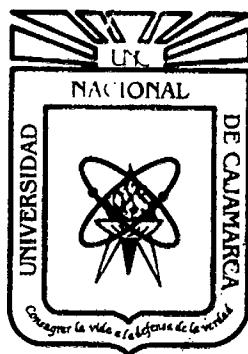


UNIVERSIDAD NACIONAL DE CAJAMARCA
FACULTAD DE INGENIERÍA
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS



PROYECTO PROFESIONAL:

**“DESARROLLO DE UN SISTEMA PARA EL CONTROL
BIBLIOGRÁFICO DE LA BIBLIOTECA MUNICIPAL JOSÉ
GÁLVEZ DE LA MUNICIPALIDAD DE CAJAMARCA”**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO DE SISTEMAS**

PRESENTADO POR EL BACHILLER:
Hernán Vargas Cueva

ASESORA:
Ing. Sandra Cecilia Rodríguez Ávila

Cajamarca - Perú
- 2013 -

Agradecimientos y dedicatoria

Agradezco a mi asesora la Ingeniera Sandra Cecilia Rodríguez Ávila por su importante apoyo en la realización de este proyecto profesional.

Agradecer a mis docentes de la Facultad de Ingeniería de la Universidad Nacional de Cajamarca, que me transmitieron sus conocimientos teóricos y sus experiencias profesionales, lo que ha enriquecido mi formación profesional.

A mis amigos(as), compañeros(as) de trabajo, en particular a los de la Municipalidad Provincial de Cajamarca, en particular a determinado personal de la Oficina de la Biblioteca Municipal "José Gálvez"

*A mi Madre, por ser una mujer digna
que siempre luchó por la superación de sus
hijos, pese a la adversidad,*

*A mi Padre que en vida, siempre nos
inculco desde pequeños a luchar por la
justicia y a defender el medio ambiente.*

Y a mis Hermanos. . .

Índice

| | |
|--|-----------|
| Agradecimientos y dedicatoria | II |
| Índice | IV |
| Resumen | VI |
| Introducción | 1 |
| Capítulo 1. Marco Referencial | 2 |
| 1.1. Objetivos | 3 |
| A) Objetivo General | 3 |
| B) Objetivos Específicos | 3 |
| 1.2. Antecedentes | 4 |
| A) Reseña histórica de la Municipalidad Provincial de Cajamarca. | 4 |
| B) Giro del Negocio. | 13 |
| Capítulo 2. Marco Teórico | 15 |
| 2.1. conceptos | 15 |
| 2.2. Metodología | 24 |
| A) El Proceso Unificado | 25 |
| B) Características del Proceso Unificado | 26 |
| a) Desarrollo basado en componentes. | 26 |
| b) El Proceso Unificado está dirigido por Casos de Uso. | 26 |
| c) El Proceso Unificado está centrado en la arquitectura | 27 |
| d) El Proceso Unificado es iterativo e incremental | 27 |
| C) Las mejores prácticas de la ingeniería de software | 28 |
| a) Desarrolle iterativamente | 28 |
| b) Administre requerimientos | 28 |
| c) Use arquitectura de componentes | 28 |
| d) Modele visualmente | 28 |
| e) Verifique calidad | 29 |
| f) Controle cambios | 29 |
| D) Elementos principales del modelado | 29 |
| E) Descripción del Proceso | 29 |
| F) Flujos de trabajo fundamentales | 30 |
| a) Captura de Requerimientos | 30 |
| b) Análisis | 32 |
| c) Diseño | 32 |
| d) Implementación | 33 |
| e) Prueba | 33 |
| G) Fases | 33 |
| a) Fase de Inicio | 34 |
| b) Fase de elaboración | 34 |
| c) Fase de Construcción | 34 |
| d) Fase de Transición | 35 |
| 2.3. Modelado | 35 |
| 2.4. UML | 36 |
| A) Introducción al UML | 36 |
| B) Objetivos de UML | 36 |

| | | |
|--|---|-----------|
| C) | Breve recorrido por el UML | 37 |
| a) | Vistas | 37 |
| b) | Diagramas | 37 |
| D) | Conceptos previos | 39 |
| a) | Paquetes | 39 |
| b) | Notas | 39 |
| 2.5. | Diagramas de clase | 39 |
| A) | Clases | 39 |
| B) | Relaciones | 40 |
| a) | Asociaciones | 41 |
| b) | Clases de Asociaciones | 42 |
| c) | Asociaciones reflexivas | 42 |
| d) | Agregaciones | 43 |
| e) | Composiciones | 43 |
| f) | Dependencia | 43 |
| 2.6. | Diagramas de casos de uso | 44 |
| A) | Actor | 45 |
| B) | Casos de Uso | 45 |
| C) | Relaciones entre los casos de uso | 45 |
| 2.7. | Diagrama de Estados | 46 |
| 2.8. | Diagrama de secuencias | 46 |
| 2.9. | Diagrama de Comunicaciones | 47 |
| 2.10. | Diagrama de actividades | 48 |
| 2.11. | Diagrama de componentes | 49 |
| 2.12. | Diagrama de distribución | 50 |
| CAPITULO 3. DESARROLLO METODOLÓGICO | | 54 |
| 3.1.- | INTRODUCCION | 54 |
| 3.1.1. | GESTION DE LA CONFIGURACION | 54 |
| a) | Documentación | 54 |
| □ | Modelos Construidos | 54 |
| 3.2.- | FASE I: INICIO | 56 |
| 3.2.1. | MODELADO DEL NEGOCIO | 56 |
| a) | Proceso de Inscripción al Lector | 56 |
| b) | Proceso de Atención Consultas de Material Bibliográfico | 59 |
| c) | Proceso de Atención de Préstamo y Devolución de Material Bibliográfico. | 61 |
| d) | Proceso de Administrar el Material Bibliográfico. | 65 |
| e) | Modelo de Objetos | 69 |
| f) | Proceso General de Consulta y Atención de Material Bibliográfico | 70 |
| b) | Visión Statement | 71 |
| 1. | Introducción | 71 |
| 2. | Descripción General | 71 |
| 3. | Requisitos del producto | 71 |
| b) | Modelo de Requisitos | 77 |
| c) | Especificación de Casos de Uso | 82 |
| 3.3. | FASE II: ELABORACIÓN | 92 |

| | |
|---|------------|
| 3.3.1. ANÁLISIS | 92 |
| 3.3.1.1. Diagrama de Análisis de Clases. | 92 |
| 3.3.1.2. Diagramas de Secuencia | 94 |
| 3.3.1.3. Diagrama de Clases | 105 |
| 3.4. FASE III: CONSTRUCCIÓN | 106 |
| 3.4.1. Diseño del Sistema | 106 |
| 3.4.1.3.1 Diseño de Pantallas del Sistema. | 119 |
| a) Inicio | 119 |
| Diagrama de Entidad Relación | 139 |
| 3.5. FASE IV: TRANSICIÓN | 136 |
| 3.5.1. IMPLEMENTACIÓN. | 136 |
| 3.5.1.1. TECNOLOGÍAS. | 136 |
| 3.5.1.2. Despliegue. | 137 |
| 3.5.1.3. Detalles de Implementación. | 137 |
| 3.5.2. Evaluación y Pruebas. | 141 |
| 3.5.2.1. Evaluación. | 141 |
| 3.5.2.2. Pruebas. | 142 |
| 3.5.2.2.1. Pruebas de validación. | 142 |
| Conclusiones y recomendaciones | 148 |
| Conclusiones. | 148 |
| Recomendaciones. | 149 |
| Referencia Bibliográfica | 150 |
| Anexos | 151 |
| Anexo 1: | 151 |
| <i>Documentacion utilizados en la Biblioteca</i> | 151 |
| Anexo 2 : | 155 |
| Relación de Tablas | 155 |
| Relación de ilustraciones | 155 |
| Anexo 3 : | 157 |
| Implementación de Código PHP | 159 |

Resumen

El Presente Proyecto Profesional denominado "Desarrollo de un Sistema para el Control Bibliográfico de la Biblioteca Municipal José Gálvez de la Municipalidad Provincial de Cajamarca", es un sistema desarrollado, en el cual entre sus bondades permite realizar el registro del material bibliográfico, el registro de lectores, emisión de carné de lector, registro de prestamos, llevar un control del inventario del material entre otras.

El desarrollo e implementación de este proyecto me ha permitido actualizar mis conocimientos investigando el UML (Lenguaje de Modelado Unificado) y el RUP (Proceso Unificado de Rational).

En el Marco Teórico detallo los fundamentos para comprender qué es el UML, presentando brevemente una introducción para luego dar conceptos sobre Orientación a Objetos, a continuación se describe los diagramas que usa (clases, objetos, casos de uso, estados, secuencias, actividades, componentes y distribución).

He creído conveniente hacer uso del Proceso Unificado de Rational (Rational Unified Process) que es un proceso de Ingeniería de Software debido a que el UML no tiene noción de proceso, el cual es una parte importante de un método RUP entonces me proporciona una aproximación disciplinada para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que cubre los requerimientos y expectativas de los usuarios finales.

Gracias al uso del RUP se desarrolla un proyecto de software exitoso que permite cubrir las expectativas del cliente, además me ha permitido trabajar de una forma económica y puntual, así mismo permite que el proceso de desarrollo sea controlado y medido para asegurar que el proyecto sea bien completado.

Este proceso es bastante extenso ya que puede manejar macro proyectos, para mi caso he seguido algunos de los pasos del proceso total guiando el desarrollo del mismo a lo largo del contenido en donde detallo los flujos de trabajo: modelado del negocio, requerimientos, análisis y diseño, implementación y despliegue.

Además fue necesario hacer uso de herramientas case que facilitan mi trabajo, para mi caso en particular he utilizado IBM Rational Software Architect for WebSphere Software 7.0 el mismo que facilita la generación de código para ciertos Lenguajes de Programación.

Introducción

Actualmente en la Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca cuenta aproximadamente con 18,500 libros así mismo cuenta con revistas, boletines y otros. Por tal razón es de necesidad que dicha oficina cuente con un sistema que le permita tener un control integral de la bibliografía existente y de sus usuarios.

UML (Unified Modeling Language) es un lenguaje para especificar, visualizar, construir y documentar los elementos de un sistema software, así como para modelado de procesos de negocio u otros sistemas no software. UML reúne una colección de las mejores prácticas en la ingeniería que han sido utilizadas con éxito para modelar sistemas grandes y complejos.

Dados los rápidos avances tecnológicos y las necesidades cada vez mayores de las organizaciones con respecto a sus sistemas de información. Eso quiere decir que debemos desarrollar sistemas, conscientes de que este va a cambiar (anticipar la evolución) y debemos hacer los cambios conscientes de que más adelante habrá más cambios (administrar el cambio). Además, dado las largas vidas útiles esperadas de los sistemas, se debe tener presente que los que participamos en el desarrollo del sistema no necesariamente somos los mismos que vamos a mantenerlo. Por esto, es importante en el desarrollo establecer y respetar convenciones y estándares que más adelante van a facilitar el entendimiento del sistema.

Cada día son más las actividades en las que intervienen aplicaciones de software.

Así como los dominios de aplicación de software han aumentado, también ha aumentado la complejidad y el tamaño de los sistemas. Esto es debido, en parte, a que las necesidades de los usuarios son cada vez más exigentes, en términos de interacción, seguridad, retroalimentación, etc., y también, debido a los avances tecnológicos que hacen posible que un sistema esté compuesto de múltiples subsistemas en los que intervienen bases de datos, interfaces gráficas, comunicación entre componentes, recuperación ante fallas, interacción con dispositivos multimedia, sistemas complejos de seguridad, etc.

CAPÍTULO 1. MARCO REFERENCIAL

Situación Problemática

La Municipalidad Provincial de Cajamarca tiene como uno de sus objetivos institucionales principales el de incentivar la cultura y aumentar el conocimiento de su población de su jurisdicción, una de las oficinas que contribuye es la Biblioteca Municipal "José Gálvez".

La Biblioteca desarrolla sus actividades diarias de forma manual, no cuenta con un Sistema de Información que le permita llevar un mejor funcionamiento más óptimo y confiable, que le permita brindar un mejor servicio de calidad a sus usuarios. La Biblioteca Municipal realiza todas sus operaciones diarias manualmente; cuando existe la necesidad de registrar un nuevo material bibliográfico se registra en el libro de registro de la Biblioteca, luego se registra en una ficha del libro, posteriormente se realiza una ficha topográfica del mismo, una ficha de autor y una ficha de materia, el cual resulta bastante tedioso, así mismo un usuario para que consulte si existe o no un material bibliográfico, lo realiza a través de fichas o de catálogos, pero el hecho de no tener todo el conocimiento de las ubicaciones de los libros en los distintos estantes, causa un poco de demora, y así el descontento del usuario o lector, para registrar un nuevo lector, se anota manualmente sus datos, se registra sus datos en una ficha de lector, se les registra en un libro de lectores, la ficha de lector se guarda en un archivo físico, y con estos procesos podemos determinar que es un problema el hecho de realizar la búsqueda manual de un lector ya que los papeles de registro y otros documentos se pueden extraviar o traspapelar. Otro problema que se presenta, aunque no con mucha frecuencia, es el extravió de los libros, esto se debe a que el control que se les da a los usuarios o lectores es muy limitado, esto trae como consecuencia el hecho de buscar en todo el inventario de libros de la Biblioteca, el cual es un archivo físico o catálogo que contiene todos los títulos, y dar de baja al libro extraviado; en este mismo contexto podemos mencionar el hecho de dar de baja libros demasiado antiguos y desfasados, o modificar su ubicación a otros estantes más específicos, para realizar esta acción también se recurre a la búsqueda manual del registro de libro en el catálogo o inventario principal, para su posterior eliminación o reubicación física, finalmente la consecuencia de este problema es el tiempo perdido al buscar manualmente en los registros del inventario o catálogo.

Así mismo para emitir los carné de lector lleva al personal de la biblioteca, entregar dentro de las 24 horas de haber gestionado el usuario, esto debido a que tiene que ingresar información en el libro de ingreso de lectores y también de llenar un formulario que queda archivado en la Biblioteca Municipal. Para realizar las estadísticas mensuales, trimestrales y anuales sobre los lectores y del tipo de material bibliográfico atendido, a la secretaria se lo lleva mucho tiempo por el hecho que se lo realiza manualmente.

Justificación del Problema

La Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca actualmente lleva el Control del material bibliográfico manualmente, donde cada uno de sus procesos no se realizan de manera informatizada, por esta razón se presenta una serie de inconvenientes dentro de la gestión bibliotecaria lo que lleva muchas veces a problemas de tipo administrativo como son: No se sabe con exactitud información sobre el total de material bibliográfico existente, aumento de trabajo, deficiencia en el manejo de información para la toma de decisiones.

Al desarrollar el sistema nos permitirá solucionar los problemas ya mencionados, lo cual se reflejara con un mejor servicio de calidad en la biblioteca.

1.1. OBJETIVOS

A) OBJETIVO GENERAL

Desarrollar un Sistema para el Control Bibliográfico de la Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca, que permita brindar un mejor servicio bibliotecario.

B) OBJETIVOS ESPECÍFICOS

- Realizar un análisis general del funcionamiento actual de la biblioteca municipal.
- Agilizar y sistematizar el trabajo dentro de la Biblioteca Municipal "José Gálvez", permitiendo contar con información veras y oportuna con reportes de acuerdo a las necesidades de los usuarios.
- Investigar sobre el Lenguaje de Modelado Unificado (UML) y el Proceso Unificado de Rational (RUP) y aplicarlo.

- Sistematizar los procesos de la Biblioteca Municipal, permitiendo tener información de manera consistente, fiable y oportuna para la toma de decisiones.
- Identificar los requerimientos y especificaciones del sistema.
- Diseñar un sistema a medida acorde a sus necesidades y realidad.
- Construir e implementar el sistema para el control de la Bibliografía de la Biblioteca Municipal.

1.2. ANTECEDENTES

A) RESEÑA HISTÓRICA DE LA MUNICIPALIDAD PROVINCIAL DE CAJAMARCA.

La Municipalidad de Cajamarca no cuenta con datos a cerca de su creación, su labor como distrito integrante de Trujillo fue imperceptible por que en la documentación de sus archivos se registran escasas obras. En 1855 fue creado el departamento de Cajamarca, a partir de allí la Municipalidad inicia una labor notable en el progreso de la localidad orientado a un civismo elevado y patriótico y con un profundo respeto a la dignidad y derechos del vecindario.

En 1864 el Consejo hace ante los Poderes Públicos las gestiones pertinentes para la implantación de alumbrado de gas en la ciudad y se dictan las medidas preliminares para la licitación administrativa de los Baños del Inca; además se organiza el mejor servicio de los mercados de abastos y sobre todo se mantiene estricto control sobre la educación pública.

En 1866 en históricas sesiones municipales se da cuenta de la alianza defensiva y ofensiva que celebramos con la República de Chile, para proteger nuestra independencia y así mismo se informa al pueblo sobre la declaratoria de guerra a España hecha el 14 de enero de 1866; la colectividad cajamarquina recurrió al Municipio y por intermedio de su Alcalde hizo saber al Supremo Gobierno que estaba dispuesta a prestar todo su apoyo en defensa de la libertad.

La municipalidad de Cajamarca tuvo un estrecho contacto con el pueblo y de manera solidaria, tanto la multitud como los dirigentes, derivaban

hacia un civismo pujante y proficuo. Esta armonía se nota en todas las decisiones públicas de los comienzos departamentales y es por que la primera medida que se toma en aquellos albores es la organización del "Censo Departamental" para conocer la verdadera cifra de la población.

A continuación se llevan a cabo numerosas obras públicas, como la edificación del camal, organización de los servicios de agua y luz, la súper vigilancia de los establecimientos públicos, especialmente los locales escolares y se crean impuestos municipales que gravan a las bodegas y billares , se organizó el registro de comerciantes y conductores de cantinas que debían pagar, como contraloría municipal, finalmente se dispuso que las deudas pendientes que hubieran a favor del Municipio pasaran en asignaciones a formar fondos destinados a terminar el puente de la independencia. El Municipio súper vigilaba la distribución del dinero en las obras de educación pública, acondicionado establecimientos y locales como se desprende de las actas referentes a las obras escolares en distritos alejados y de la presentación de un plano con su respectivo presupuesto para reparaciones del frontis del convento de la Merced que hace el director de la Escuela Normal de Cajamarca ante el Concejo, por cuanto la Escuela funciona en dicho Convento.

La organización urbana de la ciudad, nomenclatura de calles, numeración de jirones y factura del plano de la ciudad, los hace el Municipio para los fines de baja policía y para conocer los puntos en que debían colocarse las lámparas de gas del alumbrado público. Se observa en estos comienzos municipales de Cajamarca una inquietud Cultural, auspiciando el Municipio espectáculos que significan cultura y solaz del pueblo, contribuyendo de tal manera a su perfeccionamiento espiritual, puesto que los viejos libros de actas registran sesiones en los que hace referencia a los acuerdos de conceder licencia para 16 funciones de la Compañía Dramática Americana siempre que la última representación sea dedicada a favor de las municipalidades y Beneficencias de Cajamarca.

Parece que durante 1868 desaparecieron algunos documentos de la Corporación, a causa de la ocupación de sus oficinas que funcionaban en una parte del local de la Prefectura, por las tropas del Coronel José Balta, aunque el Alcalde de esa época manifestó no haber tenido conocimiento inmediato de

tal pérdida, lo que hace sospechar que no se solidarizó con lo que se podría llamar un descuido institucional.

En 1868 se produjeron dos entredichos jurisdiccionales, una se refiere al Prefecto de dicho año en que pide que el Concejo lo considere como Presidente de la Junta de Registro, contestándosele que no puede ser admitido sino como simple ciudadano y que en caso de insistir se suspendería los actos electorales mientras dirima la cuestión el Supremo Gobierno. La otra es relativa a la exigencia del Prefecto para que la Municipalidad entregue el instrumental perteneciente a la columna de gendarmes y no habiéndolo conseguido usó la fuerza pública para intimidar al Alcalde. Como resultado final de este entredicho el Supremo Gobierno reconoció como legal el cuerpo edilicio de esa época, resolución que se transcribió al Prefecto y se comunicó al pueblo por bando público.

En 1879, se ve interrumpida y opacada la labor municipal con la ocupación de Cajamarca por las fuerzas chilenas, sembrando un trastorno social que dio como resultado una gran pérdida de documentos y archivos y por otra parte el resurgimiento de la voluntad municipal para continuar su obra y superarla.

En 1934, cuando era Alcalde el Dr. Aurelio Zaldívar, se adquiere el Palacio Municipal, que es una bella casa de factura semi colonial ubicada a media cuadra de la plaza de Armas, que consta de amplios y cómodos departamentos donde funcionaba el Concejo Provincial de Cajamarca. La Municipalidad Provincial de Cajamarca está ubicada dentro de la Región Cajamarca y rige por la Ley Orgánica de Municipalidades Ley N° 27972 y por la Ley de Presupuesto del sector Público, teniendo domicilio legal a la actualidad en Av. La Alameda Qhapac Ñan s/n.

RESEÑA HISTÓRICA DE LA BIBLIOTECA "JÓSE GÁLVEZ"

Antecedentes

La ley N° 4506 del 24 de marzo de 1920 estableció bibliotecas populares en todas las capitales de provincias bajo la dirección inmediata de los respectivos concejos provinciales. Ordenó que en el local ocupado por el Concejo, se

designase una habitación debidamente arreglada para que en ella se estableciera la biblioteca y encomendó al secretario municipal, se encargue de su organización y dirección, en cuanto a la parte económica de la institución, se limitó a prescribir que los concejos provinciales consignaran en sus respectivos presupuestos, partidas especiales para el establecimiento y sostenimiento de las referidas bibliotecas.

Sin embargo, la ley 4506 no se aplicó. Después de expedida, ni el Congreso ni el Poder Ejecutivo se preocuparon de su cumplimiento.

Situación que causó honda preocupación en algunos ciudadanos notables de Cajamarca, pues habían transcurrido algo más de 20 años desde la desaparición de la Biblioteca Popular de Vicente Pita y Cajamarca pese a su numerosa población estudiantil no contaba con un centro cultural de esta categoría.

En 1928, Nazario Chávez Afiaga desde las columnas del diario "El Perú", inicio una intensa campaña en favor de la creación de una biblioteca.

"Su fundación decía en uno de sus artículos, no requiere si no buena voluntad del elemento dirigente y de los que viven con intensidad el momento actual, medios indispensables para alimentar ideales como éste que constituyen, a decir verdad, el termómetro de la cultura de las poblaciones. En cuanto al nombre con que se podría bautizar a la biblioteca no sería otro que el de "Biblioteca Amalia Puga", por móviles inherentes a la fama de esta insigne cajamarquina... De este modo, no sólo se ganaría para Cajamarca un centro representativo de ilustración, sino que la juventud de los distintos círculos encontrarían en ella, un sitio de recreo y formación espiritual..."

Los artículos que con cierta regularidad aparecían en "El Perú", rápidamente impactaron en un grupo de estudiantes del Colegio San Ramón, quienes agrupados bajo la dirección de don Aurelio Bazán Peralta, fundaron la asociación " Pro Biblioteca popular", integrada por Nicolás Chávarri, Luis Sousa, Víctor Sousa, Renán Arce y Aurelio Vásquez quienes, amparados en el decreto dado por Leguía que instituía el " Día del Libro", organizaron una serie de actividades, cuyo fondos recaudados servirían para la creación de la anhelada biblioteca popular.

Con los pocos recursos económicos obtenidos por erogación, y con la donación de algunos libros, en mayo de 1929, en una tienda del mercado la Merced, cedida por el alcalde municipal, Alejandro Ortiz Pérez, la biblioteca abrió sus puertas al público. Sin embargo, y pese a la incomodidad que ofrecía el local, por falta de luz y el inapropiado mobiliario que utilizaba, la biblioteca funcionó dos años con notable incremento de libros y gran afluencia de lectores. Situación que obligó al Comité a pedir a la Municipalidad un nuevo local más céntrico y con mejores condiciones que el anterior. El alcalde cedió entonces una de las tiendas de la Municipalidad, ubicada en el jirón Junín, local que a la postre, tampoco resultó adecuado.

Habían transcurrido ocho años desde su inauguración y la biblioteca no podía funcionar adecuadamente por problemas de local, de mobiliario y, sobre todo, no contaba con una subvención determinada para atender sus necesidades más apremiantes. EL Comité "Pro - Biblioteca" decidió, finalmente, entregar a la Municipalidad, para su administración, los libros y los pocos enseres que poseían.

Comprendiendo las necesidades de organizar una biblioteca municipal, el alcalde gestionó ante la Junta Pro - Desocupados para realizar las obras de enlucido y pavimentación de los salones destinados a la futura Biblioteca Municipal. Por su parte, Leoncio Alva (miembro de la mencionada sociedad), reforzó la gestión edilicia presentando, el 28 de junio de 1937, un proyecto al presidente de la Junta Pro - Desocupados, donde entre otras cosas proponía " que la Junta acuerde la ejecución de la obra en referencia con fondos Pro - Desocupados, con arreglo a los estudios y presupuesto del señor Ingeniero Teófilo Córdova Muñoz, y recabe del Ministerio de Fomento la aprobación respectiva".

Fundación.

El 29 de octubre, cuando el nuevo local ubicado en la Calle Cajamarca (hoy Cruz de Piedra N° 627) se hallaba expedito, el alcalde invitó a su apertura oficial en los términos siguientes:

" Debiendo tener lugar, el día de mañana, a las 11 a.m la bendición e inauguración de la Biblioteca Municipal " José Gálvez", en uno de los departamentos de este Municipio, arreglado expresamente por la Junta

Departamental Pro - Desocupados, tengo el honor de invitar a Ud. a dicha ceremonia, la que será apadrinada por el señor Prefecto del Departamento, don Enrique Ayulo Laos y la señora Elvira de Vivas Serra, en representación del señor Presidente de la República, General Oscar R. Benavides, y la señora Amalia Puga de Losada, respectivamente"

Su inauguración, con el nombre de Biblioteca Municipal "José Gálvez " se efectuó el domingo 30 de octubre de 1938, con la asistencia del alcalde provincial, representantes de lo poderes públicos, clero, cabildo y los demás centros de cultura y público en general.

La bendición corrió a cargo del Administrador Apostólico, Monseñor Federico Pérez S. quien antes, pronunció un elocuente discurso en el que agradecía al Gobierno, miembros de la Junta Pro - Desocupados, padrinos y personas que han ayudado en diversas formas a esta obra así como a los que han enviado libros. Remarcó la importancia para Cajamarca de tan primordial necesidad, esperando que ha de prestar los importantes servicios a que está llamada a desempeñar en nuestro medio.

La concurrencia fue numerosa y distinguida; notándose a las autoridades políticas, militares, del Poder Judicial, del Clero, empleados públicos, caballeros de nuestra sociedad y públicos en general. También a los miembros del Concejo casi en pleno y empleados.

Instalada la Biblioteca, se le adjudicó para su sostenimiento, parte del presupuesto del concejo Municipal y se nombró como bibliotecario a don Job P. Chávarry, que permaneció en el cargo hasta su muerte, ocurrida en 1945.

OBJETIVOS

El objetivo principal de la biblioteca es prestar servicios a los lectores y público en general, facilitando información.

Fomentar y capacitar al Personal de la Biblioteca Municipal y B.B. de la CCB.

Fortalecer el Desarrollo Institucional de la Biblioteca Municipal.

Difundir los servicios de la Biblioteca Municipal.

Promover el uso efectivo y permanente de los servicios que ofrece la biblioteca, en las actividades inmanentes a la cultura, la investigación y programas de extensión.

Promover el desarrollo de Redes de información y fortalecimiento de la organización y servicio bibliotecario.

Organizar las unidades de servicio que convengan para hacer alcanzable a los usuarios el conjunto de recursos disponibles.

Fomentar y apoyar las actividades culturales en Cajamarca.

Coadyuvar el objetivo de la MPC para fortalecer e incrementar el turismo en Cajamarca.

Concertar una Agenda para el trabajo Cultural de Cajamarca.

Mejorar el local de la biblioteca Municipal, la seguridad y la implementación bibliográfica y de equipos.

SERVICIOS

Consulta y Lectura en Sala.

Préstamo de Libros a Domicilio (En casos especiales de estudio o investigación, previa acreditación documentada), actualmente esta suspendida todo tipo de prestamos)

Hemeroteca.

La Hemeroteca otra alternativa de consulta a disposición del público.

Servicios de Internet

Servicio a Investigadores (Salas Especiales con servicio documentado en la líneas a investigar)

Formación del área bibliográfica de Cajamarca.

- Colección Bibliográfica
- Publicaciones periódicas,
- Información Turística.
- Biblioteca infantil.
- Encuadernación y Empaste.

Funciones de los Trabajadores de la Biblioteca Municipal.

DEL JEFE DE BIBLIOTECA

- 1.- Programar y coordinar los diversos aspectos del procesamiento técnico del material bibliográfico y documental.
- 2.- Procesar e incrementar material especial (hemeroteca, fotografías, mapas, partituras musicales y otros).
- 3.- Absolver consultas bibliográficas que requiera investigación especial y elaborar bibliografías breves sobre temas y personas.
- 4.- Participar en la elaboración de políticas sobre bibliotecología.
- 5.- Participar en la formulación del presupuesto y del Plan General de la Biblioteca, preparando informes técnicos.
- 6.- Integrar el equipo de investigación, estudios y publicaciones de la Biblioteca.
- 7.- Propiciar el intercambio de publicaciones con organismos nacionales e internacionales.
- 8.- Elaborar y revisar el Boletín Bibliográfico y otras publicaciones de la Biblioteca.
- 9.- Asesorar en asuntos de su especialidad.
- 10.- Proponer normas y directivas que tiendan a reguardar el patrimonio bibliográfico y mejorar el funcionamiento de la Biblioteca.
- 11.- Controlar la clasificación y codificación del material bibliográfico y documental de acuerdo al sistema y/o métodos científicos.
- 12.- Supervisar el ingreso y egreso de material bibliográfico.
- 13.- Propiciar campañas de utilización de las bibliotecas.
- 14.- Capacitar a los técnicos y auxiliares de la Biblioteca.
- 15.- Otras funciones afines al cargo que le asigne el Sub Gerente de Educación, Cultura y Deporte.

DEL TÉCNICO DE BIBLIOTECA

- 1.- Efectuar la clasificación y catalogación del material bibliográfico.
- 2.- Realizar estadísticas mensuales sobre nuevas adquisiciones.
- 3.- Ejecutar y supervisar la realización anual del inventario de la Biblioteca.
- 4.- Colaborar en las actividades de extensión cultural.
- 5.- Supervisar la alfabetización de las fichas.
- 6.- Absolver consultas simples y proponer bibliografía sobre materias específicas.
- 7.- Ofrecer servicios de información a los usuarios mediante préstamos de material bibliográfico.
- 8.- Apoyar en la programación de adquisición de material bibliográfico.
- 9.- Participar en la elaboración del Boletín Bibliográfico.
- 10.- Confeccionar el Registro de las suscripciones.
- 11.- Otras funciones afines al cargo que le asigne el Jefe de Biblioteca.

DEL TÉCNICO ADMINISTRATIVO

- 1.- Estudiar y participar en la elaboración de normas, procedimientos e investigaciones de procesos técnicos.
- 2.- Analizar y emitir opinión técnica sobre expedientes puestos a su consideración.
- 3.- Absolver consultas de carácter técnico del área de su competencia.
- 4.- Participar en la programación de actividades técnico administrativas, reuniones y comisiones de trabajo.
- 5.- Otras funciones afines al cargo que le asigne el Jefe de Biblioteca.

DEL AUXILIAR ADMINISTRATIVO

- 1.- Ejecutar actividades de recepción, clasificación, registro, distribución y archivo de los documentos.
- 2.- Ejecutar y verificar la actualización de registros, fichas y documentos técnicos de la Biblioteca.

- 3.- Coordinar actividades administrativas sencillas.
- 4.- Colaborar con la programación de actividades técnicas administrativas y en reuniones de trabajo
- 5.- Verificar el cumplimiento de disposiciones y procedimientos.
- 6.- Otras funciones afines al cargo que se le asigne el Jefe de Biblioteca.

DEL AUXILIAR DE BIBLIOTECA

- 1.- Recibir e ingresar la colección bibliográfica y documental.
- 2.- Acomodar en estantes y anaqueles libros y/o revistas, catálogos y ficheros.
- 3.- Orientar a los lectores en el uso de catálogos y/o ficheros de biblioteca.
- 4.- Mantener registro de los libros y documentos presentados así como la renovación
de los plazos vencidos.
- 5.- Vigilar el servicio en la sala de lectura, velando por la buena conservación del material bibliográfico.
- 6.- Colaborar con labores auxiliares de reprografía, (Xerox, microfichas, películas etc.
- 7.- Ordenar, clasificar y encuadernar las Normas Legales del Diario Oficial el Peruano.
- 8.- Distribuir a las Unidades Orgánicas indicadas el Diario Oficial el Peruano.
- 9.- Otras funciones afines al cargo que le asigne el Coordinador de Biblioteca.

B) GIRO DEL NEGOCIO.

La Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca es considerada una de las más importantes de Cajamarca, por el invaluable patrimonio bibliográfico y documental que su acervo contempla en el área de Cajamarca.

La Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca tiene como una de sus funciones de proporcionar el conocimiento y el sustento documental para la consulta de sus usuarios.

Actualmente la Biblioteca Municipal "José Gálvez" cuenta con más de 18,000 volúmenes y dentro de su acervo se contemplan varias áreas. Entre las áreas

están de: Cajamarca, Historia del Perú, Ciencias Sociales, Medicina, Agropecuaria, Derecho, Publicaciones Periódicas, Temas Generales, Libros de Consulta.

En el área de Cajamarca se encuentra la Colección Especial, que son obras valiosas publicadas que data del siglo XVIII.

El área de Publicaciones Periódicas es de importancia vital para los usuarios, ya que les permite mantenerse actualizados.

Existe, además, la sección de Libros de Consulta donde se encuentran los diccionarios, enciclopedias, atlas, etc.

En resumen, entre las materias incluidas en el acervo bibliográfico de la biblioteca se encuentran de Cajamarca, Historia del Perú, Ciencias Sociales, Medicina, Agropecuaria, Derecho, entre otras.

CAPÍTULO 2. MARCO TEÓRICO

2.1. CONCEPTOS

2.1.1. ASPECTOS DE LA INGENIERÍA DE SOFTWARE

El reciente aumento de aplicaciones en donde se utiliza la computadora ha sido posible debido gracias a que el hardware tiende a la tendencia de bajar su costo, por lo cual la demanda de software ha crecido de forma exponencial.

Esto implica que son necesarias las técnicas y tecnologías eficientes de Ingeniería de Software para resolver los múltiples problemas que se derivan de las aplicaciones en donde se desarrollan sistemas de información.

La Ingeniería de Software tiene como principal objetivo servir como base para la producción de software de calidad, lo cual se logra definiendo el proceso del software, el cual comprende las actividades involucradas en la producción del software.

La Ingeniería del software se refiere a la producción de sistemas de software desarrollados por grupos y no solo por individuos. Utiliza distintos principios de Ingeniería en el desarrollo de estos sistemas e incluye aspectos técnicos y no-técnicos. Además de contemplar aspectos de computación, los Ingenieros deben ser capaces de comunicarse con otros miembros de su grupo de desarrollo. Cada Ingeniero debe comprender la importancia de la gestión del proyecto y deben de apreciar los problemas que los usuarios del software podrían encontrar en su uso. De la misma forma, se debe tener en cuenta la calidad del software y su confiabilidad, ya que no es igual producir software para un sistema de matrícula, que un sistema de software para el control de un reactor nuclear. (1)

La Ingeniería de Software implica seguir en cualquier proyecto de software una metodología de desarrollo y la utilización de distintas técnicas y herramientas.

La Ingeniería de Software es una disciplina de la Ingeniería en donde se resuelven problemas relacionados con el desarrollo de sistemas de software. Esta disciplina no es solo programación ni computación. Los Ingenieros deben de ser profesionales que deben de utilizar la teoría de otras disciplinas y aplicarlas de forma efectiva en la solución de problemas complejos de sistemas de software. El software no solo significa "programas de computadora" asociados con una aplicación o producto. Además de esto, implica la documentación necesaria para

dar seguimiento, la instalación, el uso y el mantenimiento del software. En sistemas complejos de gran tamaño, el esfuerzo necesario para escribir esta documentación es a menudo tanto como lo es el desarrollo del software. No es posible presentar una solución global o precisa a todos los problemas de la Ingeniería de software o presentar una solución única para resolver los problemas de la Ingeniería de software. Cada proyecto de software presenta distintos problemas en su desarrollo, los cuales involucran personas, equipo, usuarios del software y ambiente de la aplicación. Por estas razones, cada proyecto debe resolver el problema de la producción del software teniendo en cuenta las distintas metodologías y técnicas de desarrollo, pero sin descuidar el aspecto humano, del usuario del software y del ambiente para el cual se pretende desarrollar el software (2).

2.1.2. LAS TENDENCIAS EN LA INGENIERÍA DE SOFTWARE.

Cambio de enfoque de la programación de pequeña escala hacia la programación a gran escala.

Evolución de lenguajes de programación de alto nivel.

2.1.3. ASPECTOS DE LA TECNOLOGÍA ORIENTADA A OBJETOS.

2.1.3.1. Breve Reseña Histórica.

La tecnología de objetos data de los años 60 cuando surge la necesidad de describir y simular fenómenos como sistemas de comunicación, redes neuronales, sistemas administrativos, etc.

En 1961 Krystin Nygaard con la idea de desarrollar un lenguaje de doble propósito (descripción de sistema y simulación programable), crea SIMULA I. Los usuarios descubrieron que también proveía de nuevas y poderosas facilidades cuando era usado para otros propósitos, aparte de la simulación, tales como el prototipo y otras aplicaciones.

En 1967 se creó SIMULA 67 y en él se implementaron por primera vez los conceptos de clase, objeto y herencia, que en adelante serían elementos centrales en los Lenguajes Orientado a Objetos. SIMULA 67 es una extensión del lenguaje ALGOL 60 y diseñado en 1967 por Ole Johnen Dahi y Krystin Nygaard, de la Universidad de Oslo y del Centro de Computación Noruego (Norsk Regasentra).

En 1968 se crea el lenguaje Simula 68 como contribución pragmática a la exploración del significado de la noción de tipos de datos. Transcurrió casi una década para que ese lenguaje escandinavo inspirara a los investigadores de Xerox Parc a inventar al lenguaje y ambiente de programación Smalltalk.

En 1970 se crea el SMALLTALK, éste fue el mayor desarrollo de los lenguajes Orientados a Objetos. El proyecto de investigación se realizó en la Corporación Xerox, Centro de Investigación de Palo Alto (PARC - Palo Alto Research Center) y fue dirigido por Allan Kay. Empezó en la década de los 70's y tuvo como meta, mas que el lenguaje de programación, una completa interfaz gráfica y una herramienta de desarrollo integrada.

Xerox PARC fue la pionera en el desarrollo y utilización de los componentes estándar de las modernas interfaces gráficas, como ventanas, iconos, mouse, etc.

SMALLTALK fue el primer Lenguaje Orientado a Objetos y fue el mas importante por las herramientas de desarrollo disponibles en su ambiente, éstas incluyen visualizadores de clases (class browsers) que permiten editar el código del programa e inspeccionadores de objetos (Objects inspectors).

Smalltalk encendió el interés académico por la orientación a objetos aprovechando un cierto agotamiento en los trabajos que buscaban un mecanismo matemáticamente más sólido basado en los llamados tipos abstractos de datos. Para los investigadores en semántica de tipos abstractos de datos, las nociones orientadas a objetos carecían de claridad conceptual; para los programadores que buscaban sistemas más flexibles, sobre todo para desarrollar interfaces gráficas más amigables, Smalltalk mostraba un potencial extraordinario si su velocidad de ejecución podía ser mejorado. Se suscitó una intensa competencia por casar los nuevos conceptos con C, el lenguaje imperativo más eficiente y barato para la programación de sistemas de software de infraestructura (systems programming en la jerga de la época, en contraposición a applications programming) de la época. Así nació C++ que pasó a ser un éxito tanto en el mundo académico como en el mundo comercial. La aparición de Java (1), que puede considerarse como una versión de C++ simplificada, depurada y orientada a las necesidades del mundo interconectado de internet en general y el WWW en particular, terminan de despejar las dudas; la programación orientada por objetos se establece firmemente como el paradigma de preferencia para los nuevos desarrollos en

interfaces gráficas, aplicaciones multimedia, paquetes de simulación, herramientas de análisis estadístico, manejadores de bases de datos, paquetes integrados y aplicaciones como comercio electrónico.

2.1.3.2. Fundamentos de la Tecnología Orientado a Objetos.

A) Objeto:

Es una entidad del mundo real que posee determinados atributos e identidad.

Objeto es un agregado de datos y operaciones que manipulan esos datos (objeto representante)

Un objeto es:

- Una cosa tangible y/o visible.
- Algo que puede ser entendido intelectualmente.
- Algo sobre lo que puede ser dirigido un pensamiento o una acción.

Un objeto representa un ítem, una unidad o una entidad individual e identificable, sea real o abstracta, con un papel bien definido en el dominio del problema (3).

Un objeto tiene estado, comportamiento e identidad; la estructura y el comportamiento de objetos similares son definidos en sus clases; los términos instancia y objeto son intercambiables.

El estado de un objeto incluye todas las propiedades del objeto (usualmente estáticas), más los valores actuales de cada una de estas propiedades (usualmente dinámicos).

El comportamiento es la forma como un objeto actúa y reacciona, en términos de sus cambios de estado y del paso de mensajes.

El comportamiento de un objeto representa su actividad visible y comprobable externamente. La actividad puede ser realizada por un objeto sobre otro o sobre él mismo.

Los objetos del mundo real son las únicas clases de objetos que son de interés durante la especificación de requerimientos del software.

Otras importantes clases de objetos son invenciones del proceso de desarrollo de software, cuyos relacionamientos con otros objetos sirven como mecanismos para obtener comportamientos de mayor complejidad.

B) Identidad

Es independiente de las propiedades del objeto, lo cual implica independencia de valor y de estructura.

No cambia durante toda la vida del objeto. Además, un oid no se reutiliza aunque el objeto deje de existir.

No se tiene ningún control sobre los oids y su manipulación resulta transparente.

Sin embargo, es preciso contar con algún medio para hacer referencia a un objeto utilizando referencias del dominio (valores de atributos).

C) Comportamiento

Los mensajes navegan por los enlaces, a priori en ambas direcciones.

Estado y comportamiento están relacionados.

Ejemplo: no es posible aterrizar un avión si no está volando. Está volando como consecuencia de haber despegado del suelo.

D) Comunicación

Un sistema informático puede verse como un conjunto de objetos autónomos y concurrentes que trabajan de manera coordinada en la consecución de un fin específico.

El comportamiento global se basa pues en la comunicación entre los objetos que la componen.

E) Clasificación

El concepto de clasificación permite agrupar objetos con la misma estructura de datos o atributos y el mismo comportamiento u operaciones en clases.

Una clase es un conjunto de objetos que comparten una estructura común y un comportamiento común.

F) Encapsulamiento.

El encapsulamiento (llamado también data hiding) consiste en la separación de los aspectos externos de un objeto, accesibles por otros objetos, de los detalles internos de la implementación de aquel objeto, que quedan ocultos de los demás.

Significa también que los datos y el código que los manipula son definidos juntos, y que los datos no pueden ser separados de o accedidos separadamente del código asociado.

La encapsulación esconde los detalles de la implementación de un objeto.

Polimorfismo es cuando objetos de clases distintas, pero con una superclase común, pueden responder de manera diferente a un mismo mensaje.

Al proceso de colocar en un compartimento los elementos de una abstracción que constituyen su estructura y comportamiento se le llama encapsulación.

Aquellas porciones de un programa que no son parte de una dada definición no pueden acceder cualquier dato dentro de la definición.

El dato se considera así encapsulado junto con el código.

La encapsulación se logra a través del ocultamiento de información, que consiste en esconder todos los secretos de un objeto que no contribuyen a las características esenciales de éste.

La encapsulación sirve para separar la interfaz (parte pública) de una abstracción de su implementación (parte privada).

G) Herencia

Heredar significa compartir los atributos y las operaciones entre las clases con base en un relacionamiento jerárquico.

Una clase puede ser definida en forma amplia y después ser refinada en sucesivas subclases.

Cada subclase incorpora o hereda, todas las propiedades de su superclase e incrementa sus propias y exclusivas características.

Las propiedades de la superclase no necesitan ser repetidas en cada subclase.

Permite factorar las propiedades y operaciones comunes a las diferentes clases y colocarlas dentro de una superclase común, para poderlas utilizar después sin necesidad de volverlas a crear.

Una clase comparte la estructura y comportamiento definidos en una (herencia simple) o más (herencia múltiple) clases.

La clase que recibe la herencia se le llama sub-clase o clase derivada.

La clase de la cual se hereda se le llama super-clase o clase base.

La herencia sirve para definir una jerarquía de tipo es un entre clases en donde una clase derivada hereda de una o más clases base generalizadas.

Típicamente, una clase derivada especializa su(s) clase(s) base incrementando o sobreponiendo el comportamiento y estructura existente.

H) Abstracción

Abstracción (en orientación a objetos) consiste en la concentración en los aspectos esenciales, propios, de una entidad, ignorando sus propiedades accidentales.

La abstracción se enfoca en las características esenciales de algún objeto, relativo a la perspectiva del espectador.

Significa concentrarse en lo que un objeto es y hace, antes de decidir cómo él debe ser implementado.

El uso de la abstracción durante el análisis, significa lidiar apenas con los conceptos del dominio de la aplicación, y no tener que tomar decisiones sobre el diseño y la implementación, antes del problema ser comprendido.

La abstracción denota las características esenciales de un objeto, que lo distinguen de las otras clases de objetos, permitiendo reconocer claramente sus fronteras conceptuales, con relación a la perspectiva del observador.

I) Polimorfismo

Polimorfismo es cuando objetos de clases distintas, pero con una superclase común, pueden responder de manera diferente a un mismo mensaje.

Es la característica que permite en la OO tener operaciones con el mismo nombre asociadas a diferentes objetos (clases), pero actuando de forma diferente.

Polimorfismo significa también que el remitente de un mensaje a otro objeto (al hacerse un paso de mensaje), no necesita conocer la instancia (objeto) de la clase receptora. El objeto receptor puede pertenecer a una clase arbitraria.

El receptor del estímulo es quien determina como un estímulo deberá ser interpretado, no el remitente. El remitente solamente necesita saber que otra instancia puede desempeñar un determinado comportamiento y no a qué clase la instancia pertenece y, por lo tanto, qué operación realmente efectúa la acción.

J) Concurrencia

La concurrencia permite a distintos objetos actuar al mismo tiempo.

Significa que en el dominio de la aplicación (sistema referente), los objetos pueden actuar de forma independiente y al mismo tiempo, es decir, que puede haber concurrencia.

La concurrencia de objetos reales es un hecho que existe, y su representación en software es realizada de forma adecuada usando la técnica de orientación a objetos.

K) Persistencia.

La persistencia almacena el estado de un objeto a través del tiempo y/o espacio.

Significa que un objeto creado y operado durante el tiempo de ejecución de un programa, puede continuar existiendo inclusive después de que el programa que lo creó dejó de operar.

La persistencia de un objeto en OO está relacionada con aplicaciones de bases de datos, donde los objetos creados son almacenados en la base durante largos períodos de tiempo.

La persistencia de los objetos designa la capacidad de un objeto trascender en el espacio/tiempo.

Un objeto persistente conserva su estado en un sistema de almacenamiento permanente

Podremos después reconstruirlo (materialización del objeto).

Por defecto no los consideramos persistentes

L) Simulación.

La simulación orientada a objetos se refiere a la construcción de modelos de simulación usando herramientas de diseño y programación orientadas a objetos; donde el diseño se refiere específicamente a la construcción del modelo del sistema usando entidades que son naturales en el sistema real; y, la programación se refiere al uso de las herramientas (librerías de clases) y estructuras de datos, proporcionados por algún lenguaje de programación orientado a objetos, que permita implementar el modelo de simulación.

M) Reutilización.

La reutilización no depende esencialmente de aspectos tecnológicos sino también de los organizativos y otros. No obstante la orientación a objetos ofrece un conjunto de particularidades que facilitan el desarrollo de la reusabilidad, en tres niveles de reutilización desde el punto de vista arquitectónico que pueden ser llevados a cabo teniendo en cuenta los principios de la orientación a objetos. Estos niveles son las bibliotecas de clases, los patrones de diseño y las arquitecturas.

2.1.3.3. Ventajas del Paradigma Orientado a Objetos

Gestionar la Complejidad.

Ofrecer mecanismos de abstracción permitiendo modelizar un sistema a diferentes niveles y utilizar el mismo formalismo en diferentes etapas del ciclo de vida del sistema (4)

Facilitar la reutilización de modelos ya existentes.

Producir modelos fáciles de leer y validar autorizando la descripción de interfaces de alta nivel entre componentes.

2.1.3.4. Desventajas del Paradigma Orientado a Objetos

La curva de aprendizaje requiere educación y entrenamiento constante.

La tecnología esta en continua evolución.

La gente necesita tiempo para cambiar de paradigma.

La gente no desea cambiar porque todavía cree en el paradigma estructurado.

2.1.3.5. Tecnología Orientado a Objetos en el Futuro

- Estructuración de sistemas de una manera eficiente.
- **Fácil de gestionar.**
 - Para prueba, para adaptaciones, para reuso.
 - Componentes poco acoplados y altamente coherentes.
- **Fácil de comprender.**
 - Datos y acciones juntas, protección de la información.

- Correspondencia inmediata con los objetos del mundo real.

- **Sin embargo.**

- La estructura de un sistema esta distribuido entre los objetos.

- La concurrencia de un objeto es distribuida en objetos más pequeños.

2.1.3.4. PROCESO DE DESARROLLO DE SOFTWARE

QUE ES UN PROCESO DE DESARROLLO DE SOFTWARE.

Describe un conjunto de actividades que deben realizarse en un determinado orden (5).

Define Que hacer, como hacerlo, Cuando hacerlo y el Motivo por el cual debe ser hecho.

Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

Debe ser:

Reproducible

Definido

Medible en cuanto a rendimiento

Optimizable.

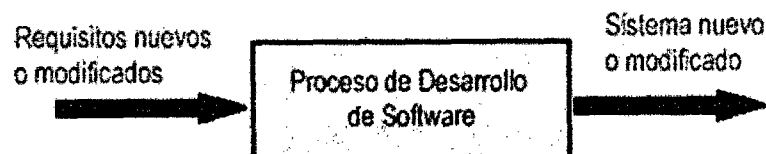


Ilustración 1: Proceso de desarrollo de software

No existe un proceso de software universal. Las Características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

2.2. METODOLOGÍA

Una metodología es un proceso para producir software de forma organizada, empleando una colección de técnicas y convenciones de notación predefinidas.

(James Rumbaugh.)

Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software

(Mario Piattini.)

El desarrollo de un sistema se puede explicar también como:

Una secuencia de modelados que ayuda a construir, a partir de la realidad, uno o varios modelos, derivados unos de otros, con el objetivo de lograr un modelo final o sistema.

Y entonces:

Un método es una guía que define las reglas de paso de un modelo a otro para evolucionar progresivamente hasta el modelo final.

Para explicar mejor El Proceso Unificado (5) observemos las diferencias importantes entre un método y un lenguaje de modelaje. Un Método es una manera explícita de estructurar nuestro pensamiento y acciones. Un método le dice a un usuario ¿qué hacer?, ¿cómo hacerlo?, ¿cuándo hacerlo?, y ¿por qué fue hecho? (el propósito de una actividad específica). Los métodos contienen modelos, y estos modelos son utilizados para describir algo y comunicar los resultados del uso de un método. La principal diferencia entre un método y un lenguaje de modelaje es que el lenguaje de modelaje carece de un proceso o de las instrucciones para ¿qué hacer?, ¿cómo hacerlo?, ¿cuándo hacerlo?, y ¿por qué fue hecho?

Cuando construimos modelos, también estructuramos nuestros pensamientos. Un modelo siempre se refiere a algo y tiene un propósito. Si un modelo no tiene un propósito específico, causará problemas, porque nadie sabrá ¿cómo?, ni ¿por qué usarlo?. Un modelo es expresado en un *Lenguaje de Modelaje*.

Cabe mencionar que ésta no es la única metodología existente en el mercado, pero para el desarrollo del presente proyecto he creído conveniente tomar como base "El Proceso de Desarrollo Unificado" (Rational Unified Process) por ser una de las más difundidas y aceptadas, teniendo en cuenta además que cada desarrollador es libre de plantear su propia metodología.

A) EL PROCESO UNIFICADO

Un proyecto de software exitoso satisface o excede las expectativas del cliente, es desarrollado en una forma económica y puntual, y es resistente al

cambio y la adaptación. El ciclo de desarrollo debe promover la creatividad e innovación. Al mismo tiempo, el proceso de desarrollo debe ser controlado y medido para asegurar que el proyecto sea bien completado. Un ciclo de vida iterativo e incremental bien manejado proporciona el control necesario sin afectar la creatividad (5).

B) CARACTERÍSTICAS DEL PROCESO UNIFICADO

El proceso Unificado es un proceso de desarrollo de software que es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto (5).

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

El Proceso Unificado utiliza el UML para preparar todos los esquemas de un sistema de software. No obstante los verdaderos aspectos definitorios del proceso unificado se resumen en tres frases clave: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

a) Desarrollo basado en componentes.

Un componente de software puede ser definido como una pieza de software no trivial, un módulo, paquete o subsistema, que llena una función clara, tiene una frontera clara y puede ser integrado en una arquitectura bien definida.

b) El Proceso Unificado está dirigido por Casos de Uso.

En resumen, **dirigido por casos de uso** quiere decir que el proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan, y los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba.

c) El Proceso Unificado está centrado en la arquitectura

El Proceso Unificado proporciona una forma metódica y de sistema para diseñar, desarrollar y validar una arquitectura. Ofrece plantillas para descripciones de arquitectura alrededor de conceptos de múltiples vistas de arquitectura, y la captura de estilo de arquitectura, reglas de diseño y restricciones.

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como la perciben los usuarios y los inversores, y se refleja en los casos de uso. Esta arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción.

El componente del proceso de diseño contiene actividades específicas dirigidas a identificar restricciones de arquitectura y elementos significativos, así como guías sobre como tomar decisiones de arquitectura. El proceso de administración muestra como la planeación de iteraciones tempranas toma en cuenta el diseño de una arquitectura y la resolución de los riesgos técnicos mayores.

d) El Proceso Unificado es iterativo e incremental

Se afirma esto en el sentido de que el software no es liberado de una sola vez al final proyecto, sino que es desarrollado por partes, en cada una de las cuales se construye software de calidad de producción, probado e integrado que satisface los requerimientos del proyecto.

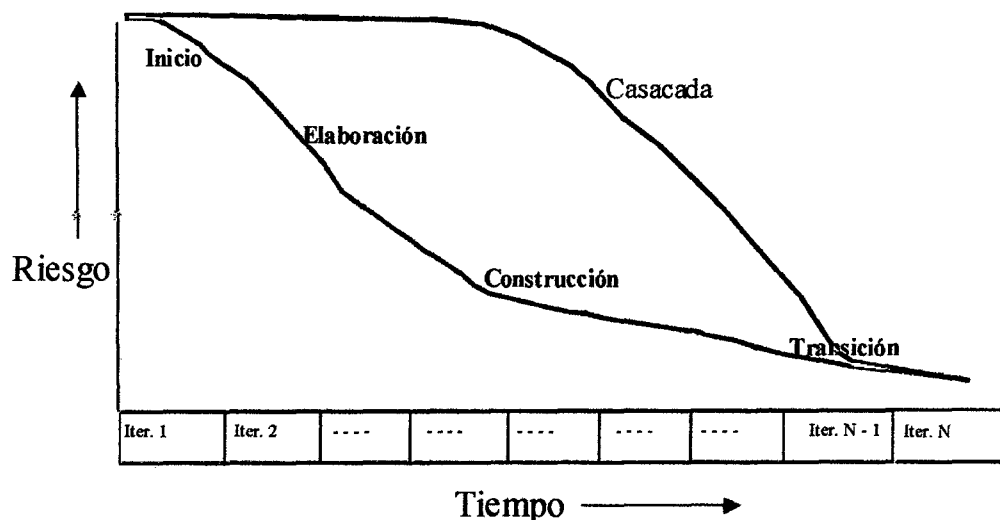


Ilustración 2: Comparación del tiempo de desarrollo con el riesgo

C) LAS MEJORES PRÁCTICAS DE LA INGENIERÍA DE SOFTWARE

El Proceso Unificado es el producto final de 3 décadas de desarrollo y por lo tanto agrupa 6 de las mejores prácticas de la ingeniería de software, que pasaremos a describir brevemente:

a) Desarrolle iterativamente

Las primeras iteraciones tienen por finalidad reducir los principales riesgos, produciendo cada una, una versión ejecutable, un incremento adicional al sistema. Cada iteración incluye integración y prueba.

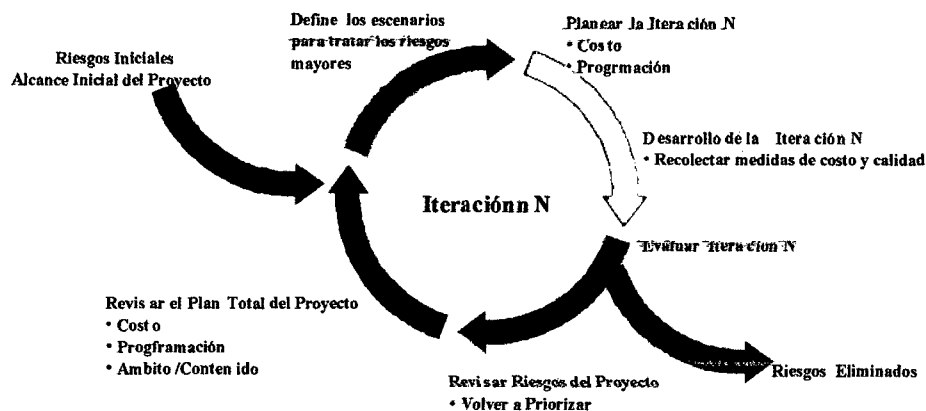


Ilustración 3: Ciclo de vida del desarrollo iterativo

b) Administre requerimientos

Los dos elementos clave detrás de un proceso iterativo controlado son la administración de requerimientos y el control del cambio. La administración de requerimientos es una aproximación de sistema para obtener, organizar, comunicar y administrar los requerimientos cambiantes de un sistema intensivo de software.

c) Use arquitectura de componentes

Las buenas arquitecturas cumplen sus requerimientos, son adaptables al cambio y basadas en componentes.

d) Modele visualmente

Muchos proyectos hoy utilizan lenguajes de programación orientados a objetos para obtener sistemas estables, reutilizables y tolerantes al cambio. Para obtener estos beneficios, es aún más importante usar tecnología orientada a objetos en el diseño.

e) Verifique calidad

Esta práctica permite medir la calidad en forma oportuna y frecuente ya que es un proceso iterativo. Resolver los problemas antes de que el software sea liberado permite un gran ahorro tanto económico como de trabajo.

f) Controle cambios

Permite administrar el trabajo de múltiples desarrolladores, equipos de trabajo, localizaciones, iteraciones, versiones. El Proceso Unificado es incremental solo si se controlan los cambios efectuados.

D) ELEMENTOS PRINCIPALES DEL MODELADO

El RUP es representado con el uso de los cuatro principales elementos de modelado.

Workers (actores, trabajadores): El quién

Activities (actividades): El cómo

Artifacts (artefactos): El qué

Workflows (flujos de trabajo): El cuándo

E) DESCRIPCIÓN DEL PROCESO

Como ya se explicó anteriormente este proceso es un proceso de desarrollo iterativo e incremental, en el sentido de que el software no es liberado de una sola vez al final proyecto, sino que es desarrollado por partes.

Las primeras fases son inicio y elaboración. Durante el inicio, se establece la razón del negocio para el proyecto y se decide el alcance del proyecto. Aquí es donde se obtiene el compromiso del patrocinador del proyecto para seguir adelante. En la elaboración, se obtienen requerimientos más detallados, se hace análisis y diseño de alto nivel para establecer una arquitectura base, y se crea el plan para la construcción.

De manera particular mostramos iteraciones solamente en la fase de construcción. En realidad, puede tener iteraciones en todas las fases, y es a menudo una buena idea hacerlo en una fase larga. Sin embargo, la construcción es la fase clave sobre la cual iterar.

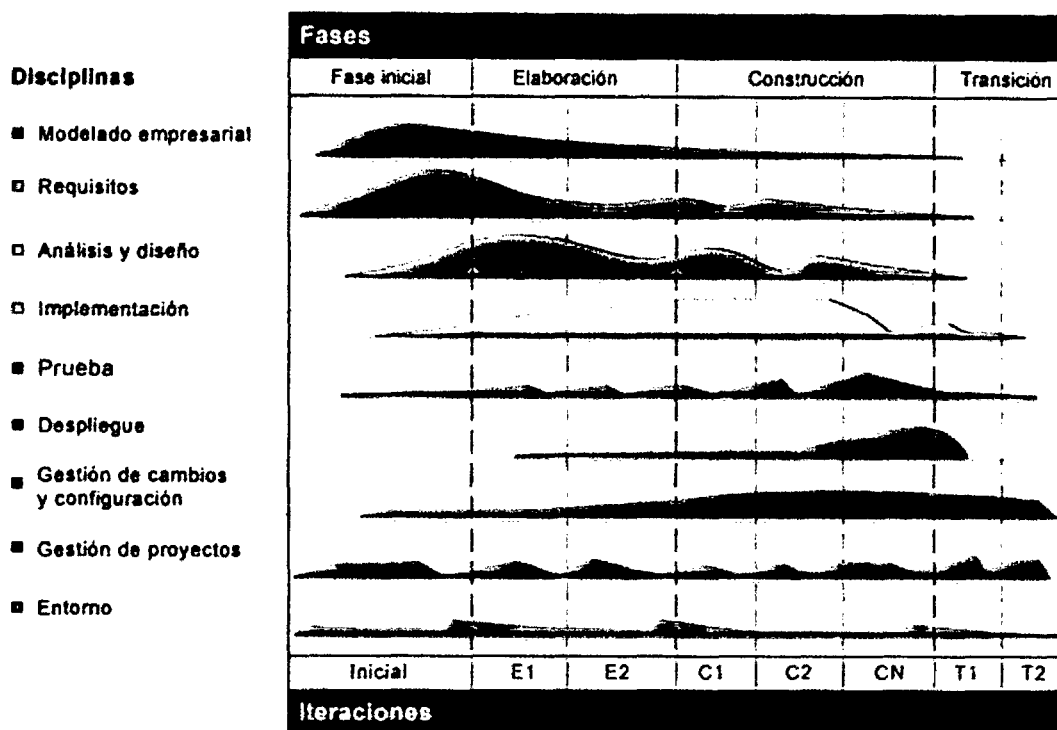


Ilustración 4: Dimensiones del Proceso Unificado

El Proceso Unificado puede ser descrito en dos dimensiones:

- A lo largo del *tiempo*, los aspectos del ciclo de vida del proceso y cómo se desarrolla.
- A lo largo de los *componentes del proceso*, que agrupa las actividades lógicamente por naturaleza.

La primera dimensión representa el aspecto *dinámico* del proceso, como es promulgado, y como es expresado en término de ciclos, fases, iteraciones e hitos.

La segunda dimensión representa el aspecto *estático* del proceso: como es descrito en términos de los componentes del proceso, actividades, flujo de trabajo, artificios y trabajadores.

F) FLUJOS DE TRABAJO FUNDAMENTALES

a) Captura de Requerimientos

Llamamos captura de requisitos al proceso de averiguar normalmente en circunstancias difíciles lo que se debe construir.

La meta de la captura de requerimientos es describir qué es lo que el sistema debería hacer y permite a los desarrolladores y al cliente ponerse

de acuerdo en esa descripción. Para alcanzar esto, se debe delimitar el sistema, definir sus alrededores y el comportamiento que debe presentar. Los clientes y usuarios potenciales son fuentes importantes de información así como cualquier otro requerimiento del sistema que pueda existir.

La captura de requerimientos resulta en un modelo de casos de uso y algunos requerimientos suplementarios. El modelo de caso de uso es esencial para el cliente, quien necesita el modelo para validar que el sistema se transformará en lo que esperaba, y para los desarrolladores, quienes necesitan el modelo para obtener un mejor entendimiento de los requerimientos del sistema.

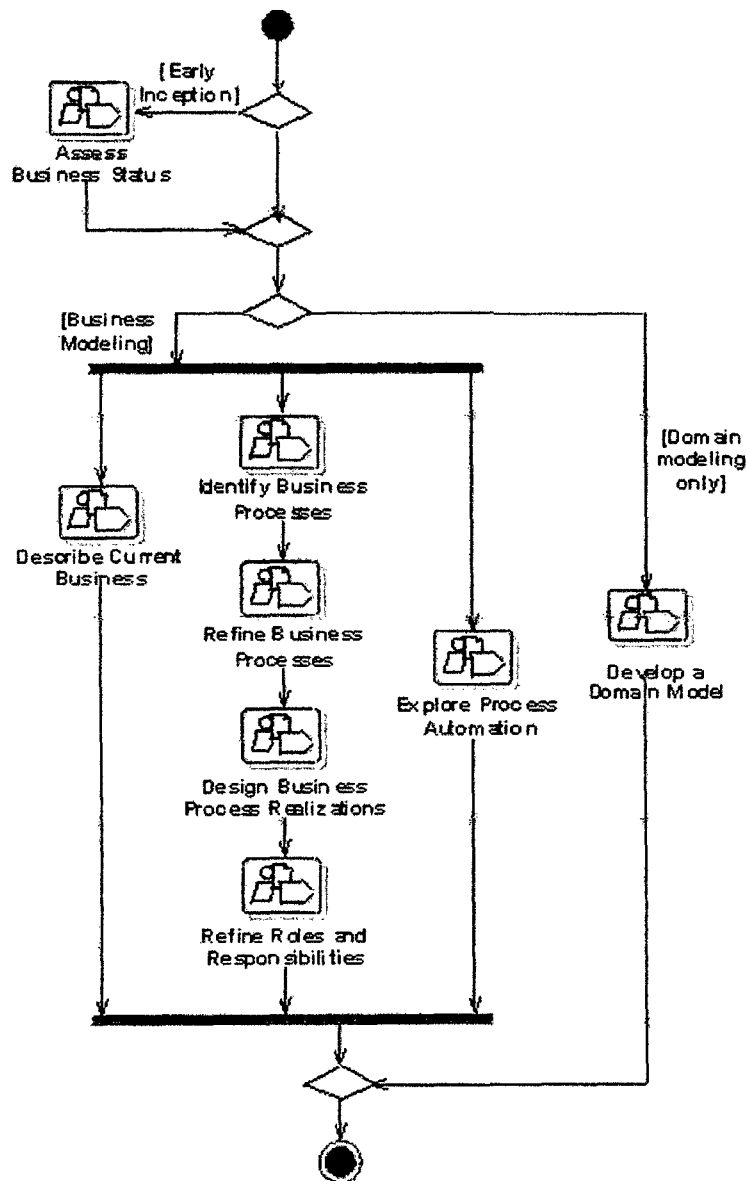


Ilustración 5: Flujo de trabajo modelado del negocio

b) Análisis

Aquí analizamos los requisitos refinándolos y estructurándolos para conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y nos ayude a estructurar el sistema entero.

La fase de análisis trata con las abstracciones primarias (clases y objetos) y los mecanismos presentes en el dominio del problema. Las clases que modelan estos son identificadas, con las relaciones de las unas con las otras, y son descritas en el diagrama de clases del UML. La colaboración entre las clases para realizar los casos de uso también es descrita mediante cualquiera de los modelos dinámicos en el UML. En el análisis, solamente las clases que están en el dominio del problema (conceptos del mundo real) son modeladas, no las clases técnicas que definen detalles y soluciones en el sistema de software, tales como clases para interfaces de usuarios, bases de datos, comunicación, concurrencia, y así sucesivamente.

c) Diseño

En el diseño modelamos el sistema y encontramos su forma para que soporte todos los requisitos que se le supone. Una entrada esencial en el diseño es el resultado del análisis, quien proporciona una descripción detallada de los requisitos y lo que es más importante, impone una estructura del sistema que debemos esforzarnos por conservar cuando demos forma al sistema.

El diseño resulta en un modelo de diseño que sirve como una abstracción del código fuente; esto es, el modelo de diseño actúa como un plano de cómo el código fuente está estructurado y escrito. El diseño también resulta en unas descripciones internas de los casos de uso, o realizaciones de los casos de uso, que describen como los casos de uso son realizados en términos de las clases/objetos participantes.

El modelo de diseño consiste de clases de diseño estructuradas en paquetes de diseño; también contiene descripciones de como los objetos de estas clases de diseño colaboran para ejecutar los casos de uso.

d) Implementación

Aquí empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Los propósitos de la implementación son:

- Planificar las integraciones de sistema necesarias en cada iteración
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño.
- Probar los componentes individualmente, y a continuación integrarlos y compilarlos para luego ser enlazados en uno o en más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones del sistema.

En la implementación las clases de la fase de diseño son convertidas a código actual en un lenguaje de programación orientado a objetos (la utilización de un lenguaje procedural no es recomendada).

e) Prueba

En el flujo de trabajo de la prueba verificamos el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros.

G) FASES

Estas conforman la organización dinámica del proceso a lo largo del tiempo. El ciclo de vida del software está dividido en ciclos, cada ciclo trabaja en una nueva generación del producto. El Proceso Unificado divide un ciclo de desarrollo en cuatro *fases* consecutivas:

- Fase de Inicio (Inception Phase).
- Fase de Elaboración (Elaboration Phase).
- Fase de Construcción (Construction Phase).
- Fase de Transición (Transition Phase).

Cada fase es construida con hitos (un punto en el tiempo en el cual ciertas decisiones críticas deben ser tomadas), bien definidos, y por lo tanto metas claves han sido alcanzadas. Cada fase tiene un propósito específico.

a) Fase de Inicio

Durante la fase de inicio se trabaja en el caso del negocio para el proyecto (cuánto costará y cuánto traerá de beneficio) y también se establecen los casos de uso del negocio y se delimita el alcance del proyecto. Para lograr esto debe identificar todas las entidades externas con las que el sistema interactúa (actores) y definir la naturaleza de esta interacción a un nivel alto. Esto involucra identificar todos los casos de uso y describir unos cuantos significantes.

b) Fase de elaboración

En este punto desea obtener un mejor entendimiento del problema.

- ¿Qué es lo que realmente se va a construir?
- ¿Cómo se va a construir?
- ¿Qué tecnología se va a usar?

En las decisiones de qué situaciones deben ser vistas durante esta etapa, se debe guiar primero y sobre todo por los riesgos del proyecto. ¿Cuáles son las cosas que podrían salir mal? Mientras mayor el riesgo, debe ser mayor la atención que se le debe prestar.

c) Fase de Construcción

El propósito primordial de esta fase es dejar listo un producto de software en versión su versión operativa inicial, a veces llamada "versión beta". El producto debería tener la calidad adecuada para su aplicación y asegurarse de cumplir los requisitos. La construcción debería tener lugar dentro de los límites del plan de negocio.

Durante la fase de construcción, iterativamente e incrementalmente se desarrolla un producto completo que está listo para pasar a su comunidad de usuarios. Cada iteración es un mini-proyecto. Hace análisis, diseño, codificación, pruebas, e integración para los casos de uso asignados a cada iteración.

Finaliza la iteración con una demostración al usuario y se realizan pruebas del sistema para confirmar que los casos de uso han sido construidos correctamente.

d) Fase de Transición

Durante la fase de transición se hace la transición del software a la comunidad de usuarios. Una vez que el producto ha sido puesto en las manos de los usuarios, a menudo surgen problemas que requieren desarrollo adicional para ajustar el sistema, corregir errores no detectados, o finalizar algunas de las características que hallan sido pospuestas. Esta fase típicamente inicia con una versión "beta" de los sistemas.

El proyecto usualmente recibirá información de los usuarios para:

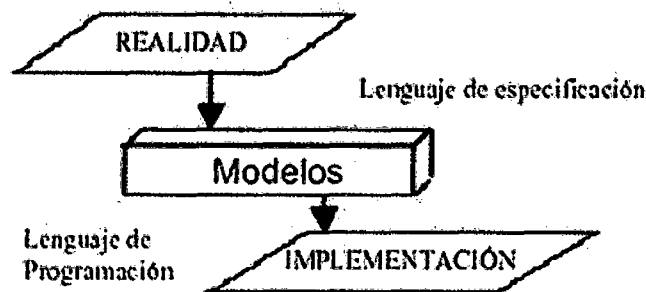
- Determinar si el sistema hace lo que demandan sus usuarios y el negocio.
- Descubrir riesgos inesperados.
- Anotar problemas no resueltos y encontrar fallos.
- Eliminar ambigüedades y lagunas en la documentación del usuario.
- Centrarse en áreas en las que los usuarios muestren deficiencias y necesiten información o formación.

2.3. MODELADO

El modelaje y la programación están altamente integrados. Los sistemas se han vuelto más grandes y distribuidos a través de muchas computadoras a través de arquitecturas Cliente/Servidor (con Internet como la última arquitectura Cliente/Servidor). La necesidad de integrar sistemas complejos en un ambiente distribuido requiere que los sistemas tengan algunos modelos comunes. Construir modelos de los sistemas antes de implementarlos se volverá tan normal y aceptado en la comunidad de ingeniería del software como lo son en las otras disciplinas de la ingeniería (4).

Un modelo es básicamente una abstracción que incluye lo esencial de un problema complejo o estructura, filtrando los detalles no esenciales, de forma que el problema se hace más comprensible. No se trata, por tanto, de una representación alternativa del sistema. Idealmente, una "representación" de un sistema debería mantener toda la información de la entidad que representa; por

contra, una abstracción deliberadamente simplifica, quedándose con las características más sobresalientes.



- Los modelos son representaciones semánticas simplificadas de un sistema para analizarlo y comprenderlo a fin de diseñarlo mejor.

Ilustración 6: Proceso de Modelado.

2.4. UML

A) INTRODUCCIÓN AL UML

UML es un lenguaje para especificar, visualizar, construir y documentar los componentes de los sistemas (software, modelos de negocio y otros sistemas diferentes al software) (5).

Es importante incidir en el hecho de que UML se trata de un lenguaje de modelado, no un método de Análisis y Diseño Orientado a Objetos (ADOO). Un método está compuesto de un lenguaje de modelado y de un proceso. El lenguaje de modelado aporta los elementos de modelado, es decir, los conceptos y la semántica, la notación (normalmente gráfica) y una serie de recomendaciones sobre cómo usarlo. Por otro lado, la parte de proceso de un método de ADOO indica los pasos a seguir para realizar un diseño.

El UML es un Lenguaje. Un lenguaje proporciona un vocabulario y las reglas para combinar las palabras de ese vocabulario para la comunicación.

B) OBJETIVOS DE UML

- Ofrecer a los usuarios un lenguaje visual de modelado expresivo y listo para ser utilizado: de esta forma, los usuarios pueden desarrollar e intercambiar modelos con significado.

- Ofrecer mecanismos de extensión y especialización para ampliar el conjunto de conceptos clave: UML puede ser extendido para cubrir nuevas necesidades o dominios específicos.
- Ser independiente de los lenguajes de programación y los procesos de desarrollo.
- Contribuir al crecimiento del mercado de herramientas: La herramienta que soporta el lenguaje de modelado más utilizado y considerado como estándar sale beneficiada al tener más cuota de mercado.
- Da soporte a conceptos de alto nivel de desarrollo: Tales como patterns, frameworks o componentes. La inclusión de estos conceptos beneficia a la orientación al objeto y, en especial, a la reutilización del software.

C) BREVE RECORRIDO POR EL UML

UML puede ser utilizado para el modelaje del negocio, software en todas las fases de desarrollo y para todos los tipos de sistemas, y modelaje en general de cualquier construcción que tenga una estructura estática y un comportamiento dinámico. Ahora daremos un vistazo a las diferentes partes del UML:

a) Vistas

Una vista no es un gráfico, pero es una abstracción que consiste en una serie de diagramas.

b) Diagramas

Son los gráficos que describen los contenidos en una vista. El UML tiene nueve tipos diferentes de diagramas que son utilizados en combinación para proporcionar todas las vistas del sistema.

| Área | Vista | Diagramas | Conceptos Principales |
|-------------|-------------------------|--------------------------|---|
| Estructural | Vista estática | Diagrama de clases | Clase, asociación, generalización, dependencia, realización, interfaz |
| | Vista de casos de uso | Diagrama de casos de uso | Caso de uso, actor, asociación, extensión, inclusión, generalización de casos de uso. |
| | Vista de implementación | Diagrama de componentes | Componente, interfaz, dependencia, realización |

| | | | |
|-------------------|-----------------------------|--------------------------|---|
| | Vista de despliegue | Diagrama de despliegue | Nodo, componente, dependencia, localización |
| Dinámica | Vista de máquina de estados | Diagrama de estados | Estado, evento, transición, acción |
| | Vista de actividad | Diagrama de actividad | Estado, actividad, transición de terminación, división, unión |
| | Vista de interacción | Diagrama de secuencia | Interacción, objeto, mensaje, activación |
| | | Diagrama de colaboración | Colaboración, interacción, rol de colaboración, mensaje |
| | Vista de Gestión del modelo | Diagrama de clases | Paquete, subsistema, modelo |
| Extensión del UML | Todas | Todos | Restricción, estereotipo, valores etiquetados |

Tabla 1: Vistas y diagramas del UML

A continuación mostramos cuáles son los diagramas del UML y cuál es su estructura:

- ❖ Diagrama de Casos de Uso
- ❖ Diagrama de Estructura Estática
 - ✓ Diagramas de Clase
 - ✓ Diagrama de Objetos
- ❖ Diagramas de Comportamiento
 - ✓ Diagrama de Estados
 - ✓ Diagrama de Actividad
 - ✓ Diagramas de Interacción
 - Diagrama de Secuencia
 - Diagrama de Colaboración
- ❖ Diagramas de implementación
 - ✓ Diagrama de Componentes
 - ✓ Diagrama de Despliegue

D) CONCEPTOS PREVIOS

Estos conceptos definen características que permiten al UML organizar y extender sus diagramas

a) Paquetes

En algunas ocasiones se encontrará con la necesidad de organizar los elementos en un grupo. Se representa por una carpeta tabular como se muestra en la figura:

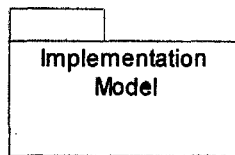


Ilustración 7: Representación UML de un paquete

b) Notas

La nota es un rectángulo con una esquina doblada, y dentro del rectángulo se coloca la explicación. La nota se adjunta al elemento conectándolos mediante una línea discontinua.



Ilustración 8: Representación de una nota

2.5. DIAGRAMAS DE CLASE

Un diagrama de clases describe la vista estática del sistema. Aunque tiene similitudes con un modelo de datos (entidad-relación), recuerde que las clases no solo muestran la estructura de la información, sino que describen también el comportamiento. Un propósito de los diagramas de clases es definir una base para otros diagramas donde otros aspectos del sistema son mostrados (tales como los estados de los objetos o la colaboración entre ellos mostrados en los diagramas dinámicos).

A) CLASES

Podemos decir que una clase es una descripción de un grupo de objetos con propiedades (atributos), comportamiento (operaciones), relaciones a otros objetos, y semántica comunes. Por lo tanto, una clase es una plantilla para crear objetos.

◆ Identificación de las Clases

Clases de entidad. Las clases de entidad típicamente son clases que son necesarias por el sistema para realizar alguna responsabilidad. Las clases de entidad típicamente son encontradas temprano en la etapa de elaboración. Son llamadas a menudo clases de dominio porque usualmente se refieren a abstracciones de entidades del mundo real.

Clases de frontera. Las clases de frontera manejan la comunicación entre el entorno del sistema y el interior del mismo. Pueden proporcionar la interfaz a un usuario u otro sistema (la interfaz a un actor). Constituyen la parte dependiente del entorno. Las clases de frontera son usadas para modelar las interfaces del sistema.

Clases de control. Las clases de control coordinan los eventos necesarios para realizar el comportamiento especificado en el caso de uso. Puede pensar de una clase de control como la que "corre" o "ejecuta" el caso de uso. Las clases de control típicamente son clases dependientes de la aplicación.

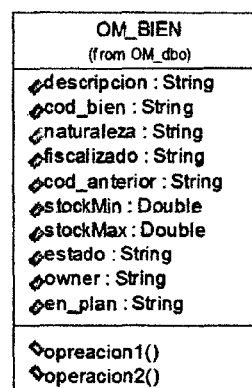


Ilustración 9: Representación UML de una clase

B) RELACIONES

Una vez que ya se han identificado las clases y han sido plasmadas en un diagrama, es necesario que estas muestren la forma como se relacionan o comunican, para ello hacemos uso de las relaciones. Si no añadimos estas relaciones lo único que habremos descubierto es tan sólo una lista de elementos, en lugar de una representación de un área de conocimiento

a) Asociaciones

Cuando las clases se conectan entre si en forma conceptual, esta conexión se conoce como asociación.

Una asociación entre clases significa que hay un enlace entre objetos de las clases asociadas. El número de objetos conectados depende de la multiplicidad de la asociación, la cual es discutida posteriormente. En el UML, las relaciones de asociación son mostradas como líneas sólidas que conectan las clases asociadas, como se muestra a continuación.

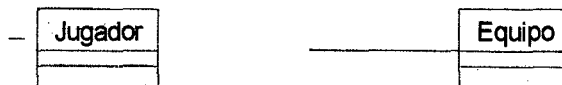


Ilustración 10: representación de una asociación entre dos clases

Los nombres son añadidos si son necesarios para agregar claridad al modelo.

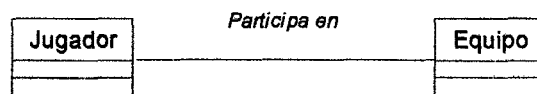


Ilustración 11: Nombrando a una asociación

La asociación podría funcionar en dirección inversa, y pueden aparecer las dos asociaciones entre clases en el mismo diagrama.

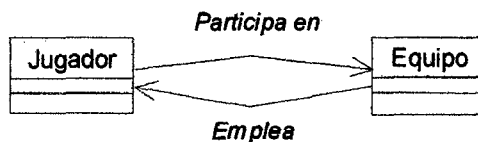


Ilustración 12: Cómo representar dos asociaciones entre dos clases

Cuando una clase se asocia con otra, cada una de ellas juega un papel dentro de la asociación.

El extremo de una asociación donde se conecta a una clase se llama el rol de la asociación. Los nombres de roles pueden ser usados en vez de los nombres. Un nombre de rol puede ser puesto en uno o ambos extremos de la línea de asociación. No es necesario tener ambos un nombre de rol y un nombre de asociación.

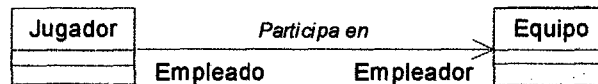


Figura 1: Representación de los roles

b) Clases de Asociaciones

La clase en asociación no está conectada a ninguno de los extremos de la asociación, sino a la asociación misma. La clase en asociación es una clase normal; puede tener atributos, operaciones y otras asociaciones. Cada enlace de la asociación está relacionado con la clase en asociación.

En otras palabras podríamos decir que se puede concebir a una clase de asociación de la misma forma que lo haría con una clase estándar, y utilizará una línea discontinua para conectarla a la línea de asociación. Así también, ésta puede tener asociaciones con otras clases.

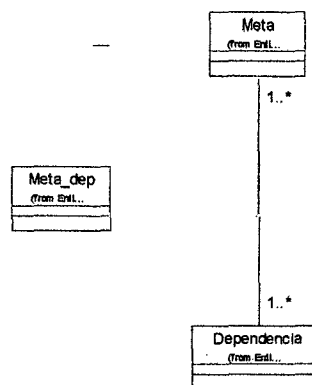


Ilustración 13: Clase de asociación

c) Asociaciones reflexivas

En ocasiones múltiples objetos pertenecientes a la misma clase pueden tener que comunicarse entre sí. Esto es mostrado en el diagrama de clase como una asociación o agregación reflexiva.

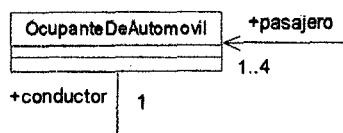


Ilustración 14: Asociación reflexiva

d) Agregaciones

Una agregación es una forma especializada de asociación, en la cual un todo está relacionado a sus partes. La agregación es conocida como una relación "parte de" o de contenimiento. La notación del UML para una relación de agregación es una asociación con un diamante cerca de la clase que denota el agregado (todo).

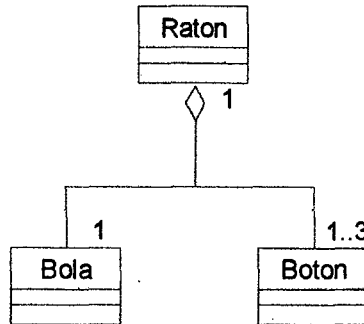


Ilustración 15: Representación de una agregación

e) Composiciones

Una composición es un tipo muy representativo de una agregación en donde cada componente puede pertenecer tan solo a un todo. El contenimiento por valor llamado composición, indica que las partes viven dentro del todo y por lo tanto serán destruidas con el todo. La composición implica una restricción sobre el valor de la multiplicidad en el lado del agregado: sólo puede tomar los valores 0 o 1 y se representa en los diagramas por un rombo de color negro.

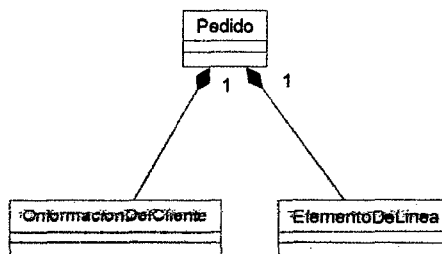


Ilustración 16: Representación de una composición

f) Dependencia

Este tipo de relación se da cuando una clase utiliza a otra. La dependencia implica una conexión semántica entre dos elementos, un cambio en el elemento independiente afectará al dependiente. Una relación de

dependencia es mostrada por una flecha punteada apuntando del dependiente al independiente.

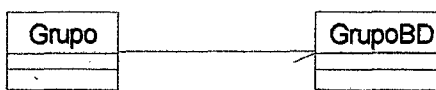


Ilustración 17: Relación de dependencia

2.6. DIAGRAMAS DE CASOS DE USO

Los Casos de Uso son una técnica para capturar información de cómo un sistema o negocio trabaja actualmente, o de cómo se desea que trabaje. (5)

El objetivo de estos diagramas es definir el comportamiento de una entidad sin especificar su estructura interna; una **entidad** en este contexto puede ser un sistema, un subsistema, o una clase.

No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos

En el modelaje de casos de uso, el sistema se mira como una "caja negra" que proporciona casos de uso. La forma en que el sistema trabaja, cómo son implementados los casos de uso y cómo trabajan internamente no es importante. De hecho, cuando el modelaje de casos de uso es hecho temprano en el proyecto, los desarrolladores no tienen idea de cómo los casos de uso serán implementados.

En un diagrama de casos de uso se representa la relación entre los actores y los casos de uso de una entidad. Por lo tanto, está formado por actores, un conjunto de casos de uso encerrados en un sistema acotado, asociaciones comunicativas entre los actores y los casos de uso, y generalizaciones entre los casos de uso.

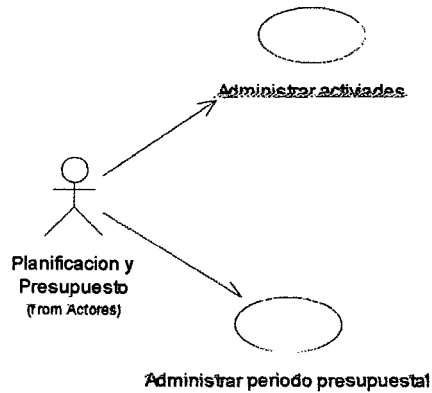


Ilustración 18: Diagrama de Casos de Uso

A) ACTOR

Un actor es un rol de un objeto u objetos externo a la entidad y que interactúa directamente con ella. Es el encargado de modelar las partes externas de la entidad. Si la entidad es un sistema los actores pueden representar tanto usuarios humanos como otros sistemas (6).

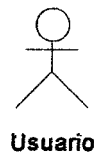


Ilustración 19: Notación UML de un actor

B) CASOS DE USO

Los casos de uso modelan un diálogo entre un actor y el sistema. Representan la funcionalidad proporcionada por el sistema; es decir, qué capacidades serán proporcionadas a un actor por el sistema. La colección de casos de uso para un sistema constituye todas las formas definidas en que el sistema será usado.



Administrar periodo presupuestal

Ilustración 20: Notación UML de un caso de uso

C) RELACIONES ENTRE LOS CASOS DE USO

Entre casos de uso pueden haber dos tipos de relaciones: incluye y extends. Mediante la relaciones "incluye" (generalizaciones con el estereotipo

«include»), la secuencia de comportamiento descrita en un caso de uso se incluye en la secuencia de otro caso de uso. Mediante las relaciones “extends” (generalizaciones con el estereotipo «extends»), un caso de uso se puede ampliar con el comportamiento definido en otro caso de uso. La relación “extends” incluye una condición para la ampliación y una referencia a un punto de extensión del caso de uso con el que está relacionado. Un **punto de extensión** es una ubicación dentro de un caso de uso en el cual se pueden insertar secuencias de acciones de otros casos de uso.

2.7. DIAGRAMA DE ESTADOS

Es un elemento de comportamiento, muestra la forma en que las partes de un modelo UML cambian con el tiempo. Un diagrama de estados es típicamente un complemento de la descripción de una clase. Muestra todos los estados posibles que los objetos de la clase puedan tener, y qué eventos causan un cambio de estado.

Los diagramas de estado capturan el ciclo de vida de objetos, subsistemas y sistemas. Describen los estados que un objeto puede tener y cómo los eventos (mensajes recibidos, tiempo que transcurre, errores, y condiciones que se vuelven ciertas) afectan esos estados a lo largo del tiempo.

Un evento puede ser otro objeto que envía un mensaje – por ejemplo, que el tiempo especificado se ha terminado – o que alguna otra condición ha sido cumplida. Un cambio de estado es llamado transición. Una transición puede tener también una acción conectada a él para especificar qué sería hecho en conexión con el estado de transición. Los diagramas de estados no son dibujados para todas las clases, solamente para aquellas que tienen una serie de estados bien definidos y en donde el comportamiento de la clase es afectado y cambiado por los estados diferentes. Los diagramas de estados pueden también ser dibujados para el sistema en su totalidad.

2.8. DIAGRAMA DE SECUENCIAS

La idea primordial es que las interacciones entre objetos se realizan en una secuencia establecida y que la secuencia se toma su tiempo de ir del principio al fin. Al momento de crear un sistema tendrá que especificar la secuencia, y para ello, utilizará el **Diagrama de Secuencias**.

Un diagrama de secuencia muestra una colaboración dinámica entre una serie de objetos. El aspecto importante de este diagrama es mostrar una secuencia de mensajes enviados entre los objetos. También son mostradas las interacciones entre los objetos, algo que sucederá en un punto específico de la ejecución de un sistema. Los diagramas consisten en una serie de objetos mostrados con líneas verticales. El tiempo pasa descendentemente en el diagrama, y el diagrama muestra el intercambio de mensajes entre los objetos a medida que pasa el tiempo en la secuencia o función. Los mensajes son mostrados como líneas con flechas de mensajes entre las líneas verticales de los objetos.

El **Diagrama de Secuencias** consta de objetos que se representan del modo usual: rectángulo con nombre (subrayado), mensajes representados por líneas continuas con una punta de fecha y el tiempo representado como una progresión vertical, necesarios para realizar la funcionalidad del escenario.

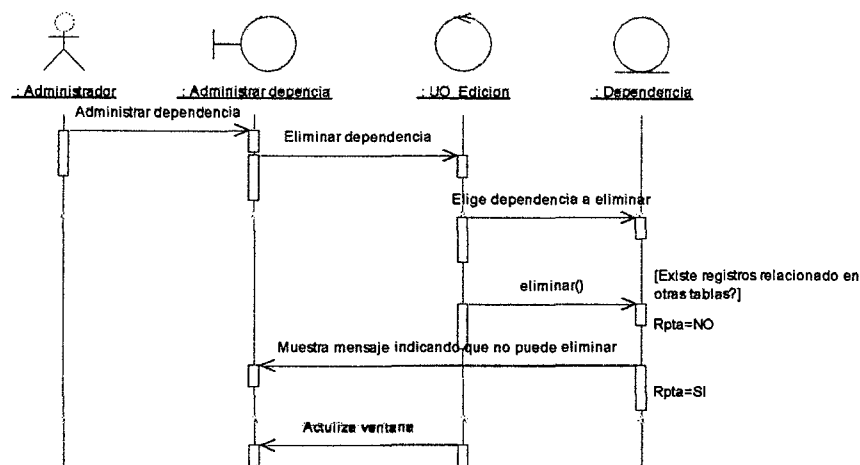


Ilustración 21: Ejemplo de un diagrama de secuencias mostrando el conjunto básico de símbolos

La relación de los diagramas de secuencias con los de casos de uso, se pone de manifiesto a través de los diagramas de secuencias de instancias y genéricos.

2.9. DIAGRAMA DE COMUNICACIONES

Un diagrama de comunicación (antes de colaboración) muestra una colaboración dinámica, como el diagrama de secuencia. Es a menudo una elección mostrar una colaboración ya sea con un diagrama de secuencia o un diagrama de colaboración. Además de mostrar el intercambio de mensajes (llamado interacción), el diagrama de colaboración muestra los objetos y sus relaciones (a veces referidos como el contexto).

"Ambos sistemas son similares. De hecho son semánticamente equivalentes. Esto significa que representan la misma información, y podrá convertir un diagrama de secuencias en un diagrama de colaboración equivalente y viceversa" (7).

Los diagramas de secuencias destacan la sucesión de las interacciones, y los diagramas de colaboración destacan el contexto y organización general de los objetos que interactúan. Además en los diagramas de colaboración, aparece las relaciones entre los roles de los objetos.

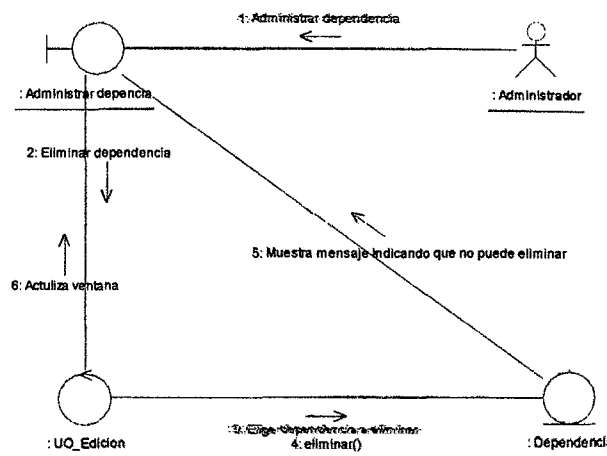


Ilustración 22: Diagrama de colaboraciones

2.10. DIAGRAMA DE ACTIVIDADES

"Los diagramas de actividades son una parte inesperada de UML. Los diagramas son particularmente útiles en conexión con el flujo de trabajo y para describir comportamiento que tiene una gran cantidad de procesamiento paralelo." (4)

Un diagrama de actividades muestra el flujo secuencial de las actividades. El diagrama de actividades es utilizado típicamente para describir las actividades realizadas en una operación, aunque puede ser también utilizado para describir otros diagramas, tal como un caso de uso o de interacción.

"Un diagrama de actividades ha sido diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso" (6). Este diagrama de actividades es un caso especial de diagrama de estados, en el cual los estados en cuestión, representan actividades que esta realizando el objeto al que se refiere el diagrama.

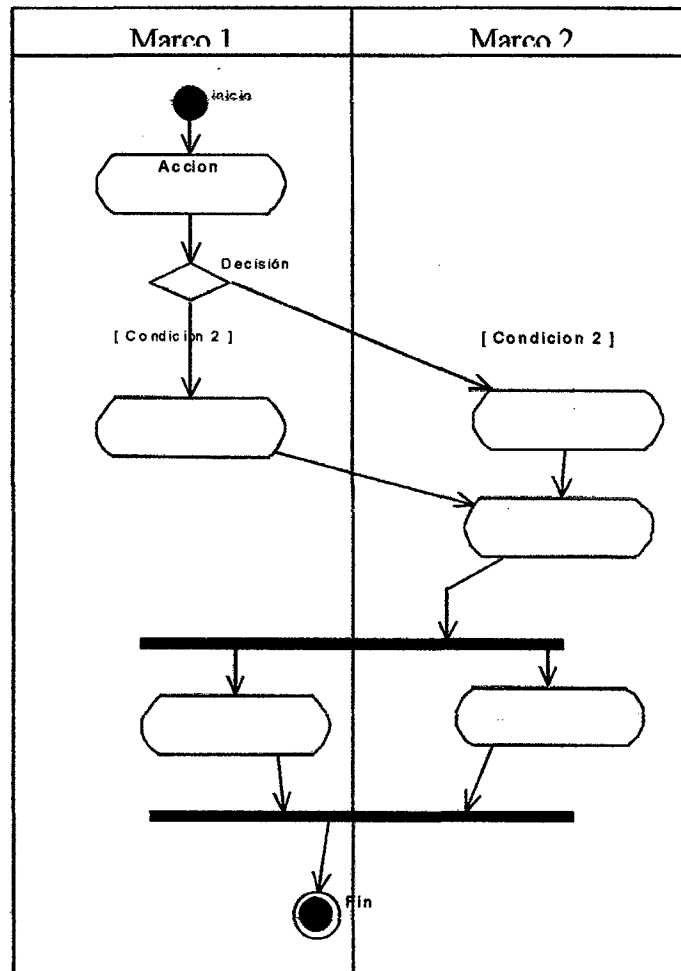


Ilustración 23: Diagrama de Actividades

2.11. DIAGRAMA DE COMPONENTES

Este diagrama representa una entidad real: un componente de software, además muestra la estructura física del código en términos de los componentes de código. Un componente puede ser un componente de código fuente, un componente binario, o un componente ejecutable. Un componente contiene información sobre la clase lógica o las clases que implementa, creando un mapeo de la vista lógica a la vista de componentes. Las dependencias entre los componentes son mostradas, haciendo fácil de analizar cómo los otros componentes son afectados por un cambio en uno de los componentes (6).

El diagrama de componentes describe los componentes de software y sus dependencias entre sí, representando la estructura del código. Los componentes son la implementación en la arquitectura física de los conceptos y la funcionalidad descrita en la arquitectura lógica (clases, objetos, sus relaciones, y

colaboraciones). Los componentes son típicamente los archivos de implementación en el ambiente de desarrollo.

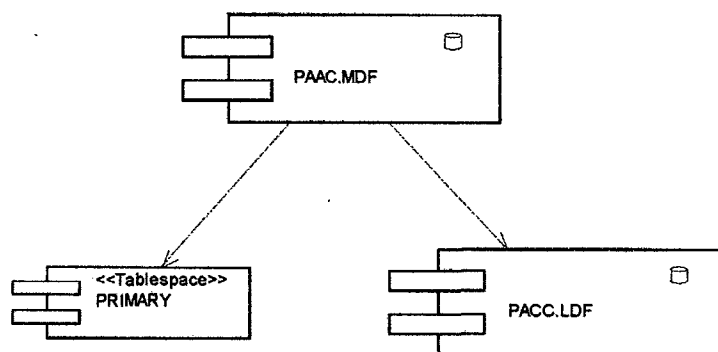


Ilustración 24: Diagrama de Componentes

2.12. DIAGRAMA DE DISTRIBUCIÓN

Los diagramas de componentes, que tratamos en el punto anterior, y los de distribución o distribución conforman las vistas físicas o de implementación. Estas vistas modelan la estructura de la implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos, como se verá en este punto. Estas vistas proporcionan una oportunidad de establecer correspondencias entre las clases y los componentes de implementación y nodos (6).

El diagrama de distribución presenta la arquitectura de tiempo de ejecución de los procesadores, dispositivos, y los componentes de software que se ejecutan en esa arquitectura. Es la descripción física última de la topología del sistema, y describe la estructura de las unidades de hardware y el software que se ejecuta en cada unidad. En una arquitectura así, debe ser posible mirar un nodo específico de la topología, ver qué componentes se ejecutan en el nodo, ver qué elementos lógicos están implementados en el componente, y finalmente trazar esos elementos al análisis de requerimientos inicial del sistema.

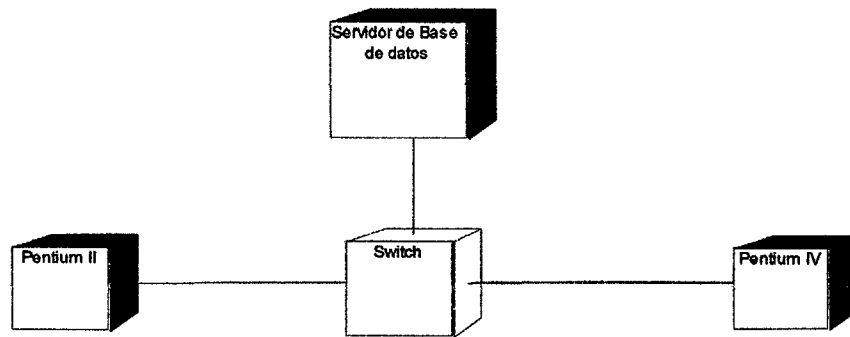


Ilustración 25: Diagrama de Despliegue

2.5. PRUEBAS DE SOFTWARE.

2.5.1. Pruebas de Caja Negra

También conocidas como Pruebas de Comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una "Caja Negra" cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas. No obstante, como el estudio de todas las posibles entradas y salidas de un programa sería impracticable se selecciona un conjunto de ellas sobre las que se realizan las pruebas. Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en nuestro sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. Entonces, dado que la prueba exhaustiva es imposible, el objetivo final es pues, encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible. Al igual que ocurría con las técnicas de Caja Blanca, para confeccionar los casos de prueba de Caja Negra existen distintos criterios. Algunos de ellos son:

1. Particiones de Equivalencia: La partición de equivalencia es un método de prueba de Caja Negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. En otras palabras, este método intenta dividir el dominio de entrada de un programa en un número finito de clases de equivalencia. De tal modo que se pueda asumir razonablemente que una prueba realizada con un valor representativo de cada

clase es equivalente a una prueba realizada con cualquier otro valor de dicha clase. Esto quiere decir que si el caso de prueba correspondiente a una clase de equivalencia detecta un error, el resto de los casos de prueba de dicha clase de equivalencia deben detectar el mismo error. Y viceversa, si un caso de prueba no ha detectado ningún error, es de esperar que ninguno de los casos de prueba correspondientes a la misma clase de equivalencia encuentre ningún error. El diseño de casos de prueba según esta técnica consta de dos pasos:

1. Identificar las clases de equivalencia.
2. Identificar los casos de prueba.

2 .Análisis de Valores Límite: La experiencia muestra que los casos de prueba que exploran las condiciones límite producen mejor resultado que aquellos que no lo hacen. Las condiciones límite son aquellas que se hayan en los márgenes de la clase de equivalencia, tanto de entrada como de salida. Por ello, se ha desarrollado el análisis de valores límite como técnica de prueba. Esta técnica nos lleva a elegir los casos de prueba que ejerciten los valores límite.

Por lo tanto, el análisis de valores límite complementa la técnica de partición de equivalencia de manera que:

- En lugar de seleccionar cualquier caso de prueba de las clases válidas e inválidas, se eligen los casos de prueba en los extremos.
- En lugar de centrarse sólo en el dominio de entrada, los casos de prueba se diseñan también considerando el dominio de salida.

Las pautas para desarrollar casos de prueba con esta técnica son:

- Si una condición de entrada especifica un rango de valores, se diseñarán casos de prueba para los dos límites del rango, y otros dos casos para situaciones justo por debajo y por encima de los extremos.
- Si una condición de entrada especifica un número de valores, se diseñan dos casos de prueba para los valores mínimo y máximo, además de otros dos casos

de prueba para valores justo por encima del máximo y justo por debajo del mínimo. - Aplicar las reglas anteriores a los datos de salida.

- Si la entrada o salida de un programa es un conjunto ordenado, habrá que prestar atención a los elementos primero y último del conjunto.

CAPITULO 3. DESARROLLO METODOLÓGICO

3.1.- INTRODUCCION

3.1.1. GESTION DE LA CONFIGURACION

a) Documentación

Toda la documentación sobre el Sistema de Biblioteca de la Municipalidad Provincial de Cajamarca está recogida en este documento, desarrollando cada fase del proceso (inicio y elaboración) en un subcapítulo distinto dentro de este capítulo. Por otro lado, los resultados de cada flujo de trabajo realizado se documentan como sub ítem en cada subcapítulos dentro de cada una de las dos fases ejecutadas.

- **Modelos Construidos**

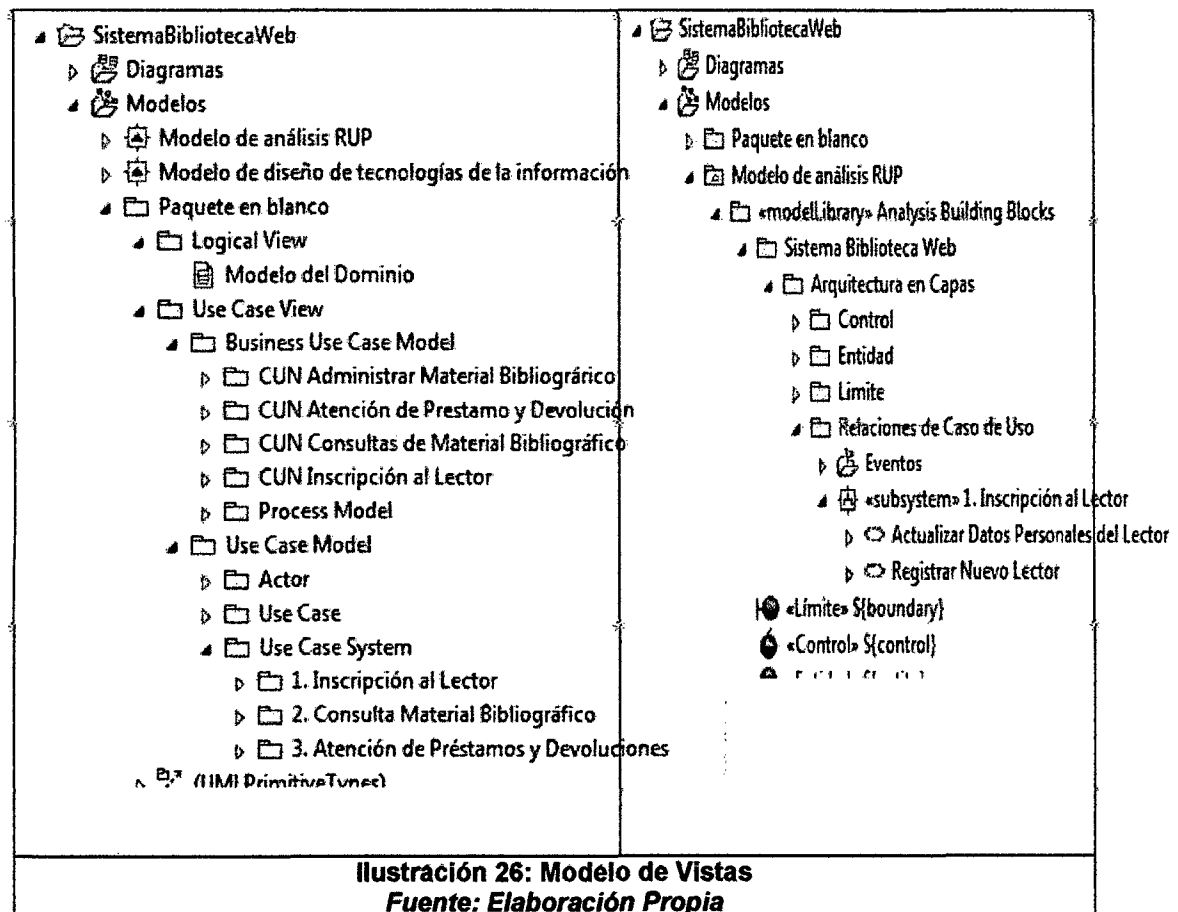
Los modelos que se generan como resultado de la aplicación del proceso de desarrollo, se va construir utilizando la herramienta **Rational Software Architect de IBM**. Los modelos lo voy a estructurar siguiendo el esquema propuesto por Rational. Los modelos se organizan en cuatro vistas según el aspecto del sistema que pretenden representar:

- **Vista de Casos de Uso.** Comportamiento visible del sistema y del entorno. En esta vista se separa en dos paquetes, el primero es el *paquete de modelo de casos de uso del negocio*, el cual incluye un paquete por cada proceso que, a su vez, contendrá los diagramas de actividades para describir el flujo de tareas del proceso. El segundo *paquete contiene el modelo de casos de uso del sistema*, donde incluiremos un paquete por cada proceso de sistema y su correspondiente diagrama de casos de uso.
- **Vista Lógica.** Entidades del entorno y del sistema (clases); estructura lógica y comportamiento interno. Esta vista se divide también en paquetes, uno para cada flujo de trabajo que opera sobre la lógica del sistema: **Modelo del Dominio, Modelo del Análisis y Modelo del Diseño**
 - El *Modelo del Dominio* incluye el diagrama de clases del dominio.
 - El *Modelo del Análisis* incluye un paquete por cada subsistema del análisis que se haya identificado y un diagrama con todas las clases de entidad identificadas (clases persistentes). Internamente, cada subsistema incluye su modelo de clases (interfaces, controladores y entidades utilizadas) en uno o varios diagramas. Dependiendo de la

complejidad del subsistema, éste puede estar dividido en subsistemas más pequeños, cada uno en su paquete.

- El Modelo del Diseño incluye un paquete con los diagramas UML necesarios para describir la arquitectura, aunque la descripción completa de la arquitectura se especificará en un documento separado.
- **Vista de Componentes.** Software del sistema (estructura y relaciones con elementos lógicos). En esta vista se deben enumerar los componentes software desarrollados en el flujo de trabajo de implementación, pero dado que este trabajo cae fuera del ámbito de este proyecto, no se abordara.
- **Vista de Despliegue.** Equipos físicos y redes que los conectan (estructura y relaciones con componentes software). La vista de despliegue consiste en un único diagrama que especifica la arquitectura física del sistema.

La siguiente figura muestra la estructura de los modelos desarrollados.



3.2.- FASE I: INICIO

En la fase de inicio se realiza las actividades del RUP y la planificación inicial del sistema. En la Biblioteca de la Municipalidad Provincial de Cajamarca donde se administra la circulación de Material Bibliográfico entre la comunidad ciudadana se analizaron los distintos problemas vigentes.

3.2.1. MODELADO DEL NEGOCIO

El negocio se inicia a partir de cuestionarios y entrevistas con el personal implicado en el sistema. En principio, he recogido la información a partir de mi conocimiento sobre el funcionamiento del proceso de Consulta y Préstamos de Material Bibliográfico, validando los resultados con personal de las diferentes Áreas implicadas en este proceso, como personal que labora en la atención de la Biblioteca.

El Proceso del Sistema de Control Bibliográfico de la Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca se ha visto que está compuesto de varios subprocesos, siendo estos el Proceso Inscripción al Lector, Consulta de Material Bibliográfico y el Proceso Atención de Préstamo y Devoluciones del Material Bibliográfico y el Proceso de Administración del Material Bibliográfico y el Proceso de Ingreso de Material Bibliográfico.

a) Proceso de Inscripción al Lector

La persona que se acerca a inscribirse como lector lo puede hacer bajo dos modalidades. Los que vienen por primera vez y los que renuevan carnet o piden duplicado por perdida.

Los que vienen por primera vez, estas personas se acercan a la ventanilla. Tendrán que llenar una ficha con sus respectivos datos, para luego hacer un pago si son niños (as) menores de 5 años no pagan, si tienen entre 6 años y 15 años de edad realizan un pago S/. 1.90 Nuevos Soles y los que tienen mas de 16 años de edad realizan un pago de S/.4.10 Nuevos Soles en caja, después de llenar la ficha con sus datos se presenta a la ventanilla de inscripción para de esta manera hacer el registro y emitir el carnet y luego registrar la ficha en el padrón de lectores de la Biblioteca Municipal.

Los lectores que vienen a renovar carnet o piden duplicado por pérdida se deberán acercar a la ventanilla y solicitar que el encargado busque sus datos en el padrón para verificar que no tenga ningún antecedente (maltrato de libro, hurto u otra infracción en contra del servicio de biblioteca), sino tiene ningún antecedente se le dará orden para que haga el pago en caja por la renovación o duplicado de carnet y así emitir el nuevo carnet.

1. Actores del Negocio




| | |
|--|--|
|  <p style="text-align: center;">Actor</p> <p><i>Figura N° 1, Actor del Negocio, Persona</i></p> | <p>Es la persona que realiza solicitud de inscripción para poder ser Lector de la Biblioteca Municipal.</p> |
|  <p style="text-align: center;">Actor</p> <p><i>Figura N° 2, Actor del Negocio, Bibliotecario</i></p> | <p>Es el trabajador de la Biblioteca Municipal, que atiende la solicitud de inscripción.</p> |
|  <p style="text-align: center;">Actor</p> <p><i>Figura N° 3, Actor del Negocio, Cajero</i></p> | <p>Es el trabajador del Área de Tesorería Municipal, que se encarga de hacer el cobro por concepto de Inscripción.</p> |

Tabla 2: Actores del Negocio en el Proceso de Inscripción al lector.

2. Casos de Uso del Negocio

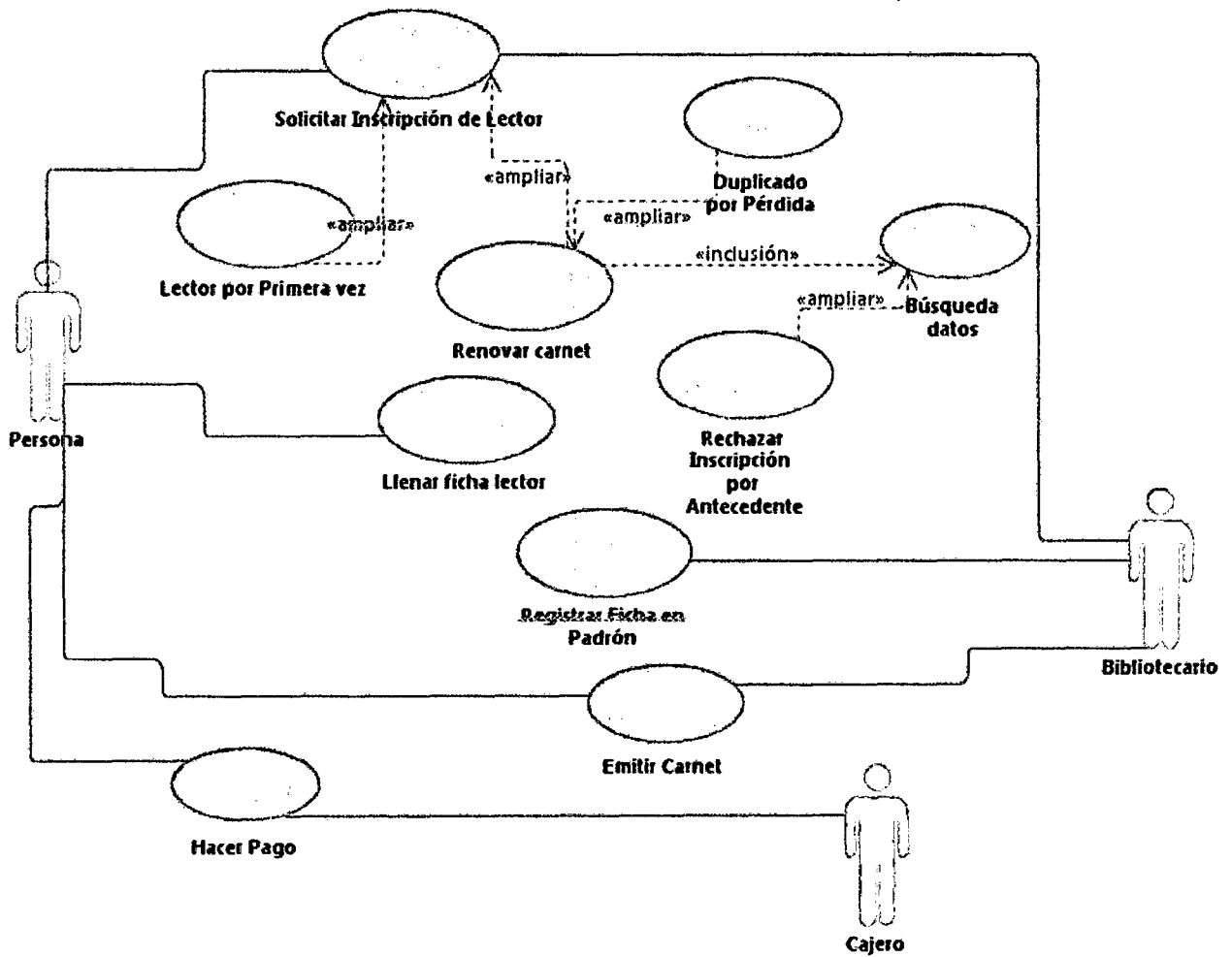


Ilustración 27: Caso de Uso del Negocio, Inscripción al Lector.

Fuente: Elaboración Propia

3. Diagrama de Actividades Proceso de Inscripción al Lector

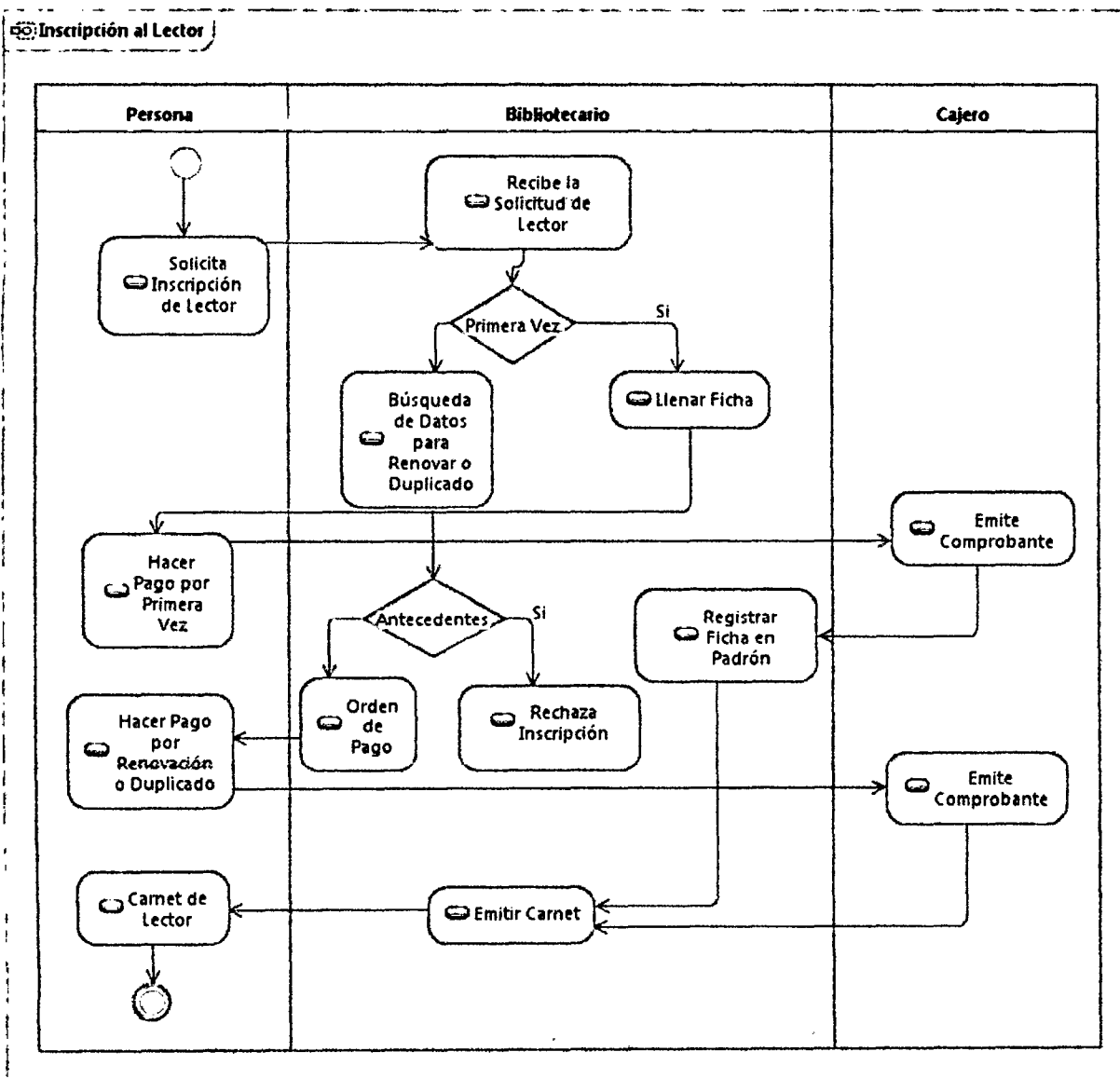


Ilustración 28: Diagrama de Actividades, Inscripción al Lector
Fuente Elaboración Propia

b) Proceso de Atención Consultas de Material Bibliográfico

El proceso de atención al lector se inicia cuando se acerca a la ventanilla de consulta, en donde se encuentran las fichas de los materiales bibliográficos clasificadas por materias y nombre de Autores. Cuando el lector ya encontró el material bibliográfico de su requerimiento toma nota de los datos de la publicación y extrae una ficha de pedido para llenar los datos y posteriormente entregarla al bibliotecario.

1. Actores del Negocio

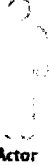
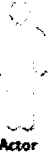
| | |
|--|---|
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 4, Actor del Negocio, Bibliotecario</i></p> | <p>Es la persona que atiende las consultas referentes al material Bibliográfico, Además actualiza información de fichas de los Materiales Bibliográficos.</p> |
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 5, Actor del Negocio, Lector</i></p> | <p>Es quien realiza la consulta del material bibliográfico según su requerimiento.</p> |

Tabla 3: Actores del Negocio en el Proceso de Atención Consultas de Material Bibliográfico.

2. Casos de Uso del Negocio

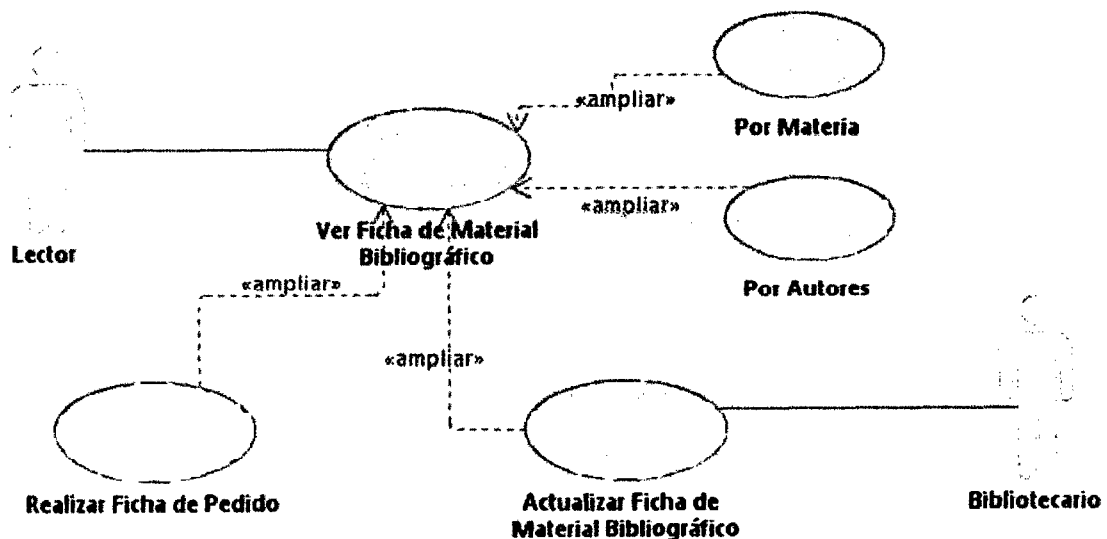


Ilustración 29: Caso de Uso del Negocio, Consulta de Material Bibliográfico

Fuente: Elaboración Propia

3. Diagrama de Actividades Proceso de Consultas de Material Bibliográfico

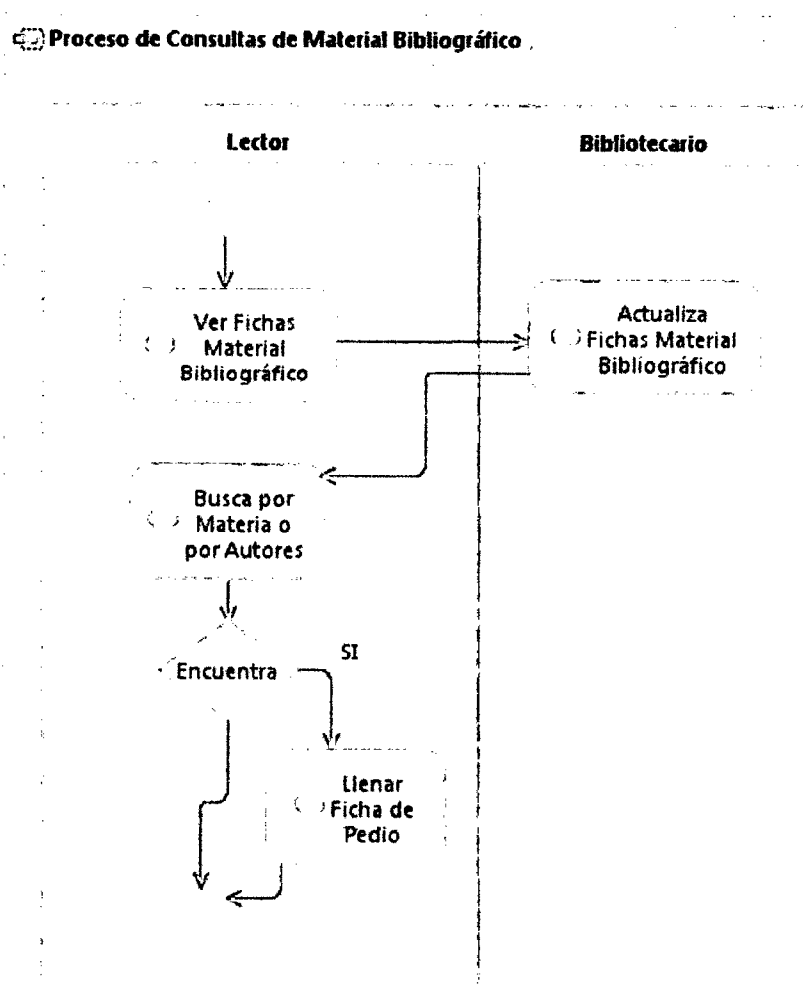


Ilustración 30: Diagrama de Actividades, Proceso Consultas de Material Bibliográfico
Fuente Elaboración Propia

C) Proceso de Atención de Préstamo y Devolución de Material Bibliográfico.

El bibliotecario recepciona la ficha de pedido préstamo y carnet, e irá a buscar en los estantes de las publicaciones el material bibliográfico solicitado en la ficha, para luego entregar al lector y retener el carnet conjuntamente con la ficha. En caso que el material bibliotecario se encuentre en la sala de lectura, el bibliotecario comunicará al lector y en ciertas ocasiones el bibliotecario le brinda al lector material bibliográfico alternativos para el tema de búsqueda. De aceptar el material bibliográfico alternativa le hará llenar los datos de la misma propuesta, en caso contrario se entregará el carnet para abortar el proceso atención al lector.

Cuando el lector entregue el material bibliográfico ya sea por días establecidos o haya concluido la lectura, se deberá devolver el material bibliográfico al bibliotecario, éste revisará la publicación y de estar en buenas condiciones entregará el carnet al lector y concluirá el proceso de atención al lector. En caso el material bibliográfico este deteriorado se le aplicará la sanción correspondiente. También indicar si ha pasado la fecha límite de devolución, se tendrá que pagar multa.

1. Actores del Negocio




| | |
|--|--|
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 6, Actor del Negocio, Bibliotecario</i></p> | <p>Es la persona que atiende los préstamos y devoluciones, sanciona al lector incumplido.</p> |
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 7, Actor del Negocio, Lector</i></p> | <p>Es quien hace la solicitud del material bibliográfico mediante la entrega de ficha de pedido, y luego devuelve el material bibliográfico.</p> |
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 8, Actor del Negocio, Cajero</i></p> | <p>Es el trabajador del Área de Tesorería Municipal, que se encarga de hacer el cobro por concepto de Multa.</p> |

Tabla 4: Actores del Negocio en el Proceso de Atención de Préstamo y Devolución de Material Bibliográfico.

2. Casos de Uso del Negocio

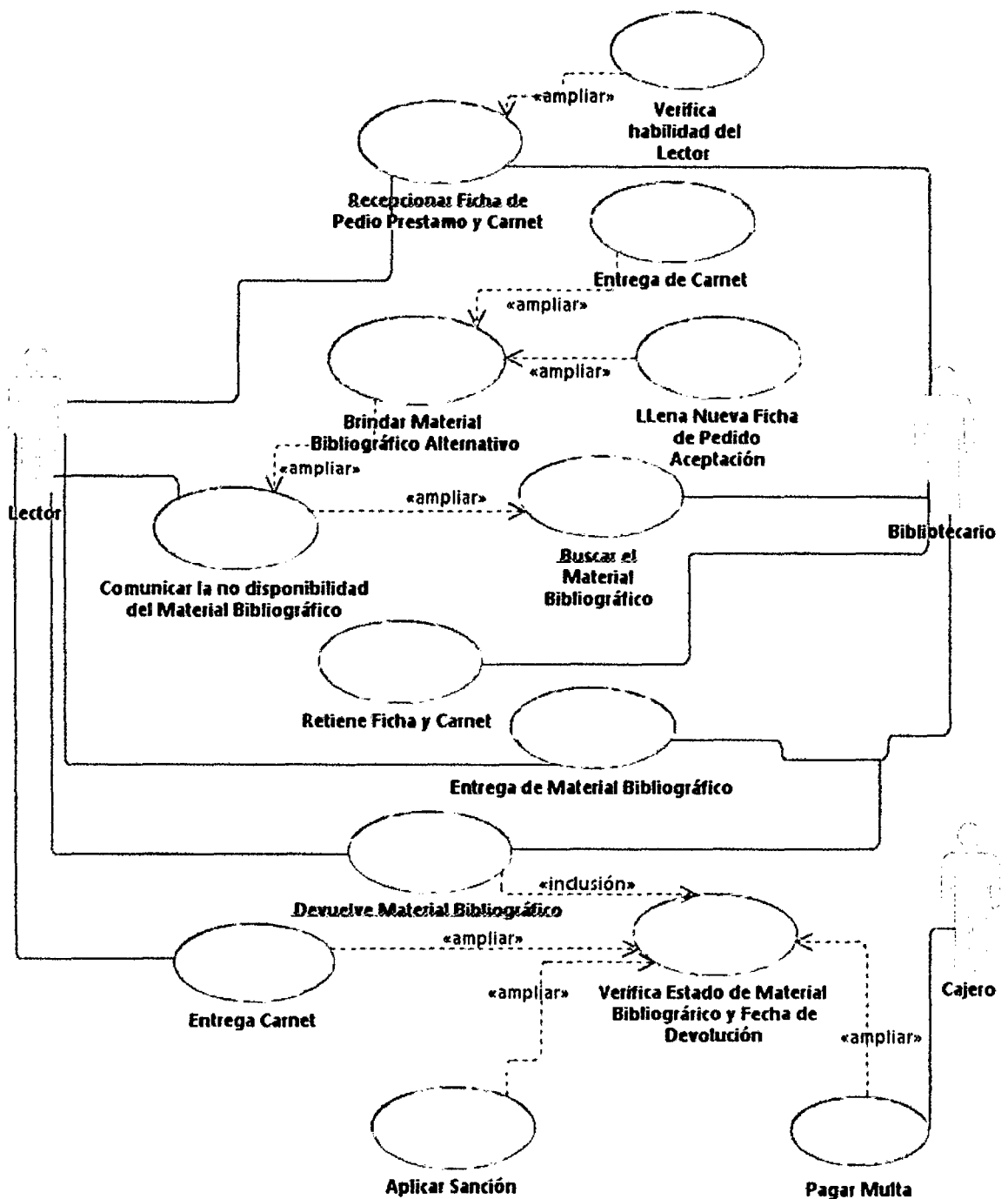


Ilustración 31: Caso de Uso del Negocio, Proceso Atención de Préstamos y Devoluciones del Material Bibliográfico

Fuente : Elaboración Propia

3. Diagrama de Actividades de Proceso de Préstamo y Devolución de Material Bibliográfico

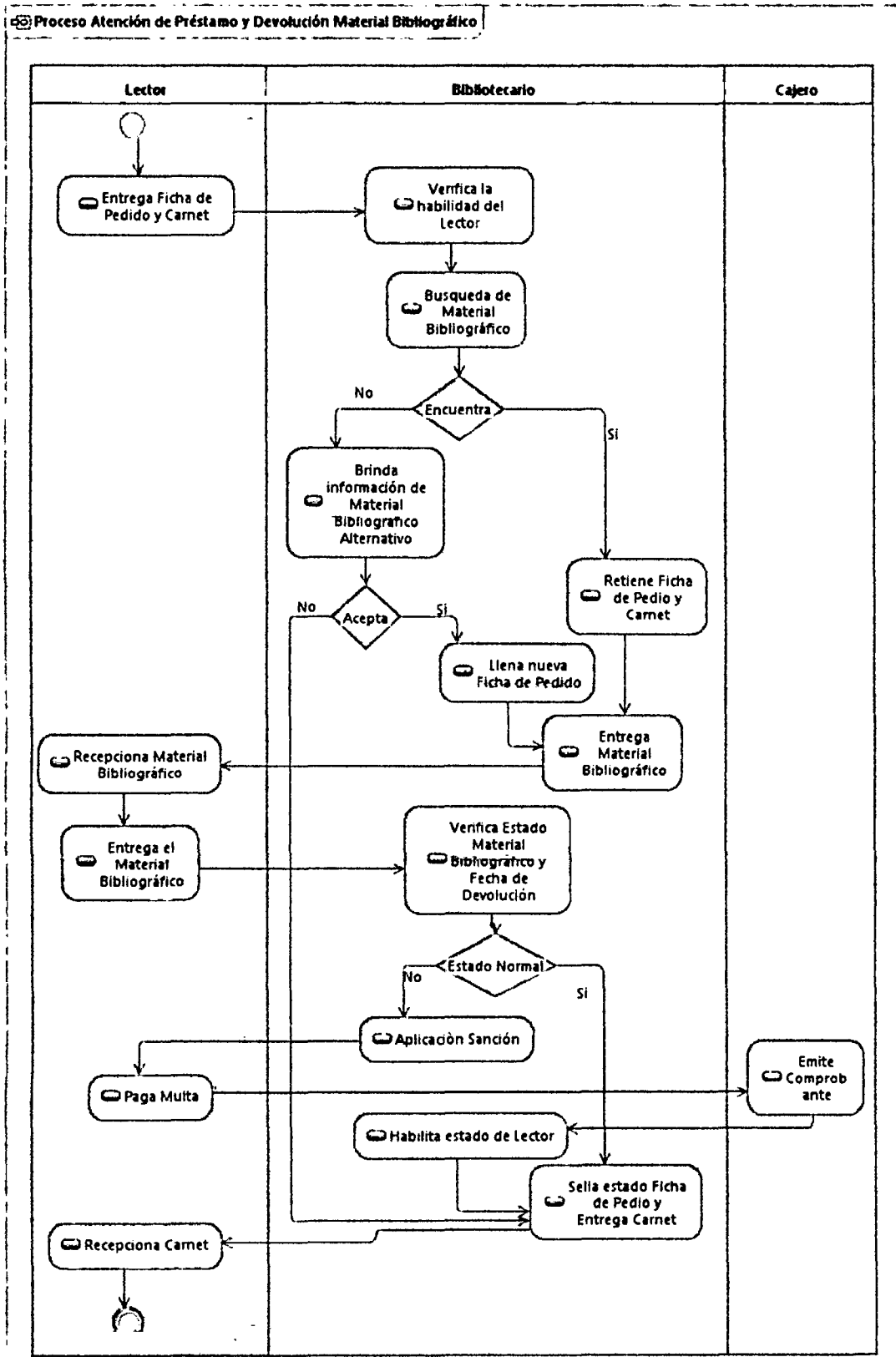


Ilustración 32: Diagrama de Actividades Proceso Atención de Préstamo y Devolución del Material Bibliográfico.

Fuente: Elaboración Propia

Hernán Vargas Cueva

d) Proceso de Administrar el Material Bibliográfico.

El proceso de Administrar el Material Bibliográfico, comienza por adquisición, aquí se elaboran los pedidos de material Bibliográfico, luego de pasar por el área de Abastecimiento de la Municipalidad y luego darse el proceso de adjudicación según ley, se receptiona el material de lectura. O también se puede adquirir el material bibliográfico por donación de alguna entidad externa.

Para elaborar estos pedidos, se consulta el inventario y estado del material bibliográfico, el cual es actualizado por los bibliotecarios, para saber cuáles deben ser remplazados, ya sea porque son ediciones pasadas y se requiere de nuevas ediciones o porque el material ya está demasiado deteriorado.

Cada cierto período, la biblioteca es implementada de a poco con materiales bibliográficos más modernos y nuevas ediciones, con los que los lectores utilizan como consulta para sus trabajos o investigaciones.

Todo material bibliográfico que ingresa debe quedar registrado en la biblioteca para que luego pueda ser prestado a los lectores para su uso. Después de ser adquirido el material, se pasa hacer la catalogación y registro en las fichas de cada material bibliográfico. El material bibliográfico que se adquiere debe ser catalogado antes de ser registrado, la catalogación consiste en agregar una nueva entrada en el sistema decimal Dewey de la biblioteca. Este sistema es parecido a un índice del material bibliográfico (libros), en el que se encuentran clasificados por categoría o materia, y dentro de cada materia, esta se encuentra subdividida de acuerdo al tema del libro. Para catalogar se utilizan los datos del material bibliográfico y el sistema Dewey.

Después de que se cataloga el material bibliográfico, éste pasa a ser registrado, para lo cual se llena un formulario llamado "Hoja de Ingreso", que contiene los datos del material bibliográfico.

1. Actores del Negocio


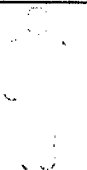




| | |
|---|---|
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 9, Actor del Negocio, Jefe de Biblioteca</i></p> | <p>Es quien tiene la administración de la biblioteca y el control de altas (catalogación y registro) y bajas del Material Bibliográfico. Es capaz de realizar cambios en las fichas de Materiales Bibliográficos.</p> |
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 10, Actor del Negocio, Bibliotecario</i></p> | <p>Informa el inventario y estado del material Bibliográfico.</p> |
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 11, Actor del Negocio, Entidad Externa</i></p> | <p>Entidad que realiza Donaciones del material bibliográfico.</p> |
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 12, Actor del Negocio, Unidad Logística</i></p> | <p>Unidad que se encarga de realizar el proceso de compras de bienes y servicios de la Municipalidad Provincial de Cajamarca.</p> |
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 13, Actor del Negocio, Lector</i></p> | <p>Realizan las sugerencias de la adquisición de nuevos materiales bibliográficos de acuerdo a sus necesidades y mundo actual.</p> |
|  <p style="text-align: center;">Actor</p> <p style="text-align: center;"><i>Figura N° 14, Actor del Negocio, Editorial</i></p> | <p>Envía catálogo de Material Bibliográfico actualizados a solicitud de la administración de la biblioteca</p> |

Tabla 5: Actores de Negocio

2. Casos de Uso del Negocio

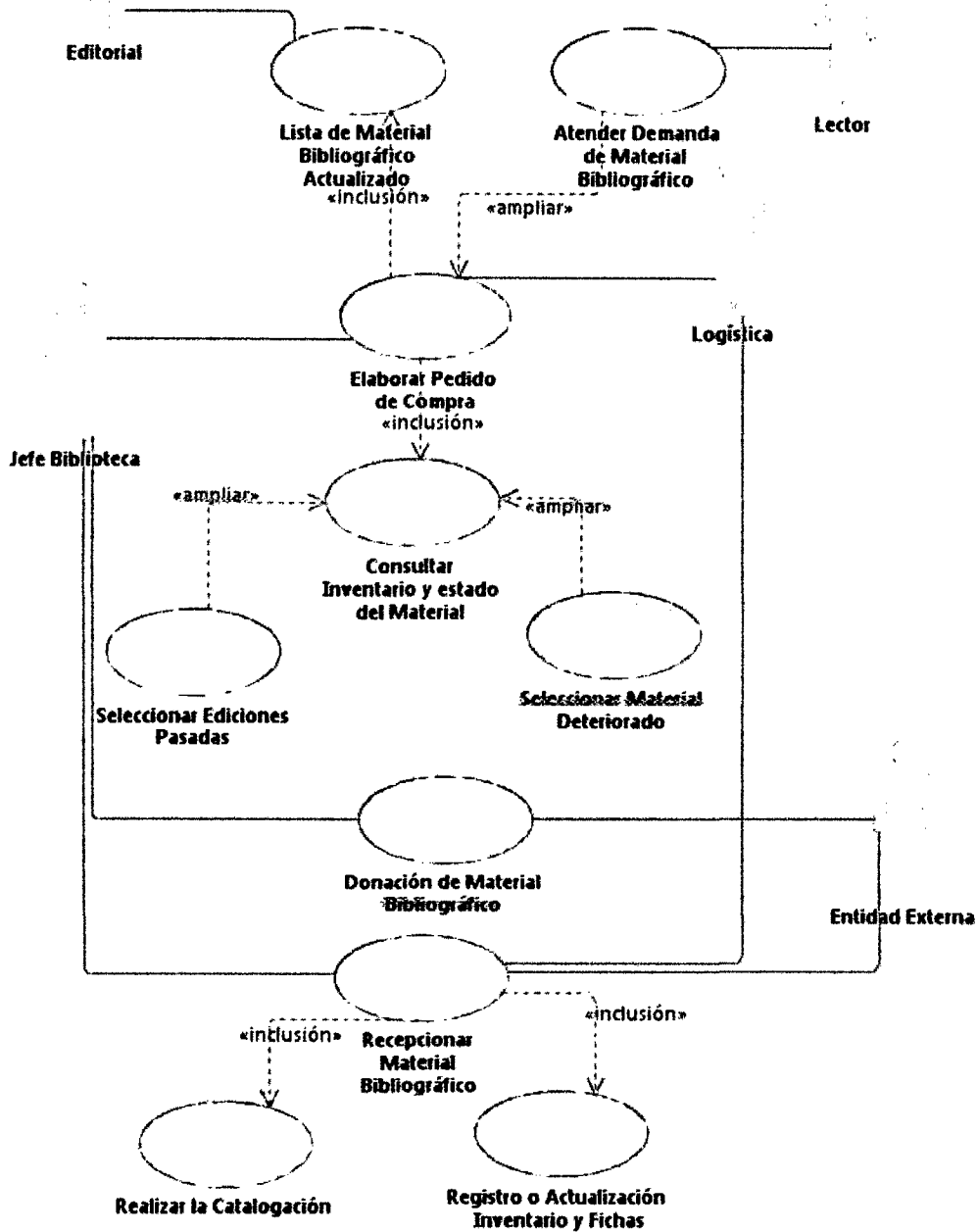


Ilustración 33: Caso de uso del Negocio, Proceso de Administrar el Material Bibliográfico
 Fuente: Elaboración Propia

3. Diagrama de Actividad del Proceso Administrar el Material Bibliográfico

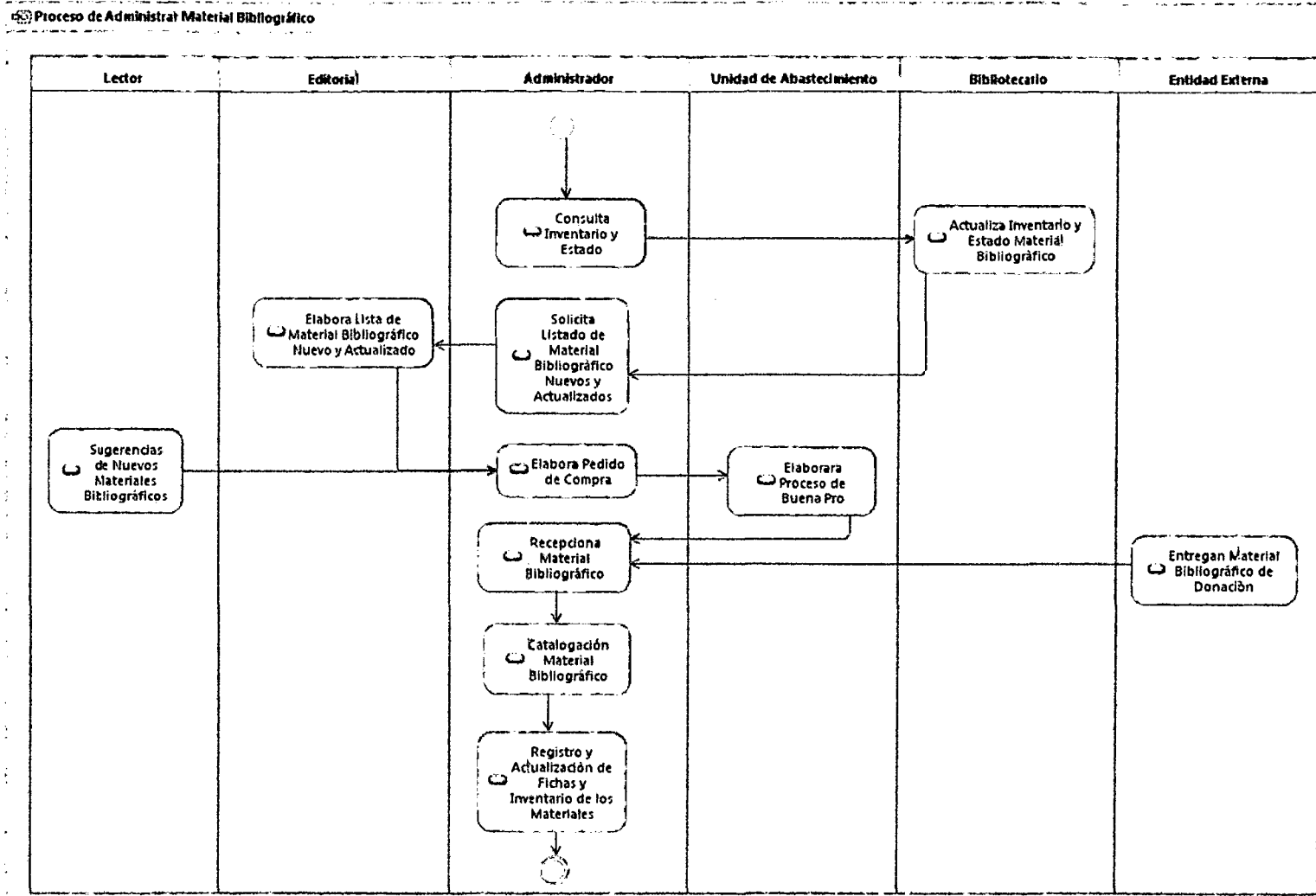


Ilustración 34: Diagrama de Actividades Proceso de Administrar el Material Bibliográfico

Fuente: Elaboración Propia

Hernán Vargas Cueva

e) Modelo de Objetos

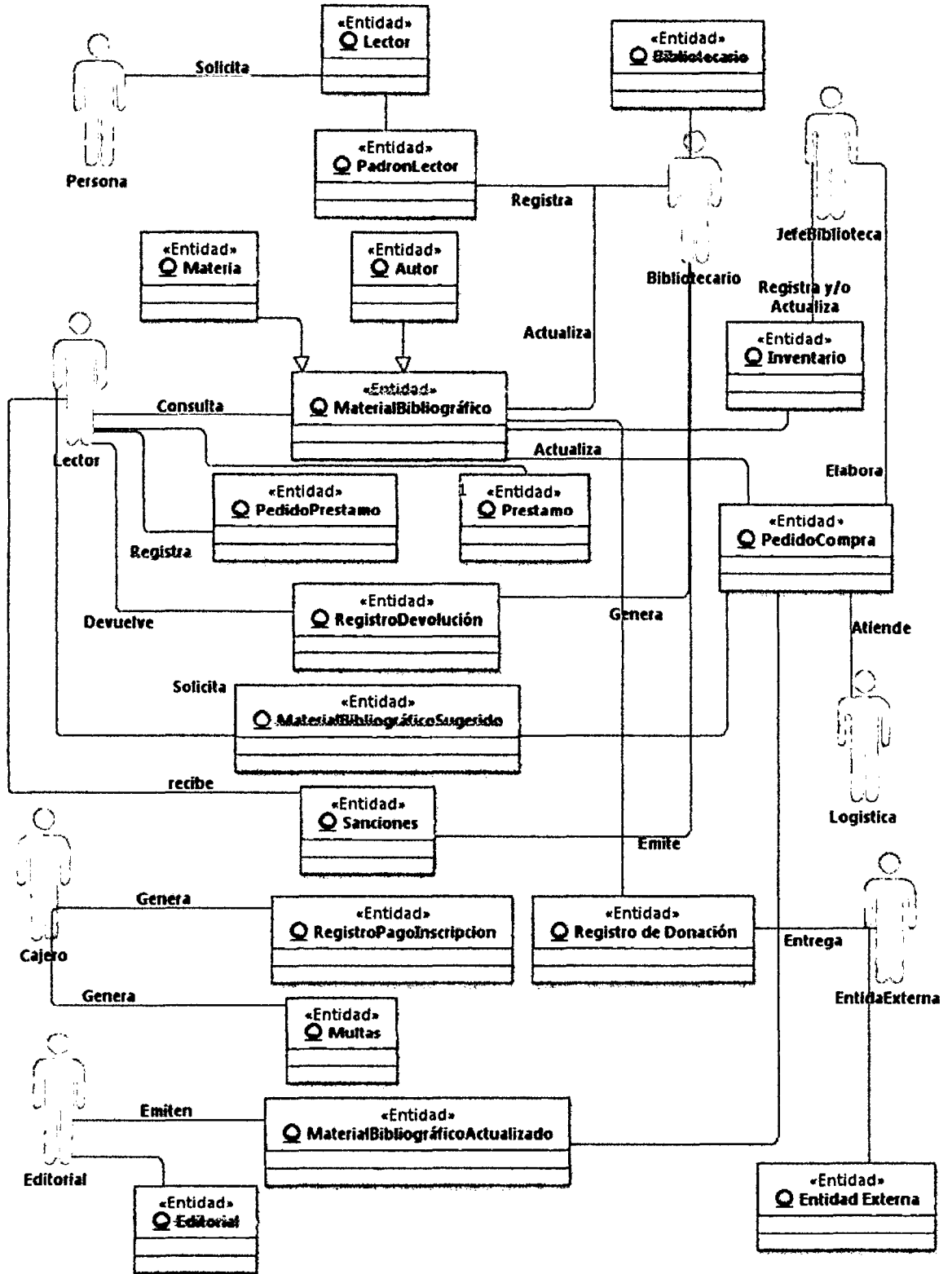


Ilustración 35: Modelo de Objetos del Negocio
Fuente: Elaboración Propia

f) Proceso General de Consulta y Atención de Material Bibliográfico

Proceso General de Biblioteca

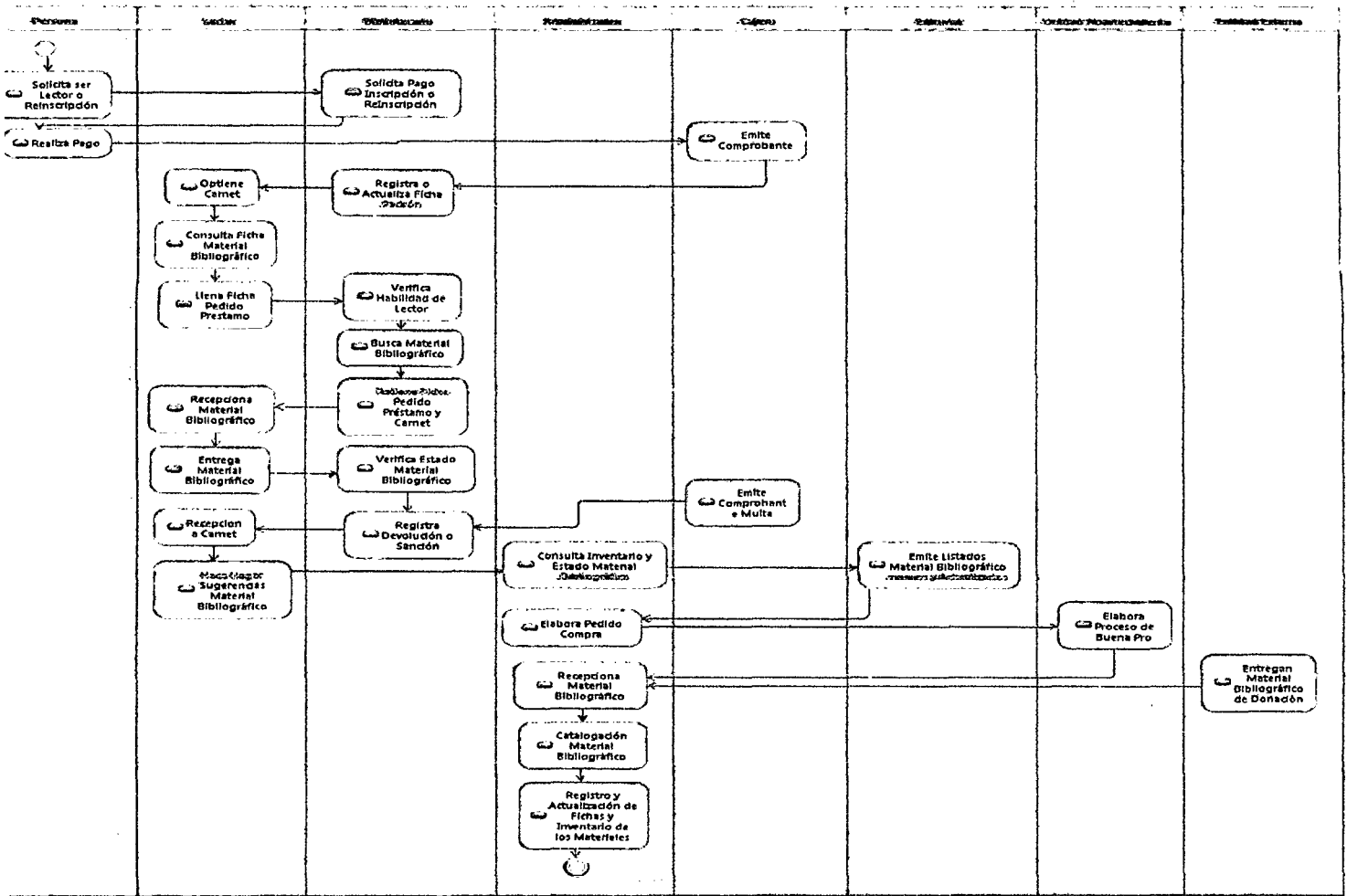


Ilustración 36: Proceso General de Consulta y Atención de Material Bibliográfico
Fuente: Elaboración Propia

b) Visión Statement

1. Introducción

1.1. Propósito del documento

Esta especificación de requisitos del producto establece los requisitos funcionales y no funcionales para el diseño y desarrollo del Sistema de Control Bibliográfico para la ciudadanía de Cajamarca.

1.2. Ámbito del documento

Este documento especifica todas las características operacionales y físicas del Sistema de Control Bibliográfico.

2. Descripción General

2.1. Descripción general del proyecto

Este proyecto tiene como objetivo final realizar el análisis y diseño de un Sistema de Control Bibliográfico, para mejorar la integridad de la información que se genera en el proceso de Consulta y Atención en los Préstamos y Devoluciones del Material Bibliográfico de la Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca. Sin embargo, no se pretende llegar a la idea de una Biblioteca virtual sin Papeles.

Durante el proyecto se seguirá, además de los procesos de gestión habituales, un proceso de desarrollo basado en UML (definido precisamente en este proyecto).

2.2. Descripción general del producto

El producto será un sistema que permita automatizar la información del proceso de Inscripción Lector, Consulta, Atención de Préstamos y Devoluciones, así como la Administración del Material Bibliográfico, que ofrecerá una mejor atención de calidad a los usuarios y seguridad en los datos.

3. Requisitos del producto

3.1. Requisitos de seguridad física

Los equipos servidores encargados de prestar los servicios del Sistema de Control Bibliográfico, deben permanecer en una habitación específicamente reservada para ellos y cuyo acceso esté restringido al

personal autorizado. Los equipos cliente deberán de recibir un mantenimiento preventivo y correctivo tanto de hardware y software.

3.2. Requisitos de seguridad lógica

En caso de permitir control remoto de los servidores de Sistema de Control Bibliográfico, éste deberá hacerse sobre un protocolo seguro que garantice, al menos, la autenticación del administrador (único usuario con acceso permitido al servidor), la privacidad y la integridad.

Por otro lado, cualquier usuario que quiera acceder al Sistema de Control Bibliográfico deberá identificarse frente al sistema.

Todas las acciones que modifiquen de alguna manera el estado del Sistema de Control Bibliográfico, quedarán registradas con la identidad del usuario autenticado en la entrada al sistema, evitando de esta manera el repudio.

Un usuario sólo podrá realizar una operación si se le ha asignado el rol que le autoriza a ello (es decir, cada operación tendrá asociada una lista de roles autorizados).

Un usuario sólo podrá consultar una información si presenta el rol que le autoriza a ello (es decir, cada información tendrá asociada un conjunto de roles autorizados).

3.3. Requisitos de la interfaz externa

3.3.1. Requisitos de la interfaz de usuario

La interfaz de usuario se presentará en una Interface Web. La navegación deberá ser intuitiva, presentando en todas las páginas web, menús desplegables para acceder a los distintos módulos del Sistema de Control Bibliográfico.

3.4. Características del producto

3.4.1. Arquitectura de desarrollo del Sistema

La arquitectura que tendrá el sistema es una arquitectura en capas, en esta arquitectura la carga se divide en tres partes o capas con un reparto claro de funciones: una capa de presentación (interfaz de usuario), otra **capa empresarial** (donde se encuentra modelado el negocio) y otra **capa de datos** (persistencia). Una capa solamente tiene relación con la siguiente.

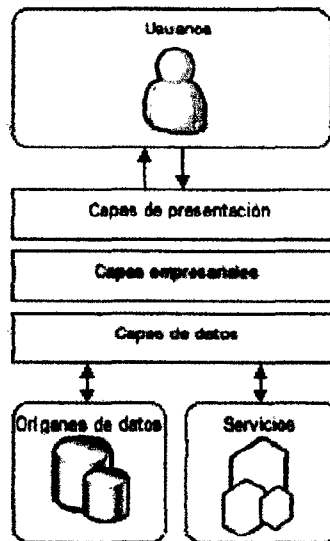


Ilustración 37: Arquitectura en Capas

La Arquitectura en capas es una de las más usadas actualmente, esta arquitectura permite muy fácilmente cambiar la vista de presentación de los usuarios, así se presenta la posibilidad de desplegar la solución en Web. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

3.4.2. Lenguaje de programación, herramientas de desarrollo y administración de base de datos.

- El lenguaje de programación a emplear será **PHP 5**, este lenguaje de programación es de uso libre, además es una tecnología que es utilizada en el desarrollo de todo tipo de aplicaciones, y permite su despliegue en diferentes plataformas de sistemas operativos, a través de la ejecución del compilador al lado del Servidor. Este lenguaje ha sido escogido ya que presenta muchas ventajas y principalmente por qué se viene utilizando para los demás sistemas y no implicaría costos para la institución. (8)
- La herramienta para el desarrollo del sistema es **Eclipse con Plugin para PHP**, es un entorno de desarrollo integrado que permite desarrollar aplicaciones en php, es uno de los más usados en el mercado y es de uso libre por lo que lo he escogido.
- La Herramienta para la Administración de la Base de Datos es **MySQL Server 5.1**, éste es un sistema administrador de base de

datos relacionales, multiusuario, multihilo, además actualmente es empleado en muchas instituciones, así mismo en la Municipalidad Provincial de Cajamarca. Esta herramienta se utilizará ya que para todas las dependencias de la municipalidad viene funcionando como motor de base de datos, por sus ventajas propias que presenta, como la de ser de uso libre, lo que no implicaría costos para la institución además de poder desplegarse en diferentes plataformas de sistemas operativos.

3.4.3. Plataformas soportadas

La Interfaz con el usuario va a ser presentada en Web, PHP se ejecuta en un Compilador con el Servidor Web Apache, y ésta se puede ejecutar sobre varios sistemas operativos, así el sistema podrá ser ejecutado en varias plataformas. Por otro lado como la arquitectura elegida es en capas, ésta permite que la presentación del sistema pueda darse en el futuro dependiendo de las necesidades que se tengan, vía Móvil, permitiendo así desplegarse el sistema sobre un navegador Web Móvil.

1.4.4. Funcionalidades del Sistema

1.4.4.1. Inscripción al Lector

- El bibliotecario permitirá registrar o dar de alta a un nuevo lector (por primera vez).
- El bibliotecario o el lector podrá actualizar datos personales del lector.
- En caso que el lector solicite renovar carnet ya sea por perdida o por no habilidad de lector, el sistema verificará el estado actual y mostrará un aviso visual si el lector esta hábil o no.
- El bibliotecario podrá actualizar el estado de hábil de un lector por alguna sanción y/o multa.
- El sistema emitirá un formato digital para imprimir el carnet del lector.

1.4.4.2. Consulta de Material Bibliográfico

- Buscar Material Bibliográfico por título, autor, editorial, y/o Materia.
- Buscar que el Material Bibliográfico esté Disponible al menos con un Ejemplar.
- Los lectores podrán solicitar Material Bibliográfico, a través de un registro de Pedido de Préstamo, dependiendo de la existencia y disponibilidad del Material Bibliográfico

1.4.4.3. Atención de Préstamos y Devolución de Material Bibliográfico

Este proceso se subdivide en dos como es lo de Préstamo y Devolución:

Prestamos:

- El sistema presentará todos los pedidos de préstamo en cola y volverá a verificar la existencia de al menos un ejemplar del Material Bibliográfico antes del registro de préstamo.
- El lector podrá solicitar una cantidad limitada de Material Bibliográfico.
- La disponibilidad o no disponibilidad del Material Bibliográfico será comunicado al usuario a través de un mensaje visual.
- El periodo de préstamo puede ser en horas dentro de la sala de lector o un máximo de préstamo es de n días(n cantidad de días variable).
- En caso de que el periodo de préstamo expire un día no lectivo, se asumirá que finaliza al día siguiente.
- El usuario deberá portar Carnet de Lector y debe estar hábil como lector para retirar Material Bibliográfico.
- No podrán retirar Material Bibliográfico los usuarios con sanciones en vigor.
- Los bibliotecarios tendrán un reporte de devolución para conocer la fecha de expiración. Y a los lectores se le comunicará vía correo electrónico, un día antes de la fecha de vencimiento.

- El listado de Material Bibliográfico en préstamo será administrado por el Sistema de Control de Material Bibliográfico.

Devolución:

- El lector deberá devolver el Material Bibliográfico antes de la expiración del periodo de préstamo y tendrá que registrar la devolución.
- En caso de retraso en la devolución de Material Bibliográfico, el sistema mostrara un reporte de alerta.
- El bibliotecario podrá actualizar el estado de no hábil de un lector *por alguna sanción y/o multa, así como el registro de las mismas.*
- El sistema de control de Biblioteca permitirá exportar y consultar el comprobante de Pago por efecto de Sanción.
- Solo el Sistema de Control podrá anular la sanción.

3.4.4.4. Administración del Material Bibliográfico

- El administrador podrá registrar altas ya sea por orden de compra o donación.
- El sistema emitirá reporte del Inventario Actualizado.
- El bibliotecario podrá actualizar el estado del material Bibliográfico.
- El administrador podrá consultar el estado actual del material Bibliográfico.
- El administrador podrá generar una Orden de Compra.
- El administrador podrá bajas al material bibliográfico por motivo de *pérdida o el estado está deteriorado.*

1.4.5. Otras funciones del sistema

El sistema debe proporcionar funciones para la creación, modificación y eliminación de los roles del sistema y así los encargados de cada función

podrán autenticarse, así como para la asignación y modificación de dichos roles a los usuarios del sistema. En el sistema existirán aplicaciones que contienen módulos, los cuales, a su vez, tienen una serie de operaciones disponibles. Las aplicaciones, servicios y operaciones dentro de estos servicios tendrán asignados un conjunto de roles autorizados, de modo que sólo los usuarios que presenten alguno de dichos roles podrán hacer uso de la aplicación, servicio u operación. Así mismo, el sistema debe impedir que un usuario consulte información que no le concierne, es decir, un usuario sólo puede consultar los datos que le corresponda. Por supuesto, el sistema debe proporcionar funciones para la creación, modificación y eliminación de usuarios del sistema. Todas estas funciones sólo podrán ser realizadas por los administradores del sistema.

b) Modelo de Requisitos

El modelo de requisitos lo presento a continuación.

Casos de uso Inscripción al Lector. Estos casos de uso son.

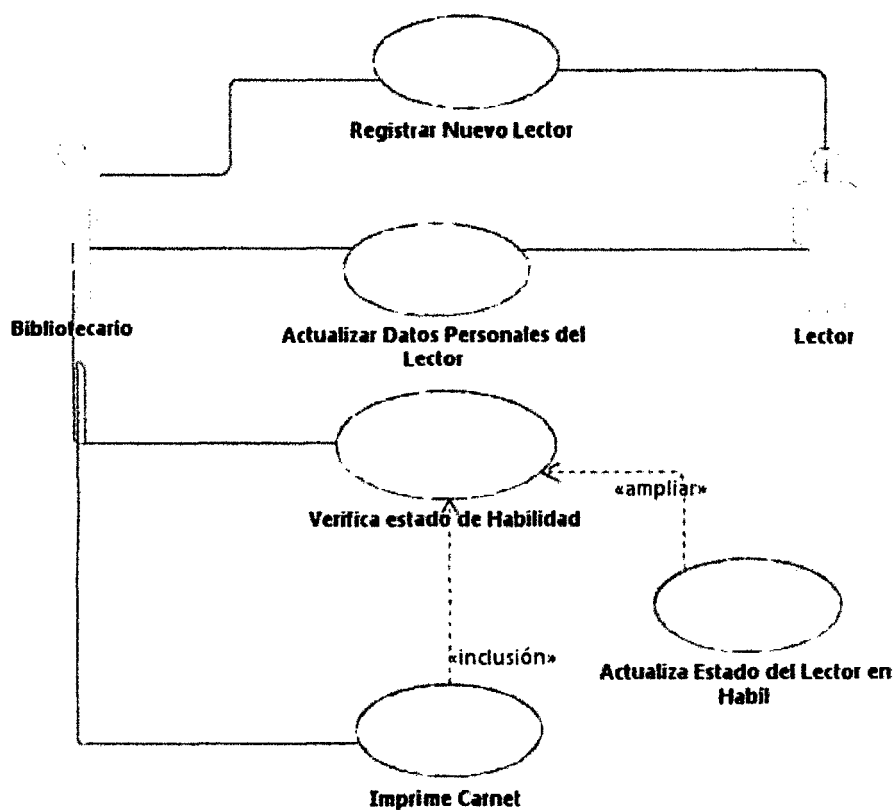


Ilustración 38: Casos de Uso Inscripción al Lector

Fuente: Elaboración Propia

Casos de uso Consulta Material Bibliográfico. Estos casos de uso son.

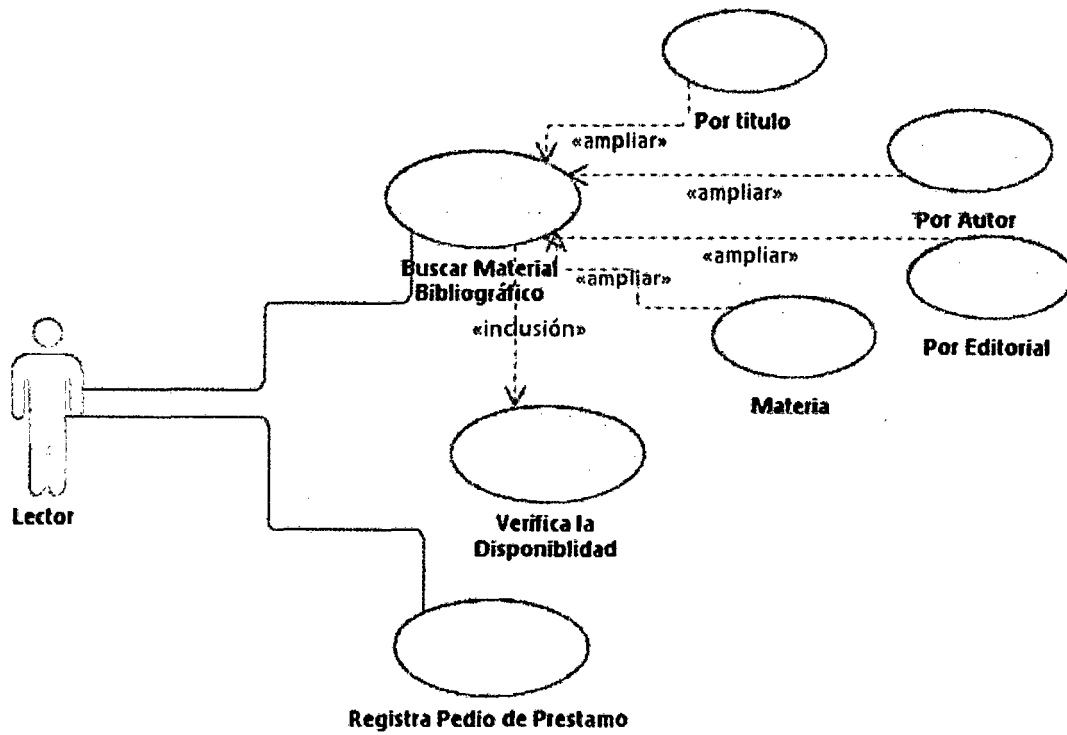


Ilustración 39: , Casos de Uso Consulta Material Bibliográfico

Fuente: Elaboración Propia

Casos de Uso Atención de Préstamos y Devoluciones. Estos casos de uso corresponden al proceso de atención de Préstamos y Devoluciones de Material Bibliográfico.

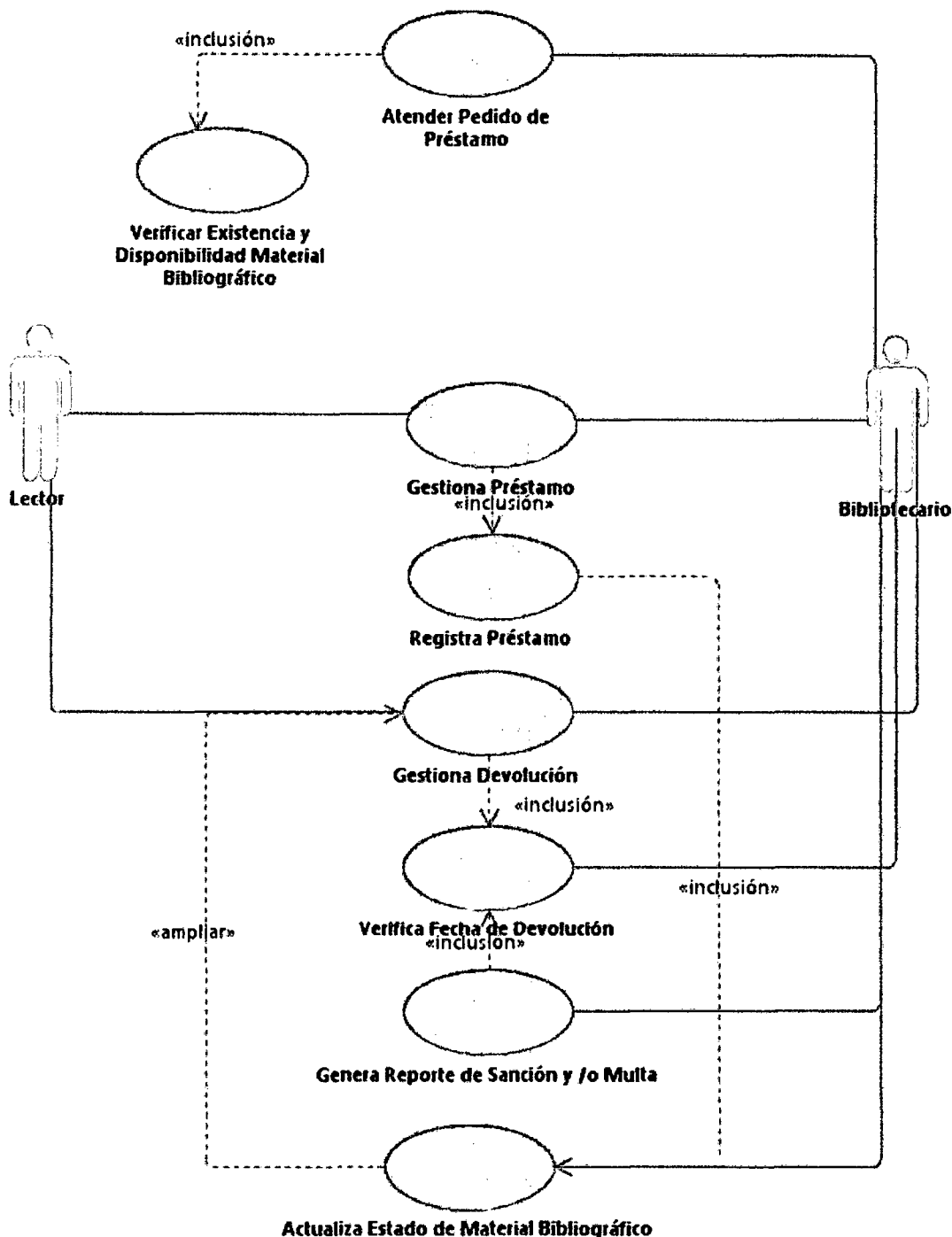


Ilustración 40: Atención de Préstamos y Devoluciones

Fuente: Elaboración Propia

Casos de Uso Administración. Estos casos de uso corresponden al proceso de administración del Material Bibliográfico.

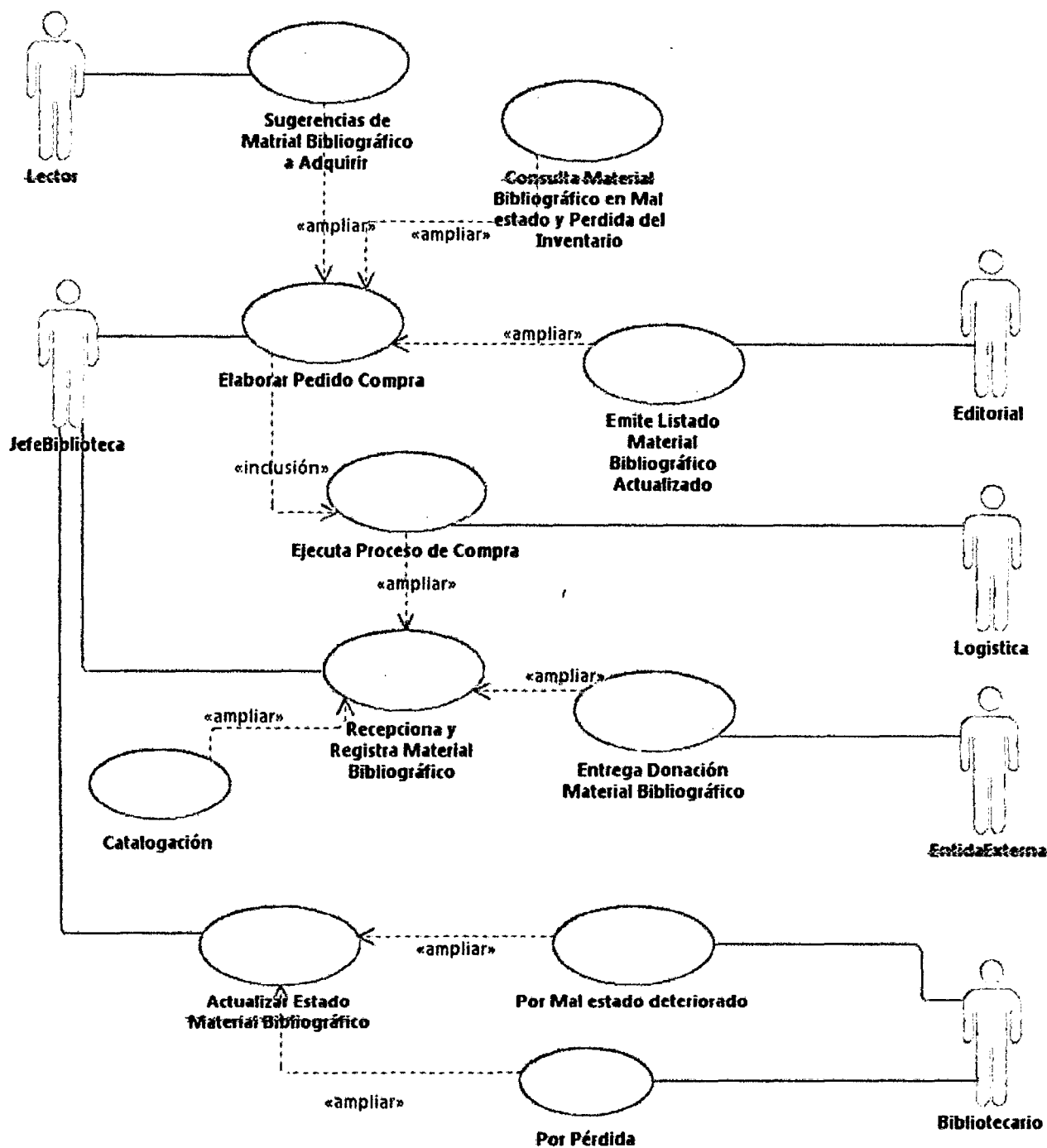


Ilustración 41: Casos de Uso Administración
Fuente: Elaboración Propia

Casos de Uso Seguridad. Estos casos de uso corresponden a la seguridad del sistema.

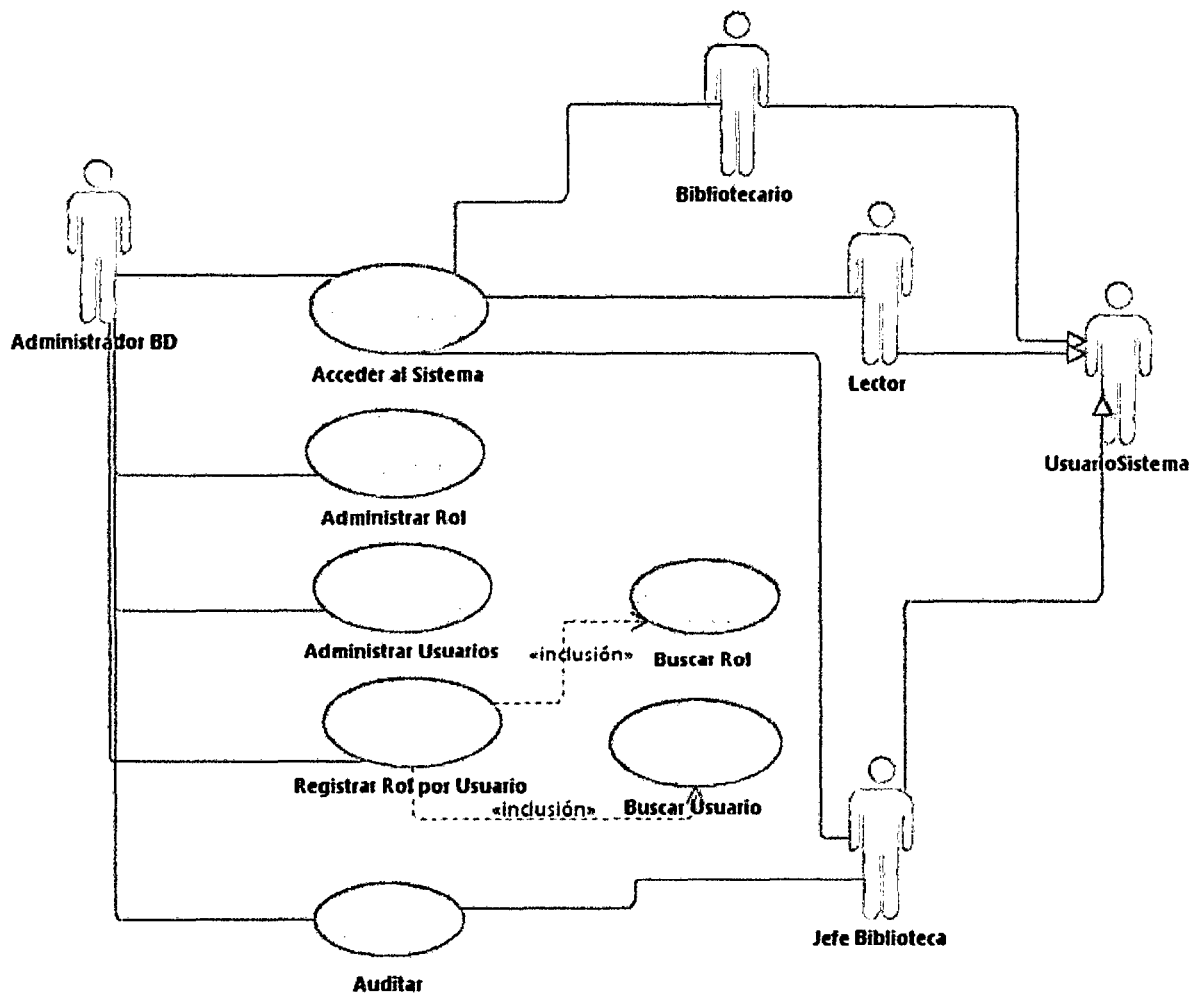


Ilustración 42: Casos de Uso Seguridad
Fuente: Elaboración Propia

c) Especificación de Casos de Uso

Detalle de caso de uso expandido

Un caso de uso expandido describe un proceso más a fondo que el de alto nivel. La diferencia básica con el caso de uso de alto nivel consiste en que tiene una sección destinada al curso normal de los eventos que los describe paso a paso.

| | | |
|--|--|---|
| Caso de Uso: | Registrar Nuevo Lector | |
| Actores: | Bibliotecario, Lector | |
| Propósito: | Obtener información exacta del Lector | |
| Resumen: | El lector se acerca a una de las terminales de la biblioteca, se le muestra por pantalla un formulario donde se le piden los datos personales (nombres, apellidos, fecha Nacimiento) | |
| Tipo: | Principal | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | |
| 1 | Lector accede al sistema de Biblioteca. | |
| 2 | Se le muestra un formulario donde se le pide todos los datos personales, de manera obligatoria cada uno. | |
| 3 | Lector o Bibliotecario llena el formulario. | |
| 4 | Lector o Bibliotecario "ENVIA" el formulario llenado. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 1.- El usuario introduce sus datos personales | | 3.- Respuesta del ingreso es <i>satisfactorio</i> |
| 2.- Si el lector no paga inscripción no puede realizar ninguna solicitud | | |
| Curso Alternativo | | |

4.- Actualiza la ventana de inicio para siguiente lector, si opera el bibliotecario.

Tabla 6: Caso de Uso Expandido: Registrar Nuevo Lector.

| | | |
|--|---|--|
| Caso de Uso: | Verifica estado de Habilidad | |
| Actores: | Bibliotecario | |
| Propósito: | Confirmar que el lector no tiene restricciones de ningún tipo, sanción o multa | |
| Resumen: | El Bibliotecario realiza una búsqueda del lector y verifica si no existen sanciones vigentes o multa, caso contrario se le comunica al lector o través de un mensaje por email se le notifica al lector su estado y la cancelación de solicitud de reinscripción. | |
| Tipo: | Principal | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | |
| 1 | El Bibliotecario usa los datos personales del Lector y comienza la búsqueda. | |
| 2 | Verifica el estado del Lector. | |
| 3 | Realiza un reporte del estado del lector, y se lo muestra a través de un mensaje por pantalla al Usuario y por correo electrónico. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 1.- El bibliotecario consulta al posible lector. | | 2.- búsqueda terminada |
| 3.- bibliotecario verifica si tiene sanciones o multas. | | 4.- muestra estado de sanciones y multas |
| 5.- bibliotecario permite la acción de la siguiente búsqueda | | |

Curso Alternativo

Tabla 7: Caso de Uso Expandido: Verifica estado de Habilidad.

| | | |
|---|--|---|
| Caso de Uso: | Registra Pedido de Préstamo | |
| Actores: | Lector | |
| Propósito: | Realizar la solicitud del Material Bibliográfico. | |
| Resumen: | El lector llega a la Biblioteca y accede a una de las terminales de consulta, en el sistema realiza el llenado del formulario, en el que se le solicita el Título y Autor del Material Bibliográfico, Tipo de Material Bibliográfico y el Código del Material Bibliográfico. | |
| Tipo: | Principal | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | |
| 1 | Se le muestra al lector por pantalla otro formulario de llenado con los datos precisos del Material Bibliográfico, una vez consultado. | |
| 2 | El lector llena el formulario con todos los datos del Material Bibliográfico (título, autor y tipo del Material Bibliográfico) y el código del material Bibliográfico. | |
| 3 | El usuario "ENVIA" el formulario llenado. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 2.- El lector introduce datos del material bibliográfico. | | 1.- genera el identificador de pedido de Préstamo donde se registran los datos del usuario. |
| 4.- El lector visualiza el estado de existencia. | | 3.- Envía reporte a lector mostrando estado de existencia |

Curso Alternativo

Tabla 8: Caso de Uso Expandido: Registra Pedido de Préstamo.

| | | |
|--|---|---|
| Caso de Uso: | Verifica existencia y disponibilidad de Material Bibliográfico | |
| Actores: | Bibliotecario | |
| Propósito: | Confirmar la existencia y disponibilidad del Material Bibliográfico. | |
| Resumen: | El Bibliotecario de Material Bibliográfico usando los datos de la pedido de préstamo realiza la búsqueda del Material Bibliográfico, con esta confirmación realiza un reporte que se le muestra por pantalla al Usuario el detalle (existencia y disponibilidad). | |
| Tipo: | Principal | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | Categoría |
| 1 | El Responsable del Material Bibliográfico realiza la búsqueda del Material bibliográfico usando los datos especificados en el formulario. | |
| 2 | Verifica el estado del Material Bibliográfico. | |
| 3 | Realiza un reporte detallado, envía el reporte de estado al Usuario a través de un mensaje por pantalla. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 1.- El bibliotecario selecciona el pedido de préstamo. | | 2.- Brinda reporte de las disposiciones de los materiales bibliográficos 3.- Envía notificación a usuario mostrando estado de existencia |

Curso Alternativo

Tabla 9: Caso de Uso Expandido: verifica existencia y disponibilidad de Material Bibliográfico.

| | | |
|---|--|--|
| Caso de Uso: | Gestionar Préstamo | |
| Actores: | Lector, Bibliotecario | |
| Propósito: | Generar el préstamo del Material Bibliográfico | |
| Resumen: | El Bibliotecario ubica el material solicitado por el pedido de préstamo. El bibliotecario le solicita al lector su carnet, el Bibliotecario verifica los datos del lector, luego realiza el préstamo entregando el Material Bibliográfico al lector. El usuario se marcha con el material. | |
| Tipo: | Principal | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | |
| 1 | El Bibliotecario realiza la búsqueda del Material Bibliográfico en estantes. | |
| 2 | El Bibliotecario pide documento personal y Carnet. | |
| 3 | El Lector hace entrega de los documentos. | |
| 4 | El Bibliotecario Recepciona los documentos. | |
| 5 | El Bibliotecario hace la entrega del Material Bibliográfico. | |
| 6 | El Bibliotecario ingresa los datos del Material Bibliográfico prestado a través del pedido de préstamo. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 1.- El bibliotecario consulta pedido de préstamo. | | 2.- Búsqueda de material y ejemplar según tipo |

| | |
|-------------------|--|
| | 3.- Envía reporte de a lector mostrando estado de existencia, código de libro y titulo |
| Curso Alternativo | |

Tabla 10: Caso de Uso Expandido: Gestionar Préstamo

| | | |
|---|--|--|
| Caso de Uso: | Registrar Préstamo | |
| Actores: | Bibliotecario | |
| Propósito: | Realizar un registro detallado del préstamo. | |
| Resumen: | El bibliotecario registra a través de un formulario, usando los datos de la solicitud pedido de préstamo de Material Bibliográfico, realiza un registro minucioso con los datos obtenidos. | |
| Tipo: | Esencial | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | |
| 1 | El bibliotecario accede al sistema. | |
| 2 | El bibliotecario cambia el estado del Material Bibliográfico con los datos obtenidos (título o autor o materia). | |
| 3 | Con la fecha actual genera automáticamente una fecha exacta de devolución. | |
| 4 | Realiza un reporte con la fecha de devolución. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 1.- El lector entrega garantías al bibliotecario. | | 3.- El sistema genera la fecha de devolución del material bibliográfico. |
| 2.- El bibliotecario registra que garantías está recibiendo y | | 4.- El sistema actualiza en el estado |

| | |
|---|--------------------------------------|
| paralelamente confirma el pedido de préstamo. | del material bibliográfico a prestar |
| Curso Alternativo | |

Tabla 11: Caso de Uso Expandido: Registrar Préstamo

| | | |
|---------------------------|---|--|
| Caso de Uso: | Gestionar Devolución | |
| Actores: | Lector, Bibliotecario | |
| Propósito: | Finalizar el proceso de préstamo. | |
| Resumen: | El lector llega con el Material Bibliográfico prestado a la Biblioteca, el Bibliotecario verifica el préstamo y registra la devolución en el sistema, el sistema verifica las reglas de préstamo para el usuario, si existen faltas genera un reporte automáticamente, dependiendo de esta acción el Bibliotecario devuelve las garantías o no. | |
| Tipo: | Principal | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | |
| 1 | El lector entrega el Material Bibliográfico prestado. | |
| 2 | El Bibliotecario recibe el Material Bibliográfico | |
| 3 | El Bibliotecario verifica la fecha de préstamo y la fecha devolución anteriormente asignada. | |
| 4 | Dependiendo de la fecha de devolución asignada y la actual, se realiza la entrega de los documentos dejados como garantía. | |
| 5 | El bibliotecario ingresa los datos del Material Bibliográfico devuelto, en base al registro de | |

| | | |
|---|-----------|--|
| | préstamo. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 1.-El bibliotecario recibe la entrega palpable del material bibliográfico prestado. | | 2.- el sistema imprime un reporte donde se encuentra nombre de lector, código del material bibliográfico, título, fecha de préstamo y fecha de devolución. |
| 3.- El bibliotecario hace la entrega del reporte impreso al lector. | | |
| Curso Alternativo | | |

Tabla 12: Caso de Uso Expandido: Gestionar Devolución.

| | | |
|---------------------------|---|--|
| Caso de Uso: | Verificar fecha de Devolución | |
| Actores: | Bibliotecario, Lector | |
| Propósito: | Confirmar que la fecha de devolución coincide con la fecha de devolución asignada por el sistema o está dentro de los límites. | |
| Resumen: | El Bibliotecario realiza la búsqueda de la fecha de devolución asignada al Usuario, al confirmar la coincidencia o no de las fechas realiza un reporte y se lo muestra por pantalla al bibliotecario. | |
| Tipo: | Esencial | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | |
| 1 | Bibliotecario accede al sistema. | |
| 2 | Realiza la búsqueda de la fecha asignada por sistema al lector | |
| 3 | Confirma la coincidencia o no coincidencia de fechas. | |
| 4 | Realiza un reporte de sanción o multa y se lo | |

| | | |
|---|--|---|
| | muestra por pantalla al Bibliotecario. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 1.- El bibliotecario pide al sistema el reporte de préstamo del lector. 4.- El bibliotecario realiza la devolución de las garantías y gestiona la sanción, que puede o no realizarse | | 3.- el sistema responde con la siguiente información: fecha de préstamo y devolución y/o el estado generado en el atributo de sanciones o multas. |
| Curso Alternativo | | |

Tabla 13: Caso de Uso Expandido: Verificar fecha de Devolución.

| | | |
|---------------------------------------|--|---------------------------------------|
| Caso de Uso: | Actualizar estado de Material Bibliográfico | |
| Actores: | Bibliotecario | |
| Propósito: | Mantener el control detallado y la información del Material Bibliográfico | |
| Resumen: | El Bibliotecario accede al listado de material Bibliográfico y cambia el estado de este. | |
| Tipo: | Esencial | |
| REFERENCIA CRUZADA | | |
| Nº Ref. | Condición | |
| 1 | El Responsable de Material Bibliográfico accede al sistema | |
| 2 | Ingresa al listado de Material Bibliográfico | |
| 3 | Cambia el estado del Material Bibliográfico de prestado o ya devuelto como corresponde. | |
| CURSO DE LOS EVENTOS | | |
| Acción del actor | | Respuesta del Sistema |
| 1.- El bibliotecario envía al sistema | | 2.- el sistema informa al responsable |

| | |
|-------------------|--|
| código de libro | de material bibliográfico que ha actualizado al estado actual del material bibliográfico prestado o devuelto |
| Curso Alternativo | |

Tabla 14: Caso de Uso Expandido: Actualizar estado de Material Bibliográfico.

3.3. FASE II: ELABORACIÓN

3.3.1. ANÁLISIS

3.3.1.1. Diagrama de Análisis de Clases.

➤ INTERFACES

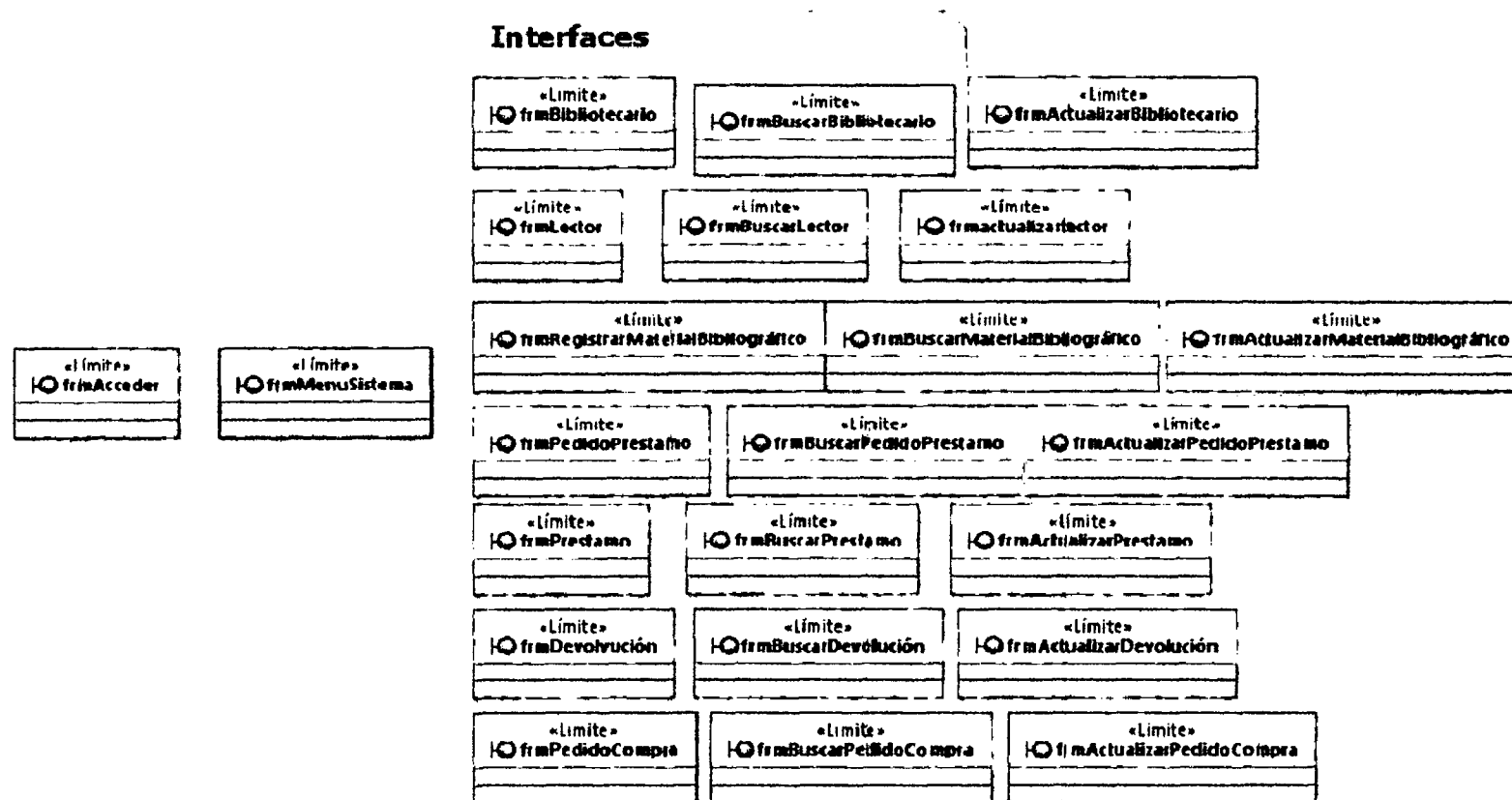


Ilustración 43: Diagrama de Interfaces
Fuente: Elaboración Propia

➤ **Controles**

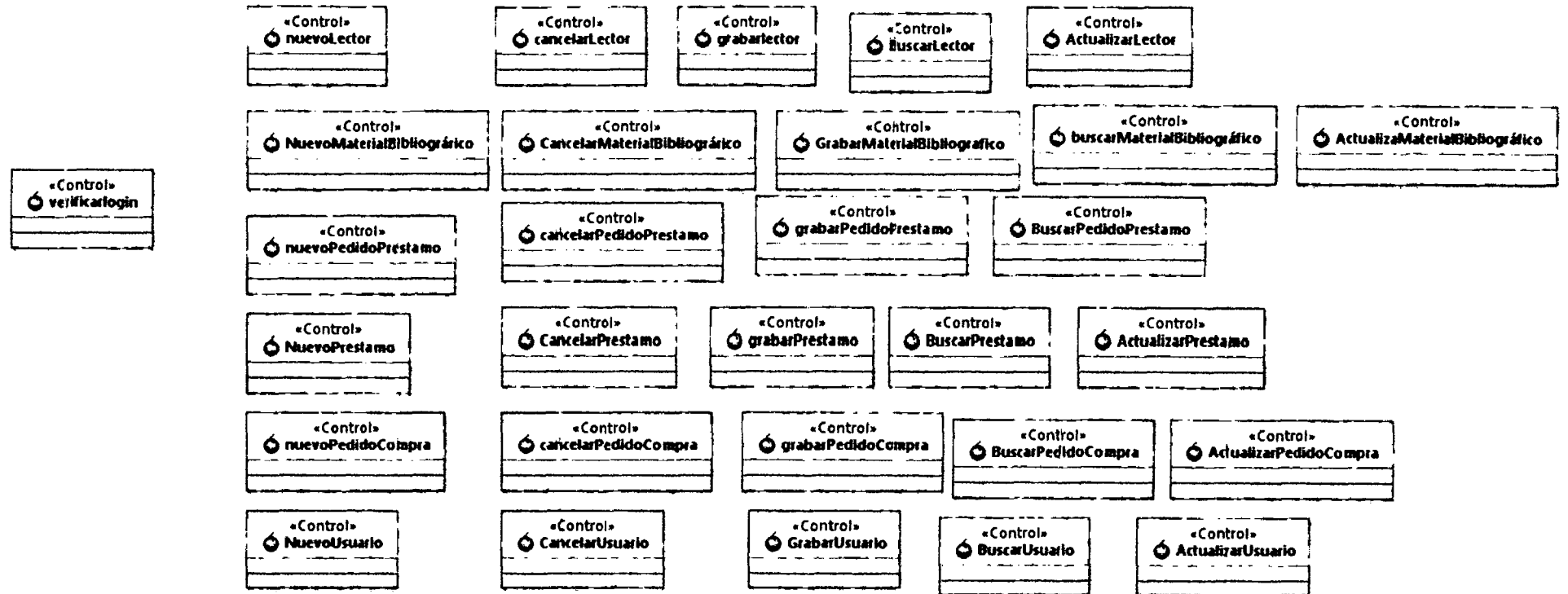


Ilustración 44: Diagrama de Controles
Fuente: Elaboración Propia

3.3.1.2. Diagramas de Secuencia

a) Inscripción al Lector

1. DS Registrar Nuevo Lector

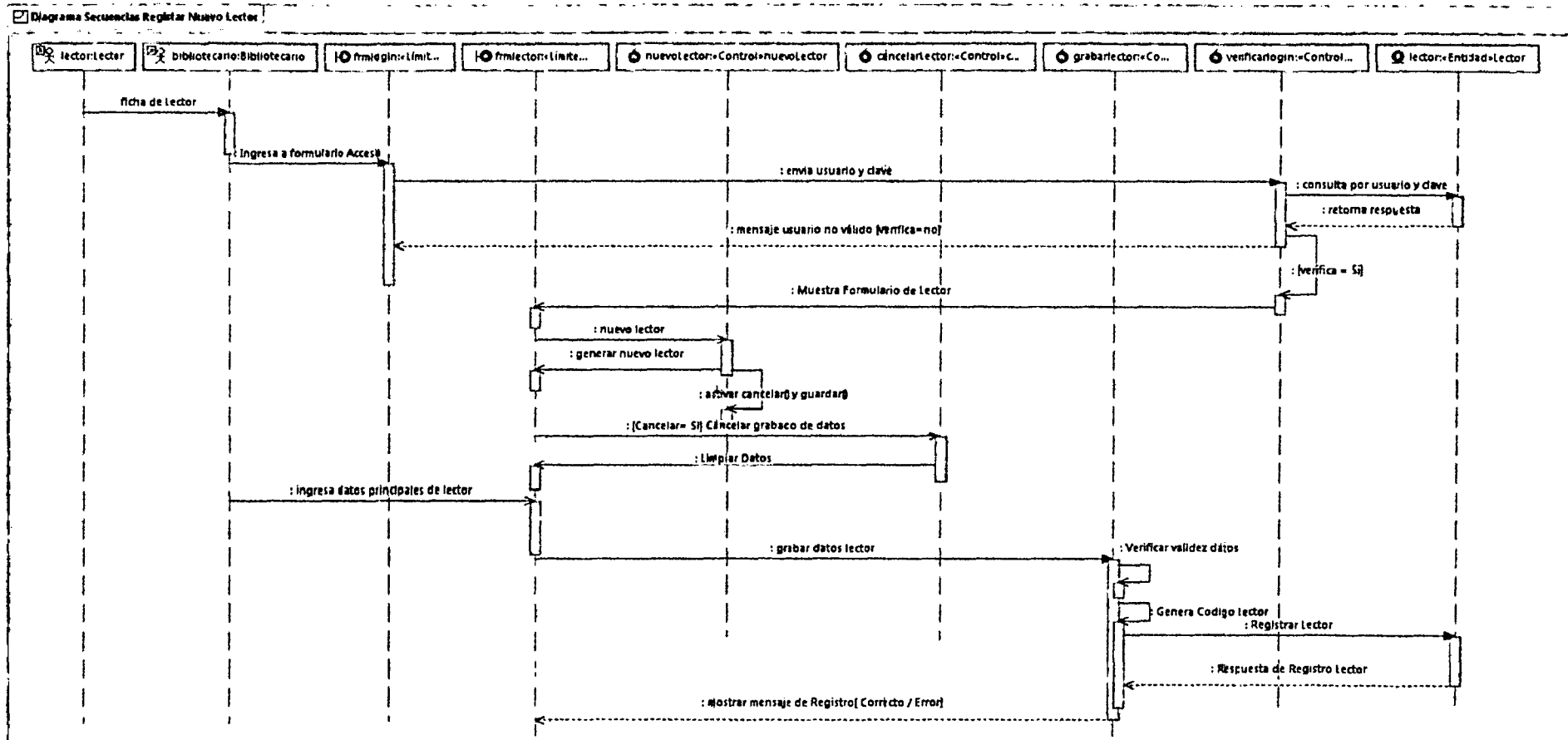


Ilustración 45: Diagrama de Secuencia, Inscribir nuevo lector

Fuente: Elaboración Propia

2. DS Actualizar Datos Personales del Lector

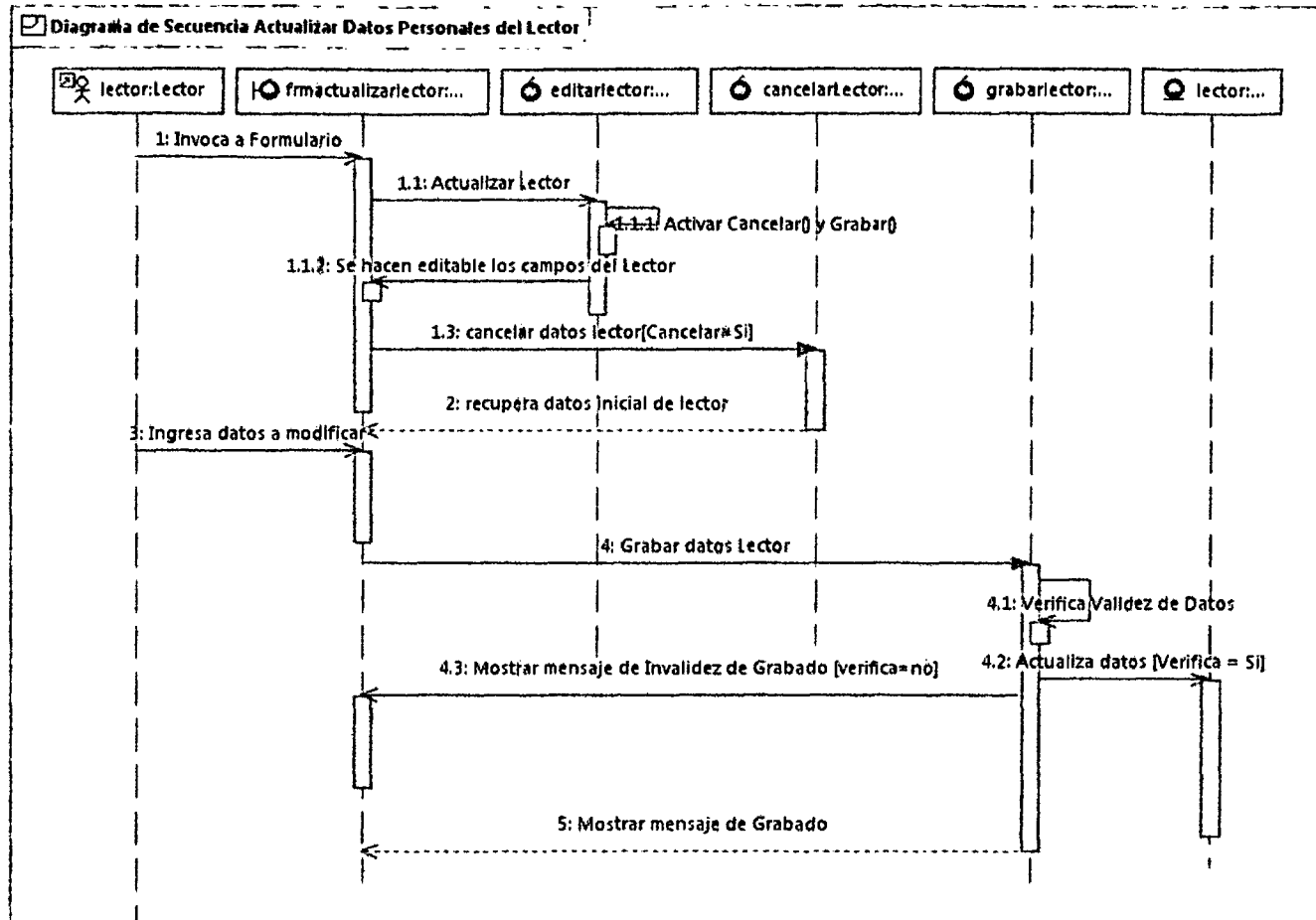


Ilustración 46: Diagrama de Secuencia, Actualizar Datos Personales del Lector

Fuente: Elaboración Propia

3. DS Verifica Estado de Habilidad

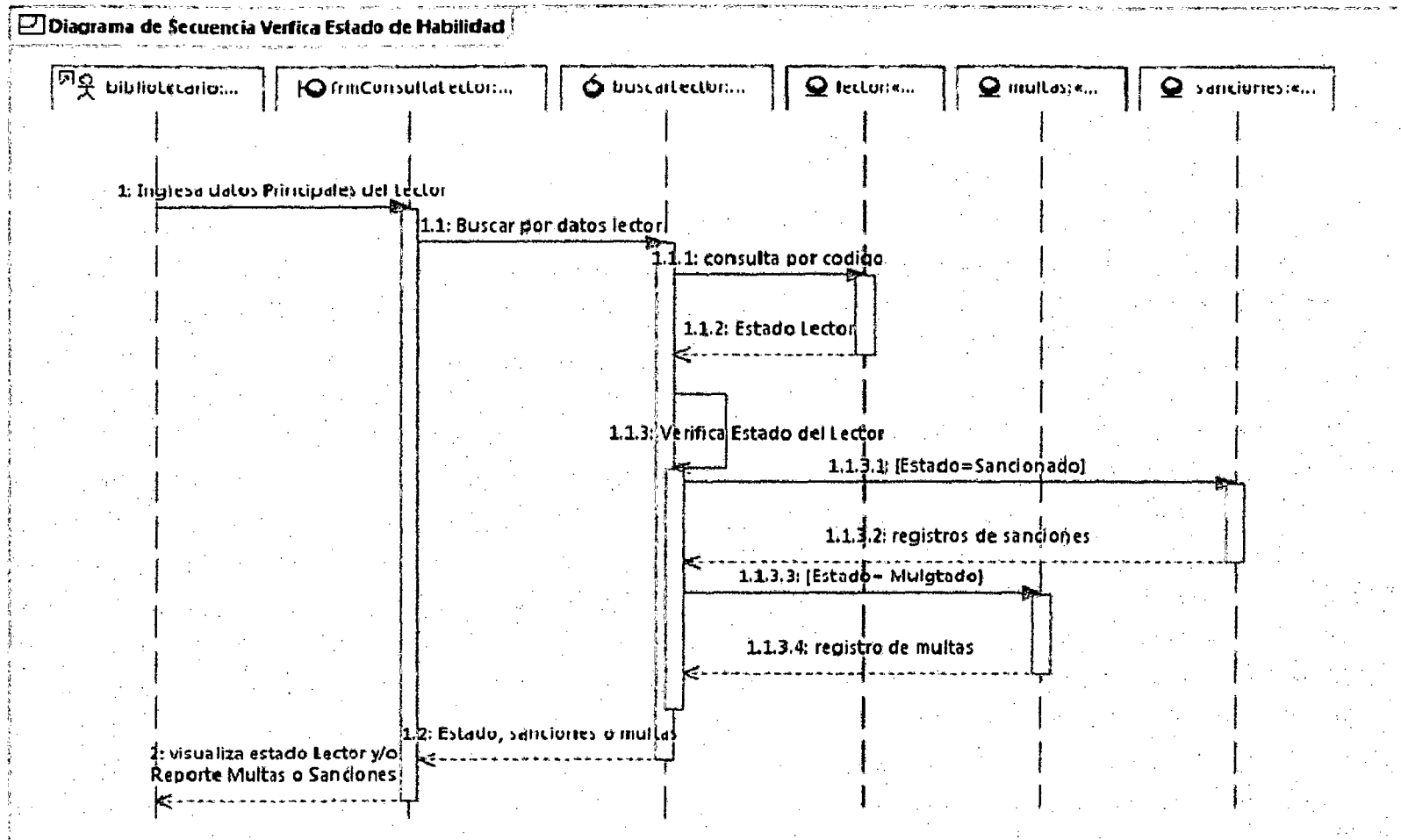


Ilustración 47: Diagrama de Secuencia, Verificar Estado de Habilidad
Fuente: Elaboración Propia

b) Atención de Consulta de Material Bibliográfico

1. DS Buscar Material Bibliográfico

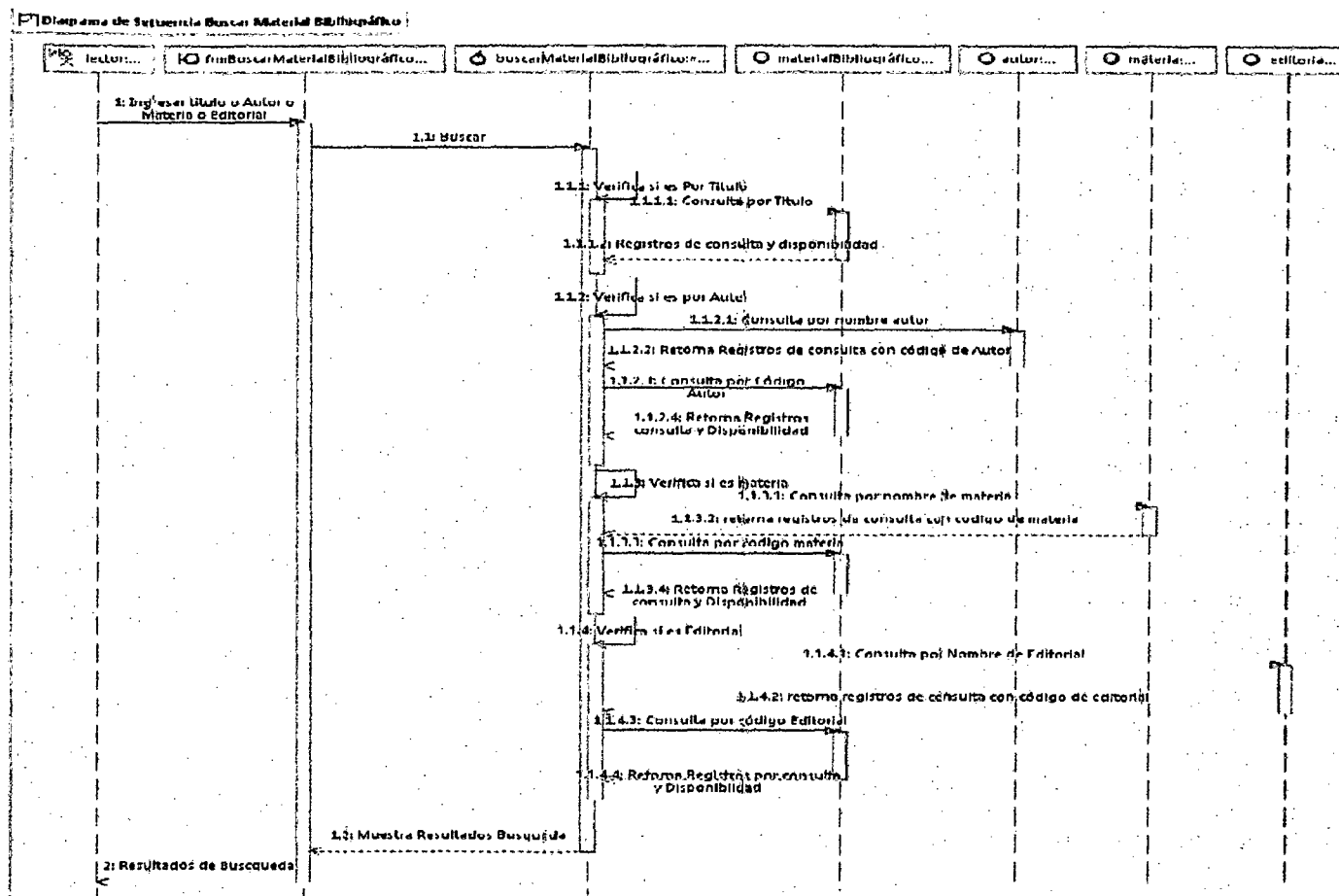


Ilustración 48: Diagrama de Secuencia, Consultar Material Bibliográfico
Fuente: Elaboración Propia

2. DS Registra pedido de Préstamo

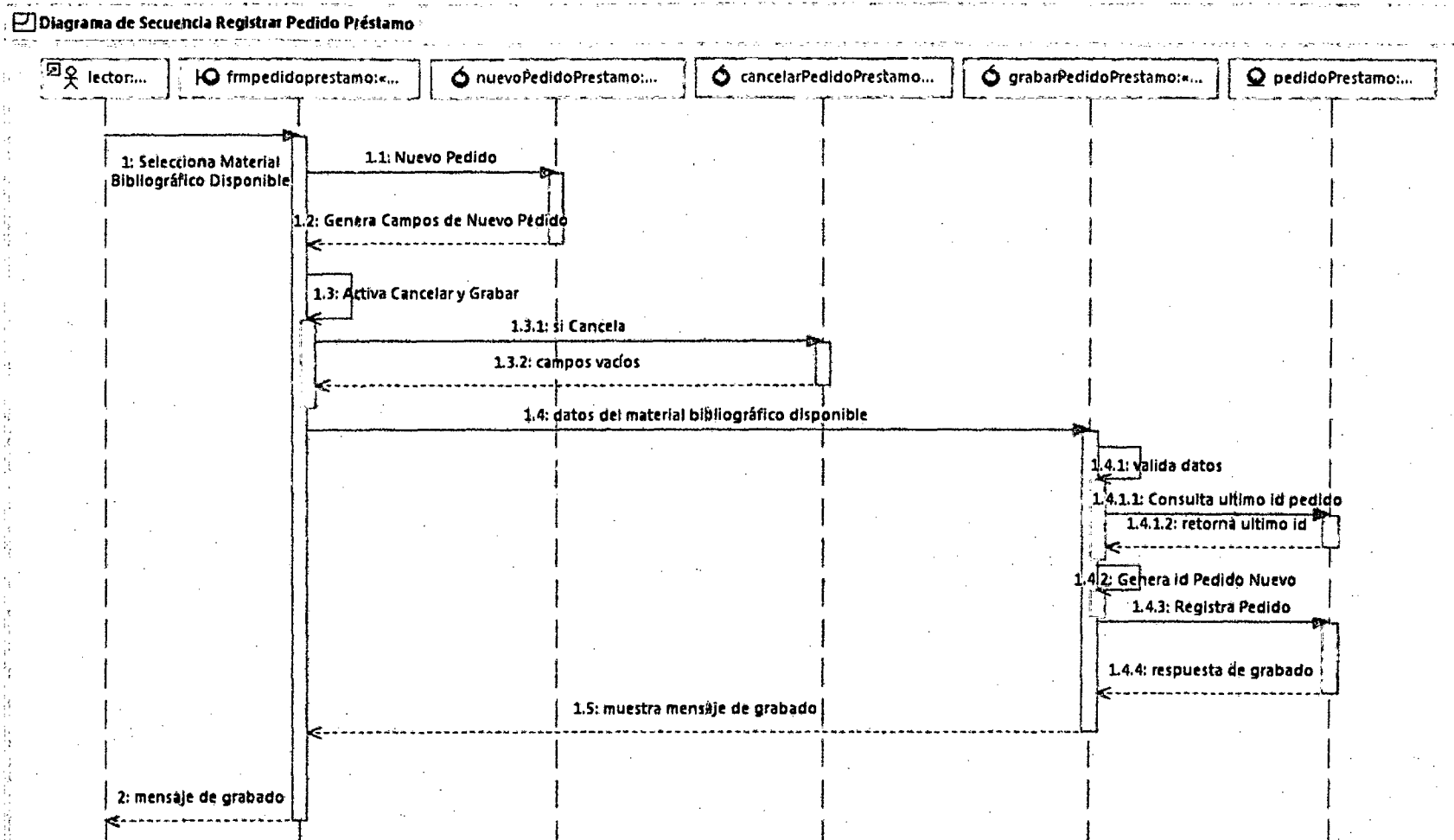


Ilustración 49: Diagrama de Secuencia, Registra pedido Préstamo
Fuente: Elaboración Propia

c) Atención de Préstamos y Devoluciones

1. DS Atender pedido de Préstamo

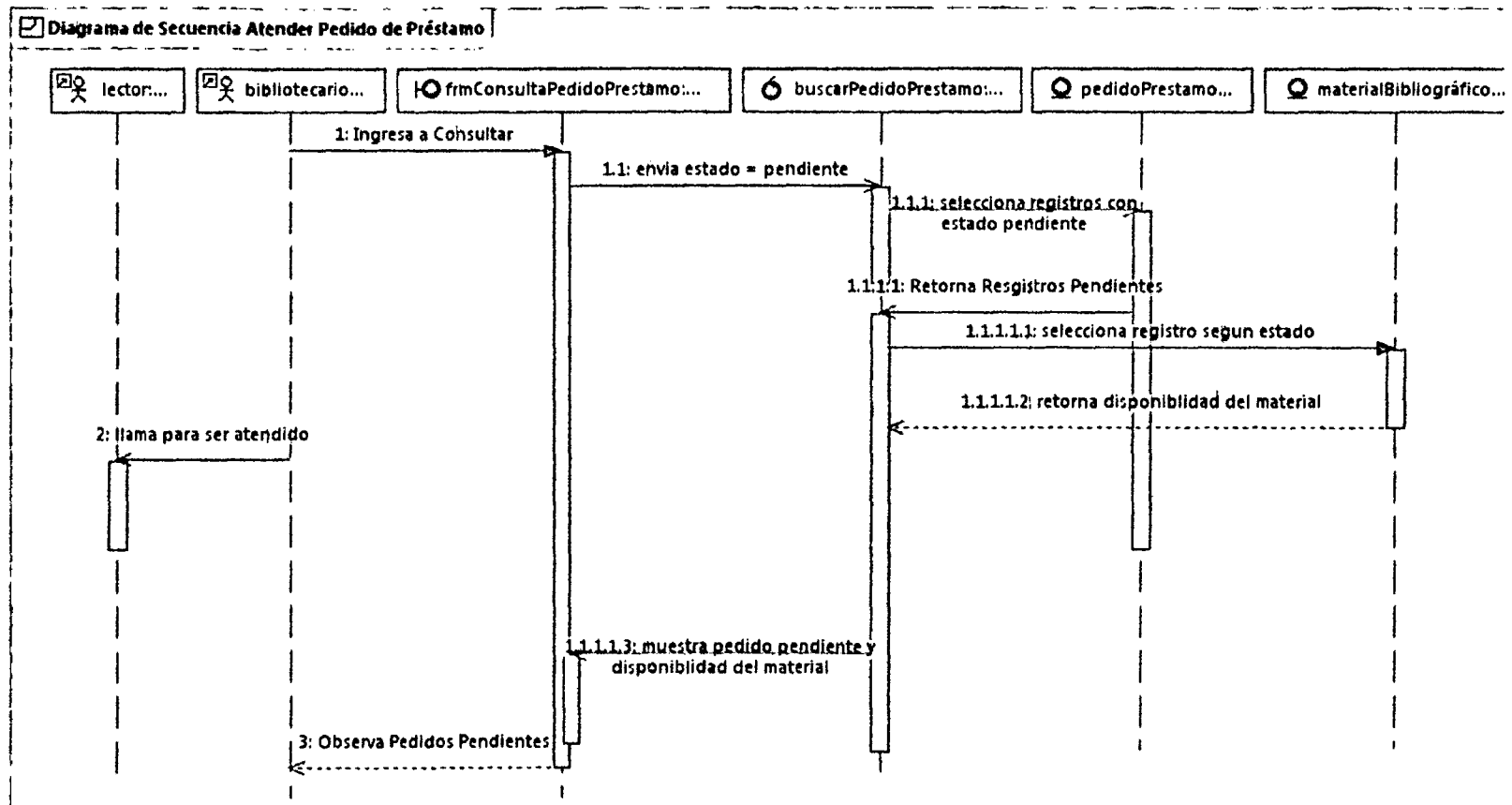


Ilustración 50: Diagrama de Secuencia, Atender pedido de Préstamo

Fuente: Elaboración Propia

2) DS Gestionar y Registro de Préstamo

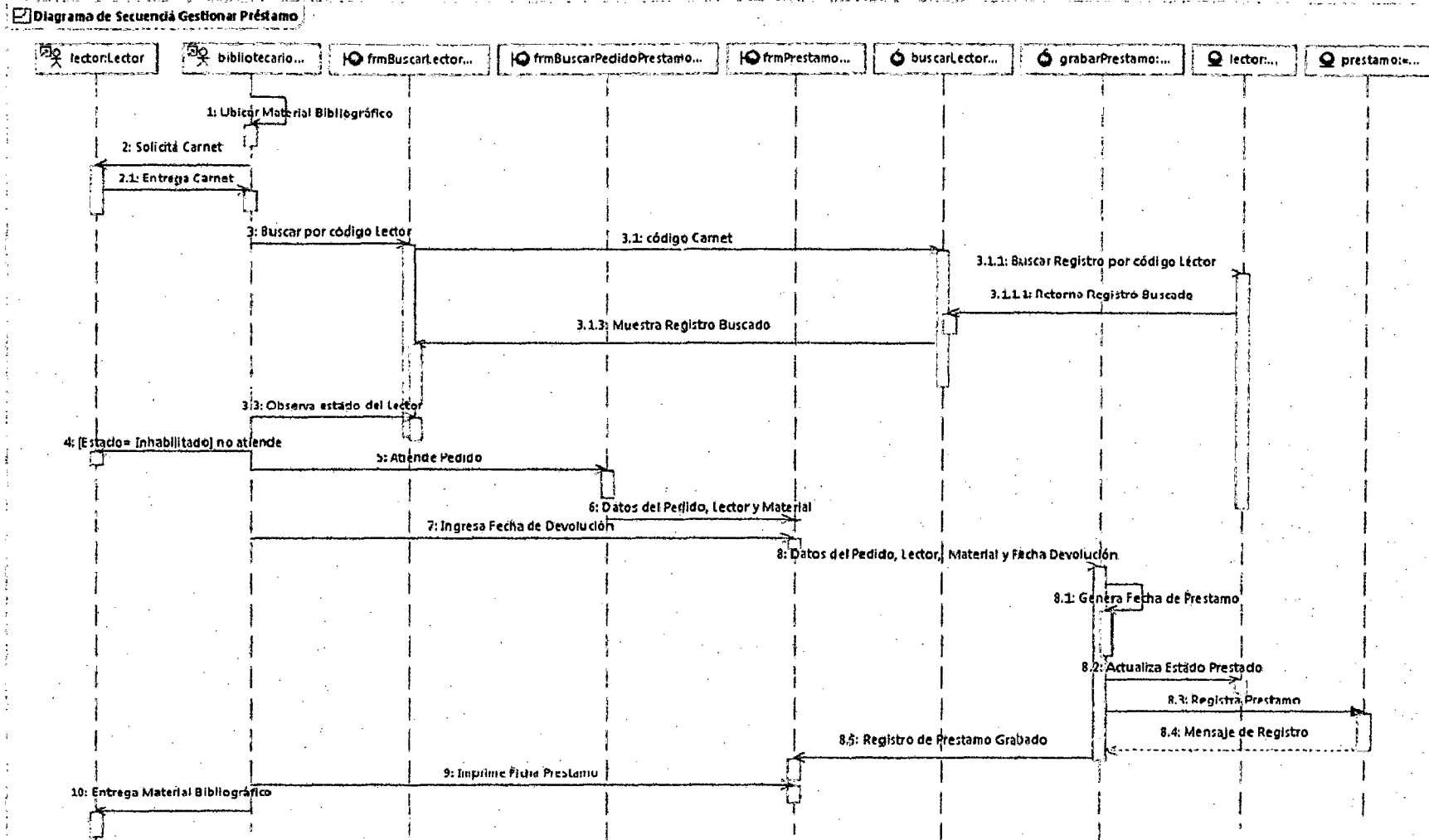


Ilustración 51: Diagrama de Secuencia, Gestionar y Registrar Préstamo

Fuente: Elaboración Propia

3) DS Gestionar Devolución

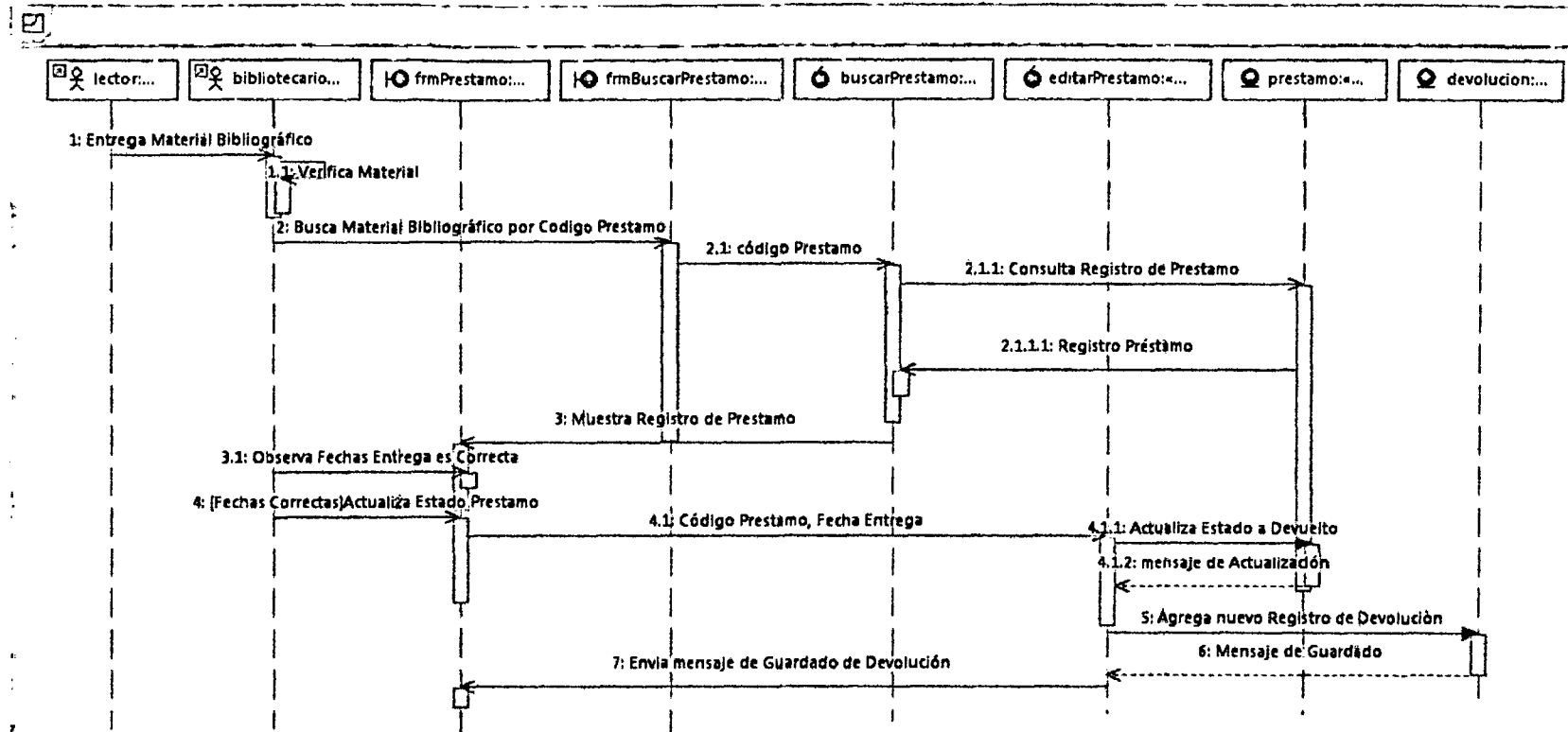


Ilustración 52: Diagrama de Secuencia, Gestionar Devoluciones

Fuente: Elaboración Propia

d) Administración de Material Bibliográfico.

1. DS Elaborar Pedido de Compra

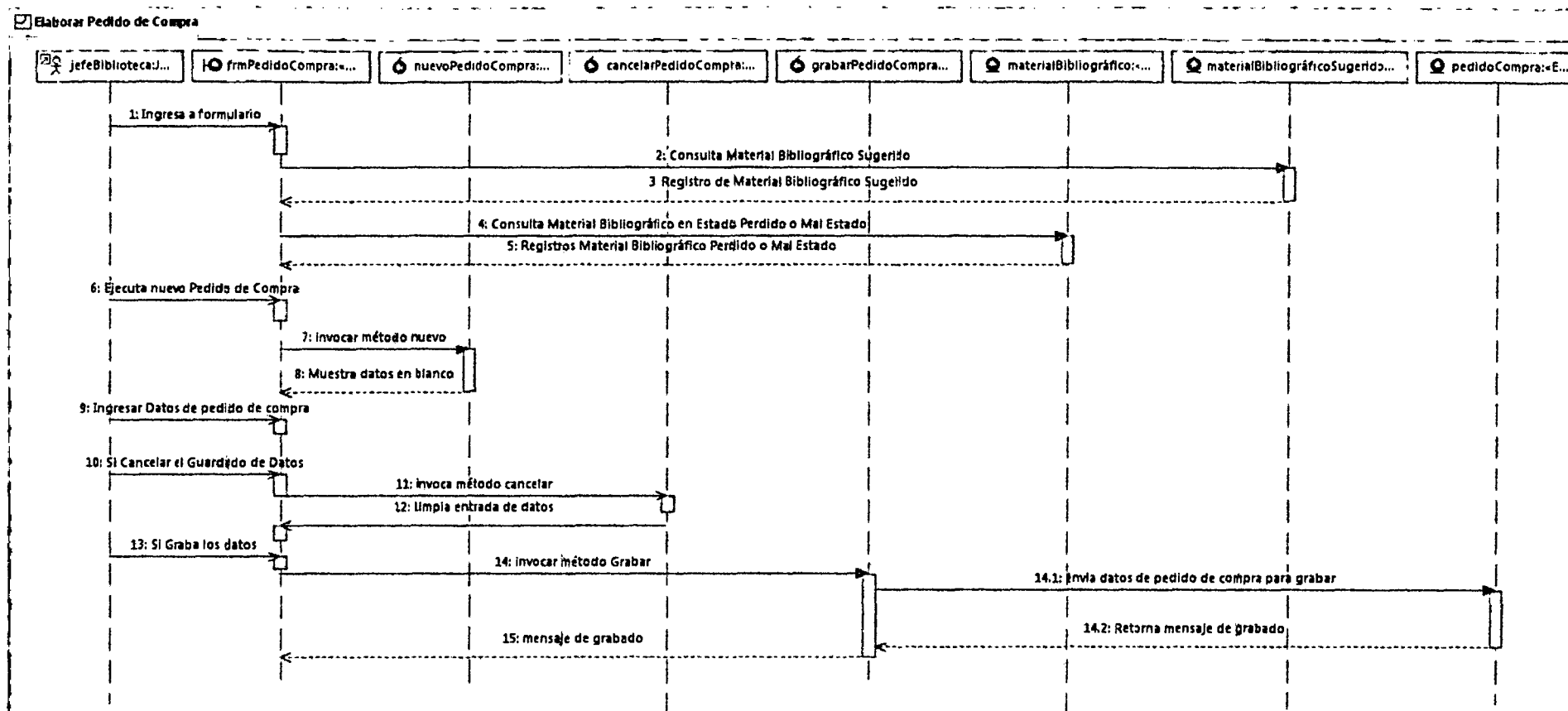


Ilustración 53: Diagrama de Secuencia, Elaborar Pedido de Compra

Fuente: Elaboración Propia

2. DS Recepciona y Registra Material Bibliográfico.

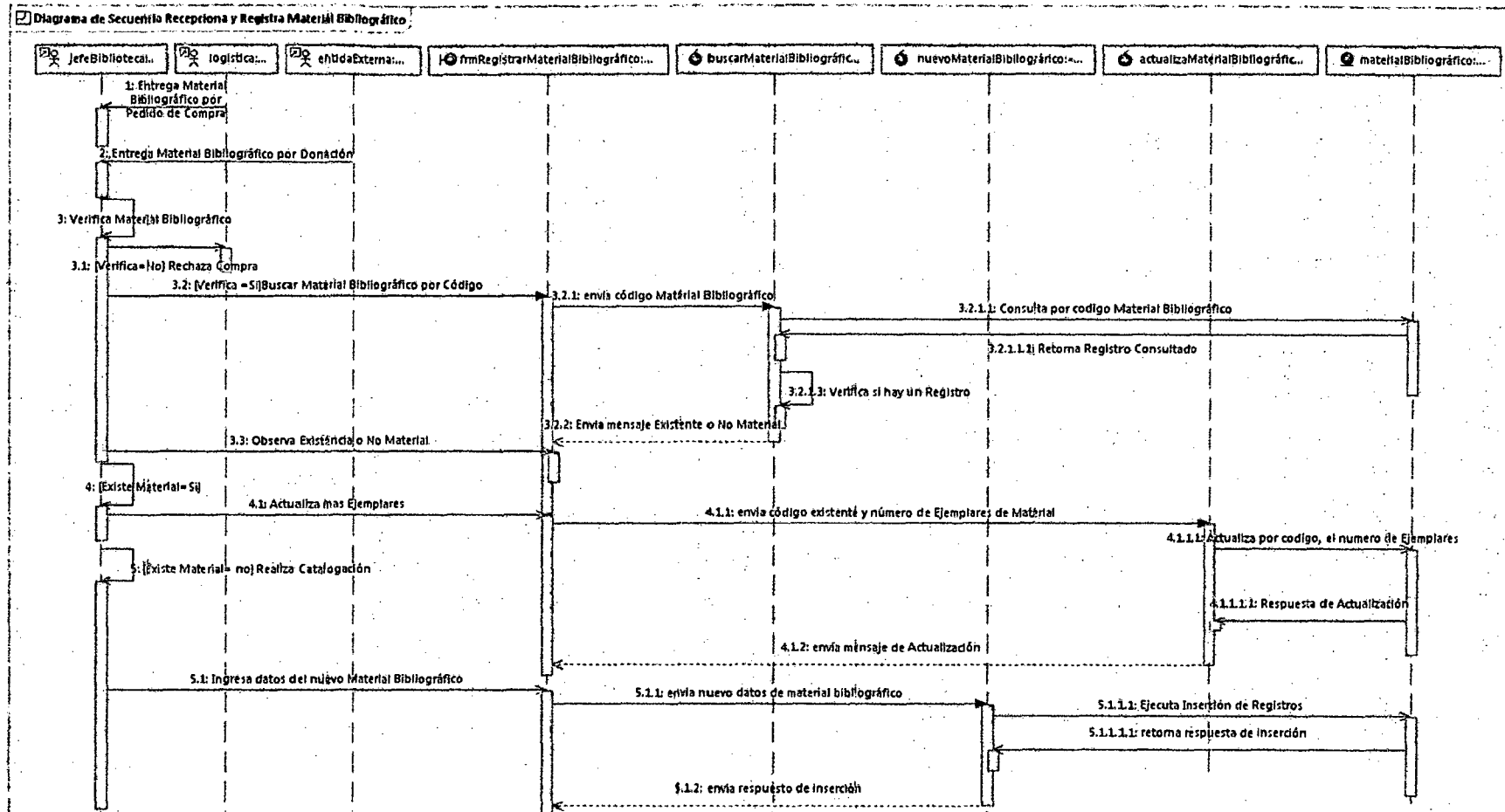


Ilustración 54: Diagrama de Secuencia, Recepciona y Registra Material Bibliográfico

Fuente: Elaboración Propia

d. Seguridad

1. DS – Acceder al Sistema: Ingresar, Cancelar

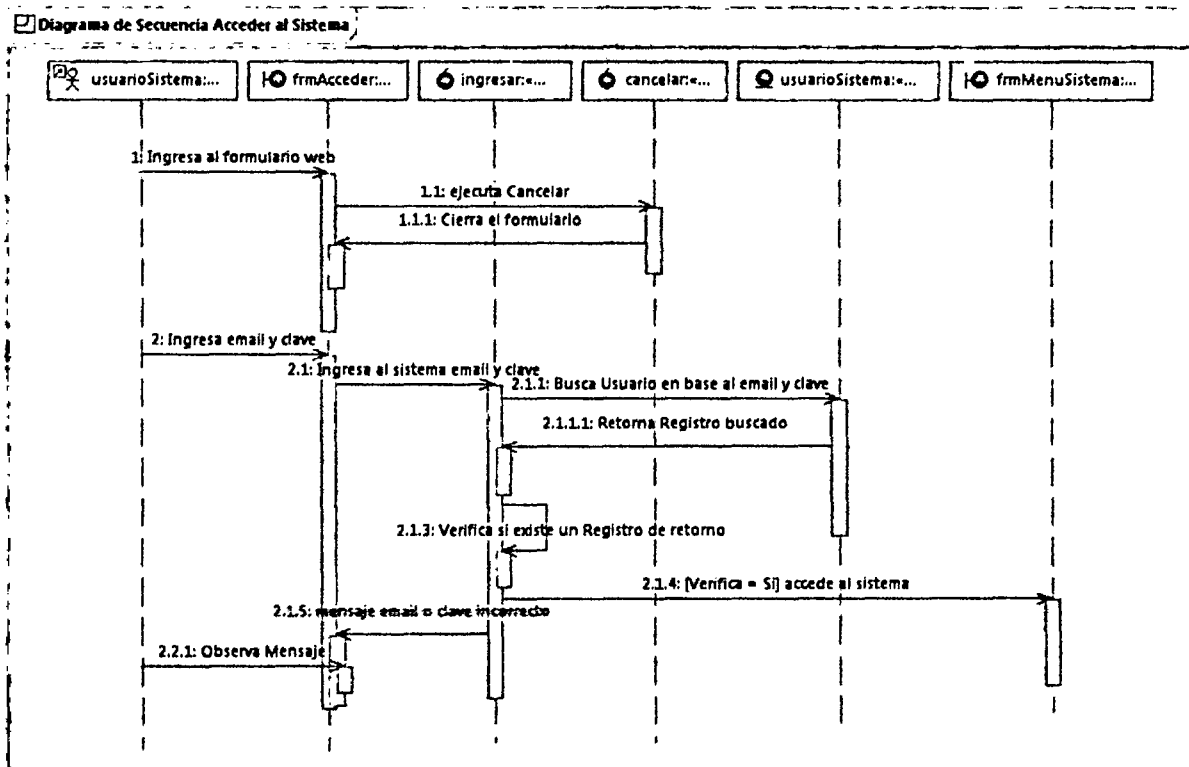


Ilustración 55: Diagrama de Secuencia, Acceder al Sistema

Fuente: Elaboración Propia

2. DS – Auditoria

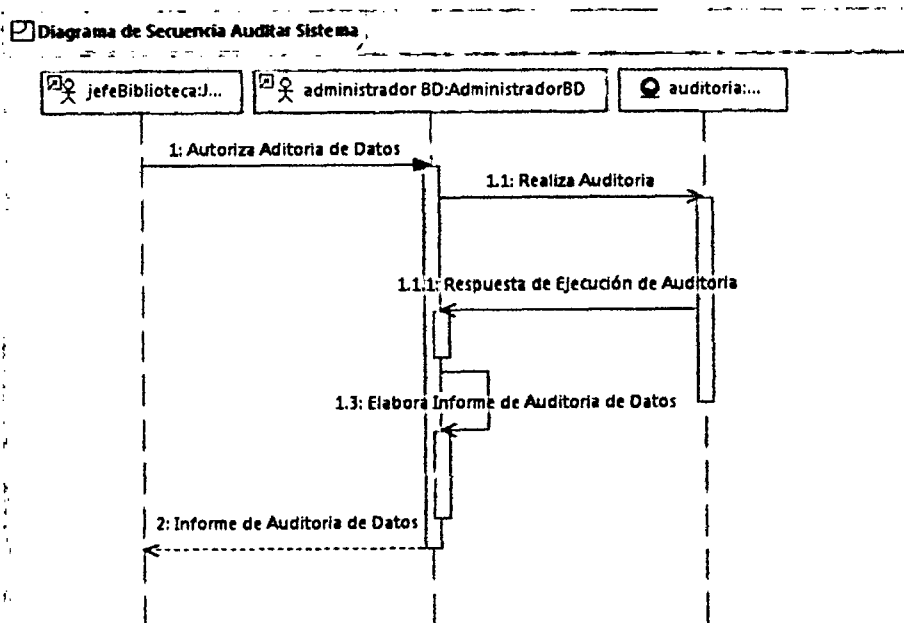


Ilustración 56: Diagrama de Secuencia, Auditoria

Fuente: Elaboración Propia

3.3.1.3. Diagrama de Clases

Diagrama de Clases de Inscripción al lector, Consulta y Atención de Préstamos y Devoluciones

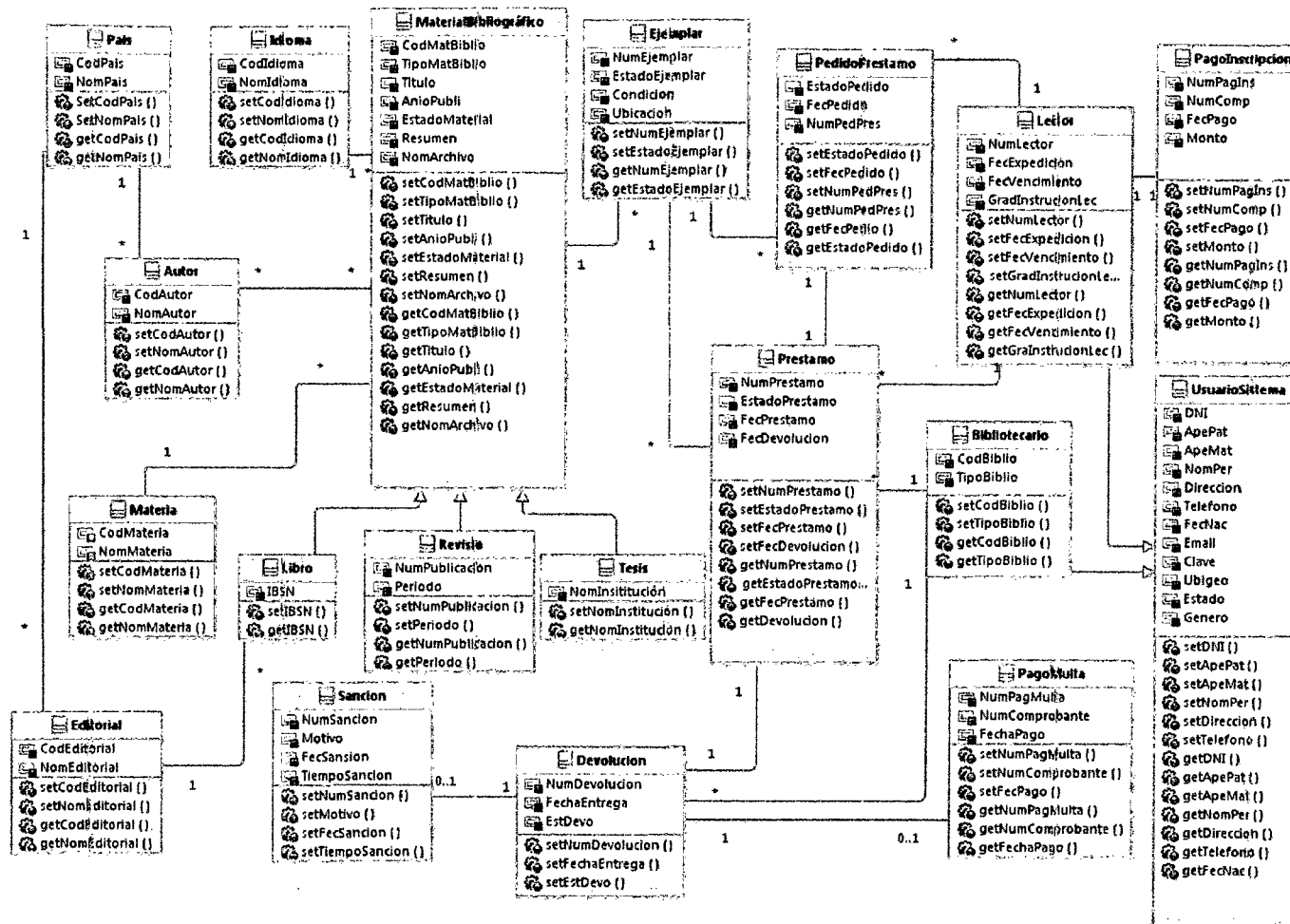


Ilustración 57: Diagrama de Clases
Fuente: Elaboración Propia

3.4. FASE III: CONSTRUCCIÓN

3.4.1. Diseño del Sistema

3.4.1.1. Introducción.

En esta fase se realiza una descripción del proyecto con un mayor nivel de detalle, teniendo en cuenta las características de un entorno Web.

A partir de los modelos obtenidos en la fase anterior, análisis, se plantea como llevar a cabo la implementación de la aplicación Web pero, sin entrar en los detalles de una tecnología concreta. Definiendo la arquitectura que define los principales componentes de la aplicación en el contexto de un entorno Web.

3.4.1.2. Diagrama de Paquetes de Diseño.

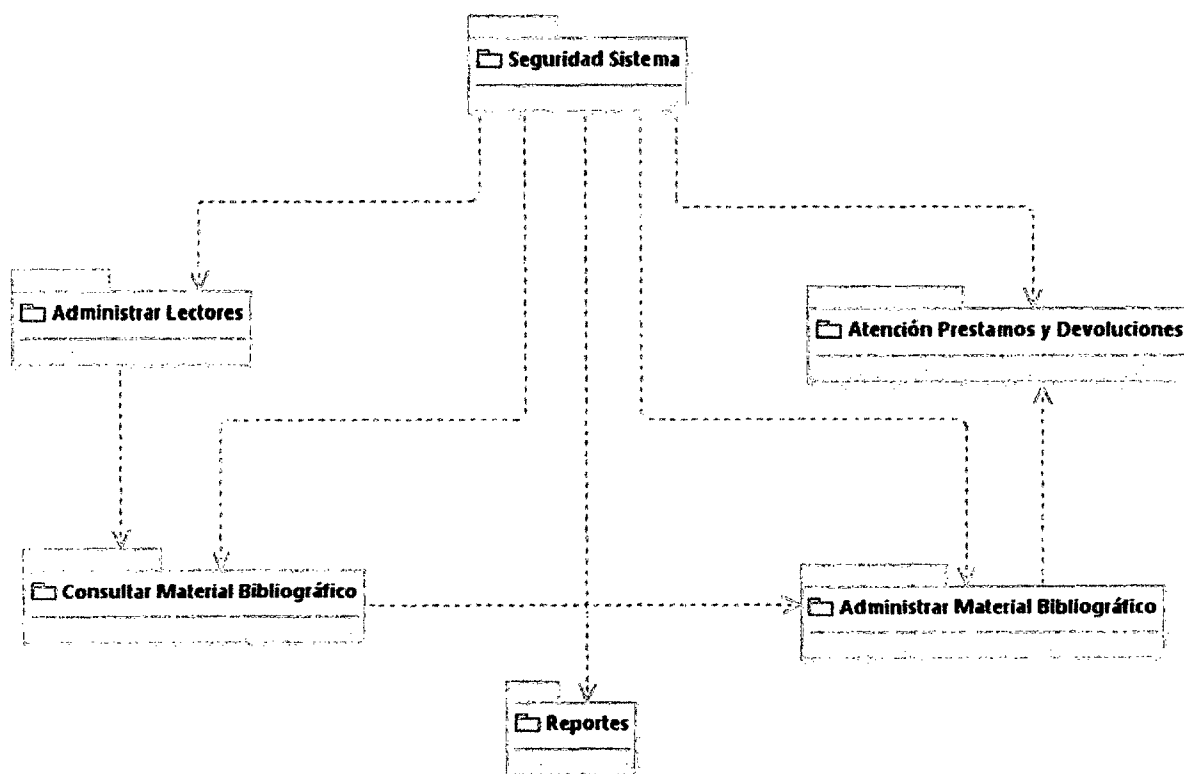


Ilustración 58: Diagrama de Paquetes de Diseño
Fuente: Elaboración Propia

3.4.1.2. Diagrama de Clases de Diseño.

a) Diagrama de Clases de Diseño para Seguridad Sistema.

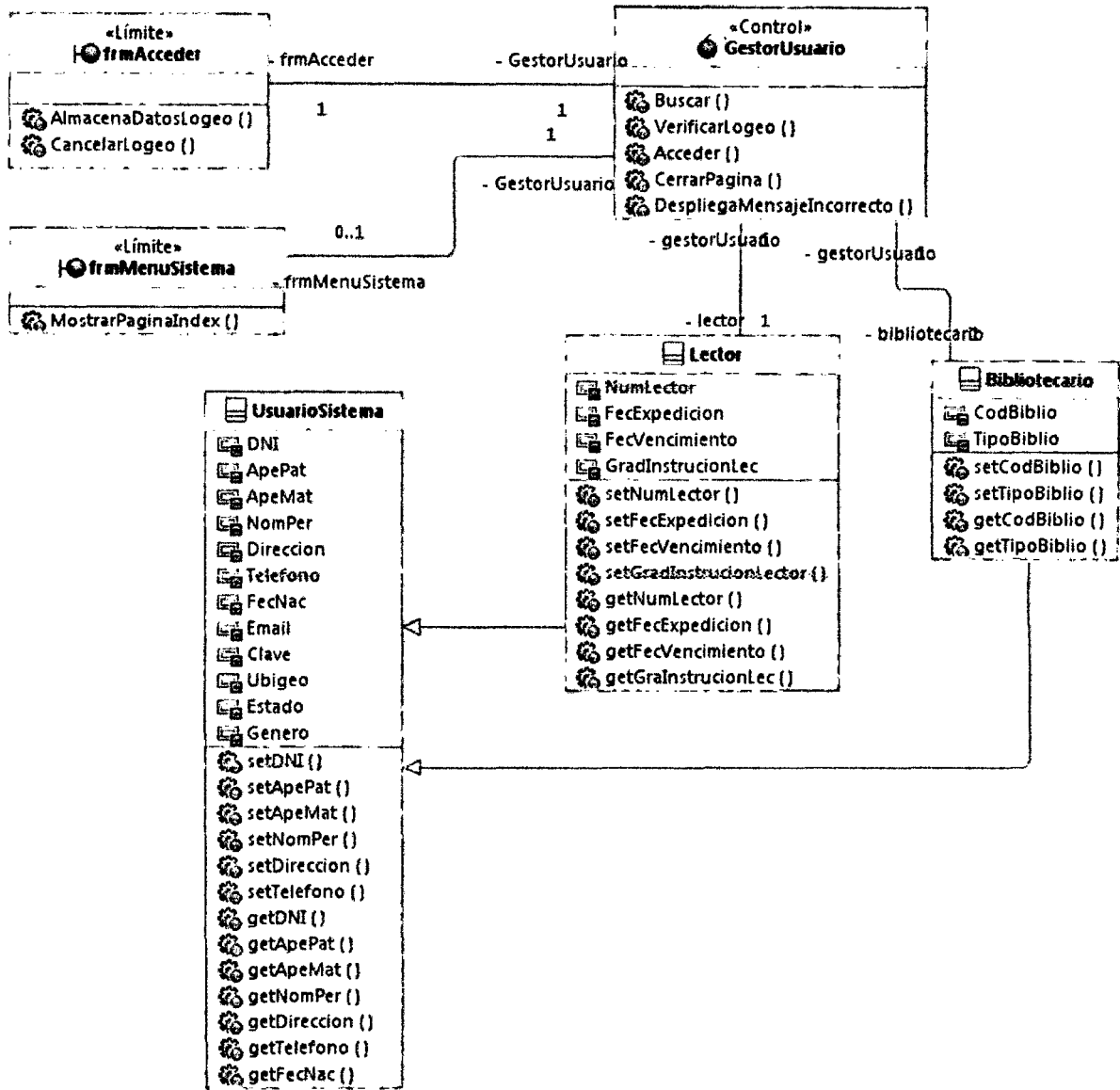


Ilustración 59: Diagrama de Clases de Diseño para Seguridad Sistema
Fuente: Elaboración Propia

b) Diagrama de Clases de Diseño para Administrar Lectores.

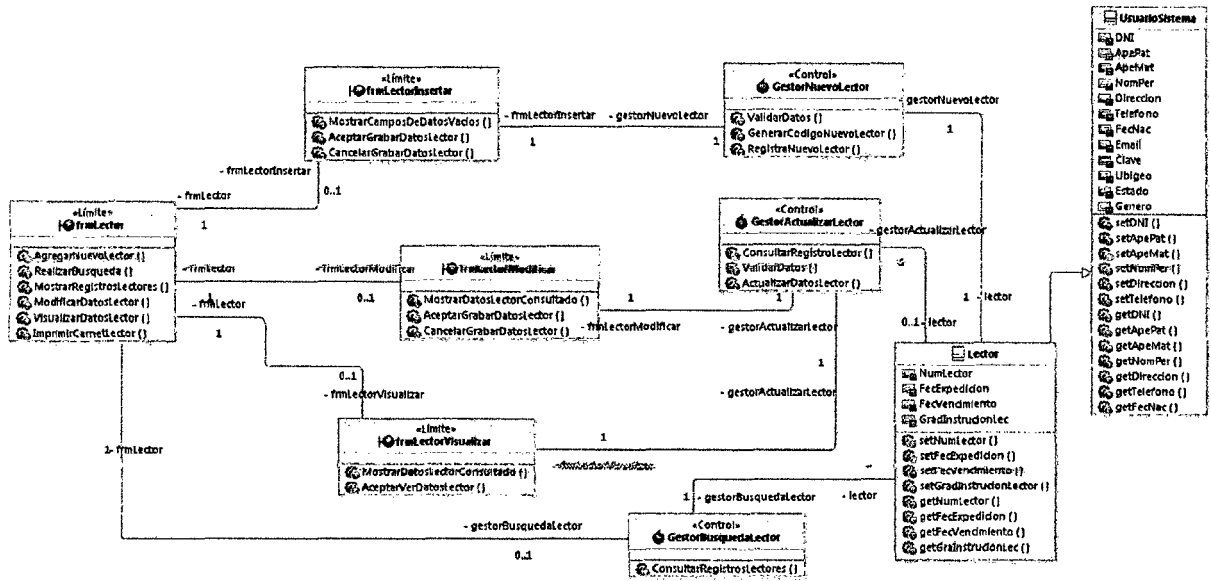


Ilustración 60: Diagrama de Clases de Diseño para Administrar Lectores. Fuente: Elaboración Propia

c) Diagrama de Clases de Diseño para Consultar Material Bibliográfico.



Ilustración 61: : Diagrama de Clases de Diseño para Consultar Material Bibliográfico. Fuente: Elaboración Propia

d) Diagrama de Clases de Diseño para Administrar Material Bibliográfico.

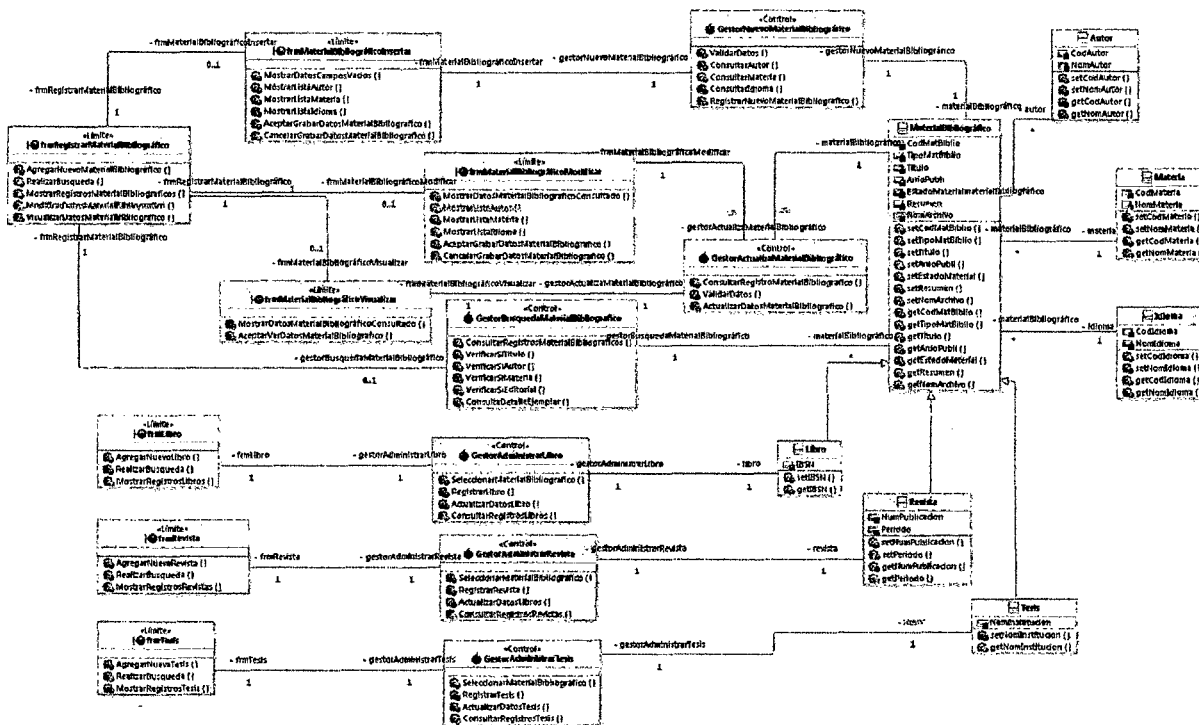


Ilustración 62: : Diagrama de Clases de Diseño para Administrar Material Bibliográfico. Fuente: Elaboración Propia

e) Diagrama de Clases de Diseño para Atención Préstamos y Devoluciones.

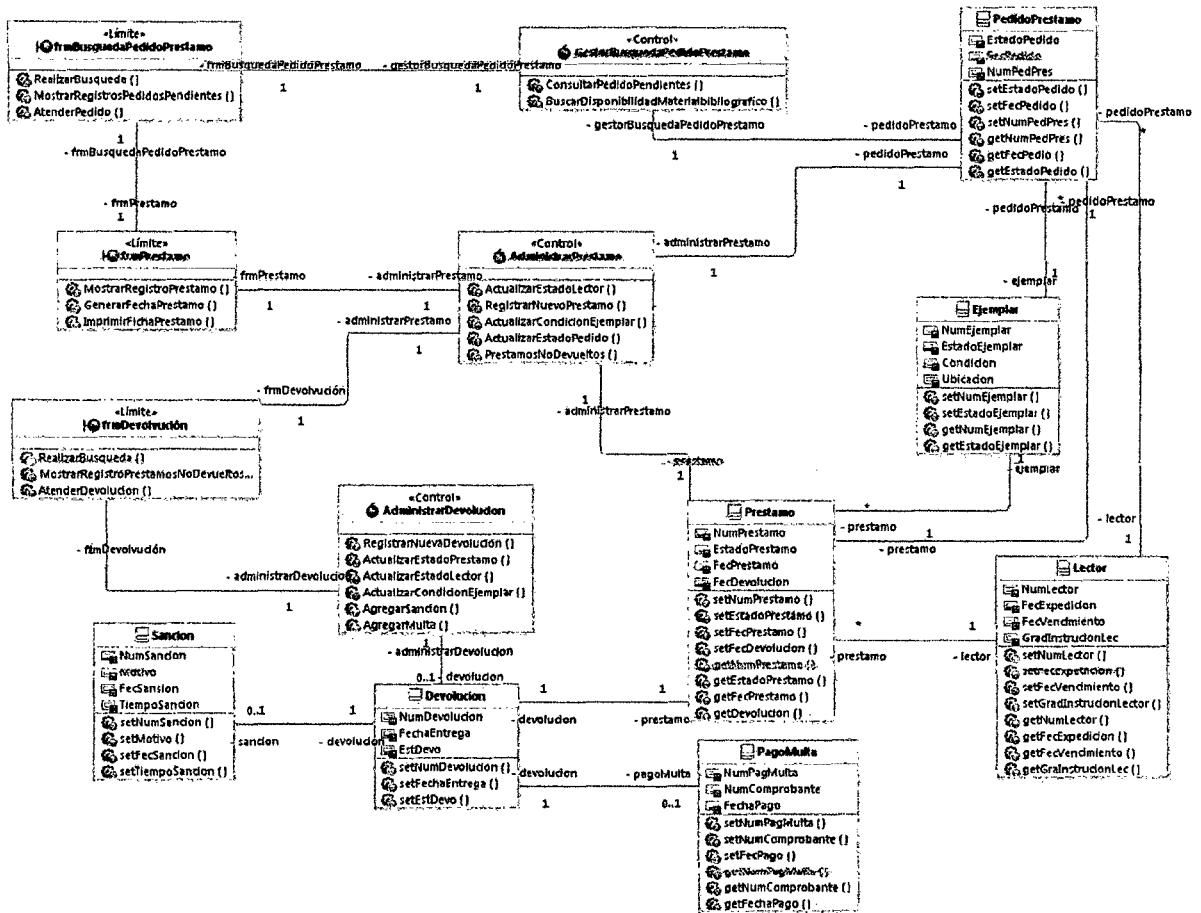


Ilustración 63: Diagrama de Clases de Diseño para Atención de Préstamos y Devoluciones.
Fuente: Elaboración Propia

3.4.1.3. Diagramas de Secuencia de Diseño.

a) Inscripción al Lector.

1. DS Registrar Nuevo Lector.

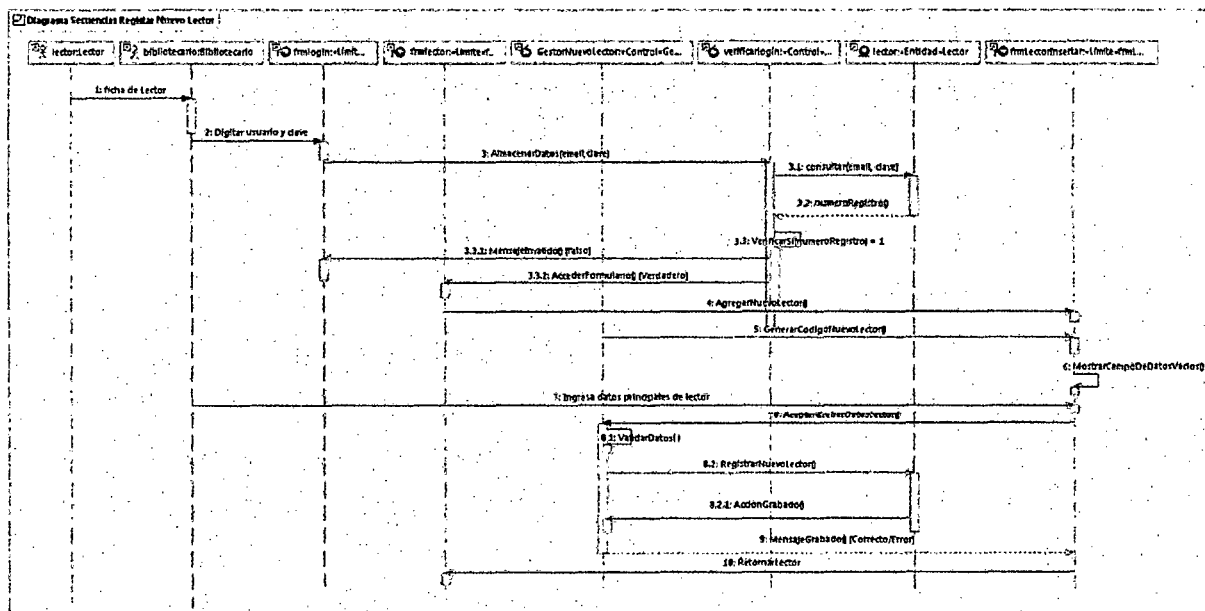


Ilustración 64: Diagrama de Secuencia de Diseño, Inscribir nuevo lector

Fuente: Elaboración Propia

b) Atención de Consulta de Material Bibliográfico.

1. DS Buscar Material Bibliográfico.

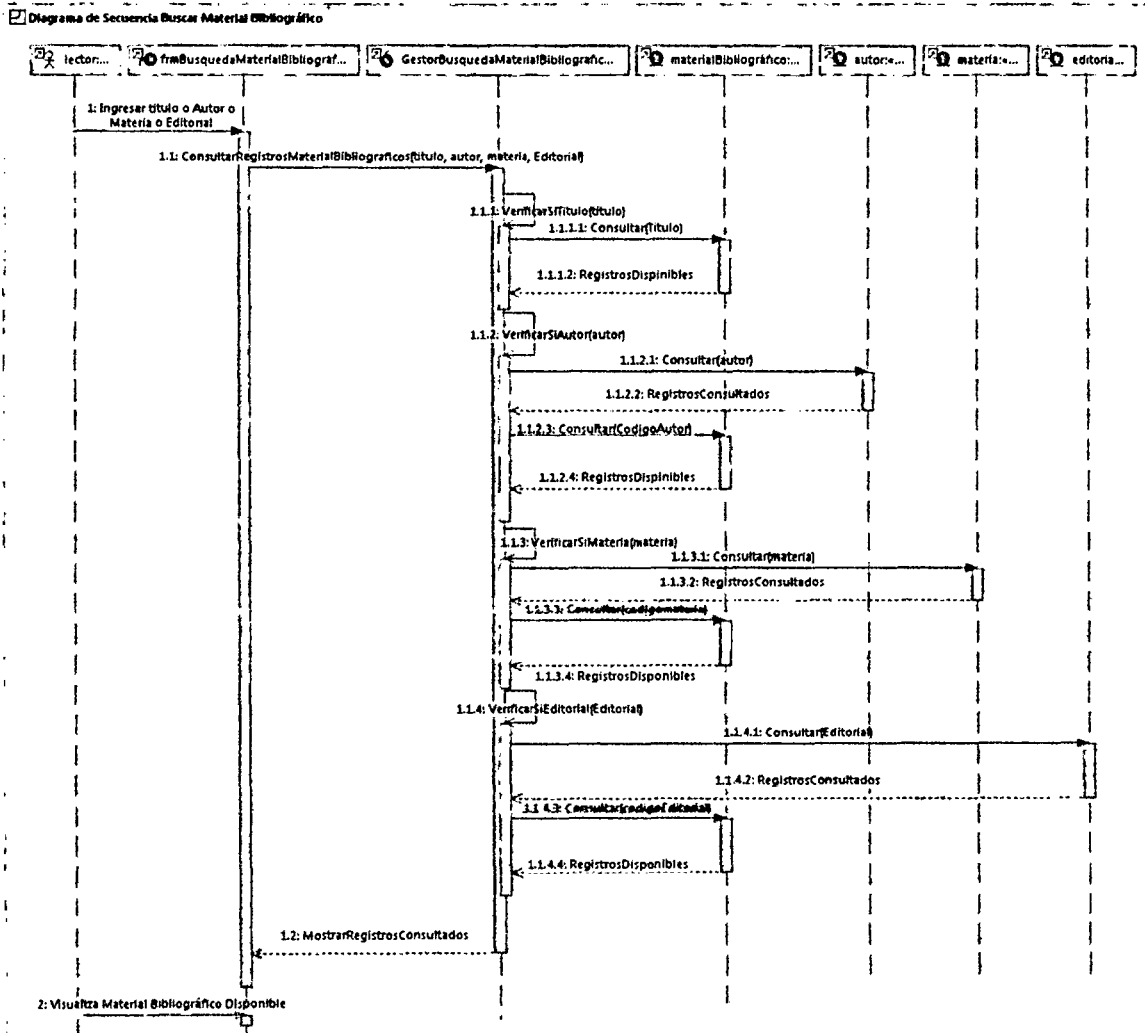


Ilustración 65: Diagrama de Secuencia, Buscar Material Bibliográfico
Fuente: Elaboración Propia.

2. DS Registra Pedido de Préstamo.

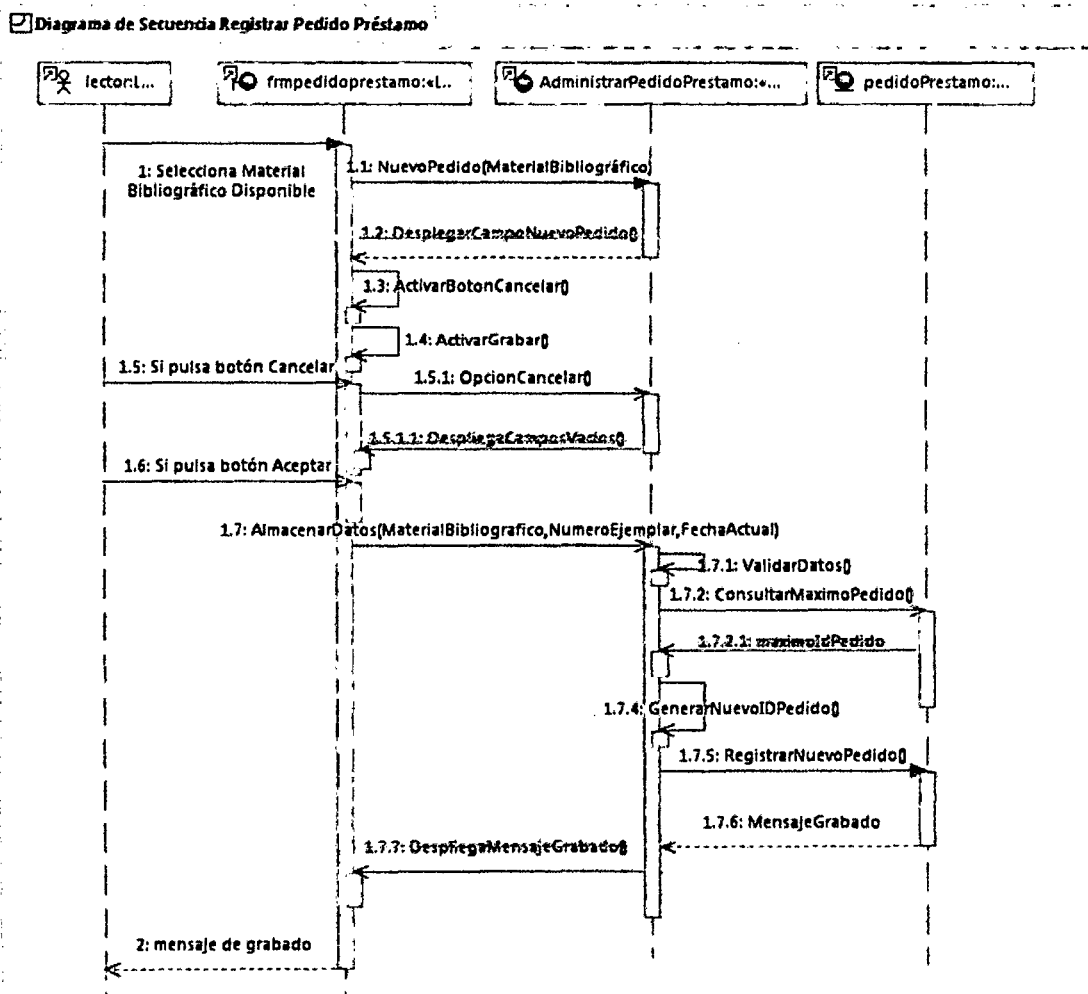


Ilustración 66: Diagrama de Secuencia, Registra pedido Préstamo
Fuente: Elaboración Propia.

c) Atención de Préstamos y Devoluciones.

1. DS Atender pedido de Préstamo.

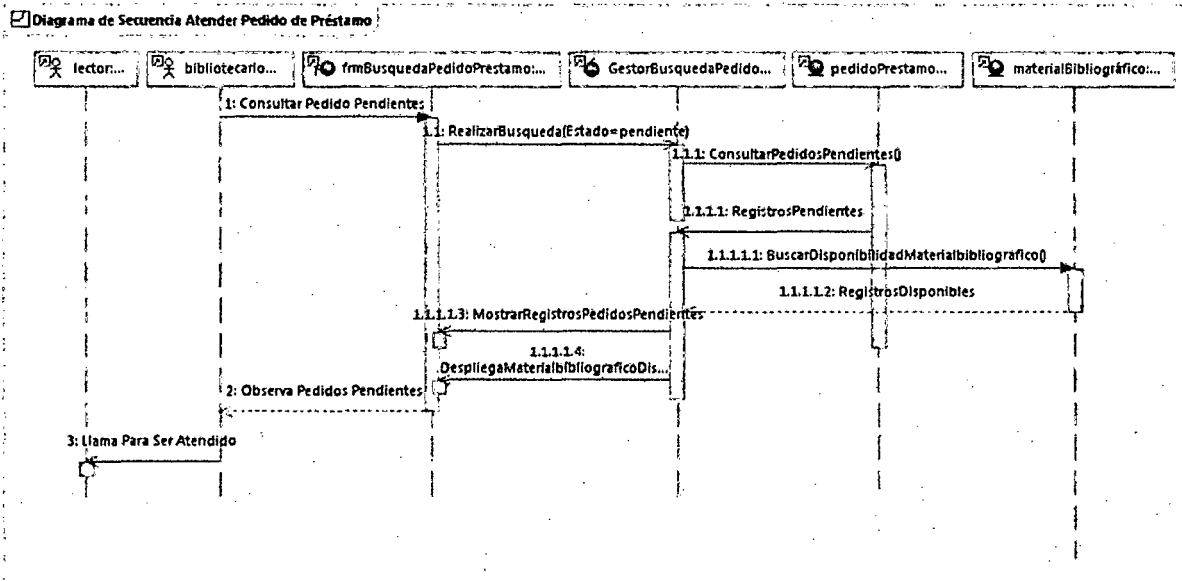


Ilustración 67: Diagrama de Secuencia, Atender pedido de Préstamo
Fuente: Elaboración Propia

2. DS Gestionar y Registro de Préstamo.

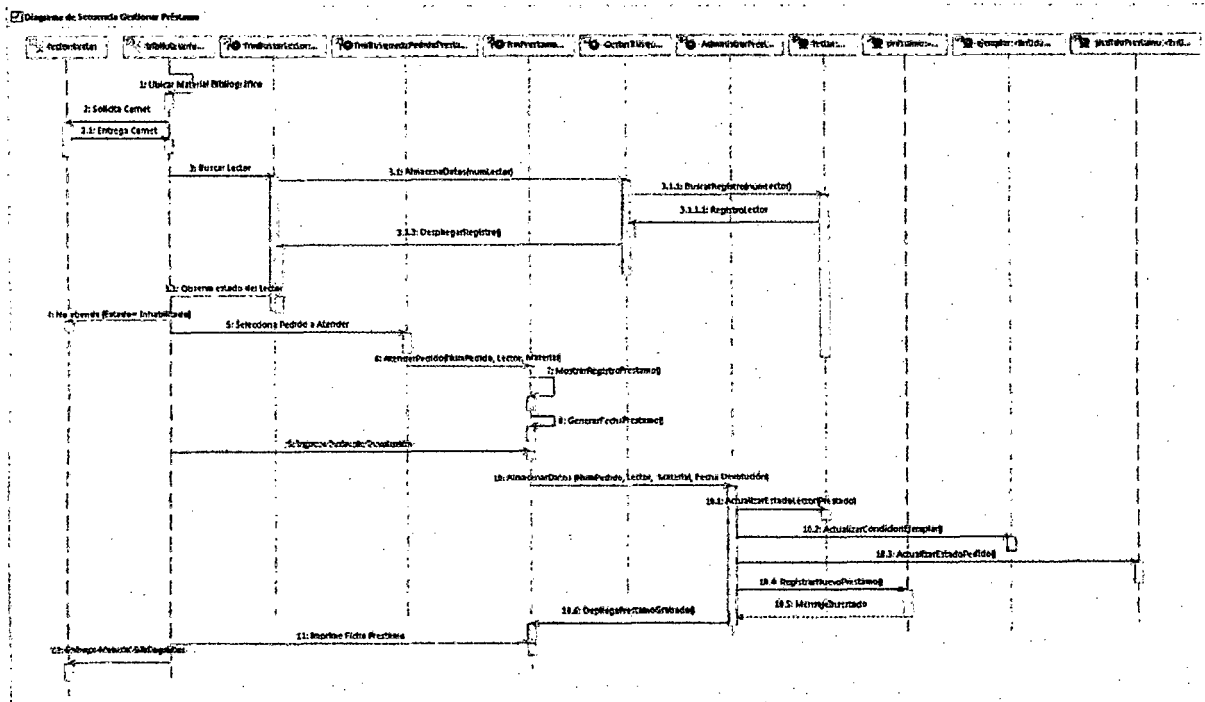


Ilustración 68: Diagrama de Secuencia, Gestionar y Registro Préstamo
Fuente: Elaboración Propia

d) Administración de Material Bibliográfico.

1. DS Elaborar Pedido de Compra.

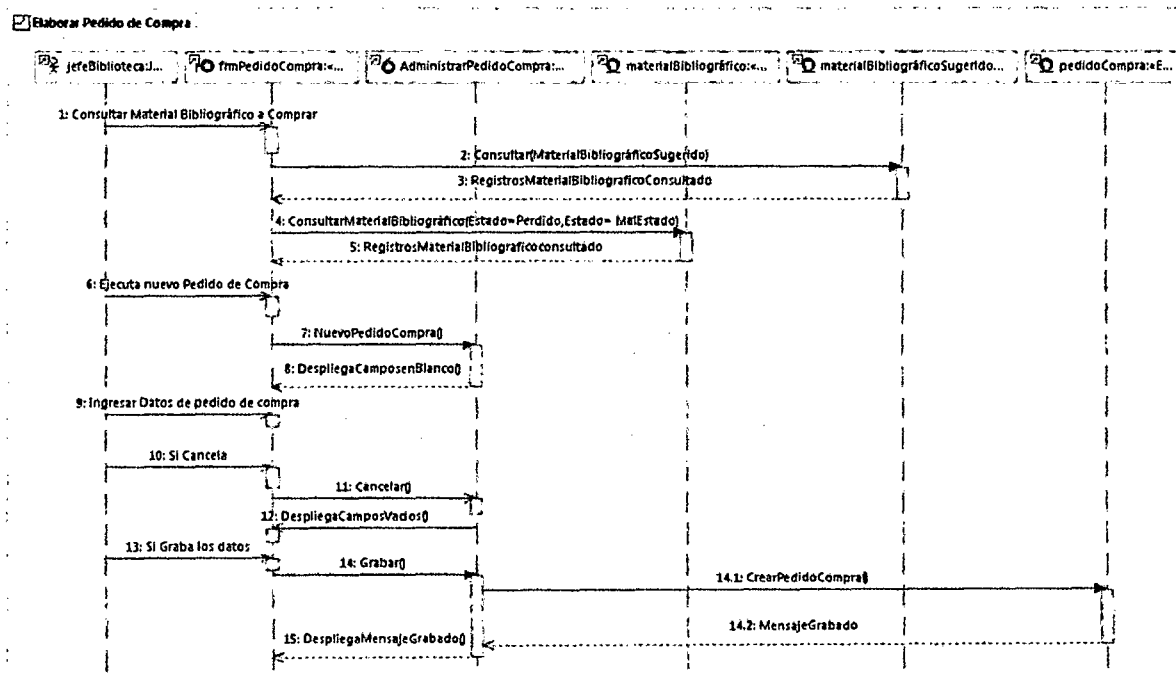


Ilustración 69: Diagrama de Secuencia, Elaborar Pedido de Compra
 Fuente: Elaboración Propia.

2. DS Elaborar Recepciona y Registra Material Bibliográfico.

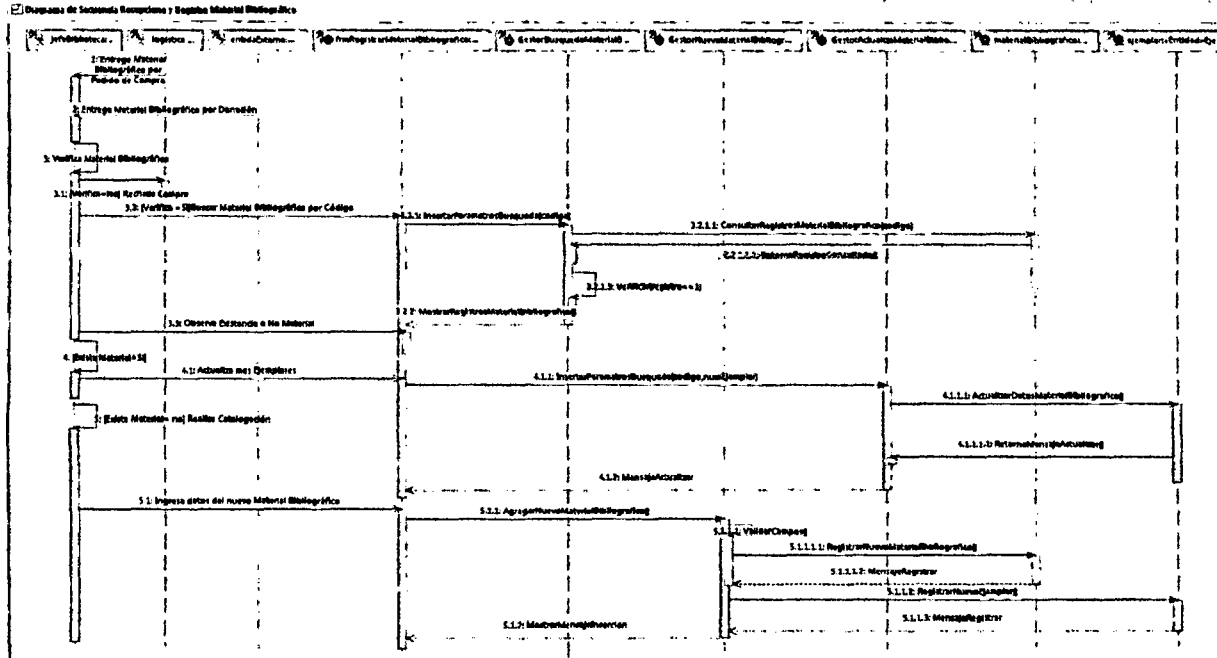


Ilustración 70: Diagrama de Secuencia, Recepciona y Registra Material Bibliográfico

Fuente: Elaboración Propia.

d. Seguridad.

1. DS – Acceder al Sistema

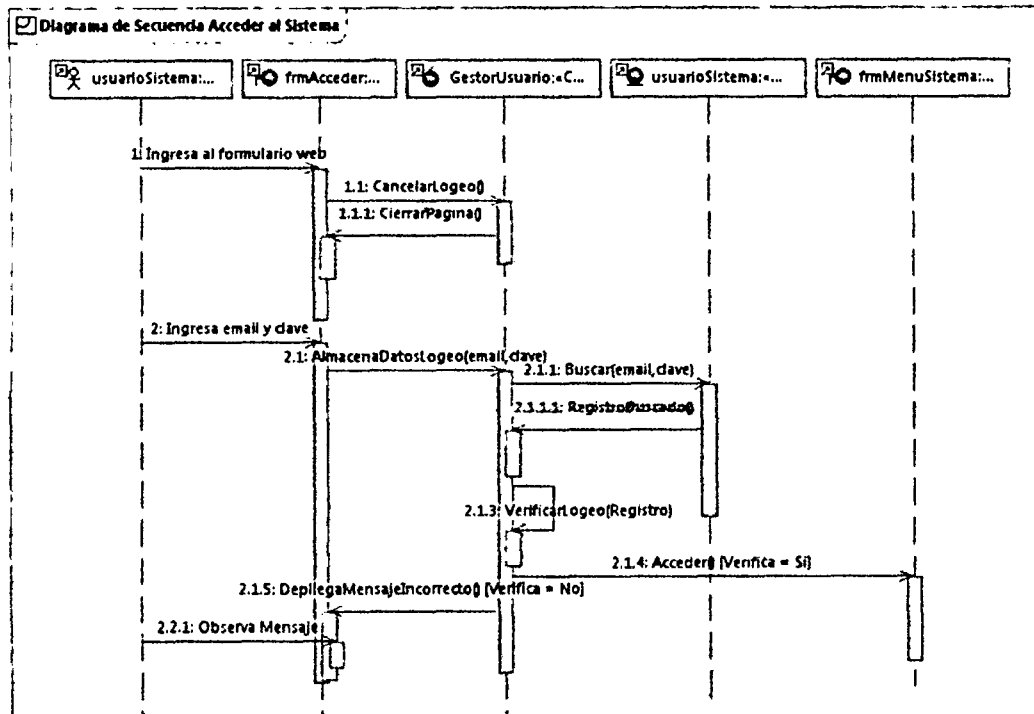


Ilustración 71: Diagrama de Secuencia, Acceder al Sistema

Fuente: Elaboración Propia

2. DS – Auditoria.

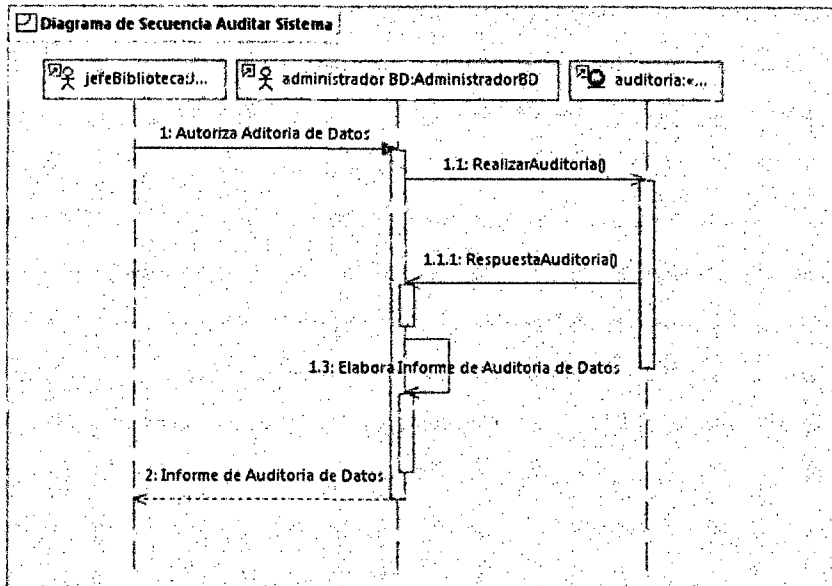


Ilustración 72: Diagrama de Secuencia, Auditoría
Fuente: Elaboración Propia

3.4.1.4. Arquitectura multicapa.

Una arquitectura multicapa es un conjunto ordenado de subsistemas, cada uno de los cuales están constituidos en términos de los que tiene por debajo y proporciona la base de la implementación de aquellos que están por encima de él.

Los objetos de cada capa suelen ser independientes, aunque suelen haber dependencias entre objetos de distintas capas.

Existe una relación cliente/servidor entre las capas inferiores, que son las que proporcionan los servicios, y las capas superiores, los usuarios de estos servicios. El diseño de esta aplicación se basa en una arquitectura multicapa, más concretamente en una arquitectura de tres capas:

- ✓ Capa de presentación, también conocida como interfaz gráfica, se encarga de la presentación de los resultados al usuario y la recogida de información del usuario al sistema.

- ✓ Capa de negocio o lógica de la aplicación, proporciona la funcionalidad de la aplicación. Es el encargado de realizar todas las operaciones a nivel aplicación.
- ✓ Capa de datos o persistencia, este nivel es el encargado de almacenar toda la información de nuestra aplicación, además de asegurar el acceso a la información de una forma controlada y segura. Este nivel lo forman la base de datos y el software de gestión de base de datos.

3.4.1.2.1. DIAGRAMA DE COMPONENTES

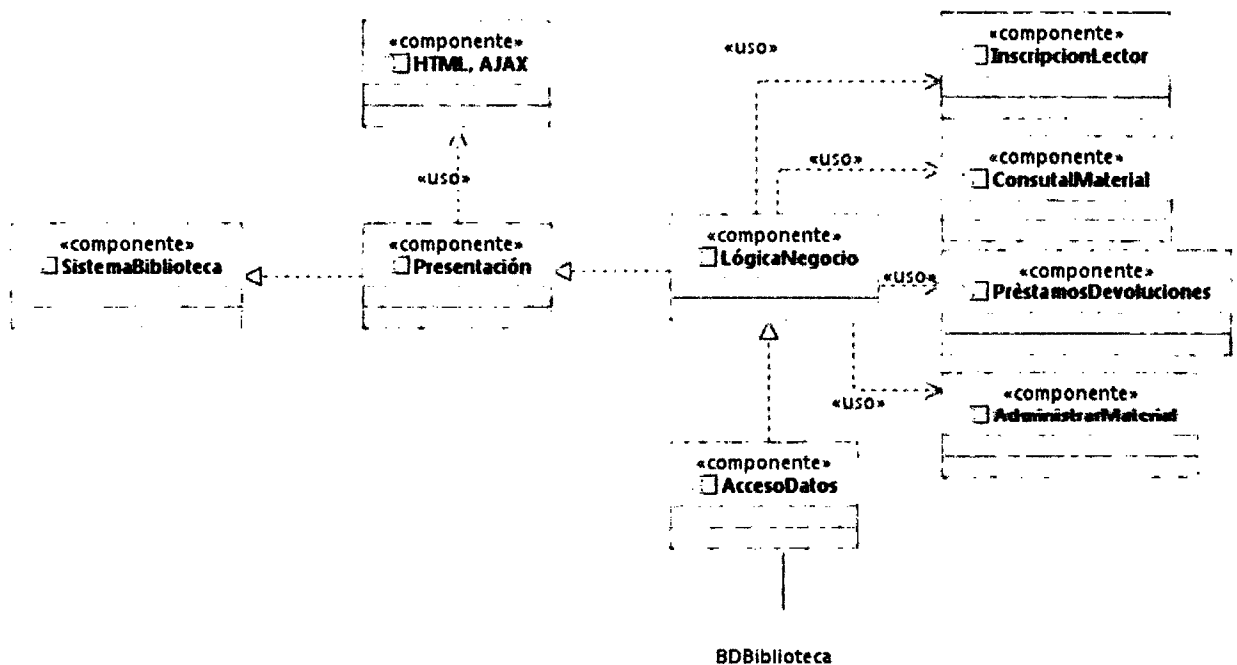


Ilustración 73: Figura N°, Diagrama de Componentes
Fuente: Elaboración Propia

3.4.1.3. NIVEL DE PRESENTACIÓN

En nuestra aplicación existen tres perfiles distintos por lo tanto tres vistas, la del usuario Lector, usuario Bibliotecario y la el Jefe de Biblioteca. El diseño de la interfaz será similar en estas tres vistas, con la diferencia de que dependiendo del usuario este disponga de más o menos opciones en su menú.

La organización de nuestra aplicación Web será la siguiente:

Parte superior, contendrá el logotipo junto al nombre de la institución Municipalidad Provincial de Cajamarca "Biblioteca Municipal".

El usuario lector, dispondrá de las opciones de búsqueda de material bibliográfico, y realizar un pedido de préstamo.

El usuario bibliotecario podrá completar atender los pedidos de préstamos a través de los préstamos y devoluciones del material bibliográfico, sanciones y multas.

El usuario Jefe de Biblioteca, se le añaden las opciones de gestión del Material bibliográfico, Lectores, Pedidos Compras de Material Bibliográfico. Que incorporan tanto las opciones de ingresos y salidas así como cambio de estados, Material Bibliográfico, Lectores y Pedidos de Compras, y permite realizar todo tipo de consultas.

3.4.1.3.1 Diseño de Pantallas del Sistema.

a) Inicio



Ilustración 74: Diseño de pantalla de inicio
Fuente: Elaboración Propia

b) Gestión Lector

BUSCAR LECTOR

Apeŕido Paterno

Apeŕido Materno

Nombre

buscar limpiar agregar

Cant. de Lectores Mostrados

| REGISTRO DE LECTORES | | | | | | |
|----------------------|----------|----------------|----------------|------------|----------------|--|
| ITEM | CÓDIGO | APELLIDOS | NOMBRE (S) | ESTADO | F. VENCIMIENTO | |
| 1 | L0000001 | ROJAS VALENCIA | MARELA | Activo | 2013-12-05 | |
| 2 | L0000099 | MENDOZA SANTOS | CARLOS EDUARDO | Activo | 2013-12-05 | |
| 3 | L0000100 | QUISPE MALCA | ANDERSON | Sancionado | 2013-12-05 | |
| 4 | L0000101 | LEON MENDOZA | ANA | Sancionado | 2013-12-05 | |
| 5 | L0000009 | ROJAS VALENCIA | DANTE | Sancionado | 2013-12-05 | |

Ilustración 75: Diseño de pantalla Buscar y Mantenimiento de Lector
Fuente: Elaboración Propia

INSERTAR LECTOR

Código lector

Apeŕido paterno

Apeŕido materno

Nombre (s)

DNI

Dirección

Ubigeo

Fecha de expedición

Fecha de vencimiento Buscar Material Bibliográfico

Grado de instrucción

Fecha de nacimiento

Estado del lector

Genero del lector

Teléfono

Email - Correo

Contraseña

- DEBE COMPLETAR EL CAMPO "PATERNO"
- DEBE COMPLETAR EL CAMPO "MATERNO"
- DEBE COMPLETAR EL CAMPO "NOMBRE"
- DEBE COMPLETAR EL CAMPO "DNI"
- DEBE COMPLETAR EL CAMPO "DIRECCION"
- DEBE COMPLETAR EL CAMPO "UBIGEO"
- DEBE COMPLETAR EL CAMPO "FECHAEXPEDICION"
- DEBE COMPLETAR EL CAMPO "FECHAVENCIMIENTO"
- DEBE COMPLETAR EL CAMPO "NACIMIENTO"
- DEBE COMPLETAR EL CAMPO "TELEFONO"
- DEBE COMPLETAR EL CAMPO "EMAIL"
- DEBE COMPLETAR EL CAMPO "PASS"
- DEBE ELIGIR UN VALOR DEL COMBO "CBOGRADO"
- DEBE ELIGIR UN VALOR DEL COMBO "CBOESTADO"
- DEBE ELIGIR UN VALOR DEL COMBO "CBOSEXO"

aceptar limpiar cancelar

Ilustración 76: Diseño de pantalla Insertar nuevo Lector
Fuente: Elaboración Propia

MODIFICAR LECTOR

| | |
|----------------------|--|
| Apellido paterno | <input type="text" value="ROJAS"/> |
| Apellido Materno | <input type="text" value="VALENCIA"/> |
| Nombre (s) | <input type="text" value="MARELA"/> |
| DNI | <input type="text" value="74190384"/> |
| Dirección | <input type="text" value="JR. ICA 123"/> |
| Ubigeo | <input type="text" value="010000"/> |
| Fecha de expedición | <input type="text" value="2012-12-05"/> |
| Fecha de vencimiento | <input type="text" value="2013-12-05"/> |
| Grado de instrucción | <input type="text" value="Universidad"/> |
| Fecha de Nacimiento | <input type="text" value="2008-12-05"/> |
| Estado del lector | <input type="text" value="Activo"/> |
| Genero del lector | <input type="text" value="Femenino"/> |
| Teléfono | <input type="text" value="413241"/> |
| Email - Correo | <input type="text" value="MARY@UNC.EDU.PE"/> |
| Contraseña | <input type="text" value="123"/> |

Ilustración 77: Diseño de pantalla Modificar Lector

Fuente: Elaboración Propia.

MODIFICAR LECTOR

LOS DATOS DEL LECTOR HAN SIDO MODIFICADOS CORRECTAMENTE

| | |
|----------------------|-----------------|
| Código lector | 10000001 |
| Apellido paterno | ROJAS |
| Apellido materno | VALENCIA |
| Nombre (s) | MARELA |
| DNI | 74190384 |
| Dirección | JR. ICA 123 |
| Ubigeo | 010000 |
| Fecha de expedición | 2012-12-05 |
| Fecha de vencimiento | 2013-12-05 |
| Grado de instrucción | Universidad |
| Fecha de nacimiento | 2008-12-05 |
| Estado | Activo |
| Genero | Femenino |
| Teléfono | 413241 |
| Email | MARY@UNC.EDU.PE |
| Contraseña | 123 |

Ilustración 78: Diseño de pantalla Guardar Datos del Lector Modificado

Fuente: Elaboración Propia

c) Gestión de Pedido de Préstamo

Buscar Material Bibliográfico

PEDIDO DE PRÉSTAMOS

BUSQUEDA DE MATERIAL BIBLIOGRÁFICO

| | |
|---------|---|
| TÍTULO | |
| AUTOR | |
| MATERIA | <input type="text" value="Seleccione una materia"/> |

No. Material Bibliográficos encontrados
Mostrados

| RELACION DE MATERIAL BIBLIOGRÁFICO | | | | | | |
|------------------------------------|--------|--------------------------------|--------|------------|------------------|---|
| ITEM | CÓDIGO | TÍTULO | AUTOR | MATERIA | CANT. EJEMPLARES | |
| 1 | MB001 | DERIVADAS | BENERO | MATEMATICA | 1 | <input type="button" value="Ver Ejemplares y Realizar Pedido"/> |
| 2 | MB002 | LIMITES DERIVADAS E INTEGRALES | BENERO | MATEMATICA | 0 | <input type="button" value="Ver Ejemplares y Realizar Pedido"/> |

Ilustración 79: Diseño de pantalla Buscar y Mantenimiento de Pedido Préstamos

Fuente: Elaboración Propia

Detalles del Material Bibliográfico y hacer Pedido

DETALLES MATERIAL BIBLIOGRÁFICO

| | |
|-----------------------------|------------|
| Código | MB001 |
| Tipo Material Bibliográfico | Libro |
| Título | DERIVADAS |
| Año de publicación | 1997 |
| Autor | BENERO |
| Materia | MATEMATICA |
| Idioma | Japones |

| RELACION DE EJEMPLARES | | | | |
|------------------------|--------|--------------|---|---|
| ITEM | CÓDIGO | NO. EJEMPLAR | ESTADO | REALIZAR PEDIDO |
| 1 | MB001 | 1 | <input type="button" value="Disponible"/> | <input type="button" value="REALIZAR PEDIDO DE PRESTAMO DE ESTE EJEMPLAR"/> |

Ilustración 80: Diseño de pantalla Detalle del Material Bibliográfico a realizar Pedido de Prestamo

Fuente: Elaboración Propia

Pedido Realizado.

| PEDIDO DE PRESTAMO | |
|--------------------------------------|------------|
| SU PEDIDO A SIDO REALIZADO CON EXITO | |
| Código | MB001 |
| Tipo Material Bibliográfico | Libro |
| Título | DERIVADAS |
| Año de publicación | 1997 |
| Autor | BENERO |
| Materia | MATEMATICA |
| Idioma | Japones |

aceptar

Ilustración 81: Diseño de pantalla Pedido de Prestamo
Fuente: Elaboración Propia

b) Gestión de Préstamos

| BUSQUEDA DE PRESTAMO | | | | | |
|--|----------------------|----------------------|-----------------------|---|---|
| LECTOR | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| TITULO | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| AUTOR | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| | | | | <input type="button" value="buscar"/> | <input type="button" value="limpiar"/> |
| No. Material Bibliograficos encontrados <input type="text"/> | | | | Mostrados <input type="text" value="1-10"/> | |
| RELACION DE PRESTAMOS PENDIENTES | | | | | |
| ITEM | CÓDIGO | NUM. EJEMPLAR | LECTOR | FECHA PEDIDO | ESTADO |
| 1 | MB001 | 1 | ROJAS VALENCIA MARELA | 2012-12-06 | Pendiente <input type="button" value=""/> |
| <input type="button" value="ATENDER PEDIDO DE PRESTAMO"/> | | | | | |

Ilustración 82: Diseño de pantalla Buscar y Mantenimiento de Préstamos
Fuente: Elaboración Propia

Buscar Lector

LECTOR

TITULO

AUTOR

Apellido paterno

Apellido materno

Nombres

No. Material Bibliográficos encontrados

| ITEM | CÓDIGO | NUM. EJEMPLAR | LECTOR |
|------|--------|---------------|---------|
| 1 | MB001 | 1 | ROJAS V |

| Código | Lector |
|----------|-------------------------------|
| L0000001 | ROJAS VALENCIA MARELA |
| L0000009 | ROJAS VALENCIA DARTE |
| L0000099 | MENDOZA SANTOS CARLOS EDUARDO |
| L0000100 | QUISPE BALCA ANDERSON |
| L0000101 | LEON MEHDOZA ANA |

Ilustración 83: Diseño de pantalla Buscar Lector

Fuente: Elaboración Propia

Realizar Préstamo

INSEJAR PRESTAMO

Fecha de préstamo

Fecha de devolución

Código material bibliográfico

Número de ejemplar

Materia

Título

Autor

Apellido paterno

Apellido materno

Nombre

Ilustración 84: Diseño de pantalla Realizar Préstamo

Fuente: Elaboración Propia

Se grabó el Préstamo

| GUARDAR PRESTAMO | |
|-------------------------------|------------|
| Fecha de préstamo | 2012/12/05 |
| Fecha de evolución | 2012/11/06 |
| Código material bibliográfico | MB001 |
| Número de ejemplar | 1 |
| Materia | MATEMATICA |
| Título | DERIVADAS |
| Autor | GENERO |
| Apellido paterno | ROJAS |
| Apellido materno | VALENCIA |
| Nombres | MARELA |

EL PRESTAMO HA SIDO GRABADO CORRECTAMENTE

aceptar

Ilustración 85: Diseño de pantalla Guarda Préstamo
Fuente: Elaboración Propia

Retorna a Buscar más Pedidos de Prestamos Pendientes

| BÚSQUEDA DE PRESTAMO | |
|----------------------|----------------------|
| LECTOR | <input type="text"/> |
| TÍTULO | <input type="text"/> |
| AUTOR | <input type="text"/> |

No. Material Bibliograficos encontrados Mostrados

| RELACION DE PRESTAMOS PENDIENTES | | | | | |
|--|--------|---------------|--------|--------------|--------|
| ITEM | CÓDIGO | NUM. EJEMPLAR | LECTOR | FECHA PEDIDO | ESTADO |
| NO HAY NINGÚN PRESTAMO PENDIENTE QUE CUMPLA CON LOS CRITERIOS DE BÚSQUEDA | | | | | |

Ilustración 86: Diseño de pantalla Retorna despues de Realizar Pedido de Préstamo
Fuente: Elaboración Propia

e) Gestión de Devoluciones

Atender Devolución

BUSQUEDA DE DEVOLUCION

TITULO

AUTOR

No. Material Bibliograficos encontrados Mostrados

RELACION DE DEVOLUCIONES PENDIENTES

| ITEM | CODIGO | NO. EJEMPLAR | LECTOR | F. PRESTAMO | F. DEVOLUCION | ESTADO |
|------|--------|--------------|-----------------------|-------------|---------------|---|
| 1 | MB001 | 1 | ROJAS VALENCIA MARELA | 2012-12-06 | | <input type="button" value="Prestado"/> |

Ilustración 87: Diseño de pantalla Buscar y atender devolución de Préstamo
Fuente: Elaboración Propia

Realizar Devolución

REALIZAR DEVOLUCION

Fecha de prestamo

Fecha de devolución

Fecha de entrega

Estado libro de devolución

Código material bibliográfico

Número de ejemplar

Materia

Título

Autor

Apellido paterno

Apellido materno

Nombre

Ilustración 88: Diseño de pantalla Realizar Devolución
Fuente: Elaboración Propia

Guardar Devolución

| GUARDAR DEVOLUCION | |
|-------------------------------|------------|
| Fecha de prestamo | 2012-12-05 |
| Fecha de devolución | 2012-11-05 |
| Fecha de entrega | 2012-12-06 |
| Código material bibliográfico | MB001 |
| Número de ejemplar | 1 |
| Materia | MATEMATICA |
| Título | DERIVADAS |
| Autor | BENIERO |
| Apellido paterno | ROJAS |
| Apellido materno | VALENCIA |
| Nombres | MARELA |

LA DEVOLUCION HA SIDO GRABADO CORRECTAMENTE

Ilustración 89: Diseño de pantalla Buscar y Mantenimiento de Lector
Fuente: Elaboración Propia

Retorna de Prestamos Pendientes

| BUSQUEDA DE DEVOLUCION | |
|------------------------|----------------------|
| TITULO | <input type="text"/> |
| AUTOR | <input type="text"/> |

No. Material Bibliograficos encontrados Mostrados

| RELACION DE DEVOLUCIONES PENDIENTES | | | | | | |
|---|--------|--------------|--------|-------------|---------------|--------|
| ITEM | CODIGO | NO. EJEMPLAR | LECTOR | F. PRESTAMO | F. DEVOLUCION | ESTADO |
| NO HAY NINGUN MATERIAL BIBLIOGRAFICO QUE CUMPLA CON LOS CRITERIOS DE BUSQUEDA | | | | | | |

Ilustración 90: Diseño de pantalla Buscar y Retrona Préstamos Pendientes
Fuente: Elaboración Propia

f) Gestión de Pago de Inscripción

REGISTRAR PAGO DE INSCRIPCIÓN

Apeido paterno

Apeido materno

Nombres

Nro. de inscripciones encontradas: Mostrados: 1-10

| ITEM | NO. PAGO | LECTOR | FECHA DE PAGO | MONTO |
|------|----------|------------------|---------------|-------|
| 1 | 1 | LEON MENDOZA ANA | 2012-12-05 | 3.00 |

Ilustración 91: Diseño de pantalla Buscar y Mantenimiento de Pago de Inscripción

Fuente: Elaboración Propia

REGISTRAR PAGO DE INSCRIPCIÓN

Número de pago

Número de comprobante

Fecha de pago

Monto

Código de lector

Ilustración 92: Diseño de pantalla Insertar pago de Inscripción

Fuente: Elaboración Propia

g) Gestión Bibliotecario

BUSCAR BIBLIOTECARIO

Apeido Paterno

Apeido Materno

Nombre

Cant. de Bibliotecarios: Mostrados: 1-10

| ITEM | CÓDIGO | APELLIDOS | NOMBRE (S) | ESTADO | TIPO |
|------|--------|--------------|--------------|--|-----------------------|
| 1 | B0001 | MENDOZA MARN | MIGUEL ANGEL | <input type="button" value="Permiso"/> | Bibliotecario |
| 2 | B0002 | MARN CHAVEZ | ESTHER | <input type="button" value="Activo"/> | Jefe de Bibliotecario |

Ilustración 93: Diseño de pantalla Buscar y Mantenimiento de Bibliotecario

Fuente: Elaboración Propia

INSERTAR BIBLIOTECARIO

| | | |
|--------------------------|---------------------------------|--|
| Código bibliotecario | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "CODBIBLIO" |
| Apeñdo paterno | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "PATERNO" |
| Apeñdo materno | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "MATERNO" |
| Nombre (s) | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "NOMBRE" |
| Dirección | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "DIRECCION" |
| Teléfono | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "TELEFONO" |
| Fecha de nacimiento | 2012/12/06 <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "NACIMIENTO" |
| Email - Correo | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "EMAIL" |
| Contraseña | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "PASS" |
| Ubigeo | <input type="text"/> | <input type="radio"/> DEBE COMPLETAR EL CAMPO "UBIGEO" |
| Estado del bibliotecario | <input type="text"/> | <input type="radio"/> DEBE ELIGIR UN VALOR DEL COMBO "CBOESTADO" |
| Genero del Bibliotecario | <input type="text"/> | <input type="radio"/> DEBE ELIGIR UN VALOR DEL COMBO "CBOSEXO" |
| DNI | <input type="text"/> | <input type="radio"/> DEBE ELIGIR UN VALOR DEL COMBO "CBO TIPO" |
| Tipo | <input type="text"/> | |

aceptar
 limpiar
 cancelar

Ilustración 94: Diseño de pantalla Insertar Bibliotecario
Fuente: Elaboración Propia

MODIFICAR BIBLIOTECARIO

| | | |
|---------------------|--|--|
| Apeñdo paterno | MARN | |
| Apeñdo materno | CHAVEZ | |
| Nombre (s) | ESTHER | |
| Dirección | JR. ATAHUALPA | |
| Teléfono | 43124132 | |
| Fecha de nacimiento | 2012-12-03 <input type="text"/> | |
| Email - Correo | ESTHERMARN@MPC.GOB.PE | |
| Contraseña | 123456 | |
| Ubigeo | 010107 <input type="text"/> | |
| Estado | Activo <input type="text"/> | |
| Genero | Femenino <input type="text"/> | |
| DNI | 42623989 | |
| Tipo | Jefe de bibliotecario <input type="text"/> | |

aceptar
 limpiar
 cancelar

Ilustración 95: Diseño de pantalla Modificar Bibliotecario
Fuente: Elaboración Propia

| MODIFICAR BIBLIOTECARIO | |
|---|------------------------|
| LOS DATOS DEL BIBLIOTECARIO HAN SIDO MODIFICADOS CORRECTAMENTE | |
| Código bibliotecario | 80002 |
| Apellido paterno | MARRI |
| Apellido materno | CHAVEZ |
| Nombre (s) | ESTHER |
| Dirección | JR. ATAHUALPA |
| teléfono | 43124132 |
| Fecha de nacimiento | 2012-12-03 |
| Email | ESTHERMARRI@MPC.GOB.PE |
| Constraseña | 123456 |
| Ubigeo | 010107 |
| Estado | Activo |
| Genero | Femenino |
| DNI | 42623989 |
| Email | ESTHERMARRI@MPC.GOB.PE |
| Tipo | Jefe de Bibliotecario |

Ilustración 96: Diseño de pantalla Confirmar el Grabado de la Modificación
Fuente: Elaboración Propia

h) Gestión Autores

| BUSCAR AUTOR | |
|--------------|---|
| Nombre | <input type="text"/> |
| País | <input type="text" value="Seleccione un País"/> |

Nro. de autores encontrados Mostrados

| RELACION DE AUTORES | | | |
|---------------------|--------|--------|------|
| ITEM | CÓDIGO | AUTOR | PAÍS |
| 1 | A0001 | BEHERO | Peru |

Ilustración 97: Diseño de pantalla Buscar y Mantenimiento de Autores
Fuente: Elaboración Propia

MODIFICAR AUTOR

NOMBRE

PAÍS

aceptar
 limpiar
 cancelar

Ilustración 98: Diseño de pantalla Modificar Autor
Fuente: Elaboración Propia

INSERTAR AUTOR

CÓDIGO

NOMBRE

PAÍS

aceptar
 limpiar
 cancelar

Ilustración 99: Diseño de pantalla Insertar Autor
Fuente: Elaboración Propia

INICIO >> AUTOR >> MODIFICAR AUTOR

LOS DATOS DEL AUTOR HAN SIDO MODIFICADOS CORRECTAMENTE

Código A0001

Nombre BENERO

País Peru

aceptar

Ilustración 100: Diseño de pantalla Buscar Modificar Autor
Fuente: Elaboración Propia

i) Gestión Editorial

BUSCAR EDITORIAL

Nombre

País

buscar
 limpiar
 agregar

Nro. de editoriales encontradas Mostrados: 1-20

| RELATIVOS DE EDITORIAL | | | |
|------------------------|--------|---------------------|--------|
| ITEM | CÓDIGO | EDITORIAL | PAÍS |
| 1 | E0001 | EDITORIAL UNICA S.A | México |
| 2 | E0002 | CALLAO | Peru |

Ilustración 101: Diseño de pantalla Buscar y Mantenimiento de Editorial
Fuente: Elaboración Propia

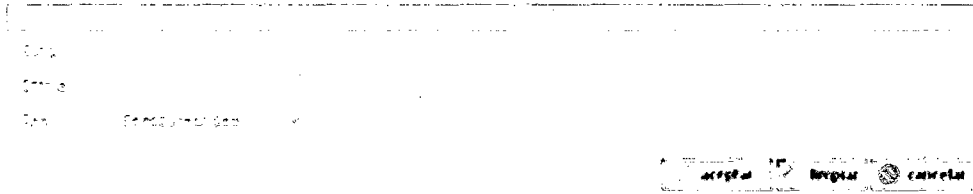


Ilustración 102: Diseño de pantalla Insertar Editorial

Fuente: Elaboración Propia



Ilustración 103: Diseño de pantalla Modificar Editorial

Fuente: Elaboración Propia

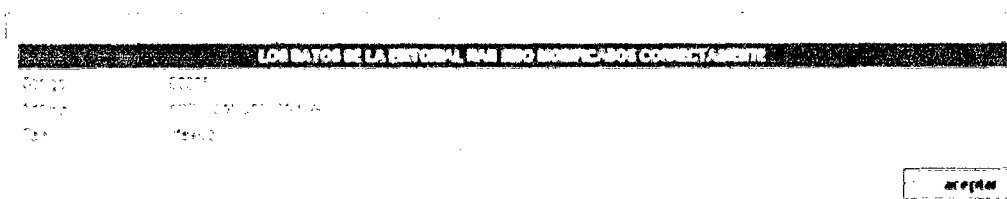


Ilustración 104: Diseño de pantalla Buscar y Mantenimiento de Editorial

Fuente: Elaboración Propia

g) Gestión Materia

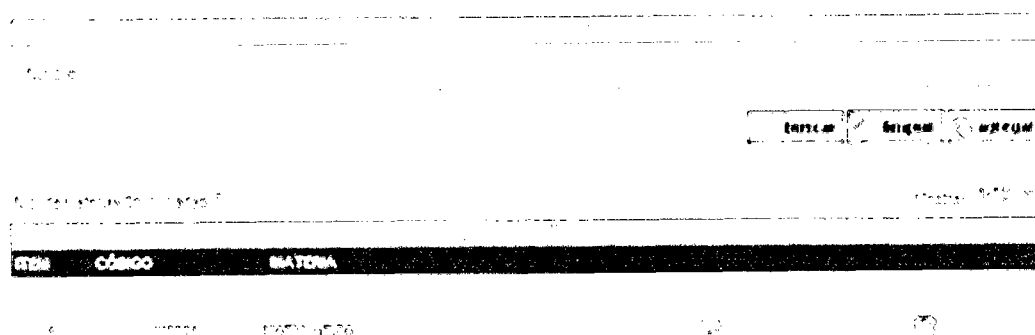


Ilustración 105: Diseño de pantalla Buscar y Mantenimiento de Materia

Fuente: Elaboración Propia

Ilustración 106: Diseño de pantalla Insertar Materia

Fuente: Elaboración Propia

Ilustración 107: Diseño de pantalla Modificar Materia

Fuente: Elaboración Propia

Ilustración 108: Diseño de pantalla guardar Datos de Materia

Fuente: Elaboración Propia

k) Gestión Material Bibliográfico

| ITEM | CÓDIGO | TÍTULO | AUTOR | MATERIA | ESTADO |
|------|--------|--------------------------------|-------|------------|--------|
| 1 | 10001 | DERIVADAS | CERRO | MATEMÁTICA | ACTIVO |
| 2 | 10002 | LIMITES DERIVADAS E INTEGRALES | CERRO | MATEMÁTICA | ACTIVO |

Ilustración 109: Diseño de pantalla Buscar y Mantenimiento de Material Bibliográfico

Fuente: Elaboración Propia

INSERTAR MATERIAL BIBLIOGRAFICO

| | | |
|-------------------------------|------------------------|---|
| Código material bibliográfico | <input type="text"/> | <input type="checkbox"/> DEBE COMPLETAR EL CAMPO "COORDINADOR" |
| Tipo material bibliográfico | Seleccione un tipo | <input type="checkbox"/> DEBE COMPLETAR EL CAMPO "TITULO" |
| Título material bibliográfico | <input type="text"/> | <input type="checkbox"/> DEBE COMPLETAR EL CAMPO "FECHA PUBLICACION" |
| Año de publicación | <input type="text"/> | <input type="checkbox"/> DEBE ELIJIR UN VALOR DEL CONJUNTO "CODOTPO" |
| Estado del material | Seleccione un estado | <input type="checkbox"/> DEBE ELIJIR UN VALOR DEL CONJUNTO "CODOSTADO" |
| Autor | Seleccione un autor | <input type="checkbox"/> DEBE ELIJIR UN VALOR DEL CONJUNTO "CODIAUTOR" |
| Materia | Seleccione una materia | <input type="checkbox"/> DEBE ELIJIR UN VALOR DEL CONJUNTO "CODMATERIA" |
| Idioma | Seleccione un idioma | <input type="checkbox"/> DEBE ELIJIR UN VALOR DEL CONJUNTO "CODIDIOMA" |
| Resumen | <input type="text"/> | |
| Nombre del archivo | <input type="text"/> | |

aceptar limpiar cancelar

Ilustración 110: Diseño de pantalla Insertar Material Bibliográfico
Fuente: Elaboración Propia

MODIFICAR MATERIAL BIBLIOGRAFICO

| | |
|-------------------------------|----------------------------------|
| Tipo Material Bibliográfico | Revisó |
| Título Material Bibliográfico | FUNCIONES DERIVADAS E INTEGRALES |
| Año de Publicación: | 2009 |
| Estado Material: | Aprobó |
| Autor | GENERO |
| Materia | MATEMATICA |
| Idioma | Español |
| Resumen | <input type="text"/> |
| Nombre Archivo | <input type="text"/> |

aceptar limpiar cancelar

Ilustración 111: Diseño de pantalla Modificar Material Bibliográfico
Fuente: Elaboración Propia

LOS DATOS DEL MATERIAL BIBLIOGRÁFICO SON LOS SIGUIENTES:

Organización bibliográfica: 10000
 Tipo de material: 10000
 Tipo de obra bibliográfica: 10000
 Año de publicación: 2000
 Setor de origen: 10000
 Autor: 10000
 Título: 10000
 Lugar: 10000
 País: 10000
 Clasificación: 10000

Ilustración 112: Diseño de pantalla Guardar Datos de Material Bibliográfico

Fuente: Elaboración Propia

Libros

Tipo de obra bibliográfica:

| ID | CÓDIGO | TÍTULO | AUTOR | EDITORIAL |
|----|--------|--------|-------|-----------|
| 1 | 10000 | 10000 | 10000 | 10000 |

Ilustración 113: Diseño de pantalla Buscar y Mantenimiento de Libros

Fuente: Elaboración Propia

Ilustración 114: Diseño de pantalla de Agregar Libro

Fuente: Elaboración Propia

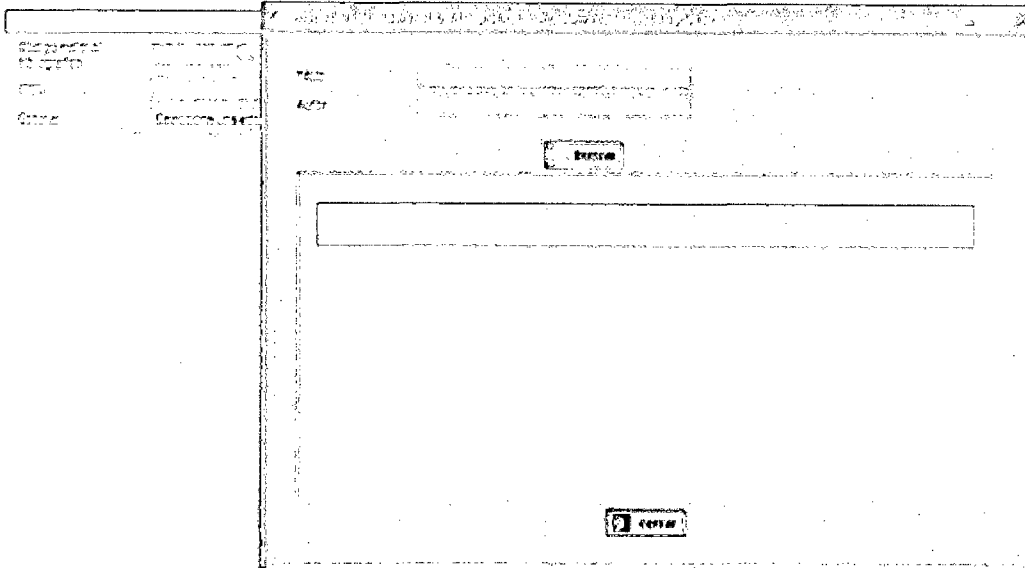


Ilustración 115: Diseño de pantalla Buscar Material Bibliográfico
Fuente: Elaboración Propia

Revistas

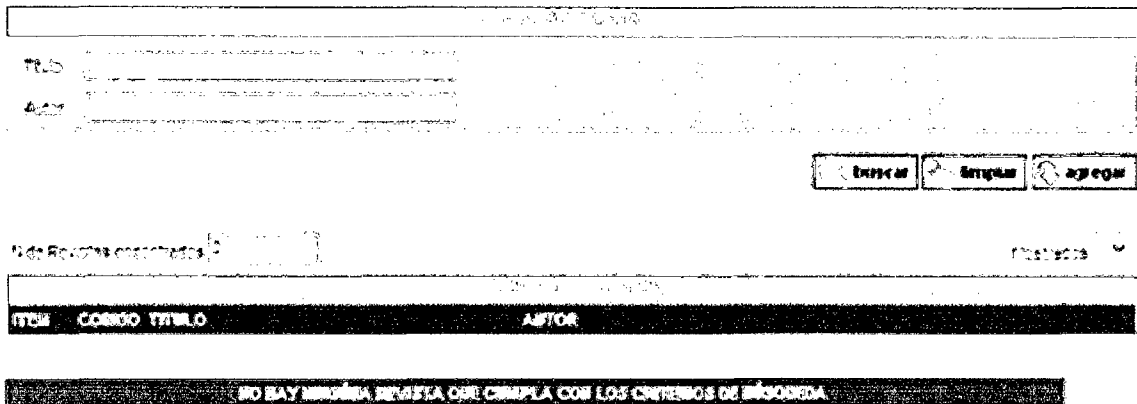


Ilustración 116: Diseño de pantalla Buscar y Mantenimiento de Revistas
Fuente: Elaboración Propia

Tesis

SEARCH

Título

Autor

Buscar Borrar Agregar

Más Tesis relacionadas

Mostrar

| ITEM | CODIGO | TITULO | AUTOR |
|------|--------|--------|-------|
|------|--------|--------|-------|

BIBLIOTECA MUNICIPAL JOSÉ GÁLVEZ DE LA MUNICIPALIDAD PROVINCIAL DE CAJAMARCA

Ilustración 117: Diseño de pantalla de Tesis
Fuente: Elaboración Propia

3.4.1.4. Nivel de lógica de la aplicación.

Este nivel estará formado por un conjunto de funciones que agrupare en paquetes y estas a las vez contienen clases que están en archivos php, estas funciones sustituyen a los métodos que tendrían las clases correspondientes a los diagramas de clases. Su funcionalidad hará posible las operaciones descritas en apartados anteriores, conexión con la base de datos, manipulación de la base de datos, validación de usuarios lectores, Bibliotecarios, Consultas de Material Bibliográfico, Pedidos de Prestamos, Préstamos y Devoluciones, Pedidos de Compras, Sanciones y Multas.

La separación por capas nos permite tener una independencia entre niveles, de esta manera el mantenimiento y las posibles ampliaciones de la aplicación se realizarán de una forma sencilla, rápida y limpia.

3.4.1.5. Nivel de datos o persistencia.

Este nivel estará formado por una base de datos relacional, la cual, almacenará todos los datos relacionados con nuestra aplicación Web.

3.4.1.6. Diagrama entidad relación.

A continuación tenemos el diagrama entidad relación que vamos a utilizar para la creación de nuestra base de datos.

Diagrama de Entidad Relación

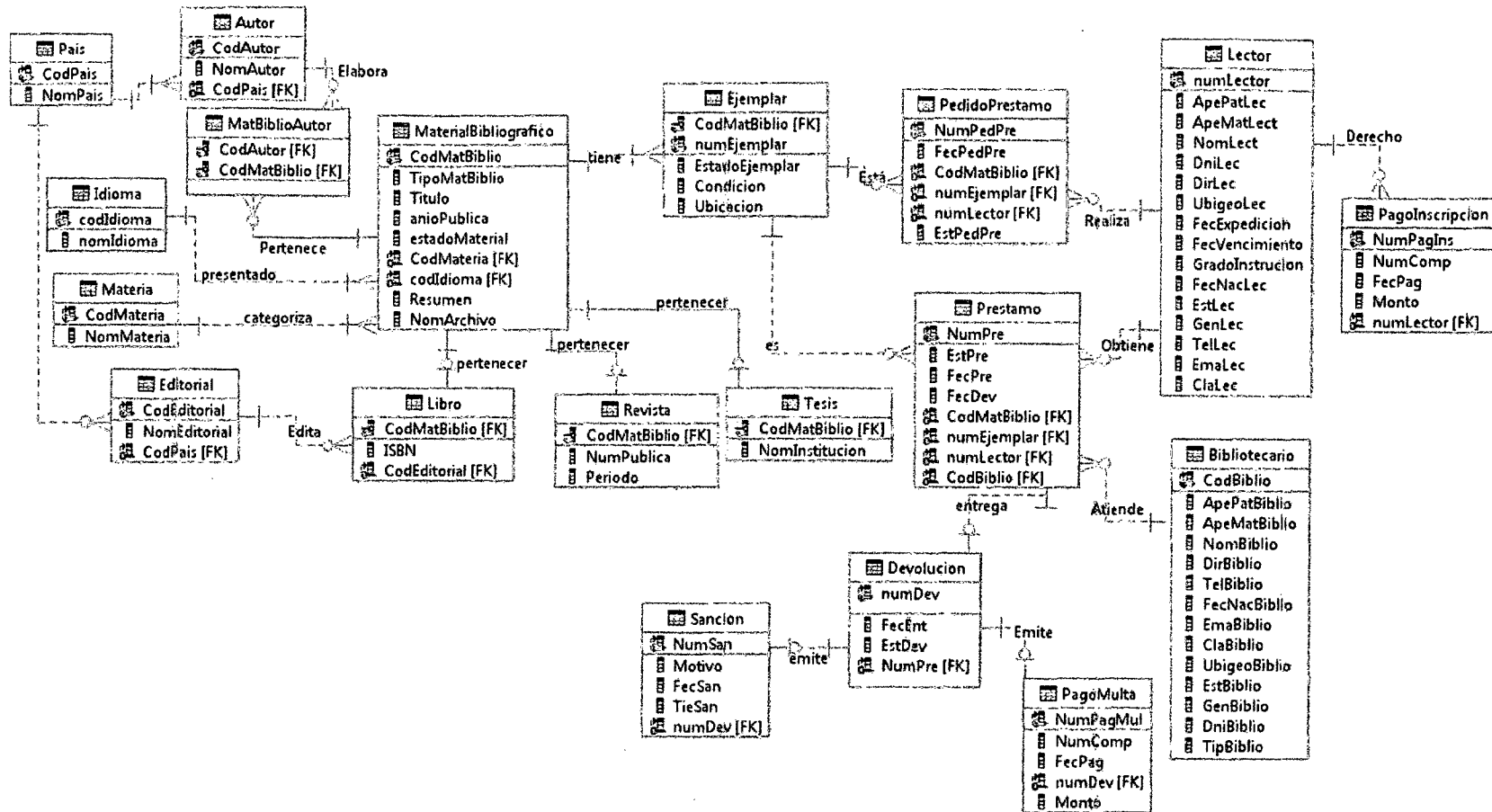


Ilustración 118: Diagrama de Entidad Relación.

Fuente: Elaboración Propia

3.4.1.7. Diseño lógico.

```
CREATE TABLE Lector (  
    numLector VARCHAR(8) NOT NULL,  
    ApePatLec VARCHAR(40),  
    ApeMatLect VARCHAR(40),  
    NomLect VARCHAR(40),  
    DniLec CHAR(8),  
    DirLec VARCHAR(60),  
    UbigeoLec CHAR(6),  
    FecExpedicion DATE,  
    FecVencimiento DATE,  
    GradInstrucion VARCHAR(20),  
    FecNacLec DATE,  
    EstLec CHAR(2),  
    GenLec CHAR(1),  
    TelLec VARCHAR(15),  
    EmaLec VARCHAR(30),  
    ClaLec VARCHAR(15),  
    PRIMARY KEY (numLector)  
);
```

```
CREATE TABLE Materia (  
    CodMateria CHAR(10) NOT NULL,  
    NomMateria VARCHAR(40),  
    PRIMARY KEY (CodMateria)  
);
```

```
CREATE TABLE Tesis (  
    NomInstitucion VARCHAR(60),  
    CodMatBiblio CHAR(15) NOT NULL,  
    Tesista CHAR(5),  
    PRIMARY KEY (CodMatBiblio)  
);
```

```
CREATE TABLE Libro (  
    CodMatBiblio CHAR(15) NOT NULL,  
    ISBN CHAR(20),  
    CodEditorial CHAR(5) NOT NULL,  
    PRIMARY KEY (CodMatBiblio)  
);
```

```
CREATE TABLE Autor (  
    CodAutor CHAR(5) NOT NULL,  
    NomAutor VARCHAR(50),  
    CodPais CHAR(3) NOT NULL,  
    PRIMARY KEY (CodAutor)  
);
```

```
CREATE TABLE Devolucion (  
    FecEnt DATE,
```

```
numDev INT NOT NULL,  
EstDev CHAR(1),  
NumPre INT NOT NULL,  
PRIMARY KEY (numDev)  
);
```

```
CREATE TABLE Bibliotecario (  
ApePatBiblio VARCHAR(40),  
CodBiblio CHAR(10) NOT NULL,  
ApeMatBiblio VARCHAR(40),  
NomBiblio VARCHAR(40),  
DirBiblio VARCHAR(60),  
TelBiblio VARCHAR(15),  
FecNacBiblio DATE,  
EmaBiblio VARCHAR(30),  
ClaBiblio VARCHAR(15),  
UbigeoBiblio CHAR(6),  
EstBiblio CHAR(2),  
GenBiblio CHAR(1),  
DniBiblio CHAR(8),  
TipBiblio CHAR(1),  
PRIMARY KEY (CodBiblio)  
);
```

```
CREATE TABLE PagoInscripcion (  
NumComp CHAR(15),  
NumPagIns INT NOT NULL,  
FecPag DATE,  
Monto DOUBLE,  
numLector VARCHAR(8) NOT NULL,  
PRIMARY KEY (NumPagIns)  
);
```

```
CREATE TABLE Pais (  
CodPais CHAR(3) NOT NULL,  
NomPais CHAR(5),  
PRIMARY KEY (CodPais)  
);
```

```
CREATE TABLE Idioma (  
codIdioma CHAR(3) NOT NULL,  
nomIdioma CHAR(5),  
PRIMARY KEY (codIdioma)  
);
```

```
CREATE TABLE PedidoPrestamo (  
FecPedPre DATE,  
NumPedPre CHAR(5) NOT NULL,  
CodMatBiblio CHAR(15) NOT NULL,  
numEjemplar CHAR(5) NOT NULL,  
numLector VARCHAR(8) NOT NULL,
```

```
EstPedPre CHAR(1),
PRIMARY KEY (NumPedPre)
);

CREATE TABLE MaterialBibliografico (
  TipoMatBiblio CHAR(2),
  Titulo CHAR(5),
  CodMatBiblio CHAR(15) NOT NULL,
  anioPublica CHAR(5),
  estadoMaterial CHAR(5),
  CodAutor CHAR(5) NOT NULL,
  CodMateria CHAR(10) NOT NULL,
  codIdioma CHAR(3) NOT NULL,
  Resumen TEXT,
  NomArchivo VARCHAR(100),
  PRIMARY KEY (CodMatBiblio)
);

CREATE TABLE Editorial (
  CodEditorial CHAR(5) NOT NULL,
  NomEditorial CHAR(5),
  CodPais CHAR(5) NOT NULL,
  PRIMARY KEY (CodEditorial)
);

CREATE TABLE Sancion (
  Motivo TEXT,
  NumSan INT NOT NULL,
  FecSan DATE,
  TieSan INT,
  numDev INT NOT NULL,
  PRIMARY KEY (NumSan)
);

CREATE TABLE PagoMulta (
  NumComp CHAR(15),
  NumPagMul INT NOT NULL,
  FecPag DATETIME,
  numDev INT NOT NULL,
  Monto DOUBLE,
  PRIMARY KEY (NumPagMul)
);

CREATE TABLE Revista (
  CodMatBiblio CHAR(15) NOT NULL,
  NumPublica VARCHAR(10),
  Periodo CHAR(5),
  PRIMARY KEY (CodMatBiblio)
);
```

```
CREATE TABLE Ejemplar (  
    CodMatBiblio CHAR(15) NOT NULL,  
    numEjemplar CHAR(5) NOT NULL,  
    EstadoEjemplar CHAR(5),  
    PRIMARY KEY (CodMatBiblio,numEjemplar)  
);  
  
CREATE TABLE Prestamo (  
    NumPre INT NOT NULL,  
    EstPre CHAR(1),  
    FecPre DATETIME,  
    FecDev DATETIME,  
    CodMatBiblio CHAR(15) NOT NULL,  
    numEjemplar CHAR(5) NOT NULL,  
    numLector VARCHAR(8) NOT NULL,  
    CodBiblio CHAR(10) NOT NULL,  
    PRIMARY KEY (NumPre)  
);  
  
CREATE INDEX Lector__IDX ON Lector (null);  
  
ALTER TABLE Sancion ADD PRIMARY KEY (NumSan);  
  
ALTER TABLE Bibliotecario ADD PRIMARY KEY (CodBiblio);  
  
ALTER TABLE Autor ADD PRIMARY KEY (CodAutor);  
  
ALTER TABLE Ejemplar ADD PRIMARY KEY (CodMatBiblio, numEjemplar);  
  
ALTER TABLE MaterialBibliografico ADD PRIMARY KEY (CodMatBiblio);  
  
ALTER TABLE Editorial ADD PRIMARY KEY (CodEditorial);  
  
ALTER TABLE PagoMulta ADD PRIMARY KEY (NumPagMul);  
  
ALTER TABLE Prestamo ADD PRIMARY KEY (NumPre);  
  
ALTER TABLE Libro ADD CONSTRAINT Libro_MaterialBibliografico_FK FOREIGN  
KEY (CodMatBiblio)  
    REFERENCES MaterialBibliografico (CodMatBiblio)  
    ON DELETE CASCADE;  
  
ALTER TABLE Prestamo ADD CONSTRAINT Prestamo_Bibliotecario_FK FOREIGN  
KEY (CodBiblio)  
    REFERENCES Bibliotecario (CodBiblio)  
    ON DELETE RESTRICT;  
  
ALTER TABLE MaterialBibliografico ADD CONSTRAINT  
MaterialBibliografico_Idioma_FK FOREIGN KEY (codIdioma)  
    REFERENCES Idioma (codIdioma)  
    ON DELETE RESTRICT;
```

```
ALTER TABLE MaterialBibliografico ADD CONSTRAINT
MaterialBibliografico_Autor_FK FOREIGN KEY (CodAutor)
REFERENCES Autor (CodAutor)
ON DELETE RESTRICT;

ALTER TABLE Tesis ADD CONSTRAINT Tesis_MaterialBibliografico_FK FOREIGN
KEY (CodMatBiblio)
REFERENCES MaterialBibliografico (CodMatBiblio)
ON DELETE CASCADE;

ALTER TABLE Editorial ADD CONSTRAINT Editorial_Pais_FK FOREIGN KEY
(CodPais)
REFERENCES Pais (CodPais)
ON DELETE RESTRICT;

ALTER TABLE PagoMulta ADD CONSTRAINT PagoMulta_Devolucion_FK FOREIGN
KEY (numDev)
REFERENCES Devolucion (numDev)
ON DELETE RESTRICT;

ALTER TABLE PagoInscripcion ADD CONSTRAINT PagoInscripcion_Lector_FK
FOREIGN KEY (numLector)
REFERENCES Lector (numLector)
ON DELETE RESTRICT;

ALTER TABLE PedidoPrestamo ADD CONSTRAINT PedidoPrestamo_Ejemplar_FK
FOREIGN KEY (CodMatBiblio, numEjemplar)
REFERENCES Ejemplar (CodMatBiblio, numEjemplar)
ON DELETE RESTRICT;

ALTER TABLE Prestamo ADD CONSTRAINT Prestamo_Ejemplar_FK FOREIGN KEY
(CodMatBiblio, numEjemplar)
REFERENCES Ejemplar (CodMatBiblio, numEjemplar)
ON DELETE RESTRICT;

ALTER TABLE Devolucion ADD CONSTRAINT Devolucion_Prestamo_FK FOREIGN
KEY (NumPre)
REFERENCES Prestamo (NumPre)
ON DELETE RESTRICT;

ALTER TABLE Sancion ADD CONSTRAINT Sancion_Devolucion_FK FOREIGN KEY
(numDev)
REFERENCES Devolucion (numDev)
ON DELETE RESTRICT;

ALTER TABLE Revista ADD CONSTRAINT Revista_MaterialBibliografico_FK
FOREIGN KEY (CodMatBiblio)
REFERENCES MaterialBibliografico (CodMatBiblio)
ON DELETE CASCADE;
```

```
ALTER TABLE Libro ADD CONSTRAINT Libro_Editorial_FK FOREIGN KEY  
(CodEditorial)  
REFERENCES Editorial (CodEditorial)  
ON DELETE RESTRICT;
```

```
ALTER TABLE PedidoPrestamo ADD CONSTRAINT PedidoPrestamo_Lector_FK  
FOREIGN KEY (numLector)  
REFERENCES Lector (numLector)  
ON DELETE RESTRICT;
```

```
ALTER TABLE Prestamo ADD CONSTRAINT Prestamo_Lector_FK FOREIGN KEY  
(numLector)  
REFERENCES Lector (numLector)  
ON DELETE RESTRICT;
```

```
ALTER TABLE MaterialBibliografico ADD CONSTRAINT  
MaterialBibliografico_Materia_FK FOREIGN KEY (CodMateria)  
REFERENCES Materia (CodMateria)  
ON DELETE RESTRICT;
```

```
ALTER TABLE Autor ADD CONSTRAINT Autor_Pais_FK FOREIGN KEY (CodPais)  
REFERENCES Pais (CodPais)  
ON DELETE RESTRICT;
```

```
ALTER TABLE Ejemplar ADD CONSTRAINT Ejemplar_MaterialBibliografico_FK  
FOREIGN KEY (CodMatBiblio)  
REFERENCES MaterialBibliografico (CodMatBiblio)  
ON DELETE CASCADE;
```


3.5. FASE IV: TRANSICIÓN

3.5.1. IMPLEMENTACIÓN.

3.5.1.1. TECNOLOGÍAS.

Para poder realizar toda nuestra aplicación he utilizado distintas tecnologías y lenguajes de programación. Inicialmente he necesitado simular un servidor Web para poder utilizar nuestra aplicación y ver las respuestas que luego se darán en un servidor real, dentro de la red.

Para ello hemos optado por instalar XAMPP 1.7.3, que incorpora en un solo programa:

- Apache versión 2.2.14, es el servidor Web.
- PHP versión 5.3.1, lenguaje de programación interpretado, para webs dinámicas.
- MySQL, junto con PhpMyADMIN que nos permite la creación y gestión de nuestras bases de datos.

En nuestro diseño de interfaces he utilizado la programación con el lenguaje HTML y CSS, para darle una apariencia correcta a todas las páginas y que nos permita con el mínimo trabajo posible realizar modificaciones.

Con JavaScript he realizado las comprobaciones de nuestros formularios, para así evitar que se dejen en blanco o se rellenen con espacios en blanco, además de comprobar el correo y nombre identificativo de los usuarios que se van a registrar.

PHP ha sido el lenguaje elegido para desarrollar la lógica de la aplicación, ya que es uno de los lenguajes más utilizados en la red, el cual nos permite la realización de páginas dinámicas de una forma rápida y potente.

Para el nivel de persistencia, he utilizado MySQL, que conjuntamente con PhpMyADMIN he podido crear y administrar la base de datos de una forma cómoda y visual.

3.5.1.2. Despliegue.

DIAGRAMA DE DESPLIEGUE

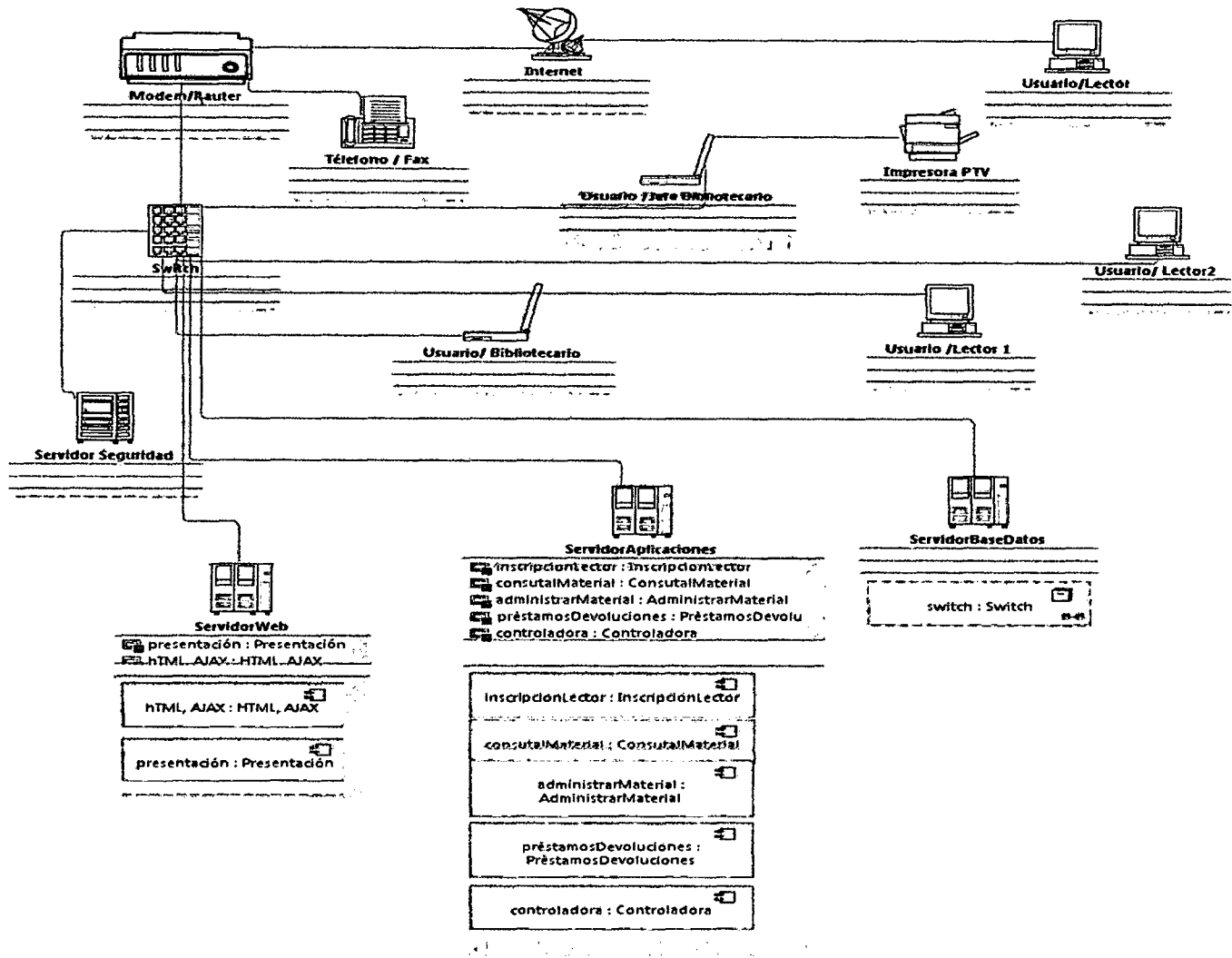


Ilustración 119: Diagrama de Despliegue

Fuente: Elaboración Propia

3.5.1.3. Detalles de Implementación.

El índice o página principal es accesible por cualquier tipo de usuario, desde aquí si el visitante ya estaba registrado podrá acceder a la opción de identificación, de lo contrario deberá efectuar la inscripción (previo requisitos como lector y Bibliotecario, Jefe de Biblioteca) en el sistema para poder proceder a las consultas del material bibliográfico o atender los préstamos y devoluciones que se ofrecen en nuestra aplicación.

Una vez el usuario identificado, si este es el jefe de Biblioteca, podrá acceder a las opciones de gestión de la aplicación. Vamos a proceder a detallar algunas de las funcionalidades más significativas que he implementado en la aplicación de nuestra Aplicación Web del Sistema de Biblioteca.

a) Acceso restringido.

El siguiente fragmento de código PHP, lo he incluido en la plantilla utilizada para crear todas las páginas correspondientes a la parte de gestión de nuestra aplicación.

Desde este código se comprueba que el usuario registrado sea administrador, en el caso de no estar registrado lo redirigimos a la página acceso_restringido.php, donde aparece un mensaje con el aviso de que no está registrado y le damos la opción con un link de poder ir a la página de identificación. Si está registrado pero, no es administrador le mostrará el menú con las opciones limitadas.

Por último si se comprueba que el usuario está registrado y es administrador no enviará al

Índice de administración, en el fichero index-admin.php.

Código conexión a la Base de Datos:

```
<?php
class MySQL{
private $conexion;
public function MySQL(){
    if(!isset($this->conexion)){
$this->conexion = (mysql_connect("localhost","biblioteca","biblio123")) or
die(mysql_error());
mysql_select_db("BDBiblioteca",$this->conexion) or die(mysql_error());
    }
}
public function consulta($consulta){
    $resultado = mysql_query($consulta,$this->conexion);
    if(!$resultado){
        echo 'MySQL Error: ' . mysql_error();
        exit;
    }
    return $resultado;
}
}
public function fetch_array($consulta)
```

```
{
    return mysql_fetch_array($consulta);
}
public function num_rows($consulta){
    return mysql_num_rows($consulta);
}
}
?>
```

Acceso_restringido.php

```
<?php
if (isset($_SESSION)) {
    session_start();
}
$MM_authorizedUsers = "administrador";
$MM_donotCheckaccess = "false";
// *** Restringir acceso: Asingar o denegar acceso a la pagina
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {
    // Para seguridad, iniciar por seguridad como usuario no autorizado
    $isValid = False;
    if (!empty($UserName)) {
        $arrUsers = Explode(",", $strUsers);
        $arrGroups = Explode(",", $strGroups);
        if (in_array($UserName, $arrUsers)) {
            $isValid = true;
        }
        if (in_array($UserGroup, $arrGroups)) {
            $isValid = true;
        }
        if (($strUsers == "") && false) {
            $isValid = true;
        }
    }
    return $isValid;
}
$MM_restrictGoTo = "../Acceso_restringido.php";
if(!((isset($_SESSION['MM_Username'])) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'],
$_SESSION['MM_UserGroup'])))) {
    $MM_qsChar = "?";
    $MM_referrer = $_SERVER['PHP_SELF'];
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
    if (isset($_QUERY_STRING) && strlen($_QUERY_STRING) > 0)
        $MM_referrer .= "?" . $_QUERY_STRING;
    $MM_restrictGoTo = $MM_restrictGoTo. $MM_qsChar . "accesscheck="
urlencode($MM_referrer);
    header("Location: ". $MM_restrictGoTo);
    exit;
}
?>
```

Mas Código ver Anexo de Código de Programas

b) Salir.

Desde el fichero salir.php destruimos las variables de sesión que hemos creado, y la sesión, enviamos al usuario a la página inicial, index.php.

```
<?php
session_start();
session_unset();
session_destroy();
header("location: index.php");
?>
```

c) Comprobación de campos.

Este pequeño script nos permite comprobar que el campo del formulario, no se encuentra vacío, ni contiene espacios en blanco, si es así continua con la siguiente parte del script, hasta llegar a la finalización del mismo.

En caso contrario lanzara un mensaje advirtiéndolo que no se han introducido los datos de forma correcta y dejando el foco en el campo correspondiente.

```
function valida_envia(){
//valido el producto
nombre1=document.fvalida.iden.value
if (nombre1==null || nombre1.length==0 || /^[^\s]+$/.test(nombre1))
{
alert("Tiene que escribir el nombre del usuario")
document.fvalida.iden.focus()
return 0;
}
}
}
```

d) Validación del correo electrónico.

La parte de este script va a comprobar que la dirección de correo introducida sea correcta, es decir que contenga un string antes del @ después otro string un punto y después del punto otro string.

Si no es así nos lanzará el mensaje de alerta, dejando el foco en el campo del correo.

Si los datos introducidos cumplen con estos parámetros continuaremos con la siguiente parte del script verificará que el siguiente campo y el del correo sean iguales, si es correcto continuará con el programa.

En caso negativo lanzaremos el mensaje de alerta avisando del error y dejando el foco en el campo erróneo.

```
valor=document.getElementById("correo").value;
var b=/^[^\s]+@[^\s]+\.[^\s]+\.[^\s]+$/
if(!(b.test(valor))){

alert("El correo no es correcto")
document.fvalida.contrasenya.focus()
return 0;
}
if (document.fvalida.correo.value!=document.fvalida.correo2.value){
alert("Los correos no son iguales")
document.fvalida.contrasenya2.focus()
return 0;
}
```

3.5.2. Evaluación y Pruebas.

3.5.2.1. Evaluación.

En esta fase de desarrollo de mi aplicación voy a evaluar tanto el funcionamiento como la usabilidad, pasando por la compatibilidad con los distintos navegadores.

Existen una serie de pautas que se deben cumplir para que el nivel de usabilidad de la aplicación sea alto:

- Debemos ponernos en lugar del usuario y tener en cuenta en que modo desearíamos encontrar la información.
- Pensar en las posibles limitaciones que el usuario a nivel de procesamiento, ancho de banda o resolución pudiera tener.

- Diseñar el grado de profundidad que daremos a nuestros contenidos y definir un árbol de navegación atractivo sin excesivos enlaces.

-Obtener un diseño visual con importante carga semántica, que transmita lo que uno propone y que cree un impacto visual sin que produzca confusión.

Con estas recomendaciones he tratado de realizar la aplicación, existen muchas recomendaciones para crear un nivel alto de usabilidad pero, para mi parecer, estas son las más importantes.

Al seguir estas recomendaciones desde el principio del proyecto la usabilidad se ha ido incorporando de un modo interactivo, iterativo e incremental. En el resultado final se nota que se ha buscado una interface sencilla, agradable y cómoda para que el usuario final se desenvuelva con soltura.

3.5.2.2. Pruebas.

A medida que he ido creando la aplicación se ha ido realizando pruebas para comprobar la navegabilidad, para ello he utilizado un programa llamado Xenu, que permite comprobar si existen enlaces rotos.

Otras pruebas a realizar son la validación de los estándares desde la página del W3C

3.5.2.2.1. Pruebas de validación.

Las pruebas de validación de formato se pueden realizar directamente desde la página de la organización W3C, nos permite validar el código XHTML y las hojas de estilo en cascada CSS.

Para realizar la validación del código lo podremos realizar desde el servidor en que resida nuestra web, copiando directamente el código en la página o mediante un upload de la misma.

La mayoría de errores que aparecen en las páginas de la aplicación, son debidos a la mezcla de código HTML con PHP, aunque son muy pocos. Lo mismo pasa con los warnings.

La hoja de estilos da algunos errores, pero tras modificar las líneas correspondientes a los avisos, pasamos sin ningún problema la validación.

Clases de Equivalencia.

a) Para ingresar el AUTOR. Existen tres Clases de Equivalencia.

1. Combinado números y letras.
2. Solamente letras.
3. Combinado números, letras y códigos ASCII.

Realizamos la prueba para la Primera clase de equivalencia.

Ingresamos los datos de Autor.

The screenshot shows a web application interface with a navigation menu at the top: Inicio, Mantenimiento, Administrar Mat. Biblio., Administrar Lectores, Bibliotecarios, Prestamos y Devoluciones, Reportes, and Cerrar Sesión. Below the menu is a form titled 'REGISTRAR AUTOR'. The form contains three fields: 'CÓDIGO' with the value 'A0006', 'NOMBRE' with the value 'HERNAN VARGAS 12', and 'PAIS/CIUDAD' with a dropdown menu showing 'Seleccione un País'. At the bottom right of the form are three buttons: 'Aceptar', 'Regresar', and 'Cancelar'.

Ilustración 120: Pantalla para Ingresar los datos de Autor

Fuente: Elaboración Propia.

No es posible ingresar datos de un autor con números.

The screenshot shows the same 'REGISTRAR AUTOR' form as in Illustration 120. However, the 'NOMBRE' field now contains 'HERNAN VARGAS 12' and the 'PAIS/CIUDAD' dropdown shows 'Seleccione un País'. Two red error messages are displayed above the form: 'EL CAMPO "NOMBRE" SOLO PUEDE TENER TEXTO' and 'DEBE ELEGIR UN VALOR DEL COMBO "CIUDADES"'. The 'Aceptar', 'Regresar', and 'Cancelar' buttons are still present at the bottom right.

Ilustración 121: Pantalla donde muestra los datos ingresados de Autor no son correctos.

Fuente: Elaboración Propia

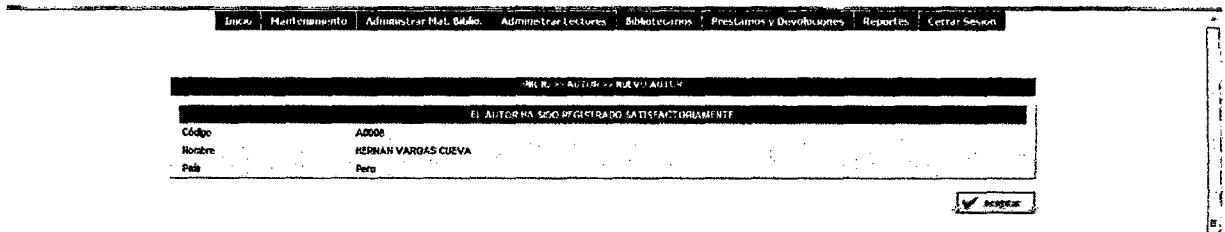
Realizamos la prueba para la Segunda clase de equivalencia.

The screenshot shows the 'REGISTRAR AUTOR' form with the 'CÓDIGO' field containing 'A0006', the 'NOMBRE' field containing 'HERNAN VARGAS CUEVA', and the 'PAIS/CIUDAD' dropdown menu showing 'Peru'. The 'Aceptar', 'Regresar', and 'Cancelar' buttons are located at the bottom right of the form.

Ilustración 122: Pantalla para ingresar los datos de Autor.

Fuente: Elaboración Propia

Acepta los datos correctamente.

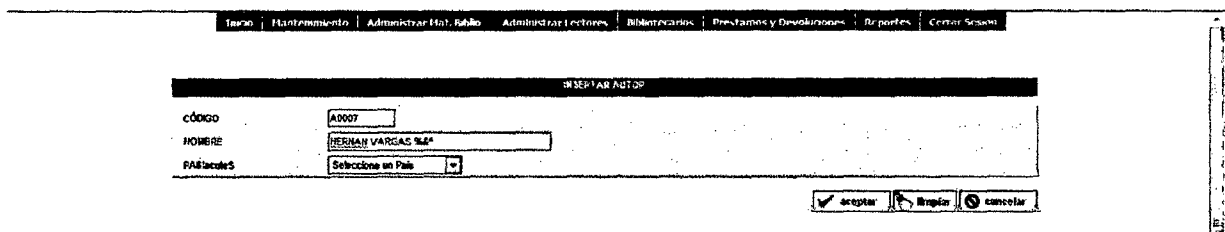


The screenshot shows a web application interface with a navigation menu at the top: Inicio, Mantenimiento, Administrar Mat. Biblio., Administrar Lectores, Bibliotecarios, Prestamos y Devoluciones, Reportes, and Cerrar Sesión. The main content area has a title bar that reads "IN. B. >> Autor >> NUEVO AUTOR". Below this, a message states "EL AUTOR HA SIDO REGISTRADO SATISFACTORIAMENTE". A table displays the entered data: Código: A0007, Nombre: HERNAN VARGAS CUEVA, and País: Peru. An "Aceptar" button is visible at the bottom right.

Ilustración 123: Pantalla donde los datos ingresados de Autor son correctos.

Fuente: Elaboración Propia

Realizamos la prueba para la Tercera clase de equivalencia

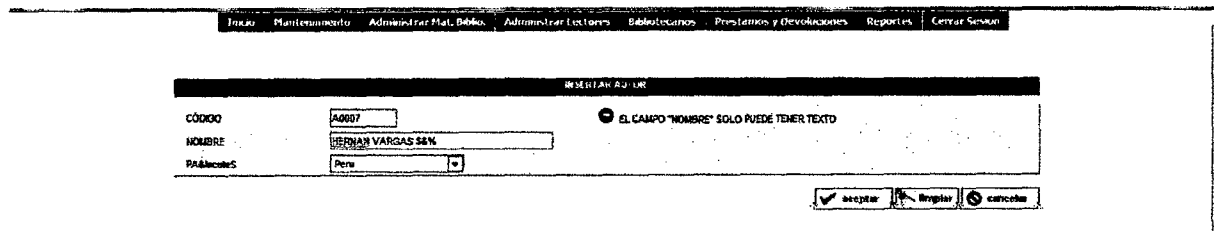


The screenshot shows the "INSERTAR AUTOR" form. The title bar reads "IN. B. >> Autor >> INSERTAR AUTOR". The form contains three fields: "CÓDIGO" with the value "A0007", "NOMBRE" with the value "HERNAN VARGAS SENA", and "PAÍSES" with a dropdown menu showing "Seleccione un País". At the bottom right, there are three buttons: "Aceptar", "Imprimir", and "Cancelar".

Ilustración 124: Pantalla para Ingresar los datos de Autor.

Fuente: Elaboración Propia

Los datos ingresados no son correctos para autor.



The screenshot shows the "INSERTAR AUTOR" form with an error message. The title bar reads "IN. B. >> Autor >> INSERTAR AUTOR". The form fields are: "CÓDIGO" with "A0007", "NOMBRE" with "HERNAN VARGAS SEN", and "PAÍSES" with "Peru". A red error message icon and text state "EL CAMPO 'NOMBRE' SOLO PUEDE TENER TEXTO". At the bottom right, there are three buttons: "Aceptar", "Imprimir", and "Cancelar".

Ilustración 125: Pantalla donde muestra que los datos ingresados de autor no son correctos.

Fuente: Elaboración Propia

b) Para reservar Material Bibliográfico por parte de un Lector.

- 1: Que el lector se encuentre Sancionado o de baja.
- 2: Que el lector se encuentre activo.

Se intenta realizar pedido, estando lector sancionado.

Panel de Usuario | Administrar | Cerrar Sesión

Cursos de programación

TÍTULO:

AUTOR:

MATERIA:

No. Material Búsqueda encontrada: Mostrados: 1-10

| ITEM | CODIGO | TITULO | AUTOR | MATERIA | CANT. EJEMPLARES |
|------|---|--------------------------------|-----------------------------------|-------------------------------|------------------|
| 1 | 612.002898272001 | CS PARA ESTUDIANTES | DOUGLAS BELL & MIKE PARR | COMPUTACION E INFORMATICA | 2 |
| 2 | MI001 | DERIVADAS | VENERO | MATEMATICA | 1 |
| 3 | MI0015/INSTRUCCIONES PARA PRESENTAR Y OFENDER | KATY J. FLORES | COMPUTACION E INFORMATICA | 4 | |
| 4 | MI002 | LIMITES DERIVADAS E INTEGRALES | VENERO | MATEMATICA | 2 |
| 5 | MI003 | REDES Y CONECTIVIDAD | UNIVERSIDAD NACIONAL DE CAJAMARCA | DESARROLLO DE SOLUCIONES JAVA | 2 |
| 6 | MI004 | JAVA 4.0 | UNIVERSIDAD NACIONAL DE CAJAMARCA | DESARROLLO DE SOLUCIONES JAVA | 0 |

Ilustración 126: Pantalla donde muestra el listado de material bibliográfico disponible.

Fuente: Elaboración Propia

El sistema no acepta que un lector que se encuentra sancionado realice una reserva.

Panel de Usuario | Administrar | Cerrar Sesión

NO PUEDE REALIZAR RESERVA DEBIDO A QUE EL USUARIO MENCIONADO SE HALLA SANCCIONADO

Código: 612.002898272001

Tipo Material Bibliográfica: Libro

Título: CS PARA ESTUDIANTES

Año de publicación: 2010

Autor: DOUGLAS BELL & MIKE PARR

Materia: COMPUTACION E INFORMATICA

Formato: Español

Reservas: No necesita tener disponibilidad previa sobre programación, ya que el libro se entrega con los conceptos de cada curso.

Archivo: CSMPARUESTUDANTE.PDF

Ilustración 127: Pantalla donde muestra que un lector sancionado, no puede realizar una reserva.

Fuente: Elaboración Propia

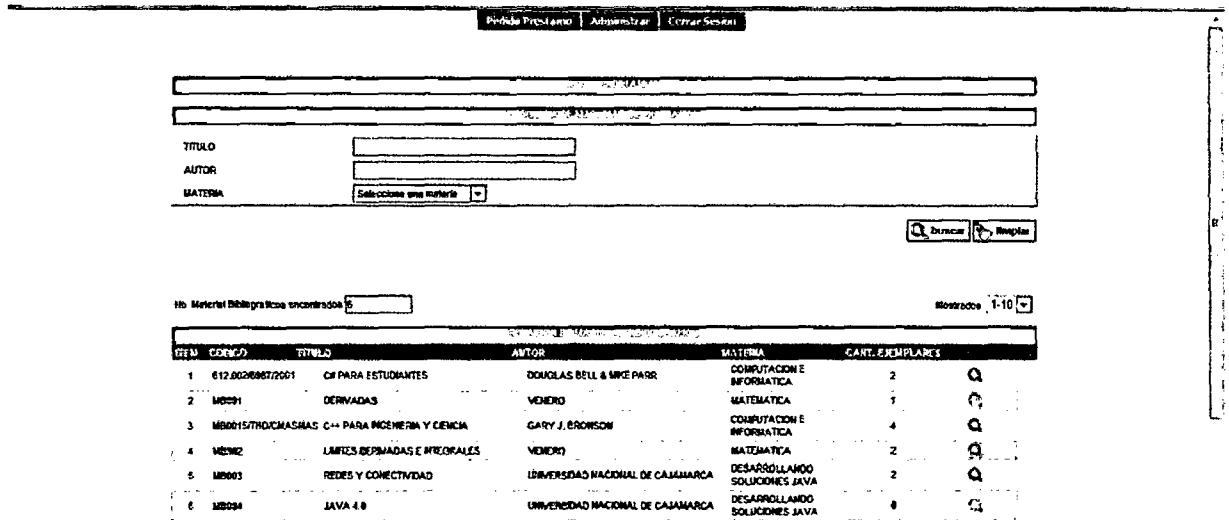


Ilustración 128: Pantalla donde muestra el listado de material bibliográfico, para ser reservado.

Fuente: Elaboración Propia

Realizar pedido de lector estando habilitado.

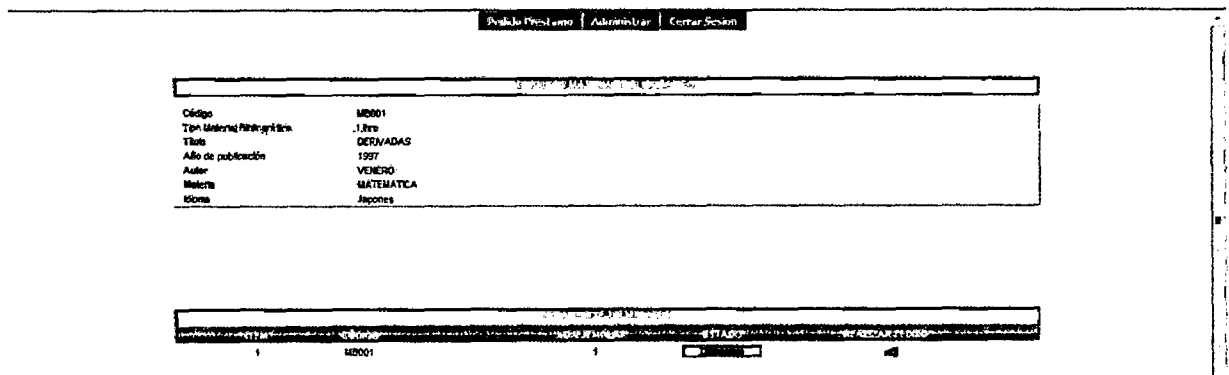


Ilustración 129: Pantalla donde muestra el material bibliográfico seleccionado para ser reservado.

Fuente: Elaboración Propia

El sistema acepta solamente la reserva por parte de un lector, que se encuentra activo.

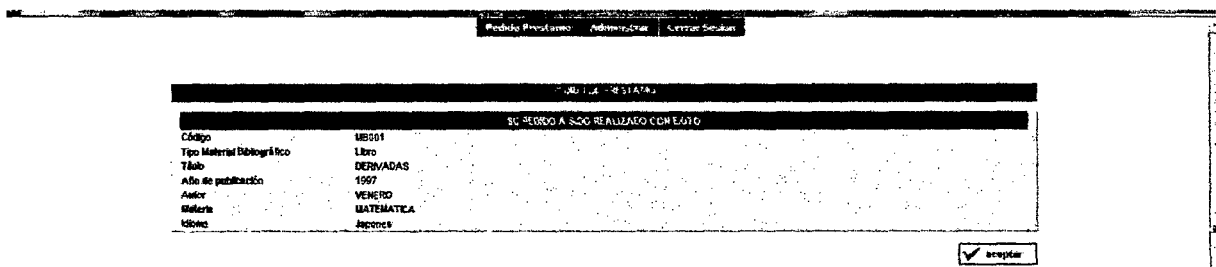


Ilustración 130: Pantalla donde muestra que la reserva realizada por un lector habilitado se realizó con éxito.

Fuente: Elaboración Propia

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES.

- En el presente Proyecto Profesional se ha utilizado el Lenguaje Unificado de Modelado (UML) apoyado en la Metodología RUP, la herramienta case IBM Rational Software Architect for Websphere Software 7.0, obteniendo como resultado final el sistema: "Para el Control de Material Bibliográfico de la Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca".
- Se logró realizar el análisis y diseño del Sistema para el Control Bibliográfico de la Biblioteca Municipal "José Gálvez" de la Municipalidad Provincial de Cajamarca, lo cual permite entre otras cosas manejar todo el proceso de Gestión Bibliotecaria, desde la Inscripción al Lector, Consulta del Material Bibliográfico, Préstamos y Devoluciones y Administración del Material Bibliográfico.
- Se implementó el Sistema de Biblioteca, basado en el análisis y diseño propuesto; con dicha implementación el servicio de biblioteca que brinda la municipalidad de Cajamarca ha mejorado considerablemente, tales como reducirá el tiempo de búsqueda de material bibliográfico y trámite obtención de información y un mejoramiento considerable de las actividades que actualmente se realiza manualmente, ya que los procesos de información serán automatizados y permitirá de una manera eficiente y confiable gestionar el proceso de Biblioteca.
- La interfaz de usuario se diseño de manera amigable, empleando la información necesaria en las pantallas con el propósito de que el usuario puede interactuar con el sistema sin dificultad.
- Antes del Funcionamiento Real del Sistema, se procedió a realizar la prueba, validaciones necesarias de la Aplicación Web del Sistema de Biblioteca, el cual funcionó de manera adecuada.

RECOMENDACIONES.

- Se recomienda al jefe y al personal involucrado en el servicio de Biblioteca de la Municipalidad Provincial de Cajamarca que al utilizar la Aplicación Web del Sistema de Biblioteca, lo realice de manera adecuada para que obtengan mayores beneficios en el desarrollo de las labores para lo cual está diseñado el sistema.
- Capacitar al personal que va utilizar el Sistema implementado, teniendo en cuenta el nivel de acceso que se tienen al mismo, para que se dé un buen uso y obtener así los resultados en beneficio de los lectores y de los ciudadanos en general.
- El personal que utilice el Sistema de Biblioteca debe apoyarse del manual de usuario para un mejor entendimiento del funcionamiento del Sistema y así no tener inconvenientes.
- Deberán brindar un mantenimiento adecuado al sistema para su buen funcionamiento.
- Realizar las Copias de Seguridad Periódicamente de la base de datos para que tengan un respaldo de los datos y no tener inconvenientes con pérdidas de información.
- El presenta proyecto profesional no ha hecho uso de patrones de diseño de software, pero en futuros mantenimientos o evoluciones de dicho sistema el diseñador podría utilizar dichos patrones

REFERENCIA BIBLIOGRÁFICA

- [1] **Aguilar, Luis Joyanes.** *Programación Orientado a Objetos.* Madrid España : McGraw Hill, 1998.
- [2] **Pressman, Roger S.** *Ingeniería del Software Un Enfoque Práctico .* Madrid : Concepción Fernandez Madrid, 1998.
- [3] **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El UML Manual de Referencia.* Madrid : Addison Wesley, 2000.
- [4] **Fowler Martin, Kendall Scout.** *UML Gota a Gota.* México : Wesley Longman de Mexico, 1999.
- [5] **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Pearson Educación S.A, 2000.
- [6] **Schach, Sthepen R.** *Analisis y Diseño Orientado a Objetos Con UML y Proceso Unificado.* Mexico : McGraw Hill Interamericana., 2005.
- [7] **Avila, César Liza.** *Modelado con UML Principios y Aplicaciones.* Perú : s.n., 2001.
- [8] **Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Hannes Magnusson, Georg Richter, Damien Seguy, Jakub Vrana.** *Manual de PHP.* Madrid : Philip Olson, 2007.
- [9] **Grady Booch, James Rumbaugh.** *The Unified Modeling Language User Guide.* USA : Addison Wesley Longman., 1999.
- [10] **Luke Welling, Laura Thomson.** *Desarrollo Web con PHP y MySQL.* Madrid. : Grupo Anaya S.A., 2005.
- [11] **Schmuller, Joseph.** *Aprendiendo UML en 24 Horas.* México : Pearson Educación., 2000.

ANEXOS

ANEXO 1:

DOCUMENTACION UTILIZADOS EN LA BIBLIOTECA

a) Carné de Lector


Parte Anterior

| | | | |
|---------------------------------------|--------------------------|--|----------|
| Municipalidad Provincial Cajamarca | | BIBLIOTECA PUBLICA MUNICIPAL "JOSÉ GÁLVEZ" CAJAMARCA | |
| TARJETA DE LECTOR | | | |
| Fecha de expedición: | 13 de setiembre del 2012 | | |
| Fecha de vencimiento: | 13 de setiembre del 2014 | | |
| N° | 26 822 | DNI: | 48367395 |
| A.P. | BRIONES | | |
| A.M. | FRIETO | | |
| Nombre | FRANKI ELI | | |
| Cajamarca, | 13 de setiembre | del 2012 | |
| Firma del lector: | | | |
| | | | |


Parte Posterior

| | | |
|--|--|-------|
| TARJETA DE LECTOR INTRASFERIBLE | | 00158 |
| Esta tarjeta debe ser presentada cada vez que el lector solicite los servicios de la Biblioteca. Deberá abonar una multa el titular que extravié esta tarjeta. | | |
| MUNICIPALIDAD PROVINCIAL DE CAJAMARCA | | |
| | | |
| MANUEL UNGAIT CUENTAS Jefe de Biblioteca Municipal Jefatura de Biblioteca | | |

b) Ficha de Préstamo.

| | | | |
|---|--|---|----------------------------------|
|  Municipalidad Provincial Cajamarca | | BIBLIOTECA PÚBLICA MUNICIPAL "JOSÉ GÁLVEZ" | |
| | | NUMERO: _____ | Materia: _____ |
| Mañana <input type="checkbox"/> | | Autor: _____ | Título: _____ |
| Tarde <input type="checkbox"/> | | Apellidos: _____ | Nombres: _____ |
| Noche <input type="checkbox"/> | | Dirección: _____ | Lug. Est/Ocup. _____ |
| SEXO | | Fecha _____ de _____ del _____ | Grado de Inst. _____ Edad: _____ |
| M <input type="checkbox"/> F <input type="checkbox"/> | | N° 004537 | |
| N° CARNET: _____ | | FIRMA DEL LECTOR(A) _____ | |

c) Ficha de Registro de Material Bibliográfico.

| | | |
|--|---|-------|
|  MUNICIPALIDAD PROVINCIAL DE CAJAMARCA | BIBLIOTECA PUBLICA MUNICIPAL "JOSE GALVEZ" PASAJE ATAHUALPA N° 628 N° Ingreso _____ | |
| | Apellidos y Nombres: _____ | |
| Titulo, Subtitulo: _____ | | _____ |
| Lugar de Impresión: _____ | | _____ |
| Editorial o Imprente: _____ | | _____ |
| Año de Imposición: _____ | | _____ |
| N° de páginas: _____ | | _____ |
| Temas Principales: _____ | | _____ |
| _____ | | _____ |

d) Sistema Dewey

El Sistema de Clasificación Decimal Dewey (también llamado CDD) es un sistema de clasificación de bibliotecas.

Fue desarrollado por Melvil Dewey, bibliotecario del Amherst College en Massachusetts, EE. UU., en 1876 y desde ese momento ha sido enormemente modificado y ampliado en el transcurso de sus veintitrés principales ediciones que han ocurrido hasta 2012. Durante este tiempo y desde 1894 también se han desarrollado 14 ediciones abreviadas, basadas en la Edición mayor desarrollada generalmente un año antes.

Clases principales

Las 10 grandes clases que lo conforman son (basadas en la Edición 21):

- 000 - Obras generales.
- 100 - Filosofía y psicología.
- 200 - Religión, teología.
- 300 - Ciencias sociales, ciencias políticas.
- 400 - Lenguaje y Lingüística.
- 500 - Ciencias puras (Matemáticas, ciencias naturales, etc)
- 600 - Ciencias aplicadas: medicina, tecnología.
- 700 - Bellas artes, juegos, deportes.
- 800 - Literatura.
- 900 - Geografía, historia.

Estas categorías principales se subdividen a su vez cada una de ellas en diez clases, en un modelo jerárquico decimal, de diversos niveles. El primer nivel (también llamado sumario) comprende estos diez grupos, en segundo sumario estaría formado por cien grupos, diez por cada uno de los diez anteriores. El tercer nivel abriría un abanico de mil posibilidades, e incluso podríamos seguir añadiendo más si fueran necesarias. Biblioteca del congreso de estados unidos Funciona como biblioteca de investigación del Congreso de Estados Unidos, y es de facto la Biblioteca Nacional de este país.

Otros niveles

Su estructura se basa en un modelo jerárquico decimal que abarca desde los temas más amplios hasta los más concretos, cada una de las diez clases principales se divide a su vez en diez Divisiones y cada una de estas en diez

secciones; Así, cada nivel inferior estará subordinado al nivel superior, algo que se denomina Fuerza Jerárquica.

800 - Literatura

880 - Literatura eslava

882 - Literatura rusa

Así sucesivamente. Como Puede observarse, cada nivel es una especialidad del anterior.

Su notación esta desarrollada completamente en números arábigos, aunque en varias partes del esquema se sugiera el uso de letras del alfabeto para la distinción en algunas temáticas, como en el caso de la literatura.

En el año 1895, Dewey permitió a los belgas Paul Otlet y Henri La Fontaine traducir y adaptar su sistema para el proyecto del Repertorio Bibliográfico Universal. De esta forma desarrollaron un esquema llamado Manual del Repertorio Bibliográfico Universal en el año 1905 del que posteriormente surge la Clasificación Decimal Universal.

El sistema Dewey es actualizado constantemente por la Online Computer Library Center (OCLC) desde 1988 Cuando adquiere los derechos al comprar la editorial Forest Press. Las últimas ediciones en inglés son la Edición 23 (2011) y la Edición Abreviada 14 (2004), las cuales son las primeras ediciones que también se desarrollaron en formato Web para consulta en línea. En español existen varias traducciones:

- Sistema de Clasificación Decimal Dewey, Edición 15 Estándar (1955)
- Clasificación Decimal Dewey: para pequeñas bibliotecas públicas y escolares (1967) (basada en la edición inglesa: Introduction to Dewey decimal classification for british schools)
- Sistema de Clasificación Decimal, basado en la 18^a edición con adiciones de la 19^a edición (1980)
- Sistema de Clasificación Decimal, Edición 20 (1995)
- Sistema de Clasificación Decimal, Edición 21 (2000)

- Clasificación Decimal Dewey Abreviada e Índice Relativa, Edición Abreviada 14 (2008)
- Sistema de Clasificación Decimal Dewey, Edición 22 (actualmente en traducción, 2010)

Curiosidades

Isaac Asimov, famoso erudito y escritor de ciencia ficción, tiene registros en todas las clasificaciones del sistema Dewey, excepto en la de Filosofía.

En 1931 para el desarrollo de la Edición 13 (1932), Melvil Dewey invitó a Ranganathan para que desarrollara los números correspondientes a la Literatura y cultura de la India, invitación que éste rechazó por estar en ese momento desarrollando su propio Sistema de Clasificación: la Clasificación colonada.

Se empieza a denominar Clasificación Decimal de Dewey a partir de la Edición 15 Estandar (1951), antes sólo era conocida como Clasificación Decimal. Siendo precisamente esta Edición la que estuvo a punto de ser la última por la crisis que se desató con su publicación.

Relación de Tablas:

| | |
|---|----|
| Tabla 1: Vistas y diagramas del UML..... | 38 |
| Tabla 2: Actores del Negocio en el Proceso de Inscripción al lector..... | 57 |
| Tabla 3: Actores del Negocio en el Proceso de Atención Consultas de Material Bibliográfico. | 60 |
| Tabla 4: Actores del Negocio en el Proceso de Atención de Préstamo y Devolución de Material Bibliográfico. | 62 |
| Tabla 5: Actores de Negocio..... | 66 |
| Tabla 6: Caso de Uso Expandido: Registrar Nuevo Lector..... | 83 |
| Tabla 7: Caso de Uso Expandido: Verifica estado de Habilidad. | 84 |
| Tabla 8: Caso de Uso Expandido: Registra Pedido de Préstamo..... | 85 |
| Tabla 9: Caso de Uso Expandido: verifica existencia y disponibilidad de Material Bibliográfico. | 86 |
| Tabla 10: Caso de Uso Expandido: Gestionar Préstamo..... | 87 |
| Tabla 11: Caso de Uso Expandido: Registrar Préstamo..... | 88 |
| Tabla 12: Caso de Uso Expandido: Gestionar Devolución. | 89 |
| Tabla 13: Caso de Uso Expandido: Verificar fecha de Devolución..... | 90 |
| Tabla 14: Caso de Uso Expandido: Actualizar estado de Material Bibliográfico. | 91 |

Relación de Ilustraciones:

| | |
|---|----|
| Ilustración 1: Proceso de desarrollo de software | 24 |
| Ilustración 2: Comparación del tiempo de desarrollo con el riesgo | 27 |
| Ilustración 3: Ciclo de vida del desarrollo Iterativo | 28 |

| | |
|---|----|
| Ilustración 4: Dimensiones del Proceso Unificado | 30 |
| Ilustración 5: Flujo de trabajo modelado del negocio | 31 |
| Ilustración 6: Proceso de Modelado | 36 |
| Ilustración 7: Representación UML de un paquete | 39 |
| Ilustración 8: Representación de una nota | 39 |
| Ilustración 9: Representación UML de una clase | 40 |
| Ilustración 10: representación de una asociación entre dos clases | 41 |
| Ilustración 11: Nombrando a una asociación | 41 |
| Ilustración 12: Cómo representar dos asociaciones entre dos clases | 41 |
| Ilustración 13: Clase de asociación | 42 |
| Ilustración 14: Asociación reflexiva | 42 |
| Ilustración 15: Representación de una agregación | 43 |
| Ilustración 16: Representación de una composición | 43 |
| Ilustración 17: Relación de dependencia | 44 |
| Ilustración 18: Diagrama de Casos de Uso | 45 |
| Ilustración 19: Notación UML de un actor | 45 |
| Ilustración 20: Notación UML de un caso de uso | 45 |
| Ilustración 21: Ejemplo de un diagrama de secuencias mostrando el conjunto básico de símbolos | 47 |
| Ilustración 22: Diagrama de colaboraciones | 48 |
| Ilustración 23: Diagrama de Actividades | 49 |
| Ilustración 24: Diagrama de Componentes | 50 |
| Ilustración 25: Diagrama de Despliegue | 51 |
| Ilustración 26: Modelo de Vistas | 55 |
| Ilustración 27: Caso de Uso del Negocio, Inscripción al Lector | 58 |
| Ilustración 28: Diagrama de Actividades, Inscripción al Lector | 59 |
| Ilustración 29: Caso de Uso del Negocio, Consulta de Material Bibliográfico | 60 |
| Ilustración 30: Diagrama de Actividades, Proceso Consultas de Material Bibliográfico | 61 |
| Ilustración 31: Caso de Uso del Negocio, Proceso Atención de Prestamos y Devoluciones del Material Bibliográfico | 63 |
| Ilustración 32: Diagrama de Actividades Proceso Atención de Prestamo y Devolución del Material Bibliográfico | 64 |
| Ilustración 33: Caso de uso del Negocio, Proceso de Administrar el Material Bibliográfico | 67 |
| Ilustración 34: Diagrama de Actividades Proceso de Administrar el Material Bibliográfico | 68 |
| Ilustración 35: Modelo de Objetos del Negocio | 69 |
| Ilustración 36: Proceso General de Consulta y Atención de Material Bibliográfico | 70 |
| Ilustración 37: Arquitectura en Capas | 73 |
| Ilustración 38: Casos de Uso Inscripción al Lector | 77 |
| Ilustración 39: , Casos de Uso Consulta Material Bibliográfico | 78 |
| Ilustración 40: Atención de Préstamos y Devoluciones | 79 |
| Ilustración 41: Casos de Uso Administración | 80 |
| Ilustración 42: Casos de Uso Seguridad | 81 |
| Ilustración 43: Diagrama de Interfaces | 92 |
| Ilustración 44: Diagrama de Controles | 93 |
| Ilustración 45: Diagrama de Secuencia, Inscribir nuevo lector | 94 |
| Ilustración 46: Diagrama de Secuencia, Actualizar Datos Personales del Lector | 95 |

| | |
|--|-----|
| Ilustración 47: Diagrama de Secuencia, Verificar Estado de Habilidad | 96 |
| Ilustración 48: Diagrama de Secuencia, Consultar Material Bibliográfico | 97 |
| Ilustración 49: Diagrama de Secuencia, Registra pedido Préstamo | 98 |
| Ilustración 50: Diagrama de Secuencia, Atender pedido de Préstamo | 99 |
| Ilustración 51: Diagrama de Secuencia, Gestionar y Registra Préstamo | 100 |
| Ilustración 52: Diagrama de Secuencia, Gestionar Devoluciones..... | 101 |
| Ilustración 53: Diagrama de Secuencia, Elaborar Pedido de Compra | 102 |
| Ilustración 54: Diagrama de Secuencia, Recepciona y Registra Material Bibliográfico | 103 |
| Ilustración 55: Diagrama de Secuencia, Acceder al Sistema..... | 104 |
| Ilustración 56: Diagrama de Secuencia, Auditoria | 104 |
| Ilustración 57: Diagrama de Clases | 105 |
| Ilustración 58: Diagrama de Paquetes de Diseño | 106 |
| Ilustración 59: Diagrama de Clases de Diseño para Seguridad Sistema | 107 |
| Ilustración 60: Diagrama de Clases de Diseño para Administrar Lectores. | 108 |
| Ilustración 61: : Diagrama de Clases de Diseño para Consultar Material Bibliográfico. | 108 |
| Ilustración 62: : Diagrama de Clases de Diseño para Administrar Material Bibliográfico. | 109 |
| Ilustración 63: Diagrama de Clases de Diseño para Atención de Préstamos y Devoluciones. | 110 |
| Ilustración 64: Diagrama de Secuencia de Diseño, Inscribir nuevo lector..... | 111 |
| Ilustración 65: Diagrama de Secuencia, Buscar Material Bibliográfico | 112 |
| Ilustración 66: Diagrama de Secuencia, Registra pedido Préstamo | 113 |
| Ilustración 67: Diagrama de Secuencia, Atender pedido de Préstamo | 114 |
| Ilustración 68: Diagrama de Secuencia, Gestionar y Registra Préstamo | 114 |
| Ilustración 69: Diagrama de Secuencia, Elaborar Pedido de Compra | 115 |
| Ilustración 70: Diagrama de Secuencia, Recepciona y Registra Material Bibliográfico | 116 |
| Ilustración 71: Diagrama de Secuencia, Acceder al Sistema..... | 116 |
| Ilustración 72: Diagrama de Secuencia, Auditoria | 117 |
| Ilustración 73: Figura N°, Diagrama de Componentes | 118 |
| Ilustración 74: Diseño de pantalla de inicio | 119 |
| Ilustración 75: Diseño de pantalla Buscar y Mantenimiento de Lector | 120 |
| Ilustración 76: Diseño de pantalla Insertar nuevo Lector | 120 |
| Ilustración 77: Diseño de pantalla Modificar Lector..... | 121 |
| Ilustración 78: Diseño de pantalla Guardar Datos del Lector Modificado | 121 |
| Ilustración 79: Diseño de pantalla Buscar y Mantenimiento de Pedido Préstamos | 122 |
| Ilustración 80: Diseño de pantalla Detalle del Material Bibliográfico a realizar Pedido de Préstamo..... | 122 |
| Ilustración 81: Diseño de pantalla Pedido de Préstamo..... | 123 |
| Ilustración 82: Diseño de pantalla Buscar y Mantenimiento de Préstamos | 123 |
| Ilustración 83: Diseño de pantalla Buscar Lector | 124 |
| Ilustración 84: Diseño de pantalla Realizar Préstamo..... | 124 |
| Ilustración 85: Diseño de pantalla Guarda Préstamo..... | 125 |
| Ilustración 86: Diseño de pantalla Retorna despues de Realizar Pedido de Préstamo..... | 125 |
| Ilustración 87: Diseño de pantalla Buscar y atender devolución de Préstamo .. | 126 |
| Ilustración 88: Diseño de pantalla Realizar Devolución..... | 126 |
| Ilustración 89: Diseño de pantalla Buscar y Mantenimiento de Lector | 127 |

| | |
|--|-----|
| Ilustración 90: Diseño de pantalla Buscar y Retrona Préstamos Pendientes.... | 127 |
| Ilustración 91: Diseño de pantalla Buscar y Mantenimiento de Pago de Inscripción..... | 128 |
| Ilustración 92: Diseño de pantalla Insertar pago de Inscripción | 128 |
| Ilustración 93: Diseño de pantalla Buscar y Mantenimiento de Bibliotecario | 128 |
| Ilustración 94: Diseño de pantalla Insertar Bibliotecario..... | 129 |
| Ilustración 95: Diseño de pantalla Modificar Bibliotecario | 129 |
| Ilustración 96: Diseño de pantalla Confirmar el Grabado de la Modificación | 130 |
| Ilustración 97: Diseño de pantalla Buscar y Mantenimiento de Autores..... | 130 |
| Ilustración 98: Diseño de pantalla Modificar Autor | 131 |
| Ilustración 99: Diseño de pantalla Insertar Autor | 131 |
| Ilustración 100: Diseño de pantalla Buscar Modificar Autor | 131 |
| Ilustración 101: Diseño de pantalla Buscar y Mantenimiento de Editorial | 131 |
| Ilustración 102: Diseño de pantalla Insertar Editorial | 132 |
| Ilustración 103: Diseño de pantalla Modificar Editorial..... | 132 |
| Ilustración 104: Diseño de pantalla Buscar y Mantenimiento de Editorial | 132 |
| Ilustración 105: Diseño de pantalla Buscar y Mantenimiento de Materia | 132 |
| Ilustración 106: Diseño de pantalla Insertar Materia | 133 |
| Ilustración 107: Diseño de pantalla Modificar Materia..... | 133 |
| Ilustración 108: Diseño de pantalla guardar Datos de Materia..... | 133 |
| Ilustración 109: Diseño de pantalla Buscar y Mantenimiento de Material Bibliográfico | 133 |
| Ilustración 110: Diseño de pantalla Insertar Material Bibliográfico..... | 134 |
| Ilustración 111: Diseño de pantalla Modificar Material Bibliográfico | 134 |
| Ilustración 112: Diseño de pantalla Guardar Datos de Material Bibliográfico.... | 135 |
| Ilustración 113: Diseño de pantalla Buscar y Mantenimiento de Libros | 135 |
| Ilustración 114: Diseño de pantalla de Agregar Libro..... | 135 |
| Ilustración 115: Diseño de pantalla Buscar Material Bibliográfico..... | 136 |
| Ilustración 116: Diseño de pantalla Buscar y Mantenimiento de Revistas | 136 |
| Ilustración 117: Diseño de pantalla de Tesis..... | 137 |
| Ilustración 118: Diagrama de Entidad Relación. | 139 |
| Ilustración 119: Diagrama de Despliegue | 137 |
| Ilustración 120: Pantalla para Ingresar los datos de Autor..... | 143 |
| Ilustración 121: Pantalla donde muestra los datos ingresados de Autor no son correctos. | 143 |
| Ilustración 122: Pantalla para ingresarlos datos de Autor..... | 143 |
| Ilustración 123: Pantalla donde los datos ingresados de Autor son correctos. . | 144 |
| Ilustración 124: Pantalla para Ingresar los datos de Autor..... | 144 |
| Ilustración 125: Pantalla donde muestra que los datos ingresados de autor no son correctos. | 144 |
| Ilustración 126: Pantalla donde muestra el listado de material bibliográfico disponible..... | 145 |
| Ilustración 127: Pantalla donde muestra que un lector sancionado, no puede realizar una reserva. | 145 |
| Ilustración 128: Pantalla donde muestra el listado de material bibliográfico, para ser reservado. | 146 |
| Ilustración 129: Pantalla donde muestra el material bibliográfico seleccionado para ser reservado..... | 146 |
| Ilustración 130: Pantalla donde muestra que la reserva realizada por un lector habilitado se realizo con éxito. | 147 |

1. Implementación de Código PHP para Acceso Al Sistema.

1.1. Clase Entidad

Archivo: UsuarioSistema.php

```
<?php
include("conexion.php");
class UsuarioSistema{
    private $DNI;
    private $ApePat;
    private $AoeMat;
    private $NomPer;
    private $Direccion;
    private $Telefono;
    private $FecNac;
    private $Email;
    private $Clave;
    private $Ubigeo;
    private $Estado;
    private $Genero;

    //Set y Get de la clase

    function setDNI($DNI) {
        $this->DNI = $DNI;
    }
    function setApePat($ApePat) {
        $this->ApePat = $ApePat;
    }
    function setApeMat($ApeMat) {
        $this->ApeMat = $ApeMat;
    }
    function setNomPer($NomPer) {
        $this->NomPer = $NomPer;
    }
    function setDireccion($Direccion) {
        $this->Direccion = $Direccion;
    }
    function setTelefono($Telefono) {
        $this->Telefono = $Telefono;
    }
    function setFecNac($FecNac) {
        $this->FecNac = $FecNac;
    }
    function setEmail($Email) {
        $this->Email = $Email;
    }
    function setClave($Clave) {
        $this->Clave = $Clave;
    }
    function setUbigeo($Ubigeo) {
        $this->Ubigeo = $Ubigeo;
    }
    function setEstado($Estado) {
        $this->Estado = $Estado;
    }
    function setGenero($Genero) {
        $this->Genero = $Genero;
    }
}

function getDNI() {
```



```
        return $this->DNI;
    }
    function getApePat() {
        return $this->ApePat;
    }
    function getApeMat() {
        return $this->ApeMat;
    }
    function getNomPer() {
        return $this->NomPer;
    }
    function getDireccion() {
        return $this->Direccion;
    }
    function getTelefono() {
        return $this->Telefono;
    }
    function getFecNac() {
        return $this->FecNac;
    }
    function getEmail() {
        return $this->Email;
    }
    function getClave() {
        return $this->Clave;
    }
    function getUbigeo() {
        return $this->Ubigeo;
    }
    function getEstado() {
        return $this->Estado;
    }
    function getGenero() {
        return $this->Genero;
    }
}
}
```

```
class Lector extends UsuarioSistema{

    private $NumLector;
    private $FecExpedicion;
    private $FecVencimiento;
    private $GradInstruccionLec;
    //Set y Get de la clase

    function setNumLector($NumLector) {
        $this->NumLector = $NumLector;
    }
    function setFecExpedicion($FecExpedicion) {
        $this->FecExpedicion = $FecExpedicion;
    }
    function setFecVencimiento($FecVencimiento) {
        $this->FecVencimiento = $FecVencimiento;
    }
    function setGradInstruccionLec($GradInstruccionLec) {
        $this->GradInstruccionLec = $GradInstruccionLec;
    }
    function getNumLector() {
        return $this->NumLector;
    }
    function getFecExpedicion() {
```

```
        return $this->FecExpedicion;
    }
    function getVencimiento() {
        return $this->Vencimiento;
    }
    function getGradInstruccionLec() {
        return $this->GradInstruccionLec;
    }
}

class Bibliotecario extends UsuarioSistema{

    private $CodBiblio;
    private $TipoBiblio;
    //Set y Get de la clase

    function setCodBiblio($CodBiblio) {
        $this->CodBiblio = $CodBiblio;
    }
    function setTipoBiblio($TipoBiblio) {
        $this->TipoBiblio=$TipoBiblio;
    }
    function getCodBiblio() {
        return $this->CodBiblio;
    }
    function getTipoBiblio() {
        return $this->TipoBiblio;
    }
}

?>
```

1.2. Clase Control

Archivo: GestorUsuario.php

```
<?php
include("conexion.php");
$conexion= new conexion();
class GestorUsuario {

    //Metodo utilizado para buscar un Usuario, de tipo Lector, Bibliotecario o
    Jefe Bibliotecario
    function buscar($_USER,$_PASSWORD,$_ID_USER) {
        /**
        * Primero buscamos dentro de la tabla lector:
        */

        $consulta="SELECT COUNT(*) as filas FROM lector WHERE emalec='$_USER' AND
        clalect='$_PASSWORD'";
        $query_lector=mysql_query($consulta);
        if(mysql_result($query_lector,0,"filas")==0){ //no encotro el lector
            /**
            * Aca buscamos dentro de la tabla bibliotecario:
            */
            $consulta="SELECT COUNT(*) as filas FROM bibliotecario WHERE
            EmaBiblio='$_USER' AND ClaBiblio='$_PASSWORD'";
            $query_bibliotecario=mysql_query($consulta);
            if(mysql_result($query_bibliotecario,0,"filas")==0){//no se encontro el
            bibliotecario

            return 0; //no se encotro a ninguno de los tres..
        }
    }
}
```

```
    }else{//se encontro bibliotecario
        $consulta="SELECT * FROM bibliotecario WHERE EmaBiblio='$_USER'
AND ClaBiblio='$_PASSWORD'";
        $query_bibliotecario=mysql_query($consulta);
        $_NOMBRE_USER = mysql_result($query_bibliotecario,0,"ApePatBiblio")."
        =
        ".mysql_result($query_bibliotecario,0,"ApeMatBiblio")."
        ".mysql_result($query_bibliotecario,0,"NomBiblio");
        $_ID_USER = mysql_result($query_bibliotecario,0,"codbiblio");
        return mysql_result($query_bibliotecario,0,"TipBiblio");
//retorna el tipo de bibliotecario--> 1:bibliotecario 2:jefebibliotecario
    }
    }else{ //SI encontro un lector
        $consulta="SELECT * FROM lector WHERE EmaLec='$_USER' AND
ClaLec='$_PASSWORD'";
        $query_lector=mysql_query($consulta);
        $_NOMBRE_USER = mysql_result($query_lector,0,"ApePatLec")."
        ".mysql_result($query_lector,0,"ApeMatLec")."
        ".mysql_result($query_lector,0,"NomLect");
        $_ID_USER = mysql_result($query_lector,0,"numLector");
        return 3;
    }
}

//Metodo para verificar el tipo de Usuario que accede
function VerificarLogeo($_USER,$_PASSWORD) {
    $_NOMBRE_USER="";
    $_ID_USER="";
    if(!empty($_USER) and !empty($_PASSWORD)){
        $resp = Buscar();
        switch($resp){
            case (0): //no hay lector ni bibliotecario
                $url = "index.php";
                break;
            case (1): // bibliotecario
                $_SESSION["Biblio"] = $_ID_USER;
                $_SESSION["ID"] = $_ID_USER;
                $_SESSION["USER"] = $_NOMBRE_USER;
                $url = "principal.php";
                break;
            case (2): //jefe bibliotecario
                $_SESSION["JefeBiblio"] = $_ID_USER;
                $_SESSION["ID"] = $_ID_USER;
                $_SESSION["USER"] = $_NOMBRE_USER;
                $url = "principal.php";
                break;
            case (3): //lector
                $_SESSION["Lector"] = $_ID_USER;
                $_SESSION["ID"] = $_ID_USER;
                $_SESSION["USER"] = $_NOMBRE_USER;
                $url = "principal.php";
                break;
        }
        Acceder();
    }
}
}else{
    DespliegaMensajeIncorrecto();
    CerrarPagina();
}
}
```

```
    }
    function Acceder()
    {
        session_start();
        if(isset($_SESSION["JefeBiblio"])    or    isset($_SESSION["Biblio"])    or
isset($_SESSION["Lector"]))
        header("location: principal.php");
    }

    function CerrarPagina()
    {
        echo    "<script    lenguaje='javascript'
type='text/javascript'>window.close();</script>";
    }

    function DespliegaMensajeIncorrecto()
    {
        header("location: index.php");
    }

}
?>
```

1.3. Clases Interface

Archivo: frmAcceder.php

```
<?
header('Cache-Control: no-cache');
header('Pragma: no-cache');
session_start();

function AlmacenaDatosLogeo()
{ include("Entidad/UsuarioSistema.php");
  $Usuario = new UsuarioSistema();
  $Usuario->setEmail(loggear.usuario);
  $Usuario->setClave(loggear.password);
  include("Control/GestorUsuario.php");
  $GestorUsuario = new GestorUsuario();
  $GestorUsuario.VerificaLogeo($Usuario->setEmail(),$Usuario->setClave());
}
function CancelarLogeo()
{
    header("location:index.php");
}

if(isset($_SESSION["JefeBiblio"])    or    isset($_SESSION["Biblio"])    or
isset($_SESSION["Lector"]))
    header("location: principal.php");
?>
<html>
<head>
    <title>Sistema Biblioteca Web</title>
    <style type="text/css">
        body {
            background-color: rgb(255, 255,255);
            margin: 0px;
        }
    .paginaindex{
```

```
        position: absolute;
        background-color:#2C81A9;
        margin: 0px;
        width: 100%;
    }
    .formulario,.logear{
        color: white;
        margin: 4px;
        font-family: helvetica;
        font-size:9pt;
    }
    .input{
        border:solid 1px #000;
        background-color:#fff;
        width:150px;
        font-size:8pt;
    }
</style>
</head>
<body>
<div class="paginaindex" align="center">
    <form id="logear" name="logear" method="post" action="AlmacenarDatosLogeo()"
class="formulario">
        <table class="fuente8" cellspacing=0 cellpadding=3 border=0>
            <tr class="logear">
                <td align="center">
                    USUARIO
                    <input name="usuario" id="usuario" maxlength="50"
type="text" class="input"/>
                </td>
                <td align="center">
                    CONTRASE&Ntilde;A
                    <input name="password" id="password" maxlength="15"
type="password" class="input"/>
                </td>
                <td align="center">
                    <input class="entrar" type="submit" value="Acceder"/>
                </td>
                <td align="center">
                    <input class="cancelar" type="cancel" value="Cancelar"
onclick="CancelarLogeo()"/>
                </td>
            </tr>
        </table>
    </form>
</div>
<iframe src="inicio.php" name="principal" title="principal" width="100%" height="90%"
frameborder=0 scrolling="no" style="margin-left: 0px; margin-right: 0px; margin-top:
2px; margin-bottom: 0px;"></iframe>
</body>
</html>
Archivo: frmMenuSistema
<?
header('Cache-Control: no-cache');
header('Pragma: no-cache');
session_start();
function MostrarPaginaIndex()
{
if(isset($_SESSION['Biblio']))
    $_TIPO = 1;
elseif(isset($_SESSION['JefeBiblio']))
```

```
        $_TIPO = 2;
elseif(isset($_SESSION["Lector"]))
    $_TIPO = 3;
else
    header("location:index.php");
}
MostrarPaginaIndex();
?>
<html>
<head>
    <title>Biblioteca</title>
    <script language="JavaScript" src="menu/JSCookMenu.js"></script>
    <link rel="stylesheet" href="menu/theme.css" type="text/css"/>
    <script language="JavaScript" src="menu/theme.js"></script>
    <?php
        switch($_TIPO){
            case (1):{?>
                <script language="JavaScript">
                    var MenuPrincipal = [
                        [null,'Inicio','inicio.php','principal','Inicio'],
                        [null,'Administrar Mat. Biblio.',null,null,'Materiales
Bibliograficos',
                                [null,'Material
Bibliografico','./materialbibliografico/index.php','principal','Materiales
Bibliograficos'],
                                [null,'Ejemplar','./ejemplar/index.php','principal','Ejemplares']
                                ],
                        [null,'Administrar Lectores',null,null,'Lectores',
                                [null,'Lectores','./lector/index.php','principal','Lectores'],
                                [null,'Pago
Inscripcion','./pagoInscripcion/index.php','principal','Pagos de Inscripcion']
                                ],
                        [null,'Prestamos y Devoluciones',null,null,'Prestamos y
Devoluciones',
                                [null,'Prestamos','./prestamo/index.php','principal','Prestamos'],
                                [null,'Devoluciones','./devolucion/index.php','principal','Devoluciones'],
                                [null,'Sanciones','./sancion/index.php','principal','Sanciones'],
                                [null,'Pago
Multa','./pagomulta/index.php','principal','Pagos por Multa']
                                ],
                        [null,'Cerrar Sesion','cerrarsesion.php',null,'Cerrar
Sesion']
                                ];</script>
                <?
                $inicio = "inicio.php";
                break;
            }
            case (2):{?>
                <script language="JavaScript">
                    var MenuPrincipal = [
                        [null,'Inicio','inicio.php','principal','Inicio'],
                        [null,'Mantenimiento',null,null,'Administrar',
                                [null,'Autor','./autor/index.php','principal','Autores'],
```

```
[null,'Editorial','./editorial/index.php','principal','Editoriales'],
[null,'Libro','./libro/index.php','principal','Libros'],
[null,'Materia','./materia/index.php','principal','Materias'],
[null,'Revista','./revista/index.php','principal','Revistas'],
[null,'Tesis','./tesis/index.php','principal','Tesis']
],
[null,'Administrar Mat. Biblio.',null,null,'Materiales
Bibliograficos',
[null,'Material
Bibliografico','./materialbibliografico/index.php','principal','Materiales
Bibliograficos'],
[null,'Ejemplar','./ejemplar/index.php','principal','Ejemplares']
],
[null,'Administrar Lectores',null,null,'Lectores',
[null,'Lectores','./lector/index.php','principal','Lectores'],
[null,'Pago
Inscripcion','./pagoInscripcion/index.php','principal','Pagos de Inscripcion']
],
[null,'Bibliotecarios','bibliotecario','principal','Bibliotecarios'],
[null,'Prestamos y Devoluciones',null,null,'Prestamos y
Devoluciones',
[null,'Prestamos','./prestamo/index.php','principal','Prestamos'],
[null,'Devoluciones','./devolucion/index.php','principal','Devoluciones'],
[null,'Sanciones','./sancion/index.php','principal','Sanciones'],
[null,'Pago
Multa','./pagomulta/index.php','principal','Pagos por Multa']
],
[null,'Reportes',null,null,'Reportes',
[null,'Inventario
Libros','./reporte_inventarios/index.php','principal','Inventarios de Libros'],
[null,'Reporte
Prestamos','./reporte_prestamos/index.php','principal','Reporte de Libros prestados']
],
[null,'Cerrar Sesion','cerrarsesion.php',null,'Cerrar
Sesion']
];</script>
<?
$inicio = "inicio.php";
break;
}
case (3):{?>
<script language="JavaScript" type="text/javascript">
var MenuPrincipal = [
[null,'Pedido
Prestamo','./pedido_prestamo/index.php','principal','Pedido Prestamo'],
```

```
[null, 'Administrar', './lector/modificar_lector1.php', 'principal', 'Adminstrar']
,
          [null, 'Cerrar Sesion', 'cerrarsesion.php', null, 'Cerrar
Sesion']
]
</script><?
$inicio = "./pedido_prestamo/index.php";
break;
}
}
?>

<style type="text/css">
body { background-color: rgb(255, 255,255);
background-image: url(images/superior.png);
background-repeat: no-repeat;
margin: 0px;
}

#MenuAplicacion { margin-left: 10px;
margin-top: 0px;
}
</style>
</head>
<body>
<div id="MenuAplicacion" align="center"></div>
<script language="JavaScript">
cmDraw ('MenuAplicacion', MenuPrincipal, 'hbr', cmThemeGray, 'ThemeGray');
</script>
<iframe src="<?echo $inicio ?>" name="principal" title="principal" width="100%"
height="1050px" frameborder=0 scrolling="no" style="margin-left: 0px; margin-right:
0px; margin-top: 2px; margin-bottom: 0px;"></iframe>
</body>
</html>
```

2. Implementación de Código PHP para Realizar Atención de Préstamos.

2.1. Clase Entidad.

Archivo: PedidoPrestamo.php

```
<?
include("conexion.php");
class PedidoPrestamo{
private $NumPedPres;
private $EstadoPedido;
private $FecPedido;

//Set y Get de la clase

function setNumPedPres($NumPedPres) {
$this->NumPedPres =$NumPedPres;
}

function setEstadoPedido($EstadoPedido) {
$this->EstadoPedido =$EstadoPedido;
}

function setFecPedido($FecPedido) {
```



```
        $this->FecPedido = $FecPedido;
    }

    function getNumPedPres() {
        return $this->NumPedPres;
    }

    function getEstadoPedido() {
        return $this->EstadoPedido;
    }
    function getFecPedido() {
        return $this->FecPedido;
    }
}

?>
```

Archivo: Lector.php

```
<?php
include("conexion.php");
class Lector{
    private $NumLector;
    private $FecExpedicion;
    private $FecVencimiento;
    private $GradInstruccionLec;

    function setNumLector($NumLector) {
        $this->NumLector = $NumLector;
    }
    function setFecExpedicion($FecExpedicion) {
        $this->FecExpedicion = $FecExpedicion;
    }
    function setFecVencimiento($FecVencimiento) {
        $this->FecVencimiento = $FecVencimiento;
    }
    function setGradInstruccionLec($GradInstruccionLec) {
        $this->GradInstruccionLec = $GradInstruccionLec;
    }
}
}
```

Archivo: Ejemplar.php

```
<?php
include("conexion.php");
class Ejemplar{
    private $NumEjemplar;
    private $EstadoEjemplar;
    private $Condicion;
    private $Ubicacion;
    //Set y Get de la clase

    function setNumEjemplar($NumEjemplar) {
        $this->NumEjemplar = $NumEjemplar;
    }
    function setEstadoEjemplar($EstadoEjemplar) {
        $this->EstadoEjemplar = $EstadoEjemplar;
    }
    function setCondicion($Condicion) {
        $this->Condicion = $Condicion;
    }
    function setUbicacion($Ubicacion) {
        $this->Ubicacion = $Ubicacion;
    }
}
```

```
function getNumEjemplar() {
    return $this->NumEjemplar;
}
function getEstadoEjemplar() {
    return $this->EstadoEjemplar;
}
function getCondicion() {
    return $this->Condicion;
}
function getUbicacion() {
    return $this->Ubicacion;
}
}
?>
Archivo: Prestamo.php
<?php
include("conexion.php");
class Prestamo{
    private $NumPrestamo;
    private $EstadoPrestamo;
    private $FecPrestamo;
    private $FecDevolucion;

    function setNumPrestamo($NumPrestamo) {
        $this->NumPrestamo = $NumPrestamo;
    }
    function setEstadoPrestamo($EstadoPrestamo) {
        $this->EstadoPrestamo = $EstadoPrestamo;
    }
    function setFecPrestamo($FecPrestamo) {
        $this->FecPrestamo = $FecPrestamo;
    }
    function setFecDevolucion($FecDevolucion) {
        $this->FecDevolucion = $FecDevolucion;
    }
    function getNumPrestamo() {
        return $this->NumPrestamo;
    }
    function getEstadoPrestamo() {
        return $this->EstadoPrestamo;
    }
    function getFecPrestamo() {
        return $this->FecPrestamo;
    }
    function getFecDevolucion() {
        return $this->FecDevolucion;
    }
}
?>
```

2.2. Clase Control.

Archivo: GestorBusquedaPedidoPrestamo.php

```
class GestorBusquedaPedidoPrestamo {
    //Metodo utilizado para obtener los pedidos Pendientes

    function ConsultaPedidoPendientes($titulo,$autor,$lector)
    {
        $PedidoPrestamo = new PedidoPrestamo();
    }
}
```

```
$cadena_busqueda=$_POST["cadena_busqueda"];

$where="1=1";
if ($titulo <> "") { $where.=" AND materialbibliografico.titulo LIKE
'%'$titulo%' ";}
if ($autor <> "") { $where.=" AND autor.nomautor LIKE '%$autor%' ";}
if ($lector <> "") {$where.=" AND lector.numlector='$lector'";}

$where.= " AND pedidoprestamo.estpedpre='1'";
$inner_join = " AND autor.codautor=materialbibliografico.codautor AND
materialbibliografico.codmatbiblio=ejemplar.codmatbiblio AND
ejemplar.numejemplar=pedidoprestamo.numejemplar AND
lector.numlector=pedidoprestamo.numlector";
$where.= $inner_join." ORDER BY fecpedpre ASC";

$query_busqueda="SELECT count(*) as filas FROM
materialbibliografico,autor,ejemplar,pedidoprestamo,lector WHERE ".$where;
$rs_busqueda=mysql_query($query_busqueda);
$filas=mysql_result($rs_busqueda,0,"filas");

return $filas;
}

function BuscarDisponibilidadMaterialBibliografico($MaterialBibliografico)
{
    $sql = "SELECT * FROM MaterialBibliografico,Ejemplar WHERE
MaterialBibliografico.CodMatBiblio=Ejemplar.CodMatBiblio and Ejemplar.condicion=1 and
CodMatBiblio=$MaterialBibliografico->GetCodMatBiblio";
    try {
        //Ejecutamos la sentencia
        $rs = mysql_query($sql);
        //Creamos un array que almacenara los datos de la sentencia
        $registros = array();
        //Recorremos el resultado de la consulta y lo almacenamos en el array
        while ($reg = mysql_fetch_array($rs)) {
            array_push($registros, $reg);
        }
        //Liberamos recursos
        mysql_free_result($rs);
        mysql_close($cn);
    } catch (exception $e) {
        try {
            mysql_free_result($rs);
        } catch (exception $e) {
        }
        try {
            mysql_close($cn);
        } catch (exception $e) {
        }
    }
    return $registros;
}
}
```

Archivo: AdministrarPrestamo.php

```
<?php
include("conexion.php");
function ActualizarEstadoLector($numlector)
{
    $Lector=new Lector();
    $Lector->setNumLector($numlector);
    //2 estado Prestado
    $Lector->setEstado('2');
    $rpta=false;
    try {
        //Creamos un objeto de la clase conexion
        $miconexion = new Conexion();
        //Obtenemos la conexion
        $cn = $miconexion->conectar();
        //Comenzamos la transaccion
        mysql_query("BEGIN", $cn);
        //Elaboramos la sentencia
        $sql = "UPDATE Lector SET EstadoLector='$Lector->getEstado()' WHERE
codigoProducto=$Lector->getNumLector()";
        //Ejecutamos la sentencia
        $result = mysql_query($sql, $cn);
        $rpta;
        if (!$result) {
            //Si no obtiene resultados anulamos la transaccion
            mysql_query("ROLLBACK", $cn);
            $rpta = false;
        } else {
            //Si obtiene resultados confirmamos la transaccion
            mysql_query("COMMIT", $cn);
            $rpta = true;
        }
        //Cerramos la conexion
        mysql_close($cn);
    } catch (exception $e) {
        try {
            mysql_query("ROLLBACK", $cn);
        } catch (exception $e1) {
        }
        try {
            mysql_close($cn);
        } catch (exception $e1) {
        }
        $rpta = false;
    }
    return $rpta;
}
function RegistrarNuevoPrestamo()
{
    $fechaprestamo=$_POST["Afechaprestamo"];
    $fechadevolucion=$_POST["Afechadevolucion"];
    $codmatbiblio=$_POST["Acodmatbiblio"];
    $numejemplar=$_POST["Anumejemplar"];
    $numlector=$_POST["numlector"];
    $numpedidoprestamo=$_POST["numpedidoprestamo"];
}
```

```
$rpta=false;
try {
    //Creamos un objeto de la clase conexion
    $miconexion = new Conexion();
    //Obtenemos la conexion
    $cn = $miconexion->conectar();
    //Comenzamos la transaccion
    mysql_query("BEGIN", $cn);
    //Elaboramos la sentencia
    $sql = "INSERT INTO prestamo
(EstPre,FecPre,FecDev,CodMatBiblio,numEjemplar,numLector,CodBiblio)
VALUES('1','$fechaprestamo','$fechadevolucion','$codmatbiblio','$numejemplar','$numle
ctor','$_USER')";
    //Ejecutamos la sentencia
    $result = mysql_query($sql, $cn);
    if (!$result) {
        //Si no obtiene resultados anulamos la transaccion
        mysql_query("ROLLBACK", $cn);
        $rpta = false;
    } else {
        //Si obtiene resultados confirmamos la transaccion
        mysql_query("COMMIT", $cn);
        $rpta = true;
    }
    //Cerramos la conexion
    mysql_close($cn);
} catch (exception $e) {
    try {
        mysql_query("ROLLBACK", $cn);
    } catch (exception $e1) {
    }
    try {
        mysql_close($cn);
    } catch (exception $e1) {
    }
    $rpta = false;
}
return $rpta;
}

function ActualizarCondicionEjemplar($numejemplar)
{
    $Ejemplar=new Ejemplar();
    $Ejemplar->setNumLector($numejemplar);
    //2 estado Prestado
    $Ejemplar->setCondicion('2');
    $rpta=false;
    try {
        //Creamos un objeto de la clase conexion
        $miconexion = new Conexion();
        //Obtenemos la conexion
        $cn = $miconexion->conectar();
        //Comenzamos la transaccion
        mysql_query("BEGIN", $cn);
        //Elaboramos la sentencia
        $sql = "UPDATE Ejemplar SET Condicion='$Ejemplar->Condicion()' WHERE
NumEjemplar=$Ejemplar->getNumEjemplar()";
        //Ejecutamos la sentencia
        $result = mysql_query($sql, $cn);
    }
}
```

```
$rpta;
if (!$result) {
    //Si no obtiene resultados anulamos la transaccion
    mysql_query("ROLLBACK", $cn);
    $rpta = false;
} else {
    //Si obtiene resultados confirmamos la transaccion
    mysql_query("COMMIT", $cn);
    $rpta = true;
}
//Cerramos la conexion
mysql_close($cn);
} catch (exception $e) {
    try {
        mysql_query("ROLLBACK", $cn);
    } catch (exception $e1) {

    }
    try {
        mysql_close($cn);
    } catch (exception $e1) {

    }
    $rpta = false;
}
return $rpta;
}
function ActualizarEstadoPedido($numpedidoprestamo)
{
    $PedidoPrestamo=new PedidoPrestamo();
    $PedidoPrestamo->setNumPedPres($numpedidoprestamo);
    //2 estado Atendido
    $PedidoPrestamo->setEstadoPedido('2');
    $rpta=false;
    try {
        //Creamos un objeto de la clase conexion
        $miconexion = new Conexion();
        //Obtenemos la conexion
        $cn = $miconexion->conectar();
        //Comenzamos la transaccion
        mysql_query("BEGIN", $cn);
        //Elaboramos la sentencia
        $sql = "UPDATE PedidoPrestamo SET EstadoPedido='$PedidoPrestamo-
>getEstadoPedido()' WHERE NumPedPres=$PedidoPrestamo->getNumPedPres()";
        //Ejecutamos la sentencia
        $result = mysql_query($sql, $cn);
        $rpta;
        if (!$result) {
            //Si no obtiene resultados anulamos la transaccion
            mysql_query("ROLLBACK", $cn);
            $rpta = false;
        } else {
            //Si obtiene resultados confirmamos la transaccion
            mysql_query("COMMIT", $cn);
            $rpta = true;
        }
        //Cerramos la conexion
        mysql_close($cn);
    } catch (exception $e) {
        try {
            mysql_query("ROLLBACK", $cn);
```

```
    } catch (exception $e1) {  
    }  
    try {  
        mysql_close($cn);  
    } catch (exception $e1) {  
    }  
    $rpta = false;  
}   
return $rpta;  
}
```

?>

2.3. Clase Interface

Archivo: frmBusquedaPedidoPrestamo.php

```
<?php  
header('Cache-Control: no-cache');  
header('Pragma: no-cache');  
session_start();  
if(!isset($_SESSION["JefeBiblio"]) and !isset($_SESSION["Biblio"]))  
    header("location: ../principal.php");  
include ("../conectar.php");  
  
$titulo=$_POST["titulo"];  
$autor=$_POST["autor"];  
$lector=$_POST["lector"];  
  
$cadena_busqueda=$_POST["cadena_busqueda"];  
  
function RealizarBusqueda($titulo,$autor,$lector)  
{ include("Control/GestorBusquedaPedidoPrestamo.php");  
  $GestorBusquedaPedidoPrestamo= new $GestorBusquedaPedidoPrestamo();  
  $GestorBusquedaPedidoPrestamo->ConsultaPedidoPendientes($titulo,$autor,$lector);  
}  
  
function MostrarRegistrosPedidosPendientes($titulo,$autor,$lector)  
{  
    $res_resultado= RealizarBusqueda($titulo,$autor,$lector);  
    $contador=0;  
    while ($contador <  
mysql_num_rows($res_resultado)) {  
        if ($contador % 2) {  
$fondolinea="itemParTabla"; } else { $fondolinea="itemImparTabla"; }?>  
        <tr class="<?php echo $fondolinea?>">  
            <td class="aCentro" width="4%"><? echo  
$contador+1;?></td>  
            <?  
            $numpedpre =  
mysql_result($res_resultado, $contador, "pedidoprestamo.numpedpre");  
            ?>  
            <input type="hidden" id="numpedpre"  
name="numpedpre" value="<?echo $numpedpre?>"/>  
            <?  
            $codmat = mysql_result($res_resultado,  
$contador, "codmatbiblio");  
            ?>  
    }  
}
```

```

                                <td width="10%"><div align="left"><?
echo $codmat?></div></td>
                                <?
                                $numejemplar =
mysql_result($res_resultado,$contador,"numejemplar");
                                ?>
                                <td width="10%"><div align="left"><?
echo $numejemplar?></div></td>
                                <?
                                include("Entidad/Lector.php");
                                $Lector= new $Lector;
                                $lector = $Lector->getApePat()."
                                ".$Lector->getApeMat()." ".$Lector->getNomPer();
                                ?>
                                <td width="25%"><div align="left"><?
echo $lector?> </div></td>
                                <?
                                ?>
                                <td width="10%"><div align="left"><?
echo mysql_result($res_resultado,$contador,"fecpedpre")?> </div></td>
                                <?
                                $estado =
mysql_result($res_resultado,$contador,"EstPedPre");
                                if($estado=='1'){
                                    $nombreestado = "Pendiente";
                                    $color = "#DFDF00";
                                }
                                ?>
                                <td width="5%"><div align="center"
style="background-color:<?echo $color?>"><?echo $nombreestado?></div></td>
                                <td
                                width="5%"><div
                                align="left">&nbsp;</div></td>
                                <td width="5%"><div align="center"><a
href="#">','<?echo $numejemplar?>','<?echo
$numlector?>','<?echo $numpedpre?>')" title="Atender Pedido" /></a></div></td>
                                <td width="5%"><div align="center"><a
href="#">')" title="Anular Pedido"
/></a></div></td>
                                <? $contador++;
                                }
                                function AtenderPedido($codmat,$numejemplar,$numlector,$numpedpre)
                                {
                                    parent.location.href="nuevo_prestamo.php?codMatBiblio=" + codmat +
                                    "&numEjemplar=" + numejemplar + "&numLector=" + numlector + "&numpedpre=" + numpedpre
                                    + "&cadena_busqueda=<? echo $cadena_busqueda?>";
                                }
                                ?>
                                <html>
                                <head>
                                <title>Libros</title>
                                <link href="../estilos/estilos.css" type="text/css" rel="stylesheet" />
                                <script language="javascript">
                                function inicio() {
                                    var numfilas=document.getElementById("numfilas").value;

```



```
var indi=parent.document.getElementById("iniciopagina").value;
var contador=1;
var indice=0;
if (indi>numfilas) {
    indi=1;
}
parent.document.form_busqueda.filas.value=numfilas;
parent.document.form_busqueda.paginas.innerHTML="";
while (contador<=numfilas) {
    texto=contador + "-" + parseInt(contador+9);
    if (indi==contador) {

parent.document.form_busqueda.paginas.options[indice]=new          Option
(texto,contador);

parent.document.form_busqueda.paginas.options[indice].selected=true;
    } else {

parent.document.form_busqueda.paginas.options[indice]=new          Option
(texto,contador);

    }
    indice++;
    contador=contador+10;
}
}

function eliminar_prestamo(codpre){
    parent.location.href="anular_prestamo.php?codpret=" + codpre;
}
</script>
</head>

<body onload=inicio()>
    <div id="pagina">
        <div id="zonaContenido">
            <div align="center">
                <table class="fuente8" width="87%" cellpadding=3
border=0 ID="Table1">
                    <input type="hidden" name="numfilas" id="numfilas" value="<? echo
$filas ?>">
                        <? $iniciopagina=$_POST["iniciopagina"];
if (empty($iniciopagina)) {
$iniciopagina=$_GET["iniciopagina"]; } else { $iniciopagina=$iniciopagina-1;}
if (empty($iniciopagina)) { $iniciopagina=0; }
if ($iniciopagina>$filas) { $iniciopagina=0; }
if ($filas > 0) {
MostrarRegistrosPedidosPendientes();
}
                        </table>
                    <? } else { ?>
                        <table class="fuente8" width="87%" cellpadding=3
border=0>
                            <tr>
                                <td width="100%" class="mensaje"><?php
echo "No hay ning&uacute;n prestamo pendiente que cumpla con los criterios de
b&uacute;squeda";?></td>
                            </tr>
                        </table>
                    </div>
                </div>
            </div>
        </div>
    </body>
```

```

                                <? } ?>
                            </div>
                        </div>
                    </div>
                </div>
            </body>
        </html>
Archivo: frmPrestamo.php
<?php
header('Cache-Control: no-cache');
header('Pragma: no-cache');
session_start();
if(!isset($_SESSION["JefeBiblio"]) and !isset($_SESSION["Biblio"]))
    header("location: ../principal.php");
include ("../conectar.php");
$codmatbiblio=$_GET["codMatBiblio"];
$numejemplar=$_GET["numEjemplar"];
$numlector=$_GET["numLector"];
$numpedidoprestamo=$_GET["numpedpre"];
$cadena_busqueda=$_GET["cadena_busqueda"];

function MostrarRegistroPrestamo()
{ $GestorMateria= new GestorMateria();
  $Lector = new Lector();

    ?>
    <table class="fuente8" width="98%" cellspacing=0 cellpadding=3 border=0/>
        <tr>
            <td width="15%">Fecha de prestamo</td>
            <td width="30%"><input name="Afechaprestamo"
id="fechaprestamo" value="<?echo GenerarFechaPrestamo();" maxlength="9"
class="cajaPequena" type="text" readonly="yes" /></td>
        </tr>
        <tr>
            <td>Fecha de devoluci&oacute;n</td>
            <td><input name="Afechadevolucion"
type="text" class="cajaPequena" id="fechadevolucion" size="10" maxlength="10"
readonly="true" /> 
                <script type="text/javascript">
                    Calendar.setup(
                        {
                            inputField      :
"fechadevolucion",
                            ifFormat      :
"%Y/%m/%d",
                            button       :
"Image1"
                        }
                    );
                </script></td>
        </tr>
        <tr>
            <td width="15%">C&oacute;digo material
bibliogr&aacute;fico</td>
            <td width="30%"><input name="Acodmatbiblio"
id="codmatbiblio" value="<?echo $codmatbiblio?" maxlength="9" class="cajaPequena"
type="text" readonly="yes" /></td>
            <td width="5%">&nbsp;</td>
            <td width="5%">&nbsp;</td>
        </tr>
    </table>
}
?>

```

```

        <td width="6%" align="right"></td>
        <td width="55%" rowspan="15" align="left"
valign="top"><ul id="lista-errores"></ul></td>
        </tr>
        <tr>
        <td width="17%">N&uacute;mero de
ejemplar</td>
        <td><input
        id="numejemplar"
name="Anumejemplar" value="<?echo $numejemplar?>" maxlength="5" class="cajaPequena"
type="text" readonly="yes" /></td>
        </tr>
        <?
        $query = $GestorMateria->Listar();
        ?>
        <tr>
        <td width="17%">Materia</td>
        <td><input id="materia" name="Amateria"
value="<?echo mysql_result($query,0,"nommateria")?>" maxlength="5" class="cajaMedia"
type="text" readonly="yes" /></td>
        </tr>
        <tr>
        <td width="17%">T&iacute;tulo</td>
        <td><input id="titulo" name="Atitulo"
value="<?echo mysql_result($query,0,"titulo")?>" maxlength="5" class="cajaGrande"
type="text" readonly="yes" /></td>
        </tr>
        <tr>
        <td width="17%">Autor</td>
        <td><input id="autor" name="Aautor"
value="<?echo mysql_result($query,0,"nomautor")?>" maxlength="5" class="cajaGrande"
type="text" readonly="yes" /></td>
        </tr>
        <?
        $query = $Lector->setNumLector($numlector);
        ?>
        <tr>
        <td width="17%">Apellido paterno</td>
        <td><input id="al" name="Apl"
value="<?echo mysql_result($query,0,"ApePatLec")?>" maxlength="5" class="cajaGrande"
type="text" readonly="yes" /></td>
        </tr>
        <tr>
        <td width="17%">Apellido materno</td>
        <td><input id="ml" name="Aml"
value="<?echo mysql_result($query,0,"ApeMatLect")?>" maxlength="5" class="cajaGrande"
type="text" readonly="yes" /></td>
        </tr>
        <tr>
        <td width="17%">Nombre</td>
        <td><input id="nl" name="Anl"
value="<?echo mysql_result($query,0,"NomLect")?>" maxlength="5" class="cajaGrande"
type="text" readonly="yes" /></td>
        </tr>
    </table>
<? }

function GenerarFechaPrestamo()
{
    return date("Y")."." / ".date("m")."." / ".date(d);

```

```
}
function ImprimirFichaPrestamo($numPrestamo,$nlector,$codmatb)
{
    $Prestamo= new Prestamo();
    $Prestamo->setNumPrestamo($numPrestamo);
    $Lector = new Lector();
    $Lector->setNumLector($nlector);
    $MaterialBibliografico= new MaterialBibliografico();
    $MaterialBibliografico->setCodMatBiblio($codMatBiblio);

    location.href=" ../fpdf/imprimir_ticket_html.php?numPrestamo=$Prestamo-
>getNumPrestamo()+"&FecPre=" + $Prestamo->getFecPre() + "&FecDev=" + $Prestamo-
>getFecDev + "&numlector=" + Lector->getNumLector() + "&CodMatBiblio=" +
$MaterialBibliografico->getCodMatBiblio();
}
}

?>
<html>
    <head>
        <title>Principal</title>
        <link href=" ../estilos/estilos.css" type="text/css" rel="stylesheet"/>
        <link href=" ../calendario/calendar-blue.css" rel="stylesheet"
type="text/css" />
        <script type="text/JavaScript" language="javascript"
src=" ../calendario/calendar.js"></script>
        <script type="text/JavaScript" language="javascript"
src=" ../calendario/lang/calendar-sp.js"></script>
        <script type="text/JavaScript" language="javascript"
src=" ../calendario/calendar-setup.js"></script>
        <script type="text/javascript" src=" ../funciones/validar.js"></script>
        <script language="javascript">

            function cancelar() {
                location.href="index.php";
            }

            var cursor;
            if (document.all) {
                // EstÃ¡ utilizando EXPLORER
                cursor='hand';
            } else {
                // EstÃ¡ utilizando MOZILLA/NETSCAPE
                cursor='pointer';
            }

            function limpiar() {
                document.getElementById("codigo").value="";
                document.getElementById("nombre").value="";
                document.formulario.cboPais.options[0].selected = true;
            }
        </script>
    </head>
    <body>
        <div id="pagina">
            <div id="zonaContenido">
                <div align="center">
                    <div id="tituloForm" class="header">INSERTAR PRESTAMO
                </div>
            </div>
        </div>
    </body>
</html>
```

```
        <div id="frmBusqueda">
            <form id="formulario" name="formulario" method="post"
action="AdministrarPrestamo.php" enctype="multipart/form-data">
                <input id="accion" name="accion" value="alta"
type="hidden"/>
                    <?
                        MostrarRegistroPrestamo();
                    ?>
                <input type="hidden" name="numlector"
id="Anumlector" value="<?echo $numlector?>"/>
                <input type="hidden" name="numpedidoprestamo"
id="Anumpedidoprestamo" value="<?echo $numpedidoprestamo?>"/>
            </div>
                <div id="botonBusqueda">
                    
                    
                    <input type="hidden" name="id" id="id" value="">
                </div>
            </form>
        </div>
    </div>
</body>
</html>
```



MANUAL DE USUARIO
DEL SISTEMA DE BIBLIOTECA
DE LA MUNICIPALIDAD PROVINCIAL DE CAJAMARCA



**Municipalidad
Provincial de
Cajamarca**

Versión 1.0

Cajamarca, Abril del 2013



Tabla de Contenidos

| | |
|--|-----------|
| 1. INTRODUCCIÓN | 3 |
| 2. INSTALACIÓN Y CONFIGURACIÓN INICIAL DEL SISTEMA..... | 4 |
| 3. GESTIÓN DE INSCRIPCIÓN DE LECTOR..... | 6 |
| 4. GESTIÓN DE CONSULTA DE MATERIAL BIBLIOGRÁFICO Y PEDIDO DE PRÉSTAMO POR PARTE DEL LECTOR..... | 10 |
| 5. GESTIÓN PRÉSTAMOS Y DEVOLUCIONES. | 12 |
| 6. GESTIÓN DE ADMINISTRACIÓN MATERIAL BIBLIOGRÁFICO..... | 15 |
| 7. GESTIÓN DE COPIAS DE SEGURIDAD | 21 |



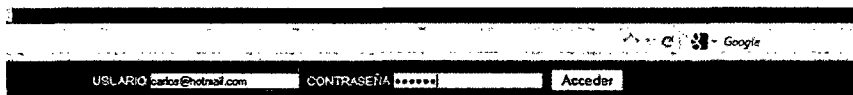
1. Introducción

El sistema es una aplicación para controlar principalmente, la Inscripción de Lector, Consultas de Material Bibliográfico y como módulo principal la gestión de Préstamos y Devoluciones de Material Bibliográfico destinado para ser usado en la Biblioteca de la Municipalidad Provincial de Cajamarca

La aplicación se a desarrollado con herramientas bajo licencia GPL. Está desarrollada sobre entorno Web, lo que la hace ser muy versátil. Es independiente del se sistema operativo y además permite el trabajo en red. Las funciones principales del sistema son:

- Gestión de Inscripción de Lector.
- Gestión de Consulta de Material Bibliográfico y Pedido de Préstamo por parte del Lector.
- Gestión Préstamos y Devoluciones.
- Gestión de Administración Material Bibliográfico.
- Gestión de copias de seguridad.

El funcionamiento a través de entorno Web permite su uso multiplataforma, tanto en sistemas operativos Windows como Linux. El software ha sido desarrollado en lenguaje PHP y utilizando como motor de base de datos MySQL.



**Municipalidad
Provincial de
Cajamarca**

Biblioteca Pública Municipal - JOSE GALVES

Versión 1.0

© 2012

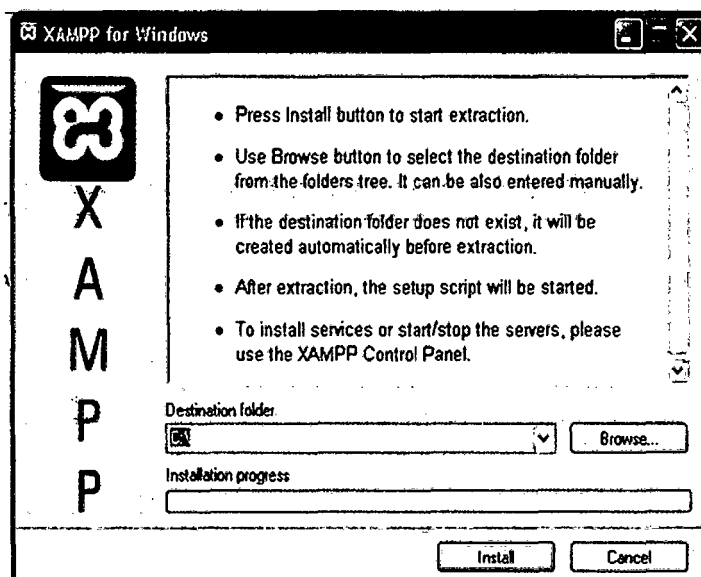
Resolución óptima 1024 x 768 píxeles

Aplicación optimizada para Firefox

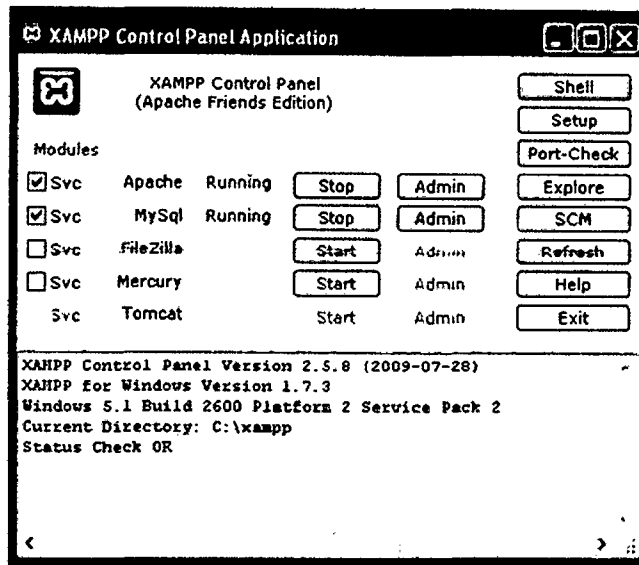


2. Instalación y Configuración inicial del Sistema de Biblioteca

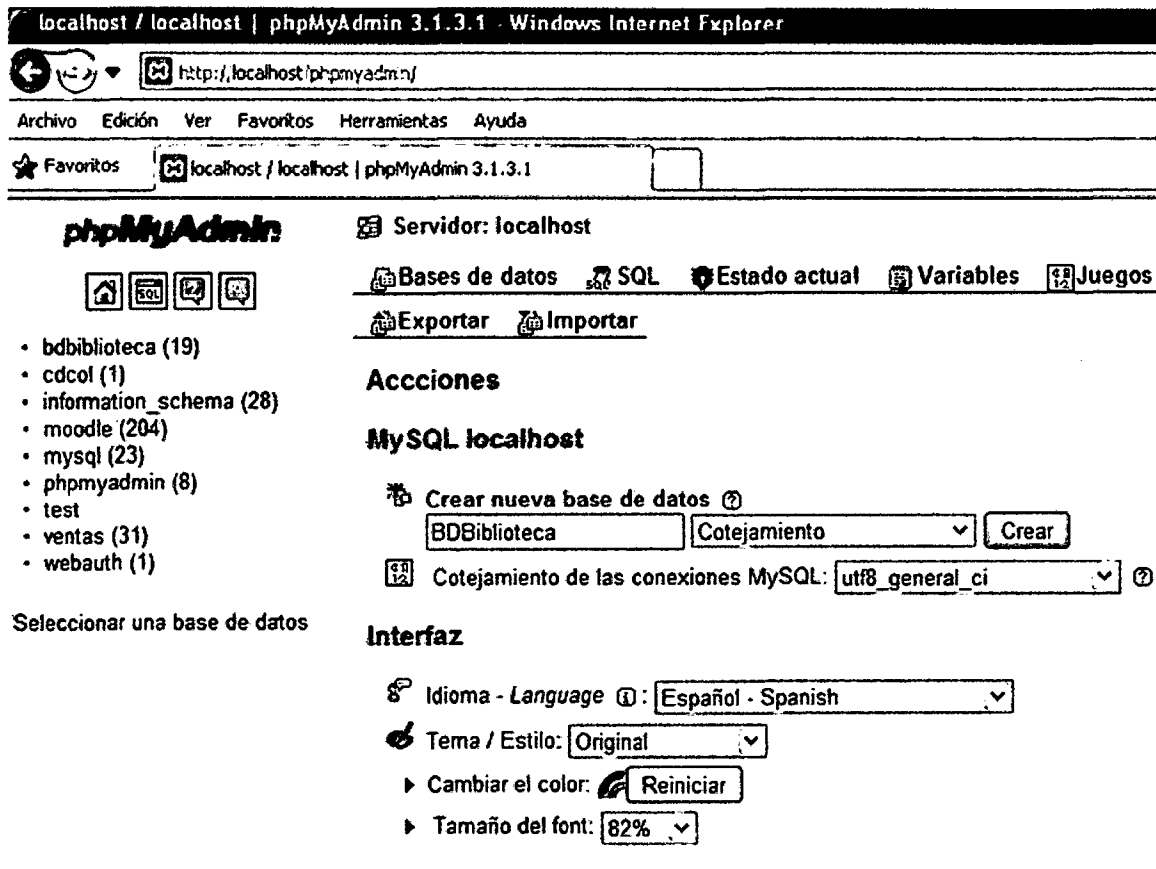
Para poder iniciar a utilizar el sistema requiere de algunas tareas de instalación y configuración básicas, estas tareas son la de Instalar Apache, MySql y Php, puede usarse algún paquete que contenga estos programas, en mi caso utilizado XAMP



Una vez terminada la instalación, el instalador te va a preguntar si deseas abrir el panel de control. Desde esa nueva ventana se va a poder prender y apagar el servidor Apache y MySQL (PHP se inicia cuando inicias Apache) cada vez que lo necesites.



Una vez instalado XAMPP se debe crear la Base de datos en MySQL





El manual de usuarios de Sistema de Biblioteca, asume que tienes instalado correctamente, y dependiendo al logeo del usuario del sistema se va observar los menús; a continuación te indicamos las actividades iniciales para utilizar la correcta utilización del sistema:

Para comenzar a utilizar el Sistema de Biblioteca, se requiere ingresar la siguiente información en la secuencia que especificamos:

1. Ubigeo
2. País
3. Idioma
4. Autor
5. Editorial
6. Materia

El detalle de cómo ingresar esta información la encuentras en este documento en los capítulos correspondientes, por favor referirse a ellos para el detalle.

3. Gestión de Inscripción de Lector.

3.1. Lector

Antes de poder ingresar al módulo a la ventana de Lector, el bibliotecario tendrá que autenticarse como tal.

The image shows a login form with two input fields: 'USUARIO' and 'CONTRASEÑA'. The password field contains several asterisks. To the right of the password field is a button labeled 'Acceder'. Below the login form is a horizontal navigation menu with four items: 'Inicio', 'Mantenimiento', 'Administrar Mat. Biblio.', and 'Administrar Lectores'.

El sistema permite administrar una nómina de lectores, para lo cual debes ingresar al menú "Administrar Lector" del menú Principal cuando se haya accedido como usuario Bibliotecario, que tiene el privilegio de crear nuevo usuario.

Se desplegará la siguiente pantalla con las opciones de buscar, limpiar e ingresar un nuevo registro de lector:



BUSCAR LECTOR

ApeEdo Paterno

ApeEdo Materno

Nombre

Cant. de Lectores Mostrados

RELACION DE LECTORES

| ITEM | CÓDIGO | APELLIDOS | NOMBRE (S) | ESTADO | F. VENCIMIENTO | | |
|------|----------|----------------|----------------|------------|----------------|--------------------------|--------------------------|
| 1 | L0000001 | ROJAS VALENCIA | MARIELA | Activo | 2013-12-05 | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | L0000099 | MENDOZA SANTOS | CARLOS EDUARDO | Activo | 2013-12-05 | <input type="checkbox"/> | <input type="checkbox"/> |
| 3 | L0000100 | QUISPE MALCA | ANDERSON | Sancionado | 2013-12-05 | <input type="checkbox"/> | <input type="checkbox"/> |
| 4 | L0000101 | LEON MENDOZA | ANA | Sancionado | 2013-12-05 | <input type="checkbox"/> | <input type="checkbox"/> |
| 5 | L0000009 | ROJAS VALENCIA | DANTE | Sancionado | 2013-12-05 | <input type="checkbox"/> | <input type="checkbox"/> |

3.1.1. Agregar: Al presionar el Botón "agregar" se desplegará la siguiente pantalla para hacer el ingreso:

INGRESAR LECTOR

Código lector

ApeEdo paterno

ApeEdo materno

Nombre (s)

Dni

Dirección

Ubigeo

Fecha de expedición

Fecha de vencimiento

Grado de instrucción

Fecha de nacimiento

Estado del lector

Genero del lector

Teléfono

Email - Correo

Contraseña

? Diciembre, 2012

| Smn | Lun | Mar | Mie | Jue | Vie | Sab | Dom |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 45 | | | | | | 1 | 2 |
| 46 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 47 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 48 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 49 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 50 | 31 | | | | | | |


Seleccione fecha




- DEBE COMPLETAR EL CAMPO "PATERNO"
- DEBE COMPLETAR EL CAMPO "MATERNO"
- DEBE COMPLETAR EL CAMPO "NOMBRE"
- DEBE COMPLETAR EL CAMPO "DNI"
- DEBE COMPLETAR EL CAMPO "DIRECCION"
- DEBE COMPLETAR EL CAMPO "UBIGEO"
- DEBE COMPLETAR EL CAMPO "FECHAEXPEDICION"
- DEBE COMPLETAR EL CAMPO "FECHAVENCIMIENTO"
- DEBE COMPLETAR EL CAMPO "NACIMIENTO"
- DEBE COMPLETAR EL CAMPO "TELEFONO"
- DEBE COMPLETAR EL CAMPO "EMAIL"
- DEBE COMPLETAR EL CAMPO "PASS"
- DEBE ELIGIR UN VALOR DEL COMBO "CBOGRADO"
- DEBE ELIGIR UN VALOR DEL COMBO "CBOESTADO"
- DEBE ELIGIR UN VALOR DEL COMBO "CBOSEXO"

Cada campo es validado, sino pasa la validación no se podrá agregar.



Al presionar el botón aceptar se despliega la pantalla con la lista de todos los registros ingresados.

3.1.2. Modificar: Al hacer clic en el botón Modificar de tipo imagen  que pertenece a un lector, se presentará la pantalla para poder modificar los cambios.

| MODIFICAR LECTOR | |
|----------------------|--|
| Apellido paterno | ROJAS |
| Apellido Materno | VALENCIA |
| Nombre (s) | MARELA |
| DNI | 74190384 |
| Dirección | JR. ICA 123 |
| Ubigeo | 010000 |
| Fecha de expedición | 2012-12-05  |
| Fecha de vencimiento | 2013-12-05  |
| Grado de instrucción | Universidad |
| Fecha de nacimiento | 2008-12-05  |
| Estado del lector | Activo |
| Genero del lector | Femenino |
| Teléfono | 413241 |
| Email - Correo | MARY@UNC.EDU.PE |
| Contraseña | 123 |

aceptar limpiar cancelar

Finalmente se presiona el botón aceptar, para poder actualizar los datos

| MODIFICAR LECTOR | |
|---|-----------------|
| LOS DATOS DEL LECTOR HAN SIDO MODIFICADOS CORRECTAMENTE | |
| Código lector | L0000001 |
| Apellido paterno | ROJAS |
| Apellido materno | VALENCIA |
| Nombre (s) | MARELA |
| DNI | 74190384 |
| Dirección | JR. ICA 123 |
| Ubigeo | 010000 |
| Fecha de expedición | 2012-12-05 |
| Fecha de vencimiento | 2013-12-05 |
| Grado de instrucción | Universidad |
| Fecha de nacimiento | 2008-12-05 |
| Estado | Activo |
| Genero | Femenino |
| Teléfono | 413241 |
| Email | MARY@UNC.EDU.PE |
| Contraseña | 123 |

aceptar



3.1.3. Buscar: el botón buscar se debe presionar cuando usted desee filtrar los datos del lector, ya sea por apellidos o nombres.

| | | | | | | | |
|---|----------|----------------|----------|------------|------------|--|--|
| 3 | L0000100 | QUISE MALCA | ANDERSON | Sancionado | 2013-12-04 | | |
| 4 | L0000101 | LEON MENDOZA | ANA | Sancionado | 2013-12-05 | | |
| 5 | L0000029 | ROJAS VALENCIA | DAJTE | Baja | 2013-12-05 | | |

En la columna estado se observa 3 estados: si su carnet está vigente, presenta verde, si es amarillo está sancionado y si esta de color rojo está de baja, ósea inactivo.

3.2. Pago de Inscripción

Este módulo sirve para registrar el pago que hace el lector, cuando se inscribe por primera vez o se reinscribe pasado la fecha de vencimiento.

BUSQUEDA DE PAGOS DE INSCRIPCION

Apellido paterno

Apellido materno

Nombres

Nº de inscripciones encontradas Mostrados 1-10 ▾

| DETALLE DE PAGOS DE INSCRIPCION | | | | |
|---------------------------------|----------|------------------|---------------|-------|
| ITEM | NO. PAGO | LECTOR | FECHA DE PAGO | MONTO |
| 1 | 1 | LEON MENDOZA ANA | 2012-12-05 | 3.00 |

3.2.1. Buscar: Este botón se ejecuta cuando usted desea buscar un lector ya sea por apellidos o nombres, el cual hará filtro en la lista de pagos de inscripción.

3.2.2. Limpiar: Blanquea los campos que se ha buscado, para hacer una nueva búsqueda.

3.2.3. N° de inscripciones encontradas: Campo que sirve para informarnos el total de registro que tiene los pagos de inscripción.

3.2.4. Lista Mostrados: Cuando existe demasiados registros esta lista nos podrá limitar el número de registros por página.

Nóta: Los botones, Buscar, Limpiar , caja de texto de N° de Inscripciones encontradas, como la lista de Mostrados, tienen la misma



funcionalidad para todas las interfaces principales, por lo cual ya no se detallará en lo sucesivo.

3.2.5. **Agregar:** Es el principal botón de este módulo para poder registrar el pago de inscripción de un lector

INSERTAR PAGO DE INSCRIPCIÓN

| | | |
|-----------------------|--------------------------------|----------------------------------|
| Número de pago | <input type="text" value="2"/> | |
| Número de comprobante | <input type="text"/> | |
| Fecha de pago | <input type="text"/> | <input type="button" value="📅"/> |
| Monto | <input type="text"/> | |
| Código de lector | <input type="text"/> | <input type="button" value="🔍"/> |

4. Gestión de Consulta de Material Bibliográfico y Pedido de Préstamo por parte del Lector.

Esta ventana de consulta de Material bibliográfico se mostrará siempre y cuando el lector esté inscripto y activo.

4.1. **Buscar:** El lector podrá hacer consulta por título y/o autor y/o Materia, para saber si existe disponibilidad del material de la relación de material bibliográfico.

RELACION DE PRESTAMOS


BUSQUEDA DE MATERIAL BIBLIOGRAFICO

| | | |
|---------|---|--|
| TITULO | <input type="text"/> | |
| AUTOR | <input type="text"/> | |
| MATERIA | <input type="text" value="Seleccione una materia"/> | |


No. Material Bibliograficos encontrados Mostrados

| ITEM | CÓDIGO | TITULO | AUTOR | MATERIA | CANT. EJEMPLARES | |
|------|--------|--------------------------------|---------|------------|------------------|---|
| 1 | MB001 | DERIVADAS | BEINERO | MATEMATICA | 1 | <input type="button" value="🔍"/> |
| 2 | MB002 | LIMITES DERIVADAS E INTEGRALES | BEINERO | MATEMATICA | 0 | <input type="button" value="Ver Ejemplares y Realizar Pedido"/> |




4.2. Realizar Pedido: El botón de tipo imagen  ver Ejemplar y Realizar pedido, nos muestra la ventana con información detallada del material bibliográfico y la relación de ejemplares de dicho material, y con la opción de poder hacer un pedido de préstamo.

| DETALLES MATERIAL BIBLIOGRAFICO | |
|---------------------------------|------------|
| Código | MB001 |
| Tipo Material Bibliográfico | Libro |
| Título | DERIVADAS |
| Año de publicación | 1997 |
| Autor | BENERO |
| Materia | MATEMATICA |
| Idioma | Japones |

| RELACION DE EJEMPLARES | | | | |
|------------------------|--------|--------------|------------|---|
| ITEM | CÓDIGO | NO. EJEMPLAR | ESTADO | REALIZAR PEDIDO |
| 1 | MB001 | 1 | Disponible |  |

REALIZAR PEDIDO DE PRESTAMO DE ESTE EJEMPLAR

4.3. Realizar Pedido: El botón de tipo imagen , que se presenta en cada registro de la relación de Ejemplares de Dicho Material Bibliográfico, servirá para que el lector pueda realizar un pedido de préstamo para que sea atendido por el bibliotecario. Entonces una vez que usted hace click en dicho botón se mostrará la siguiente ventana. Que nos muestra el mensaje de que su pedido ha sido realizado con éxito.

| PEDIDO DE PRESTAMO | |
|--------------------------------------|------------|
| SU PEDIDO A SIDO REALIZADO CON EXITO | |
| Código | MB001 |
| Tipo Material Bibliográfico | Libro |
| Título | DERIVADAS |
| Año de publicación | 1997 |
| Autor | BENERO |
| Materia | MATEMATICA |
| Idioma | Japones |

aceptar

Por ultimo usted debe hacer clic en el botón aceptar para que pueda regresar al módulo de consulta.



5. Gestión Préstamos y Devoluciones.

La Gestión de Préstamos de Material Bibliográfico así como Devoluciones se presentará cuando el usuario acceda como tipo de Bibliotecario.

A screenshot of a login form. It features two input fields: one labeled 'USUARIO' and another labeled 'CONTRASEÑA' with a masked password '*****'. To the right of the password field is a button labeled 'Acceder'.

Luego nos mostrará un menú para poder Administrar los Préstamos y Devoluciones, así como también Sanciones y Pago por Multa.

A screenshot of a navigation menu. The top bar contains 'Préstamos y Devoluciones' and 'Cerrar Sesión'. Below this, a dropdown menu is open, listing 'Préstamos', 'Devoluciones', 'Sanciones', and 'Pago por Multa'.

5.1. Préstamos.

Una vez que el lector ha hecho su consulta y ha encontrado el material que necesita para llevar a casa o a sala, realiza un pedido de préstamo. El bibliotecario le presentará una relación de los préstamos pendientes (campo estado de color amarillo), para poder hacer el despacho del material bibliográfico; en esta relación se presentará en forma ascendente por el campo de fecha de pedido, con la lógica de que "el primero en llegar a la cola debe ser el primero en ser atenderse"



RESERVA DE PRÉSTAMO

LECTOR

TÍTULO

AUTOR

No. Material Bibliograficos encontrados Mostrados

| ITEM | CÓDIGO | NUM. EJEMPLAR | LECTOR | FECHA PEDIDO | ESTADO |
|------|--------|---------------|-----------------------|--------------|--------|
| 1 | MB001 | 1 | ROJAS VALENCIA MARELA | 2012-12-06 | |

5.1.1 Atender Pedido de Préstamo: El bibliotecario procede a hacer clic en el botón de tipo imagen , el cual procede a ver una ventana de insertar préstamo, la mayoría de los campos ya están llenados, lo cual el usuario solo debe llenar solamente la fecha de devolución.

INSERTAR PRÉSTAMO

Fecha de prestamo

Fecha de devolución

Código material bibliográfico Noviembre, 2012

Número de ejemplar

Materia

Título

Autor

Apeñdo paterno

Apeñdo materno

Nombre

5.1.2. Aceptar: Una vez llenado el campo fecha devolución, procederá hacer clic en el botón aceptar para luego ver en otra ventana todos los datos grabados y nos muestra el mensaje de que el préstamo ha sido grabado correctamente., como último paso de este módulo hacer click en el botón aceptar para poder regresar al módulo relación de pedido de préstamos pendientes, para poder seguir atendiendo.



| GUARDAR PRESTAMO | |
|-------------------------------|------------|
| Fecha de prestamo | 2012/12/06 |
| Fecha de evolucion | 2012/11/06 |
| Código material bibliográfico | MB001 |
| Número de ejemplar | 1 |
| Materia | MATEMATICA |
| Título | DERIVADAS |
| Autor | BENERO |
| Apellido paterno | ROJAS |
| Apellido materno | VALENCIA |
| Nombres | MARELA |

EL PRESTAMO HA SIDO GRABADO CORRECTAMENTE

aceptar

5.2. **Devoluciones:** En esta ventana se presenta la relación de todos los préstamos generados que aún no devuelven el material bibliográfico, para esto se filtra los registro de préstamos con el estado igual a “prestado”, como se muestra en la figura

| EN BUSCA DE DEVOLUCION | |
|------------------------|----------------------|
| TITULO | <input type="text"/> |
| AUTOR | <input type="text"/> |

No. Material Bibliograficos encontrados Mostrados 1-10

| RELACION DE DEVOLUCIONES PENDIENTES | | | | | |
|-------------------------------------|--------|--------------|-----------------------|-------------|----------------------|
| ITEM | CODIGO | NO. EJEMPLAR | LECTOR | F. PRESTAMO | F. DEVOLUCION ESTADO |
| 1 | MB001 | 1 | ROJAS VALENCIA MARELA | 2012-12-06 | Prestado |

5.2.1. **Atender devolución de préstamo:** El botón atender devolución de préstamo , estará activo siempre y cuando la fecha actual no supere la fecha establecida de la devolución, si esto sucede se mostrará la fecha de devolución de color rojo. Pero si la fecha está dentro de lo establecido, se podrá hacer click y nos mostrará la ventana de realizar devolución.



| REALIZAR DEVOLUCION | |
|-------------------------------|----------------------|
| Fecha de prestamo | 2012-12-06 |
| Fecha de devolución | 2012-11-06 |
| Fecha de entrega | 2012-12-06 |
| Estado libro de devolución | Seleccione un estado |
| Código material bibliográfico | MB001 |
| Número de ejemplar | 1 |
| Materia | MATEMATICA |
| Título | DERIVADAS |
| Autor | BENERO |
| Apellido paterno | ROJAS |
| Apellido materno | VALENCIA |
| Nombre | MARELA |

La mayoría de campos ya están llenados por datos, el bibliotecario solo seleccionará una opción del estado de libro de que está devolviendo el material Bibliográfico.

Una vez que realiza click en el botón aceptar, nos mostrará el mensaje de "La Devolución ha sido Grabado correctamente"

| GUARDAR DEVOLUCION | |
|-------------------------------|------------|
| Fecha de prestamo | 2012-12-06 |
| Fecha de devolución | 2012-11-06 |
| Fecha de entrega | 2012-12-06 |
| Código material bibliográfico | MB001 |
| Número de ejemplar | 1 |
| Materia | MATEMATICA |
| Título | DERIVADAS |
| Autor | BENERO |
| Apellido paterno | ROJAS |
| Apellido materno | VALENCIA |
| Nombres | MARELA |

LA DEVOLUCION HA SIDO GRABADO CORRECTAMENTE

Por último hacer click en el botón aceptar, para retornar al módulo de devolución.

6. Gestión de Administración Material Bibliográfico.

La gestión de Administración de Material Bibliográfico involucra el trabajo del usuario del Jefe de Biblioteca, el cual al acceder al sistema podrá observar un menú que podrá observar los demás módulos para cuestión de supervisión.



- Inicio
- Mantenimiento
- Administrar Mat. Biblio.
- Administrar Lectores
- Bibliotecarios
- Préstamos y Devoluciones
- Cerrar Sesión



6.1. Material Bibliográfico

Siendo su principal tarea de realizar las entradas y salidas del material bibliográfico, para ello el jefe de biblioteca hacer click en el menú "Administrar Mat. Biblio". y nos mostrará la ventana.

BUSCAR MATERIAL BIBLIOGRAFICO

Autor:

Título:

Materia:

- buscar
- limpiar
- agregar

Cant. de Material Bibliográfico

Mostrados

| RELACION DE MATERIAL BIBLIOGRAFICO | | | |
|------------------------------------|--------------------------------------|--------|---|
| ITEM | CÓDIGO TÍTULO | AUTOR | MATERIA ESTADO |
| 1 | MB001 DERIVADAS | BENERO | MATEMATICA Activo |
| 2 | MB002 LÍMITES DERIVADAS E INTEGRALES | BENERO | MATEMATICA Activo |

El campo estado tiene dos colores, verde cuando el material bibliográfico está activo y lo contrario en estado inactivo.

6.1.1. Agregar: El jefe de biblioteca podrá hacer clic en este botón cuando desea agregar un nuevo material Bibliográfico y procederá a llenar los campos.

INSERTAR MATERIAL BIBLIOGRAFICO

Código material bibliográfico:

Tipo material bibliográfico:

Título material bibliográfico:

Año de publicación:

Estado del material

Autor

Materia

Idioma

Resumen

Nombre del archivo

? Diciembre, 2012 x

| Sem | Lun | Mar | Mie | Jue | Wie |
|-----|-----|-----|-----|-----|----------|
| 48 | | | | | 1 2 |
| 49 | 3 | 4 | 5 | 6 | 7 8 9 |
| 50 | 10 | 11 | 12 | 13 | 14 15 16 |
| 51 | 17 | 18 | 19 | 20 | 21 22 23 |
| 52 | 24 | 25 | 26 | 27 | 28 29 30 |
| | | 31 | | | |

Seleccione fecha

- aceptar
- limpiar
- cancelar

Hernán Vargas Cueva


Página 16




El principal campo a tener en cuenta es el código material bibliográfico, que el bibliotecario está entrenado a catalogar.

Una vez llenado los campos y seleccionado las opciones como tipo de material entre otros campos, se procede hacer click en el botón aceptar. El cual nos mostrar otra ventana donde se observa los campos llenados y el mensaje de que el registro de material bibliográfico ha sido grabado satisfactoriamente.

Si el jefe de biblioteca no llena todos los campos principales como código, título del material entre otros no podrá ser registrado, y nos sale los mensajes de advertencia.


6.1.2. Modificar: El botón de imagen modificar  será utilizado cuando usted desea actualizar algún campo, principalmente el estado del material bibliográfico.

6.1.3. Visualizar: Este botón de imagen  , nos lleva a una ventana en donde solo se observa los campos en estado de lectora,



| VER MATERIAL BIBLIOGRAFICO | |
|-----------------------------|------------|
| Código | MB001 |
| Tipo Material Bibliográfico | Libro |
| Título | DERIVADAS |
| Año de publicación | 1997 |
| Estado de material | Activo |
| Autor | BENERO |
| Materia | MATEMATICA |
| Idioma | Japones |
| Resumen | |
| Archivo | |

aceptar

6.1.4. **Eliminar:** Este botón de imagen , nos da la posibilidad de poder eliminar un registro cuando el material bibliográfico nunca haya sido prestado.

| ELIMINAR MATERIAL BIBLIOGRAFICO | |
|---------------------------------|--------------------------------|
| Código | MB002 |
| Tipo Material Bibliográfico | Revista |
| Título | LIMITES DERIVADAS E INTEGRALES |
| Año de publicación | 2009 |
| Estado de material | Activo |
| Autor | BENERO |
| Materia | MATEMATICA |
| Idioma | Español |
| Resumen | |
| Archivo | |

aceptar cancelar

Pero antes nos advierte si deseamos eliminar o no. Si usted hace click en el botón aceptar, se eliminará definitivamente.

6.2. Libro:

Módulo que permite clasificar al material bibliográfico como libro, y permite ingresar datos inherentes del libro.

| BUSCAR LIBRO | | | | |
|-------------------------|--------------------------------|---|--------|-----------|
| Título | <input type="text"/> | | | |
| Autor | <input type="text"/> | | | |
| Editorial | <input type="text"/> | | | |
| | | <input type="button" value="buscar"/> <input type="button" value="limpiar"/> <input type="button" value="agregar"/> | | |
| N de Libros encontrados | <input type="text" value="1"/> | Mostrados <input type="text" value="1-10"/> | | |
| RELACION DE LIBROS | | | | |
| ITEM | CÓDIGO | TÍTULO | AUTOR | EDITORIAL |
| 1 | MB001 | DERIVADAS | BENERO | CALLAO |

Este módulo permite agregar, modificar y buscar un libro determinado.



INSERIR LIBRO

| | | |
|-------------------------------|--------------------------|--|
| Código material bibliográfico | <input type="text"/> | |
| ISBN | <input type="text"/> | |
| Editorial | Seleccione una editorial | |

aceptar limpiar cancelar

6.3. Revista:

Módulo que permite clasificar al material bibliográfico como revista, y permite ingresar datos inherentes de una revista.

BUSCAR REVISTA

| | |
|--------|----------------------|
| Título | <input type="text"/> |
| Autor | <input type="text"/> |

buscar limpiar agregar

fi de revistas encontrados Mostrados

RELACION DE REVISTAS

| ITEM | CODIGO | TITULO | AUTOR | |
|------|--------|-------------------------------|--------|--|
| 1 | MB002 | LMITES DERIVADAS E INTEGRALES | BENERO | |

Este módulo permite agregar, modificar y buscar una revista.

INSERIR REVISTA

| | | |
|-------------------------------|--------------------------------------|--|
| Código material bibliográfico | <input type="text" value="MB002"/> | |
| Número de publicación | <input type="text" value="12"/> | |
| Periodo | <input type="text" value="SEMENAL"/> | |

aceptar limpiar cancelar

6.4. Tesis:

Módulo que permite clasificar al material bibliográfico como tesis, y permite ingresar datos inherentes de una revista.



BUSCAR TESIS

Título

Autor

N de Tesis encontrados Mostrados

RELACION DE TESIS

| ITEM | CODIGO | TITULO | AUTOR |
|---|--------|--------|-------|
| NO HAY NINGUNA TESIS QUE CUMPLA CON LOS CRITERIOS DE BÚSQUEDA | | | |

6.5. Autor

Módulo que permite registrar los autores de los materiales bibliográficos. y permite realizar un ingreso, búsqueda, editar un determinado autor.

BUSCAR AUTOR

Nombre

País

Nro. de autores encontrados Mostrados

RELACION DE AUTORES

| ITEM | CODIGO | AUTOR | PAIS |
|------|--------|--------|------|
| 1 | A0001 | BENERO | Peru |

6.6. Materia

Módulo que permite registrar las materias que clasifica a los materiales bibliográficos. y permite realizar un ingreso, búsqueda, editar una determinada materia.



BUSCAR MATERIA

Nombre

Nro. de Materias Encontradas Mostrar

RELACION DE MATERIAS

| ITEM | CÓDIGO | MATERIA | | | |
|------|--------|------------|--|--|--|
| 1 | M0001 | MATEMATICA | | | |

6.7. Editorial

Módulo que permite registrar las editoriales que pertenece un determinado libro, y permite realizar un ingreso, búsqueda, editar una determinada materia.

BUSCAR EDITORIAL

Nombre

País

Nro. de editoriales encontradas Mostrados

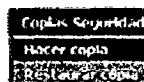
RELACION DE EDITORIALES

| ITEM | CÓDIGO | EDITORIAL | PAÍS | | | |
|------|--------|----------------------|--------|--|--|--|
| 1 | E0001 | EDITORIAL LRIUSA S.A | Mexico | | | |
| 2 | E0002 | CALLAO | Peru | | | |

7. Gestión de copias de seguridad.

7.1 Hacer Copia

Esta opción permite hacer una copia de los datos almacenados en la Base de Datos, para lo cual debe seleccionar la opción "Hacer Copia" del menú "Copias de Seguridad"



Luego ingrese en el campo denominación el nombre y la fecha que dará al respaldo y presione aceptar.



NUEVA COPIA DE SEGURIDAD

Denominacion

Fecha

Hora

Finalmente cuando termina esta copia de seguridad, crea un archivo SQL en el directorio "copias".

7.2. Restaurar Copia

Para restaurar una copia de seguridad debe ingresar a la opción "restaurar copia" del menú "Copias Seguridad".

Se desplegará la siguiente pantalla con las opciones de buscar o imprimir nuevo registro:

BUSCAR COPIAS DE SEGURIDAD

Fecha de inicio

Fecha de fin

Denominacion de la copia

N de copias de seguridad encontradas Mostradas 1-10 v

RELACION DE COPIAS DE SEGURIDAD

| ITEM | DENOMINACION | FECHA | HORA |
|------|--------------|-------|------|
|------|--------------|-------|------|