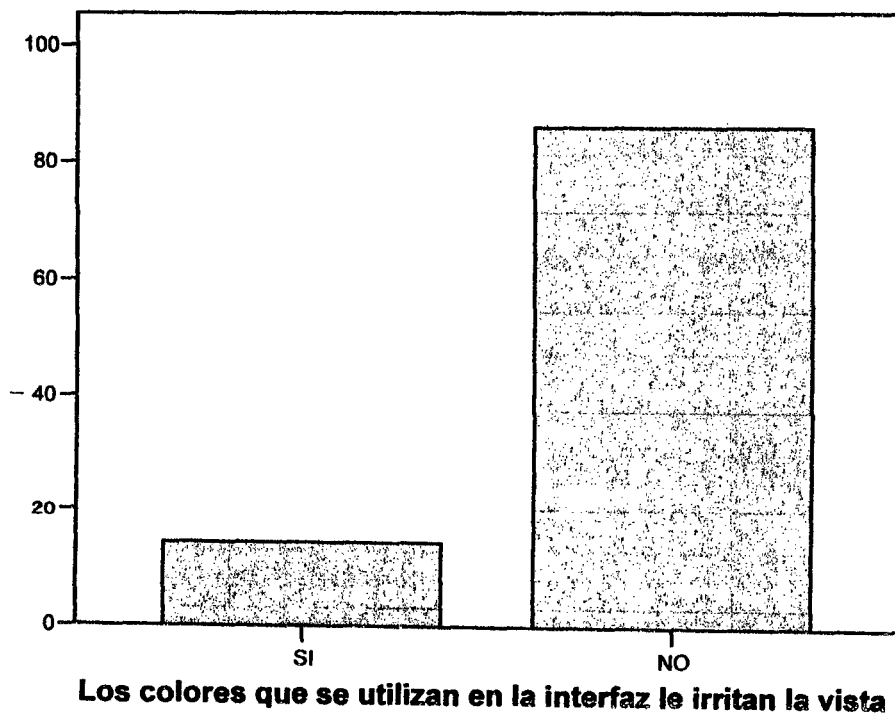
➤ **A la pregunta:** ¿Es fácil la navegación en el sistema?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	SI	6	85,7	85,7	85,7
	NO	1	14,3	14,3	100,0
	Total	7	100,0	100,0	



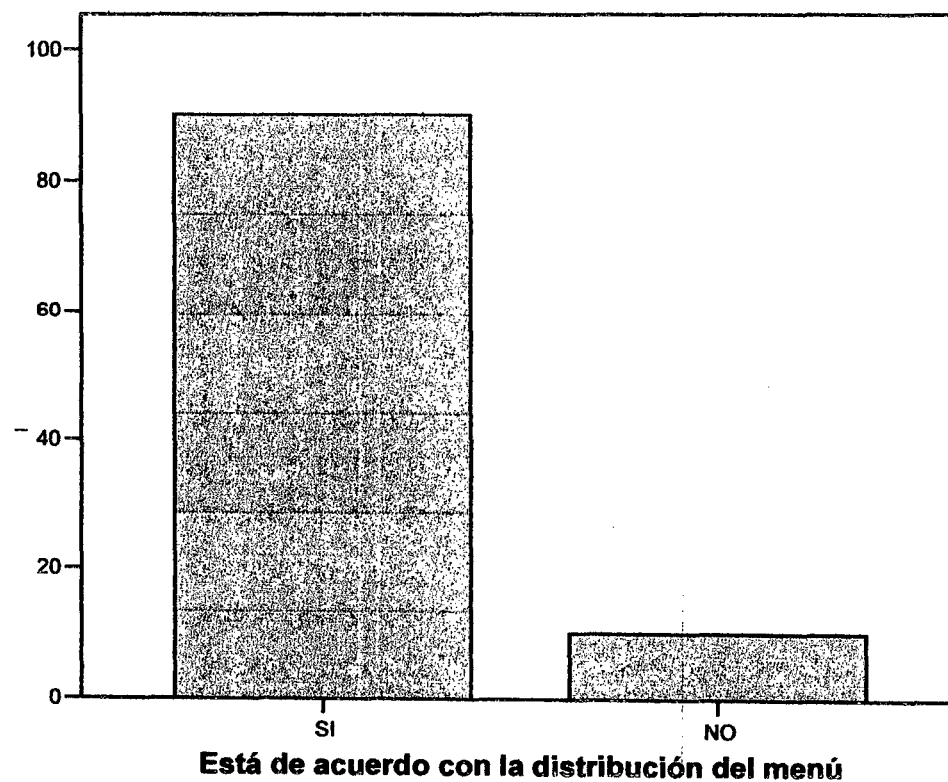
➤ **A la pregunta:** ¿Los colores que se utilizan en la interfaz le irritan la vista?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	SI	1	14,3	14,3	14,3
	NO	6	85,7	85,7	100,0
	Total	7	100,0	100,0	



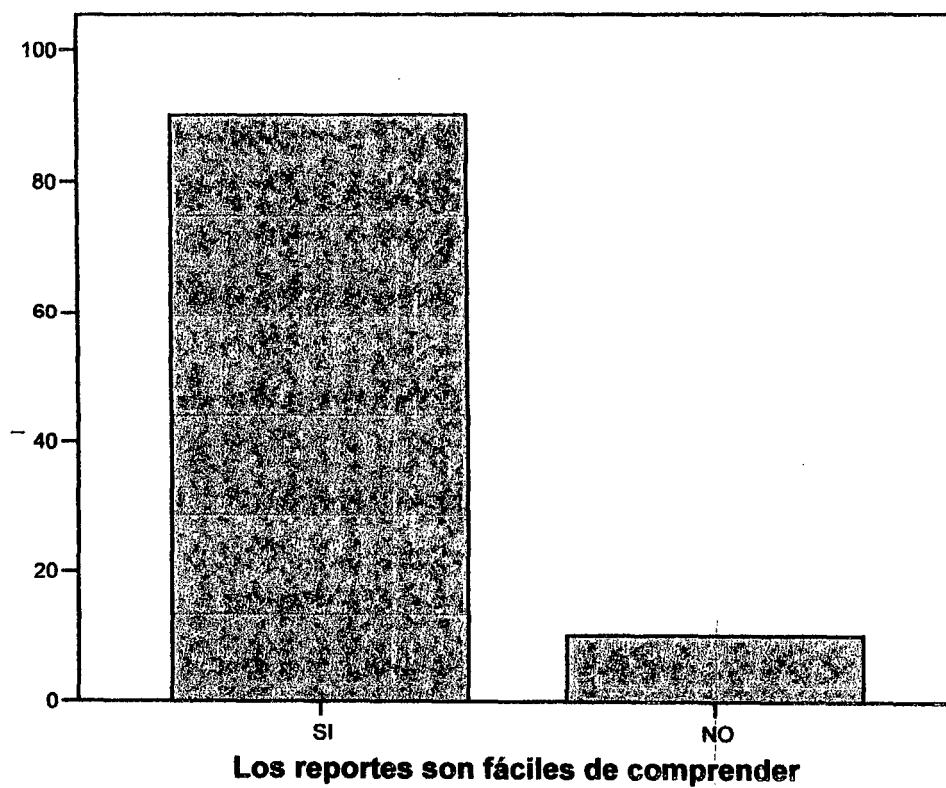
➤ **A la pregunta:** ¿Está de acuerdo con la distribución del menú?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	SI	6	85,7	85,7	85,7
	NO	1	14,3	14,3	100,0
	Total	7	100,0	100,0	



➤ **A la pregunta:** ¿Los reportes son fáciles de comprender?

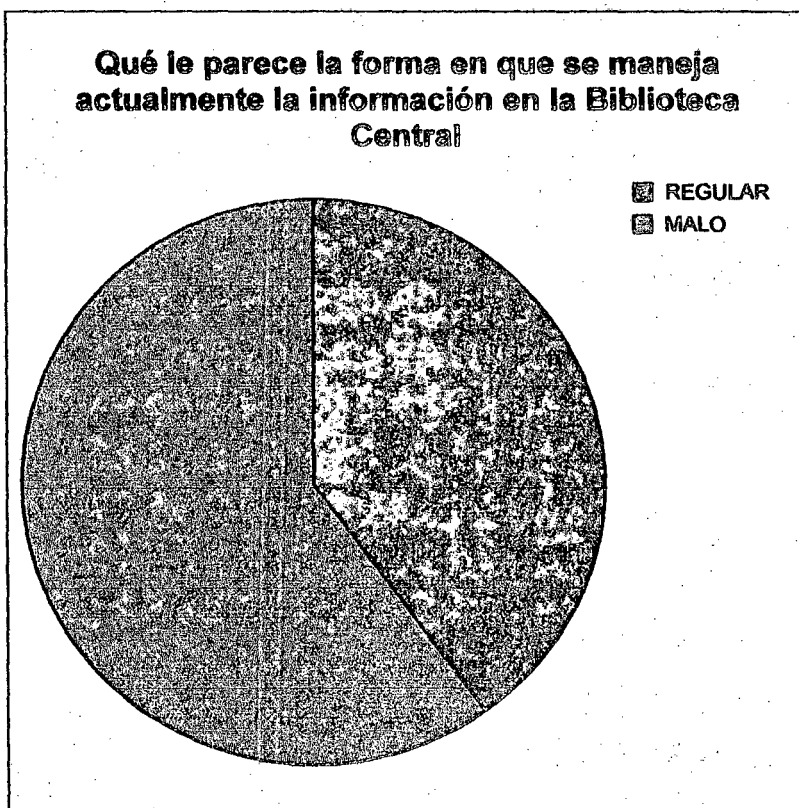
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	SI	6	85,7	85,7	85,7
	NO	1	14,3	14,3	100,0
	Total	7	100,0	100,0	



ANEXO 13: RESULTADOS DE LAS ENCUESTAS PRE – TEST

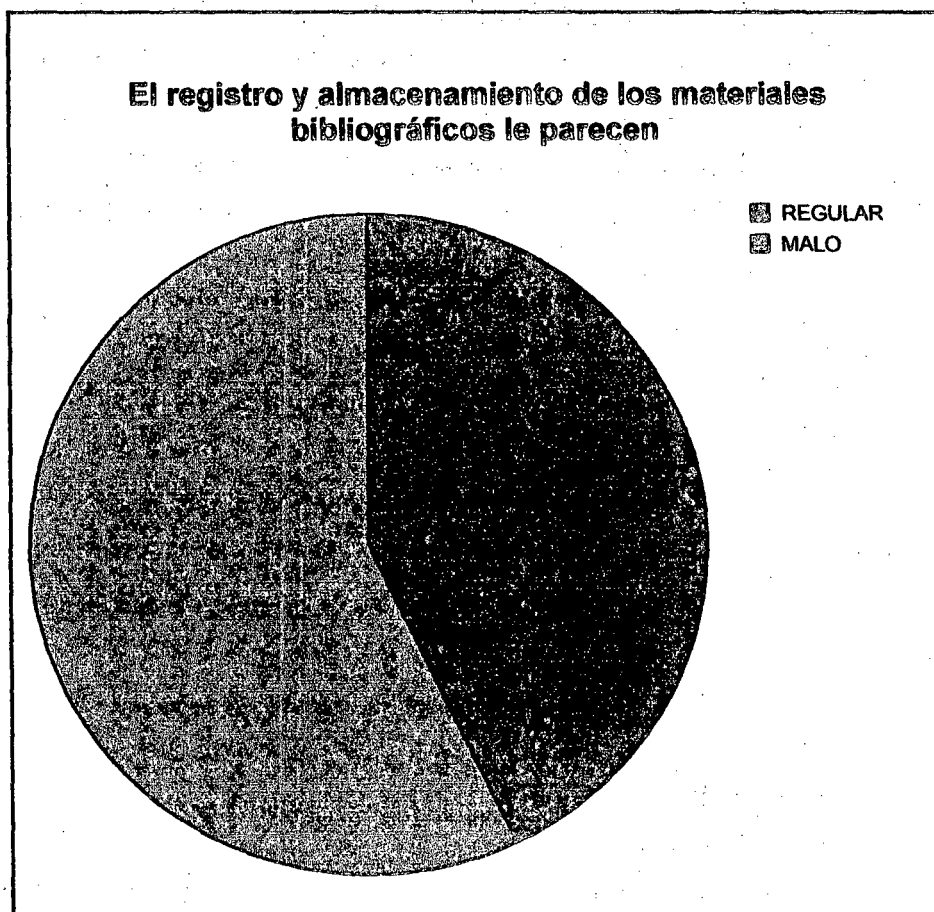
- **A la pregunta:** ¿Qué le parece la forma en que se maneja actualmente la información en la Biblioteca Central?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	REGULAR	3	42,9	42,9	42,9
	MALO	4	57,1	57,1	100,0
	Total	7	100,0	100,0	



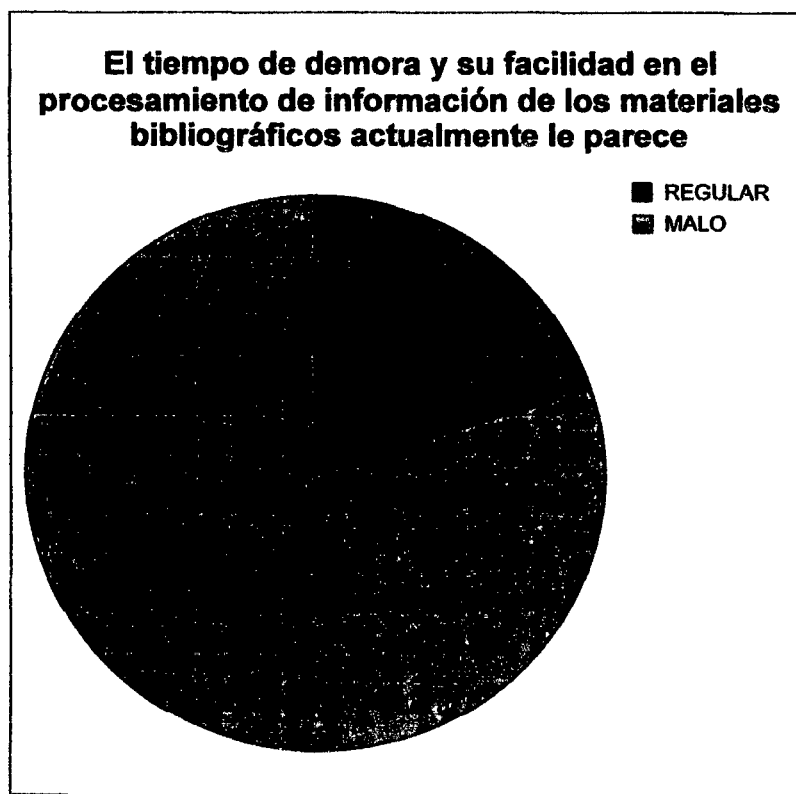
- A la pregunta: El registro y almacenamiento de los materiales bibliográficos le parecen:

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	REGULAR	3	42,9	42,9	42,9
	MALO	4	57,1	57,1	100,0
	Total	7	100,0	100,0	



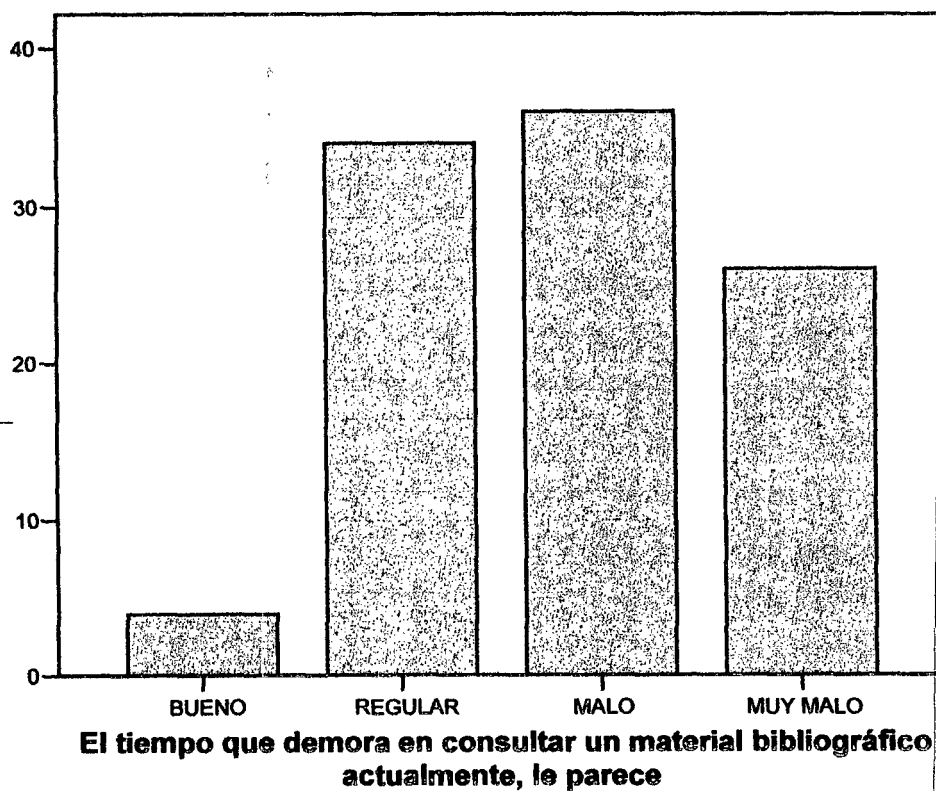
- **A la pregunta:** El tiempo de demora y su facilidad en el procesamiento de información de los materiales bibliográficos actualmente le parece:

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	REGULAR	1	14,3	14,3	14,3
	MALO	6	85,7	85,7	100,0
	Total	7	100,0	100,0	



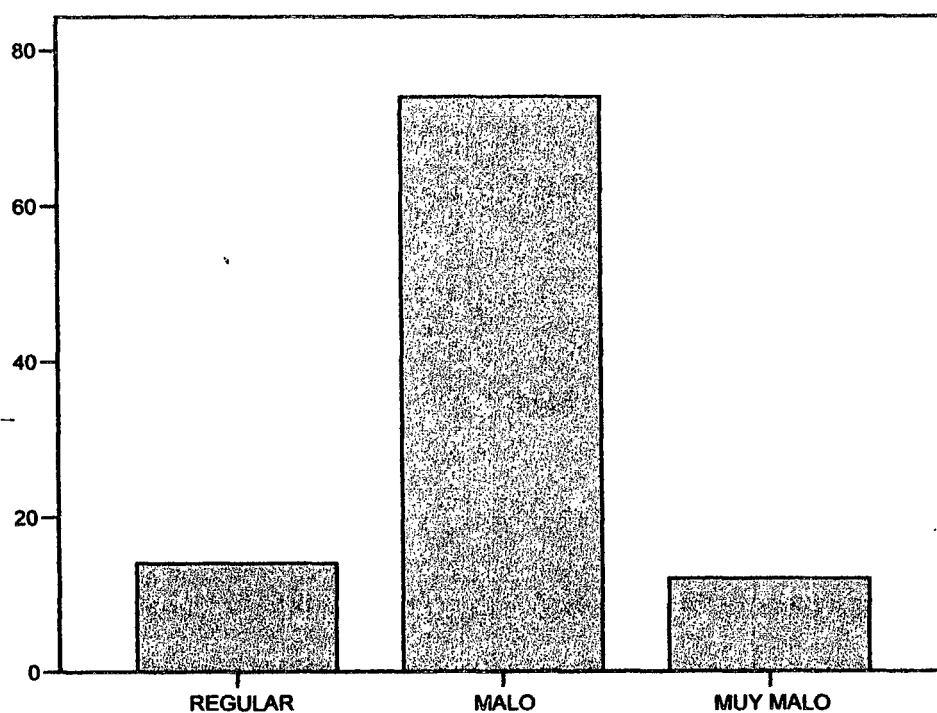
- **A la pregunta:** El tiempo que demora en consultar un material bibliográfico actualmente, le parece:

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	BUENO	20	4,0	4,0	4,0
	REGULAR	172	34,3	34,3	38,2
	MALO	180	35,9	35,9	74,1
	MUY MALO	130	25,9	25,9	100,0
	Total	502	100,0	100,0	



- **A la pregunta:** El tiempo de demora y su facilidad para realizar el préstamo de un material bibliográfico, le parece:

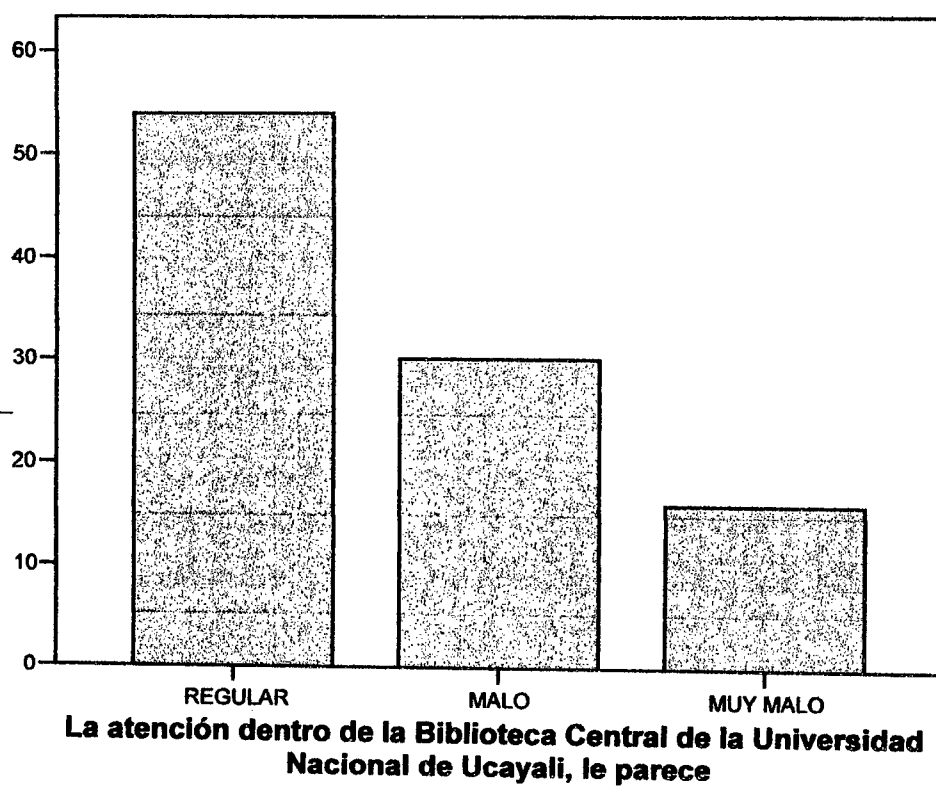
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	REGULAR	70	13,9	13,9	13,9
	MALO	371	73,9	73,9	87,8
	MUY MALO	61	12,2	12,2	100,0
	Total	502	100,0	100,0	



El tiempo de demora y su facilidad para realizar el préstamo de un material bibliográfico, le parece

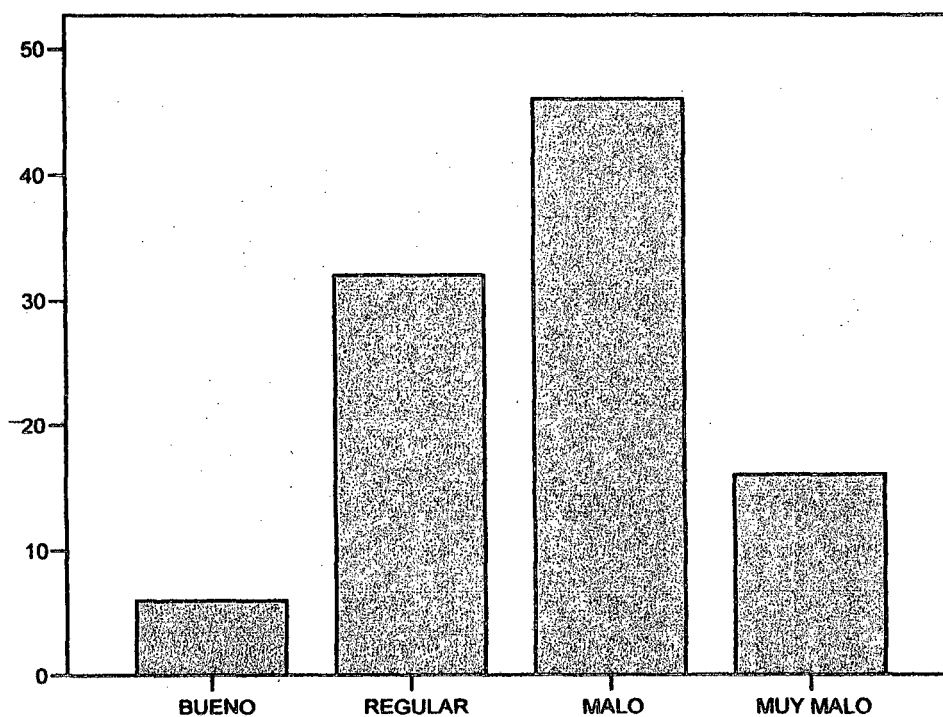
- **A la pregunta:** La atención dentro de la Biblioteca Central de la Universidad Nacional de Ucayali, le parece:

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	REGULAR	271	54,0	54,0	54,0
	MALO	150	39,9	39,9	83,9
	MUY MALO	81	16,1	16,1	100,0
	Total	502	100,0	100,0	



- **A la pregunta:** El registro de asistencia a Biblioteca Central de la Universidad Nacional de Ucayali, actualmente, le parece:

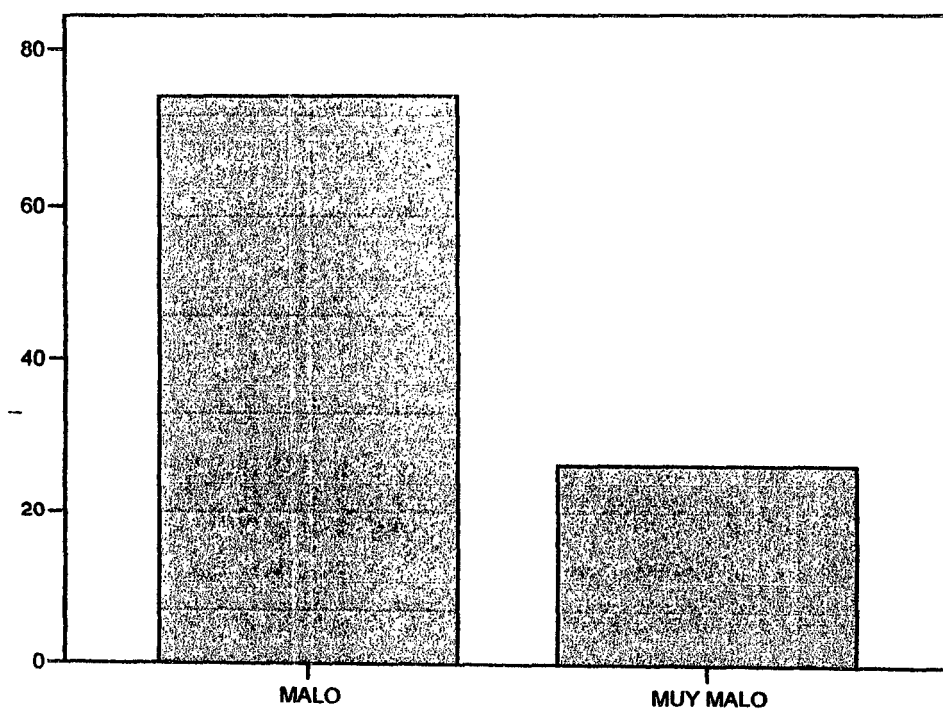
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	BUENO	30	6,0	6,0	6,0
	REGULAR	161	32,1	32,1	38,0
	MALO	231	46,0	46,0	84,1
	MUY MALO	80	15,9	15,9	100,0
	Total	502	100,0	100,0	



El registro de asistencia a Biblioteca Central de la Universidad Nacional de Ucayali, actualmente, le parece

- **A la pregunta: La reserva de material bibliográfico en la Biblioteca Central de Ucayali actualmente le parece:**

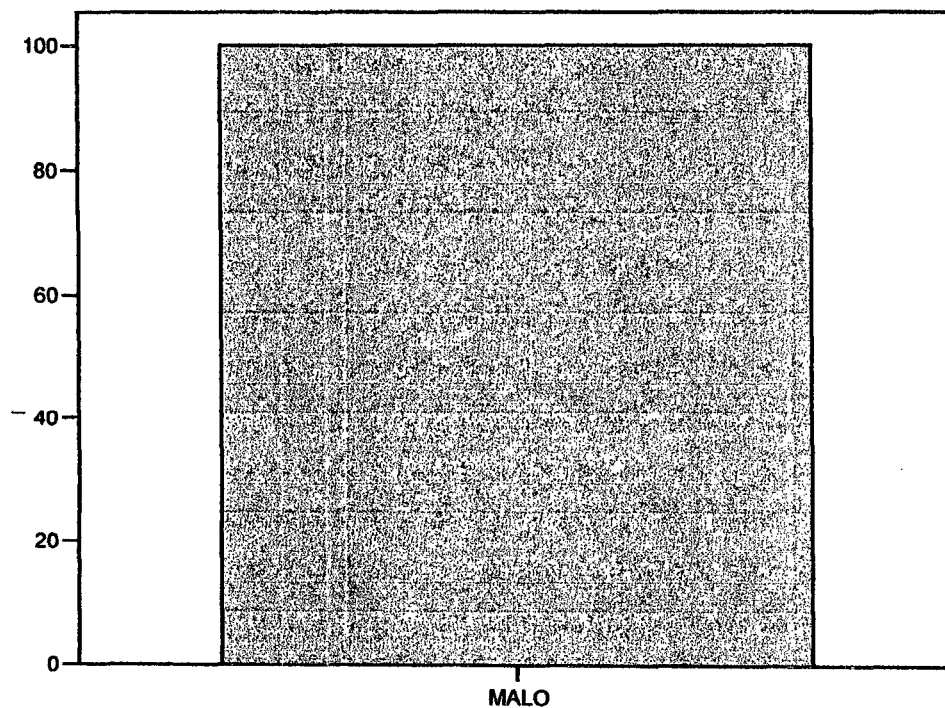
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MALO	372	74,1	74,1	74,1
	MUY MALO	130	25,9	25,9	100,0
	Total	502	100,0	100,0	



La reserva de material bibliográfico en la Biblioteca Central de Ucayali actualmente le parece

- **A la pregunta:** Planeación basada con información en tiempo real, integra y altamente confiable.

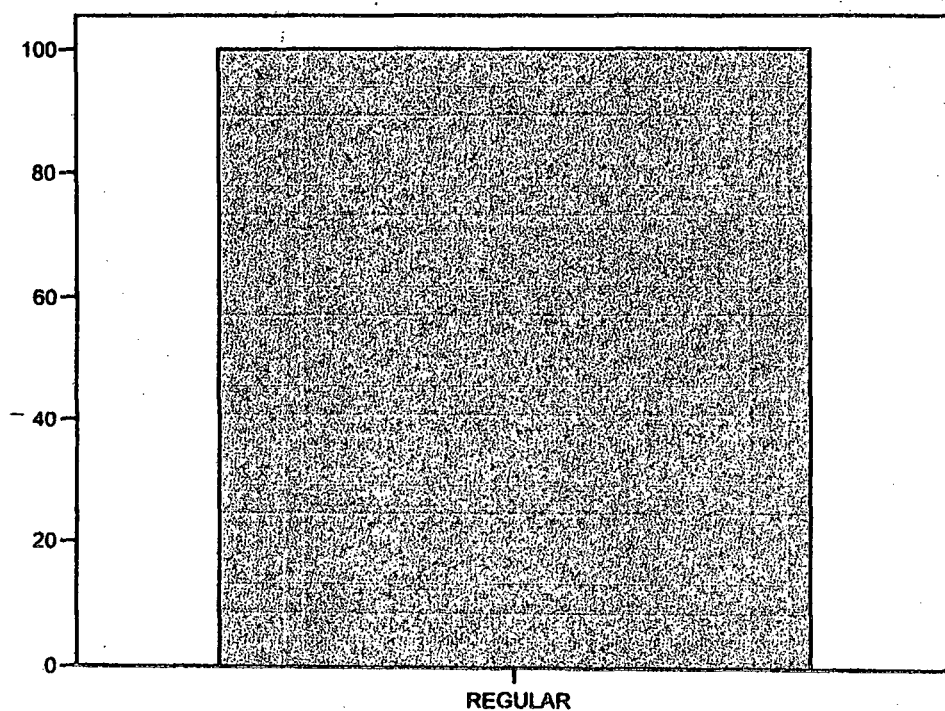
	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos MALO	1	100,0	100,0	100,0



Planeación basada con información en tiempo real, integra y altamente confiable

- **A la pregunta:** Se establecen y definen niveles de autoridad y responsabilidad dentro de la organización.

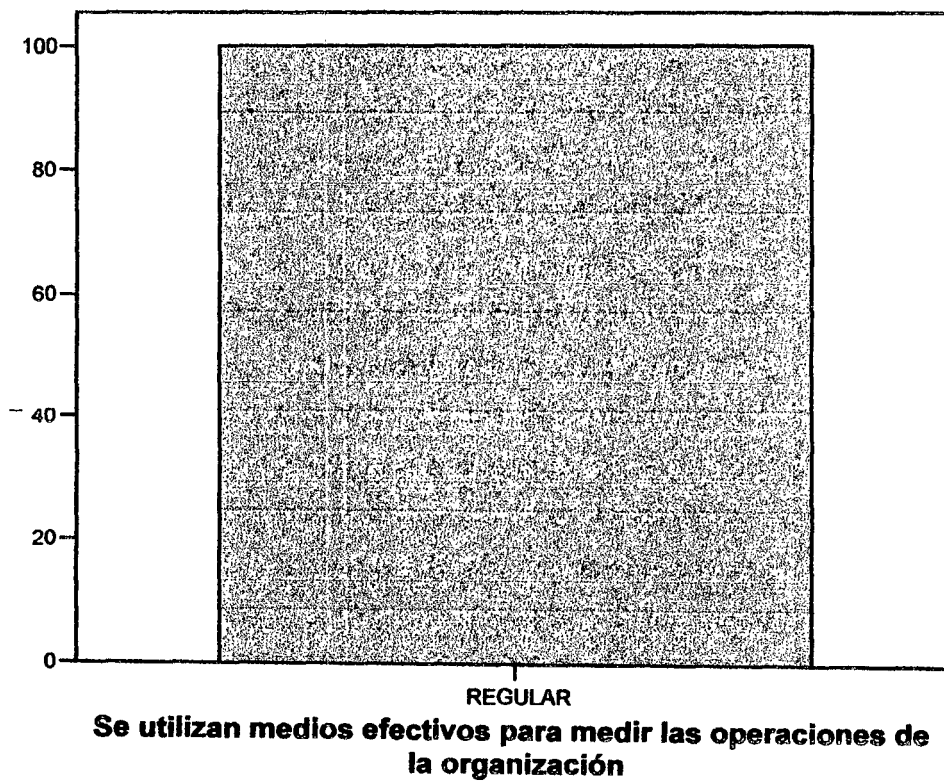
	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos REGULAR	1	100,0	100,0	100,0



Se establecen y definen niveles de autoridad y responsabilidad dentro de la organización

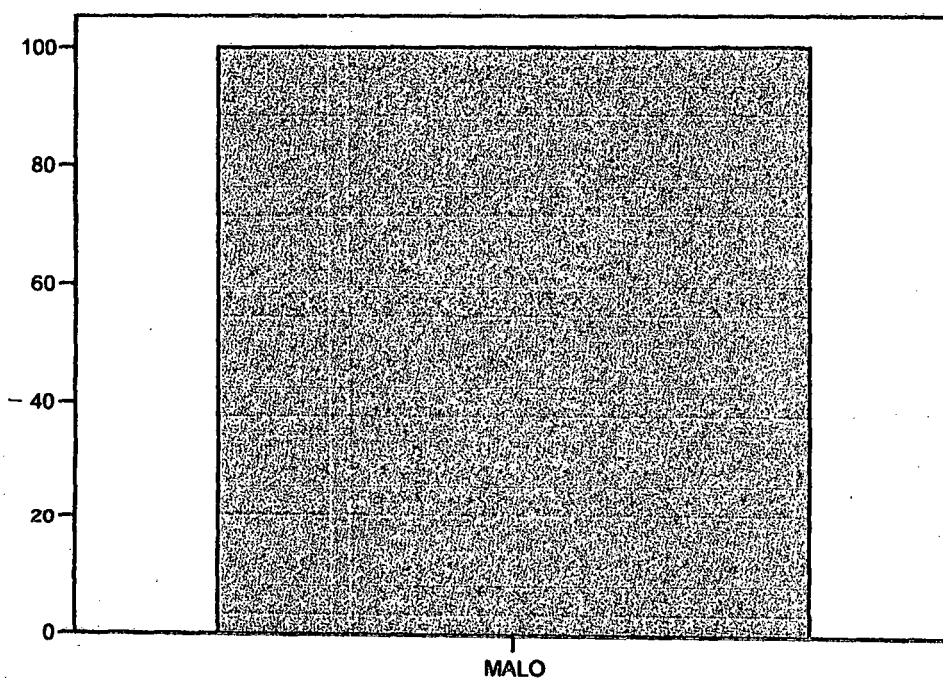
- **A la pregunta:** Se utilizan medios efectivos para medir las operaciones de la organización.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos REGULAR	1	100,0	100,0	100,0



- A la pregunta: En base a los resultados del control se toman ajustes o medidas necesarias para mejorar los procesos de la organización.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos MALO	1	100,0	100,0	100,0

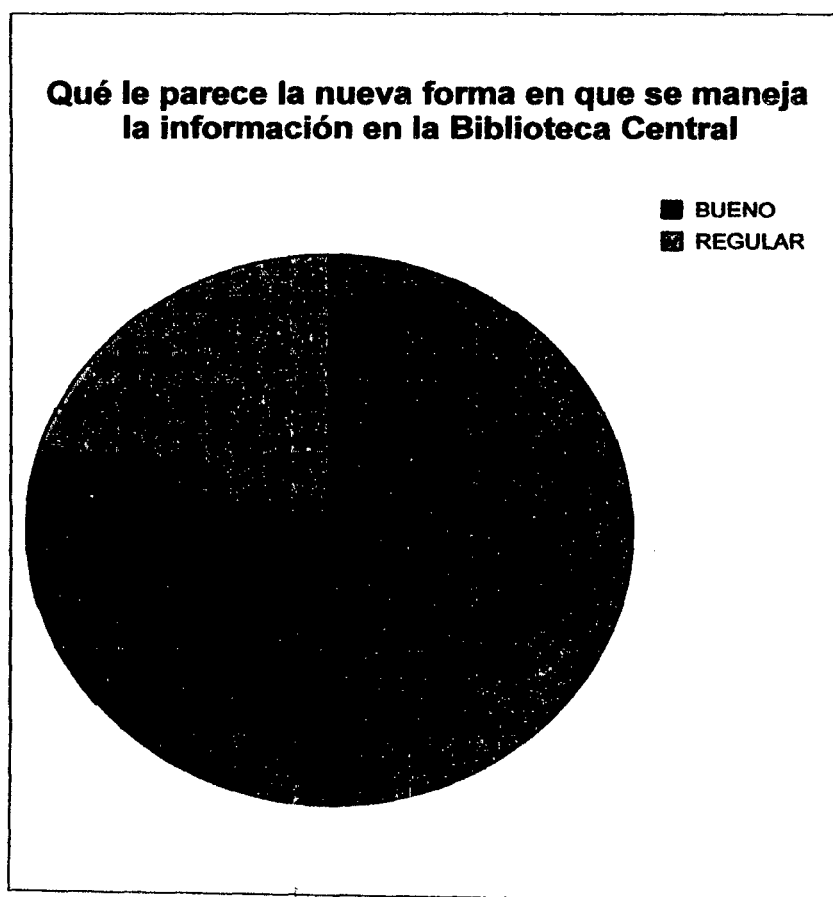


En base a los resultados del control se toman ajustes o medidas necesarias para mejorar los procesos de la organización

ANEXO 14: RESULTADOS DE LAS ENCUESTAS POST – TEST

- **A la pregunta:** ¿Qué le parece la nueva forma en que se maneja la información en la Biblioteca Central?

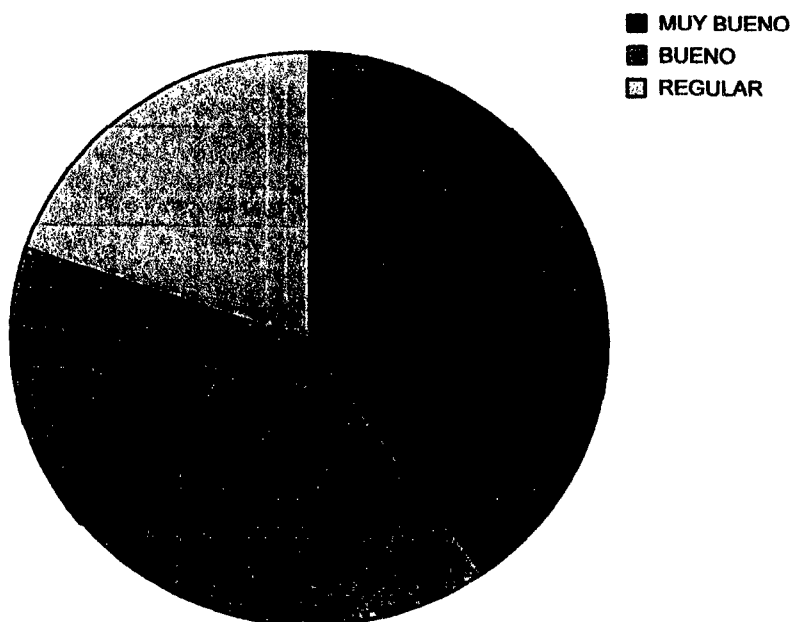
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	BUENO	6	85,7	85,7	85,7
	REGULAR	1	14,3	14,3	100,0
	Total	7	100,0	100,0	



- **A la pregunta:** El registro y almacenamiento de los materiales bibliográficos en una base de datos centralizada le parecen:

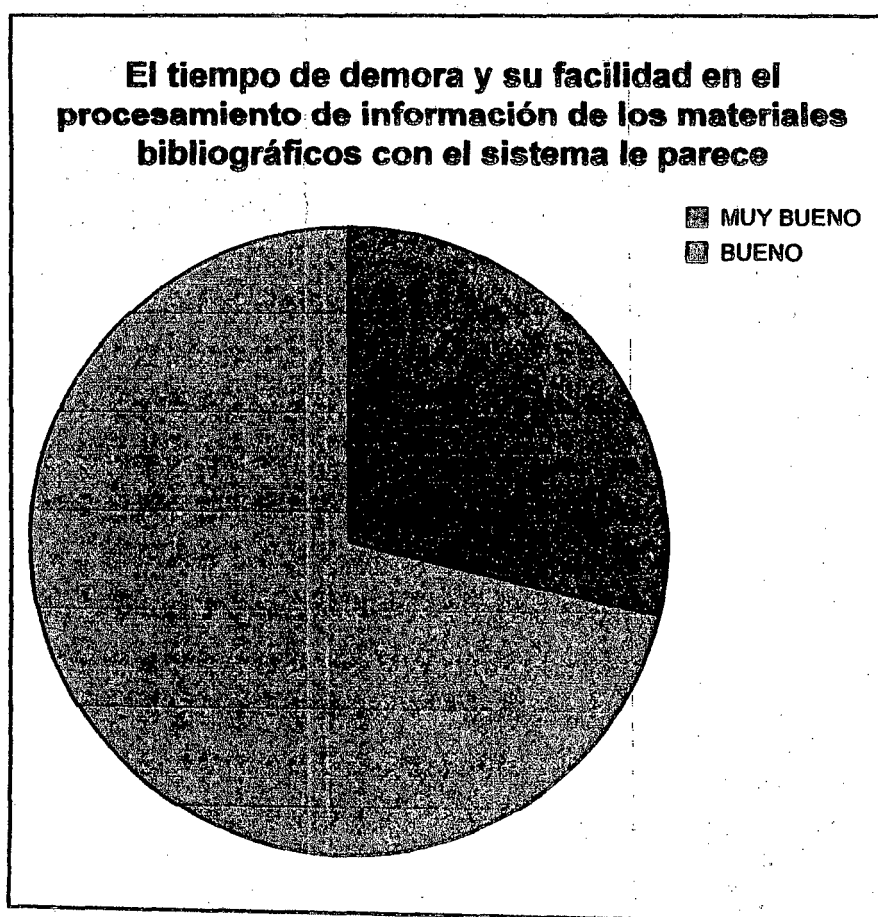
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	3	42,9	42,9	42,9
	BUENO	3	42,9	42,9	85,7
	REGULAR	1	14,3	14,3	100,0
	Total	7	100,0	100,0	

El registro y almacenamiento de los materiales bibliográficos en una base de datos centralizada le parecen



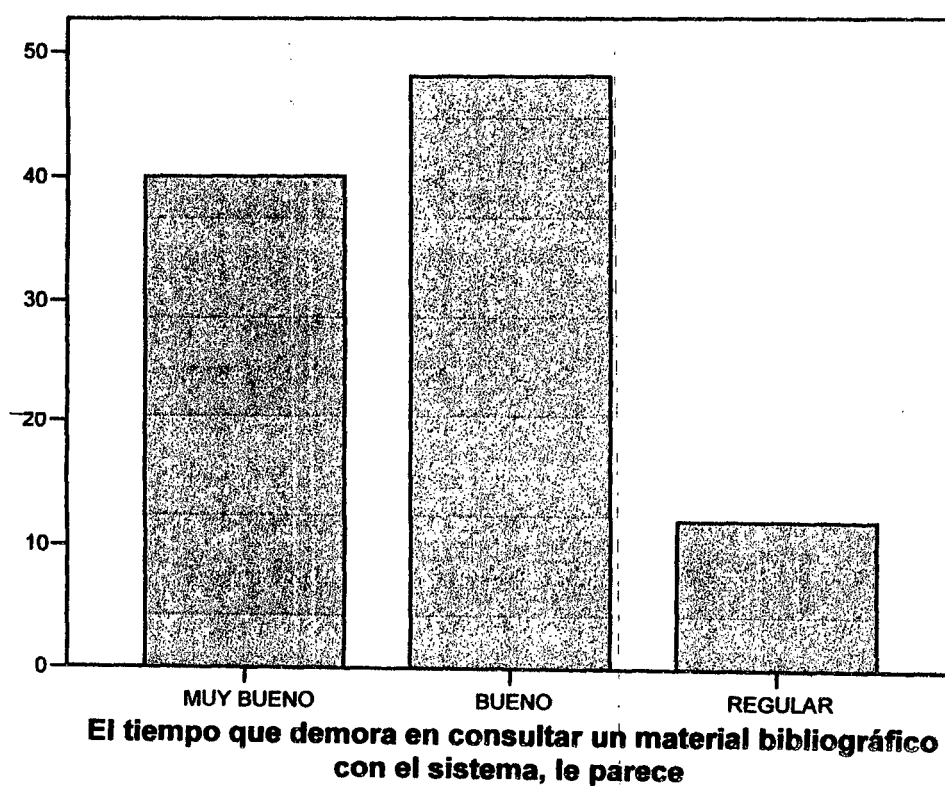
- **A la pregunta: El tiempo de demora y su facilidad en el procesamiento de información de los materiales bibliográficos con el sistema le parece:**

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	2	28,6	28,6	28,6
	BUENO	5	71,4	71,4	100,0
	Total	7	100,0	100,0	



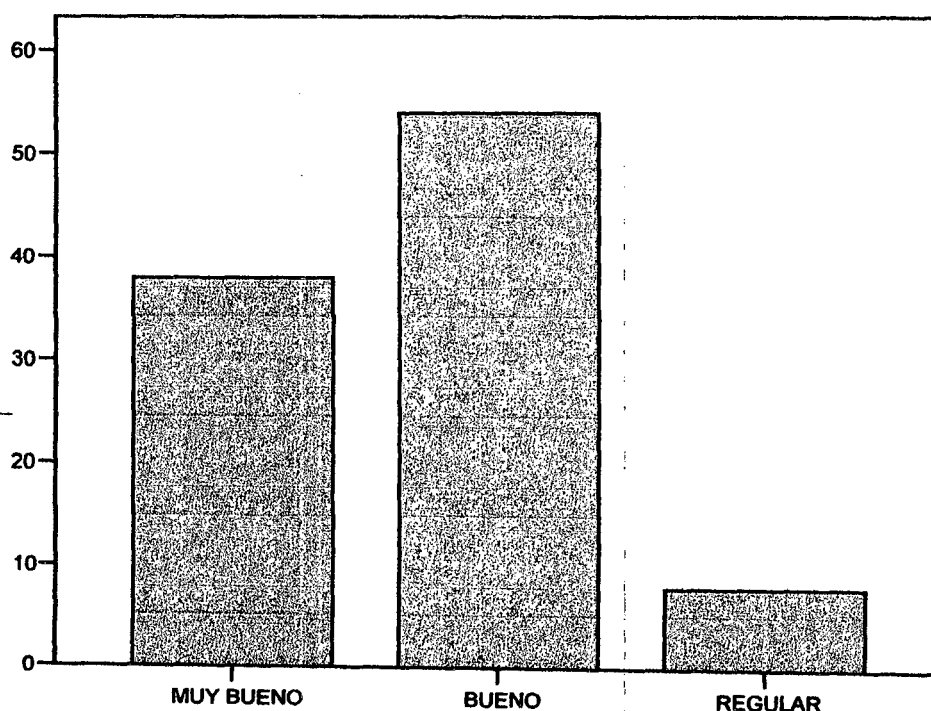
- **A la pregunta:** El tiempo que demora en consultar un material bibliográfico con el sistema, le parece:

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	200	39,8	39,8	39,8
	BUENO	242	48,2	48,2	88,0
	REGULAR	60	12,0	12,0	100,0
	Total	502	100,0	100,0	



- **A la pregunta:** El tiempo de demora y su facilidad para realizar el préstamo de un material bibliográfico con el sistema, le parece:

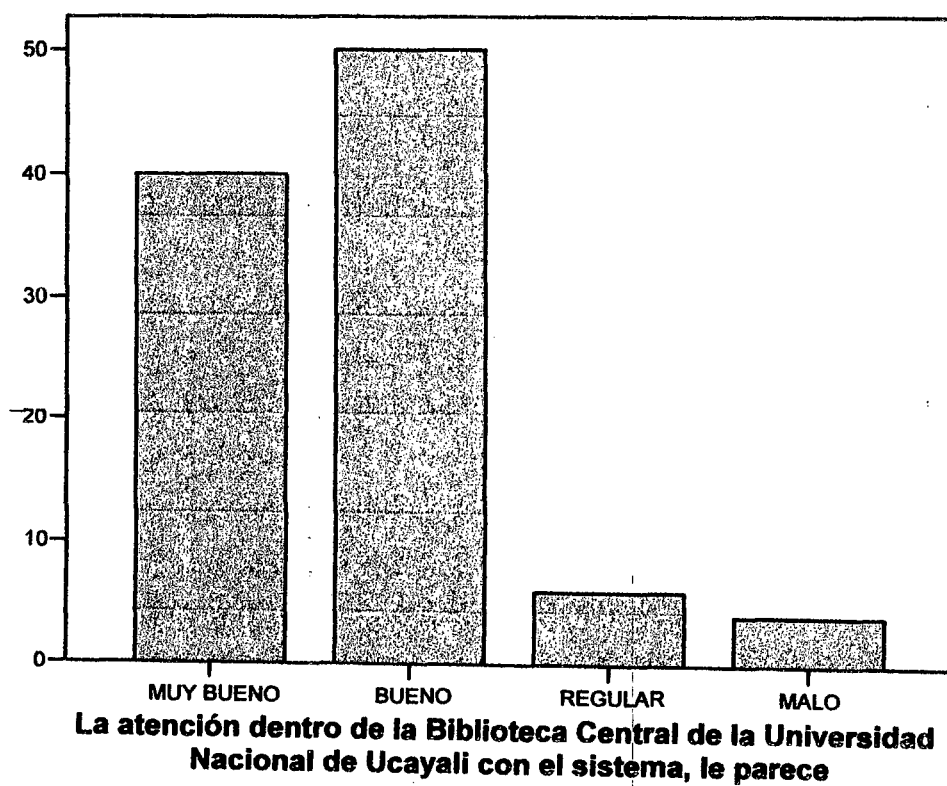
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	192	38,2	38,2	38,2
	BUENO	270	53,8	53,8	92,0
	REGULAR	40	8,0	8,0	100,0
	Total	502	100,0	100,0	



El tiempo de demora y su facilidad para realizar el préstamo de un material bibliográfico con el sistema, le parece

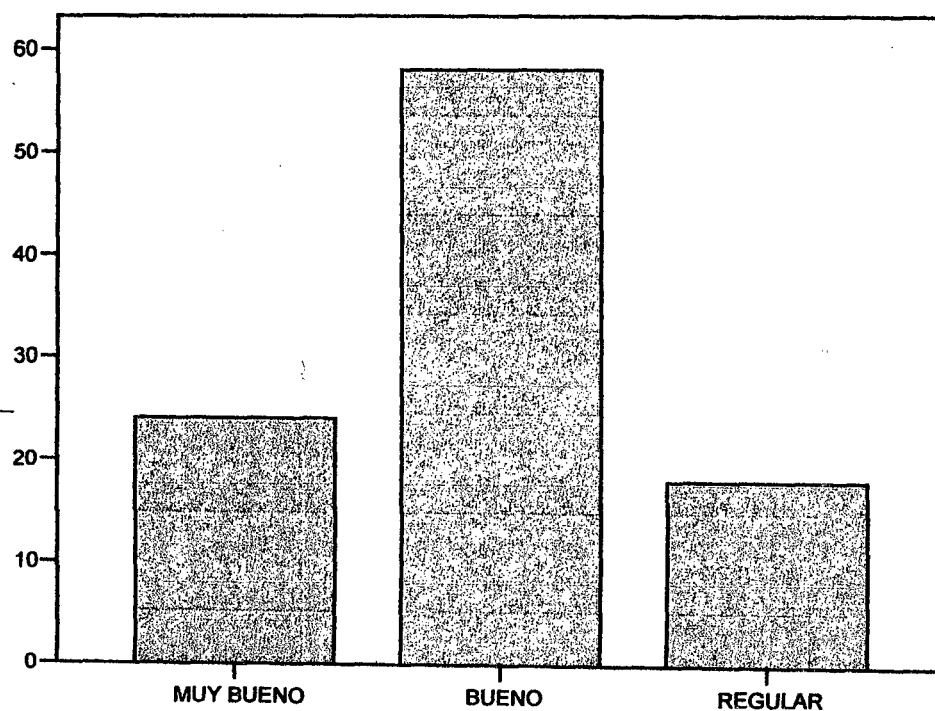
- **A la pregunta:** La atención dentro de la Biblioteca Central de la Universidad Nacional de Ucayali con el sistema, le parece:

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	200	39,8	39,8	39,8
	BUENO	251	50,0	50,0	89,8
	REGULAR	31	6,2	6,2	96,0
	MALO	20	4,0	4,0	100,0
	Total	502	100,0	100,0	



- **A la pregunta:** El registro de asistencia a Biblioteca Central de la Universidad Nacional de Ucayali con el sistema, le parece:

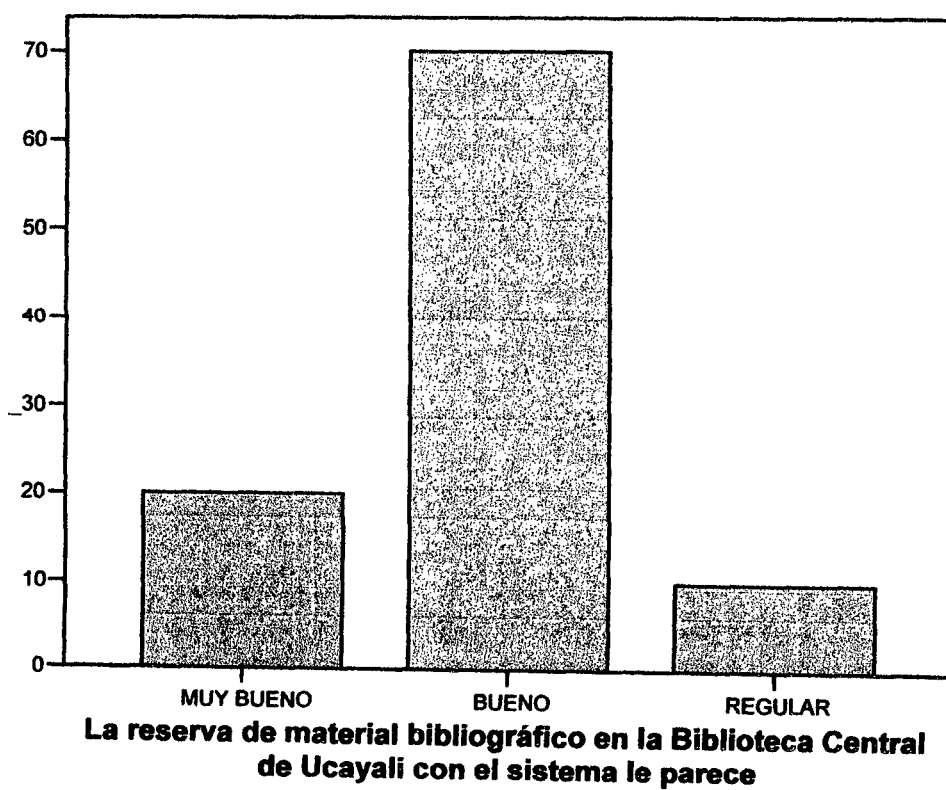
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	120	23,9	23,9	23,9
	BUENO	292	58,2	58,2	82,1
	REGULAR	90	17,9	17,9	100,0
	Total	502	100,0	100,0	



El registro de asistencia a Biblioteca Central de la Universidad Nacional de Ucayali con el sistema, le parece

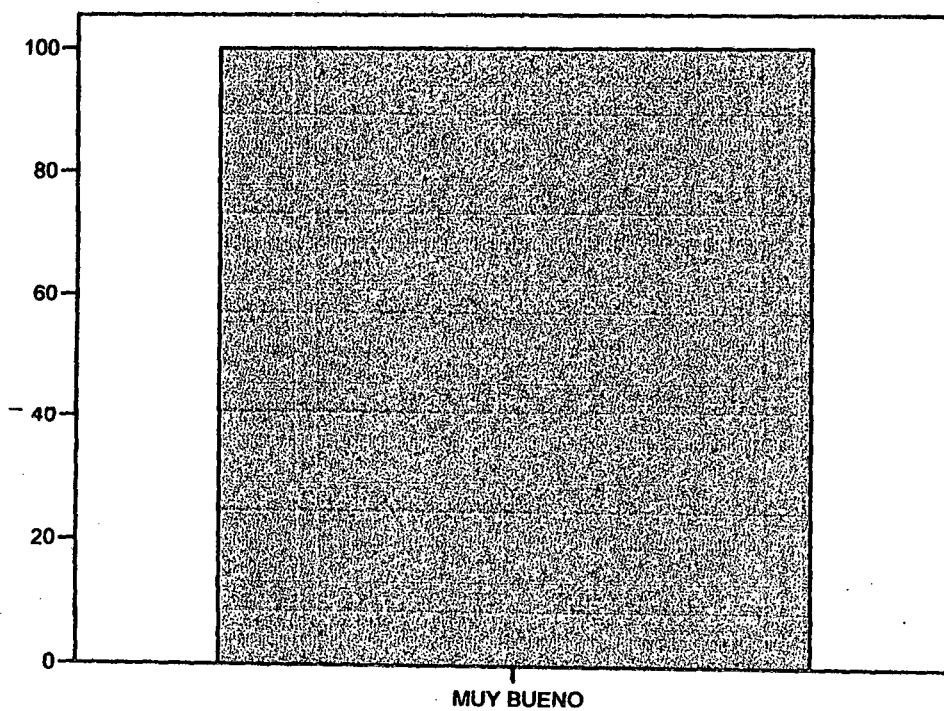
- **A la pregunta:** La reserva de material bibliográfico en la Biblioteca Central de Ucayali con el sistema le parece:

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	100	19,9	19,9	19,9
	BUENO	352	70,1	70,1	90,0
	REGULAR	50	10,0	10,0	100,0
	Total	502	100,0	100,0	



- **A la pregunta:** Planeación basada con información en tiempo real, integra y altamente confiable.

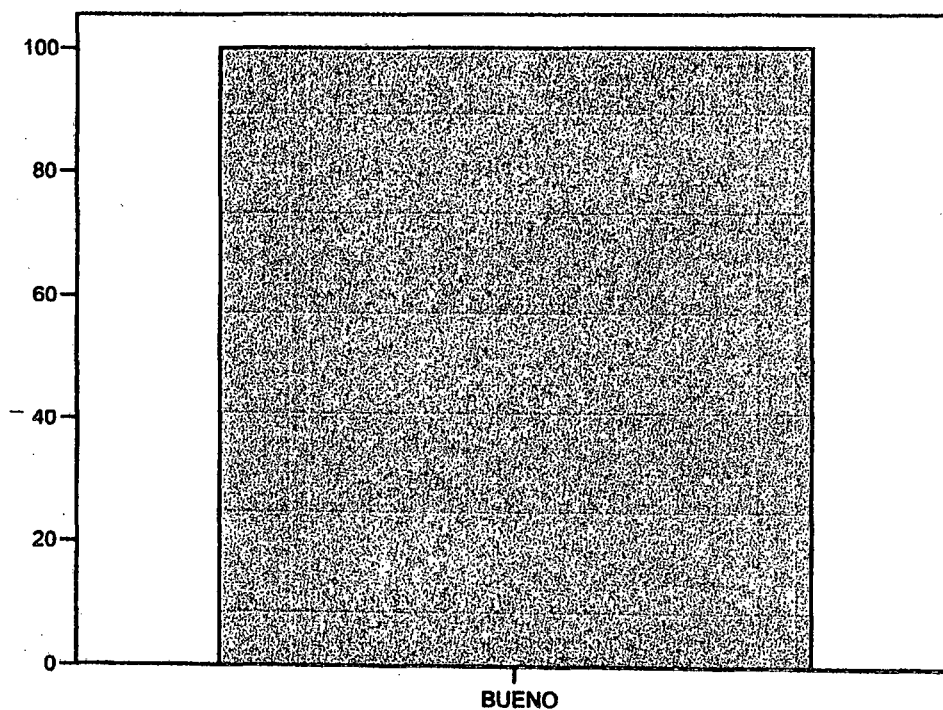
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	1	100,0	100,0	100,0



Planeación basada con información en tiempo real, integra y altamente confiable

- **A la pregunta:** Se establecen y definen niveles de autoridad y responsabilidad dentro de la organización.

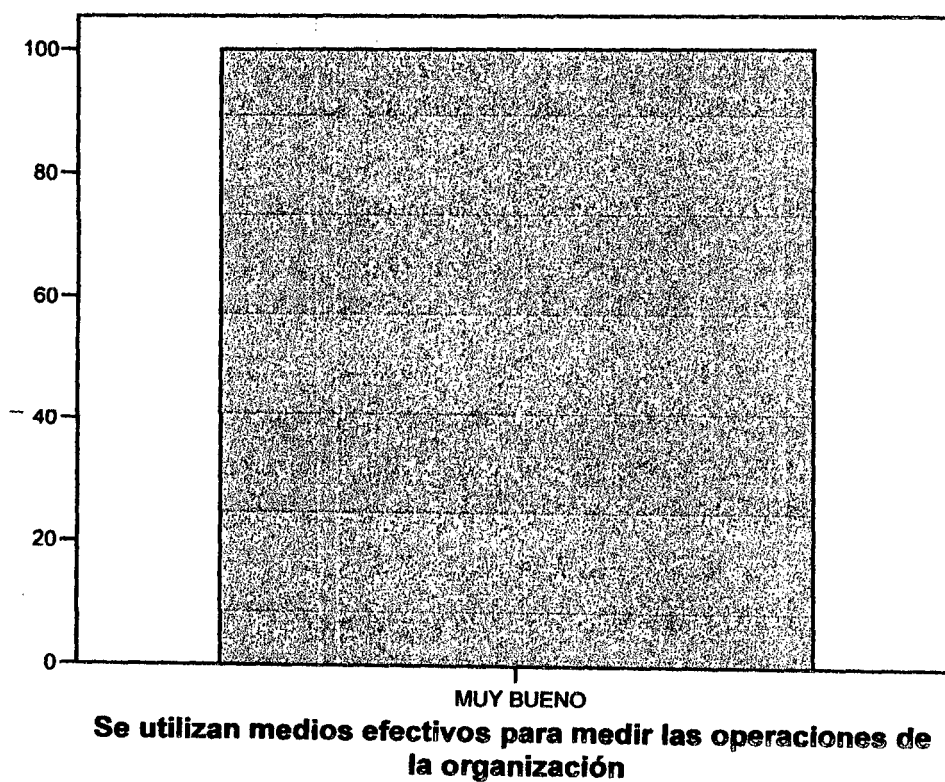
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	BUENO	1	100,0	100,0	100,0



Se establecen y definen niveles de autoridad y responsabilidad dentro de la organización

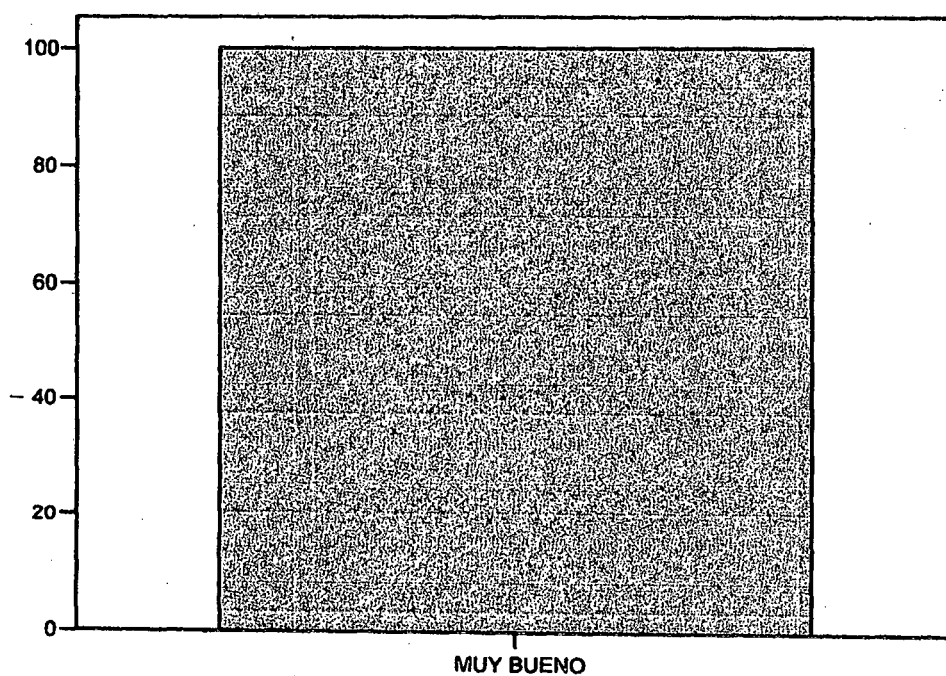
- **A la pregunta:** Se utilizan medios efectivos para medir las operaciones de la organización.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	1	100,0	100,0	100,0

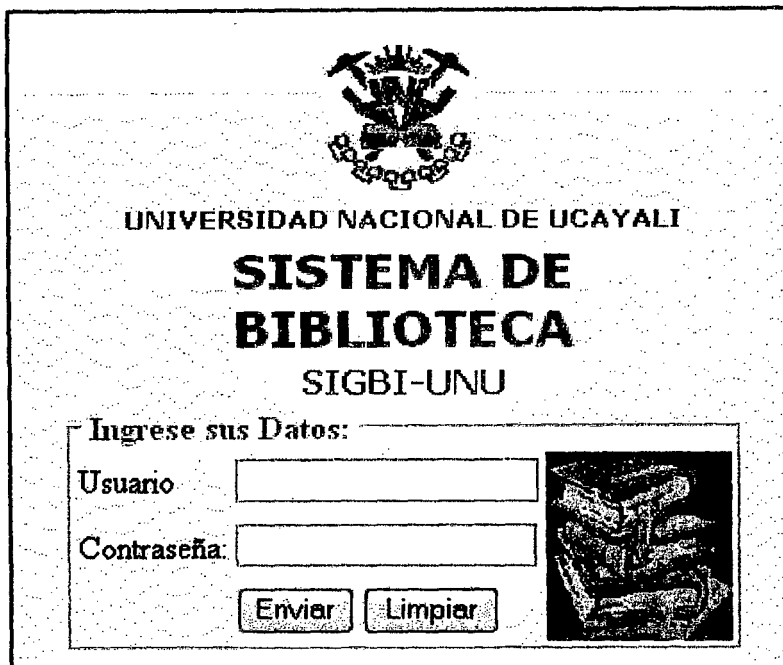


- **A la pregunta:** En base a los resultados del control se toman ajustes o medidas necesarias para mejorar los procesos de la organización.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	MUY BUENO	1	100,0	100,0	100,0



En base a los resultados del control se toman ajustes o medidas necesarias para mejorar los procesos de la organización

ANEXO 15: MANUAL DEL SISTEMA**1. Página de inicio (Login)****1.1. Descripción.**

Esta ventana sirve para ingresar al sistema de biblioteca a la vez registrarse. Se deberá solicitar el nombre del usuario y su respectiva clave de acceso al administrador del sistema para poder manipulo.

1.2. Funciones.

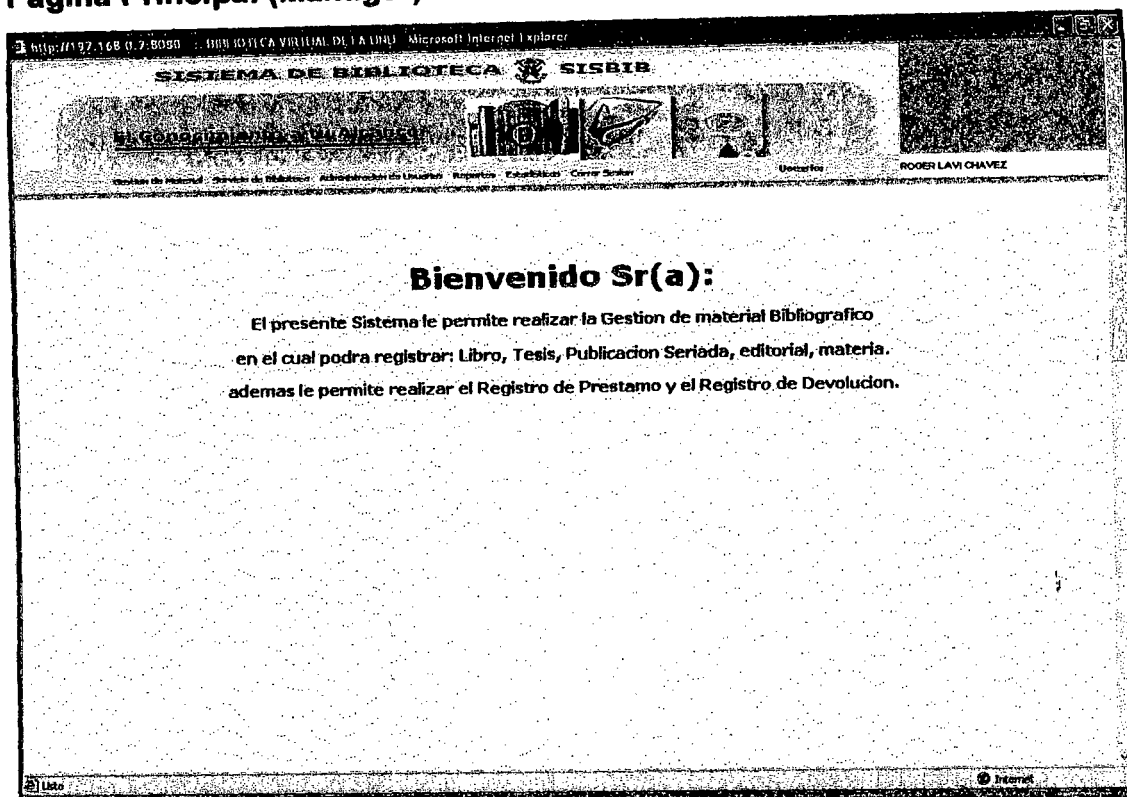
La ventana consta de botones y cajas de texto donde se introducirá la información necesaria para ingresar "Log in" al sistema, estas son:

- a) **Caja de texto "Usuario":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- b) **Caja de texto "Contraseña":** Se tendrá que redactarlo, ingresando la información a través del teclado. Los caracteres son ocultados mientras se redacta para protección.
- c) **Botón "Enviar":** Esta culminara la actividad de "Log in" acceso al sistema. Si la clave o usuario son correctas podrá ingresar al

sistema página principal "Manager", caso contrario tendrá que probar otra vez.

d) **Botón "Limpiar"**: Este le permitirá borrar todo rápidamente.

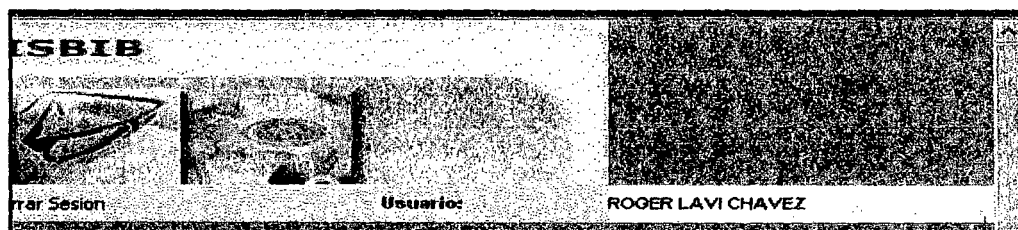
2. Página Principal (Manager)



2.1. Descripción.

Esta barra es constante no cambia es constante el proceso, además permite el acceso directo a todas las actividades del sistema.

a) En la parte superior Izquierda de la ventana se encuentra la el nombre del personal que esta utilizando el sistema en ese momento, se tubo que iniciar en la ventana de inicio "Log in". Como por ejemplo se muestra en al la imagen.



- b) En la parte superior derecha se encuentra los menús principales que el usuario principalmente va ha manipular.



- Gestión de Material.
- Servicio de Biblioteca.
- Administración de Usuarios.
- Reportes
- Estadísticas.
- Cerrar Sesión.

2.2. Barras de Menús.

2.2.1. Gestión de Material.

Este menú contienen los submenús de "Registrar" y "Mantenimiento" los cuales, cada uno, abren submenús. Estos activan nuevas ventanas. Para poder activarlas solo es necesario arrastrar el curso, sobre el menú y hacer clic derecho en el submenú.

2.2.1.1. Registrar.

En el este submenú permite el acceso a nuevas ventanas, cuya función exclusiva es la de ingresar información a la base de datos del sistema.

SISTEMA DE BIBLIOTECA  SISBIB	
El Compromiso a la Calidad	
Acciones de Mantenimiento	Servicio de Biblioteca Administración de Usuarios Reportes Estadísticas Cerrar Sesión
Mantenimiento	Libro Publicación Seriada Testis Autor Editorial Materia

A) Libro.

Este submenú permite el acceso a la ventana “Registrar libro” en la cual aparecerá instantes después de seleccionar esta barra de menú con el cursor.

Ventana "Registro de Libro".

REGISTRAR LIBRO			
Autogenerado	VIEBRE LEON SORIA		Nuevo
	EL SER HUMANO Y SUS PATOLOGIAS		
	PATOLOGIA HUMANO		
	PATERNSALIONS		
1	ARGENTINA		
AOAPEA			Nuevo
1998	200	SI	
SI	NO		
21	25	5	
			Nuevo

➤ **Descripción.**

La barra de principal se mantiene y mantiene las mismas características. Mas abajo en la parte inferior central se vera un formulario vacío que se llenará para registrar "crear" un libro.

➤ **Función.**

El formulario consta de botones y cajas de texto donde se introducirá la información necesaria para poder registrar "crear" un libro estas son:

- a) **Caja de texto "Código":** Este es automático no es necesario tratar de ingresar los datos.
- b) **Caja de texto "Autor":** Esta tiene una aplicación que le ayudara a encontrar la información deseada rápidamente. Si el autor ingresado se encuentra registrado aparecerá una lista donde se podrá seccionar el autor sin necesidad de ingresar el nombre por completo.

REGISTRAR LIBRO	
Código	Autogenerado
Autor	HEBREO Nuevo
Título	HEBREO LEON SORIA
Subtítulo	PATOLOGIA HUMANO
Editor	PATERNSLIONS
ISBN	
Fecha	

En caso que no este registrado se tiene un botón "Nuevo" a lado donde tan solo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Autor.

- c) **Caja de texto "Título":** Se tendrá que redactarlo, ingresando la información a través del teclado.

- d) **Caja de texto "Subtítulo"**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- e) **Caja de texto "Otros Autores"**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- f) **Caja de texto "Numero de edición"**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- g) **Caja de texto "Lugar de edición"**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- h) **Caja de texto "Editorial"**: Esta tiene una aplicación que le ayudara a encontrar la información deseada rápidamente. Si la editorial ingresada se encuentra registrado aparecerá una lista donde se podrá seccionar el autor sin necesidad de ingresar el nombre por completo.

The image shows a screenshot of a web form titled "REGISTRAR LIBRO". The form has several input fields and a "Nuevo" button. The fields are labeled on the left side of the form:

- Autor:** HEBREO
- Título:** EL SER HUMANO Y SUS PATOLOGIAS
- Subtítulo:** PATOLOGIA HUMANO
- Otros Autores:** PATERNSILIONS
- Edición:** 1
- Lugar de Edición:** ARGENTINA
- Editorial:** MEGABYTE
- Editorial:** AOAPEA
- Editorial:** PONTIFICIA UNIVERSIDAD CATOLICA
- Editorial:** NAVARRETE
- Edición:** 21
- Edición:** 25
- Edición:** 5

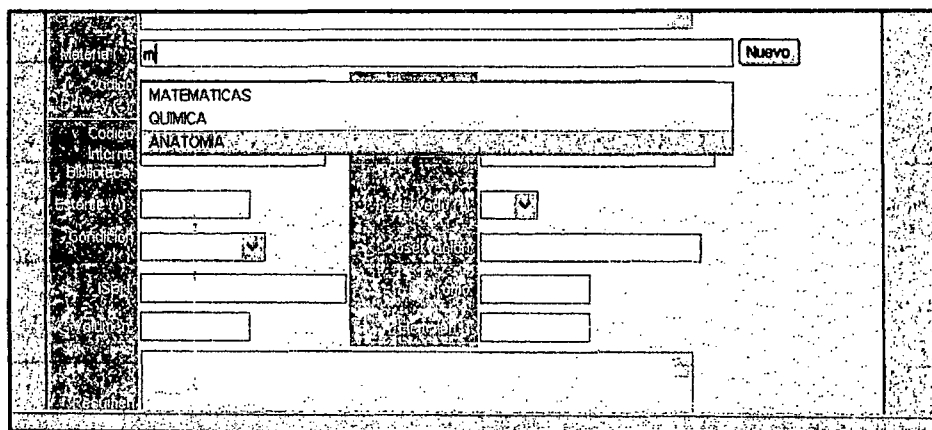
The "Nuevo" button is located next to the "Autor" field.

En caso que no este registrado se tiene un botón a lado donde tan solo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Editorial.

- i) **Caja de texto "Año de edición"**: Se tendrá que redactarlo, ingresando la información a través del teclado.

- j) **Caja de texto "Nº de página":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- k) **Caja de texto "Tabla":** Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las respuestas SI, NO las cuales se seleccionara con el cursor para indicar si el material contiene tablas, cuadros, gráficos estadísticos, etc. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- l) **Caja de texto "Ilustraciones":** Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las respuestas SI, NO las cuales se seleccionara con el cursor para indicar si este material contiene imágenes, fotos, dibujos, etc. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- m) **Caja de texto "CD":** Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las respuestas SI, NO las cuales se seleccionara con el cursor para indicar si este material vino con CD. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- n) **Caja de texto "Largo":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- o) **Caja de texto "Ancho":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- p) **Caja de texto "Espesor":** Se tendrá que redactarlo, ingresando la información a través del teclado.

- q) **Caja de texto “Descriptores”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- r) **Caja de texto “Materia”:** Esta tiene una aplicación que le ayudara a encontrar la información rápidamente. Si la materia ingresada se encuentra registrada aparecerá una lista donde se podrá seccionar la materia sin necesidad de ingresar el nombre por completo.



En caso que no este registrado se tiene un botón a lado donde tan solo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Materia.

- s) **Caja de texto “Código Dewey”:** Tendrá que redactarlo, ingresando la información a través del teclado.
- t) **Caja de texto “Código Málaga”:** Tendrá que ser redactado, ingresando la información a través del teclado.
- u) **Caja de texto “Código interno de Biblioteca”:** Tendrá que ser redactado, ingresando la información a través del teclado.
- v) **Caja de texto “Código de Almacén”:** Tendrá que ser redactado, ingresando la información a través del teclado.

- w) **Caja de texto "Estante"**: Tendrá que ser redactado, ingresando la información a través del teclado.
- x) **Caja de texto "Reservado"**: Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las respuestas SI, NO las cuales se seleccionara con el cursor el cual indicara si este es un libro único o sea que no se puede prestar a domicilio. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- y) **Caja de texto "Condición"**: Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las condiciones del libro Donado, Comprado, transferido las cuales se seleccionara con el cursor. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- z) **Caja de texto "Observación"**: Tendrá que ser redactado, ingresando la información a través del teclado.
- aa) **Caja de texto "Tomo"**: Tendrá que redactarlo, ingresando la información a través del teclado.
- bb) **Caja de texto "Volumen"**: Tendrá que redactarlo, ingresando la información a través del teclado.
- cc) **Caja de texto "Código Internacional ISBN"**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- dd) **Caja de texto "Ejemplar"**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- ee) **Caja de texto "Resumen"**: Se tendrá que redactarlo, ingresando la información a través del teclado.

- ff) Botón “Registrar”: Una vez completado el formulario se arrastrara el cursor y se hará clic derecho sobre el botón “Registrar” para ingresar los datos al sistema. Instantes después el sistema le dará una respuesta; aparecerá una cuadro dialogo que le recordará los datos que no ingresó y luego retornara de nuevo a la ventana del formulario, caso contrario le dará una respuesta de “Éxito en la operación”.

NOTA:

- En el formulario existe botones con el nombre “Nuevo” estos se utilizaran para el caso de no existir Autor, Editorial o Materia, entonces al hacer clic aparecerá una ventana para registrar el nuevo autor o nueva editorial o nueva materia, es por eso que es opcional.

- En el formulario al costo de los nombre de los caja de texto a llenar aparece en algunos entre paréntesis un asteriscos (*) esto significa que esos campos a llenar son necesariamente

obligatorio, si el caso no se llenara según los datos entonces se ingresará con un punto (.).

b) Publicación Seria da.

Este submenú permite el acceso a la ventana "Registrar publicación seria da" en la cual aparecerá instantes después de seleccionar esta barra de menú.

The screenshot shows a web interface for a library system. At the top, there is a header with the text "El Conocimiento a tu Alcance" and a logo. Below the header, there is a navigation bar with links: "Gestión de Materia", "Servicio de Biblioteca", "Reportes", and "Cerrar Sesión". On the right side of the navigation bar, there is a "Usuario:" label. In the center, there is a large button labeled "Publicación Seria da". To the left of this button, there is a sidebar menu with options: "Libro", "Tests", "Autor", "Editorial", and "Materia". The main content area displays the text "Bienvenido Sr(a):".

Ventana "Registrar publicación Seria da".

The screenshot shows a window titled "REGISTRAR PUBLICACION SERIADA". Inside the window, there is a form with several input fields and buttons. The form is organized into sections. The first section has a label "Autogenerado" and a "Nuevo" button. The second section has a "Nuevo" button. The third section has a "Nuevo" button. The fourth section has a "Nuevo" button. The form includes various input fields for text, numbers, and dates, as well as checkboxes and radio buttons. The background of the form has a wavy pattern.

➤ Descripción.

La barra de principal se mantiene y mantiene las mismas características. Mas abajo en la parte inferior central se vera un formulario vacío que se llenará para registrar "crear" una publicación seriada.

➤ **Funciones.**

El formulario consta de botones y cajas de texto donde se introducirá la información necesaria para poder registrar "crear" un libro estas son:

- a) **Caja de texto "Código":** Este es automático no es necesario tratar de ingresar los datos.
- b) **Caja de texto "Autor":** Esta tiene una aplicación que le ayudara a encontrar la información deseada rápidamente. Si el autor ingresado se encuentra registrado aparecerá una lista donde se podrá seccionar el autor sin necesidad de ingresar el nombre por completo.

En caso que no este registrado se tiene un botón "Nuevo" a lado donde tan sólo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Autor.

- c) **Caja de texto "Titulo":** Tendrá que redactarlo, ingresando la información a través del teclado.

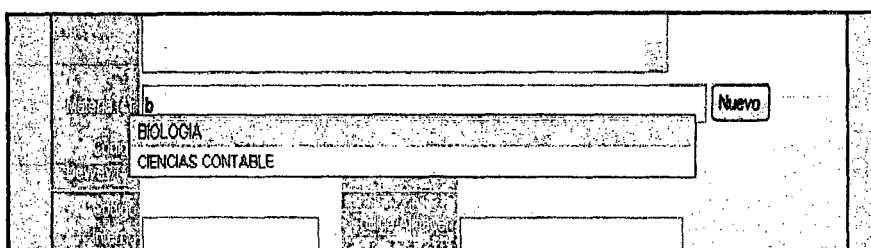
- d) **Caja de texto “Subtitulo”:** Tendrá que redactarlo, ingresando la información a través del teclado.
- e) **Caja de texto “Otros Autores”:** Tendrá que redactarlo, ingresando la información a través del teclado.
- f) **Caja de texto “Numero de edición”:** Tendrá que redactarlo, ingresando la información a través del teclado.
- g) **Caja de texto “Lugar de edición”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- h) **Caja de texto “Editorial”:** Esta tiene una aplicación que le ayudara a encontrar la información deseada rápidamente. Si la editorial ingresada se encuentra registrado aparecerá una lista donde se podrá seccionar el autor sin necesidad de ingresar el nombre por completo.

The screenshot shows a software interface with a dark sidebar on the left containing labels: "Otros Autores", "Edición No.", "Editorial (*)", "Año de Edición", and "Largo con". The main area contains several input fields and a dropdown menu. The "Edición No." field contains the number "2". The "Editorial (*)" field contains the text "UNIVERSIDAD" and has a "Nuevo" button to its right. Below it, a dropdown menu is open, showing a list of universities: "PONTIFICIA UNIVERSIDAD CATOLICA" and "UNIVERSIDAD NACIONAL DE UCAYALI". The "Largo con" field is partially visible at the bottom.

En caso que no este registrado se tiene un botón a lado donde tan solo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Editorial.

- i) **Caja de texto “Año de edición”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- j) **Caja de texto “Nº de página”:** Se tendrá que redactarlo, ingresando la información a través del teclado.

- k) **Caja de texto “Largo”**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- l) **Caja de texto “Ancho”**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- m) **Caja de texto “Espesor”**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- n) **Caja de texto “Descriptores”**: Se tendrá que redactarlo, ingresando la información a través del teclado.
- o) **Caja de texto “Materia”**: Esta tiene una aplicación que le ayudara a encontrar la información rápidamente. Si la materia ingresada se encuentra registrada aparecerá una lista donde se podrá seccionar la materia sin necesidad de ingresar el nombre por completo.



En caso que no este registrado se tiene un botón a lado donde tan solo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Materia.

- p) **Caja de texto “Código Dewey”**: Tendrá que redactarlo, ingresando la información a través del teclado.
- q) **Caja de texto “Código Málaga”**: Tendrá que ser redactado, ingresando la información a través del teclado.

- r) **Caja de texto "Código interno de Biblioteca":** Tendrá que ser redactado, ingresando la información a través del teclado.
- s) **Caja de texto "Código de Almacén":** Tendrá que ser redactado, ingresando la información a través del teclado.
- t) **Caja de texto "Estante":** Tendrá que ser redactado, ingresando la información a través del teclado.
- u) **Caja de texto "Reservado":** Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las respuestas SI, NO las cuales se seleccionara con el cursor el cual indicara si este es un libro único o sea que no se puede prestar a domicilio. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- v) **Caja de texto "Condición":** Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las condiciones del libro Donado, Comprado, transferido las cuales se seleccionara con el cursor. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- w) **Caja de texto "Observación":** Tendrá que ser redactado, ingresando la información a través del teclado.
- x) **Caja de texto "Año de circulación":** Tendrá que redactarlo, ingresando la información a través del teclado.
- y) **Caja de texto "Número":** Tendrá que redactarlo, ingresando la información a través del teclado.

- z) Caja de texto de Selección "Periodicidad":** Tendrá que seleccionar si es Diaria, Semanal, Mensual, Bimestral, Trimestral, Semestral o Anual.
- aa)Caja de texto "Tomo":** Tendrá que redactarlo, ingresando la información a través del teclado.
- bb) Caja de texto "Volumen":** Tendrá que redactarlo, ingresando la información a través del teclado.
- cc)Caja de texto "Código Internacional ISSN":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- dd) Caja de texto "Ejemplar":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- ee)Caja de texto "Resumen":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- ff) Botón "Registrar":** Una vez completado el formulario se arrastrara el cursor y se hará clic derecho sobre el botón "Registrar" para ingresar los datos al sistema. Instantes después el sistema le dará una respuesta; aparecerá una cuadro dialogo que le recordará los datos que no ingresó y luego retornara de nuevo a la ventana del formulario, caso contrario le dará una respuesta un formulario con la figura del código de barras y el icono imprimir el cual permite la impresión del código de barras como se muestra en la siguiente figura.

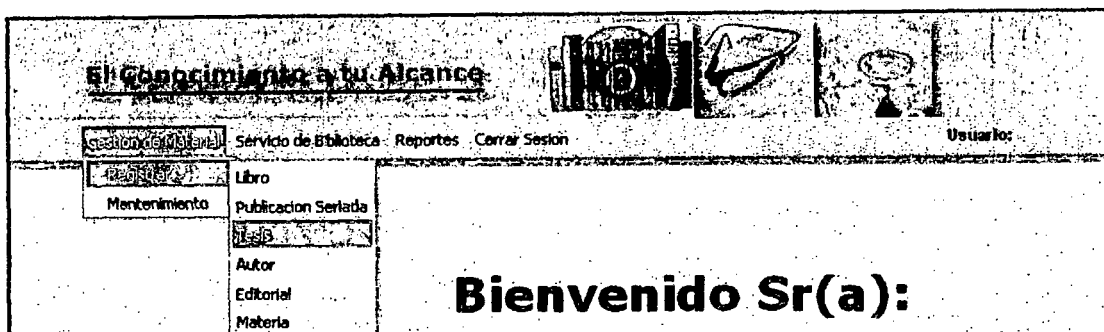
NOTA:

- En el formulario existe botones con el nombre "Nuevo" estos se utilizaran para el caso de no existir Autor, Editorial o Materia, entonces al hacer clic aparecerá una ventana para registrar el nuevo autor o nueva editorial o nueva materia, es por eso que es opcional.

- En el formulario al costo de los nombre de los caja de texto a llenar aparece en algunos entre paréntesis un asteriscos (*) esto significa que esos campos a llenar son necesariamente obligatorio, si el caso no se llenara según los datos entonces se ingresará con un punto (.).

c) Tesis.

Este submenú permite el acceso a la ventana "Registrar Tesis" en la cual aparecerá instantes después de seleccionar esta barra de menú con el cursor.



Ventana "Registrar Tesis".

➤ Descripción.

La barra de principal se mantiene y mantiene las mismas características. Mas abajo en la parte inferior central se vera un formulario vacío que se llenará para registrar "crear" una tesis.

➤ Funciones.

El formulario consta de botones y cajas de texto donde se introducirá la información necesaria para poder registrar "crear" un libro estas son:

- a) **Caja de texto "Autor":** Esta tiene una aplicación que le ayudara a encontrar la información deseada rápidamente. Si el autor ingresado se encuentra registrado aparecerá una lista donde se podrá seccionar el autor sin necesidad de ingresar el nombre por completo.
- b) **Caja de texto "Código":** Este es automático no es necesario tratar de ingresar los datos.

The image shows a screenshot of a web form titled "REGISTRAR TESIS". Inside the form, there is a section for the "Autor" (Author) field. It features a dropdown menu with the text "Autogenerado" and a list of authors, including "JAVIER LUNA VELASQUEZ". To the right of the dropdown menu is a button labeled "Nuevo".

En caso que no este registrado se tiene un botón "Nuevo" a lado donde tan solo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Autor.

- c) **Caja de texto "Título":** Tendrá que redactarlo, ingresando la información a través del teclado.
- d) **Caja de texto "Subtítulo":** Tendrá que redactarlo, ingresando la información a través del teclado.
- e) **Caja de texto "Otros Autores":** Tendrá que redactarlo, ingresando la información a través del teclado.
- f) **Caja de texto "Numero de edición":** Tendrá que redactarlo, ingresando la información a través del teclado.

- g) **Caja de texto “Lugar de edición”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- h) **Caja de texto “Editorial”:** Esta tiene una aplicación que le ayudara a encontrar la información deseada rápidamente. Si la editorial ingresada se encuentra registrado aparecerá una lista donde se podrá seccionar el autor sin necesidad de ingresar el nombre por completo.

The screenshot shows a form with several input fields. The 'Editorial' field contains the text 'UNIVERSIDAD'. Below this field, a list of search results is displayed: 'PONTIFICIA UNIVERSIDAD CATOLICA' and 'UNIVERSIDAD NACIONAL DE LICA YALI'. To the right of the list, there is a button labeled 'Nuevo'. The interface also shows other fields like 'Lugar de edición' and 'Año de edición'.

En caso que no este registrado se tiene un botón a lado donde tan solo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Editorial.

- i) **Caja de texto “Año de edición”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- j) **Caja de texto “N° de página”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- k) **Caja de texto “Largo”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- l) **Caja de texto “Ancho”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- m) **Caja de texto “Espesor”:** Se tendrá que redactarlo, ingresando la información a través del teclado.

n) **Caja de texto "Descriptores":** Se tendrá que redactarlo, ingresando la información a través del teclado.

o) **Caja de texto "Materia":** Esta tiene una aplicación que le ayudara a encontrar la información rápidamente. Si la materia ingresada se encuentra registrada aparecerá una lista donde se podrá seccionar la materia sin necesidad de ingresar el nombre por completo.

En caso que no este registrado se tiene un botón a lado donde tan solo arrastrando el cursor y haciendo clic en el se podrá abrir la ventana de Registrar Materia.

p) **Caja de texto "Código Dewey":** Tendrá que redactarlo, ingresando la información a través del teclado.

q) **Caja de texto "Código Malaga":** Tendrá que ser redactado, ingresando la información a través del teclado.

r) **Caja de texto "Código interno de Biblioteca":** Tendrá que ser redactado, ingresando la información a través del teclado.

s) **Caja de texto "Código de Almacén":** Tendrá que ser redactado, ingresando la información a través del teclado.

t) **Caja de texto "Estante":** Tendrá que ser redactado, ingresando la información a través del teclado.

- u) **Caja de texto “Reservado”:** Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las respuestas SI, NO las cuales se seleccionara con el cursor el cual indicara si este es un libro único o sea que no se puede prestar a domicilio. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- v) **Caja de texto “Condición”:** Se observará que esta aplicación tiene una pestaña la cual abrir un cuadro que le mostrar las condiciones del libro Donado, Comprado, transferido las cuales se seleccionara con el cursor. Son las únicas opciones que se tiene para ingresar, no se podrá ni deberá ingresar información desde el teclado.
- w) **Caja de texto “Observación”:** Tendrá que ser redactado, ingresando la información a través del teclado.
- x) **Caja de texto “Facultad”:** Tendrá que redactarlo, ingresando la información a través del teclado.
- y) **Caja de texto “Grado”:** Tendrá que redactarlo, ingresando la información a través del teclado.
- z) **Botón “Registrar”:** Una vez completado el formulario se arrastrara el cursor y se hará clic derecho sobre el botón “Registrar” para ingresar los datos al sistema. Instantes después el sistema le dará una respuesta; aparecerá una cuadro dialogo que le recordará los datos que no ingresó y luego retornara de nuevo a la ventana del formulario, caso contrario le dará una respuesta un formulario con la figura del código de barras y el icono imprimir el cual permite la impresión del código de barras como se muestra en la siguiente figura.

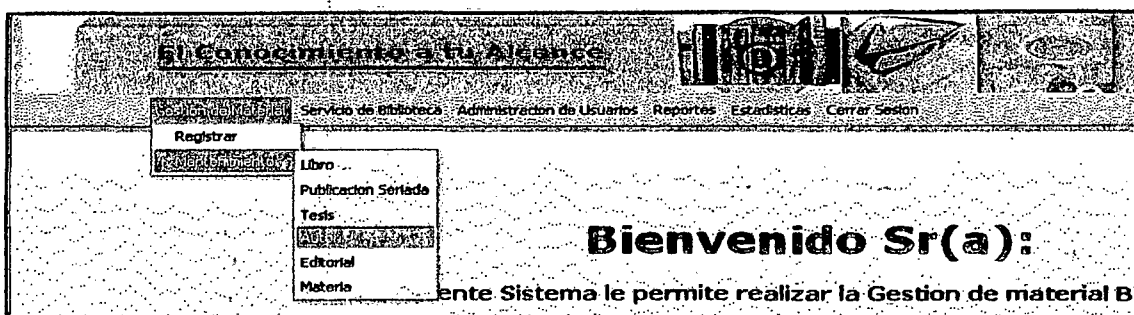
NOTA:

- En el formulario existe botones con el nombre "Nuevo" estos se utilizaran para el caso de no existir Autor, Editorial o Materia, entonces al hacer clic aparecerá una ventana para registrar el nuevo autor o nueva editorial o nueva materia, es por eso que es opcional.

- En el formulario al costo de los nombre de los caja de texto a llenar aparece en algunos entre paréntesis un asteriscos(*) esto significa que esos campos a llenar son necesariamente obligatorio, si el caso no se llenara según los datos entonces se ingresará con un punto(.).

d) Autor.

Este submenú permite el acceso a la ventana "Registrar Autor" en la cual, aparecerá instantes después de seleccionar esta barra de menú.



Ventana "Registrar Autor".

➤ Descripción.

La barra de principal se mantiene y mantiene las mismas características. Mas abajo en la parte inferior central se vera un formulario vacío que se llenará para registrar "crear" un autor.

➤ Funciones.

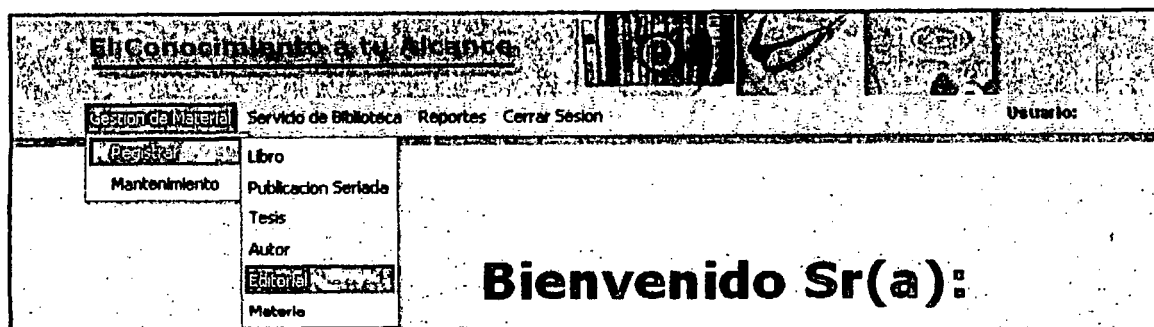
El formulario consta de botones y cajas de texto donde se introducirá la información necesaria para poder registrar "crear" un libro estas son:

- Caja de texto "Código":** Este es automático no es necesario tratar de ingresar los datos.
- Caja de texto "Nombres":** Se tendrá que redactarlo, ingresando la información a través del teclado.

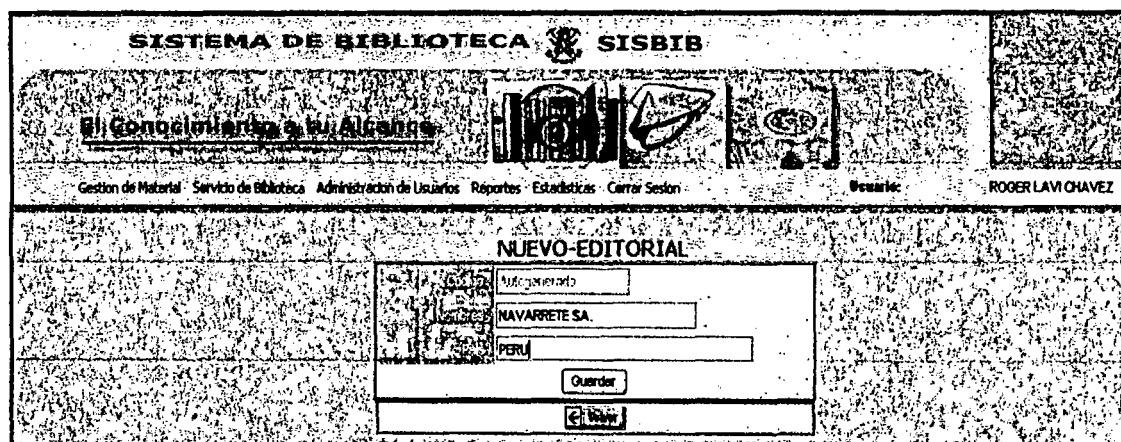
- c) **Caja de texto “Apellido Paterno”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- d) **Caja de texto “Apellido Materno”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- e) **Caja de texto “País”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- f) **Botón “Registrar”:** Una vez completado el formulario se arrastrara el cursor y se hará clic derecho sobre el botón “Registrar” para ingresar los datos al sistema. Instantes después el sistema le dará una respuesta; aparecerá una cuadro dialogo que le recordará los datos que no ingresó, luego retornara de nuevo a la ventana del formulario, caso contrario le dará una respuesta de “éxito en la operación”.
- g) **Botón “Volver”:** Este botón es opcional, al hacer click, aparecerá la lista de todos los autores registrados en el sistema.

e) Editorial

Este submenú permite el acceso a la ventana “Registrar Editorial” en la cual aparecerá instantes después de seleccionar esta barra de menú.



Venta “Registrar Editorial”



➤ Descripción.

La barra de principal se mantiene y mantiene las mismas características. Mas abajo en la parte inferior central se vera un formulario vacío que se llenará para registrar “crear” una editorial.

➤ Funciones.

El formulario consta de botones y cajas de texto donde se introducirá la información necesaria para poder registrar “crear” un libro estas son:

- a) **Caja de texto "Código":** Este es automático no es necesario tratar de ingresar los datos.
- b) **Caja de texto "Nombres":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- c) **Caja de texto "País":** Se tendrá que redactarlo, ingresando la información a través del teclado.
- d) **Botón "Registrar":** Una vez completado el formulario se arrastrara el cursor y se hará clic derecho sobre el botón "Registrar" para ingresar los datos al sistema. Instantes después el sistema le dará una respuesta; aparecerá una cuadro dialogo que le recordará los datos que no ingresó, luego retornara de nuevo a la ventana del formulario, caso contrario le dará una respuesta de "éxito en la operación".
- e) **Botón "Volver":** Este botón es opcional, al hacer click, aparecerá la lista de todas las editoriales registradas en el sistema.

f) Materia.

Este submenú permite el acceso a la ventana "Registrar Materia" en la cual aparecerá instantes después de seleccionar esta barra de menú.



Ventana "Registrar Materia".



➤ Descripción.

La barra de inicio se mantiene y mantiene las mismas características. Se vera un formulario vacío que se llenará para registrar "crear" una materia.

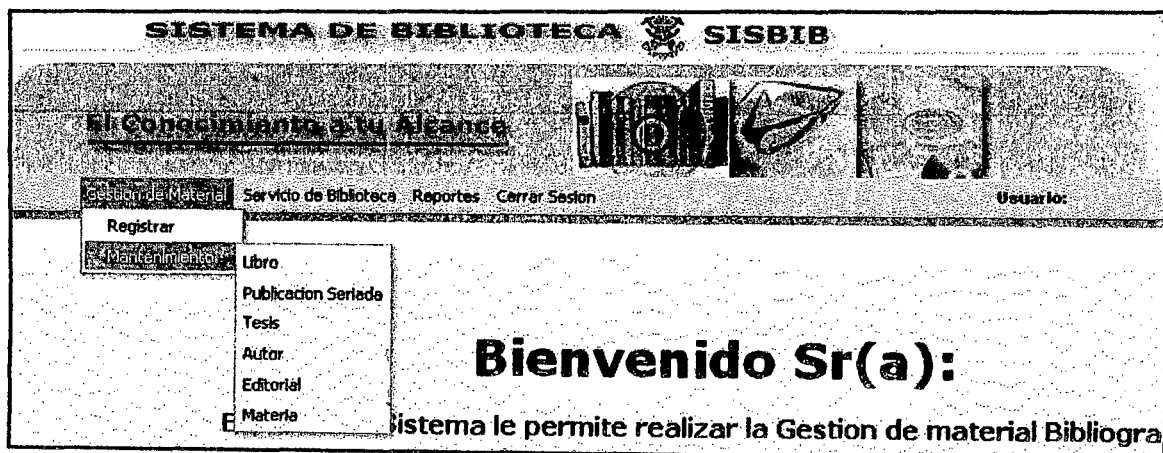
➤ Funciones.

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para poder registrar "crear" un libro, estas son:

- a) **Caja de texto "Código":** Este es automático no es necesario tratar de ingresar los datos.
- b) **Caja de texto "Descripción":** Se tendrá que redactar el nombre de la materia.
- c) **Botón "Registrar":** Una vez completado el formulario se arrastrara el cursor y se hará clic derecho sobre el botón "Registrar" para ingresar los datos al sistema. Instantes después el sistema le dará una respuesta; aparecerá una cuadro dialogo que le recordará los datos que no ingresó, luego retornara de nuevo a la ventana del formulario, caso contrario le dará una respuesta de "éxito en la operación".
- d) **Botón "Volver":** Este botón es opcional, al hacer click, aparecerá la lista de todas las materias registrados en el sistema.

2.2.1.2. Mantenimiento.

En el este submenú permite el acceso a nuevas ventanas, cuya función exclusiva es la modificar, corregir o eliminar la información de la base de datos del sistema.



a) Consulta de Material.

Ventana "Consulta de Material".

The screenshot shows a window titled "CONSULTAR MATERIAL". Inside, there is a search bar with two buttons: "BUSCAR" and "BUSCAR POR". Both buttons have a small downward arrow next to them, indicating they are dropdown menus.

➤ Descripción.

Antes de hacer el mantenimiento del material buscamos el material el tipo de material, si es Libro, Tesis o Publicación seriada; una vez encontrado se pasa a la modificación de los datos.

➤ Función.

El formulario consta de cajas de texto de selección el nos permite realizar la búsqueda si es libro, tesis o publicación seriada, además la búsqueda es por autor o por código Dewey o Málaga.

a) Caja de texto de Selección "Buscar": Escogemos si es Libro, Tesis o Publicación Seriada.

This screenshot shows the "CONSULTAR MATERIAL" window with the "Buscar" dropdown menu set to "Libro". The "Buscar por" dropdown menu is set to "Autor", and the text "HEBREO LEON SORIA" is entered in the search field. The "Buscar" button is visible on the right.

b) Caja de texto de Selección "Buscar por": Escogemos para que la búsqueda sea por Autor o por Código Dewey-Málaga.

This screenshot shows the "CONSULTAR MATERIAL" window with the "Buscar por" dropdown menu set to "Dewey-Málaga". The "Buscar" dropdown menu is set to "Libro". The search field is empty, and the "Buscar" button is visible on the right.

c) Botón "Buscar": Ejecuta la búsqueda según los parámetros que ingresamos anteriormente. Mostrándonos una lista de registro si es que existe datos con esos parámetros.

CONSULTAR MATERIAL

Código	CLASIFICADOR	TÍTULO	AUTOR	ESTADO	OPCIONES
000000001L	qwe qwe2S	SDFS	LEON SORIA HEBREO	Disponible	

SUBTÍTULO:	LO QUE UN DIA FUE	
EDITORIAL:	MEGABYTE	
EDICIÓN:	ASD 2323	
AUTORES:	edasd	
RESUMEN:	AEDA	
Nº PAGINAS:	123	
libro de Ejemplar: 1		

000000004L	sdd1 fde34	HOLH	LEON SORIA HEBREO	Disponible	
000000005L	44646 es34	EL SER HUMANO Y SUS PATOLOGIAS	LEON SORIA HEBREO	Disponible	
000000006L	44646 es34	EL SER HUMANO Y SUS PATOLOGIAS	LEON SORIA HEBREO	Reservado	
000000007L	44646 es34	EL SER HUMANO Y SUS PATOLOGIAS	LEON SORIA HEBREO	Prestado	

d) Botón “Ver Detalle”: Muestra los detalles del registro.

e) Botón “Lápiz”: Llama el formulario para modificar los datos del material ya sea Libro, Tesis o Publicación seriada.

MANTENER LIBRO

000000001L
Nuevo

HEBREO LEON SORIA

SDFS

LO QUE UN DIA FUE

edasd

2

ASD

MEGABYTE

2323

123

SI

SI

1

2

3

ASD

CIENCIAS SOCIALES

qwe

qwe2S

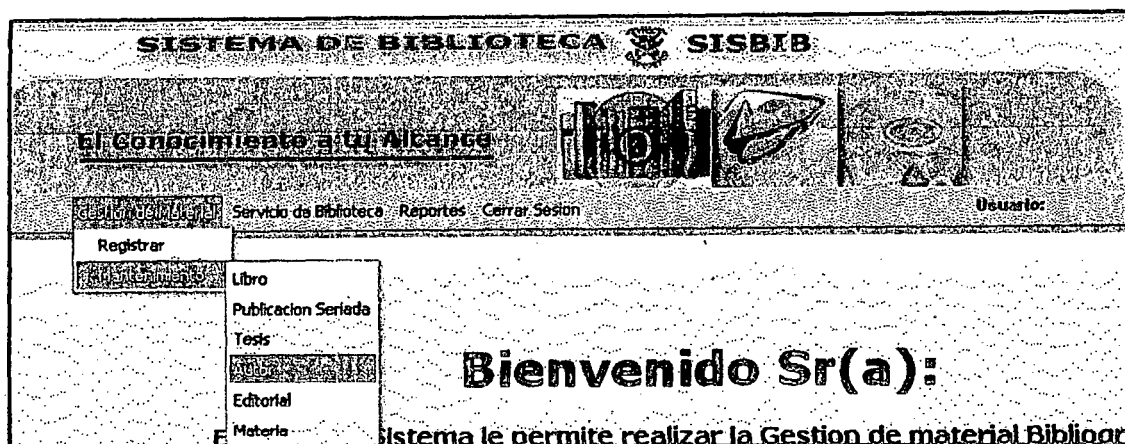
265

etmd

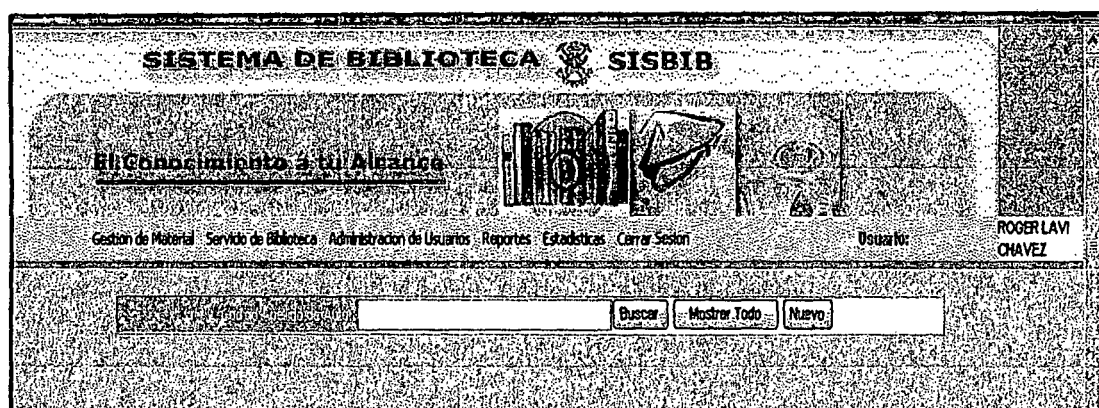
Nuevo
Nuevo

b) Autor.

Este submenú permite el acceso a la ventana “Mantener Autor” en la cual aparecerá instantes después de seleccionar esta barra de menú.



Ventana "Manteniendo Autor".



➤ Descripción.

La barra principal del menú se mantiene y tiene las mismas características. Mas abajo en la parte inferior central se vera una caja de texto con tres botones que posteriormente indicaremos sus funciones.

➤ Funciones.

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para poder mantener "modificar o eliminar", básicamente el proceso consiste en buscar al autor y luego realizar la modificación u eliminación según las necesidades del usuario:

- a) **Caja de texto "Ingrese Apellido de autor":** Esta aparecerá por defecto al inicio, tiene una aplicación que le ayudara a simplificar. Puede ingresar por ejemplo la letra "a" el sistema buscará todo los autores cuyo nombre o apellidos contenga la letra "a" la búsqueda se hará mas exacta cuando digitamos con mayor exactitud el nombre u apellidos que se desea buscar; esto funcionará al hacer click en el botón "Buscar", desplegándose una lista de todos los autores que contengan la palabra que se digito en la caja de texto.

- b) **Botón Buscar:** Una vez ingresado el nombre u apellido del autor a modificar o eliminar presionamos el botón "buscar" para que el sistema realice la búsqueda según la palabra ingresada en la caja de texto mostrándonos una lista de los autores coincidente con la palabra, si existe mas de quince registros coincidente con la búsqueda el sistema muestra agrupando los registros de quince en quince como se muestra en la figura siguiente:

Código Autor	Nombre	Apellido	País	Modificar
00000001	ARNOLDO	MONDADORI	ESPAÑA	X
00000002	MARIA TERESA	ANGUERA	ESPAÑA	X
00000004	JALIME	ARAHAI	ESPAÑA	X
00000006	RAUL	DORRA	RUSIA	X
00000007	ENRIQUE ED	LA MOZ BRITO	PERU	X
00000009	VICTOR	VASQUEZ VILLANUEVA	PERU	X
00000010	LUIS EDUARDO	ACOSTA HOYOS	ARGENTINA	X
00000011	PERNADO	ARIAS GALICIA	MEXICO	X
00000012	ROBERTO B	AVILA ACOSTA	PERU	X
00000014	WAYNE C	BOOTH	ESPAÑA	X
00000020	LAURA	CAZARES HERNANDEZ	MEXICO	X
00000021	PIERGORGIO	CORBETTA	ESPAÑA	X
00000022	JOSE MARIA	DESANTES QUANTER	ESPAÑA	X
00000024	RAUL	FERNANDEZ GARCIA	EE UU	X
00000025	YSABEL	GRACIDA JUAREZ	MEXICO	X

- c) **Botón “Mostrar Todo”:** Al hacer clic en este botón nos muestra todos los autores registrados en el sistema ordenados alfabéticamente ascendente según el apellido paterno del autor.

Código Autor	Nombre	Apellidos	País	Mantener
00001904	NUFFIELD		ESPAÑA	X
00000535	ADELA	ABAD DE SERVIN	MEXICO	X
00001358	JÓRGE	ABADIE LINARES	PERU	X
00000062	LISA	ABBOTT	USA	X
00000869	ROBERTO	ABUSADA	PERU	X
00000559	JOSE	ACOSTA EYZAGUIRRE	PERU	X
00000010	LUIS EDUARDO	ACOSTA HOYOS	ARGENTINA	X
00001290	BEATRIZ	ACTIZ	ARGENTINA	X
00002066	ARTHUR	ADAMSÓN	ESPAÑA	X
00000232	VICTOR	AFANASIEV	ARGENTINA	X
00000245	V. G.	AFANASSIEV	RUSIA	X
00000609	V.	AFANAVIEV	RUSIA	X
00000426	ALFREDO	AGÜERO OSCAR	PERU	X
00000261	GUIDO	AGUILAR	MEXICO	X
00001905	J	AGUILAR	ESPAÑA	X

- d) **Botones de desplazamiento de registros:** Esto sirve para mostrar la siguiente lista de autores para el caso de que exista mas de quince registro según el criterio de búsqueda que le hemos dado; estos botones son:

 **Botón “Adelante”:** Permite avanzar los siguientes quince registros de la búsqueda.

 **Botón “Últimos Registros”:** Permite avanzar hasta los últimos registros de la búsqueda.

 **Botón “Atrás”:** Permite desplazarnos hacia atrás mostrando los quince registros anteriores.

 **Botón “Primeros Registros”:** Permite desplazarnos hacia los quince primeros registros.

- e) **Botón “Modificar”**: Este se encuentra diferenciado por un lápiz que se muestra en la lista de nuestra búsqueda; al hacer clic sobre el lápiz nos muestra el formulario para poder modificar al autor que hemos seleccionado.

00001823	TERRENCE J	AKAI	MEXICO		X
00000470	TIMIAS	AKUST NUGKAI	PERU		X
00000780	GERMAN	ALARCO T	PERU		X

MODIFICAR-AUTOR

Código	Autogenerado
Nombre	TIMIAS
Apellido Paterno	AKUST
Apellido Materno	NUGKAI
País	PERU

- f) **El Botón “Volver”**: nos permite regresar a la pantalla anterior mostrándonos todos los registros de autores que existe en el sistema.

Una vez modificado los campos de nombre o apellido paterno o materno o país, para guardar los cambios hacemos click en el botón **“Guardar”**. Mostrándonos el siguiente mensaje de conformidad que los datos se ha guardado correctamente.



Al hacer clic en regresar volvemos a la pantalla inicial del mantenimiento del autor mostrándonos todos los autores que existe en el sistema.

- g) **Botón “Eliminar”:** Este se encuentra diferenciado por unos equis “X” que se muestra; al hacer clic sobre la equis nos muestra una ventana de alerta preguntándonos: “Si realmente deseamos eliminar el registro seleccionado?”.

ID del Autor	Nombre	Apellidos	País	Mantener
00001509	MARIA LUISA	AGUILAR HURTADO	PERU	X
00001582	PERNANDO	AGUIRRE PEREZ	PERU	X
00000999	PERNANDO	AGUIRRE PEREZ	PERU	X
00001823	TERRENCE J	AKAI	MEXICO	X
00000470	TIMIAS	AKUST NUKKAI	PERU	X
00000780	GERMAN	ALARCO T	PERU	X
00000881	RODOLFO	ALARCON	PERU	X
00001678	EMILIO		ESPAÑA	X
00001979	FERNANDO		USA	X
00000574	JULIO		PERU	X
00001511	ADRIAN		MEXICO	X
00001070	GIORGIO		PERU	X
00000864	JAVIER		PERU	X
00000202	JOSE ANTONIO	ALCANTARA	ESPAÑA	X
00001076	LORENA	ALCAZAR	PERU	X

Microsoft Windows Explorer

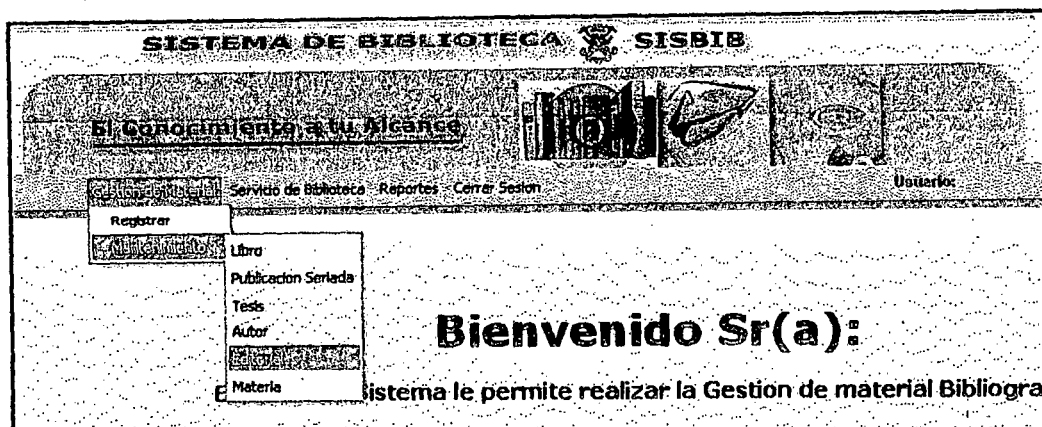
¿Realmente desea eliminar el registro?

Si hacemos clic en el botón “Aceptar” el sistema verificará si el autor se encuentra en un registro de un material, si el caso no se encuentra, se procede a eliminar mostrando el mensaje de que se eliminó con éxito, caso contrario el sistema mostrará un mensaje que el autor no se puede eliminar porque se encuentra registrado en un material.

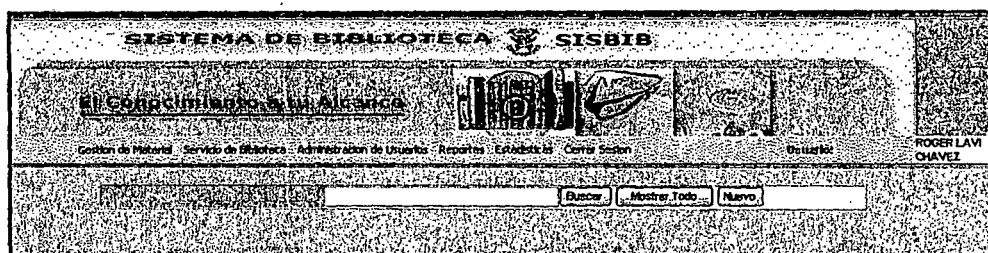
Si hacemos clic en el botón "Cancelar", no se procede a la eliminación del registro cerrándose la ventana de alerta.

c) Editorial.

Este submenú permite el acceso a la ventana "Mantener Editorial" en la cual aparecerá instantes después de seleccionar esta barra de menú con el cursor.



Venta "Mantenimiento Editorial".



➤ Descripción.

La barra principal del menú se mantiene y tiene las mismas características. Mas abajo en la parte inferior central se vera una caja de texto vacía con tres botones que posteriormente indicaremos sus funciones.

➤ Funciones.

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para poder mantener "modificar o eliminar", básicamente el proceso consiste en

buscar la editorial y luego realizar la modificación u eliminación según las necesidades del usuario:

- a) **Caja de texto “Ingrese Nombre de la editorial”:** Esta aparecerá por defecto al inicio, tiene una aplicación que le ayudara a simplificar. Puede ingresar por ejemplo la letra “a” el sistema buscará todas las editoriales cuyo nombre la letra “a” la búsqueda se hará mas exacta cuando digitamos con mayor exactitud el nombre de la editorial que se desea buscar; esto funcionará al hacer click en el botón “Buscar”, desplegándose una lista de todas las editoriales que contengan la palabra que se digito en la caja de texto.

The image shows a search interface with a text input field on the left, followed by three buttons: 'Buscar', 'Mostrar Todo', and 'Nuevo'.

- b) **Botón Buscar:** Una vez ingresado el nombre de la editorial a modificar o eliminar presionamos el botón “buscar” para que el sistema realice la búsqueda según la palabra ingresada en la caja de texto mostrándonos una lista de las editoriales coincidente con la palabra, si existe mas de quince registros coincidente con la búsqueda el sistema muestra agrupando los registros de quince en quince como se muestra en la figura siguiente:

The image shows a table of search results. The table has four columns: 'Código Editorial', 'Nombre', 'País', and 'Modificar'. The data is as follows:

Código Editorial	Nombre	País	Modificar
00001	CATEDRA S.A.	ESPAÑA	X
00002	ASESORANDINA	PERU	X
00003	AMAZONAS	PERU	X
00005	OMEGA S.A.	ESPAÑA	X
00006	IMPRENTA UNION	PERU	X
00010	ESTUDIOS Y EDICIONES R.A.	PERU	X
00012	GEDISA S.A.	ESPAÑA	X
00013	EDEGRAF	PERU	X
00015	COFAB S.A.	ESPAÑA	X
00016	SINTESIS S.A.	ESPAÑA	X
00017	INTERNACIONAL THOMSON EDITORES S.A.	MEXICO	X
00018	EDERE S.A.	MEXICO	X
00020	ARCO LIBROS S.L.	ESPAÑA	X
00021	MANTARO	PERU	X
00023	BIBLOTECA NUEVA	ESPAÑA	X

- c) **Botón “Mostrar Todo”:** Al hacer clic en este botón nos muestra todas las editoriales registradas en el sistema ordenados alfabéticamente ascendentes.

Codigo Editorial	Nombre	País	Modificar
00003	AMAZONAS	PERU	X
00020	ARCO LIBROS S.L.	ESPAÑA	X
00002	ASESORANDINA	PERU	X
00023	BIBLOTECA NUEVA	ESPAÑA	X
00027	CAAP	PERU	X
00001	CATEDRA S.A.	ESPAÑA	X
00015	COFAS S.A.	ESPAÑA	X
00029	CONCYTEC S.A.	PERU	X
00004	COSMOS	RUSIA	X
00019	ECOE	COLOMBIA	X
00013	EDEGRAF	PERU	X
00010	EDEFE S.A.	MEXICO	X
00010	ESTUDIOS Y EDICIONES S.A.	PERU	X
00024	FUNDACION CARBAJAL S.A.	COLOMBIA	X
00022	G. HERRERA	PERU	X

- d) **Botones de desplazamiento de registros:** Esto sirve para mostrar la siguiente lista de editoriales para el caso de que exista mas de quince registro según el criterio de búsqueda que le hemos dado; estos botones son:



Botón “Adelante”: Permite avanzar los siguientes quince registros de la búsqueda.



Botón “Últimos Registros”: Permite avanzar hasta los últimos registros de la búsqueda.



Botón “Atrás”: Permite desplazarnos hacia atrás mostrando los quince registros anteriores.



Botón “Primeros Registros”: Permite desplazarnos hacia los quince primeros registros.

- e) **Botón “Modificar”:** Este se encuentra diferenciado por un lápiz que se muestra en la lista de nuestra búsqueda; al

hacer clic sobre el lápiz nos muestra el formulario para poder modificar la editorial que hemos seleccionado.

00015	COFAS S.A.	ESPAÑA		X
00029	CONCYTEC S.A.	PERU		X
00004	COSMOS	RUSIA		X
00019	ECOE	COLOMBIA		X

MODIFICAR-EDITORIAL

Autogenerada

COSMOS

RUSIA

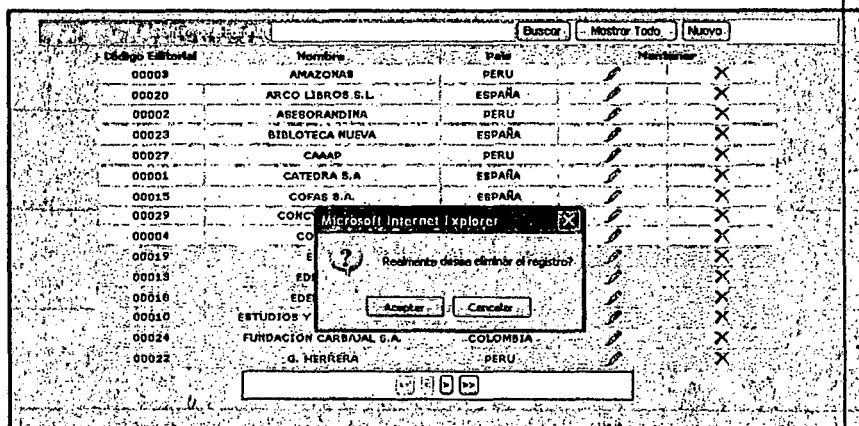
- f) El Botón “Volver”: nos permite regresar a la pantalla anterior mostrándonos todos los registros de editoriales que existe en el sistema.

Una vez modificado los campos de nombre o país de la editorial, para guardar los cambios hacemos click en el botón “Guardar”. Mostrándonos el siguiente mensaje de conformidad que los datos se ha guardado correctamente.



Al hacer clic en regresar volvemos a la pantalla inicial del mantenimiento del autor mostrándonos todas las editoriales que existe en el sistema.

- g) **Botón “Eliminar”:** Este se encuentra diferenciado por unos equis “X” que se muestra; al hacer clic sobre la equis nos muestra un mensaje de alerta preguntándonos: “Si realmente deseamos eliminar el registro seleccionado?”.

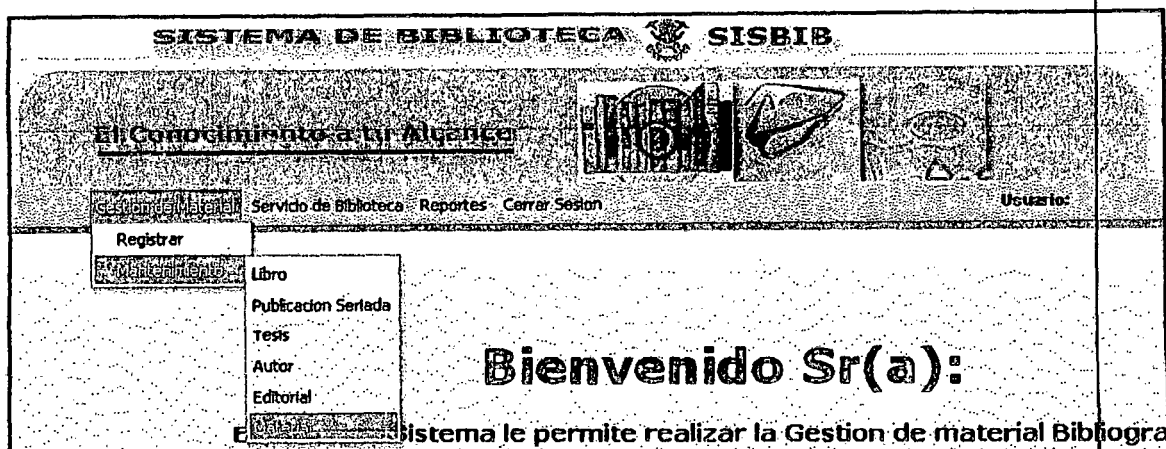


Si hacemos clic en el botón “Aceptar” el sistema verificará si la editorial se encuentra en un registro de un material, si el caso no se encuentra, se procede a eliminar mostrando el mensaje de que se eliminó con éxito, caso contrario el sistema mostrará un mensaje que la editorial no se puede eliminar porque se encuentra registrado en un material.

Si hacemos clic en el botón “Cancelar”, no se procede a la eliminación del registro cerrándose la ventana de alerta.

d) **Materia.**

Este submenú permite el acceso a la ventana “Mantener Materia” el cual aparecerá instantes después de seleccionar esta barra de menú.



Ventana "Mantenimiento Materia".



➤ Descripción.

La barra principal del menú se mantiene y tiene las mismas características. Mas abajo en la parte inferior central se vera una caja de texto vacía con tres botones que posteriormente indicaremos sus funciones.

➤ Funciones.

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para poder mantener "modificar o eliminar", básicamente el proceso consiste en buscar la materia y luego realizar la modificación u eliminación según las necesidades del usuario:

- a) Caja de texto "Ingrese Nombre de la materia": Esta aparecerá por defecto al inicio, tiene una aplicación que le ayudara a simplificar. Puede ingresar por ejemplo la letra "a" el

sistema buscará todas las materias cuyo nombre la letra "a" la búsqueda se hará mas exacta cuando digitamos con mayor exactitud el nombre de la materia que se desea buscar; esto funcionará al hacer clic en el botón "Buscar", desplegándose una lista de todas las editoriales que contengan la palabra que se digito en la caja de texto.

- b) **Botón Buscar:** Una vez ingresado el nombre de la materia a modificar o eliminar presionamos el botón "buscar" para que el sistema realice la búsqueda según la palabra ingresada en la caja de texto mostrándonos una lista de las materias coincidente con la palabra, si existe mas de quince registros coincidente con la búsqueda el sistema muestra agrupando los registros de quince en quince como se muestra en la figura siguiente:

Codigo Materia	Nombre	Mantener
00001	MATEMATICAS	X
00002	CIENCIAS HUMANAS	X
00003	METODOLOGIA DE LA OBSERVACION	X
00004	INVESTIGACION CIENTIFICA	X
00007	REDACION DE DOCUMENTOS	X
00017	PLANIFICACION	X
00019	TECNICAS CUANTITATIVAS	X
00022	EDUCACION	X
00023	CREATIVIDAD	X
00025	GUIA DE TESIS	X
00027	RECOLECCION DE DATOS	X
00028	MANEJO DE DATOS	X
00029	HIPOTESIS CIENTIFICA	X
00031	DISEÑOS EXPERIMENTALES	X
00032	BIBLIOGRAFIAS	X

- c) **Botón "Mostrar Todo":** Al hacer clic en este botón nos muestra todas las materias registrados en el sistema ordenados alfabéticamente ascendentes.

Buscar				Mostrar Todo	Nuevo
Código Materia	Nombre				
00062	ADMINISTRACION			X	
00026	AGRICULTURA			X	
00030	AMAZONIA			X	
00059	ANIMACION			X	
00063	BASE DE DATOS			X	
00064	BIBLIOTECOLOGIA			X	
00032	BIBLIOGRAFIAS			X	
00033	CIBERNETICA			X	
00024	CIENCIA			X	
00002	CIENCIAS HUMANAS			X	
00076	CIENCIAS POLITICAS			X	
00073	CIENCIAS SOCIALES			X	
00052	CIRCUITOS DIGITALES			X	
00066	COLECCIONES DE ENSAYOS			X	
00030	COMERCIO			X	

d) **Botones de desplazamiento de registros:** Esto sirve para mostrar la siguiente lista de materias para el caso de que exista mas de quince registro según el criterio de búsqueda que le hemos dado; estos botones son:

 **Botón "Adelante":** Permite avanzar los siguientes quince registros de la búsqueda.

 **Botón "Últimos Registros":** Permite avanzar hasta los últimos registros de la búsqueda.

 **Botón "Atrás":** Permite desplazarnos hacia atrás mostrando los quince registros anteriores.

 **Botón "Primeros Registros":** Permite desplazarnos hacia los quince primeros registros.

e) **Botón "Modificar":** Este se encuentra diferenciado por un lápiz que se muestra en la lista de nuestra búsqueda; al hacer clic sobre el lápiz nos muestra el formulario para poder modificar la materia que hemos seleccionado.

00033	CIBERNETICA			X
00024	CIENCIA			X
00002	CIENCIAS HUMANAS			X
00076	CIENCIAS POLITICAS			X

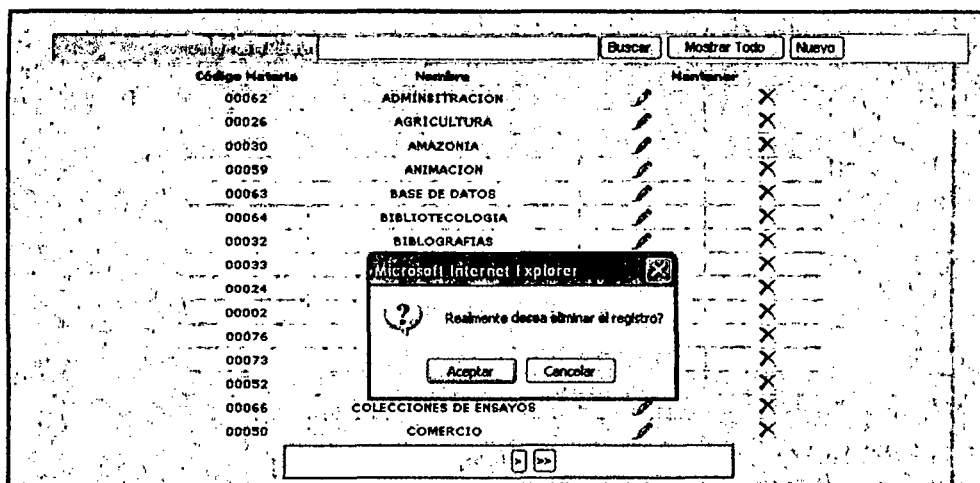
- f) **El Botón “Volver”:** nos permite regresar a la pantalla anterior mostrándonos todos los registros de materias que existe en el sistema.

Una vez modificado los campos de nombre de la materia, para guardar los cambios hacemos click en el botón **“Guardar”**. Mostrándonos el siguiente mensaje de conformidad que los datos se ha guardado correctamente.



Al hacer clic en regresar volvemos a la pantalla inicial del mantenimiento del autor mostrándonos todas las materias que existe en el sistema ordenado alfabéticamente ascendente.

- g) **Botón “Eliminar”:** Este se encuentra diferenciado por unos equis “X” que se muestra; al hacer clic sobre la equis nos muestra un mensaje de alerta preguntándonos: “Si realmente deseamos eliminar el registro seleccionado?”.



Si hacemos clic en el botón “Aceptar” el sistema verificará si la materia se encuentra en un registro de un material, si el caso no se encuentra, se procede a eliminar mostrando el mensaje de que se eliminó con éxito, caso contrario el sistema mostrará un mensaje que la materia no se puede eliminar porque se encuentra registrado en un material.

Si hacemos clic en el botón “Cancelar”, no se procede a la eliminación del registro cerrándose la ventana de alerta.

2.2.2. Servicio de biblioteca.

a) Registrar Préstamo.

Este submenú permite el acceso a la ventana “Registrar Préstamo” o sea cuando se quiere registrar el préstamo de algún material para

que el alumno se llevé. Este aparecerá instantes después de seleccionar esta barra de menú.

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestion de Material Administración de Usuarios Reportes Estadísticas Cerrar Sesión Usuarios ROGER LAVI CHAVEZ

Validar Ingreso
Registrar Devolucion
Validar Prestamo
Registrar Autorizacion
Registrar Lecturas

Bienvenido Sr(a):

El presente Sistema le permite realizar la Gestion de material Bibliografico

Ventana "Registrar Préstamo".

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestion de Material Servicio de Biblioteca Reportes Cerrar Sesión Usuarios ROGER LAVI CHAVEZ

REGISTRAR PRESTAMO

Consultar

➤ Descripción.

La barra de inicio se mantiene y mantiene las mismas características. Se vera una caja de texto vacía que se llenará para registrar el préstamo de algún material.

➤ **Función.**

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para registrar el préstamo, estas son:

a) **Caja de texto “Código de Usuario”:** Se tendrá que redactarlo, ingresando la información a través del teclado.

b) **Caja de texto “Código de Material”:** Se tendrá que redactarlo, ingresando la información a través del teclado.

The screenshot shows a window titled "REGISTRAR PRESTAMO". It contains two input fields: "Código de Usuario" with the value "0002020143" and "Código de Material" with the value "0000000011". Below these fields is a button labeled "Consultar".

c) **Botón “Consultar”:** Este abrir un cuadro que muestra la información del alumno y del material que se esta prestando.

The screenshot shows a window titled "REGISTRAR PRESTAMO" displaying the details of a loan. It lists the following information:

- Código de Usuario: 0002020143
- Nombre: EDSON MIGUEL
- Apellido: LINARES RODRIGUEZ
- Código de Material: 0000000011
- Titulo: LA ERA DE CIBERNETICA
- Auto: AMA
- Edición: BRUNO
- Temática: INGENIERIA

 At the bottom of the window is a button labeled "Registrar".

d) **Botón “Registrar”:** Este finalizara la transacción del préstamo ingresando al sistema la información. Apareciendo instantes después un respuesta de éxito.

b) Registrar Devolución.

Ventana "Registrar Devolución".

➤ Descripción.

La barra de inicio se mantiene y mantiene las mismas características. Se vera una caja de texto vacía que se llenará para consultar el préstamo de algún material

➤ Función.

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para hacer la devolución del material, estas son:

- a) Caja de texto "Código de Usuario": Se tendrá que redactarlo, ingresando la información a través del teclado.
- b) Caja de texto "Código de Material": Se tendrá que redactarlo, ingresando la información a través del teclado.

CONSULTA DE DEVOLUCION		
Código de Usuario	0002020140	
Código de Material	000000008L	<input type="button" value="Consultar"/>
Regresar		

- c) Botón "Consultar": Este abrir un cuadro que muestra la información del alumno y del material que se ha prestando.

REGISTRAR DEVOLUCIÓN						
Alumno		HORMAZA CUBA, JORGE LUIS				
Código de Usuario	Código de Material	Alumno	Material	Fecha de Devolución	Cantidad	Acciones
0603070001	000000008L	ANA ROSA OLMEDO HOYOS	EL SER HUMANO	06/03/2007	0	  
Regresar						<input type="button" value="Devolver Material y Multa"/>

- d) Botón "Devolver Material Multa": Es la imagen redondeada de color rojo. Registra la devolución del material y el pago completo de la multa, devolviendo éxito si es que realizo correctamente la operación.

REGISTRAR DEVOLUCIÓN

Nombre: HORMAZA CUBA, JORGE LUIS

Código Prestamo	Código Material	Auto	Título	Fecha de Devolución	Cant	Acciones
0603070001	000000008L	ANA ROSA OLMEDO HOYOS	EL SER HUMANO	06/03/2007	0	  

[Devolver Solo Material](#)

[Regresar](#)

- e) Botón “Devolver solo material”: Es la imagen redondeada de color rojo. Devuelve el material pero la multa queda pendiente, devolviendo éxito si es que realizo correctamente la operación.

REGISTRAR DEVOLUCIÓN

Nombre: HORMAZA CUBA, JORGE LUIS

Código Prestamo	Código Material	Auto	Título	Fecha de Devolución	Cant	Acciones
0603070001	000000008L	ANA ROSA OLMEDO HOYOS	EL SER HUMANO	06/03/2007	0	  

[Reportar perdida](#)

[Regresar](#)

- f) Botón “Reportar Perdida”: Es la imagen redondeada de color rojo. Da de baja el material quedando pendiente la devolución del libro.

c) Validar Préstamo.



El Conocimiento a tu Alcance.

Gestion de Material [Registrar Prestamo](#) [Reportes](#) [Cerrar Sesión](#) Usuario:

[Registrar Prestamo](#)

[Registrar Devolucion](#)

[Validar Prestamo](#)

Ventana “Validar Préstamo”.

El Conocimiento a tu Alcance

Gestion de Material Servicio de Biblioteca Reportes Cerrar Sesión

Usuario: ROGER LAVI CHAVEZ

➤ **Descripción.**

La barra de inicio se mantiene y mantiene las mismas características. Se vera una caja de texto vacía que se llenará para validar el préstamo de algún material para ello el usuario externo debió haber hecho una reservación del material.

➤ **Función.**

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para hacer la validación del material, estas son:

- a) **Caja de texto “Código de Usuario”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- b) **Botón “Buscar”:** Este ejecuta la búsqueda según el código y muestra un cuadro que muestra la información del alumno y del material que se ha reservado.

VALIDAR PRESTAMO

CODIGO	TITULO	AUTOR	OPCIONES
545ss	HKJH	LEON SORIA HEBREO	<input type="button" value="Validar"/>

SUBTITULO: TODO MVC333
 EDITORIAL: PONTIFICIA UNIVERSIDAD CATOLICA
 EDICION: GHJGH, 1545
 AUTORES: GKGJHDSADADADAD
 RESUMEN: JIOJIOASD
 No PAGINAS: 12
 Hora de Reserva: 11:43:55 AM

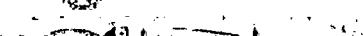
- c) **Botón “Validar”:** Este ejecuta la validación del material que ha realizado la reserva.

d) Botón “Ver Detalle”: Muestra el detalle del material reservado por el usuario.

2.2.3. Consultar Material en Línea.

Consultar Material.

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance


Consulta de Libros en Linea

Ventana “Consultar Material en Línea”.

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance



Consulta de Libros en Línea

Libro
▼

Autor
▼

CODIGO	TITULO	AUTOR	OPCIONES
as25 err5	EL SER HUMANO	OLMEDO HOYOS ANA ROSA	 
qwe qwe25	SDFSD	LEON SORIA HEBREO	 
545ss 6545SDaa	HKJH	LEON SORIA HEBREO	 
sddf fds34	HKJH	LEON SORIA HEBREO	 
44646 as34	EL SER HUMANO Y SUS PATOLOGIAS	LEON SORIA HEBREO	 

➤ **Descripción.**

Este escenario se vera a través de la WEB, para realizar la consulta del material en línea; esta búsqueda puede realizarse por varios parámetros como se muestra en la figura anterior.

➤ **Función.**

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para hacer la validación del material, estas son:

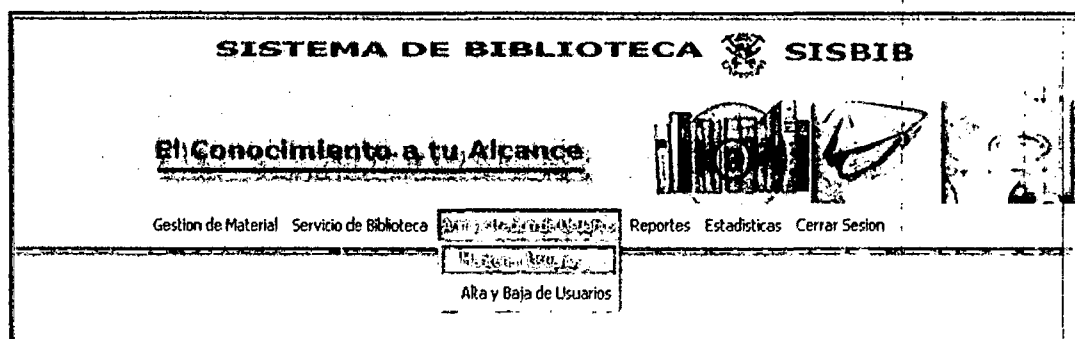
- a) **Caja de texto de selección “Buscar”:** Se tendrá que seleccionar el tipo de material Libro, Tesis o Publicación seriada a buscar.
- b) **Caja de texto de selección “Consultar Por”:** Se tendrá que seleccionar si se buscara por Autor, Título, Editorial, Resumen o Materia.
- c) **Caja de texto “Información a Consultar”:** Se tendrá que redactar la información a buscar.
- d) **Botón “Buscar”:** Ejecuta la búsqueda según los parámetros que hayamos ingresado.

Consulta de Libros en Línea			
<input type="text" value="as25 err5"/>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Libro ▼</div>	<input type="text" value="a"/>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Autor ▼</div>
<input type="button" value="Consultar"/>			
CODIGO as25 err5	TITULO EL SER HUMANO	AUTOR OLMEDO HOYOS ANA ROSA	OPCIONES
SUBTITULO: PATOLOGIA DEL SER HUMANO			
EDITORIAL: AGAPEA			
EDICION: INGLATEERA , 1985			
AUTORES: RATERNS/PETERSEN			
RESUMEN: TODA LA PATOLOGIA DEL SER HUMANO			
Nº PAGINAS: 180			
Nro de Ejemplares: 1			

2.2.4. Gestión de usuarios.

Mantener Usuarios.

Para ello nos dirigimos al menú “Administración de Usuarios”, luego al submenú “Mantener usuario” que permite el acceso a la ventana cuando se quiere registrar al usuario que harán uso del sistema. Este aparecerá instantes después de seleccionar esta barra de menú.



Ventana “Mantener Usuarios”.

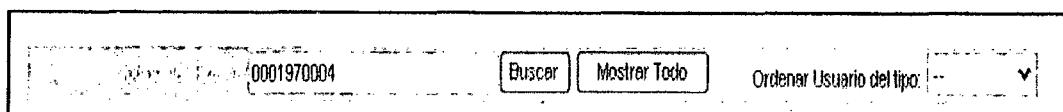
➤ Descripción.

La barra de inicio se mantiene y mantiene las mismas características. Se vera una caja de texto vacía que se llenará con el código del usuario.

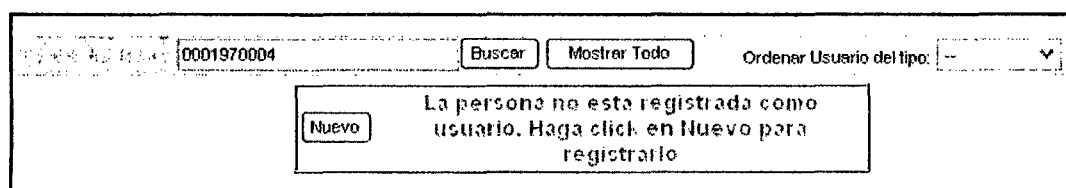
➤ Función.

El formulario consta de botones y de cajas de texto donde se introducirá la información necesaria para mantener al usuario, estas son:

- a) **Caja de texto “Código de Usuario”:** Se tendrá que redactarlo, ingresando la información a través del teclado.



- b) **Botón “Buscar”:** Una vez digitado el código del usuario presionamos en el botón “Buscar”, el sistema buscará al usuario con este código y si este no está registrado aparecerá un mensaje de color rojo de la siguiente forma “La persona no está registrada como usuario, Haga clic en Nuevo para registrarlo”.



- e) **Botón “Nuevo”:** Al hacer clic en “Nuevo” nos aparecerá un formulario con los datos del usuario.

En este formulario se ingresará la contraseña y el tipo de usuario; estos pueden ser: Alumnos, Docentes, Trabajadores, Biblioteca admin., Biblioteca Gestión, Biblioteca Préstamo, Biblioteca Ingreso; estos son los tipos de usuarios que maneja el sistema; para terminar el proceso de registro de usuario debemos hacer clic en el botón “Guardar”, el sistema automáticamente dará de “Alta” a este usuario.

Ventana de Registro de Nuevo Usuario del Sistema

NUEVO- USUARIO

Nombre	<input type="text"/>
Apellido	<input type="text"/>
Correo Electrónico	<input type="text"/>
Contraseña	<input type="password"/>
Confirmar Contraseña	<input type="password"/>
Tipo de Usuario	<input type="text" value="Alumno"/>
<input type="button" value="Guardar"/>	
<input type="button" value="← Volver"/>	

Nota: El botón “Volver”, tiene la misma función que el botón “Mostrar Todo”, regresa al formulario mostrándonos todos los registros de los usuarios del sistema.

Código Usuario	Nombre	Apellidos	Tipo	Estado	Mantener
0000014105	LEONARDO	MEDINA AZAÑEDO	Docente	Alta	X
0002020140	JORGE LUIS	HORMAZA CUBA	Alumno	Alta	X
0002020142	ROGER	LAVI CHAVEZ	Biblioteca Admin	Alta	X
0002020146	JAVIER	LUHA VELASQUEZ	Biblioteca Prestamo	Alta	X
0002020156	ALEXANDER CLEN	RIVA CCACCAYCUCHO	Biblioteca Gestión	Alta	X
0002020171	YUMI	TOMIHAGA GARCIA	Biblioteca Ingreso	Alta	X

f) Botón “Mostrar Todo”: Al hacer clic en este botón nos muestra todos los usuarios registrados en el sistema.

g) Caja de Selección “Ordenar usuario del tipo”: Al hacer clic en la caja de selección nos filtra los registros según lo escogido.

<input type="text"/> <input type="button" value="Buscar"/> <input type="button" value="Mostrar Todo"/> Ordenar Usuario del tipo: --					
Código Usuario	Nombre	Apellidos	Tipo	Estado	Mantener
0002020142	ROGER	LAVI CHAVEZ	Biblioteca Admin	Alta	
0002020146	JAVIER	LUNA VELASQUEZ	Biblioteca Prestamo	Alta	
0002020156	ALEXANDER CLEH	RIVA CCACCAYCUCHO	Biblioteca Gestión	Alta	
0002020171	YUMI	TOMINAGA GARCIA	Biblioteca Ingreso	Alta	

Nota:

- Para modificar el tipo o cambiar de contraseña hacemos clic en el “Lápiz” y se procede a modificar los datos del usuario, para terminar el proceso de modificar hacemos clic en el botón “Guardar”.

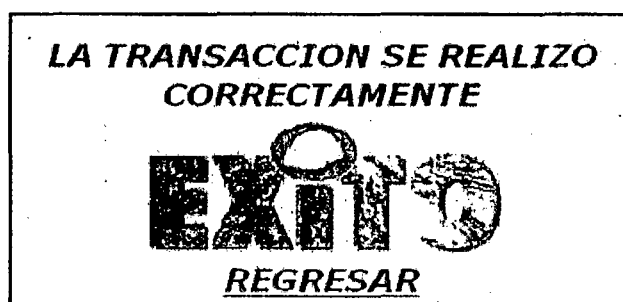
MODIFICAR- USUARIO

Biblioteca Prestamo ▼



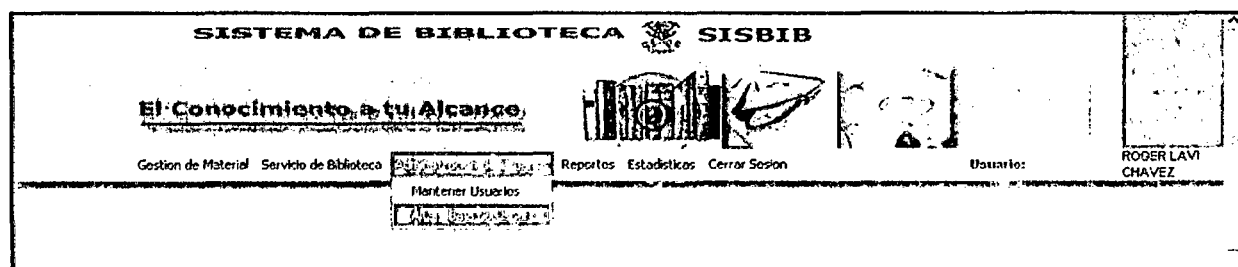
Para Eliminar al usuario hacemos clic en la “Equis” y aparecerá un mensaje preguntando si estamos seguro de eliminar el registro”. Si hacemos clic en el botón “Aceptar” elimina el registro y aparecerá un mensaje de “Éxito” de la operación; si hacemos clic en el botón “Cancelar”, no se procede a la eliminación del registro cerrándose la ventana de alerta.

Código Usuario	Nombre	Apellidos	Tipo	Estado	Mantener
0000014105	LEONARDO			Alta	X
0002020140	JORGE LUIS			Alta	X
0002020142	ROGER		Admin	Alta	X
0002020146	JAVIER		Estamo	Alta	X
0002020156	ALEXANDER CLEN		Estión	Alta	X
0002020171	YUMI		ngreso	Alta	X



Alta y Baja de Usuarios.

Para ello nos dirigimos al menú “Administración de Usuarios”, luego al submenú “Alta y Baja de Usuarios” que permite el acceso a la ventana cuando se quiere dar de “Alta” o “Baja” al usuario que utilizan el sistema. Este aparecerá instantes después de seleccionar esta barra de menú.



Ventana “Alta y Baja de Usuarios”.

ALTA / BAJA DE USUARIO	
<input type="text" value="Ingrese Código de Usuario"/>	<input type="button" value="Buscar"/>

➤ **Función.**

El formulario consta de un botón y caja de texto donde se introducirá la información necesaria para dar de alta o baja al usuario, estas son:

- **Caja de texto “Código de Usuario”:** Se tendrá que redactarlo, ingresando la información a través del teclado.
- **Botón “Buscar”:** Una vez ingresado el código del usuario hacemos clic en el botón “Buscar”, el sistema buscara los datos del usuario con este código.
- **Nota:** Si el sistema encuentra los datos del usuario con este código, muestra el siguiente formulario:

ALTA / BAJA DE USUARIO

Introduce el código de Usuario:



El Usuario esta de Alta

Desea darle de Baja ?

El sistema mostrara el estado del usuario “Baja” o “Alta” si esta de “Alta”: Hacemos clic en el botón “Dar de Baja”, el sistema mostrará un mensaje de éxito de la operación.

Si esta de “Baja” hacemos clic en el botón “Dar de Alta”, el sistema mostrará un mensaje de éxito de la operación.

Ahora si luego de digitar el código del usuario y lo buscamos y el sistema no encuentra los datos aparecerá como esta grafica:

ALTA / BAJA DE USUARIO

La persona no se encuentra como usuario en la Base de datos

Por el cual el usuario no esta registrado como usuario del sistema:

2.2.5. Reportes.

a) Asistencia.

Diaria.

SISTEMA DE BIBLIOTECA
SISBIB

El Conocimiento a tu Alcance

Gestion de Material Servicio de Biblioteca Administracion de Usuarios

Estadísticas
Cerrar Sesión

Bienvenido Sr(a):

Ventana "Asistencia Diaria".

Universidad
Nacional de
Ucayali

REPORTE DE BIBLIOTECA

VISITAS DIARIAS

Fecha: 06/03/2007

Hora: 06:12:20 AM

Seleccione usuario ▼

Fecha de visitas:

Marzo, 2007

sem	Lun	Mar	Mié	Jue	Vie	Sáb	Dom
9					1	2	3
10	5	6	7	8	9	10	11
11	12	13	14	15	16	17	18
12	19	20	21	22	23	24	25
13	26	27	28	29	30	31	

Seleccionar fecha

➤ **Descripción.**

Este escenario permite ver en forma texto la asistencia diaria de los alumnos y docentes a las instalaciones de la biblioteca.

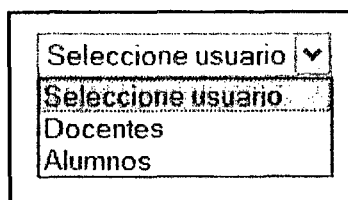
Nota: El reporte de asistencia aparecerá siempre y cuando se haya registrado la asistencia en el modulo de ingreso a la biblioteca.

Los Reportes si seleccionamos por alumno aparecerán separados por Escuelas existente en la Universidad Nacional de Ucayali.

➤ **Función.**

El formulario consta de botones y de cajas de texto Selección donde se introducirá la información necesaria para hacer el reporte de la asistencia diaria, estas son:

- a) **Caja de texto “Selección de Usuario”:** Se tendrá que seleccionar el tipo de usuario a sacar el reporte.



The image shows a screenshot of a web application's user selection dropdown menu. The menu is open, displaying the following options: 'Seleccione usuario' (with a downward arrow icon), 'Seleccione usuario', 'Docentes', and 'Alumnos'. The menu is enclosed in a rectangular border.

- b) **Caja de Texto “Fecha de visitas”:** Se tendrá que ingresar la fecha a través del botón “...” donde nos muestra un calendario para seleccionar la fecha.

Fecha de visitas:

Buscar

Marzo, 2007

sem	Lun	Mar	Mié	Jue	Vie	Sáb	Dom
9				1	2	3	
10	5	6	7	8	9	10	11
11	12	13	14	15	16	17	18
12	19	20	21	22	23	24	25
13	26	27	28	29	30	31	

Seleccionar fecha

c) Botón “Buscar”: Ejecutará la operación del reporte según los parámetros ingresados, y nos devolverá la lista de asistencia de la siguiente manera:

REPORTE DE BIBLIOTECA
VISITAS DIARIAS

Fecha: 06/03/2007
 Hora: 06:26:22 AM

Seleccione usuario: Fecha de visitas: 06/03/2007 Buscar

principal

	Nro de Visitas
CONTABILIDAD	
MAICK SILVA SALDAÑA	1
INGENIERIA DE SISTEMAS	
JORGE LUIS HORMAZA CUBA	1
MAY RAFAEL HOYOS TELLO	1
TOTAL GENERAL	3

Imprimir

Nota: Luego de mostrarnos la lista de asistencia aparecerá subrayada de color azul el link “Imprimir” al hacer clic sobre él nos permitirá imprimir el reporte.

Préstamo Por Fecha.

SISTEMA DE BIBLIOTECA SISBIB

El Conocimiento a tu Alcance

Gestion de Material Servicio de Biblioteca Administración de Usuarios

Reportes Estadísticas Cerrar Sesión

Asistencia

Préstamo por Fecha

Material mas Prestado

Usuarios multados

Ventana "Préstamo por Fecha".

The screenshot shows the SISBIB web interface. At the top, it says 'SISTEMA DE BIBLIOTECA' and 'SISBIB'. Below that is the slogan 'El Conocimiento a tu Alcance'. A navigation menu includes 'Gestion de Material', 'Servicio de Biblioteca', 'Administracion de Usuarios', 'Reportes', 'Estadísticas', and 'Cerrar Sesión'. The main content area is titled 'REPORTE DE BIBLIOTECA' and 'PRESTAMOS POR FECHA'. It features a date field 'Fecha: 06/03/2007', a time field 'Hora: 06:39:40 AM', and a search button 'Buscar'. There are also input fields for 'DESDE:' and 'HASTA:' with calendar icons.

➤ Descripción.

Este escenario permite ver en forma texto los préstamos realizados, estos reportes puede ser "desde una fecha hasta una fecha".

➤ Función.

El formulario consta de botones y de cajas de texto Selección donde se introducirá la información necesaria para hacer el reporte de los préstamos por fechas, estas son:

- Caja de Texto "DESDE":** Se ingresara la fecha a través del botón "...", el cual nos mostrará un calendario para seleccionar la fecha de inicio o desde.
- Caja de Texto "HASTA":** Se ingresara la fecha a través del botón "...", el cual nos mostrará un calendario para seleccionar la fecha de fin o hasta.
- Botón "Buscar":** Ejecutará la operación del reporte según los parámetros ingresados, y nos devolverá el número de préstamos entre las fechas ingresadas.

Material más prestado.

The screenshot shows a web interface for a library system. At the top, there is a header with the text "El Conocimiento a tu Alcance" and a logo. Below the header, there is a navigation menu with links: "Gestion de Material", "Servicio de Biblioteca", "Administración de Usuarios", "Reportes", "Estadísticas", and "Cerrar Sesión". The "Reportes" link is highlighted, and a dropdown menu is visible with options: "Asistencia", "Prestamos Por Fecha", "Material Prestado", and "Usuarios multados".

Ventana "Material mas Prestado".

The screenshot shows a window titled "SISTEMA DE BIBLIOTECA SISBIB". Below the title, there is a header with the text "El Conocimiento a tu Alcance" and a logo. Below the header, there is a navigation menu with links: "Gestion de Material", "Servicio de Biblioteca", "Administración de Usuarios", "Reportes", "Estadísticas", and "Cerrar Sesión". The "Reportes" link is highlighted, and a dropdown menu is visible with options: "Asistencia", "Prestamos Por Fecha", "Material Prestado", and "Usuarios multados". Below the navigation menu, there is a section titled "UNIVERSIDAD NACIONAL DE UCAVALI" and "REPORTES DE LOS 20 MATERIALES MAS PRESTADOS". Below this, there is a form with fields for "DESDE" and "HASTA", and a "Buscar" button. Below the form, there is a table with columns: "Ranking", "Codigo Material", "Titulo", "Año de Edición", and "Numero de Prestamos". At the bottom left, there is a "Imprimir" button.

➤ Descripción.

Este escenario permite ver en forma texto la los veinte (20) materiales mas prestados, estos reportes puede ser "desde una fecha hasta una fecha".

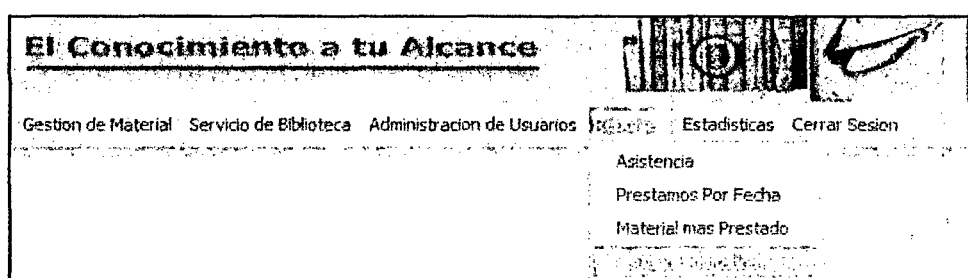
➤ Función.

El formulario consta de botones y de cajas de texto Selección donde se introducirá la información necesaria para hacer el reporte de los préstamos por fechas, estas son:

- Caja de Texto "DESDE":** Se ingresara la fecha a través del botón "...", el cual nos mostrará un calendario para seleccionar la fecha de inicio o desde.

- b) **Caja de Texto “HASTA”:** Se ingresara la fecha a través del botón “...”, el cual nos mostrará un calendario para seleccionar la fecha de fin o hasta.
- c) **Botón “Buscar”:** Ejecutará la operación del reporte según los parámetros ingresados, y nos devolverá los veinte (20) materiales mas prestados entre las fechas ingresadas.

Usuarios Multados.



Ventana “Usuarios Multados”.

Nombres	Apellidos	Titulo del Material	Multa S/.
[Empty rows for data]			

➤ Descripción.

Este escenario permite ver en forma texto los Usuarios que tienen multa según la política que maneja la Biblioteca Central de la Universidad nacional de Ucayali, estos reportes puede ser “desde una fecha hasta una fecha”.

➤ Función.

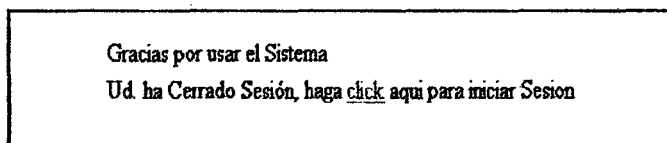
El formulario muestra con nombres, apellidos, titulo del material y la multa de los usuarios.

2.2.6. Cerrar Sesión

Este menú le permitirá salir del sistema cuando se solicite. Tan solo se deberá arrastrar el cursor sobre este menú y hacer clic para iniciar la salida del usuario del sistema "Log out".



A continuación aparecerá una pantalla que le indicara el éxito en la operación



1808T