



Capítulo 8. Implementación y Pruebas del Sistema

En este capítulo se presenta al sistema en términos de componentes, y también abarcara las pruebas efectuadas al Sistema de Biblioteca “ONLINE”, El primero es decir , ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares, que se obtuvieron durante la implementación de los artefactos del diseño, el segundo una vez que el software ha sido generado, el software debe ser probado para garantizar la calidad del mismo.

Una de las ventajas del proceso de desarrollo iterativo e incremental es la posibilidad de introducir los resultados de un ciclo anterior al iniciar el siguiente, así pues los resultados del Análisis y Diseño subsecuentemente se perfeccionan para lograr la implementación del producto de software

Inicialmente se describirán el lenguajes y el gestor de Base de Datos elegidos para la implementación del Sistema de Software, posteriormente se procede a presentar el sistema en términos de componentes, esto es : código fuente, ejecútales los mismos que son productos de la fase de implementación , en el anexo B contiene el listado del código fuente.

8.1. Elección del Lenguaje para el Desarrollo de la Biblioteca “ONLINE”

Al ser una aplicación web es necesario elegir una lenguaje de programación que se adecue de una manera optima en lado del servidor e interactúe de una manera transparente con el HTML, por esta razón y otras es que se ha elegido a **PHP “Personal Home Page”** como lenguaje de desarrollo, en función de las mejores características notable que lista a continuación (donde se toma en cuenta la sencillez, velocidad, flexibilidad y multiplataforma), como ya se explico las mejores características de los lenguajes web mas usados, como es ASP y PHP en el capítulo numero 4 [WWW8].

Diferencias Fundamentales entre PHP / ASP

	En PHP	En ASP
PALABRAS CLAVE	Todas las funciones y palabras clave se escriben en minúscula	Es indistinto. Pueden ir en mayúsculas o minúsculas.
SENSIBLE A MAYUSCULAS MINUSCULAS (Case sensitive)	Si	No
FIN DE INSTRUCCION	Punto y coma (;)	Cambio de línea
COMENTAR UNA LINEA	//Comentario	'Comentario
COMENTAR UN BLOQUE	/* Comentario */	'Hay que comentar 'cada línea por

		'separado
VARIABLES	No es necesario declararlas. Se identifican con un signo \$ delante	No es necesario declararlas.
EL CODIGO SE ESCRIBE ENTRE LAS ETIQUETAS	<? y ?>	<% y %>
PERMITE CONECTARSE A ORIGENES DE DATOS ODBC	Si	Si
CONEXION A SQLServer SIN ODBC	Si	Si
CONEXION A MySQL SIN ODBC	Si	No
PUEDE CORRER SOBRE LINUX	Si	No
PUEDE CORRER SOBRE WINDOWS	Si	Si

Otras Diferencias que son importantes tomar en cuenta son:

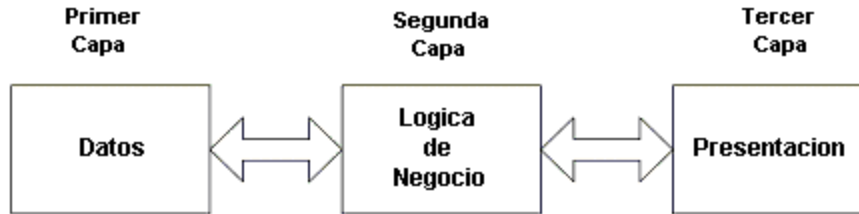
1. Velocidad.
2. Manejo de Memoria.
3. Costos.
4. Interconexión con las diferentes Bases de Datos.
5. Estilo de programación.
6. Estrategia de Migración de Plataforma Cruzada.

Según los Sitios Benchmark en todos estos aspecto PHP, supera a ASP.[WWW9]

PHP es un lenguaje Script que ejecuta en lado del servidor brindado una flexibilidad para la interconexión a Base de Datos, funciona de forma transparente con el servidor Web, funciona en diferentes plataformas y en especial en ambiente Linux sobre el Servidor Web Apache, PHP es lenguaje que contempla la programación orientada a objetos para el desarrollo.

8.2. Independencia entre la Capas

Cuando realizamos aplicaciones Web, existen un relación entre las diferentes capas (Presentación – Lógica de Negocio – Base de Datos):



- Presentación – Lógica de Negocio
- Lógica de Negocio – Base de Datos

Para lograr la independencia se recurre a herramientas y las utilizadas en el presente trabajo son:

8.2.1. Presentación - Lógica de Negocio:

La herramienta utilizada para independizar la Presentación de la Lógica de Negocio es **Smarty**, esta herramienta me permite crear Plantillas las cuales pueden ser usadas por la lógica de negocio en forma transparente, de esta manera se evita insertar código PHP en la Capa de Presentación.

Smarty es un motor de Plantillas, mas específicamente este facilita un manejo separado de la aplicación entre la lógica y el contenido de presentación, esto es mejor descrito en una situación donde el programador de aplicación y el diseñador de plantillas juegan diferentes papeles, o en la mayoría de los caso no son las mismas personas, un ejemplo, seria cuando el Diseñador de plantillas quiere cambiar de color, entonces pude hacerlo sin modificar la lógica de negocio y de la misma forma para el programador de la aplicación.

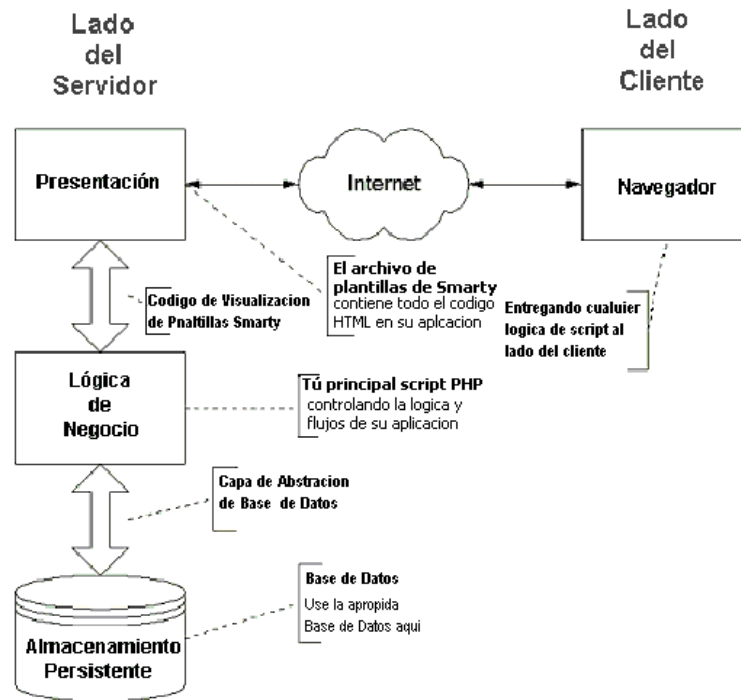
8.2.2. Lógica de Negocio – Base de Datos:

La herramienta utilizada para independizar la Lógica de Negocio del Gestor de Base de Datos es **ADODB**, esta herramienta me permite crear código PHP, con independencia del gestor de la Base de Datos, ADOdb es un conjunto de clases y Script para PHP que me permite realizar este tipo de trabajo.

Las funciones PHP de acceso a Base de Datos no son estandarizadas, esto crea una necesidad de una librería de Bases de Datos que oculte estas diferencias entre las diferentes Bases de Datos(Encapsule las Diferencias) y se pueda cambiar de base de datos en forma fácil. Actualmente **ADODB** soporta un gran número de gestores de bases de Datos(MySQL, Oracle, Microsoft SQL Server, Sybase, Sybase SQL Anywhere, Informix, PostgreSQL, FrontBase, Interbase, Foxpro, Access, ADO y ODBC).

PHP4 soporta variables de sesión, se puede almacenar la información de las usando ADOdb para una verdadera portabilidad y escalabilidad

Para una mejor Comprensión véase la siguiente grafica:



8.3. Convenios Utilizados para la Codificación

Para que cada módulo, script, clase y variable sea comprendida claramente sin necesidad de tomar tiempo en consultar los documentos de diseño, el estilo de codificación debe ser sencillo y claro, existen algunos elementos que se deben tomar en cuenta al momento de la codificación:

- Documentación Interna a Nivel Código Fuente.
- Métodos de Declaración de los Datos, Procesos y Clases.
- Técnicas de Entrada y Salida.

8.3.1. Nombre de Variables

Los nombres de las variables comenzarán entre tres o cuatro caracteres en minúscula, y así sucesivamente hasta conformar el nombre completo, la primera letra de cada palabra con mayúscula. Por ejemplo: httpLogin, numPagina, etc.

8.3.2. Nombre de Formularios

Los nombre de los formularios comenzaran con “frm” como primeras letras seguido del nombre que lo identificara con la primera letra mayúscula. Por ejemplo: frmLogin, frmPrestamos, etc.

8.3.3. Nombre de Scripts

Los Script por defecto comenzaran con tres o cuatro letras minúsculas seguido del nombre del proceso el que realizan, al igual que las variables tienen la primer letra de cada palabra con mayúsculas. Por ejemplo: admRegistrar, admModificar, admEliminar, etc.

8.3.4. Nombre de Clases

Las Clases por defecto tendrán delante del nombre que los identifique el carácter “C” y comenzaran con la primera letra mayúscula, tendrá asociados procedimiento que se especificaran los nombres de acuerdo a la función que realizan comenzando con letras minúsculas y luego con palabras hasta completar el nombre, dichos palabra empezaran con mayúsculas. Por ejemplo CPrestamos, CConsultas, CAdministradores, etc.

8.4. Elección del Gestor de Base de Datos

Las características que pueda tener el Gestor de Base de Datos es muy importante, ya que tienen que cumplir con exigencias transaccionales y de ella dependen en gran medida el buen funcionamiento de la base de datos y su vida útil, entre las mas importantes se pueden mencionar las siguientes:

- 1.- Control de redundancia de datos que evita que los usuarios tengan información repetida y que provoca problema en la administración.

2.- Restricción de accesos no autorizados para crear cuentas y asignar privilegios de accesos, evitando el ingreso de personas no autorizadas a ciertas operaciones, esta característica es vital en un gestor de Base de Datos.

3.- Representación de vínculos complejos entre datos, sin perder los enlaces manteniendo los vínculos complejos de los datos, obtener y actualizar con rapidez y eficiencia datos que estén mutuamente relacionados.

4.- Cumplir con las restricciones de integridad, como el tamaño de los datos establecidos en la base de datos, no todas las restricciones podrán ser especificadas en Gestor de Base de Datos, otras pueden requerir verificación mediante programas de actualización cuando se introducen los datos.

5.- Robustez en el respaldo y recuperación de los datos cuando ocurren fallas en el sistema, mediante subsistemas de respaldos.

Al tomar en cuenta las características generales que debe cumplir un gestor de Base de Datos, también se tienen unas características operacionales:

- Transacciones (Transactions).
- Disparadores (Triggers).
- Restricciones (Constraints).
- Replicación (Replication).
- Backup y Recuperación (Backup & Recovery).
- Reglas (Rules).
- Procedimientos Almacenados / funciones (Stored Procedures/Functions).
- Integridad Referencial.
- Outer Joins.
- Sintaxis ANSI SQL 89, 92 y 98.
- Logging.
- Extensivo y programable.

- Orientado a Objetos
- Características sofisticadas de integridad de datos.
- Tipos de datos y funciones definidos por el usuario.
- Cliente / servidor, entre otros.

Se tomo como gestores posibles alternativas a dos de gestores de base de datos mas utilizados en el mundo de 'OPEN SOURCE DATABASE SOFTWARE' aquí existen una gran disputa entre los dos mas importante que son MySQL y PostgreSQL, se realizaron las comparaciones necesarias y se concluyo que el gestor ha utilizar seria **PostgreSQL**, al cumplir ciertas características que se adecuan ha esta proyecto, las diferencias y características mas importantes son:

8.4.1. MySQL:

- Su principal objetivo de diseño fue la VELOCIDAD. Se sacrificaron algunas características esenciales en sistemas más "serios" con este fin.
- Otra característica importante es que consume MUY POCOS RECURSOS, tanto de CPU como de memoria.
- Licencia GPL a partir de la versión 3.23.19.

8.4.1.1. Ventajas:

- Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- Mejores utilidades de administración (backup, recuperación de errores, etc).
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- MySQL se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla "en caliente".

8.4.1.2. Inconvenientes:

- No soporta transacciones, "roll-backs" ni subselects.
- No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

8.4.2. PostgreSQL:

- Postgres intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.
- Licencia BSD.

8.4.2.1. Ventajas:

- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.

8.4.2.2. Inconvenientes:

- Consume BASTANTES más recursos y carga más el sistema.
- Límite del tamaño de cada fila de las tablas a 8k!!! (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).
- Es de 2 a 3 veces más lenta que MySQL.
- Menos funciones en PHP.

En cuanto a consideraciones de estabilidad del servidor, cada comparativa da datos contradictorios. En general parece que MySQL es más estable (aunque también hay gente que opina lo contrario), y que Postgres tiende a desperdiciar memoria y sobrecargar bastante el sistema (aunque de nuevo, hay opiniones distintas).

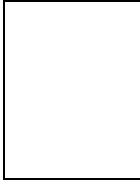
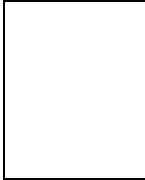
Como conclusión a la comparación entre MySQL y Postgres, parece aceptado que MySQL junto con Apache y PHP forman un buen equipo para servir páginas web con contenido dinámico, discusiones, noticias, etc. En general, sistemas en los que la

velocidad y el número de accesos concurrentes sea algo primordial, y la seguridad no sea muy importante (pueda bastar con hacer backups periódicos que se restaurarán tras una caída del servidor).

En cambio, para sistemas más serios en las que la consistencia de la BD sea **fundamental** (BD con información realmente importante, bancos, etc.) PostgreSQL es una mejor opción pese a su mayor lentitud.

8.4.3. Tabla Comparativa entre MySQL y PostgreSQL:

Se presenta continuación una de las tabla comparativas, mas completas emitidas por a principios de 2004 [www10]:

<u>Características</u>	<u>MySQL</u>	<u>PostgreSQL</u>
Version:	Mysql-4.0.xx	PostgreSQL 7.4.x
		
Licencias:	Dual: GPL y Comercial	BSD
<u>Plataformas:</u>	Linux, Solaris,HP-UX, MacOS, AIX, SCO, IRIX, FreeBSD, NetBSD, OpenBSD, Windows , BSDI, DEC, OS/2, Compaq Tru64	Linux, Solaris,HP-UX, AIX, IRIX, FreeBSD, OpenBSD, NetBSD, MacOS, SCO OpenServer, SCO Unixware, BeOS, BSDI, Compaq Tru64, QNX, Windows
SQL estándar COMPLIANCE	Media	Alta
Velocidad:	Media / Alta	Media

Estabilidad:	Alta / Muy Alta	Alta
Integridad de Datos	NO (Planeado v4.1)	Si
Característica de Seguridad	Alta	Media / Alta
Métodos de Autenticación	?	md5, crypt, password and Kerberos
Acceso SSL	Si	Si
Bloqueo y Concurrencia	Media / Alta	Alta
Soporte de Vistas	Si (Planeado v5.0)	Si
Soporte de Esquemas	NO	Si
Soporte de Sub selección	NO (Planeado v4.1)	Si
Almacenamiento de Procedimientos	Si (v5.0)	Si (PL/pgSQL, PL/Perl, PL/TCL)
Soporte de Unicode	NO (Planeado v4.0)	Si
Soporte de Disparadores	NO (Planeado v4.1)	Si
Programación de Interfaces	ODBC, JDBC, C/C++, OLEDB, Delphi, Perl, Python, PHP	ODBC, JDBC , C/C++, Embedded SQL (in C), Tcl/Tk, Perl, Python, PHP
Transacciones	Si	Si
Llaves Foráneas	Si (Planeado v4.1)	Si
Replicación	Si	Si
Ipv6 support	?	Si
Clustering	NO	NO
Backups Calientes	Si	Si

8.5. Descripción de la Arquitectura del Modelo de Implementación

Se muestran los componentes mas importantes para este trabajo. Las aplicaciones son:

- Modulo de administración de Sistema(usuarios, grupos).
- Módulo de administración Biblioteca(libros, bibliotecarios, prestamos, devoluciones y políticas).
- Modulo para reportes y estadísticas(jefes de biblioteca).
- Módulo consultas y reservas de Material(usuarios).

Modelo de Implementacion (Arquitectura General) - Sistema de Biblioteca ON-LINE

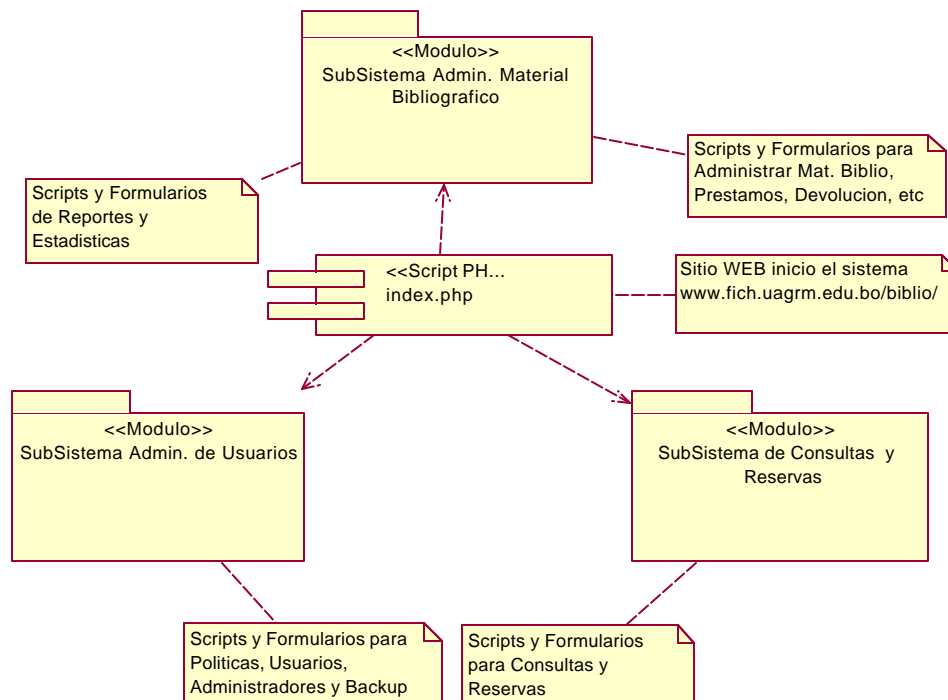


Fig. 8.1 Descripción De La Arquitectura General – Biblioteca ONLINE

8.5.1. Modelo de Implementación del Modulo de Administrar Usuario

El modulo de Administrar Usuarios utiliza las librería ADOBDDB para el acceso a Base de Datos PostgreSQL, Smarty para crear Templates HTML para PHP, en este modulo implica la implementación de procesos para Administrar Usuarios(Operaciones Alta, Baja y Modificación), Administrar Bibliotecarios(Operaciones Alta, Baja y Modificación), Administrar Políticas(Operaciones Alta, Baja y Modificación), Administrar Grupos(Privilegios) y Administrar RespalDOS.

Modelo de Implementación del Modulo de Administrar Usuarios - Biblioteca ON-LINE

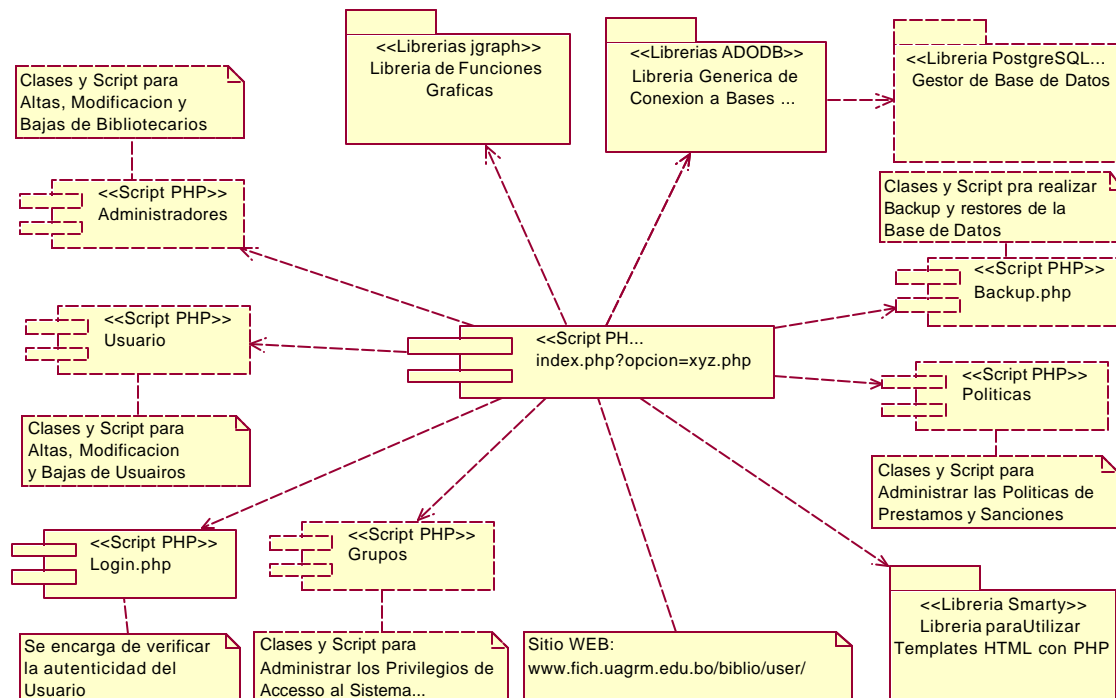


Fig. 8.2 Modelo de Implementación del Modulo Administración de Usuarios

8.5.2. Modelo de Implementación del Modulo de Administrar Material Bibliográfico

El modulo de Administrar Material Bibliográfico utiliza los siguientes paquete y librerías JpGraph (para Realizar Graficas), Smarty (que Sirve para crear templates HTML), ADODB (Que nos permite Administrar una Base de Datos) y PHP. Este Modulo se encarga de la administración de los prestamos, las devoluciones, Eliminar las Sanciones aplicadas por falta al reglamento interno.

Modelo de Implementación del Modulo de Administrar Mat. Bibliografico - Biblioteca ON-LINE

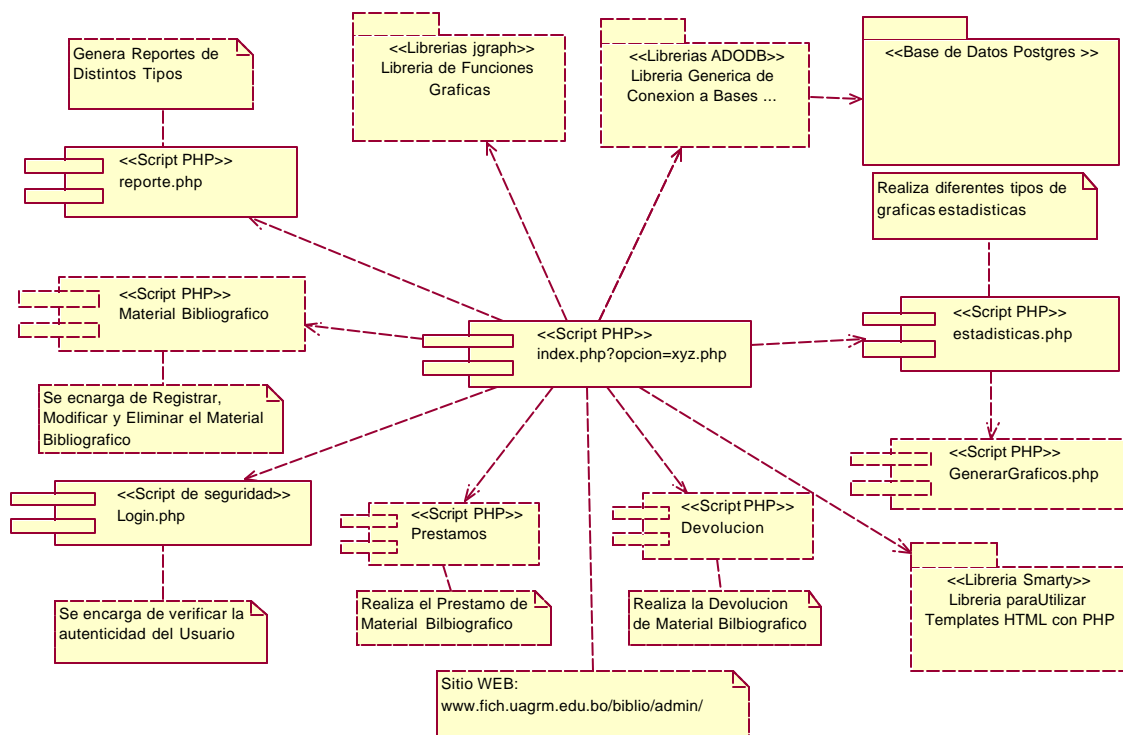


Fig. 8.3 Modelo de Implementación del Modulo Administración de Material Bibliográfico

8.5.3. Modelo de Implementación del Modulo de Consultas y Reservas

El modulo de Administrar de Consultas y Reservas al igual que la anterior utiliza el paquete Librería PHP y Base de Datos PostgreSQL, se encarga de la generación de Consultas ya sea por Autor, tema, titulo y me genera una lista de este material bibliográfico, también se pueden realizar reservas del material bibliográficos para esto se debe ingresar al sistema con su usuario y contraseña.

Modelo de Implementación del Modulo de Consultas y Reservas - Biblioteca ON-LINE

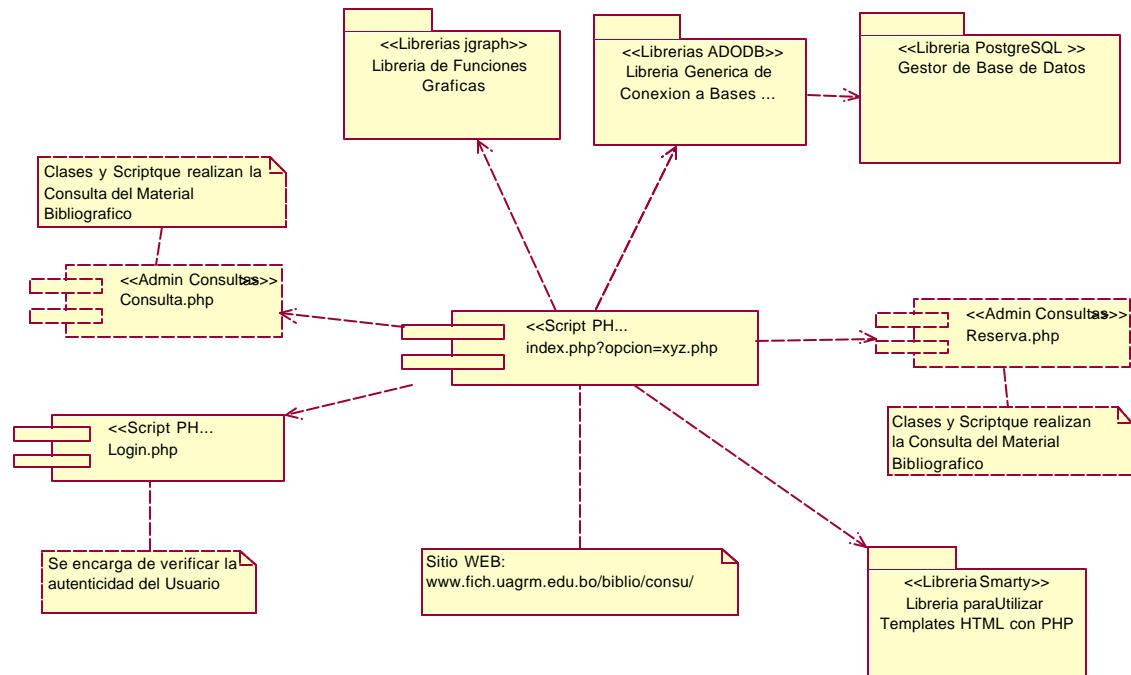


Fig. 8.4 Modelo de Implementación del Modulo de Consulta y Reservas

8.5.4. Casos de Prueba

Las pruebas de software son un elemento crítico para garantizar la calidad y representa una revisión final de las especificaciones, del diseño y de la codificación. El desarrollo de sistemas de Software implica una serie de actividades de producción en las que las posibilidades de que aparezca un fallo humano son enormes, los errores pueden empezar a darse desde el primer momento del proceso, en los que los objetivos pueden estar especificados en forma errónea o imperfecta, así como en los posteriores pasos de diseño y desarrollo debido a imposibilidad humana de trabajar y comunicarse en forma perfecta, el desarrollo de software ha de ir acompañado de una actividad que garantice la calidad [RSP2002].

Una estrategia de pruebas del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. Un software Orientado a Objetos con la aplicación del Proceso Unificado como metodología de desarrollo de software, el concepto de unidad cambia dando cabida a los casos de uso que ahora se convierten en casos de pruebas. [RSP2002] Las pruebas de integración se irán realizando dentro de cada caso de uso, donde se irán integrando las clases independiente y dependiente, este mismo criterio se aplicará a todos los casos de uso hasta obtener el sistema completo. Las pruebas de validación serán dirigidas por los casos de uso, en la cual centraremos la atención en las entradas y salidas que produzca el sistema, en esta etapa se tomará como métodos de prueba:

- Pruebas de unidad(Página Web)
- Prueba del Modelo de Diseño(USN :Unidad Semántica de Navegación)

Las cuales tomará como escenario lo caso de uso.

Los casos que se seguirán para identificar los casos de pruebas son:

1. Cada caso de prueba debe ser identificado separada y explícitamente.
2. Debe declararse el propósito de la prueba.

3. Debe desarrollar un lista de los paso a seguir como secuencia de la prueba pero además debe contener:
 - a. Definición de la listas de estados, específicos para probar las unidad.
 - b. Lista de Mensajes y Operaciones, que se ejercitaran como consecuencia de la prueba.
 - c. La lista de excepciones, que pueden ocurrir conforme al objeto se comprueba.
 - d. Lista de Condiciones externas.
 - e. Información Adicional que ayudara ala comprensión e implementación de la prueba.

8.5.5. Modelo de Prueba

Las pruebas son el proceso de ejercitar el Software con la intención de encontrar y corregir errores, las aplicaciones WEB representa un reto significativo realizar al tener diferentes tipos de sistemas operativos, Navegadores y protocolos de comunicación. [RSP2002]

Los casos de prueba especifican una forma de probar el sistema(incluyendo entrada, resultado, condiciones). Los casos de uso se transformaran en casos de pruebas, siguiendo de esta forma la línea de conducción y ejecución que dirige al proyecto.

Se encontraron errores durante la etapa de implementación de la aplicación, como en la etapas de pruebas, los cuales fueron corregidos en su totalidad.

Las pruebas de unidad(Paginas WEB) se hacen en gran medida en la etapa de implementación del caso de uso, verificando las operaciones que conlleva las consultas(Cliente / Servidor) en línea al servidor, en su partes criticas.

Las pruebas de validación se efectuaron con consultas (peticiones al servidor y Script en el cliente) visibles al usuario y salidas reconocibles del sistema (respuestas del servidor).

Las pruebas de integración se utiliza la Prueba del Modelo de Diseño(USN :Unidad Semántica de Navegación), en las cuales se verifican los caminos de navegación que tienen cada usuario en función a su nivel de Accesos, cada uno de ellos tiene una Unidad Semántica de Navegación la cual nos da una guía en la cual indica por cuales Nodos(Paginas WEB) el usuario navegara. Correcciones importantes realizadas:

1. Modificar los USN actualizando los enlaces para lograr una WoN(Forma de Navegación) mas eficientes.
2. se añadieron tablas auxiliares para el control de inicio de sesión al sitio.
3. se han reutilizados paginas y Script, reduciendo de esta forma las líneas de códigos.
4. se han probado valores críticos, por ejemplo números donde deber ser cadenas, fechas donde debe ser numero, fechas herradas, etc.
5. Ingresar en forma directa a una pagina, sin haber iniciado una sesión.

Como ejemplo de la forma de realización de los casos de pruebas a continuación se detallan los casos de pruebas mas importantes en el proyecto:

1) Caso de Prueba : Administrar Material Bibliográficos

Se requiere registrar un (Libro, tesis, etc.) Material Bibliográfico por motivo de compra de nuevos libro en la biblioteca de la Facultad Integral del Chaco, el encargado de este trabajo es don Juan Carlos Paniagua que funcionario administrativo dela U.A.G.R.M. con cargo de Bibliotecario en esta unidad, se debe registrar la transacción y emitir un comprobante registros como resultado de esta operación.

Entrada :

Se ingresaran los datos correspondientes a al libro o material bibliográfico (Código, Titulo, Autor, Editorial, etc en función al material bibliográfico), los datos del bibliotecario(Jefe de Biblioteca o Bibliotecario), la cantidad de ejemplares.

Resultado :

- El Bibliotecario recibe un comprobando de ingreso del material bibliográfico, en el cual indica los datos principales de material, la fecha y hora de la transacción y los datos del bibliotecario.
- Se ha registrado el comprobante de acuerdo a lo establecido por su tipo de Material Bibliográfico.

Condiciones :

Debe estar registrado el Administrador de biblioteca Previamente, dentro de los usuarios normales y con el Rango de Bibliotecario o Jefe de Biblioteca.

Procedimiento de Prueba :

1. Se deber dar click en el Hiperenlace “**Registrar**” dentro de la sub opciones del Menú Material Bibliográficos.
2. Se ingresan los datos correspondientes a la información del Material Bibliográfico tales como titulo, autor, edición, etc.
3. Ser verifica que hayan introducido valores en los campos que corresponde como obligatorios del formulario HTML.
4. Se presiona el Botón “**Registrar**” e “**Imprime**” el comprobante correspondiente al tipo de Material registrado.

2) Caso de Prueba : Realizar Reserva

Se requiere registrar una transacción de un algún Material Bibliográfico por motivo de realizar una reservar en la biblioteca de la Facultad Integral del Chaco, la persona que puede realizar esta reserva es un Usuario(Estudiante, Profesor, Administrativo o

Particular que este registrado en el sistema) de la U.A.G.R.M., se debe registrar la transacción y emitir un mensaje como resultado de esta operación.

Entrada :

Se ingresaran los datos correspondientes a al código del libro(s) o material bibliográfico, los datos del Usuario que esta realizando la reserva del material bibliográfico.

Resultado :

- El Usuario recibe un mensaje en el cual se informa del estado del proceso, en el cual indica los datos principales de material, la fecha y hora de la transacción, los datos del usuario. Se especifica también el periodo de duración que tiene dicha reserva.
- Se ha registrado el comprobante de acuerdo a lo establecido por el reglamento interno.

Condiciones :

Debe estar registrado el usuarios, el código del libro debe ser correcto y debe pertenecer a libro introducidos y actualmente activos, también debe existir el código del usuario que realiza la reserva, la hora y fecha del servidor de base Servidor de Paginas Web debe ser correctas.

Procedimiento de Prueba :

1. Se deber dar click en el Hiperenlace “**Reserva**” dentro de la sub opciones de Menú Usuario.
2. Se ingresan los datos correspondientes al código del Material Bibliográfico, código o registro del Usuarios etc.
3. Ser verifica que hayan introducido valores en los campos que corresponde como obligatorios del formulario HTML.

4. Se presiona el Botón “**Reservar**” emite la correspondiente respuesta si la transacción se ha completado de una manera exitosa, o caso contrario emite el correspondiente mensaje de error que se haya suscitado, (en este caso puede haber sanciones por morosidad, no existir ejemplares, etc).

3) Caso de Prueba : Realizar Préstamo

Se requiere registrar una transacción de un algún Material Bibliográfico por motivo de préstamo en la biblioteca de la Facultad Integral del Chaco, el encargado de este trabajo es don Juan Carlos Paniagua que es funcionario administrativo dela U.A.G.R.M. con cargo de Bibliotecario en esta unidad, se debe registrar la transacción y emitir un mensaje como resultado de esta operación.

Entrada :

Se ingresaran los datos correspondientes a al código la reserva de los libro(s) o material bibliográfico, los datos del bibliotecario(Jefe de Biblioteca o Bibliotecario) y los datos del Usuario que esta realizando el préstamo del material bibliográfico.

Resultado :

- El Bibliotecario recibe un mensaje en el cual se informe del estado del proceso, en el cual indica los datos principales de material, la fecha y hora de la transacción, los datos del usuario y los datos del bibliotecario, caso contraria informaría si no existe ejemplares del elemento introducido.
- Se ha registrado el comprobante de acuerdo a lo establecido por el reglamento interno.

Condiciones :

Debe estar registrado el Administrador de biblioteca Previamente, dentro de los usuarios normales y con el Rango de Bibliotecario o Jefe de Biblioteca, el código del libro debe

ser correcto y debe pertenecer a alguno de libro introducidos y actualmente activos, también debe existir el código del usuario que realiza el préstamo, donde se especifica que el préstamo se debe de realizar, si el usuario no tiene prestamos alguno, prestamos en mora, o no tienen alguna sanción en proceso, la hora y fecha del servidor de base Servidor de Paginas Web debe ser correctas.

Procedimiento de Prueba :

1. Se deber dar click en el Hiperenlace “**Prestamos**” dentro de la sub opciones de Menú Material Bibliográfico.
2. Se verifica si se ha realizado una reserva del dicho material, si se ha realizado la reserva se realiza el proceso normal, caso contrario se realiza un reserva del material y se continua con el proceso normal de préstamo.
3. Se ingresan los datos correspondientes al del reserva, numero o codigo de la reserva.
4. Ser verifica que hayan introducido valores en los campos que corresponde como obligatorios del formulario HTML.
5. Se presiona el Botón “**Prestar**” emite el correspondiente respuesta si la transacción se ha completado de una manera exitosa, o caso contrario emite el correspondiente mensaje de error que se haya suscitado.

4) Caso de Prueba : Realizar Devolución

Se requiere registrar una transacción de un algún Material Bibliográfico por motivo de devolución en la biblioteca de la Facultad Integral del Chaco, el encargado de este trabajo es don Juan Carlos Paniagua que es funcionario administrativo de la U.A.G.R.M. con cargo de Bibliotecario en esta unidad, se debe registrar la transacción y emitir un mensaje como resultado de esta operación.

Entrada :

Se ingresaran los datos correspondientes a al código del libro(s) o material bibliográfico, los datos del bibliotecario(Jefe de Biblioteca o Bibliotecario) y los datos del Usuario que esta realizando la devolución del material bibliográfico.

Resultado :

- El Bibliotecario recibe un mensaje en el cual se informa del estado del proceso, en el cual indica los datos principales de material, la fecha y hora de la transacción, los datos del usuario y los datos del bibliotecario. Se verifica si la devolución no conlleva una Sanción por algún motivo(morosidad, maltrato, deuda, etc.)
- Se ha registrado el comprobante de acuerdo a lo establecido por el reglamento interno.

Condiciones :

Debe estar registrado el Administrador de biblioteca Previamente, dentro de los usuarios normales y con el Rango de Bibliotecario o Jefe de Biblioteca, el código del libro debe ser correcto y debe pertenecer a libro introducidos y actualmente activos, también debe existir el código del usuario que realiza la devolución, la hora y fecha del servidor de base Servidor de Paginas Web debe ser correctas.

Procedimiento de Prueba :

1. Se deber dar click en el Hiperenlace **“Devolución”** dentro de la sub opciones de Menú Material Bibliográfico.
2. Se ingresan los datos correspondientes al código del Material Bibliográfico, código o registro del Usuarios etc.
3. Ser verifica que hayan introducido valores en los campos que corresponde como obligatorios del formulario HTML.
4. Se presiona el Botón **“Devolver”** emite la correspondiente respuesta si la transacción se ha completado de una manera exitosa, o caso contrario emite el

correspondiente mensaje de error que se haya suscitado, (en este caso puede haber sanciones por morosidad, etc).



Conclusiones

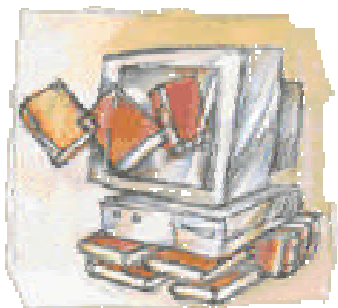
A la conclusión del presente estudio y desarrollo de la “Biblioteca ONLINE”, se ha obtenido un software que puede ser utilizado por las bibliotecas de las Facultades de la U.A.G.R.M. ya que la mayoría presentan una administración similar.

La “Biblioteca ONLINE” cuenta con características Administración de las Bibliotecas convencionales, adicionalmente tiene una capacidad para incorporar libros en formatos digitales, con lo que se convierte en una de las aplicaciones Web con capacidad carga de trabajos digitales al servidor y de administración desde cualquier lugar que se tenga acceso al sistema.

La “Biblioteca ONLINE” es un software multiplataforma en el lado del cliente al usar un navegador web, la configuración en la parte del servidor es adaptable a cualquier entorno o plataforma en el que este instalado PHP, ha sido probado para el funcionamiento en Ambiente Windows y Linux(Servidores Web, Gestores de Bases de Datos de ambas plataformas).

Durante el desarrollo del software se utilizó el Lenguaje Unificado de Modelado UML, que permitió visualizar, especificar, construir y documentar los artefactos del presente trabajo, obteniéndose de esta manera una forma estándar de escribir modelos que componen un sistema software.

Para el desarrollo del sistema fue utilizado el Proceso Unificado de Desarrollo de Software, que ofrece un conjunto de técnicas que aprovechan las ventajas del estándar UML, permitiendo así desarrollar el software iterativa e incrementalmente, mediante componentes, dirigido por casos de uso y centrado en la arquitectura.



Recomendaciones

Para que el sistema de la “Biblioteca ONLINE”, funcione de una manera regular se recomienda a las Autoridades Facultativas de esta Unidad.

Se recomienda, capacitar al personal de la Biblioteca de la Facultad Integral del Chaco con conocimientos de Internet, codificación DDC para el material bibliográfico, para la parte de administración de la Biblioteca.

Se Recomienda, La contratación de personal con conocimiento avanzado en la edición de textos digitales, para enriquecer el material bibliográfico digital con el que cuenta, de una forma regular y planificada.

Se Recomienda, La creación de un subdominio en Internet (www.fich.uagrm.edu.bo) para la publicación de documentación y biblioteca de esta unidad, en el dominio de nuestra universidad.

Capítulo 8. Implementación y Pruebas del Sistema	188
8.1. Elección del Lenguaje para el Desarrollo de la Biblioteca “ONLINE”	189
8.2. Independencia entre la Capas	191
8.2.1. Presentación - Lógica de Negocio:.....	191
8.2.2. Lógica de Negocio – Base de Datos:.....	192
8.3. Convenios Utilizados para la Codificación	193
8.3.1. Nombre de Variables	193
8.3.2. Nombre de Formularios	194
8.3.3. Nombre de Scripts	194
8.3.4. Nombre de Clases	194
8.4. Elección del Gestor de Base de Datos.....	194
8.4.1. MySQL:	196
8.4.1.1. Ventajas:	196
8.4.1.2. Inconvenientes:.....	197
8.4.2. PostgreSQL:.....	197
8.4.2.1. Ventajas:	197
8.4.2.2. Inconvenientes:.....	197
8.4.3. Tabla Comparativa entre MySQL y PostgreSQL:	198
8.5. Descripción de la Arquitectura del Modelo de Implementación	200
8.5.1. Modelo de Implementación del Modulo de Administrar Usuario	201
8.5.2. Modelo de Implementación del Modulo de Administrar Material Bibliográfico	202
8.5.3. Modelo de Implementación del Modulo de Consultas y Reservas.....	203
8.5.4. Casos de Prueba.....	204
8.5.5. Modelo de Prueba.....	205
Conclusiones.....	213
Recomendaciones.....	215