

RAE

PROGRAMA	TECNOLOGÍA EN SISTEMAS
AUTOR	MARIAYARA PÉREZ BARRETO
TITULO	APLICATIVO TECNOLÓGICO PARA EL CONTROL DE INVENTARIOS, CIRCULACIÓN Y PRÉSTAMO DE LIBROS DE LA BIBLIOTECA DEL COLEGIO EDUARDO CARRANZA
PALABRAS CLAVES	BASE DE DATOS, CICLO DE VIDA DE SOFTWARE, MICROSOFT SQL SERVER: Microsoft SQL Server, MICROSOFT VISUAL STUDIO. NET
DESCRIPCION	Este proyecto pretende organizar la información para reducir el tiempo de consulta y búsqueda de libros, mejora del control de inventario además de generar los informes requeridos por la Secretaria Distrital de Educación con respecto a los libros del Plan Vitrina.
FUENTES BIBLIOGRÁFICAS	<p>ACIMED. SciELO: una metodología para la publicación electrónica, 2001. vol.9 supl.4, 9-22 p. ISSN 1024-9435.</p> <p>BOB BRANCHEK (Editor), et al. Edición Especial Microsoft SQL Server 6.5, Prentice Hall, Madrid, España. 1997. 26p</p> <p>DOMINGO BUONOCORE Diccionario de Bibliotecología, Marymar Ediciones, Segunda Edición Aumentada, 1976. 354p</p> <p>El modelo entidad-relación [en línea] http://www3.uji.es/~mmarques/f47/apun/node83.html [consultada febrero 17 2009]</p> <p>ELMARSÍ R Y NAVATHE S. Fundamentos de Sistemas de Bases de Datos tercera edición. Editorial Pearson. España. 2005 962p.</p> <p>FERRER A., PSET F., MORENO M. Y LLORET N. Guía Metodológica para la Implementación de una Biblioteca Digital Universitaria. Edición Trea S. L. España. 2005. 202p.</p> <p>GARCIA L. Y GARCIA E. Automatización de Bibliotecas. Editorial Arco/Libros S.L. Madrid, España. 1999. 285p.</p> <p>HAWRYSZKIEWYCZ, I.T, Análisis y Diseño de Base de Datos, Megabyte Noriega Editores, Mexico. 1994. 21 p.</p> <p>Ingeniería de Software Clase 3 [en línea] http://pisis.unalmed.edu.co/cursos/material/3004582/1/clase3_IS_0702-v1.ppt#256,1,Ingeniería de Software Clase 3 [consultada Mayo 30 2009]</p>

	<p>WILLIAM SAFFADY, Andrés Magaña García, Informática Documental para Bibliotecas, Traducido por Andrés Magaña García, Publicado por Ediciones Díaz de Santos, 1986. 309 p. ISBN 8486251478, 9788486251475</p> <p>JOSÉ YEPES(Editor), Diccionario Enciclopédico de Ciencias de la Documentación, Editorial Síntesis, S.A. 439-440p.</p> <p>PONJUÁN D.G. Biblioteca digital. Un nuevo paso en la evolución de las arquitecturas de información. Revista <i>Ciencias de la Información</i> 2002. Vol. 33, No. 1.</p>
CONTENIDO	<p>GLOSARIO INTRODUCCIÓN</p> <p>1 PLANTEAMIENTO DEL PROBLEMA</p> <p>1.1 ANTECEDENTES</p> <p>1.2 DESCRIPCIÓN DEL PROBLEMA</p> <p>1.3 JUSTIFICACION</p> <p>1.4 OBJETIVOS DE LA INVESTIGACIÓN</p> <p>1.4.1 OBJETIVO GENERAL</p> <p>1.4.2 OBJETIVOS ESPECÍFICOS</p> <p>1.5 ALCANCES Y LIMITACIONES DEL PROYECTO</p> <p>1.5.1 Alcances</p> <p>1.5.2 Limitaciones</p> <p>2 MARCO REFERENCIAL</p> <p>2.1 MARCO TEÓRICO – CONCEPTUAL</p> <p>2.1.1 Adquisición De Libros</p> <p>2.1.2 La catalogación</p> <p>2.1.3 Préstamo y Circulación</p> <p>2.1.4 Inventario</p> <p>2.1.5 Base de Datos</p> <p>2.1.6 Metodología de Desarrollo para el Aplicativo</p> <p>3 METODOLOGIA</p>

	3.1 ENFOQUE DE LA INVESTIGACIÓN
	3.2 LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA
	3.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN
	4 DESARROLLO DEL PROYECTO
	4.1 Primera Fase (Análisis)
	4.2 Segunda Fase (Diseño)
	4.2.1 Casos de Uso
	4.2.2 Desarrollo Base De Datos
	4.3 Tercera Fase (Desarrollo)
	4.3.1 Gestión de Libros
	4.3.1.1 Editar Área
	4.3.1.2 Editar Autor
	4.3.1.3 Editar Editorial
	4.3.1.4 Editar Libro
	4.3.1.5 Editar TipoLibro
	4.3.1.6 Editar TipoUsuario
	4.3.1.7 Editar Usuario
	4.3.1.8 Borrar Area
	4.3.1.9 Borrar Autor
	4.3.1.10 Borrar Editorial
	4.3.1.11 Borrar Libro
	4.3.1.12 Borrar TipoLibro
	4.3.1.13 Borrar Usuario
	4.3.1.14 Insertar Area
	4.3.1.14 Insertar Autor
	4.3.1.15 Insertar Editorial
	4.3.1.16 Insertar Libro
	4.3.1.17 Insertar Tipo Libro
	4.3.1.18 Insertar Tipo Usuario
	4.3.1.19 Insertar Usuario
	4.3.1.20 Seleccionar AdminCorreo
	4.3.1.21 Seleccionar Area
	4.3.1.22 Seleccionar Autor
	4.3.1.23 Seleccionar Editorial
	4.3.1.24 Seleccionar Libro
	4.3.1.25 Seleccionar Préstamo
	4.3.1.26 Seleccionar Tipo Libro
	4.3.1.27 Seleccionar Tipo Usuario
	4.3.1.28 Seleccionar Usuario
	4.3.2 Gestión Prestamo de Libros

	<p>4.3.2.1 Envío Correo Solicitud Préstamo</p> <p>4.3.2.2 Actualizar cantidad de libro</p> <p>4.3.2.3 Buscar por Autor</p> <p>4.3.2.4 Buscar por Libro</p> <p>4.3.2.5 Buscar por Editorial</p> <p>4.3.2.6 Buscar por Libro</p> <p>4.3.2.7 Libro asignado usuario</p> <p>4.3.2.8 Devolucion Libro</p> <p>4.4 Cuarta Fase (Pruebas)</p> <p>4.1.1 Registrarse</p> <p>4.1.2 Solicitud Préstamo de Libro</p> <p>4.1.2.1 Búsqueda por Autor</p> <p>4.1.2.2 Búsqueda por libro</p> <p>4.1.2.2 Búsqueda por Editorial</p> <p>4.1.3 Inserción de Libro</p> <p>4.1.3 Inserción de Área</p> <p>4.1.4 Inserción de Autor</p> <p>4.1.4 Inserción de Editorial</p> <p>4.1.4 Inserción de Tipo de Libro</p> <p>4.5 Retroalimentación</p> <p>BIBLIOGRAFÍA</p> <p>ANEXOS</p>
<p>METODOLOGÍA</p>	<p>ENFOQUE DE LA INVESTIGACIÓN</p> <p>Empírico-analítico cuyo interés es el técnico, orientado a la interpretación y transformación del mundo material y el modo de conocerlo y este aplicativo está propuesto para suplir las necesidades de la biblioteca del Colegio Eduardo Carranza</p> <p>LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA</p> <p>Línea de Investigación: Tecnologías Actuales y Sociedad.</p> <p>Sub-línea de Facultad: Sistemas de Información y Comunicación.</p> <p>Campo Temático del Programa: Algorítmica, Comercio Electrónico, Web, Alm. De Información</p> <p>TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN</p> <p>Se realizó entrevista al bibliotecario con el fin de conocer las necesidades de la biblioteca del colegio Eduardo Carranza, ya que él es la persona idónea que conoce las necesidades de la biblioteca</p>

**APLICATIVO TECNOLÓGICO PARA EL CONTROL DE INVENTARIOS,
CIRCULACIÓN Y PRÉSTAMO DE LIBROS DE LA BIBLIOTECA DEL
COLEGIO EDUARDO CARRANZA**

**MARIAYARA PÉREZ BARRETO
20043170009**

**UNIVERSIDAD DE SAN BUENAVENTURA
INGENIERÍA DE SISTEMAS
TECNOLOGIA EN SISTEMAS
BOGOTÁ, D.C
2009**

**APLICATIVO TECNOLÓGICO PARA EL CONTROL DE INVENTARIOS,
CIRCULACIÓN Y PRÉSTAMO DE LIBROS DE LA BIBLIOTECA DEL
COLEGIO EDUARDO CARRANZA**

**MARIAYARA PÉREZ BARRETO
20043170009**

**Proyecto presentado como requisito para optar por el título de
Tecnólogo en sistemas**

Asesor, Emilio Barajas, Ingeniero de Sistemas

**UNIVERSIDAD DE SAN BUENAVENTURA
INGENIERÍA DE SISTEMAS
TECNOLOGIA EN SISTEMAS
BOGOTA, D.C
2009**

Nota de aceptación:

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Firma del Asesor Metodológico

Bogota, D.C. 1 Junio 2009

CONTENIDO

Pág.

GLOSARIO

INTRODUCCIÓN

5 PLANTEAMIENTO DEL PROBLEMA

5.1 ANTECEDENTES

5.2 DESCRIPCIÓN DEL PROBLEMA

5.3 JUSTIFICACION

5.4 OBJETIVOS DE LA INVESTIGACIÓN

5.4.1 OBJETIVO GENERAL

5.4.2 OBJETIVOS ESPECÍFICOS

5.5 ALCANCES Y LIMITACIONES DEL PROYECTO

1.5.1 Alcances

1.5.2 Limitaciones

6 MARCO REFERENCIAL

6.1 MARCO TEÓRICO – CONCEPTUAL

2.1.1 Adquisición De Libros

2.1.2 La catalogación

2.1.3 Préstamo y Circulación

2.1.4 Inventario

2.1.5 Base de Datos

2.1.6 Metodología de Desarrollo para el Aplicativo

7 METODOLOGIA

- 7.1 ENFOQUE DE LA INVESTIGACIÓN
- 7.2 LÍNEA DE INVESTIGACIÓN DE USB /
SUB-LÍNEA DE FACULTAD /
CAMPO TEMÁTICO DEL PROGRAMA
- 7.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

8 DESARROLLO DEL PROYECTO

- 8.1 Primera Fase (Análisis)
- 8.2 Segunda Fase (Diseño)

4.2.1 Casos de Uso

4.2.2 Desarrollo Base De Datos

4.3 Tercera Fase (Desarrollo)

4.3.1 Gestión de Libros

4.3.1.1 Editar Área

4.3.1.2 Editar Autor

4.3.1.3 Editar Editorial

4.3.1.4 Editar Libro

4.3.1.5 Editar TipoLibro

4.3.1.6 Editar TipoUsuario

4.3.1.7 Editar Usuario

4.3.1.8 Borrar Area

4.3.1.9 Borrar Autor

4.3.1.10 Borrar Editorial

4.3.1.11 Borrar Libro

4.3.1.12 Borrar TipoLibro

4.3.1.13 Borrar Usuario

- 4.3.1.14 Insertar Area
- 4.3.1.14 Insertar Autor
- 4.3.1.15 Insertar Editorial
- 4.3.1.16 Insertar Libro
- 4.3.1.17 Insertar Tipo Libro
- 4.3.1.18 Insertar Tipo Usuario
- 4.3.1.19 Insertar Usuario
- 4.3.1.20 Seleccionar AdminCorreo
- 4.3.1.21 Seleccionar Area
- 4.3.1.22 Seleccionar Autor
- 4.3.1.23 Seleccionar Editorial
- 4.3.1.24 Seleccionar Libro
- 4.3.1.25 Seleccionar Préstamo
- 4.3.1.26 Seleccionar Tipo Libro
- 4.3.1.27 Seleccionar Tipo Usuario
- 4.3.1.28 Seleccionar Usuario
- 4.3.2 Gestión Prestamo de Libros
 - 4.3.2.1 Envio Correo Solicitud Préstamo
 - 4.3.2.2 Actualizar cantidad de libro
 - 4.3.2.3 Buscar por Autor
 - 4.3.2.4 Buscar por Libro
 - 4.3.2.5 Buscar por Editorial
 - 4.3.2.6 Buscar por Libro
 - 4.3.2.7 Libro asignado usuario

4.3.2.8 Devolucion Libro

4.4 Cuarta Fase (Pruebas)

4.1.1 Registrarse

4.1.2 Solicitud Préstamo de Libro

4.1.2.1 Búsqueda por Autor

4.1.2.2 Búsqueda por libro

4.1.2.2 Búsqueda por Editorial

4.1.3 Inserción de Libro

4.1.3 Inserción de Área

4.1.4 Inserción de Autor

4.1.4 Inserción de Editorial

4.1.4 Inserción de Tipo de Libro

4.5 Retroalimentación

BIBLIOGRAFÍA

ANEXOS

LISTA DE TABLAS

	Pág.
Tabla 1: Tablas Base de Datos.....	24
Tabla 2: Campos de Tabla AdminCorreo.....	24
Tabla 3: Campos de Tabla Área.....	25
Tabla 4: Campos de Tabla Autor.....	25
Tabla 5: Campos de Tabla DetallePrestamo.....	25
Tabla 6: Campos de Tabla Editorial.....	25
Tabla 7: Campos de Tabla Libro.....	25
Tabla 8: Campos de Tabla Préstamo.....	26
Tabla 9: Campos de Tabla TipoLibro.....	26
Tabla 10: Campos de Tabla Usuario.....	26

LISTA DE FIGURAS

	pág
Figura 1: Caso De Uso Registro Usuario.....	20
Figura 2: Solicitud De Libro Por Un Usuario.....	21
Figura 3: Devolución De Libro.....	22
Figura 4: Gestión De Libros.....	23
Figura 5: Diagrama Base De Datos.....	31

Lista de Anexos

Pág

Anexo A: Encuesta

Anexo B: Manual de Usuario

GLOSARIO

BASE DE DATOS: Una base de datos es un conjunto de información organizada. Contiene una colección de los registros que usted puede buscar, clasificar y analizar rápidamente.

CICLO DE VIDA DE SOFTWARE: El proceso que se sigue para construir, entregar y hacer evolucionar el software, desde la concepción de una idea hasta la entrega y el retiro del sistema.

MICROSOFT SQL SERVER: **Microsoft SQL Server** es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea, así como de tener unas ventajas que más abajo se describen.

MICROSOFT VISUAL STUDIO. NET: Es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#. Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

INTRODUCCIÓN

La gestión de una biblioteca es una tarea complicada y ardua, más en una institución educativa donde parte de las existencias de libros son propiedad de la Secretaria Distrital de Educación (Plan vitrina educativa) y la restante son libros adquiridos por la institución, esto hace que el bibliotecario tenga que discriminar en el inventario qué cantidades son de la Secretaria y cuáles de la institución. Es aquí donde se hace uso de las nuevas tecnologías de almacenamiento y gestión de información.

Este aplicativo está diseñado en tres módulos: consulta de libros, gestión de biblioteca (inserción, búsqueda, modificación, eliminación de libros) y gestión de préstamo (sacar, devolver libro), inventario; los cuales nos permiten tener un control de circulación e inventario de los libros en la biblioteca del colegio Eduardo Carranza.

1 PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES

A lo largo de la historia las bibliotecas han ido evolucionando desde procedimientos manuales a la incorporación de técnicas proporcionadas por el ordenador, hasta el empleo de los servicios y herramientas de Internet; esta evolución está abriendo la posibilidad de la puesta en funcionamiento de bibliotecas digitales y virtuales¹.

Para consolidar las bibliotecas digitales es necesario implementar una automatización de todos sus procesos: como inventario del recurso bibliográfico, definición de las categorías de usuario, servicios conforme a unas normas de acceso, organización y servicios de los recursos, lectura o consulta en sala, gestión de préstamo domiciliario, ordenación del material bibliográfico, encuadernación y restauración del material, entre otras cosas².

Las tecnologías no han resuelto otros problemas de equidad en el acceso a la información y aún existen millones de personas (niños, jóvenes y adultos) que no tienen ni han tenido acceso a los beneficios de la automatización de una biblioteca, ya que el progreso no llega a todos simultáneamente, las transformaciones tecnológicas ocurren a grandes velocidades, pero su masificación es bastante lenta y gradual; esto impone vivencias prolongadas de transición en los cuales coexisten los nuevos y antiguos sistemas. La biblioteca digital o electrónica, en la mayoría de los casos, coexiste y coexistirá todavía con la biblioteca tradicional con acervos impresos y manuscritos³. Para esto en este proyecto se pretende realizar un proceso base para organización, donde se realizara una automatización básica de una biblioteca donde los procesos todavía son manuales, desarrollando como un inicio, para ponerse a la vanguardia de la tecnología, una aplicación tecnológica para el inventario de libros y control de procesos de una biblioteca tradicional.

Después de la segunda guerra mundial ya se planteaba la posibilidad y necesidad de utilizar los adelantos tecnológicos, para crear nuevas formas de almacenar y mantener conocimiento, Posteriormente, en los sesentas, J.C.R. Licklider desarrollo una visión sobre cómo el uso de los ordenadores revolucionaría el funcionamiento de las bibliotecas, introduciendo el indexado "inteligente" de los documentos. En 1994 Gladner habla de las bibliotecas

¹ GARCIA L. Y GARCIA E. Automatización de Bibliotecas. Editorial Arco/Libros S.L. Madrid, España. 1999. 285p

² FERRER A., PSET F., MORENO M. Y LLORET N. Guía Metodológica para la Implementación de una Biblioteca Digital Universitaria. Edición Trea S. L. España. 2005. 202p.

³ PONJUÁN D.G. Biblioteca digital. Un nuevo paseo en la evolución de las arquitecturas de información. Revista Ciencias de la Información 2002. Vol. 33, No. 1.

digitales como un equipo informático de almacenamiento y comunicaciones junto con el software y el contenido necesario para reproducir, emular y extender los servicios previstos por las bibliotecas convencionales⁴

En 1996, Borgman sugirió la siguiente definición: “Las bibliotecas digitales son un conjunto de recursos electrónicos y de potencialidades técnicas asociadas para la creación, búsqueda y utilización de información. Son una extensión ampliada de los sistemas de almacenamiento y recuperación de información que manipulan datos digitales en cualquier medio (textos, imágenes, sonidos, imágenes estáticas o dinámicas) y que existen en redes distribuidas”⁵. En los trabajos que se están realizando en la actualidad se hacen sobre bibliotecas digitales, donde ya se pretende tener la información en un formato digital, para esto primero se inicio con la consolidación de bases de datos donde se encontraban los inventarios de los recursos de las diferentes bibliotecas y con las cuales se realizaban sistemas para circulación y préstamo, como búsqueda por parte de los usuarios.

Bases de datos y sistemas especializados para inventario y control de procesos de las bibliotecas ya se encuentran en todas las grandes bibliotecas de país, las de la mayoría de las grandes universidades; donde se encuentran como catálogos de recursos bibliográficos, donde se accede a la información relevante para encontrarla ubicación, tema, estado, etc. de los libros . La Biblioteca de la Universidad Autónoma de Colombia cuenta con bases de datos referenciales y en texto completo, en las distintas áreas del conocimiento. Estas se pueden consultar de manera fácil desde la página web de la Biblioteca, y dentro del campus universitario. Como esta en el país se encuentran un gran número de bibliotecas con catálogos y bases de datos con todos los recursos que tienen a disposición la biblioteca, también el estado de los mismos, su ubicación la cantidad y los que se encuentran en préstamo en sala o domiciliario. Entre estas bibliotecas se encuentran, según el catalogo de las bibliotecas nacionales del instituto de investigación Alexander Von Humboldt, la Biblioteca Luis Ángel Arango (Banco de la República), Biblioteca Nacional de Colombia, Pontificia Universidad Javeriana, Universidad de Antioquia, Universidad de Caldas, Universidad de los Andes, Universidad de Manizales, Universidad de San Buenaventura, Universidad del Norte, Universidad Eafit, Universidad Externado de Colombia, Universidad Jorge Tadeo Lozano, Universidad Nacional de Colombia, Universidad Tecnológica de Pereira, entre otras.

Un gran proyecto donde aparte de automatizar todos los procesos de la biblioteca ya se tienen verdaderas bibliotecas virtuales, donde los recursos bibliográficos en su totalidad se encuentran incluidos dentro de las bases de datos en formatos electrónicos. Scientific Electronic Library Online - es una biblioteca virtual para Latinoamérica, el Caribe, España y Portugal. SciELO Colombia es una biblioteca electrónica que cubre una colección selecta de

⁴FERRER *ET AL*, Op. Cit., 202p.

⁵ PONJUÁN, Op. Cit., Vol. 33, No. 1.

revistas científicas colombianas de todas las áreas del conocimiento. A demás de esto cuenta con un Comité Consultivo Nacional, conformado por el Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología Francisco José de Caldas Colciencias, la Organización Panamericana de la Salud-Representación Colombia, la Universidad Nacional de Colombia y representantes de los Editores. La biblioteca es parte de un proyecto que está siendo desarrollado por la FAPESP - Fundao de Amparo Pesquisa do Estado de So Paulo, en colaboración con BIREME - Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud. El proyecto FAPESP/BIREME contempla el desarrollo de una metodología común para la preparación, almacenamiento, diseminación y evaluación de literatura científica en formato electrónico. A través del portal SciELO.org, que integra y provee acceso a la red de sitios SciELO, se puede realizar búsquedas en las colecciones de SciELO existentes en toda la red o en un país en particular⁶.

Algunas de las iniciativas para bibliotecas digitales es por ejemplo: La universidad de California en Barkeley, donde tienen un proyecto sobre un sistema de planificación del medio ambiente e información geográfica, el cual pondrá en practica una biblioteca digital como un conjuntote servicios de la información distribuida, haciendo hincapié en la indexación automatizada, la recuperación inteligente, soporte por las bases de datos distribuidas, un mejor protocolo de recuperación cliente-servidor y una mejor tecnología de obtención de datos⁷.

1.2 DESCRIPCIÓN DEL PROBLEMA

El Colegio Distrital Eduardo Carranza cuenta en la actualidad con una biblioteca compuesta por dos grupos de libros, el primero correspondiente al plan vitrina educativa los cuales son donados por la Secretaria de Educación del Distrito y el segundo los libros adquiridos por la institución.

El préstamo de libros a los estudiantes se hace llenando un libro en el cual se registra el titulo, nombre del solicitante, documento de identificación, hora de entrada y hora de salida. El estudiante debe dejar el documento de identidad mientras tenga el libro.

Actualmente el control de inventario es deficiente ya que no se lleva ningún registro de los libros en préstamos o si están actualizados. Por lo tanto no se puede determinar si aun quedan libros disponibles.

Adicionalmente el colegio por petición del plan vitrina educativa debe reportar cada año a la secretaria de educación el estado y cantidad de libros. Fruto de este informe se identifico la pérdida de aproximadamente 300 libros, lo cual generó conflictos al interior de la institución.

⁶ ACIMED, Op. Cit., vol.9 supl.4, p9-22.

⁷ ELMARSI R, Navathe S. Fundamentos de Sistemas de Bases de Datos tercera edición. Editorial Pearson. España. 2005. p962.

¿Cómo mediante las tecnologías de la información y comunicaciones se pueden optimizar el control de inventarios, circulación y préstamo de libros en la biblioteca del Colegio Distrital Eduardo Carranza?

1.3 JUSTIFICACION

Con este proyecto se pretende organizar la información para reducir el tiempo de consulta y búsqueda de libros, mejorar el control del inventario, generar los informes requeridos por la Secretaria Distrital de Educación con respecto a los libros del Plan Vitrina; con esto se facilitara la labor tanto de los bibliotecarios como de los estudiantes y se fortalece el interés de los estudiantes por las ayudas bibliográficas que tienen a su disposición.

1.4 OBJETIVOS DE LA INVESTIGACIÓN

1.4.1 OBJETIVO GENERAL

Desarrollar un aplicativo tecnológico que permite el control (inventarios, circulación y préstamo) de los libros que se encuentran en la biblioteca del Colegio Distrital Eduardo Carranza.

1.4.2 OBJETIVOS ESPECÍFICOS

- Analizar el proceso de inventario, circulación y préstamo de la biblioteca del Colegio Distrital Eduardo Carranza.
- Diseñar una base de datos para el inventario, circulación y préstamo de los recursos bibliográficos del Colegio Distrital Eduardo Carranza.
- Desarrollar un aplicativo que permitirá el control (adquisición, catalogación, inventario y circulación) de los libros de la Biblioteca del Colegio Distrital Eduardo Carranza.
- Realizar pruebas del aplicativo.

1.5 ALCANCES Y LIMITACIONES DEL PROYECTO

1.5.1 Alcances

Mediante un aplicativo tecnológico desarrollar un sistema para la consulta del material de la biblioteca del Colegio Eduardo Carnaza y que este aplicativo sea administrado por el bibliotecario, esto con el fin de optimizar los procesos de la biblioteca.

Se diseñará y desarrollará una base de datos para el inventario de libros y un sistema de circulación y préstamo de éstos; se realizarán pruebas de su funcionamiento, esto con el fin de prestar un servicio a la comunidad educativa

de la institución y con esto se determine si su implantación es viable para que cada uno de ellos pueda disfrutar este proyecto.

1.5.2 Limitaciones

El aplicativo tecnológico realizado para el colegio Eduardo Santos no está diseñado para gestionar lista de pedidos de libros, ni tampoco puede ejecutar tareas tales como: gestión de matriculas de un estudiante, control de notas por parte de los docentes, procesos académicos tales como: control de asistencia, anotaciones en observatorio sobre conducta o control sobre nomina del personal administrativo.

2 MARCO REFERENCIAL

2.1 MARCO TEÓRICO – CONCEPTUAL

Para el desarrollo del aplicativo tecnológico para el control de inventarios, circulación y préstamo de libros de la biblioteca del colegio Eduardo Carranza se debe primero conocer cuáles son los servicios generales que se realizan en cualquier biblioteca, éstos por lo general son los de: adquisición, catalogación de libros, asignación de número topográfico, préstamo, también en toda biblioteca se realiza un inventario o recuento de libros, para saber la ubicación o estado de los mismos.

2.1.1 Adquisición De Libros La adquisición de libros tiene varios criterios a tener en cuenta según la biblioteca pero en términos generales los principios son:

- **Cuantitativos:** Este criterio tiene dos vertientes: la relación documentos/usuario, que es diferente según el tipo de biblioteca universitaria, pública, especializada, etc., y según el tipo de documento monografías, publicaciones periódicas, material no librario, etc., y el número de ejemplares de cada obra.
- **Cualitativos:** Se define que clase de libros se necesitan en la biblioteca sean estos literarios académicos de texto etc.
- **Demanda:** Necesidad de libros expresada por parte de los usuarios de la biblioteca también en este punto se tiene en cuenta la circulación de libros aspecto que se tocara mas adelante.
- **Neutralidad:** La selección y adquisición de fondos será realizada por el personal bibliotecario, pero en ningún momento deben intervenir en ello otros criterios diferentes de los puramente profesionales⁸.

2.1.2 La catalogación Es el proceso de identificar libros por medio de puntos de acceso (autor, título o tema) y la asignación de una signatura topográfica esto da como resultado el asiento o registro bibliográfico, esto se hace con el fin de facilitar la identificación formal y la localización física de los libros. Para que la catalogación tenga éxito debe ceñirse a las normas de catalogación normalizadas a nivel internacional.

⁸ Selección y adquisición de libros y materiales especializados. Criterios, fuentes y métodos. Planes cooperativos y centralizados [en línea] <http://www.geocities.com/zaguan2000/304.html> [consultada: 1 mayo 2009]

La catalogación es uno de los procesos que se ha automatizado mediante procedimientos informáticos la catalogación automática se inicia en 1983 en la biblioteca del Senado.

La catalogación automática ha permitido la manipulación sencilla de grandes cantidades de libros y/o documentos esto ha facilitado el flujo de información entre bibliotecas y la desaparición progresiva de las fichas manuales, ya que actualmente, debido al crecimiento del volumen de información es imposible mantener los servicios de una unidad de información sin recurrir a las tecnologías de la información.

La catalogación realizada de manera automática facilita el trabajo a los profesionales, mejora la eficacia de los servicios a los usuarios y permite una mejor gestión de los recursos económicos y humanos.

2.1.3 Préstamo y Circulación El servicio de préstamo consiste en la entrega, por parte de la biblioteca, a una persona, institución u otra biblioteca, para que, por un tiempo dado, tenga el uso de ella con la obligación de restituirla antes del vencimiento del plazo⁹.

Existen tres modalidades en el servicio de préstamo:

- a. Préstamo personal: La biblioteca presta el libro a un usuario particular para que lo consulte fuera de ésta, para este servicio tiene que existir un reglamento que el usuario acepta y un registro previo del usuario en la biblioteca.

Antes de la aparición de los sistemas de gestión de bibliotecas este servicio de préstamo personal representaba un gran esfuerzo tanto humano como de recursos por que se tenía que hacer manualmente por medio del sistema de papeletas de préstamo.

- b. Préstamo colectivo: Se realiza entre la biblioteca y una institución con la que se firma un convenio de préstamo y a la que se le otorga un carné que se suele denominar "carné institucional". La gestión automatizada del proceso, ha supuesto un mayor avance, si cabe, que en el caso del préstamo personal, ya que el préstamo colectivo presenta la dificultad añadida de una mayor complicación para la localización de los documentos una vez prestados
- c. Préstamo ínter bibliotecario: se realiza entre bibliotecas, aunque no es un servicio que realice la biblioteca por sí sola sino entre un grupo de bibliotecas organizadas a nivel nacional o local según recomienda la IFLA (La Federación Internacional de Asociaciones e Instituciones Bibliotecarias)¹⁰.

⁹ DOMINGO BUONOCORE Diccionario de Bibliotecología, Marymar Ediciones, Segunda Edición Aumentada, 1976. p354

¹⁰ JOSÉ YEPES(Editor), Diccionario Enciclopédico de Ciencias de la Documentación, Editorial Síntesis, S.A. p 439-440.

En el préstamo de libros aparece un recurso del manejo de las bibliotecas que no es un servicio sino un instrumento para el correcto funcionamiento de las bibliotecas y es el de circulación y control de circulación, La circulación de libros es una parte muy importante de toda biblioteca dado que la mayoría de usuarios de una biblioteca tienen la intención de sacar en préstamo un libro de las instalaciones de la biblioteca, la función principal del control de circulación es la de mantener un registro de los préstamos de libros, en los registros de circulación aparece qué usuario sacó el libro, el nombre del libro, la fecha de préstamo y la fecha de entrega, esto garantiza la protección del patrimonio de la biblioteca.

El proceso manual de circulación lo explica William Saffady en su libro *Informática documental para bibliotecas* “en la mayoría de los sistemas manuales, un fichero maestro de circulación contiene tarjetas o papeletas, cada una de las cuales contiene información sobre una determinada autorización de circulación. Dichas tarjetas contienen información específica, normalmente una breve descripción bibliográfica del materia en cuestión (generalmente, título y código de identificación), fecha en que se prestó, número de identificación del lector, su nombre y en algunos casos su dirección. El fichero maestro de circulación refleja siempre el material bibliográfico de la biblioteca que en un determinado momento está en condición de préstamo. El fichero suele estar clasificado de tal modo que facilita el conocimiento de la situación actual de cualquier artículo bibliográfico. En muchas bibliotecas públicas, por ejemplo, las tarjetas se clasifican, para facilitar el proceso, en base a la fecha de devolución. A si se identifican con facilidad los materiales que no se han devuelto en la fecha debida, sobre los que hay que tomar medidas adecuadas...”¹¹ este proceso de control de circulación ayuda no sólo al servicio de préstamo de libros sino también en el de inventario o recuento dado que el control de inventario o recuento es un proceso por el cual se toma la cantidad de libros que están en préstamo y se suma al total de libros físicos que se encuentra en la biblioteca y esta cantidad se coteja con la cantidad de libros que aparecen en el catálogo de la biblioteca. Este proceso de inventario o recuento también sirve para saber qué cantidad de libros están en buen estado o deteriorados, en cuanto a la periodicidad con que se realizan los inventarios se recomienda hacerlos cada año.

2.1.4 Inventario Desde el punto de vista administrativo, En la acepción más amplia de la palabra, los inventarios son recursos utilizables que se encuentran almacenados para su uso posterior en un momento determinado, existen 5 clases de inventarios los cuales son: Inventario de Mercancías, Inventario de Productos Terminados, Inventario de Productos en Proceso de Fabricación, Inventario de Materias Primas, el tipo de inventario que mas comúnmente se aplica a una biblioteca es el de Inventario de Mercancías, el manejo de la

¹¹WILLIAM SAFFADY, Andrés Magaña García, *Informática Documental para Bibliotecas*, Traducido por Andrés Magaña García, Publicado por Ediciones Díaz de Santos, 1986. 309 p. ISBN 8486251478, 9788486251475

mercancía en este caso libros se hace sobre los criterios de: pertenencia a uno de dos grupos que son: Plan vitrina educativa, libros que son suministrados por la Secretaría Distrital de Educación y libros adquiridos por la institución, o pertenencia a una área del saber determinada. En el Colegio Eduardo Carranza los inventarios de las colecciones de las bibliotecas son obligatorios ya que son requeridos por la secretaria de educación que auditan los recursos del Colegio, para esto el bibliotecario debe saber que libros tiene en existencia y que libros en préstamo y la cantidad detallada de estos.

2.1.5 Base de Datos Las bases de datos son esenciales para el sistema de información de una organización, el cual soporta las funciones de la organización al mantener los datos para éstas y auxilia a los usuarios al interpretar los datos para tomar decisiones. La base de datos juega un papel central en este proceso.

Los usuarios que deben tomar decisiones obtienen los datos necesarios al acceder la base y registrar su decisión en ella. La localización de la base de datos y las facilidades para accederla tienen una gran relación con la efectividad del sistema de información. El fácil acceso de una variedad de datos desde cierto número de lugares permite que el sistema de información responda con rapidez a las necesidades de quienes toman decisiones en la organización, en tanto que un acceso deficiente puede, por supuesto, obstaculizar una rápida respuesta. Si no se dispone con facilidad de los datos, las decisiones quizá se retarden en forma innecesaria o se tomen con datos incompletos, lo que puede conducir a un mal desempeño posterior del sistema.¹²

Es por ello que el diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

El diseño conceptual parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos. En el esquema conceptual se hace una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD que se vaya a utilizar para manipularla. Con el modelo conceptual, lenguaje que se utiliza para describir esquemas conceptuales, se tiene como objetivo describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información

A partir del esquema conceptual se realiza el diseño lógico y como resultado se obtiene un esquema lógico. En este esquema lógico se encuentra la descripción de la estructura de la base de datos en términos de las estructuras

¹² HAWRYSZKIEWYCZ, I.T, Análisis y Diseño de Base de Datos, Megabyte Noriega Editores, Mexico. 1994. 21 p.

de datos que puede procesar un tipo de SGBD. El diseño lógico depende del tipo de SGBD que se vaya a utilizar, no depende del producto concreto.

El diseño físico parte del esquema lógico y da como resultado un esquema físico donde se hace una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos.¹³

Para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional se usa la normalización, la cual se alude a la forma en que se implementan las relaciones y el almacenamiento de datos dentro de las tablas de las bases de datos. Cuando se normaliza una tabla, se intenta limitar la cantidad de datos redundantes que existen dentro de la tabla. Existen muchos niveles diferentes, o tipos, de normalización. Con toda probabilidad, el objetivo global va a ser una base de datos tercera forma normal (3NF). En la mayoría de los casos, éste será el mejor compromiso entre extremos cuando se trata de equilibrar la normalización con la funcionalidad y la facilidad de implementación. Existen niveles más allá de la 3NF, pero en la práctica hacen confuso el diseño de las bases de datos.¹⁴

2.1.6 Metodología de Desarrollo para el Aplicativo Actualmente se encuentra en el mercado software para la gestión de bibliotecas los cuales se revisan y se toman algunos como referencia para el desarrollo de nuestra aplicación. Programa de inventarios de libros bookbag de Wakefieldsoft CCL. Versión 4.6.7 interfaz intuitiva y fácil de manejar que permite tener un inventario de libros y clasificarlos por autor, género, título etc., Xiros v2.2.1 una aplicación fácil de manejar, que automatiza todos los procesos relacionados con préstamos, devoluciones, registro de libros, usuarios y consulta en línea del material de la biblioteca y Biblioteca 2000 v1.01b programa que contiene tres módulos: Administración, Consulta, y Gestión de préstamos, búsquedas simples, generar informes en Word, realizar copias de seguridad de tus datos y protegerlos mediante clave de acceso.

El desarrollo de este aplicativo va unido a un ciclo de vida compuesto por una serie de etapas que comprenden todas las actividades, desde el momento en que surge la idea de crear el software, hasta que el producto es entregado a los usuarios. Para el desarrollo de este aplicativo se utiliza el modelo incremental. Este modelo tiene la facilidad de comenzar un diseño, código y prueba del software y si no está satisfecho el cliente o el programador da la posibilidad de comenzar con una segunda iteración sin necesidad de realizar un nuevo análisis, se pueden realizar tantas iteraciones como sean necesarias. El proceso se divide en 4 partes.

¹³ El modelo entidad-relación[en línea] <http://www3.uji.es/~mmarques/f47/apun/node83.html> [consultada febrero 17 2009]

¹⁴ Bob Branchek (Editor), et al. Edición Especial Microsoft SQL Server 6.5, Prentice Hall, Madrid, España. 1997. 26p

- **Análisis:** Necesidades del cliente para el que se creará la aplicación, el documento resultante suele tener como origen una serie de entrevistas cliente-proveedor. Se determina que elementos intervienen en el sistema a desarrollar, así como su estructura, relaciones, evolución en el tiempo, detalle de sus funcionalidades.
- **Diseño:** Aquí se definirán en detalle entidades y relaciones de las bases de datos, se pasará de casos de uso esenciales a su definición como casos expandidos reales, se seleccionará el lenguaje más adecuado, el Sistema Gestor de Bases de Datos a utilizar en su caso, librerías, configuraciones hardware, redes, etc.).
- **Código:** En este punto se empieza a codificar algoritmos y estructuras de datos, definidos en las etapas anteriores, en el correspondiente lenguaje de programación y/o para un determinado sistema gestor de bases de datos.
- **Pruebas:** El objetivo de estas pruebas es garantizar que el sistema ha sido desarrollado correctamente, sin errores de diseño y/o programación. Es conveniente que sean planteadas a nivel de cada módulo¹⁵.

¹⁵ Ingeniería de Software Clase 3 [en línea]
http://pisis.unalmed.edu.co/cursos/material/3004582/1/clase3_IS_0702-v1.ppt#256,1, Ingeniería de Software Clase 3 [consultada febrero 17 2009]

3 METODOLOGIA

3.1 ENFOQUE DE LA INVESTIGACIÓN

Empírico-analítico cuyo interés es el técnico, orientado a la interpretación y transformación del mundo material y el modo de conocerlo y este aplicativo está propuesto para suplir las necesidades de la biblioteca del Colegio Eduardo Carranza

3.2 LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA

Línea de Investigación: Tecnologías Actuales y Sociedad.

Sub-línea de Facultad: Sistemas de Información y Comunicación.

Campo Temático del Programa: Algorítmica, Comercio Electrónico, Web, Alm. De Información

3.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

Se realizó entrevista al bibliotecario con el fin de conocer las necesidades de la biblioteca del colegio Eduardo Carranza, ya que él es la persona idónea que conoce las necesidades de la biblioteca

4 DESARROLLO DEL PROYECTO

4.1 Primera Fase (Análisis)

Recolección de información, entrevista con el bibliotecario del Colegio Eduardo Carranza para el planteamiento del problema. También se realizó entrevista con bibliotecólogos de la Biblioteca Luis Ángel Arango para conocer los procesos y servicios de ésta biblioteca con el fin de tener una visión más clara para el desarrollo del aplicativo

4.2 Segunda Fase (Diseño)

4.2.1 **Casos de Uso** Los casos de uso nos ayudarán a ver cómo debería interactuar el sistema con el usuario como lo muestran las figuras 1-4

Figura 1: Caso De Uso Registro Usuario

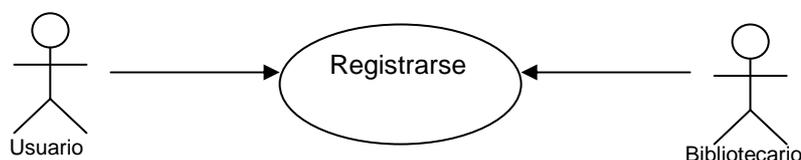


Tabla 1: Explicación Caso de Uso Registro Usuario

Actores	Usuario, Bibliotecario
Descripción	Se verifica los roles se llena el formulario de Datos personales, se permite los ingresos
Secuencia normal	El sistema evalúa si los datos introducidos son válidos. El sistema no deja finalizar el registro si los campos están vacíos.

Figura 2: Caso de Uso Solicitud De Libro Por Un Usuario

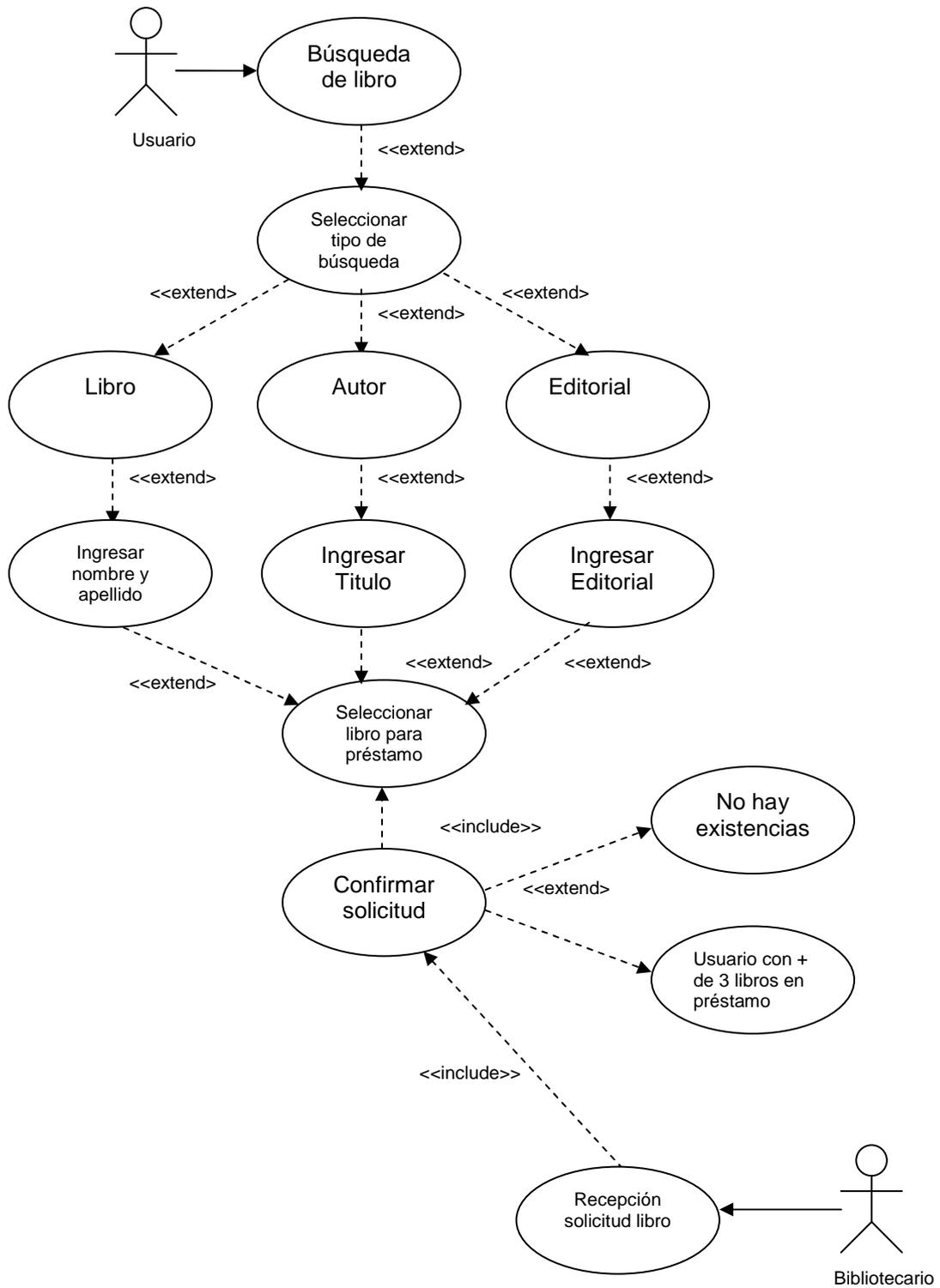


Tabla 2: Explicación Caso de Uso Solicitud De Libro Por Un Usuario

Actores	Usuario, Bibliotecario
Descripción	Un usuario registrado al sistema realiza su búsqueda por libro, autor, editorial luego de seleccionar su búsqueda se genera lista de libros, selecciona el libro lo solicita y confirma. El bibliotecario recibe la solicitud entrega el libro, el sistema actualiza el catalogo de libros
Condiciones	Cuando el usuario confirma su solicitud el sistema verifica que no exceda la cantidad de libro permitidos (3 máximo) El sistema verifica y envía un mensaje al usuario si no hay existencia de libro solicitado.

Figura 3: Caso de Uso Devolución De Libro

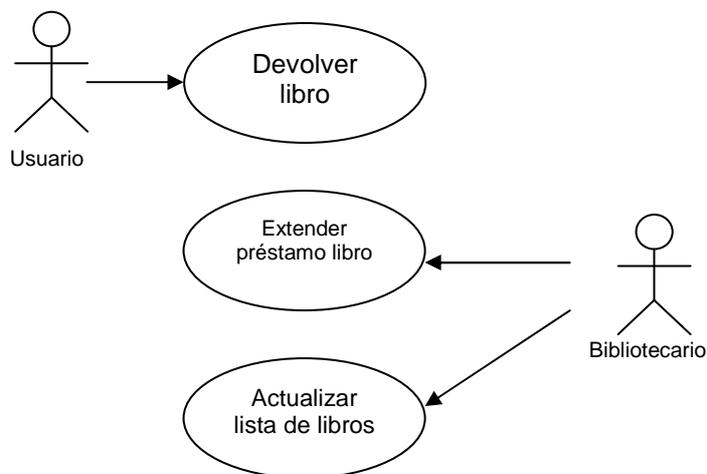


Tabla 3: Explicación Caso de Uso Devolución de libro

Actores	Usuario, Bibliotecario
Descripción	El usuario regresa el libro que tomo en préstamo, el bibliotecario lo recibe y actualiza el Catalogo de libros

Condiciones	El usuario si desea extender el préstamo el bibliotecario consulta en el sistema y genera el nuevo plazo.
--------------------	---

Figura 4: Caso de Uso Gestión De Libros

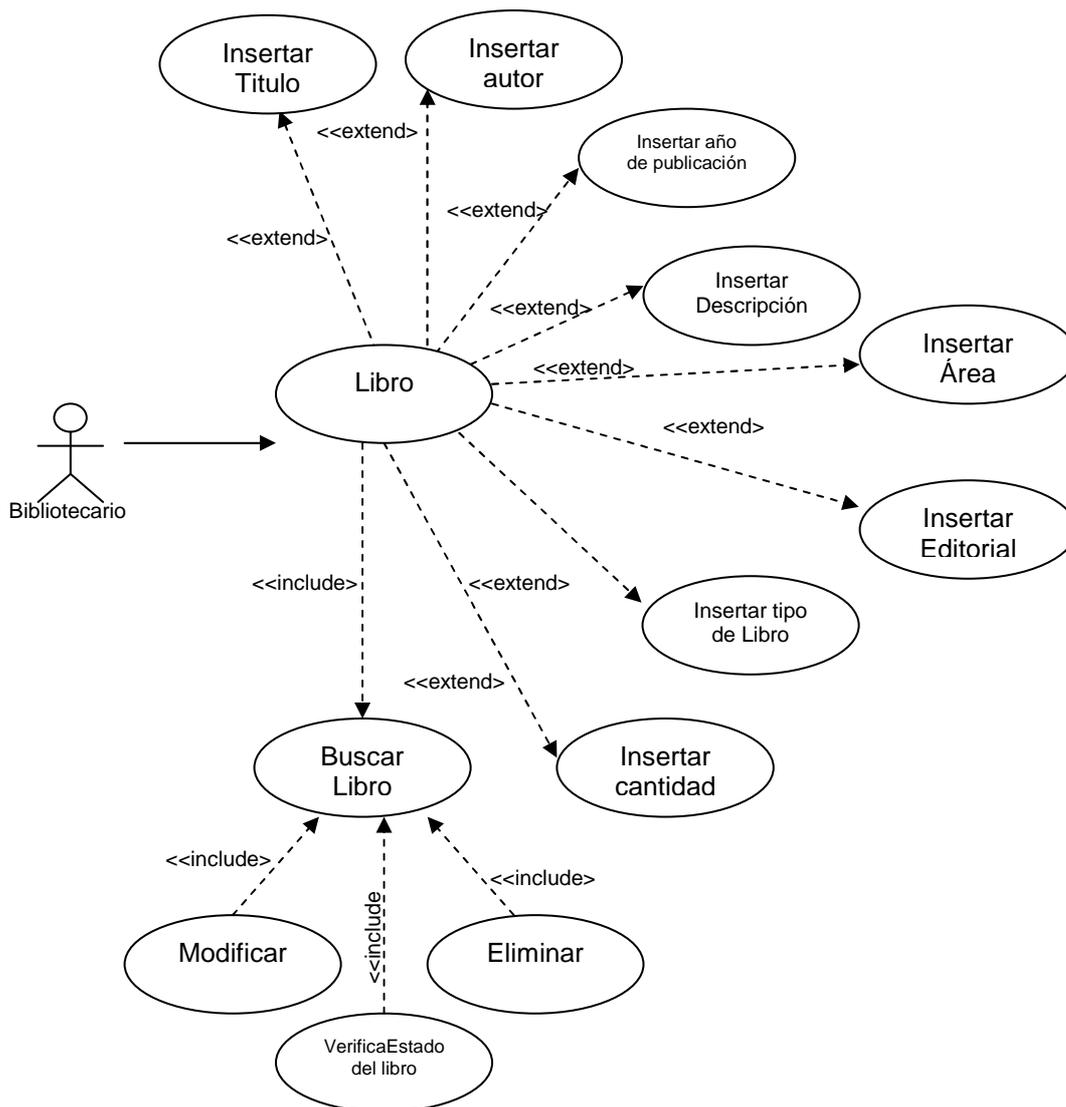


Tabla 4: Explicación Caso de Uso Gestión de libros

Actores	Bibliotecario
Descripción	El bibliotecario inserta un nuevo libro al sistema. Introduce los datos llenando los campos de ISBN, titulo, autor, editorial, tipo de libro, año de publicación, área, descripción, cantidad. El sistema guarda la información. En cualquier momento puede cancelar la inserción. Puede buscar un libro ingresando DEL

	Nº Topográfico, autor, año de publicación, editorial, título, área o descripción.
Condiciones	Si no existe autor en la base de datos continuación se introduce un nuevo autor y esto mismo sucede para editorial, año publicación, área, tipo de libro. Para modificar, eliminar, verificar estado del libro desde la opción de buscar libro.

4.2.2 Desarrollo Base De Datos

Las funciones que necesita la base de datos son las de adquisición, catalogación, préstamo, circulación e inventarios de los libros.

- Modelo Conceptual: Para desarrollar la base de datos utilizamos el modelo de entidad/relación en este modelo identificamos las entidades (cosa, persona, concepto abstracto o suceso en la que se recoge información) atributos (característica de interés o un hecho sobre la entidad) y relaciones (asociación entre entidades).
- Modelo Lógico: En este modelo utilizamos la normalización, pues nos ayuda a que cada tabla tenga datos sobre un solo tema, estos datos deben ser independientes de la información contenida en otras tablas. Se debe tener cuidado para que la información no se duplique en otras tablas y que si se elimina un registro en una tabla no afecte los registros en otra. También es importante que los campos en una tabla no se dupliquen. Se crea el diccionario de datos como lo muestran las tablas 1-10

Diccionario De Datos

Tabla 4: Tablas Base de Datos

Nombre de Tablas	Descripción
AdminCorreo	Administrador de Correo
Área	Asignatura, materia, tema del libro
Autor	Datos del autor del libro
DetallePrestamo	Detalle de registro de préstamo de los libros
Editorial	Datos de la editorial del libro
Libro	Datos del libro
Préstamo	Datos préstamo del libro
TipoLibro	Procedencia del libro
Usuario	Datos del usuario

Descripción Tablas

Tabla 5: Campos de Tabla AdminCorreo

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
CuentaCorreo	Int	Not Null	Cuenta Correo	No	No
CuentaCorreoEnvia	nvarchar(50)	Not Null	Envia Correo	No	No
Contraseña	nvarchar(50)	Not Null	Contraseña	No	No
ServidorCorreo	nvarchar(50)	Not Null	Servidor de Correo	No	No
Puerto	Int	Not Null	Puerto	No	No
SSL	Boolean	Not Null	Seguridad Autenticada	No	No

Tabla 6: Campos de Tabla Área

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
IdArea	Int	Not Null	Clave primaria	Sí	No
Descripcion	nvarchar(50)	Not Null	Nombre Área	No	No

Tabla 7: Campos de Tabla Autor

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
IdAutor	Int	Not Null	Clave primaria	Sí	No
Nombre	nvarchar(50)	Not Null	Nombres autor	No	No
Apellido	nvarchar(50)	Not Null	Apellidos autor	No	

Tabla 8: Campos de Tabla DetallePrestamo

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
IdPrestamo	Int	Not Null	Clave primaria	Sí	No
IdLibro	Int	Not Null	Relación con tabla "Libro"	No	Sí
Cantidad	Int	Not Null	Cantidad	No	No

Tabla 9: Campos de Tabla Editorial

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
idEditorial	Int	Not Null	Clave primaria	Sí	No
Editorial	nvarchar(50)	Not Null	Editorial	No	No

Tabla 10: Campos de Tabla Libro

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
IdLibro	Int	Not Null	Clave primera	Sí	No
Titulo	nvarchar(50)	Not Null	Titulo libro	No	No

Descripcion	nvarchar(50)	Not Null	Descripción del libro	No	No
AnoPublicacion	Datetime	Not Null	Año de publicación del libro	No	No
Cantidad	Int	Not Null	Existencias de un libro	No	No
idEditorial	Int	Not Null	Relación con la tabla "editorial"	No	Sí
Diarrea	Int	Not Null	Relacion con la tabla "Area"	No	Sí
IdAutor	Int	Not Null	Relación con la tabla "Autor"	No	Sí
idTipoLibro	Int	Not Null	Relación con la tabla "TipoLibro"	No	Sí

Tabla 11: Campos de Tabla Préstamo

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
IdPrestamo	Int	Not Null	Clave primaria	Sí	No
Estado	nvarchar(50)	Not Null	Cómo se encuentra el libro (prestado, libre)	No	No
FechaPrestamo	datetime	Not Null	Fecha del préstamo	No	No
IdUsuario	varchar(15)	Not Null	Relación con la tabla "Usuario"	No	Sí
FechaDevolucion	datetime	Not Null	Fecha devolución libro	No	No

Tabla 12: Campos de Tabla TipoLibro

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
idTipoLibro	Int	Not Null	Clave primaria	No	No
TipoLibro	nvarchar(50)	Not Null	Procedencia del libro	No	No

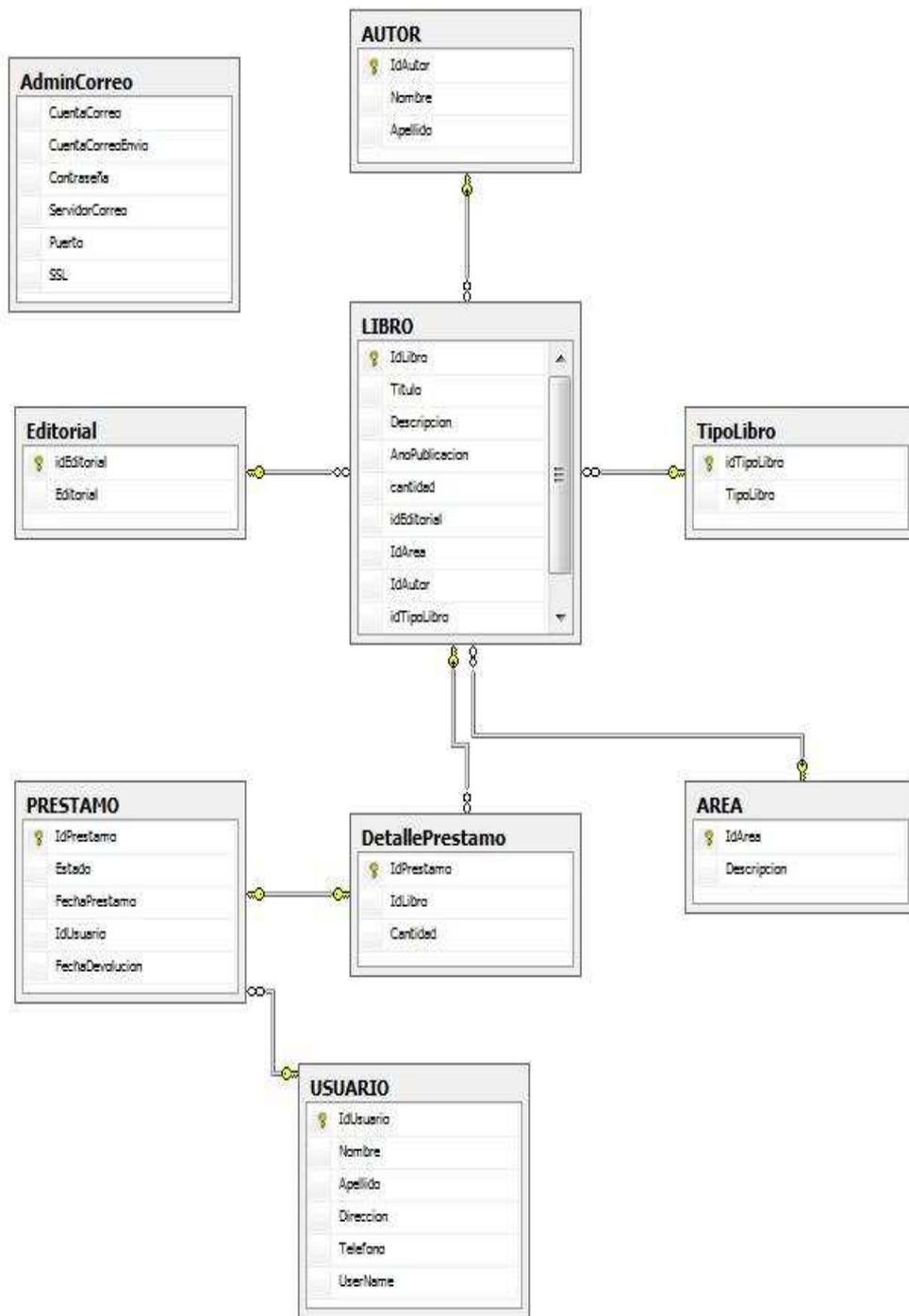
Tabla 13: Campos de Tabla Usuario

Nombre	Tipo	Null?	Descripción	Es PK?	Es FK?
IdUsuario	varchar(15)	Not Null	Clave primaria	Sí	No

Nombre	varchar(50)	Not Null	Nombres usuario	No	No
Apellido	varchar(50)	Not Null	Apellidos usuario	No	No
Direccion	varchar(50)	Not Null	Dirección del usuario	No	No
Telefono	Int	Not Null	Teléfono del usuario	No	No
UserName	varchar(50)	Not Null	Nombre de usuario	No	No

Luego de definir las tablas a cada una las identificamos con una clave primaria esto nos sirve para relacionarlas con las demás tablas, Por ejemplo, la tabla AUTOR y la tabla LIBRO se encuentra en una relación de uno a muchos, lo que significa que un autor se puede relacionar con un varios libros, como vemos en la siguiente Figura 5.

Figura 5: Diagrama Base De Datos



- Modelo Físico: Estructura del SGBD utilizando SQL Server

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[AUTOR]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[AUTOR](
    [IdAutor] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [nvarchar](50) NOT NULL,
    [Apellido] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_AUTOR] PRIMARY KEY CLUSTERED
(
    [IdAutor] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[Editorial]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[Editorial](
    [idEditorial] [int] IDENTITY(1,1) NOT NULL,
    [Editorial] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Editorial] PRIMARY KEY CLUSTERED
(
    [idEditorial] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[TipoLibro]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[TipoLibro](

```

```

        [idTipoLibro] [int] IDENTITY(1,1) NOT NULL,
        [TipoLibro] [nvarchar](50) NOT NULL,
        CONSTRAINT [PK_Tipo_Libro] PRIMARY KEY CLUSTERED
    (
        [idTipoLibro] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
    ) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[AREA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[AREA](
        [IdArea] [int] IDENTITY(1,1) NOT NULL,
        [Descripcion] [nvarchar](50) NOT NULL,
        CONSTRAINT [PK_AREA] PRIMARY KEY CLUSTERED
    (
        [IdArea] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
    ) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[USUARIO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[USUARIO](
        [IdUsuario] [varchar](15) NOT NULL,
        [Nombre] [varchar](50) NULL,
        [Apellido] [varchar](50) NULL,
        [Direccion] [varchar](50) NULL,
        [Telefono] [int] NULL,
        [UserName] [varchar](50) NULL,
        CONSTRAINT [PK_USUARIO] PRIMARY KEY CLUSTERED
    (
        [IdUsuario] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
    ) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[LIBRO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[LIBRO](
        [IdLibro] [int] NOT NULL,
        [Titulo] [nvarchar](50) NOT NULL,
        [Descripcion] [nvarchar](50) NOT NULL,
        [AnoPublicacion] [datetime] NOT NULL,

```

```

        [cantidad] [int] NOT NULL,
        [idEditorial] [int] NULL,
        [IdArea] [int] NOT NULL,
        [IdAutor] [int] NOT NULL,
        [idTipoLibro] [int] NULL,
    CONSTRAINT [PK_LIBRO] PRIMARY KEY CLUSTERED
    (
        [IdLibro] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
    ) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[PRESTAMO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[AdminCorreo](
    [CuentaCorreo] [varchar](50) NOT NULL,
    [CuentaCorreoEnvio] [varchar](50) NOT NULL,
    [Contraseña] [varchar](50) NOT NULL,
    [ServidorCorreo] [varchar](50) NOT NULL,
    [Puerto] [int] NOT NULL,
    [SSL] [bit] NOT NULL
) ON [PRIMARY]

GO
CREATE TABLE [dbo].[PRESTAMO](
    [IdPrestamo] [int] IDENTITY(1,1) NOT NULL,
    [Estado] [nvarchar](50) NOT NULL,
    [FechaPrestamo] [datetime] NOT NULL,
    [IdUsuario] [varchar](15) NULL,
    [FechaDevolucion] [datetime] NOT NULL,
    CONSTRAINT [PK_PRESTAMO] PRIMARY KEY CLUSTERED
    (
        [IdPrestamo] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
    ) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DetallePrestamo]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DetallePrestamo](
    [IdPrestamo] [int] NOT NULL,
    [IdLibro] [int] NOT NULL,
    [Cantidad] [int] NOT NULL,
    CONSTRAINT [PK_DetallePrestamo_1] PRIMARY KEY CLUSTERED
    (
        [IdPrestamo] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
    ) ON [PRIMARY]
END

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_LIBRO_AREA]') AND parent_object_id =
OBJECT_ID(N'[dbo].[LIBRO]'))
ALTER TABLE [dbo].[LIBRO] WITH CHECK ADD CONSTRAINT [FK_LIBRO_AREA]
FOREIGN KEY([IdArea])
REFERENCES [dbo].[AREA] ([IdArea])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_LIBRO_AUTOR]') AND parent_object_id =
OBJECT_ID(N'[dbo].[LIBRO]'))
ALTER TABLE [dbo].[LIBRO] WITH CHECK ADD CONSTRAINT [FK_LIBRO_AUTOR]
FOREIGN KEY([IdAutor])
REFERENCES [dbo].[AUTOR] ([IdAutor])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_LIBRO_Editorial]') AND parent_object_id =
OBJECT_ID(N'[dbo].[LIBRO]'))
ALTER TABLE [dbo].[LIBRO] WITH CHECK ADD CONSTRAINT
[FK_LIBRO_Editorial] FOREIGN KEY([idEditorial])
REFERENCES [dbo].[Editorial] ([idEditorial])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_LIBRO_Tipo_Libro]') AND parent_object_id =
OBJECT_ID(N'[dbo].[LIBRO]'))
ALTER TABLE [dbo].[LIBRO] WITH CHECK ADD CONSTRAINT
[FK_LIBRO_Tipo_Libro] FOREIGN KEY([idTipoLibro])
REFERENCES [dbo].[TipoLibro] ([idTipoLibro])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_PRESTAMO_USUARIO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[PRESTAMO]'))
ALTER TABLE [dbo].[PRESTAMO] WITH CHECK ADD CONSTRAINT
[FK_PRESTAMO_USUARIO] FOREIGN KEY([IdUsuario])
REFERENCES [dbo].[USUARIO] ([IdUsuario])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DetallePrestamo_LIBRO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[DetallePrestamo]'))
ALTER TABLE [dbo].[DetallePrestamo] WITH CHECK ADD CONSTRAINT
[FK_DetallePrestamo_LIBRO] FOREIGN KEY([IdLibro])
REFERENCES [dbo].[LIBRO] ([IdLibro])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DetallePrestamo_PRESTAMO]') AND parent_object_id
= OBJECT_ID(N'[dbo].[DetallePrestamo]'))
ALTER TABLE [dbo].[DetallePrestamo] WITH CHECK ADD CONSTRAINT
[FK_DetallePrestamo_PRESTAMO] FOREIGN KEY([IdPrestamo])
REFERENCES [dbo].[PRESTAMO] ([IdPrestamo])

```

4.3 Tercera Fase (Desarrollo)

La Tecnología de programación que se utilizará será Visual Studio .Net 2005 Professional. Ya que utiliza un ambiente completamente gráfico que facilita la creación de interfaces y la programación

Se utilizará el sistema de gestión de base de datos llamado Sql Server 2005 funciona bien con visual Studio .Net Interfaz Grafica para Administración de BD, creación de Procedimientos Almacenados, Usuarios, Vistas y Triggers.

En SQL Server 2005 Se crean stored procedure y luego los servicios web en Visual Studio. Net 2005

4.3.1 Gestión de Libros

4.3.1.1 Editar Área Stored Procedure

Procedimiento almacenado donde se utiliza update para modificar campos en este caso se esta modificando el campo descripcion

```
Create procedure [dbo].[actualizar_area]
(
    @idarea varchar(50),    'parametro'
    @descripcion varchar(50) 'parametro'
)
AS
UPDATE area SET descripcion= @descripcion 'se actualiza descripcion'
WHERE idarea= @idarea 'donde el valor que esta en idarea es = @idarea'
```

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado actualizar_area

```
Imports System.Web
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Data.SqlClient

<WebService(Namespace:="http://tempuri.org/")> _
<WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Public Class Service
    Inherits System.Web.Services.WebService

    <WebMethod()> _
    Public Function actualizar_area(ByVal idarea As String, ByVal
descripcion As String) As String
        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            Dim cmd As SqlCommand = New SqlCommand("actualizar_area",
conn)
```

```

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@idarea",
Data.SqlDbType.VarChar, 50)).Value = idarea.ToString
        cmd.Parameters.Add(New SqlParameter("@descripcion",
Data.SqlDbType.VarChar, 50)).Value = descripcion.ToString
        conn.Open()
        Dim i As Integer
        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString

    Catch ex As Exception

    End Try

End Function

```

4.3.1.2 Editar Autor

Stored Procedure

Procedimiento almacenado donde se utiliza update para modificar campos en este procedimiento se estan modificando los campos nombre y apellido

```

Create procedure [dbo].[actualizar_autor]

(
    @idautor varchar(50),    'parametro'
    @nombre varchar(50),    'parametro'
    @apellido varchar(50)   'parametro'
)
AS
UPDATE autor SET nombre= @nombre, apellido = @apellido 'se actualiza
nombre, apellido'

WHERE idautor= @idautor 'donde el valor que esta en idautor es =
@idautor'

```

Visual. Net Web service

Colocar parámetros

WebMethod se crea para llamar el procedimiento almacenado actualizar_autor

```

<WebMethod()> _
    Public Function actulizar_autor(ByVal idautor As String, ByVal
nombre As String, ByVal apellido As String) As String
        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            Dim cmd As SqlCommand = New SqlCommand("actualizar_autor",
conn)

            cmd.CommandType = Data.CommandType.StoredProcedure

```

```

        cmd.Parameters.Add(New SqlParameter("@idautor",
Data.SqlDbType.VarChar, 50)).Value = idautor.ToString
        cmd.Parameters.Add(New SqlParameter("@nombre",
Data.SqlDbType.VarChar, 50)).Value = nombre.ToString
        cmd.Parameters.Add(New SqlParameter("@apellido",
Data.SqlDbType.VarChar, 50)).Value = apellido.ToString
        conn.Open()
        Dim i As Integer
        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString

    Catch ex As Exception

    End Try

End Function

```

4.3.1.3 Editar Editorial

Stored Procedure

Procedimiento almacenado donde se utiliza update para modificar campos en este caso se esta modificando el campo editorial

```

Create procedure [dbo].[actualizar_editorial]

(
    @ideditorial varchar(50),      'parametro'
    @editorial varchar(50)        'parametro'
)
AS
UPDATE editorial SET editorial= @editorial 'se actualiza descripcion'
WHERE ideditorial= @ideditorial'donde el valor que esta en ideditorial
es = @ideditorial'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado actualizar_editorial

Colocar parámetros

```

<WebMethod(> _
    Public Function actulizar_editorial(ByVal ideditorial As String,
ByVal editorial As String) As String
        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            Dim cmd As SqlCommand = New
SqlCommand("actualizar_editorial", conn)
            cmd.CommandType = Data.CommandType.StoredProcedure

```

```

        cmd.Parameters.Add(New SqlParameter("@ideditorial",
Data.SqlDbType.VarChar, 50)).Value = ideditorial.ToString
        cmd.Parameters.Add(New SqlParameter("@editorial",
Data.SqlDbType.VarChar, 50)).Value = editorial.ToString
        conn.Open()
        Dim i As Integer
        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString

    Catch ex As Exception

    End Try

End Function

```

4.3.1.4 Editar Libro

Stored Procedure

Procedimiento almacenado donde se utiliza update para modificar campos en este caso se estan modificando titulo, descripcion, anopublicacion, cantidad, ideditorial, idarea, idautor, idtipolibro

```

create procedure [dbo].[actualizar_libro]

(
    @idlibro varchar(50),          'parametro'
    @titulo varchar(50),          'parametro'
    @descripcion varchar(50),     'parametro'
    @anopublicacion varchar(50),  'parametro'
    @cantidad varchar(20),        'parametro'
    @ideditorial int,             'parametro'
    @idarea int,                  'parametro'
    @idautor int,                 'parametro'
    @idtipolibro int              'parametro'
)
AS
UPDATE libro SET titulo= @titulo, descripcion= @descripcion,
anopublicacion= @anopublicacion, cantidad= @cantidad,
ideditorial=@ideditorial, idarea=@idarea, idautor=@idautor,
idtipolibro=@idtipolibro 'se actualiza titulo, descripcion,
anopublicacion, cantidad, ideditorial, idarea, idautor, idtipolibro'

WHERE idlibro= @idlibro 'donde el valor que esta en idlibro es =
@idlibro'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado actualizar_libroall

Colocar parámetros

```

    <WebMethod()> _
    Public Function actualizar_libroall(ByVal idlibro As String, ByVal
titulo As String, ByVal descripcion As String, ByVal anopublicacion As
String, ByVal cantidad As String, ByVal editorial As Integer, ByVal
area As Integer, ByVal autor As Integer, ByVal tipolibro As Integer)
As String
        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            Dim cmd As SqlCommand = New SqlCommand("actualizar_libro",
conn)

            cmd.CommandType = Data.CommandType.StoredProcedure
            cmd.Parameters.Add(New SqlParameter("@idlibro",
Data.SqlDbType.VarChar, 50)).Value = idlibro
            cmd.Parameters.Add(New SqlParameter("@titulo",
Data.SqlDbType.VarChar, 50)).Value = titulo
            cmd.Parameters.Add(New SqlParameter("@descripcion",
Data.SqlDbType.VarChar, 50)).Value = descripcion
            cmd.Parameters.Add(New SqlParameter("@anopublicacion",
Data.SqlDbType.DateTime)).Value = anopublicacion
            cmd.Parameters.Add(New SqlParameter("@cantidad",
Data.SqlDbType.VarChar, 50)).Value = cantidad
            cmd.Parameters.Add(New SqlParameter("@ideditorial",
Data.SqlDbType.Int)).Value = editorial
            cmd.Parameters.Add(New SqlParameter("@idarea",
Data.SqlDbType.Int)).Value = area
            cmd.Parameters.Add(New SqlParameter("@idautor",
Data.SqlDbType.Int)).Value = autor
            cmd.Parameters.Add(New SqlParameter("@idtipolibro",
Data.SqlDbType.Int)).Value = tipolibro

            conn.Open()
            Dim i As Integer
            i = cmd.ExecuteNonQuery
            conn.Close()
            Return i.ToString

        Catch ex As Exception

        End Try
    End Function

```

4.3.1.5 Editar TipoLibro

Stored Procedure

Procedimiento almacenado donde se utiliza update para modificar campos en este caso se esta modificando tipolibro

```

create procedure [dbo].[actualizar_tipolibro]
(
    @idtipolibro varchar(50),      'parametro'
    @tipolibro varchar(50)        'parametro'
)
AS

```

```
UPDATE tipolibro SET tipolibro= @tipolibro'se actualiza descripcion'  
WHERE idtipolibro= @idtipolibro'donde el valor que esta en idtipolibro  
es = @idtipolibro'
```

Visual. Net

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado actualizar_tipolibro

Colocar parámetros

```
<WebMethod()> _  
    Public Function actualizar_tipolibro(ByVal idtipolibro As String,  
    ByVal tipolibro As String) As String  
        Try  
            Dim conn As SqlConnection = New SqlConnection("data  
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")  
            Dim cmd As SqlCommand = New  
SqlCommand("actualizar_tipolibro", conn)  
            cmd.CommandType = Data.CommandType.StoredProcedure  
            cmd.Parameters.Add(New SqlParameter("@idtipolibro",  
Data.SqlDbType.VarChar, 50)).Value = idtipolibro.ToString  
            cmd.Parameters.Add(New SqlParameter("@tipolibro",  
Data.SqlDbType.VarChar, 50)).Value = tipolibro.ToString  
            conn.Open()  
            Dim i As Integer  
            i = cmd.ExecuteNonQuery  
            conn.Close()  
            Return i.ToString  
  
            Catch ex As Exception  
  
            End Try  
  
        End Function
```

4.3.1.6 Editar TipoUsuario

Procedimiento almacenado donde se utiliza update para modificar campos en este caso se esta modificando el campo descripcion

Stored Procedure

```
Create procedure [dbo].[actualizar_tipousuario]
```

```

(
    @idtipousuario varchar(50), 'parametro'
    @descripcion varchar(50) 'parametro'
)
AS
UPDATE tipousuario SET descripcion= @descripcion 'se actualiza
descripcion'
WHERE idtipousuario= @idtipousuario'donde el valor que esta en
idtipousuario es = @idtipousuario'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado actualizar_tipousuario

Colocar parámetros

```

<WebMethod()> _
    Public Function actualizar_tipousuario(ByVal idtipousuario As
String, ByVal descripcion As String) As String
        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            Dim cmd As SqlCommand = New
SqlCommand("actualizar_tipousuario", conn)
            cmd.CommandType = Data.CommandType.StoredProcedure
            cmd.Parameters.Add(New SqlParameter("@idtipousuario",
Data.SqlDbType.VarChar, 50)).Value = idtipousuario.ToString
            cmd.Parameters.Add(New SqlParameter("@descripcion",
Data.SqlDbType.VarChar, 50)).Value = descripcion.ToString
            conn.Open()
            Dim i As Integer
            i = cmd.ExecuteNonQuery
            conn.Close()
            Return i.ToString

        Catch ex As Exception

        End Try

    End Function

```

4.3.1.7 Editar Usuario

Stored Procedure

Procedimiento almacenado donde se utiliza update para modificar campos en este caso se estan modificando nombre, apellido, direccion, telefono

```

Create procedure [dbo].[actualizar_usuario]
(
    @idusuario varchar(50), 'parametro'

```

```

        @nombre varchar(50),      'parametro'
        @apellido varchar(50),    'parametro'
        @direccion varchar(50),   'parametro'
        @telefono varchar(50)     'parametro'
    )

AS
UPDATE usuario
SET nombre= @nombre, apellido= @apellido, direccion= @direccion,
telefono= @telefono 'se actualiza nombre, apellido, direccion
telefono'

WHERE idusuario= @idusuario 'donde el valor que esta en idusuario es =
@idusuario'

```

Visual. Net Web Service

Colocar parámetros

Se crea un WebMethod para llamar el procedimiento almacenado actualizar_usuario

```

<WebMethod(> _
    Public Function actualizar_usuario(ByVal idusuario As String, ByVal
nombre As String, ByVal apellido As String, ByVal direccion As String,
ByVal telefono As String) As String
        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            Dim cmd As SqlCommand = New
SqlCommand("actualizar_usuario", conn)
            cmd.CommandType = Data.CommandType.StoredProcedure
            cmd.Parameters.Add(New SqlParameter("@idusuario",
Data.SqlDbType.VarChar, 50)).Value = idusuario.ToString
            cmd.Parameters.Add(New SqlParameter("@nombre",
Data.SqlDbType.VarChar, 50)).Value = nombre.ToString
            cmd.Parameters.Add(New SqlParameter("@apellido",
Data.SqlDbType.VarChar, 50)).Value = apellido.ToString
            cmd.Parameters.Add(New SqlParameter("@direccion",
Data.SqlDbType.VarChar, 50)).Value = direccion.ToString
            cmd.Parameters.Add(New SqlParameter("@telefono",
Data.SqlDbType.VarChar, 50)).Value = telefono.ToString
            conn.Open()
            Dim i As Integer
            i = cmd.ExecuteNonQuery
            conn.Close()
            Return i.ToString

        Catch ex As Exception

        End Try

    End Function

```

4.3.1.8 Borrar Area

Stored Procedure

Procedimiento almacenado donde se utiliza delete para eliminar filas

```
create procedure [dbo].[borrar_area]
(
    @idarea varchar(50) 'parametro'
)
as
delete from area 'se borra la fila'

where idarea = @idarea 'donde el valor que esta en idarea es =
@idarea'
```

Visual. Net

Web Service

Colocar parámetros

Se crea un WebMethod para llamar el procedimiento almacenado borrar_area

```
<WebMethod()> _
Public Function borrar_area(ByVal idarea As Integer) As String
    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        Dim cmd As SqlCommand = New SqlCommand("borrar_area",
conn)

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@idarea",
Data.SqlDbType.VarChar, 50)).Value = idarea.ToString
        conn.Open()
        Dim i As Integer
        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString
    Catch ex As Exception

    End Try
End Function
```

4.3.1.9 Borrar Autor

Stored Procedure

Procedimiento almacenado donde se utiliza delete para eliminar filas

```
Create procedure [dbo].[borrar_autor]
(
    @idautor varchar(50) 'parametro'
)
```

```

as
delete from autor 'se borra la fila'

where idautor = @idautor 'donde el valor que esta en idarea es =
@idarea'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado borrar_area

Colocar parámetros

```

<WebMethod()> _
Public Function borrar_autor(ByVal idautor As Integer) As String
Try
Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
Dim cmd As SqlCommand = New SqlCommand("borrar_autor",
conn)

cmd.CommandType = Data.CommandType.StoredProcedure
cmd.Parameters.Add(New SqlParameter("@idautor",
Data.SqlDbType.VarChar, 50)).Value = idautor.ToString
conn.Open()
Dim i As Integer
i = cmd.ExecuteNonQuery
conn.Close()
Return i.ToString
Catch ex As Exception

End Try
End Function

```

4.3.1.10 Borrar Editorial

Stored Procedure

Procedimiento almacenado donde se utiliza delete para eliminar filas

```

Create procedure [dbo].[borrar_editorial]
(
@ideditorial varchar(50) 'parametro'
)
as

delete from editorial 'borra la fila'
where ideditorial = @ideditorial 'donde el valor que esta en
ideditorial es = @ideditorial'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado borrar_editorial

Colocar parámetros

```
<WebMethod()> _
Public Function borrar_editorial(ByVal ideditorial As String) As
String
    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        Dim cmd As SqlCommand = New SqlCommand("borrar_editorial",
conn)

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@ideditorial",
Data.SqlDbType.VarChar, 50)).Value = ideditorial.ToString
        conn.Open()
        Dim i As Integer
        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString
    Catch ex As Exception

    End Try
End Function
```

4.3.1.11 Borrar Libro

Stored Procedure

Procedimiento almacenado donde se utiliza delete para eliminar filas

```
Create procedure [dbo].[borrar_libro]
(
    @idLibro varchar(50) 'parametro'
)
as
delete from libro 'borra la fila'
where idlibro = @idlibro 'donde el valor que esta en idlibro es =
@idlibro'
```

Visual. Net

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado borrarlibro

Colocar parámetros

```
<WebMethod()> _
Public Function borrar_libro(ByVal idlibro As Integer) As String
    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        Dim cmd As SqlCommand = New SqlCommand("borrar_libro",
conn)

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@idlibro",
Data.SqlDbType.VarChar, 50)).Value = idlibro.ToString
        conn.Open()
        Dim i As Integer
```

```

        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString
    Catch ex As Exception

    End Try
End Function

```

4.3.1.12 Borrar TipoLibro

Stored Procedure

Procedimiento almacenado donde se utiliza delete para eliminar filas

```

Create procedure [dbo].[borrar_tipolibro]
(
    @idtipolibro varchar(50) 'parametro'
)
as
delete from tipolibro 'borra la fila'
where idtipolibro = @idtipolibro 'donde el valor que esta en
idtipolibro es = @idtipolibro'

```

Visual. Net

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado borrar_tipolibro

Colocar parámetros

```

<WebMethod()> _
    Public Function borrar_tipolibro(ByVal idtipolibro As Integer) As
String
    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        Dim cmd As SqlCommand = New SqlCommand("borrar_tipolibro",
conn)

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@idtipolibro",
Data.SqlDbType.VarChar, 50)).Value = idtipolibro.ToString
        conn.Open()
        Dim i As Integer
        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString
    Catch ex As Exception

```

```
End Try
End Function
```

4.3.1.13 Borrar Usuario

Stored Procedure

Procedimiento almacenado donde se utiliza delete para eliminar filas

```
Create procedure [dbo].[borrar_usuario]
(
    @idusuario varchar(50) 'parametro'
)
as
delete from usuario 'borra la fila'
where idusuario = @idusuario 'donde el valor que esta en idusuario es
= @idusuario'
```

Se crea un WebMethod para llamar el procedimiento almacenado borrar_usuario

Visual. Net Web Service

Colocar parámetros

```
<WebMethod()> _
Public Function borrar_usuario(ByVal idusuario As Integer) As
String
    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        Dim cmd As SqlCommand = New SqlCommand("borrar_usuario",
conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@idusuario",
Data.SqlDbType.VarChar, 50)).Value = idusuario.ToString
        conn.Open()
        Dim i As Integer
        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString
    Catch ex As Exception

    End Try
End Function
```

4.3.1.14 Insertar Area

Stored Procedure

Procedimiento almacenado donde se utiliza insert para la insercion de filas

```

Create PROCEDURE [dbo].[insert_area]
(
    @descripcion varchar(50) 'parametro'
)
as
insert into area (descripcion) 'se inserta fila'

values (@descripcion)

'valor que introducirá en @descripcion en el orden de los parametros'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado insert_area

Colocar parámetros

```

<WebMethod(> _
    Public Function insertararea(ByVal descripcion As String) As
String

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog = biblioteca;user id
=libros;pwd=yara;")
        conn.Open()
        Dim i As Integer
        Dim cmd As SqlCommand = New SqlCommand("insert_area",
conn)

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add("@descripcion", Data.SqlDbType.VarChar,
50).Value = descripcion

        i = cmd.ExecuteNonQuery
        conn.Close()

    Catch ex As Exception
        Return ex.Message
    End Try

End Function

```

4.3.1.14 Insertar Autor

Procedimiento almacenado donde se utiliza insert para la insercion de filas

Stored Procedure

```

Create procedure [dbo].[insert_autor]

(
    @nombre varchar(50),    'parametro'
    @apellido varchar(50)  'parametro'
)
as
insert into autor (nombre, apellido) 'se inserta fila'
values (@nombre, @apellido) 'valores que se introducirán en @nombre,
@apellido en el orden de los parametros'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado insert_autor

Colocar parámetros

```

    <WebMethod()> _
Public Function insertar_autor(ByVal nombre As String, ByVal apellido
As String) As String

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog = biblioteca;user id
=libros;pwd=yara;")
        conn.Open()
        Dim i As Integer
        Dim cmd As SqlCommand = New SqlCommand("insert_autor",
conn)

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add("@nombre", Data.SqlDbType.VarChar,
50).Value = nombre
        cmd.Parameters.Add("@apellido", Data.SqlDbType.VarChar,
50).Value = apellido
        i = cmd.ExecuteNonQuery
        conn.Close()

    Catch ex As Exception
        Return ex.Message
    End Try

End Function

```

4.3.1.15 Insertar Editorial

Procedimiento almacenado donde se utiliza insert para la insercion de filas

Stored Procedure

```

Create procedure [dbo].[insert_editorial]

```

```

(
    @editorial varchar(50) 'parametro'
)
as
insert into editorial (editorial) 'se inserta fila'

values (@editorial) 'valor que se introducirá en @editorial en el
orden de los parametros'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado insert_editorial

Colocar parámetros

```

    <WebMethod()> _
Public Function insertar_editorial(ByVal editorial As String) As
String

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog = biblioteca;user id
=libros;pwd=yara;")
        conn.Open()
        Dim i As Integer
        Dim cmd As SqlCommand = New SqlCommand("insert_editorial",
conn)

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add("@editorial", Data.SqlDbType.VarChar,
50).Value = editorial
        i = cmd.ExecuteNonQuery
        conn.Close()

    Catch ex As Exception
        Return ex.Message
    End Try

End Function

```

4.3.1.16 Insertar Libro

Procedimiento almacenado donde se utiliza insert para la insercion de filas

Stored Procedure

```

Create procedure [dbo].[insertar_libro]
(

```

```

        @idlibro varchar(50),           'parametro'
        @titulo varchar(50),          'parametro'
        @descripcion varchar(50),     'parametro'
        @anopublicacion varchar(50), 'parametro'
        @cantidad varchar(20),       'parametro'
        @editorial int,               'parametro'
        @area int,                    'parametro'
        @autor int,                   'parametro'
        @tipolibro int                 'parametro'
    )
as
insert into libro (idlibro, titulo, descripcion, anopublicacion,
cantidad, idEditorial, IdArea, IdAutor, idTipoLibro) 'se inserta fila'

values (@idlibro, @titulo,@descripcion, @anopublicacion, @cantidad,
@editorial, @area, @autor, @tipolibro) 'valores que se introducirán en
@idlibro, @titulo, @descripcion, @anopublicacion,@cantidad,
@editorial, @area, @autor, @tipolibro en el orden de los parametros'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado insert_libro

Colocar parámetros

```

    <WebMethod()> _
    Public Function
    insertar_libro(ByVal idlibro As String, ByVal titulo As String, ByVal
descripcion As String, ByVal anopublicacion As String, ByVal cantidad
As String, ByVal autor As Integer, ByVal area As Integer, ByVal
tipolibro As Integer, ByVal editorial As Integer) As String

        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog = biblioteca;user id
=libros;pwd=yara;")
            conn.Open()
            Dim i As Integer
            Dim cmd As SqlCommand = New SqlCommand("insertar_libro",
conn)

            cmd.CommandType = Data.CommandType.StoredProcedure
            cmd.Parameters.Add("@idlibro", Data.SqlDbType.VarChar,
50).Value = idlibro
            cmd.Parameters.Add("@titulo", Data.SqlDbType.VarChar,
50).Value = titulo
            cmd.Parameters.Add("@descripcion", Data.SqlDbType.VarChar,
50).Value = descripcion
            cmd.Parameters.Add("@anopublicacion",
Data.SqlDbType.VarChar, 50).Value = anopublicacion
            cmd.Parameters.Add("@cantidad", Data.SqlDbType.VarChar,
20).Value = cantidad
            cmd.Parameters.Add("@editorial", Data.SqlDbType.Int).Value
= editorial

```

```

        cmd.Parameters.Add("@area", Data.SqlDbType.Int).Value =
area
        cmd.Parameters.Add("@autor", Data.SqlDbType.Int).Value =
autor
        cmd.Parameters.Add("@tipolibro", Data.SqlDbType.Int).Value =
tipolibro

        i = cmd.ExecuteNonQuery
        conn.Close()

    Catch ex As Exception
        Return ex.Message
    End Try

End Function

```

4.3.1.17 Insertar Tipo Libro

Procedimiento almacenado donde se utiliza insert para la insercion de filas

Stored Procedure

```

create procedure [dbo].[insert_tipolibro]
(
    @tipolibro varchar(50) 'parametro'
)
as

insert into tipolibro (tipolibro) 'se inserta fila'
values (@tipolibro) 'valor que se introducirá en @tipolibro en el
orden de los parametros'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado insert_tipolibro

Colocar parámetros

```

<WebMethod()> _
Public Function insertar_tipolibro(ByVal tipolibro As String) As
String

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog = biblioteca;user id
=libros;pwd=yara;")
        conn.Open()
        Dim i As Integer

```

```

        Dim cmd As SqlCommand = New SqlCommand("insert_tipolibro",
conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add("@tipolibro", Data.SqlDbType.VarChar,
50).Value = tipolibro

        i = cmd.ExecuteNonQuery
        conn.Close()

    Catch ex As Exception
        Return ex.Message
    End Try

End Function

```

4.3.1.18 Insertar Tipo Usuario

Procedimiento almacenado donde se utiliza insert para la insercion de filas

Stored Procedure

```

Create procedure [dbo].[insert_tipousuario]
(
    @descripcion varchar(50) 'parametro'
)
as
insert into tipousuario (descripcion) 'se inserta fila'
values (@descripcion) 'valor que se introducirá en @descripcion en el
orden de los parametros'

```

Visual. Net

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado insert_tipousuario

Colocar parámetros

```

    <WebMethod()> _
    Public Function insertar_tipousuario(ByVal descripcion As String)
As String

        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog = biblioteca;user id
=libros;pwd=yara;")
            conn.Open()
            Dim i As Integer
            Dim cmd As SqlCommand = New
SqlCommand("insert_tipousuario", conn)
            cmd.CommandType = Data.CommandType.StoredProcedure
            cmd.Parameters.Add("@descripcion", Data.SqlDbType.VarChar,
50).Value = descripcion

```

```

        i = cmd.ExecuteNonQuery
        conn.Close()

    Catch ex As Exception
        Return ex.Message
    End Try

End Function

```

4.3.1.19 Insertar Usuario

Procedimiento almacenado donde se utiliza insert para la insercion de filas

Stored Procedure

```

Create procedure [dbo].[insertar_usuario]
(
    @IdUsuario varchar(15), 'parametro'
    @nombre varchar(50), 'parametro'
    @apellido varchar(50), 'parametro'
    @direccion varchar(50), 'parametro'
    @telefono int, 'parametro'
    @username varchar (50) 'parametro'
)
as
insert into usuario (idusuario, nombre, apellido, direccion, telefono,
username) 'se inserta fila'
values (@idusuario, @nombre, @apellido, @direccion, @telefono,
@username) 'valores que se introducirán en @idusuario, @nombre,
@direccion, @telefono, @username en el orden de los parametros'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado insert_usuario

Colocar parámetros

```

<WebMethod(> _
Public Function insertar_usuario(ByVal idusuario As String, ByVal
nombre As String, ByVal apellido As String, ByVal direccion As String,
ByVal telefono As Integer, byval username as String) As String

    Try

```

```

        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog = biblioteca;user id
=libros;pwd=yara;")
        conn.Open()
        Dim i As Integer
        Dim cmd As SqlCommand = New SqlCommand("insertar_usuario",
conn)

        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add("@idusuario", Data.SqlDbType.VarChar,
15).Value = idusuario
        cmd.Parameters.Add("@nombre", Data.SqlDbType.VarChar,
50).Value = nombre
        cmd.Parameters.Add("@apellido", Data.SqlDbType.VarChar,
50).Value = apellido
        cmd.Parameters.Add("@direccion", Data.SqlDbType.VarChar,
50).Value = direccion
        cmd.Parameters.Add("@telefono", Data.SqlDbType.Int,
50).Value = telefono
        cmd.Parameters.Add("@username", Data.SqlDbType.VarChar,
50).Value = username

        i = cmd.ExecuteNonQuery
        conn.Close()

    Catch ex As Exception
        Return ex.Message
    End Try

End Function

```

4.3.1.20 Seleccionar AdminCorreo

Procedimiento almacenado simple, sin parametros que devuelve un conjunto de filas que cumplen siempre la misma condicion.

Stored Procedure

```

Create procedure [dbo].[ sp_admin]
as
select * from admincorreo 'selección de filas de la tabla admincorreo'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado sp_admin

Colocar parámetros

```

<WebMethod()> _
    Public Function consultar_Admin() As Data.DataSet

        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog = biblioteca;user id
=libros;pwd=yara;")
            conn.Open()
            Dim cmd As SqlCommand = New SqlCommand("sp_admin", conn)

```

```

        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try

End Function

```

4.3.1.21 Seleccionar Area

Procedimiento almacenado simple, sin parametros que devuelve un conjunto de filas que cumple siempre la misma condicion.

Stored Procedure

```

Create procedure [dbo].[seleccionar_areaall]
as
select * from area 'selección de filas de la tabla area'

```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado consultar_arreall

Colocar parámetros

```

<WebMethod()> _
Public Function consultar_areaall() As Data.DataSet

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        conn.Open()
        Dim cmd As SqlCommand = New
SqlCommand("seleccionar_areaall", conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try

End Function

```

4.3.1.22 Seleccionar Autor

Procedimiento almacenado simple, sin parametros que devuelve un conjunto de filas que cumple siempre la misma condicion.

Stored Procedure

```
Create procedure [dbo].[seleccionar_autor]
as
select idautor, nombre + ' ' + apellido as nombre from autor 'selección
de filas de la tabla autor'
```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado consultar_autor

Colocar parámetros

```
<WebMethod()> _
Public Function consultar_autor() As Data.DataSet

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        conn.Open()
        Dim cmd As SqlCommand = New
SqlCommand("seleccionar_autor", conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try
End Function
```

4.3.1.23 Seleccionar Editorial

Procedimiento almacenado simple, sin parametros que devuelve un conjunto de filas que cumple siempre la misma condicion.

Stored Procedure

```
Create procedure [dbo].[seleccionar_editorial]
as
select * from editorial 'selección de filas de la tabla admincorreo'
```

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado sp_admin

Colocar parámetros

```
<WebMethod()> _
    Public Function consultar_editorial() As Data.DataSet

        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            conn.Open()
            Dim cmd As SqlCommand = New
SqlCommand("seleccionar_editorial", conn)
            cmd.CommandType = Data.CommandType.StoredProcedure
            Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
            Dim ds As New Data.DataSet
            adap.Fill(ds)
            Return ds
        Catch ex As Exception

        End Try

    End Function
```

4.3.1.24 Seleccionar Libro

Procedimiento almacenado select con inner join que permite hacer consultas multitabla

Stored Procedure

```
Create procedure [dbo].[seleccionar_libro]

as
SELECT      dbo.LIBRO.IdLibro, dbo.LIBRO.Titulo, dbo.LIBRO.Descripcion,
dbo.LIBRO.AnoPublicacion, dbo.LIBRO.cantidad, dbo.Editorial.Editorial,
            dbo.AREA.Descripcion AS Expr1, dbo.AUTOR.Nombre
+ ' ' + dbo.AUTOR.Apellido as Nombre, dbo.TipoLibro.TipoLibro
FROM        dbo.AREA INNER JOIN
            dbo.LIBRO ON dbo.AREA.IdArea = dbo.LIBRO.IdArea
INNER JOIN
            dbo.AUTOR ON dbo.LIBRO.IdAutor =
dbo.AUTOR.IdAutor INNER JOIN
            dbo.Editorial ON dbo.LIBRO.idEditorial =
dbo.Editorial.idEditorial INNER JOIN
            dbo.TipoLibro ON dbo.LIBRO.idTipoLibro =
dbo.TipoLibro.idTipoLibro
```

'esta tabla tiene muchos campos que estan relacionados mediante un id con otras tablas que contienen los valores reales y necesitamos que los id se conviertan en dichos valores reales y para esto utilizaremos joins. En este select seleccionamos los campos de los queremos sacar informacion y le damos un nombre de alias con As. En From decimos de qué tablas sacamos la información del select, aclaramos las condiciones que se tienen que cumplir para que los valores sean correctos. En esta tabla los id deben coincidir con los id de las tablas relacionadas'

Visual. Net Web Service

Se crea un WebMethod para llamar el procedimiento almacenado consultar_libro

Colocar parámetros

```
<WebMethod()> _
    Public Function consultar_libro() As Data.DataSet

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        conn.Open()
        Dim cmd As SqlCommand = New
SqlCommand("seleccionar_libro", conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try

End Function
```

4.3.1.25 Seleccionar Préstamo

Procedimiento almacenado select con inner join que permite hacer consultas multitabla

Stored Procedure

```
Create procedure [dbo].[seleccionar_prestamo]
(
```

```

        @FechaPrestamo datetime,
        @FechaDevolucion datetime
    )
as
SELECT      dbo.PRESTAMO.IdPrestamo, dbo.USUARIO.Nombre,
dbo.USUARIO.Apellido, dbo.USUARIO.Telefono, dbo.PRESTAMO.Estado,
            dbo.PRESTAMO.FechaPrestamo,
dbo.PRESTAMO.FechaDevolucion, dbo.PRESTAMO.IdUsuario
FROM        dbo.PRESTAMO INNER JOIN
            dbo.USUARIO ON dbo.PRESTAMO.IdUsuario =
dbo.USUARIO.IdUsuario
WHERE      dbo.PRESTAMO.FechaPrestamo between @FechaPrestamo and
@FechaDevolucion

```

'esta tabla tiene algunos campos que estan relacionados mediante un id con otras tablas que contienen los valores reales y necesitamos que los id se conviertan en dichos valores reales y para esto utilizaremos joins. En este select seleccionamos los campos de los queremos sacar informacion. En From decimos de qué tablas sacamos la información del select, aclaramos las condiciones que se tienen que cumplir para que los valores sean correctos. En esta tabla los id deben coincidir con los id de las tablas relacionadas'

Visual. Net

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado consultar_prestamo

Colocar parámetros

```

<WebMethod(> _
    Public Function consultar_prestamo(ByVal FechaPrestamo As Date,
ByVal Fechadevolucion As Date) As Data.DataSet

        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            conn.Open()
            Dim cmd As SqlCommand = New
SqlCommand("seleccionar_prestamo", conn)
            cmd.Parameters.Add(New SqlParameter("@FechaPrestamo",
Data.SqlDbType.DateTime)).Value = FechaPrestamo.ToString
            cmd.Parameters.Add(New SqlParameter("@Fechadevolucion",
Data.SqlDbType.DateTime)).Value = Fechadevolucion.ToString
            cmd.CommandType = Data.CommandType.StoredProcedure
            Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
            Dim ds As New Data.DataSet
            adap.Fill(ds)
            Return ds
        Catch ex As Exception

        End Try

    End Function

```

4.3.1.26 Seleccionar Tipo Libro

Procedimiento almacenado simple, sin parametros que devuelve un conjunto de filas que cumple siempre la misma condicion.

Stored Procedure

```
Create procedure [dbo].[seleccionar_tipolibro]
as
select * from tipolibro 'selección de filas de la tabla tipolibro'
```

Visual. Net

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado seleccionar_tipolibro

Colocar parámetros

```
<WebMethod(> _
Public Function consultar_tipolibro() As Data.DataSet

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        conn.Open()
        Dim cmd As SqlCommand = New
SqlCommand("seleccionar_tipolibro", conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try

End Function
```

4.3.1.27 Seleccionar Tipo Usuario

Procedimiento almacenado simple, sin parametros que devuelve un conjunto de filas que cumple siempre la misma condicion.

Stored Procedure

```

Create procedure [dbo].[seleccionar_tipousuario]
as
select * from tipousuario 'selección de filas de la tabla tipousuario'

```

Visual. Net

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado seleccionar_tipolibro

Colocar parámetros

```

<WebMethod(> _
Public Function consultar_tipousuario() As Data.DataSet

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        conn.Open()
        Dim cmd As SqlCommand = New
SqlCommand("seleccionar_tipousuario", conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try

End Function

```

4.3.1.28 Seleccionar Usuario

Procedimiento almacenado simple, sin parametros que devuelve un conjunto de filas que cumple siempre la misma condicion.

Stored Procedure

```

Create procedure [dbo].[seleccionar_usuario]
as
select * from usuario 'selección de filas de la tabla tipolibro'

```

Visual. Net

Web Service

Se crea un WebMethod para llamar el procedimiento almacenado seleccionar_usuario

Colocar parámetros

```

<WebMethod()> _
Public Function consultar_usuario() As Data.DataSet

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        conn.Open()
        Dim cmd As SqlCommand = New
SqlCommand("seleccionar_usuario", conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try
End Function

```

4.3.2 Gestión Prestamo de Libros

4.3.2.1 Envío Correo Solicitud Préstamo

```

Imports Datos.BiblioServices.Service
Public Class Correo
    Public ws As BiblioServices.Service = New BiblioServices.Service
    Public Sub EnvioCorreo(ByVal entra As String, ByVal salida As
String, ByVal contra As String, ByVal server As String, ByVal libro As
String, ByVal Cuerpo As String, ByVal puerto As Integer, ByVal ssl As
Boolean)

        Dim correo As New System.Net.Mail.MailMessage 'creamos un
nuevo mensaje de correo

        correo.From = New System.Net.Mail.MailAddress(salida) 'De
correo.To.Add(entra) 'Para
        correo.Subject = "Prestamos del libro " + libro 'Asunto
        correo.Body = Cuerpo 'Cuerpo del correo
        correo.IsBodyHtml = False 'Mostrar como HTML
        correo.Priority = System.Net.Mail.MailPriority.Normal
        'Prioridad de el correo

        'indicamos cual servidor utilizaremos
        'aquí usaremos gmail y su puerto SMTP
        Dim smtp As New System.Net.Mail.SmtpClient
        smtp.Host = server
        smtp.Port = puerto
        smtp.Credentials = New System.Net.NetworkCredential(salida,
contra)
        smtp.EnableSsl = ssl

    Try

```

```

        smtp.Send(correo)
    Catch ex As Exception
    End Try

End Sub

'Trae correo
Public Sub TraerCorreos(ByVal libro As String, ByVal cuerpo As
String)
    Dim Entra As String
    Dim Sale As String
    Dim Contra As String
    Dim Server As String
    Dim puerto As Integer
    Dim ssl As Boolean
    Entra =
ws.consultar_Admin().Tables(0).Rows(0).Item(0).ToString()
    Sale =
ws.consultar_Admin().Tables(0).Rows(0).Item(1).ToString()
    Contra =
ws.consultar_Admin().Tables(0).Rows(0).Item(2).ToString()
    Server =
ws.consultar_Admin().Tables(0).Rows(0).Item(3).ToString()
    puerto =
Convert.ToInt32(ws.consultar_Admin().Tables(0).Rows(0).Item(4).ToStrin
g())
    ssl =
Convert.ToBoolean(ws.consultar_Admin().Tables(0).Rows(0).Item(5).ToStr
ing())

    EnvioCorreo(Entra, Sale, Contra, Server, libro, cuerpo,
puerto, ssl)
End Sub
End Class

```

4.3.2.2 Actualizar cantidad de libro Stored Procedure

Procedimiento almacenado diseñado para que cuando se saque un libro se reste en la cantidad

```

Create PROCEDURE [dbo].[sp_aclibro]
(
    @busqueda AS int,
    @cant AS int
)
AS
UPDATE LIBRO
SET
    cantidad = @cant

WHERE IdLibro = @busqueda

```

Visual. Net Web Service

Colocar parámetros

Se crea este webmethod para llamar el sp_aclibro

```
<WebMethod()> _
    Public Function actulizar_Librop(ByVal busqueda As Integer, ByVal
cantidad As Integer) As String
        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            Dim cmd As SqlCommand = New SqlCommand("sp_aclibro", conn)
            cmd.CommandType = Data.CommandType.StoredProcedure
            cmd.Parameters.Add(New SqlParameter("@busqueda",
Data.SqlDbType.VarChar, 50)).Value = busqueda
            cmd.Parameters.Add(New SqlParameter("@cant",
Data.SqlDbType.VarChar, 50)).Value = cantidad
            conn.Open()
            Dim i As Integer
            i = cmd.ExecuteNonQuery
            conn.Close()
            Return i.ToString

        Catch ex As Exception

        End Try

    End Function
```

4.3.2.3 Buscar por Autor Stored Procedure

Procedimiento almacenado diseñado con select y inner join para realizar búsquedas por autor

```
Create PROCEDURE [dbo].[sp_autor]
(
    @busqueda AS varchar(60)
)
AS
SELECT     dbo.LIBRO.IdLibro, dbo.LIBRO.Titulo, dbo.LIBRO.Descripcion,
dbo.LIBRO.AnoPublicacion, dbo.LIBRO.cantidad, dbo.Editorial.Editorial,
dbo.AUTOR.Nombre, dbo.AUTOR.Apellido
FROM       dbo.AUTOR INNER JOIN
          dbo.LIBRO ON dbo.AUTOR.IdAutor = dbo.LIBRO.IdAutor INNER
JOIN
          dbo.Editorial ON dbo.LIBRO.idEditorial =
dbo.Editorial.idEditorial
WHERE      dbo.AUTOR.Nombre LIKE '%' + @busqueda + '%' OR
dbo.AUTOR.Apellido LIKE '%' + @busqueda + '%'
```

'esta tabla tiene algunos campos que estan relacionados mediante un id con otras tablas que contienen los valores reales y necesitamos que los id se conviertan en dichos valores reales y para esto utilizaremos joins. En este select seleccionamos los campos de los queremos sacar informacion. En From decimos de qué tablas sacamos la información del select, aclaramos las condiciones que se tienen que cumplir para que

los valores sean correctos. En esta tabla los id deben coincidir con los id de las tablas relacionadas'

Visual. Net Web Service Colocar parámetros

WebMethod creado para llamar el sp_autor

```
<WebMethod(> _
    Public Function consultar_AutorPres(ByVal busqueda As String) As
Data.DataSet

        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            conn.Open()
            Dim cmd As SqlCommand = New SqlCommand("sp_autor", conn)
            cmd.Parameters.Add(New SqlParameter("@busqueda",
Data.SqlDbType.VarChar, 60)).Value = busqueda.ToString
            cmd.CommandType = Data.CommandType.StoredProcedure
            Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
            Dim ds As New Data.DataSet
            adap.Fill(ds)
            Return ds
        Catch ex As Exception

        End Try

    End Function
```

4.3.2.4 Buscar por Libro Stored Procedure

Procedimiento almacenado select con inner join que permite hacer consultas multitabla

```
Create PROCEDURE [dbo].[sp_libro]
(
    @busqueda AS varchar(60)
)
AS
SELECT     dbo.LIBRO.IdLibro, dbo.LIBRO.Titulo, dbo.LIBRO.Descripcion,
dbo.LIBRO.AnoPublicacion, dbo.LIBRO.cantidad, dbo.Editorial.Editorial,
dbo.AUTOR.Nombre, dbo.AUTOR.Apellido
FROM       dbo.AUTOR INNER JOIN
          dbo.LIBRO ON dbo.AUTOR.IdAutor = dbo.LIBRO.IdAutor INNER
JOIN
          dbo.Editorial ON dbo.LIBRO.idEditorial =
dbo.Editorial.idEditorial
WHERE      dbo.LIBRO.Titulo LIKE '%' + @busqueda + '%'
```

'esta tabla tiene muchos campos que estan relacionados mediante un id con otras tablas que contienen los valores reales y necesitamos que los id se conviertan en dichos valores reales y para esto utilizaremos

joins. En este select seleccionamos los campos de los queremos sacar información. En From decimos de qué tablas sacamos la información del select, aclaramos las condiciones que se tienen que cumplir para que los valores sean correctos. En esta tabla los id deben coincidir con los id de las tablas relacionadas'

Visual. Net Web Service Colocar parámetros

WebMethod diseñado para llamar el sp_libro.

```
<WebMethod()> _
Public Function consultar_LibroPres(ByVal busqueda As String) As
Data.DataSet

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        conn.Open()
        Dim cmd As SqlCommand = New SqlCommand("sp_libro", conn)
        cmd.Parameters.Add(New SqlParameter("@busqueda",
Data.SqlDbType.VarChar, 60)).Value = busqueda.ToString
        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try

End Function
```

4.3.2.5 Buscar por Editorial Stored Procedure

Procedimiento almacenado select con inner join que permite hacer consultas multitabla

```
Create PROCEDURE [dbo].[sp_editorial]
(
    @busqueda AS varchar(60)
)
AS
SELECT     dbo.LIBRO.IdLibro, dbo.LIBRO.Titulo, dbo.LIBRO.Descripcion,
dbo.LIBRO.AnoPublicacion, dbo.LIBRO.cantidad, dbo.Editorial.Editorial,
dbo.AUTOR.Nombre, dbo.AUTOR.Apellido
FROM       dbo.AUTOR INNER JOIN
          dbo.LIBRO ON dbo.AUTOR.IdAutor = dbo.LIBRO.IdAutor INNER
JOIN
          dbo.Editorial ON dbo.LIBRO.idEditorial =
dbo.Editorial.idEditorial
WHERE      dbo.Editorial.Editorial LIKE '%' + @busqueda + '%'
```

'esta tabla tiene muchos campos que estan relacionados mediante un id con otras tablas que contienen los valores reales y necesitamos que los id se conviertan en dichos valores reales y para esto utilizaremos joins. En este select seleccionamos los campos de los queremos sacar informacion. En From decimos de qué tablas sacamos la información del select, aclaramos las condiciones que se tienen que cumplir para que los valores sean correctos. En esta tabla los id deben coincidir con los id de las tablas relacionadas'

Visual. Net Web Service Colocar parámetros

WebMethod diseñado para llamar el sp_editorial

```
<WebMethod()> _
  Public Function consultar_EditorialPres(ByVal busqueda As String) As
Data.DataSet

    Try
      Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
      conn.Open()
      Dim cmd As SqlCommand = New SqlCommand("sp_editorial",
conn)
      cmd.Parameters.Add(New SqlParameter("@busqueda",
Data.SqlDbType.VarChar, 60)).Value = busqueda.ToString
      cmd.CommandType = Data.CommandType.StoredProcedure
      Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
      Dim ds As New Data.DataSet
      adap.Fill(ds)
      Return ds
    Catch ex As Exception

    End Try

  End Function
```

4.3.2.6 Buscar por Libro Stored Procedure

Procedimiento almacenado select hacer consulta de libro

```
Create PROCEDURE [dbo].[sp_libroid]
(
    @busqueda AS int
)
AS
SELECT Titulo,Descripcion,AnoPublicacion,cantidad
FROM LIBRO
WHERE IdLibro = @busqueda
```

Visual. Net

Web Service Colocar parámetros

WebMethod diseñado para llamar el sp_libroid

```
<WebMethod()> _
Public Function consultar_Libroid(ByVal busqueda As Integer) As
Data.DataSet

    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        conn.Open()
        Dim cmd As SqlCommand = New SqlCommand("sp_libroid", conn)
        cmd.Parameters.Add(New SqlParameter("@busqueda",
Data.SqlDbType.Int)).Value = busqueda
        cmd.CommandType = Data.CommandType.StoredProcedure
        Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim ds As New Data.DataSet
        adap.Fill(ds)
        Return ds
    Catch ex As Exception

    End Try

End Function
```

Para la realización del inventario se tiene en cuenta la cantidad de libros prestados a los usuarios y los libros que se encuentran en la biblioteca Se crea un sp_consultaprestamo

4.3.2.7 Libro asignado usuario Stored Procedure

```
Create PROCEDURE sp_librosacargo
(
    @usr AS varchar(60)
)
AS
SELECT      dbo.LIBRO.Titulo, dbo.LIBRO.Descripcion,
dbo.DetallePrestamo.CantidadPrestado,
dbo.DetallePrestamo.EstadoPrestamo
FROM        dbo.DetallePrestamo INNER JOIN
            dbo.LIBRO ON dbo.DetallePrestamo.IdLibro =
            dbo.LIBRO.IdLibro
WHERE      dbo.DetallePrestamo.NombreUsuario = @usr AND
            dbo.DetallePrestamo.EstadoPrestamo = 'En prestamo'
go
```

Visual. Net Web Service Colocar parámetros

WebMethod diseñado para llamar sp_librosacargo

```
<WebMethod()> _
    Public Function consultar_Librosusuario(ByVal nombre As String)
    As Data.DataSet

        Try
            Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
            conn.Open()
            Dim i As Integer
            Dim cmd As SqlCommand = New SqlCommand("sp_librosacargo",
conn)

            cmd.CommandType = Data.CommandType.StoredProcedure
            cmd.Parameters.Add("@usr", Data.SqlDbType.VarChar,
60).Value = nombre.ToString
            i = cmd.ExecuteNonQuery
            Dim adap As SqlDataAdapter = New SqlDataAdapter(cmd)
            Dim ds As New Data.DataSet
            adap.Fill(ds)
            Return ds
            conn.Close()
        Catch ex As Exception

        End Try

    End Function
```

4.3.2.8 Devolucion Libro

Stored Procedure

```
Create PROCEDURE sp_Devolucion
(
    @codlib AS int
)
AS

DECLARE @cant AS int
SET @cant = (SELECT cantidad FROM LIBRO WHERE IdLibro = @codlib)

UPDATE DetallePrestamo
SET
    EstadoPrestamo = 'Devuelto'
WHERE IdLibro = @codlib AND EstadoPrestamo = 'En prestamo'

UPDATE LIBRO
SET
    cantidad = @cant + 1
WHERE IdLibro = @codlib
go
```

Visual. Net Web Service Colocar parámetros

WebMethod diseñado para llamar sp_devolucion

```
<WebMethod()> _
    Public Function actualizar_Devolucion(ByVal Codlib As Integer) As
String
    Try
        Dim conn As SqlConnection = New SqlConnection("data
source=COLOSSUS;initial catalog=biblioteca;user id=libros;pwd=yara;")
        Dim cmd As SqlCommand = New SqlCommand("sp_Devolucion",
conn)
        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@codlib",
Data.SqlDbType.Int)).Value = Codlib
        conn.Open()
        Dim i As Integer
        i = cmd.ExecuteNonQuery
        conn.Close()
        Return i.ToString

    Catch ex As Exception

    End Try

End Function
Try
```

4.4 Cuarta Fase (Pruebas)

Pruebas Caja Negra

4.1.1 Registrarse

Datos personales usuario

Si cualquiera de los campos no esta lleno muestra con asterisco y mensaje la necesidad de ingresar datos

Si el correo no tiene el formato determinado se muestra con asterisco el campo y mensaje de ingresar correo adecuado

Si la dirección no tiene el formato determinado se muestra con asterisco el campo y mensaje de ingresar dirección adecuada

Si el teléfono no tiene el formato determinado se muestra con asterisco el campo y mensaje de ingresar teléfono adecuado

Si todos los campos están llenos adecuadamente entonces

Datos para crear cuenta de usuario

Si cualquiera de los campos no esta lleno muestra con asterisco y mensaje la necesidad de ingresar datos

Si la contraseña no tiene el formato determinado se muestra con asterisco el campo y mensaje de ingresar contraseña adecuada

Si el correo no tiene el formato determinado se muestra con asterisco el campo y mensaje de ingresar correo adecuado

Si todos los campos están llenos adecuadamente entonces
Usuario creado

Para esto utilizamos el control **CreateUserWizard** admite la validación requerida y de comparación

4.1.2 **Solicitud Préstamo de Libro**

4.1.2.1 **Búsqueda por Autor**

- Seleccionar una opción de búsqueda
Si selecciona autor
Escribir nombre y apellido
Si nombre y apellido no coincide o no se encuentra en la tabla autor mostrar mensaje explicando situación e iniciar búsqueda.
- Seleccionar una opción de búsqueda
Escribir nombre y apellido
Lista de autores relacionados con el No. topográfico, título, descripción, año de publicación, cantidad, área, editorial
Seleccionar libro deseado
Si el libro no se encuentra disponible mensaje de no se encuentra entonces
Iniciar búsqueda
- Seleccionar una opción de búsqueda
Escribir nombre y apellido
Lista de autores relacionados con el No. topográfico, título, descripción, año de publicación, cantidad, área, editorial
Seleccionar libro deseado
Si el libro no se encuentra disponible mensaje de no se encuentra entonces
Iniciar búsqueda
- Seleccionar una opción de búsqueda
Escribir nombre y apellido
Lista de autores relacionados con el No. topográfico, título, descripción, año de publicación, cantidad, área, editorial
Seleccionar libro deseado
Si el usuario ha excedido la cantidad de libro de préstamo que debe tener mensaje donde se le diga al usuario que ha excedido la cantidad de préstamo
- Seleccionar una opción de búsqueda
Escribir nombre y apellido
Lista de autores relacionados con el No. topográfico, título, descripción, año de publicación, cantidad, área, editorial
Seleccionar libro deseado

Confirmar solicitud de préstamo

4.1.2.2 **Búsqueda por libro**

- Seleccionar una opción de búsqueda
Si selecciona libro
Escribir título
Si título no coincide o no se encuentra en la tabla autor mostrar mensaje explicando situación e iniciar búsqueda.
- Seleccionar una opción de búsqueda
Si selecciona libro
Escribir título
Lista de títulos relacionados con el No. topográfico, autor, descripción, año de publicación, cantidad, área, editorial
Seleccionar libro deseado
Si el libro no se encuentra disponible mensaje de no se encuentra entonces
Iniciar búsqueda
- Seleccionar una opción de búsqueda
Si selecciona libro
Escribir título
Lista de títulos relacionados con el No. topográfico, autor, descripción, año de publicación, cantidad, área, editorial
Seleccionar libro deseado
Si el libro no se encuentra disponible mensaje de no se encuentra entonces
Iniciar búsqueda
- Seleccionar una opción de búsqueda
Si selecciona libro
Escribir título
Lista de títulos relacionados con el No. topográfico, autor, descripción, año de publicación, cantidad, área, editorial
Seleccionar libro deseado
Si el usuario ha excedido la cantidad de libro de préstamo que debe tener mensaje donde se le diga al usuario que ha excedido la cantidad de préstamo
- Seleccionar una opción de búsqueda
Escribir nombre y apellido
Lista de autores relacionados con el No. topográfico, título, descripción, año de publicación, cantidad, área, editorial
Seleccionar libro deseado
Confirmar solicitud de préstamo

4.1.2.2 **Búsqueda por Editorial**

- Seleccionar una opción de búsqueda
Si selecciona editorial
Escribir editorial
Si editorial no coincide o no se encuentra en la tabla autor mostrar mensaje explicando situación e iniciar búsqueda.
- Seleccionar una opción de búsqueda
Si selecciona editorial
Escribir editorial
Lista de editoriales relacionadas con el No. topográfico, autor, descripción, año de publicación, cantidad, área, título
Seleccionar libro deseado
Si el libro no se encuentra disponible mensaje de no se encuentra entonces Iniciar búsqueda
- Seleccionar una opción de búsqueda
Escribir editorial
Lista de editoriales relacionadas con el No. topográfico, autor, descripción, año de publicación, cantidad, área, título
Seleccionar libro deseado
Si el libro no se encuentra disponible mensaje de no se encuentra entonces Iniciar búsqueda
- Seleccionar una opción de búsqueda
Escribir editorial
Lista de editoriales relacionadas con el No. topográfico, autor, descripción, año de publicación, cantidad, área, título
Seleccionar libro deseado
Si el usuario ha excedido la cantidad de libro de préstamo que debe tener mensaje donde se le diga al usuario que ha excedido la cantidad de préstamo
- Seleccionar una opción de búsqueda
Escribir editorial
Lista de editoriales relacionadas con el No. topográfico, autor, descripción, año de publicación, cantidad, área, título
Seleccionar libro deseado
Confirmar solicitud de préstamo

4.1.3 **Inserción de Libro**

Insertar datos de libro

Si los campos no están llenos mostrar mensaje de llenar todos los campos

Si el N^o topográfico coincide con otro almacenado mostrar mensaje para verificar el N^o topográfico

Si autor no existe en la lista desplegable ir a insertar nuevo autor

Si editorial no existe el la lista desplegable ir a insertar nuevo autor
Si tipo de libro no existe el la lista desplegable ir a insertar nuevo autor
Si área no existe el la lista desplegable ir a insertar nuevo autor.
Si todos los campos están llenos y verificados
Insertar libro

4.1.3 Inserción de Área

- Escribir área
Si existe el nombre de la área mostrar mensaje ya existe área
Iniciar inserción de área
- Escribir área
Insertar de área

4.1.4 Inserción de Autor

- Escribir datos autor
Si existe el nombre de la autor mostrar mensaje ya existe área
Iniciar inserción de área
- Escribir datos autor
Si no todos los campos están llenos mensaje llenar todos los campos
Iniciar inserción de área
- Escribir datos autor
Insertar de autor

4.1.4 Inserción de Editorial

- Escribir datos editorial
Si existe el nombre de la editorial mostrar mensaje ya existe área
Iniciar inserción de editorial
- Escribir datos editorial
Insertar de editorial

4.1.4 Inserción de Tipo de Libro

- Escribir datos tipo libro
Si existe el nombre de la tipo libro mostrar mensaje ya existe área
Iniciar inserción de área
- Escribir datos tipo de libro
Insertar de tipo de libro

4.5 Retroalimentación

Al realizar la segunda iteración se encontró en la gestión de préstamo que se necesitaba enviar un mensaje al bibliotecario con la solicitud de los libros por

parte de los usuarios la solución fue enviar e-mails desde visual.net 2005 por medio de un servidor SMTP. Además se creó una tabla AdminCorreo en donde se guardarán datos de las cuentas que envía y recibe el mensaje contraseña puerto del correo y seguridad de autenticidad se crea esta tabla con el fin de si en un futuro se cambian estos datos se hagan en la base de datos sin afectar el código en .visual.net

BIBLIOGRAFÍA

ACIMED. SciELO: una metodología para la publicación electrónica, 2001. vol.9 supl.4, 9-22 p. ISSN 1024-9435.

BOB BRANCHEK (Editor), et al. Edición Especial Microsoft SQL Server 6.5, Prentice Hall, Madrid, España. 1997. 26p

DOMINGO BUONOCORE Diccionario de Bibliotecología, Marymar Ediciones, Segunda Edición Aumentada, 1976. 354p

El modelo entidad-relación [en línea]
<http://www3.uji.es/~mmarques/f47/apun/node83.html> [consultada febrero 17 2009]

ELMARSÍ R Y NAVATHE S. Fundamentos de Sistemas de Bases de Datos tercera edición. Editorial Pearson. España. 2005 962p.

FERRER A., PSET F., MORENO M. Y LLORET N. Guía Metodológica para la Implementación de una Biblioteca Digital Universitaria. Edición Trea S. L. España. 2005. 202p.

GARCIA L. Y GARCIA E. Automatización de Bibliotecas. Editorial Arco/Libros S.L. Madrid, España. 1999. 285p.

HAWRYSZKIEWYCZ, I.T, Análisis y Diseño de Base de Datos, Megabyte Noriega Editores, Mexico. 1994. 21 p.

Ingeniería de Software Clase 3 [en línea]
http://pisis.unalmed.edu.co/cursos/material/3004582/1/clase3_IS_0702-v1.ppt#256,1, Ingeniería de Software Clase 3 [consultada Mayo 30 2009]

WILLIAM SAFFADY, Andrés Magaña García, Informática Documental para Bibliotecas, Traducido por Andrés Magaña García, Publicado por Ediciones Díaz de Santos, 1986. 309 p. ISBN 8486251478, 9788486251475

JOSÉ YEPES(Editor), Diccionario Enciclopédico de Ciencias de la Documentación, Editorial Síntesis, S.A. 439-440p.

PONJUÁN D.G. Biblioteca digital. Un nuevo paso en la evolución de las arquitecturas de información. Revista *Ciencias de la Información* 2002. Vol. 33, No. 1.

ANEXOS A
ENCUESTA

Entrevista

- ¿Cuál es su cargo en la institución?

Yo me encargo de la biblioteca de organizar los libros y de facilitárselos a los estudiantes

- ¿Realizo algún estudio para hacerse cargo de la biblioteca?

La verdad solo hice un curso de archivo pero y una inducción en el manejo de la biblioteca

- ¿Sabe usted que es la bibliotecología?

Pues creo que es el estudio de las bibliotecas pero no estoy seguro

- ¿Cómo es el manejo del préstamo de libros?

Pues aquí se maneja de la siguiente forma: Los estudiantes que necesitan libros los buscan en los ficheros que hay y llenan una ficha con el nombre del libro y el nombre de ellos yo les busco el libro y se los paso, ellos me entregan su documento de identidad el cual yo conservo con la ficha de préstamo en la ficha de préstamo coloco el nombre del estudiante el día en que saco el libro y la fecha en que tiene que entregarlo

- ¿Qué clase de libros hay en la biblioteca?

En esta biblioteca hay libros de texto, literatura general, libros de historia que son de la institución y otros libros que son de la secretaria de educación de un programa que se llama Vitrina Escolar.

- ¿Cómo realiza el inventario de libros?

Pues el inventario de libros lo hago contando la cantidad de libros que tengo

- ¿Qué le ayudaría a realizar en inventario de libros?

Me ayudaría algo con lo en lo que yo pueda ver la cantidad de libros que ay en la institución, la cantidad de libros que están en préstamo y cuales de los libros que están en préstamo no han devuelto.

ANEXO B
MANUAL DE USUARIO

Primera Pantalla



Log In

User Name:

Password:

Remember me next time.

[Registrarse](#)

El usuario necesita registrarse para ingresar al aplicativo. Cuando se da clic en Registrarse aparece la pantalla de Identificación Personal

Pantalla de Identificación Personal



Información Personal

Numero Identificación:

Nombre:

Apellido:

Dirección:

Teléfono:

El usuario debe llenar todos los campos que se muestran en la pantalla para continuar con el registro. Luego de escribir los datos requeridos en esta pantalla se da clic en siguiente

Pantalla Creación Cuenta de Usuario



Sign Up for Your New Account

User Name:

Password:

Confirm Password:

E-mail:

Security Question:

Security Answer:

En esta pantalla se llenan todos los campos para crear una cuenta de usuario clic en crear usuario

Pantalla inicio de préstamo



[Inicio](#)

[Usuarios](#)

[Consulta Préstamo](#)

Bienvenido: maria

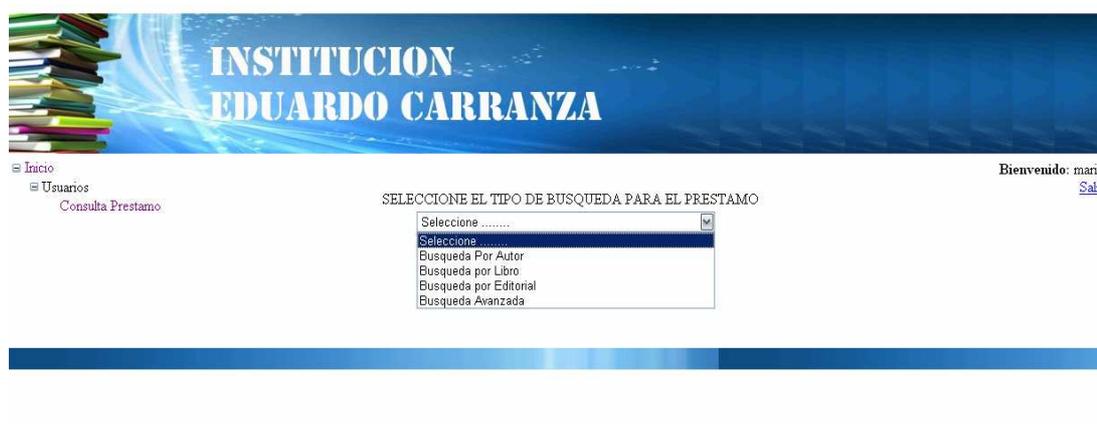
[Salir](#)

Luego de registrarse al aplicativo se muestra esta pantalla clic consulta de préstamo para iniciar con el proceso de préstamo clic en consulta de Préstamo. Si el usuario desea salir clic en salir.

Pantalla seleccionar tipo de búsqueda.



En esta pantalla seleccionamos el tipo de búsqueda para el préstamo de libro



Al dar clic en la opción por autor aparece la siguiente pantalla

Pantalla tipo de búsqueda por autor



[Inicio](#)
[Usuarios](#)
[Consulta Prestamo](#)

[Inicio](#) > [Usuarios](#) > [Consulta Prestamo](#)

Bienvenido: maria
[Salir](#)



Ingrese el nombre o apellido del autor

Buscar

En esta pantalla se escribe el nombre y apellido del autor clic en Buscar

Pantalla tipo de búsqueda por libro



[Inicio](#)
[Usuarios](#)
[Consulta Prestamo](#)

[Inicio](#) > [Usuarios](#) > [Consulta Prestamo](#)

Bienvenido: maria
[Salir](#)



Ingrese el nombre del libro

Buscar

En esta pantalla se escribe el nombre del libro clic en Buscar

Pantalla tipo de búsqueda por editorial



[Inicio](#)
[Usuarios](#)
[Consulta Prestamo](#)

[Inicio](#) > [Usuarios](#) > [Consulta Prestamo](#)

Bienvenido: maria
[Salir](#)



Ingrese el nombre de la editorial

Buscar

En esta pantalla se escribe editorial clic en Buscar



En tres pantallas anteriores en cualquier momento se puede dar clic para regresar a la pantalla seleccionar tipo de búsqueda o dar clic en salir.

Pantalla de inicio para el administrador



**INSTITUCION
EDUARDO CARRANZA**

- Inicio
- Administrador
 - Administrar Autores
 - Administrar Editorial
 - Administrar Libro
 - Administrar Area
 - Administrar TipoLibro
- Usuarios
 - Consulta Prestamo

Bienvenido: admon [Salir](#)

Pantalla ingreso autor



**INSTITUCION
EDUARDO CARRANZA**

Inicio > Administrador > Administrar Autores

- Inicio
- Administrador
 - Administrar Autores
 - Administrar Editorial
 - Administrar Libro
 - Administrar Area
 - Administrar TipoLibro
- Usuarios
 - Consulta Prestamo

Bienvenido: admon [Salir](#)

Nombre

Apellido

En esta pantalla se llenan los campos de autor y apellido es necesario llenar todos los campos

Pantalla ingreso editorial



**INSTITUCION
EDUARDO CARRANZA**

Inicio > Administrador > Administrar Editorial

- Inicio
- Administrador
 - Administrar Autores
 - Administrar Editorial
 - Administrar Libro
 - Administrar Area
 - Administrar TipoLibro
- Usuarios
 - Consulta Prestamo

Bienvenido: admon [Salir](#)

Editorial

Pantalla ingreso área



**INSTITUCION
EDUARDO CARRANZA**

Inicio > Administrador > Administrar Area Bienvenido: admon [Salir](#)

- Administrador
 - Administrar Autores
 - Administrar Editorial
 - Administrar Libro
 - Administrar Area
 - Administrar TipoLibro
- Usuarios
 - Consulta Prestamo

Area

Pantalla ingreso tipo de libro



**INSTITUCION
EDUARDO CARRANZA**

Inicio > Administrador > Administrar TipoLibro Bienvenido: admon [Salir](#)

- Administrador
 - Administrar Autores
 - Administrar Editorial
 - Administrar Libro
 - Administrar Area
 - Administrar TipoLibro
- Usuarios
 - Consulta Prestamo

Tipo Libro

Pantalla ingreso libro



**INSTITUCION
EDUARDO CARRANZA**

Inicio > Administrador > Administrar Libro Bienvenido: admon [Salir](#)

- Administrador
 - Administrar Autores
 - Administrar Editorial
 - Administrar Libro
 - Administrar Area
 - Administrar TipoLibro
- Usuarios
 - Consulta Prestamo

Codigo Libro	<input style="width: 100%;" type="text"/>
Libro	<input style="width: 100%;" type="text"/>
Descripcion	<input style="width: 100%;" type="text"/>
Año Publicacion	<input style="width: 100%;" type="text"/>
Cantidad	<input style="width: 100%;" type="text"/>
Autor	<input style="width: 100%;" type="text"/>
Editorial	<input style="width: 100%;" type="text"/>
Area	<input style="width: 100%;" type="text"/>
Tipo Libro	<input style="width: 100%;" type="text"/>

Para editar clic en administrar autor clic en editar y clic en actualizar

Pantalla Editar Autor

Inicio > Administrador > Administrar Autores

Bienvenido: admon

Administrador

- Administrar Autores
- Administrar Editorial
- Administrar Libro
- Administrar Area
- Administrar TipoLibro

Usuarios

- Consulta Prestamo

Insertar Editar

Nombre Autor
gabriel garcia

Edit Delete

Clic en editar para modificar y clic en actualizar

Pantalla Editar Editorial

Para editar clic en administrar editorial clic en editar y clic en actualizar

Inicio > Administrador > Administrar Editorial

Bienvenido: admon

Administrador

- Administrar Autores
- Administrar Editorial
- Administrar Libro
- Administrar Area
- Administrar TipoLibro

Usuarios

- Consulta Prestamo

Insertar Editar

Editorial
oveja negra

Edit Delete Select

Pantalla Área

Para editar clic en administrar área clic en editar y clic en actualizar

Inicio > Administrador > Administrar Area

Bienvenido: admon

Administrador

- Administrar Autores
- Administrar Editorial
- Administrar Libro
- Administrar Area
- Administrar TipoLibro

Usuarios

- Consulta Prestamo

Insertar Editar

Area
novela

Edit Delete Select

Pantalla Tipo Libro

Para editar clic en administrar tipo libro clic en editar y clic en actualizar

Inicio > Administrador > Administrar TipoLibro

Bienvenido: admon [Salir](#)

Administrador

- Administrar Autores
- Administrar Editorial
- Administrar Libro
- Administrar Area
- Administrar TipoLibro

Usuarios

- Consulta Prestamo

Insertar Editar

Tipo Libro
secretaria

Pantalla Libro

Para editar clic en administrar libro clic en editar y clic en actualizar

Inicio > Administrador > Administrar Libro

Bienvenido: admon [Salir](#)

Administrador

- Administrar Autores
- Administrar Editorial
- Administrar Libro
- Administrar Area
- Administrar TipoLibro

Usuarios

- Consulta Prestamo

Insertar Editar

	Codigo Libro	Nombre Libro	Descripcion	Año Publicacion	Cantidad Editorial	Area	Autor	Tipo Libro
Edit Delete	123	cien años de soledad	novela	01/01/1990 12:00:00 a.m.	2	oveja negra	novela gabriel garcia	secretaria

