

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Reporte Final del Proyecto de Integración

Licenciatura en Ingeniería en Computación

Proyecto tecnológico

Sistema de gestión para la sección de librería y papelería de la UAM Azcapotzalco

Alejandra Tapia Miranda

210301167

Asesora:

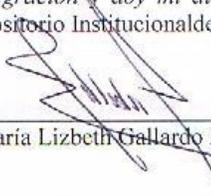
Dra. María Lizbeth Gallardo López

No. Económico: 30761

Trimestre 2014 Otoño

9 de enero de 2015

Yo, DRA. MARÍA LIZBETH GALLARDO LÓPEZ, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para supublicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.


Dra. María Lizbeth Gallardo López

Yo, ALEJANDRA TAPIA MIRANDA, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.


Alejandra Tapia Miranda

Resumen

El presente documento describe el desarrollo de un sistema de información que operará en la red local de la sección de la librería y papelería de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco (UAM-A). Este sistema de información apoyará dos procesos principales: i) Realizar solicitudes de libros o artículo de papelería que se han agotado; estas solicitudes provienen de miembros de la Comunidad Universitaria. ii) Realizar pedidos de libros a distribuidores. iii) Gestionar distribuidores. Estos dos procesos son llevados a cabo por la Coordinación de librería de la UAM-A.

Actualmente, la sección de librería cuenta con un sistema de cómputo para consultar el catálogo de libros disponibles; sin embargo, las solicitudes de los usuarios se registran en bitácoras. La Coordinación emplea bitácoras para gestionar sus proveedores, y para elaborar sus pedidos retoma las bitácoras de solicitudes, para posteriormente editar en un formulario el pedido.

En el desarrollo de este proyecto se utilizaron patrones de diseño de software, el lenguaje unificado de modelado (UML); para la implementación, se empleó el lenguaje de programación Java, la framework Struts 2, además JavaMail API para enviar correos masivos, e iText para crear y manipular archivos PDF.

Índice

1. Introducción	7
2. Antecedentes	8
3. Justificación	9
4. Objetivos	9
5. Marco teórico	10
5.1 Sistemas de información administrativos.....	10
5.2 Aplicación web.....	10
5.3 Patrones de diseño	11
6. Desarrollo del proyecto	12
6.1 Metodología empleada en el desarrollo del proyecto	12
6.2. Diseño del sistema.....	13
6.2.1 Diagramas de casos de uso	13
6.2.2 Casos de uso texto Solicitar libro	15
6.2.3 Modelo de dominio general	18
7. Arquitectura del sistema	19
7.1 Diagrama de paquetes y diagrama de clases.....	20
7.2 Patrones de diseño aplicados en LIBREP.....	24
7.3 Estructura de la base de datos.....	26
7.4 Uso del sistema.....	29
8. Hardware y software necesario	35
8.1 Software.....	35
8.1 Tecnología del desarrollo de la aplicación.....	35
8.1.2 Tecnología para la instalación y puesta en marcha la aplicación.....	35
8.2 Hardware.....	35
9. Resultados	36

10. Análisis y discusión de resultados	37
Soluciones.....	37
Limitaciones.....	37
11. Conclusiones	37
12. Perspectivas del proyecto.	40
Bibliografía	46
Anexo A	

Índice de figuras

FIGURA 1. MÓDULO DE CONSULTA DE CATALOGO	8
FIGURA 2. REGISTRO EN BITÁCORAS	8
FIGURA 3. DIAGRAMA DE CASOS DE USO GENERAL	13
FIGURA 4. DIAGRAMA DE CASOS DE USO “GESTIONAR SOLICITUDES DE LIBROS”	14
FIGURA 5. MODELO DE DOMINIO GENERAL	18
FIGURA 6. ARQUITECTURA DE APLICACIÓN WEB: CLIENTE LIGERO	19
FIGURA 7. ESTRUCTURA DE LA APLICACIÓN LIBREP	20
FIGURA 8. MODELO.....	21
FIGURA 9. CLASES GESTORAS DE LAS OPERACIONES CRUD LIBRO.....	22
FIGURA 10. CONTROLADOR . ESPEC.....	23
FIGURA 11. CONTROLADOR . SOLICITUD	23
FIGURA 12. CONTROLADOR . REGISTRO	24
FIGURA 13. PATRÓN STRATEGY EN EL MODELO	25
FIGURA 14. PATRÓN FACHADE AL AGREGAR UNA ESPECIFICACIÓN DE LIBRO	25
FIGURA 15. DIAGRAMA E-R (ENTIDAD-RELACIÓN) DE LA BASE DE DATOS DE LIBREP.....	28
FIGURA 16. MENÚ PARA ALUMNOS	29
FIGURA 17. REGISTRO DE LIBRO	30
FIGURA 18. SELECCIÓN DE UN REGISTRO	30
FIGURA 19. AGREGAR UN REGISTRO	31
FIGURA 20. MENSAJE DE ÉXITO AL REGISTRAR UN LIBRO	31
FIGURA 21. MENSAJE DE ERROR	31
FIGURA 22. MENÚ DEL ADMINISTRADOR	32
FIGURA 23. MENÚ GESTIONAR SOLICITUDES DE LIBROS.....	32
FIGURA 24. MODIFICAR LOS REGISTROS EN BASE A UNA FECHA	33
FIGURA 25. LISTADO DE REGISTROS EN BASE A UNA FECHA.....	33
FIGURA 26. MODIFICAR UN REGISTRO EN BASE A UNA CLAVE DE REGISTRO	33
FIGURA 27. BUSCAR REGISTROS PROPORCIONANDO EDITORIAL Y DEMANDA	34
FIGURA 28. LISTA DE REGISTROS	34
FIGURA 29. MENSAJE NO HAY REGISTROS CON ESA EDITORIAL Y DEMANDA PROPORCIONADA	34

1. Introducción

Los procesos automatizados son factores de vital importancia para una organización. Actualmente se observa innovación y desarrollo de tecnologías de la información que impactan a la sociedad con aportes en los distintos sectores económicos, sociales, políticos, y educativos, entre otros.

Centrados en el sector educativo y en particular la Universidad Autónoma Metropolitana Unidad Azcapotzalco (UAM-A), la sección de librería y papelería tiene un problema de ineficiencia en el proceso de solicitud de un libro o de un material de papelería que no se encuentra en existencia. El proceso de solicitud es ineficiente tanto para un alumno como para un profesor. El proceso para solicitar un libro es: i) registrar el libro que requiere en una bitácora, ii) acudir en varias ocasiones a la librería para verificar si el libro está disponible, y solo en caso afirmativo, iii) realizar la compra. En la solicitud de un material el alumno lo manifiesta de forma verbal al vendedor, pero en ocasiones esta solicitud no es registrada; no hay una bitácora disponible para este caso.

La Coordinación, por su parte, antes de poder realizar pedidos a las editoriales o a los distribuidores debe realizar el siguiente proceso: i) revisar la bitácora de solicitudes de alumnos o profesores diariamente; ii) descartar los registros incompletos o que tienen errores; iii) contabilizar las solicitudes del mismo libro; iv) clasificar los libros por editorial; v) verificar el número de libros por editorial y si el distribuidor puede surtir esa cantidad; vi) capturar el pedido correspondiente.

Este proyecto terminal tuvo como principal objetivo apoyar tanto al proceso de solicitud de libros y material de papelería, como al proceso de formular los pedidos, a través de un sistema de información, que opere en la red local de la sección de la librería y papelería de la UAM-A.

3. Justificación

El sistema de información para la sección de librería y papelería de la UAM-A, al cual llamaremos desde ahora LIBREPpermitirá: i) Automatizar procesos solicitud de libros o material de papelería; ii) tener el nombre del alumno o profesor quien realiza la solicitud; iii) establecer un mecanismo que dé prioridad en las solicitudes elaboradas por profesores; iv) establecer un mecanismo que permita saber si una solicitud es idéntica a otra, evitando así la duplicidad; v) saber cuántos alumnos requieren la misma solicitud; vi) clasificar los registros de solicitudes de libros por la editorial; vii) clasificar los registros de solicitudes material de papelería por la marca; viii) notificar cuando la cantidad de libros en solicitudes con la misma editorial pueden ser surtidos por un distribuidor; ix) generar pedidos de forma automática en formato PDF; x) informar al alumno o profesor sobre el estado de su solicitud vía correo electrónico.

4. Objetivos

4.1 Objetivo general: Diseñar e implementar un sistema de información que facilite los procesos de solicitud de libros y material de papelería por parte de los usuarios, así como el levantamiento de pedidos por parte de la Coordinación de la sección de librería de la UAM-A.

4.2 Objetivos particulares:

1. Diseñar e implementar un módulo que permita gestionar las solicitudes tanto de libros como de material de papelería, que realizan los usuarios de la librería.
2. Diseñar e implementar un módulo que permita gestionar los pedidos tanto de libros como de material de papelería, que realiza la coordinación de la librería.
3. Diseñar e implementar un módulo que permita gestionar la cartera de distribuidores tanto de libros como de material de papelería, con que cuenta la coordinación de la librería.
4. Diseñar e implementar un módulo que permita gestionar usuarios.
5. Integrar los módulos anteriores en una aplicación *web*.
6. Probar la funcionalidad de los módulos anteriores.

5. Marco teórico

¿Qué son los sistemas de información administrativa?

Los sistemas de información administrativa (MIS) son sistemas de información computarizados que funcionan debido a la decidida interacción entre las personas y las computadoras. Al requerir que las personas, el software y el hardware funcionen en concierto, los sistemas de información administrativa brindan soporte a los usuarios para realizar un espectro más amplio de tareas organizacionales, incluyendo los procesos de análisis y toma de decisiones.

Para acceder a la información, los usuarios del sistema de información administrativa comparten una base de datos común; esta almacena tanto los datos como los modelos que permiten al usuario interactuar con ellos, interpretarlos y aplicarlos. Los sistemas de información administrativa producen información que se utiliza en el proceso de toma de decisiones. También pueden ayudar a integrar algunas de las funciones de información computarizadas de una empresa [1].

¿Qué es una aplicación web?

Una aplicación web (*web-based application*) es un tipo especial de aplicación **cliente/servidor**, donde tanto el **cliente** (el navegador, explorador o visualizador) como el **servidor** (el servidor web) y el protocolo HTTP mediante el cual se comunican. El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son empleados en Internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio entre distintos ordenadores [2].

Características

El cliente web es un programa con el que interactúa el usuario para solicitar a un servidor *web* el envío de recursos (páginas *web*, imágenes, sonidos, animaciones, etc). El envío de estos recursos se realiza a través del protocolo **HTTP**. El cliente de las aplicaciones web suele estar formada por código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de *script* del navegador (*JavaScript* o *VBScript*) o mediante pequeños programas (*applets*) realizados en *Java*. Por lo tanto, la misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (imágenes, sonidos, etc.).

El servidor web es un programa que está esperando permanentemente las solicitudes por parte de los clientes *web*. El servidor de las aplicaciones *web* está formada por:

- Páginas estáticas (documentos HTML) que siempre muestran el mismo contenido.
- Recursos adicionales (audio, video, imágenes, animaciones) que se pueden emplear dentro de las páginas o estar disponibles para ser descargados y ejecutados en el cliente.
- Programas o scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas páginas. La salida de este script suele ser una página HTML estándar que se envía al navegador del cliente.

Patrones de diseño

Basado en el libro[3]. Los patrones de diseño son descripciones de clases y objetos relacionados que están particularizados para resolver un problema de diseño general en un determinado contexto. En la implementación de LIBREP se emplearon los siguientes patrones de diseño: Estrategia, Fachada y Modelo-Vista-Controlador (MVC); además de los principios básicos de la programación orientada a objetos, a saber: encapsulamiento, polimorfismo, herencia, entre otros.

Estrategia (*Strategy*). Define una familia de algoritmos, encapsula cada uno, y los hace intercambiables. Estrategia permite que el algoritmo varíe de forma independiente de los clientes que lo utilizan.

Fachada (*Fachade*) Considera un subsistema complejo para el cual se define una fachada; es decir, se define una interfaz que facilita la comunicación con el subsistema.

Modelo-Vista-Controlador (*Model-view-Controller*). Es considerado un patrón de patrones porque involucra tres patrones de diseño: Estrategia, Observador y Composición [4].

6. Desarrollo del proyecto

6.1 Metodología empleada en el desarrollo del proyecto

El Proceso Unificado (UP) se ha convertido en un proceso de desarrollo de software de gran éxito para la construcción de sistemas orientados a objetos. El UP promueve un desarrollo iterativo e incremental. El UP se organiza en iteraciones de duración fija, por ejemplo, cuatro semanas. Cada iteración aborda nuevos requisitos y amplía el sistema incrementalmente [5].

Para el desarrollo de LIBREP se definió una iteración con 2 semanas. En la fase de inicio realizaron dos iteraciones (I1, I2) en la fase de elaboración 3 iteraciones (E1, E2, E3), en la fase de construcción 6 iteraciones (C1-C6) y en la fase final transición 3 iteraciones (T1, T2, T3).

Inicio: Visión aproximada e inicio del documento de visión, análisis de la lógica del negocio, y definición del alcance del proyecto. Planificación de actividades de acuerdo al calendario escolar, e inicio de la documentación. Entrevistas con el cliente, y bosquejo de pantallas del sistema.

Elaboración: Refinamiento del documento de visión, definición de la arquitectura basada en patrones de diseño de software. Identificación de otros requisitos, y asignación de prioridad a los requisitos. Análisis iterativo a través de la especificación de los casos de uso de texto para los requisitos. Diseño e implementación de la base de datos preliminar, implementación de algunos módulos sencillos, y su documentación.

Construcción: Diseño e implementación definitiva de la base de datos, diseño e implementación iterativa de los módulos para el resto de los requisitos, y preparación para el despliegue.

Transición: Pruebas unitarias, integración de los módulos, y pruebas de sistema. Retroalimentación de la documentación, análisis de resultados en base a los objetivos propuestos. Identificación de limitantes del sistema y las posibles soluciones.

6.2 Diseño del sistema

A continuación se presentan los artefactos de diseño: diagrama de casos de uso, casos de uso de texto, y diagrama de clases, los cuales sirvieron de base para el desarrollo de LIBREP.

6.2.1 Diagramas de casos de uso

El sistema LIBREP cuenta con cinco módulos: 1) gestionar usuarios, 2) gestionar solicitudes de libros, 3) gestionar solicitudes de artículo de papelería, 4) gestionar pedidos de libros o artículo de papelería y 5) gestionar cartera de distribuidores, como lo muestra el diagrama de casos de uso de la figura 4.

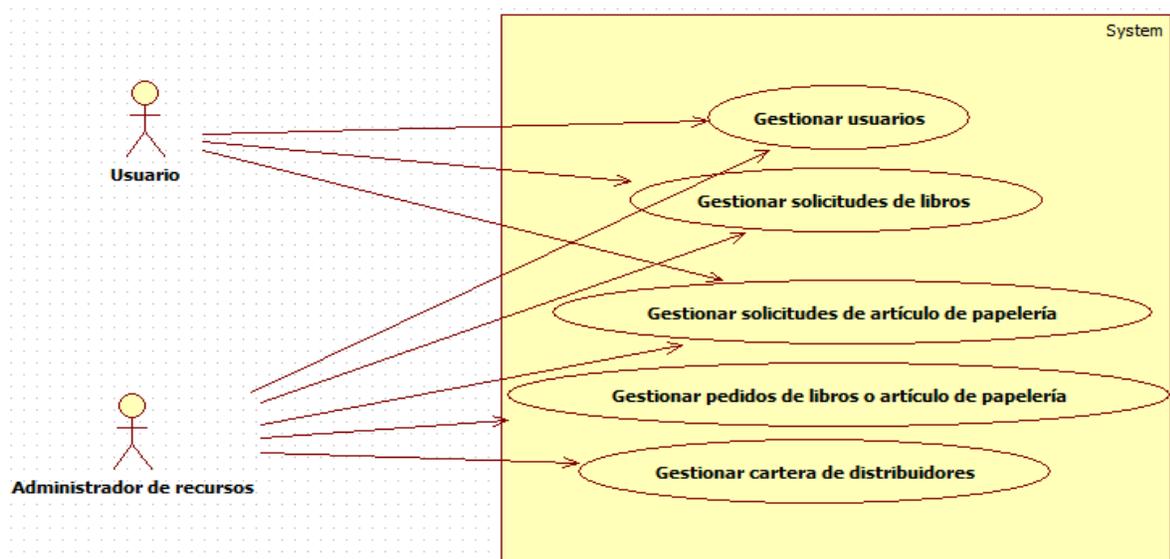


Figura 3. Diagrama de casos de uso general.

Tomaremos el caso de uso Gestionar solicitudes de libro para ejemplificar, a lo largo del documento, las diferentes etapas del diseño. Los detalles del resto de los casos de uso de la figura 3 se encuentran en la memoria de diseño del sistema LIBREP.

Gestionar solicitudes de libro. Implica 4 sub-módulos: Registrar solicitud de libro, modificar solicitud de libro, eliminar solicitud de libro y revisar solicitud de libro, verFigura 4.

1. **Registrar solicitud de libro.** Será posible si y sólo si el usuario se encuentra registrado. Si una solicitud es idéntica a una que ya está registrada, simplemente contará como un voto y dicho usuario se incorporará a dicha solicitud.
2. **Modificar solicitud de libro.** Será posible modificar una solicitud, si ésta hace referencia a un libro inexistente. Para ello se deberá proporcionar la clave única de registro del libro.
3. **Eliminar solicitud de libro.** Será posible eliminar una solicitud, si ya fue pedido por parte el administrador de recursos; o bien, el registro no cumple con lo esperado. Para ello se debe proporcionar la clave única de registro del libro.
4. **Revisar solicitudes de libro.** Incluye priorizar solicitudes, listar usuarios con la misma solicitud y listar solicitudes por editorial. **Identificar solicitudes:** Se podrán identificar aquellas solicitudes elaboradas por profesores y alumnos, para ello se emplearán colores. El sistema mostrará el número económico de los profesores en color rojo, y la matrícula de los alumnos en color verde. **Listar usuarios con la misma solicitud:** Muestra el número económico ó la matrícula de los usuarios que han hecho la misma solicitud. **Listar por editorial:** Mostrará todas las solicitudes que coincidan con la editorial, además de especificar la cantidad de solicitantes. Será posible seleccionar solicitudes para dar de alta un pedido.

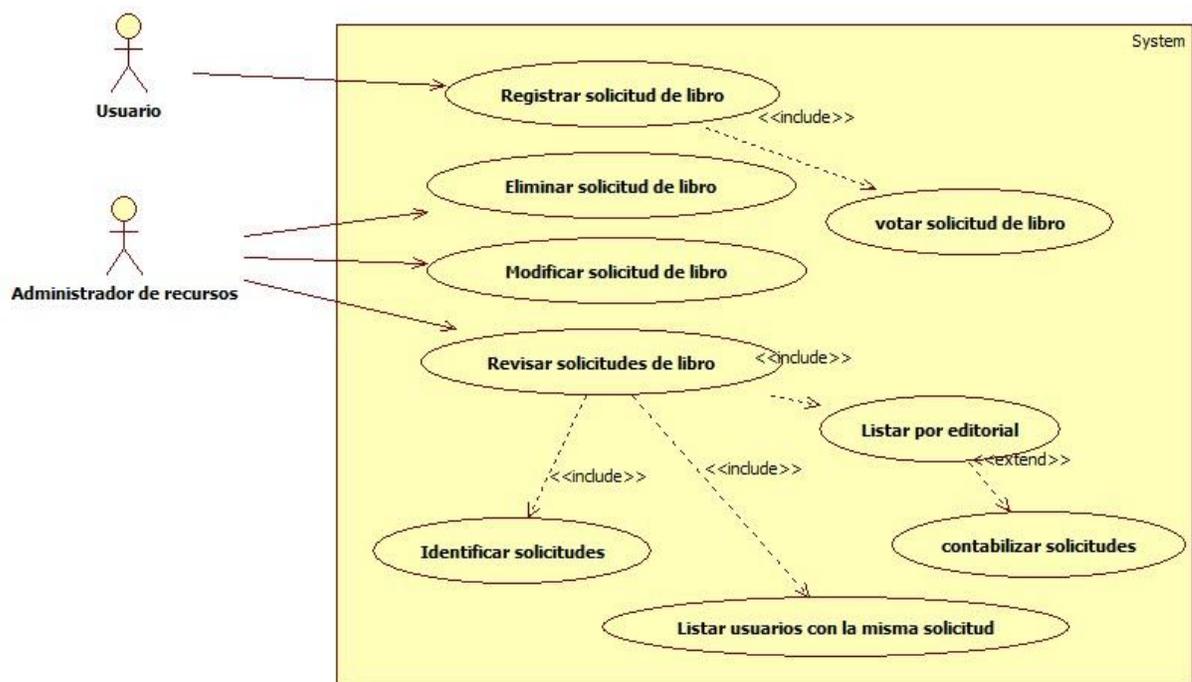


Figura 4. Diagrama de casos de uso “Gestionar solicitudes de libros”.

6.2.2 Casos de uso de texto

1. Caso de uso CU1: Solicitar libro

2. Descripción del caso de uso: Se requiere hacer una solicitud de libro, cuando éste no está disponible en la librería. Esta solicitud la realiza todo usuario interesado en la compra de un libro en particular.

3. Actores

Actor primario: Usuario de la librería de la UAM-A.

Usuario: quiere solicitar un libro específico, también quiere saber si ese libro fue pedido anteriormente con el propósito de incorporar esa solicitud a una ya existente, en lugar de generar una nueva.

Personal involucrado e intereses (Actores secundarios):

Administrador de recursos: quiere conocer el título, autor, editorial y edición del libro cuya demanda por parte de los usuarios, sea suficiente para hacer un pedido al distribuidor correspondiente.

4. Disparador: el usuario selecciona la opción de *Solicitud*

5. Precondiciones: Estar en la página principal del sistema LIBREP, y estar autenticado.

6. Pos condición: el usuario queda registrado junto con su solicitud en espera del estado de su libro (fecha próxima a llegar o información de existencia) notificado vía correo electrónico.

Garantías de éxito (postcondiciones): i) Se registra la solicitud del libro; ii) se ordenan conforme a demanda o votación del profesor; automáticamente, iii) se muestra el número de alumnos solicitando ese libro; iv) se manda al usuario un aviso (vía correo electrónico) sobre el estado de su libro, a saber: pedido en proceso ó pedido en espera.

7. Escenario principal de éxito (o flujo Básico):

1. El usuario ingresa al sistema LIBREP proporcionando Matrícula (alumnos) o número económico (profesores) y contraseña. Posteriormente se muestra un menú con la siguiente información:
 - *Solicitud libro
 - *Solicitud artículo de papelería
2. El usuario selecciona una opción *Solicitud libro*.
3. LIBREP muestra la siguiente información:
 - *Titulo
 - *Autor
 - *Editorial
 - *Edición
 - *Año
 - *ISBN
4. El usuario ingresa la información del formulario.
5. LIBREP proporciona un autocompletado para cada campo, proporcionando las solicitudes que tiene ya registradas.
6. El usuario presiona *Enviar*.
7. LIBREP valida los datos ingresados.
8. LIBREP guarda los datos ingresados.
9. LIBREP muestra un mensaje al usuario de Éxito y un recordatorio: En breve recibirás un correo electrónico informándote que el libro está disponible en la librería.
10. LIBREP automáticamente proporciona la información capturada al administrador de recursos.

8. Flujo Alternativo

1. El usuario ingresa un identificador no válido
 - 1a. LIBREP señala el error ya sea en nombre de usuario o contraseña y rechaza la entrada.
 - 1b. LIBREP muestra un ejemplo en cada campo.
- 3.El usuario no ingresa algunos campos obligatorios
 - 3a.LIBREP marca un error e indica los campos obligatorios faltantes.
 - 3b.El usuario ingresa los campos faltantes.
4. El usuario comete un error al ingresar un campo
 - 4a. LIBREP marca el error y muestra un ejemplo sobre el formato del campo.
 - 4b.El usuario ingresa los campos correctamente.
7. No es posible guardar los campos ingresados
 - 7a. LIBREP muestra un mensaje sobre el error y pide al usuario intentar más tarde o comunicarse con el administrador del sistema, mostrando el correo de quejas.

Requisitos no funcionales:

Interfaz del administrador de recursos: el texto debe ser visible a un metro de distancia se mantendrá una paleta de 4 colores.

Internacionalización del lenguaje del texto.

Mantener a la vista el inicio de sesión del usuario.

Interfaz del usuario: mantener una paleta de 4 colores. Cuando una solicitud tiene alta demanda mostrarla en color rojo. Cuando una solicitud tiene baja demanda mostrarla en color verde.

Tiempo de envío de la solicitud: 30 segundos.

Para solicitar un artículo de papelería se requiere el mismo procedimiento que para el caso de uso de texto Solicitar libro, solo en el punto (2a) la información será la siguiente: descripción del artículo, marca, especificaciones, y cantidad sugerida.

6.2.3 Modelo de dominio general.

El modelo de dominio para LIBREP se muestra en la figura 5, donde las principales entidades son: Usuario, RegistroLibro, RegistroLibroespec, SolicitudLibro, PedidoLibro, RegistroArticulo, RegistroArticuloespec, SolicitudArticulo, PedidoArticulo, Distribuidor y catalogoDistribuidor.

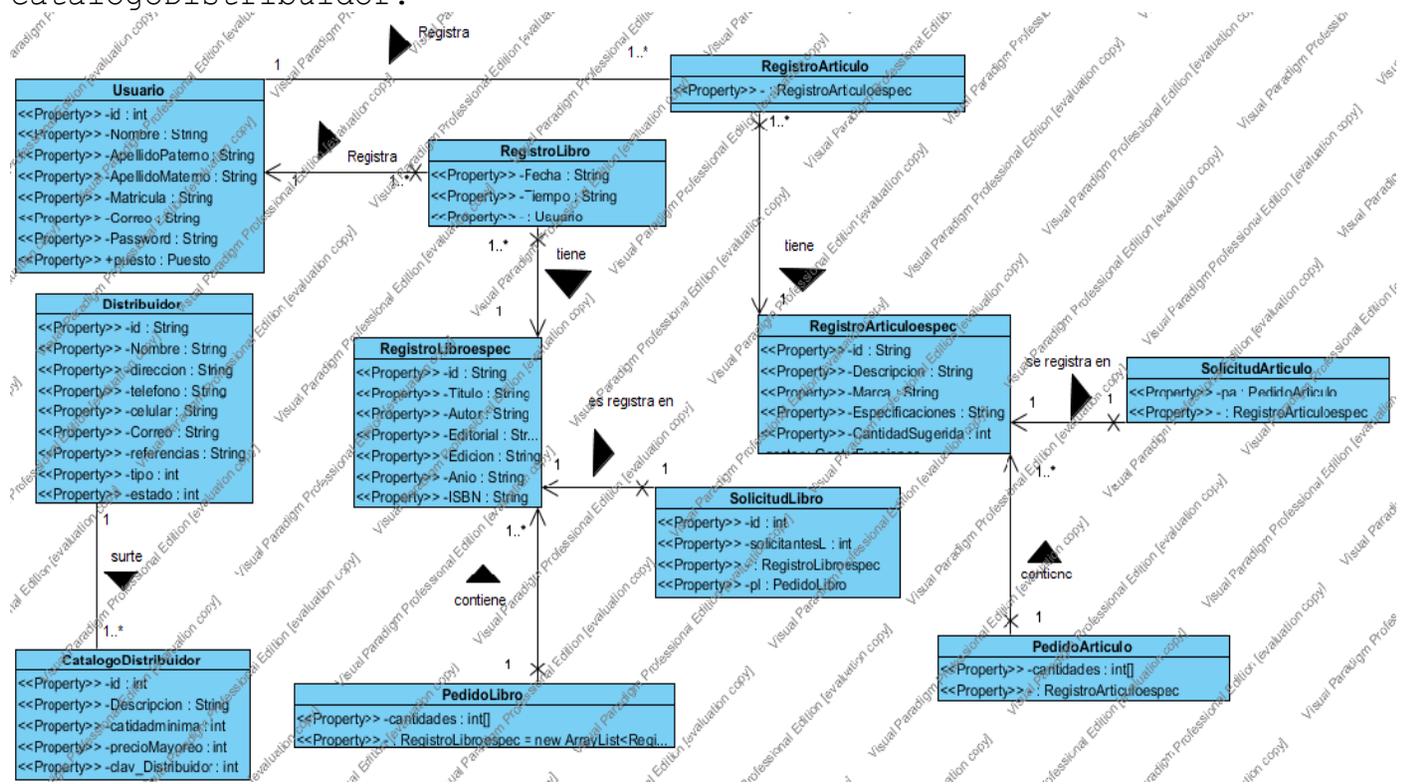


Figura 5. Modelo de dominio general.

La entidad Usuario registra una o varias especificaciones de libro. La entidad RegistroLibro registra: la fecha, hora, el usuario y la especificación del libro capturada por el usuario. SolicitudLibro lleva el control sobre el número de solicitantes para el libro especificado por el usuario. PedidoLibro registra: fecha, hora, el usuario y una lista con las especificaciones de libros con sus respectivas cantidades. Distribuidor registra la información básica de un distribuidor. CatalogoDistribuidor mantiene una descripción breve del material que distribuye, la cantidad mínima que surte y si maneja mayoreo. Es importante señalar que la misma información se recupera para la entidad artículo, pero en su propio conjunto de entidades, a saber: RegistroArticulo, RegistroArticuloespec y SolicitudArticulo.

7. Arquitectura del sistema

LIBREP presenta una arquitectura cliente-servidor: cliente ligero, ver figura 6.

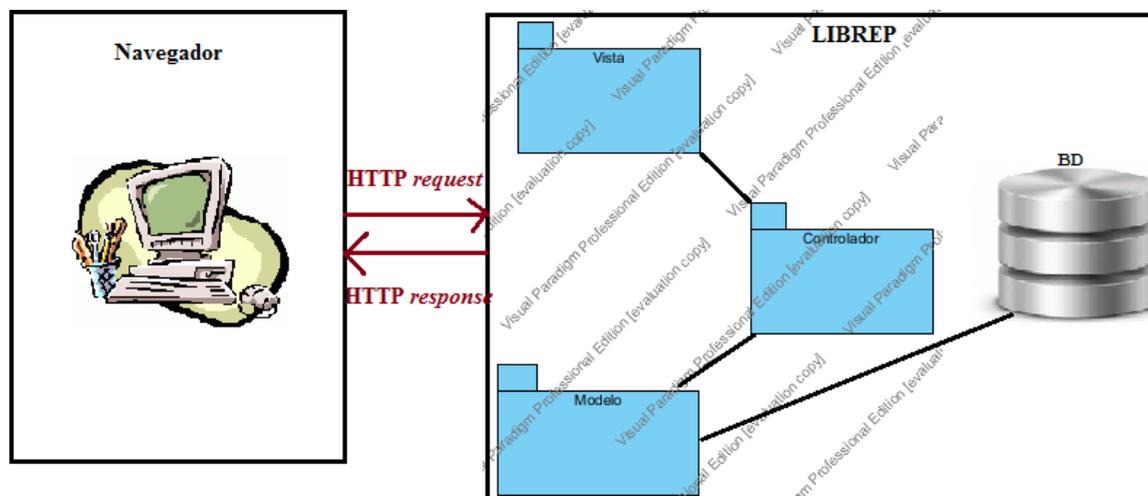


Figura 6. Arquitectura de aplicación web: Cliente ligero.

LIBREP está organizado en base al patrón MVC(Modelo-Vista-Controlador), enseguida se describen sus principales componentes.

Modelo: Contiene los datos, el estado y la lógica de la aplicación LIBREP. Esta capa se divide en:

- ✓ Datos, estado y lógica: Formado por las clases de software correspondientes a las entidades definidas en el modelo de dominio.
- ✓ Capa de datos: Contiene todas las clases de software que se encargan de gestionar la interconexión del SGBD(Sistema de Gestión de Bases de Datos) con las clases que contienen los datos, el estado y la lógica de LIBREP. Las clases de software son nombradas *GestorentidadBd* o *EntidadCRUD*, donde la *entidad* corresponde al nombre definido en el modelo de dominio. Estas clases permiten hacer las operaciones CRUD a una *entidad* determinada.
- ✓ Helpers: Contiene todas las clases de software que apoyan tanto al controlador como a la vista para agilizar algunas tareas. En LIBREP las clases de software involucradas ayudan a las vistas a cargar *select's* dinámicos; por ejemplo, obtener las editoriales de libros cuando un usuario registra una solicitud. para más detalles ver la memoria de diseño del proyecto.

Controlador: Contiene todas las clases de software que permiten gestionar todas las peticiones, validar y dirigir cualquier petición de cualquier tipo. En LIBREP los controladores están organizados conforme a una *entidad* definida en el modelo de dominio, es decir cada petición: registrar, modificar, eliminar o consultar que involucre a una *entidad* está definida en un paquete nombrado *controlador.entidad*.

Vista: Contiene todas las clases que conforman la interfaz gráfica del usuario final. Los archivos *jsp* están ubicados una carpeta llamada *Vista*.

7.1 Diagrama de paquetes y diagrama de clases

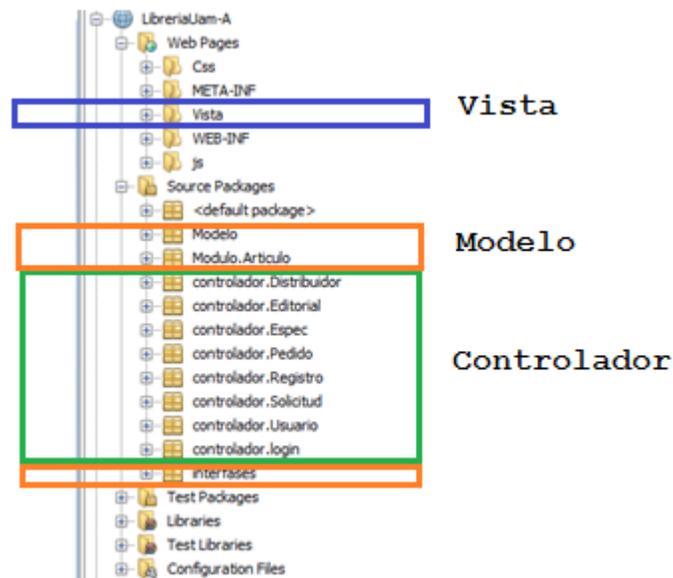


Figura 7. Estructura de la aplicación LIBREP

La aplicación LIBREP. está compuesta por once paquetes (ver figura 7) distribuidos de la manera siguiente:

Modelo. Formado por tres paquetes: `Modelo`, `Modulo.Articulo` e `Interfases`. El paquete `Modelo` y `Modulo.Articulo` tiene las clases definidas en el modelo del dominio ver Figura 5, también contiene las clases que conforman la capa de datos encargadas de realizar las operaciones CRUD y finalmente los helpers. `Interfases` solo contiene la interfaz `Puesto` que se utiliza en el patrón *Strategy*.

Controlador. Formado por ocho paquetes, En particular, para el caso de uso solicitar libro, el paquete `controlador.Espec` contiene las clases para el manejo de las peticiones: registrar, modificar, consultar y eliminar, para la entidad `RegistroLibroespec`. Esta organización es similar para los siete paquetes restantes, pero para una entidad diferente. **Vista.** La carpeta `Web pages`, contiene la carpeta `vista`, donde se encuentran los archivos `jsp` de LIBREP.

Diagrama de clases del paquete: **Modelo**

En la figura 8 se muestra una parte del Modelo concretamente el Modelo relacionado con el caso de uso `Solicitar libro`.

Basados en el caso de uso Solicitar libro, las clases involucradas son: Registro, RegistroLibro, RegistroLibroespec, Solicitud y SolicitudLibro, ver Figura 8.

RegistroLibroespec es una clase que toma toda la información proporcionada por el usuario cuando solicita un libro desde una computadora ó terminal. Esta información está compuesta por: *Título, autor, editorial, edición, año e ISBN*. Si la información capturada existe en LIBREP, la clase SolicitudLibro incorporará un voto al registro ya establecido.

Registro es una clase abstracta que tiene como atributos: *fecha, hora*, y una instancia de la clase *Usuario*. Esta información la obtiene cuando un usuario registra una solicitud de libro.

RegistroLibro es una clase que hereda de Registro, además de los atributos que hereda, tiene una instancia de la clase RegistroLibroespec. Esta clase permite saber la *fecha y la hora*, la descripción del libro y el *usuario* que realizó la solicitud.

Solicitud es una clase abstracta que tiene como atributos: su identificador único y la cantidad de solicitantes. Esta clase nos sirve para obtener la cantidad de usuarios que están solicitando el mismo libro.

SolicitudLibro es una clase que hereda de Solicitud, además de los atributos que hereda, tiene una instancia de la clase de RegistroLibroespec, esta clase permite saber la demanda que tiene un libro, de tal manera que impide que el sistema registre múltiples solicitudes de un mismo libro.

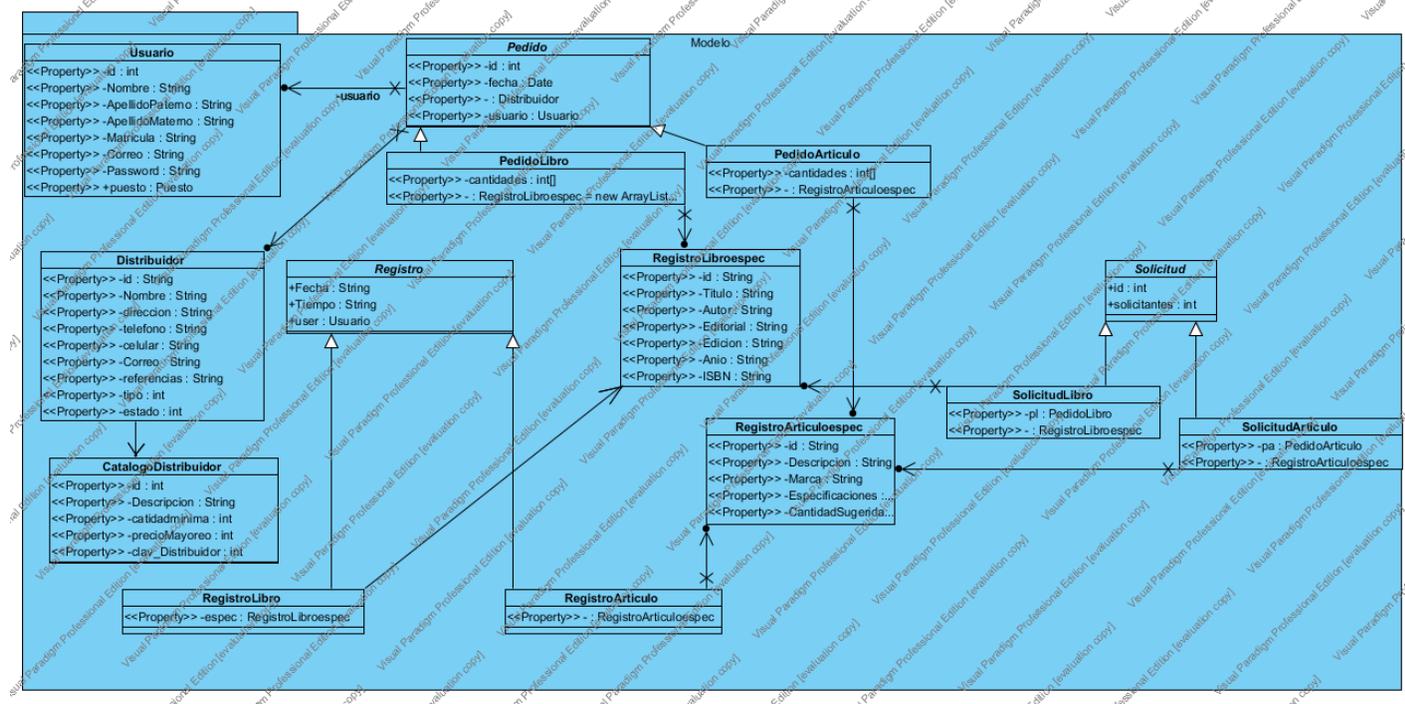


Figura 8. Modelo

La figura 9 muestra las clases gestoras: GestorRegistroLibroBD, GestorRegistroLibroespecBD, GestorSolicitudLibroBD. Los métodos principales de estos gestores son: *agregar, cargar, eliminar, buscar y modificar*; estos métodos actúan respectivamente sobre las entidades: *RegistroLibro, RegistroLibroespec*

SolicitudLibro.EstablecerConexion es la encargada de realizar la conexión a la base de datos de LIBREPa través del método Enlace .

Es importante mencionar que existe una clase gestora para cada una de las entidades de la figura 5.

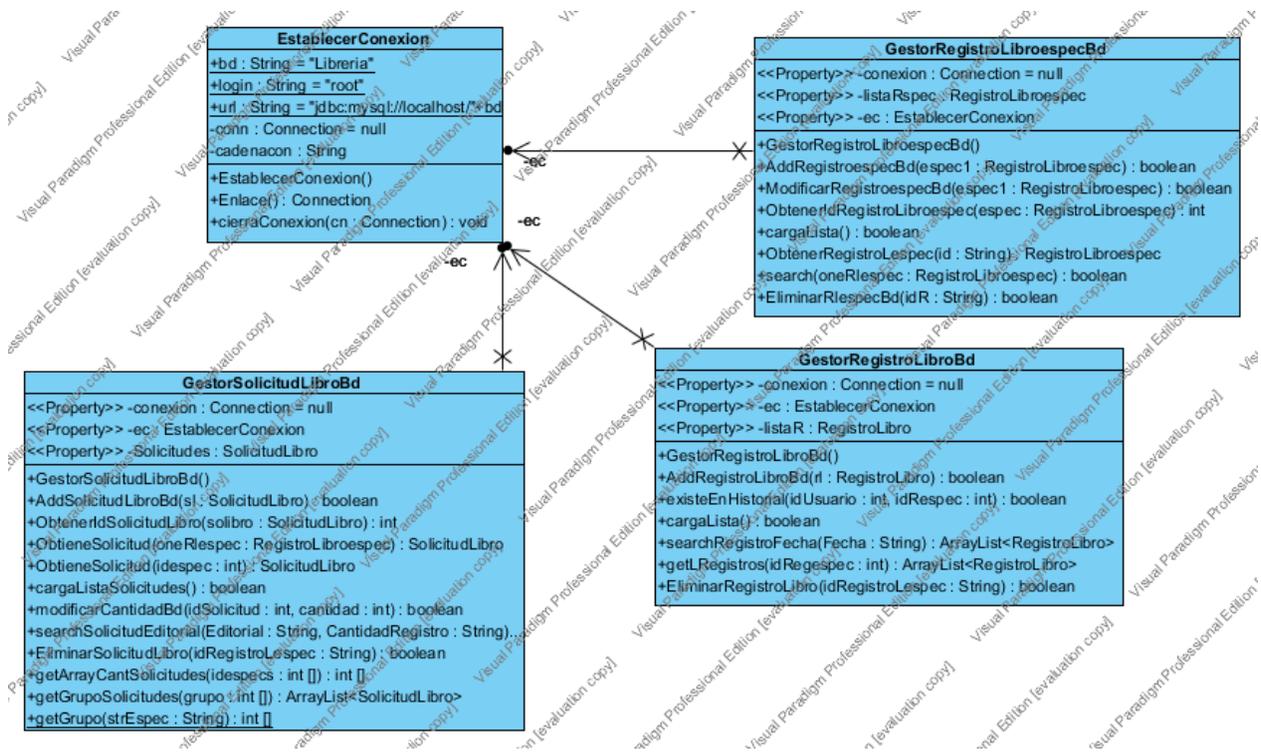


Figura 9. Clases gestoras de las operaciones *CRUD* libro.

Diagrama de clases del paquete: **controlador.Espec**

En este paquete se encuentran las clases controladoras: *ConsultaRlespec*, *EliminarRlespec* y *ModificarRlespec*, cuyos métodos actúan sobre la entidad *RegistroLibroespec*, ver Figura 10. El controlador *ConsultaRlespec* obtiene una lista de registros de tipo *RegistroLibroespec* que contienen: *Título*, *autor*, *editorial*, *edición*, *año* e *isbn*, filtrando los registros a través del nombre de editorial. El controlador *EliminarRlespec* permite borrar una instancia u objeto de tipo *RegistroLibroespec* proporcionando su identificador único. El controlador *ModificarRlespec* realiza la actualización de cualquier atributo, a saber: *Título*, *autor*, *Editorial*, *edición*, *año* e *isbn* de una entidad *RegistroLibroespec*. Es importante mencionar que estas clases controlador extienden de la clase *ActionSupport*, ya que LIBREP utiliza la Framework *Struts2*.

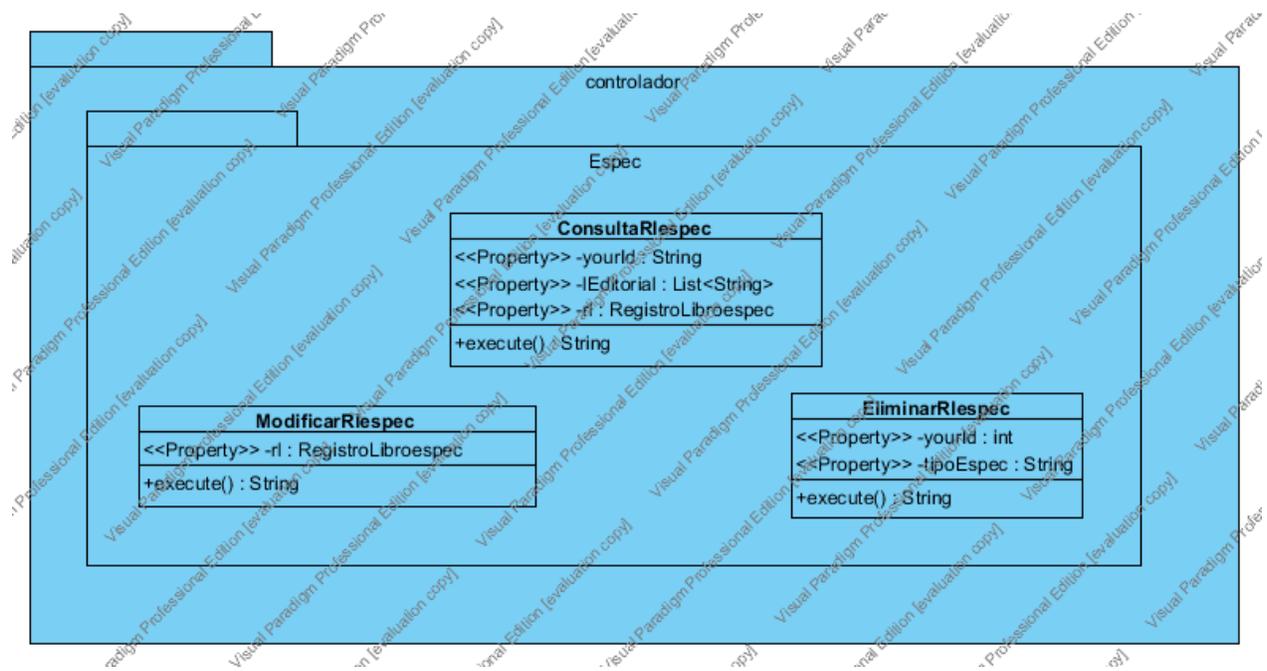


Figura 10. controlador.Espec

Diagrama de clases del paquete: controlador.solicitud

En este paquete se encuentra la clase controladora: Busqueda, de la entidad SolicitudLibro, ver Figura 11. El controlador Búsqueda obtiene una lista de los registros de tipo SolicitudLibro que contienen; *Título, autor, editorial edición, año e isbn* y la cantidad de solicitantes, filtrando los registros a través del nombre de la editorial y la demanda (cantidad de solicitudes).

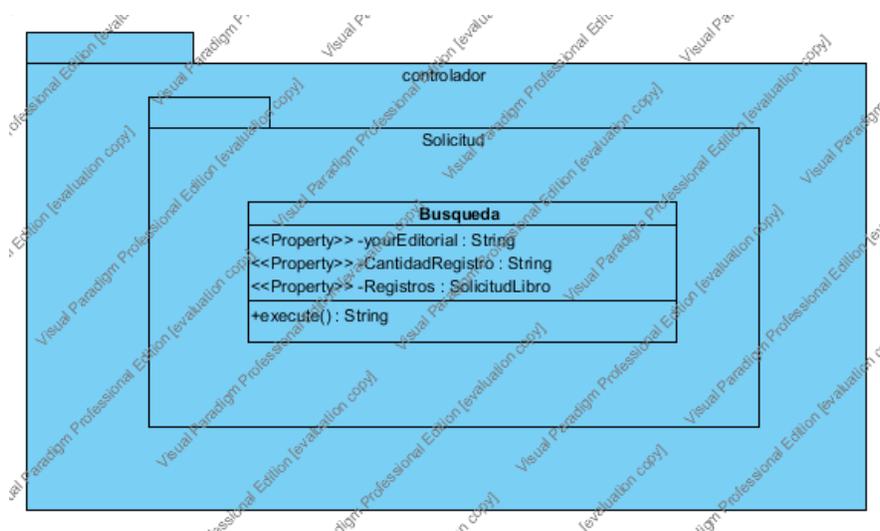


Figura 11. Controlador.Solicitud

Diagrama de clases del paquete: `controlador.registro`

En este paquete se encuentran las clases controladoras: `RegistroLibroAlumno` y `ModificarRegistrosF` de la entidad `RegistroLibro`, ver Figura 12. El `controladorRegistroLibroAlumno` permite hacer la inserción de una entidad `RegistroLibroespec`: *Título, autor, editorial, edición, año e isbn* empleando el patrón fachada. La inserción de la entidad `RegistroLibroespec` se aloja en las tablas: `RegistroLibroespec`, `SolicitudLibro` y `RegistroLibro` de la base de datos de LIBREP descritas en la sección 7.3. El `controladorModificarRegistrosF` obtiene una lista de registros de libro almacenados en LIBREP, cada registro es del tipo `RegistroLibro`, con la información siguiente: *Fecha, hora, usuario, título, autor, editorial, edición, año e isbn*. Esta lista se obtiene proporcionando una fecha.

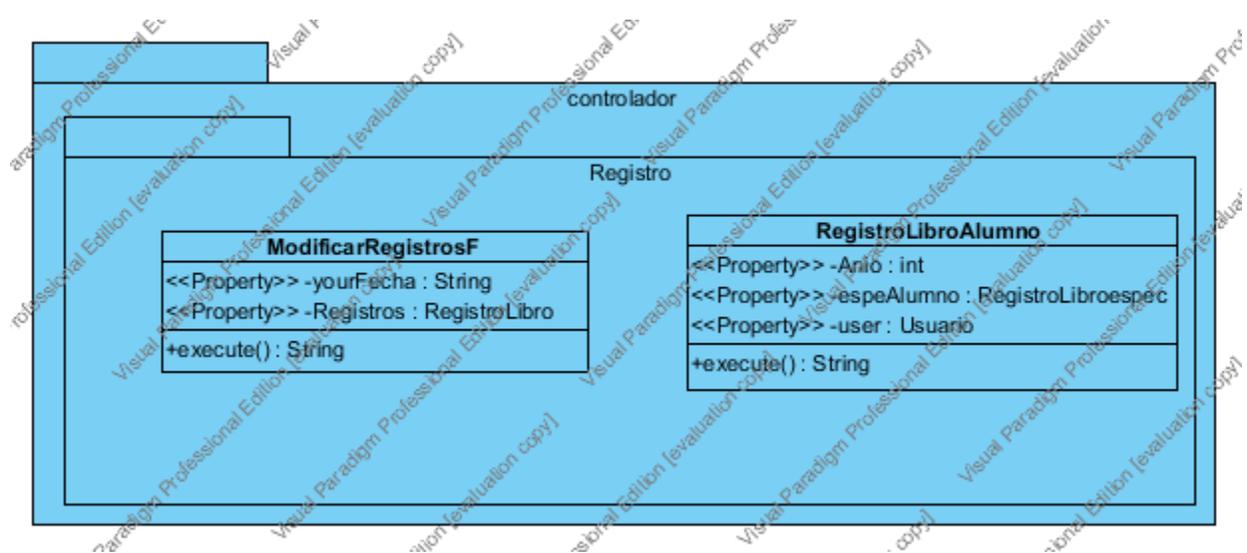


Figura 12. `Controlador.Registro`

7.2 Patrones de diseño empleados en la aplicación LIBREP

Uno de los patrones que se emplearon en las clases de software que conforman el Modelo es el patrón estrategia (*Strategy*). Por ejemplo, la Figura 13 muestra las clases: `Usuario`, `Puesto`, `Puesto_Administrador`, `Puesto_Alumno`, `Puesto_Profesor`, donde las clases `Puesto_Administrador`, `Puesto_Alumno` y `Puesto_Profesor` implementan los métodos de la interface `Puesto`. Esto es así porque la propiedad o atributo `puesto` de la clase `Usuario` puede variar.

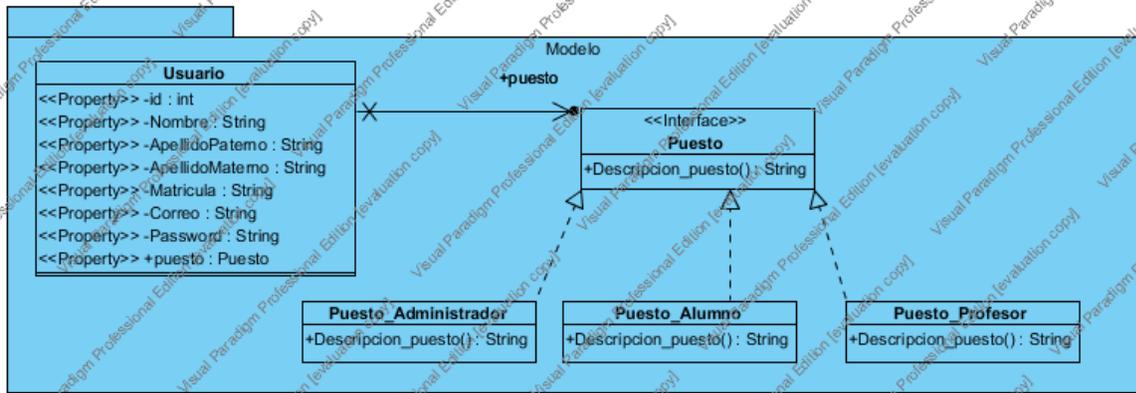


Figura 13. Patrón Strategy en el Modelo.

Otro patrón de diseño que se empleó fue fachada (*facade*), en el controlador Registrardel paquete controlador.Registro. En la figura 14 se presentan las clases: GestorFacadeRegistroLibro, GestorRegistroLibroespecBd, GestorRegistroLibroBdy GestorSolicitudLibroBd. GestorFacadeRegistroLibro a través del método InsertarRegistroLibro inserta la especificación de un libro y este, a su vez, invoca al método AddEntidadBd de las clases que tiene referencia, cada clase involucra una entidad a saber: RegistroLibroespec, SolicitudLibro y RegistroLibro. Las cuales serán alojadas en la base de datos de LIBREP después de ejecutar el método.

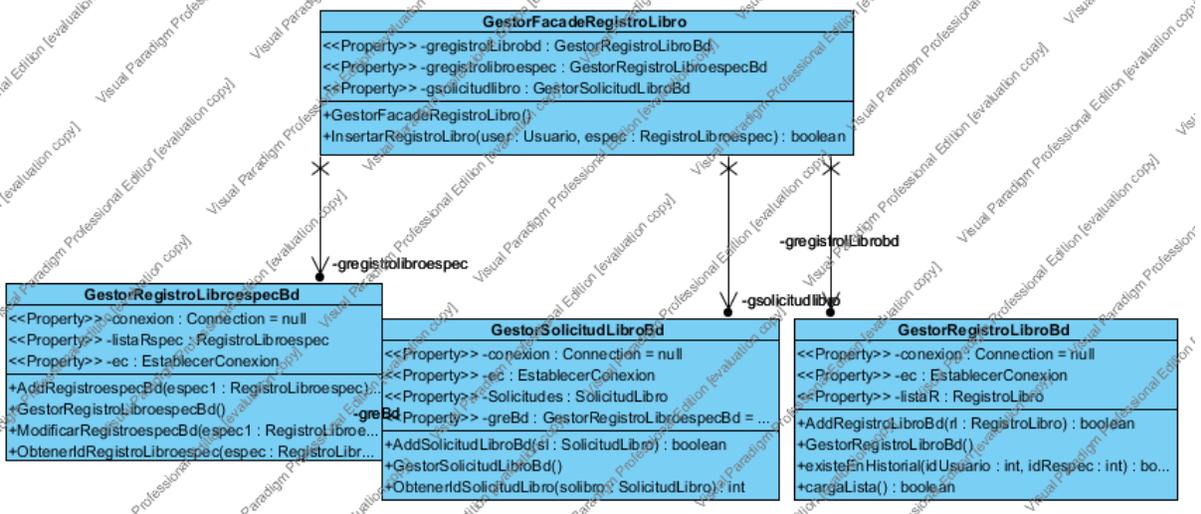


Figura 14. Patrón facade al agregar una especificación de libro.

7.3 Estructura de la base de datos.

La figura 15 muestra el modelo relacional para la base de datos del sistema LIBREP, la cual está compuesta por las tablas: Usuario, Puesto, RegistroLibro, Registro- Articulo, SolicitudLibro, SolicitudArticulo, RegistroLibroespec, RegistroArticuloespec, Distribuidor, CatalogoDistribuidor, PedidoLibro, PedidoArticulo, pedidoA_espec, pedido_espec y Editorial.

Dado que la comunidad universitaria puede utilizar LIBREP, la entidad Usuario aloja los datos de cada uno de ellos. Se eligió a Id_Usuario como *PRIMARY KEY*. Cada usuario que hace uso de la aplicación tiene un identificador único; aunque en la aplicación también se utiliza su matrícula ó su número económico, ya que estos son irrepetibles. La entidad Usuario tiene tres relaciones con las entidades siguientes: Puesto, RegistroLibro y RegistroArticulo. La primera establece que un usuario tiene un puesto ó rol en el sistema, a saber: alumno, profesor o administrador. La segunda y la tercera establecen que un usuario puede solicitar uno o varios libros o artículos que no se encuentran disponibles en Librería.

La tabla RegistroLibroespec aloja una descripción de un libro capturada por el usuario. En ella cualquier entrada es única, es decir no existen registros duplicados. Tiene cuatro relaciones con las entidades: Editorial, RegistroLibro, SolicitudLibro y pedido_espec. La primera establece que un libro tiene una editorial. La segunda establece que la descripción de un libro fue registrada por un usuario. La tercera establece que un libro tiene una cantidad específica de solicitantes. La cuarta establece que la descripción de un libro forma parte de uno o varios pedidos.

La tabla RegistroLibro incorpora fecha, hora y el usuario que capturó la descripción de un libro. Se eligió a Id_RegistroLibro como *PRIMARY KEY*. Cada usuario que registra la descripción de un libro es almacenado en esta tabla.

La tabla SolicitudLibro aloja los datos de la demanda que tiene un libro. Se eligió a Id_SolicitudLibro como *PRIMARY KEY*. Un libro que ya fue capturado como parte de una solicitud no volverá a registrarse, cada solicitud adicional se incorporará a la misma solicitud, de tal manera que será posible identificar los libros con mayor demanda.

La tabla PedidoLibro aloja los datos básicos de un pedido, incluyendo: el identificador del usuario quien lo elabora y el identificador del distribuidor a quien va dirigido. Se eligió a Id_PedidoLibro como *PRIMARY KEY*. Cada pedido está compuesto de uno o varios libros con sus respectivas descripciones, los cuales se alojan en la tabla pedido_espec. Este registro permite saber, además, la fecha y hora del registro.

Dado que cualquier descripción de libro almacenada en la tabla RegistroLibroespec puede ser incorporada a un pedido, en la tabla pedido_espec se podrá incorporar el *PRIMARY KEY* de

la tabla PedidoLibro y la cantidad. Esta tabla permite conocer las descripciones de los libros y sus respectivas cantidades para un pedido.

Las tablas Registroarticuloespec, RegistroArticulo, SolicitudArticulo, PedidoArticulo y PedidoA_espec, operan de la misma manera que las descritas para RegistroLibroespec, solo que la descripción del artículo contiene: Descripción, marca, especificaciones del artículo y cantidad sugerida; donde cada registro es alojado en la tabla RegistroArticuloespec.

La coordinación de la sección de Librería tiene varios distribuidores de libros o artículos, por lo cual la entidad Distribuidor aloja la información básica de cada uno de ellos. Se eligió a id_Distribuidor como *PRIMARY KEY*. Cada distribuidor tiene un tipo de distribución: libros o artículos. Distribuidor tiene una relación a la entidad CatalogoDistribuidor, porque cada distribuidor puede ofrecer uno o varios catálogos.

CatalogoDistribuidor aloja los datos de un catálogo que ofrece un distribuidor. Se eligió id_Catalogodistribuidor como *PRIMARY KEY*. Cada registro describe un tipo de material que surte un distribuidor, se precisa si maneja mayoreo y la cantidad mínima que surte. Esta entidad permite obtener todas las descripciones de un catálogo que surte un distribuidor.

El Anexo A, presenta la descripción de los campos de cada una de las tablas que forman la base de datos.

7.4 Uso del sistema

El uso del sistema LIBREP se explicará a partir de caso de uso Gestionar solicitudes de libros. En particular, para un usuario (profesor o alumno): solicitar libro; para el usuario (Administrador de recursos): modificar solicitud de libro y revisar solicitud de libros.

Usuario (alumno o profesor): Solicitar libro

El usuario (alumno o profesor) ya autenticado en LIBREP observará un menú con dos opciones: *solicitud libro* y *solicitud articulo* (Figura 16); donde el usuario podrá registrar su descripción de un libro o artículo de papelería que no está disponible en la Librería UAM-Azcapotzalco. En este caso la descripción de un libro está compuesta por: *título, autor, editorial, edición, año e ISBN*, ver Figura 17.



Figura 16. Menú para alumnos.

Registrar solicitud de libro

Si el usuario escoge la opción *solicitar libro*, entonces se presenta un formulario como el de la figura 17, solicitando los datos para los campos: *título, autor, editorial, edición, año e ISBN*. Todos los campos con excepción de ISBN son obligatorios; es decir, no pueden ser vacíos.

Inicio sesión 1111111
[SALIR \(X\)](#)

Solicitudes de Librería UAM-Azcapotzalco

Registra tu libro

Verifica todos los campos antes de enviar.

(*)Campos obligatorios.

Título*:
 Autor*:
 Editorial*:
 Edición*: --Selecciona la edición--
 Año*:
 ISBN:

Dudas o comentarios enviar un correo: libreria@azc.uam.mx Tel: 55-90-00-99

Figura 17. Registro de libro

Cuando el usuario proporciona el título del libro, LIBREP proporciona una lista con nombres de libros ya registrados por otros usuarios. Al seleccionar el título del libro, aparecerán el resto de los datos correspondientes a los campos del formulario; de esta manera, el usuario puede seleccionar uno de estos registros, sin escribir el nombre completo, ver Figura 18.

Inicio sesión 1111111
[SALIR \(X\)](#)

Solicitudes de Librería UAM-Azcapotzalco

Registra tu libro

Verifica todos los campos antes de enviar.

(*)Campos obligatorios.

Título*:
 Autor*:
 Editorial*:
 Edición*:

 Año*:
 ISBN:

Dudas o comentarios enviar un correo: libreria@azc.uam.mx Tel: 55-90-00-99

Figura 18. Selección de un registro

Si el libro no ha sido registrado anteriormente, el usuario debe ingresar todos y cada uno de los datos solicitados en el formulario, ver Figura 19. Es importante mencionar que la editorial se escoge de una lista de editoriales proporcionada por LIBREP. Una editorial está relacionada con uno o varios distribuidores registrados en LIBREP, de tal manera que un pedido de libros se realiza proporcionando la editorial y posteriormente se elige el distribuidor.

Solicitudes de Librería UAM-Azcapotzalco

Registra tu libro

Verifica todos los campos antes de enviar.

(*) Campos obligatorios.

Titulo*:
Autor*:
Editorial*:
Edición*:
Año*:
ISBN:

Enviar

Dudas o comentarios enviar un correo: libreria@azc.uam.mx Tel: 55-90-00-99

Figura 19. Agregar un registro.

El usuario verifica los datos ingresados y presiona enviar. Si no hay errores se muestra la Figura 20.

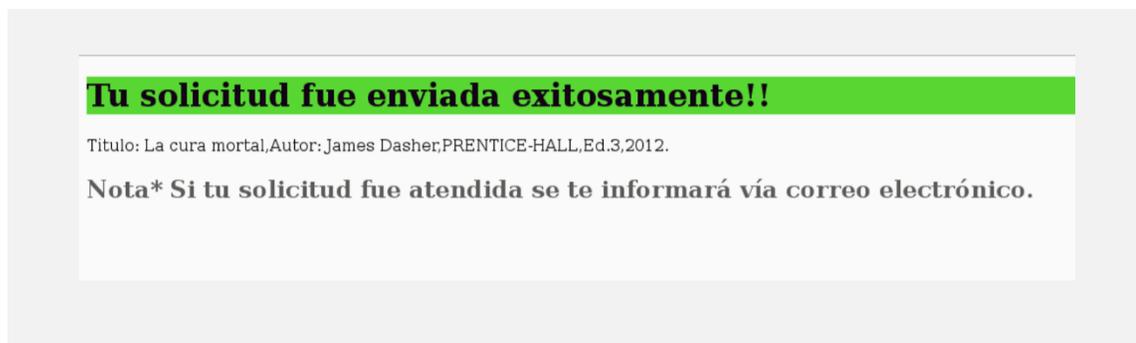


Figura 20. Mensaje de éxito al registrar un libro.

Si hubo un problema al agregar un registro se muestra un mensaje de error como el de la Figura 21. Este problema está relacionado con la conexión a la base de datos, o posiblemente la información de la descripción de un libro no se guardó de forma adecuada. En ambos casos, se recomienda repetir el proceso. Figura 17.



Figura 21. Mensaje de error.

Administrador de recursos: Modificar solicitud de libro y revisa solicitud de libros.

En la Figura 22, se muestra el menú de opciones para el administrador de recursos de la sección de Librería y papelería de la UAM-Azacapotzalco. El administrador de recursos debe verificar que su número económico aparece en la esquina superior derecha. Si no es así, el usuario debe presionar Salir (X) y autenticarse.



Figura 22. Menú del administrador

Si escoge Gestionar solicitudes, LIBREP le presentará un nuevo menú, como el de la figura 23, donde muestra 3 opciones: *modificar solicitudes*, *eliminar solicitudes* y *revisar solicitudes*.



Figura 23. Menú gestionar solicitudes de libros.

Ahora el usuario escoge Modificar solicitudes, entonces podrá modificar los campos de un registro de libro ingresado por un alumno o un profesor, a partir de proporcionar una fecha, ver Figura 24.



Figura 24. Modificar los registros en base a una fecha.

Los registros de libros son mostrados en base a la fecha ingresada en un formato (yyyy-mm-dd) año-mes-día por ejemplo 2014-11-25. Figura 25.



Figura 25. Listado de registros en base a una fecha

El administrador de recursos podrá observar los registros y el usuario (matrícula o número económico) quien lo solicitó. Si el usuario es un alumno, se muestra en color verde su matrícula, si fue un profesor se muestra en color rojo su número económico. Un registro pasará a modo edición, proporcionando su clave única de registro, ver Figura 26.



Figura 26. Modificar un registro en base a una clave de registro.

LIBREP muestra un mensaje de éxito indicando "se guardaron los cambios", de lo contrario envía un mensaje de error (ver Figura 21).

Si el administrador de recursos escoge consultar las solicitudes de libros, podrá visualizarlas en base a una editorial y una cantidad de solicitantes. Es decir, si se elige una editorial por ejemplo PRENTICE-HALL y se selecciona menores a 10, se mostrará sólo los registros con esa editorial y demanda menor a 10. Figura 27.

Inicio sesión 200
[SALIR \(X\)](#)

Solicitudes de Libreria UAM-Azcapotzalco

Buscar solicitudes por editorial

Elige la editorial: PRENTICE-HALL

Cantidad de solicitudes por editorial: Menores de 10

Buscar

[Regresar](#)

Figura 27. Buscar registros proporcionando editorial y demanda.

En la Figura 28 se muestran los registros que existen en LIBREP que cumplen con los criterios antes mencionados. En este punto, el administrador de recursos podrá seleccionar uno o varios registros para elaborar un pedido, o simplemente para verificar la clave única de un registro, sea para modificarlo o eliminarlo.

Inicio sesión 200
[SALIR \(X\)](#)

Solicitudes de Libreria UAM-Azcapotzalco

Editorial: PRENTICE-HALL

CLAVE UNICA	TITULO	AUTOR	EDITORIAL	EDICIÓN	AÑO	ISBN	CANTIDAD DE SOLICITANTES	PEDIDO
69	prueba de fuego	James Dasher	PRENTICE-HALL	1		9789876123549	1	<input type="checkbox"/>
71	La cura mortal	James Dasher	PRENTICE-HALL	3			1	<input type="checkbox"/>

Genera pedido

[Regresar](#)

Figura 28. Lista de registros.

Si no existen registros con la editorial y demanda proporcionada se muestra un mensaje como el de la Figura 29.

No hay registros

[Regresar](#)

Figura 29. Mensaje No hay registros con esa editorial y demanda proporcionada

8. Hardware y software necesario

8.1 Software

8.1.1 Tecnología del desarrollo de la aplicación

Netbeans 7.4 [6] con JRE 6. **IDE de desarrollo.** *Netbeans* es un entorno de desarrollo; es decir, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación.

Struts 2 [7]. **Framework MVC.** *Apache Struts* es un framework para el MVC, es de código abierto, y sirve para la creación de aplicaciones web Java.

8.1.2 Tecnología para la instalación y puesta en marcha de la aplicación

Apache Tomcat [8]. *Apache Tomcat* implementa las tecnologías *Java Servlet* y *JavaServer Pages*. Es de código abierto. Se optó por la versión 7.0.53 porque tiene algunas mejoras como: detección de fugas de memoria de aplicaciones Web y prevención y soporte para la inclusión de contenido externo directamente en una aplicación web.

MySQL 5 [9]. *MySQL* es un Sistema de gestión de bases de datos relacional, multihilo y multiusuario. Se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia.

JavaMail API 1.4.7 [10]. **Biblioteca para construir correos.** El API *JavaMail* proporciona una plataforma y un protocolo independientes para construir un correo y aplicaciones de mensajería. El API *JavaMail* está disponible como un paquete opcional para el uso con la plataforma Java SE y también se incluye en la plataforma Java EE.

iText [11]. **Biblioteca para crear y manipular archivos PDF.** Es una biblioteca de código abierto que permite crear y manipular documentos PDF.

Jquery 2.1.1 [12]. **Biblioteca JavaScript.** permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX para páginas web.

8.2 Hardware

Para el desarrollo de la aplicación LIBREP se empleó una computadora de escritorio con las siguientes especificaciones:

- Un procesador de 2.30 GHz
- 4 GB de memoria RAM.
- Disco duro de 650 GB.
- Un monitor de 14 pulgadas.
- Sistema operativo Debian Whezzy 7.3 32 bits.

9. Resultados

En este proyecto de integración se implementó la aplicación LIBREP compuesta por cinco módulos. A continuación se describe brevemente cada uno de ellos.

1. **Gestionar usuarios** permite: registro, modificación, eliminación y consulta de usuarios. Es importante mencionar que la identificación única de un usuario es por matrícula (alumnos) o número económico (profesores y trabajadores). De manera que que pertenecen a la UAM pueden hacer uso de LIBREP.
2. **Gestionar solicitudes de libro** permite: registro, modificación, eliminación y consulta de solicitudes de libros que han realizado los usuarios de la librería; además, el sistema impide la duplicación de solicitudes. Se pueden revisar las solicitudes con varios filtros entre ellos: editorial, fecha. También es posible observar indicadores la pantalla principal del administrador de recursos. Se muestra un indicador cuando una solicitud fue realizado por un profesor. Se muestra otro indicador cuando hay gran demanda en una solicitud de libro. Ambos indicadores son marcados en color rojo. Cuando se consultan las solicitudes de libros o de artículos, es posible seleccionar algunas ó todas ellas para realizar el pedido correspondiente de manera directa.
3. **Gestionar pedidos de libro o artículo de papelería** permite: registro, modificación, eliminación y consulta de pedidos tanto de libros como de artículos de papelería. En este momento, el módulo funciona únicamente para el caso de libros. Al generar un pedido, el administrador de recursos, puede revisar las solicitudes y observar su prioridad, puede modificar la cantidad de libros y agregar otras solicitudes a ese mismo pedido, y puede elegir al distribuidor del libro en cuestión. El pedido, una vez validado, puede ser transformado a formato PDF. Además, este módulo se encarga de enviar correos a los usuarios (profesores y alumnos) avisando que su solicitud fue atendida.
4. **Gestionar distribuidores** permite: registro, modificación, eliminación y consulta de la cartera de distribuidores de la librería. Un distribuidor puede surtir un catálogo de libros o de artículos de papelería. Este catálogo tendrá para cada producto una descripción y la cantidad mínima que surte. Este módulo permite elegir la mejor opción de un distribuidor para un pedido determinado.
5. **Gestionar solicitudes de artículo de papelería** permite: registro, eliminación y consulta de solicitudes de artículos de papelería. Se puede consultar las solicitudes por fecha. Además, se puede visualizar si una solicitud fue registrada por un profesor o por un alumno, de manera que el administrador de recursos pueda tomar la decisión de hacer el pedido.

10. Análisis y discusión de resultados.

Soluciones

LIBREP fue diseñado empleando el patrón MVC, lo cual nos ofreció algunas ventajas: i) LIBREP está implementado modularmente, y ii) sus vistas muestran en todo momento información actualizada. El patrón MVC nos proporcionó independencia entre las capas: vista, lógica del negocio y persistencia de datos. Realizar una modificación en la vista no afectó en absoluto a las otras capas.

Cada módulo que conforma a la aplicación LIBREP fue diseñado, implementado y sometido a pruebas unitarias. Posteriormente, los módulos fueron integrados y sometidos a pruebas de sistema. Para validar las entradas del usuario, se optó por hacer dos tipos de validaciones en la capa de la vista: i) usando expresiones regulares, y ii) usando *struts 2 validation-xml*.

En el módulo de gestionar pedidos de libros, y en particular registrar un pedido de libros se logró una manera sencilla para que el administrador de recursos manipule los registros. Esta manipulación involucra la selección de registros y la confirmación de los cambios que realice sobre los registros. Para controlar esto LIBREP emplea jquery.

El diseño modular de LIBREP permite su extensión a nuevas funcionalidades; además, facilita su mantenibilidad.

Limitaciones

El módulo gestionar solicitudes de artículos de papelería no se completó. En este momento solo se cuenta con las operaciones de registrar, eliminar y consultar artículos de papelería, pero no se cuenta con modificar.

En las solicitudes de artículos de papelería, fue complicado proponer un algoritmo que nos indique la coincidencia en las solicitudes. Es decir, la manera de especificar un artículo de papelería es muy variado; por ejemplo: Cartulina gruesa color: rosa y Cartulina gruesa color: rosa fosforescente. Hasta este momento LIBREP puede detectar duplicados, pero es frecuente que el sistema contenga varias solicitudes del mismo artículo de papelería.

Al agregar un catálogo para un distribuidor, LIBREP aloja la cantidad mínima que surte por material y si maneja precio de mayoreo. LIBREP no contempla costos individuales o por lote.

11. Conclusiones

LIBREP es una aplicación para mejorar algunos procesos de la sección de librería UAM-Azcapotzalco. Estos procesos son: 1) solicitud de un material (libro o artículo de papelería) agotado; 2) levantamiento de pedidos de libros o artículos; 3) Control de cartera de distribuidores;

LIBREP está compuesto por los siguientes módulos: 1) Gestionar usuarios; 2) Gestionar solicitudes de libros; 3) Gestionar solicitudes de artículos de papelería; 4) Gestionar distribuidores y 5) Gestionar pedidos de libros o artículos de papelería.

LIBREP es una aplicación web, que mantiene la independencia entre las capas siguientes: vista, lógica de negocio y persistencia de datos. De tal manera que una modificación en alguna de las capas no afecta de manera sustancial a las otras. Además, implementa los patrones de diseño: *strategy* y *fachade*. LIBREP permite registrar una solicitud de un material (libro o artículo de papelería) de manera sencilla, organizada y rápida; esto se logró empleando el patrón *fachade*. Este patrón facilita, al programador, entender las funcionalidades de la aplicación.

La lógica del negocio se planteó de tal forma que es fácil agregar nuevas funcionalidades. Por ejemplo: si se desea manejar otro tipo de solicitud, esta solicitud podría implementarse más fácilmente, incorporando sus características, gracias al patrón *Strategy*. Este patrón nos permite reutilizar código y ahorrar tiempo. Cada módulo cuenta con las operaciones básicas de: registrar, modificar, eliminar y consultar, logrando uniformidad en la aplicación.

Una característica de LIBREP es el uso de indicadores para que el administrador de recursos identifique aquellos registros que requieren ser atendidos. Estos indicadores son colores: verde y rojo. El color rojo significa que los registros son de mayor prioridad, mientras que el color verde significa que son de menor prioridad. Un registro es de alta prioridad cuando hay gran demanda en una solicitud de un libro ó de un artículo de papelería, ó cuando la solicitud de un libro ó artículo fue hecha por un profesor.

Además LIBREP permite tener un control de los usuarios, que facilita el contacto entre la coordinación y la comunidad universitaria.

LIBREP permite la manipulación directa a los registros cuando se realiza un pedido.

A modo de confirmación LIBREP proporciona el envío de mensajes al usuario a lo largo de un procedimiento por ejemplo cuando se registra un pedido.

El Proceso Unificado marcó las fases de todo el desarrollo: primero en la fase de inicio se logró tener una visión aproximada, la organización de actividades de acuerdo con el calendario escolar y las primeras entrevistas con el cliente. En la siguiente fase se refinó el documento de visión, se optó por ordenar los requisitos de mayor a menor prioridad; los casos de uso de texto se desarrollaron de forma iterativa, y se construyó una base de datos preliminar. La fase de construcción fue la de mayor duración, donde se construyó la base de datos definitiva y se diseñaron e implementaron los módulos de forma iterativa. Y por último en la fase de transición se integraron los módulos, se realizaron pruebas unitarias y pruebas de sistema. Finalmente se realizó el análisis de resultados, y la identificación de las limitaciones, así como el análisis de las posibles mejoras y extensiones de LIBREP.

En la siguiente tabla se muestra si se cumplió un objetivo, su medida y comentarios.

Módulo (objetivo)	Medida	Comentarios
Gestionar de usuarios	completo	Se cumple con las operaciones de registrar, modificar, eliminar y consultar los usuarios del sistema.
Gestionar solicitudes de libro	completo	Se cumple con las operaciones de registrar, modificar, eliminar y consultar solicitudes de libro.
Gestionar solicitudes de artículos de papelería	parcial	Se cumple con las operaciones de registrar, eliminar y consulta solicitudes de artículos de papelería. La funcionalidad de modificar solicitudes se tiene en la capa de negocio pero no se implementó la capa de presentación.
Gestionar distribuidores	completo	Se cumple con las operaciones de registrar, modificar, eliminar y consultar distribuidores. Para un catálogo de distribuidor, LIBREP cuenta con las operaciones registrar y consultar.
Gestionar pedidos de libros o artículos de papelería	parcial	En el caso de libros se cumple con las operaciones registrar, modificar, eliminar y consultar pedido de libros. En el caso de artículos, se cumple con las operaciones de eliminar y consultar. Las funcionalidades de registrar y modificar pedido de artículos se tiene en la capa de negocio pero no se implementó la capa de presentación.

12. Perspectivas del proyecto

LIBREP puede ser mejorada en los módulos actuales y puede ser extendida a nuevas funcionalidades.

El módulo de gestionar pedidos podría mejorarse, con operaciones estadísticas que obtengan información sobre i) los distribuidores a quienes más se les consume y sobre los costos de los productos que ofrecen, ii) época en que más se vende un libro o un artículo de papelería, iii) la cantidad de alumnos que no regresan por su pedido.

En el módulo de gestionar solicitudes de artículos de papelería, completar la operación de modificar. En este módulo se podría manejar un catálogo con todos los artículos de papelería con los que dispone la librería, de tal manera que al registrar uno se evite que el usuario realice la especificación.

En el módulo de gestionar solicitudes de libros, en particular en la operación de consultar solicitudes, se podría agregar más filtros para consultas avanzadas por ejemplo, por título, autor y tomar un rango de tiempo.

Una extensión de LIBREP puede ser un módulo que permita hacer cálculos estadísticos o gráficos: por ejemplo para saber en qué temporada se vende más un libro o un artículo de papelería, que editorial es favorita entre los usuarios. Este módulo serviría para mejorar la toma de decisiones de la Coordinación de la Sección de Librería y papelería de la UAM-A. También se puede pensar en extender LIBREP para que opere en una aplicación móvil.

Otra extensión es crear un módulo de consulta sobre el historial de sus solicitudes de un usuario.

Bibliografía

- [1] Kendall, Kenneth E. y Kendall, Julie E. *Análisis y diseño de sistemas*. 8va. Edición. Pearson Educación, México, 2011.
- [2] Lujan Mora, Sergio. *Programación de aplicaciones web: Historia, principios básicos y clientes web*.
- [3] Gamma Erich; Richar Helm; Ralph Johnson; Vlissides John. *Patrones de Diseño.Elementos de software orientado a objetos reusable*. Pearson Educación, S.A . Madrid, 2003.
- [4] Freeman Eric; Freeman Elisabeth; Sierra Kathy; Bates Bert. *Head first Design Patterns*. O'Really Media. 2004.
- [5] Larman Craig; *UML Y PATRONES . una introducción al análisis y diseño orientado a objetos y al proceso unificado*. 2da. Edición. Pearson Educación.
- [6]https://netbeans.org/index_es.html
- [7] <http://struts.apache.org/index.html>
- [8] <http://tomcat.apache.org/index.html>
- [9] <http://www.mysql.com/products/workbench>
- [10] <http://www.oracle.com/technetwork/java/javamail/index.html>
- [11]<http://itextpdf.com>
- [12] <http://jquery.com/>

Anexo A. Descripción de tablas

USUARIO								
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción
id_Usuario	Int			X		X	PK	Identificador del usuario
Nom_usuario	Varchar(25)			X				Nombre del usuario
ApellidoPaterno_usuario	Varchar(30)			X				Apellido Paterno
ApellidoMaterno_usuario	Varchar(30)			X				Apellido Materno
Matricula_usuario	Varchar(25)			X				Matricula si es alumno. No económico (trabajadores)
correo_usuario	Varchar(60)			X				Correo del usuario
Password_usuario	Varchar(15)			X				Password del usuario
Id_Puesto	int			X			Fk	Llave foránea a la tabla Puesto columna(id_Puesto)

PUESTO								
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción
Id_Puesto	int			X		X	Pk	Llave primaria Identificador único.
Descripcion_puesto	Varchar(45)			X				Descripción del puesto por ejemplo (alumno, administrador, profesor)

REGISTROLIBROESPEC								
	Tipo	Uniq ue	Null	Not null	Default	Autoincrement	Llave	Descripción
id_RegistroLibroespec	int			X		X	PK	Llave primaria de un especificación de libro.
Titulo_RegistroLibroespec	Varchar(80)			X				Título de la especificación del libro.
Autor_RegistroLibroespec	Varchar(80)			X				Autor de la especificación del libro
Edicion_RegistroLibroespec	Varchar(3)			X				Edición de la especificación del libro
Año_RegistroLibroespec	Varchar(10)			X				Año de la especificación de libro
ISBN_RegistroLibroespec	Varchar(15)		X					ISBN de la especificación de libro
id_editorial	int			X			FK	Llave foránea relacionada la tabla Editorial columna(id_editorial)

EDITORIAL								
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción
id_Editorial	int			X		X	Pk	Llave primaria Identificador único. De una Editorial
Descripcion_editorial	Varchar(45)			X				Descripción del editorial (Nombre) por ejemplo (PORRUA,PEARSON..Etc.)

REGISTROLIBRO								
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción
Id_RegistroLibro	Int			X		X	PK	Llave primaria. Identificador único RegistroLibro.
Fecha	Date			X				Fecha de registro libro por un usuario.
Hora	Varchar(25)		X					Hora de registro libro por un usuario. Formato (HH:mm:ss)
Id_Usuario	int			X			FK	Llave foránea. Id del Usuario quien realizo el registro. Tabla usuario columna(id_Usuario)
Id_RegistroLibroespec	int			X			FK	Llave foránea. Relacionada a la tabla RegistroLibroespec columna(id).

SOLICITUDLIBRO								
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción
id_SoliciudLibro	int			X		X	PK	Llave primaria. De una solicitud de libro.
Solicitantes_SolicitudLibro	int			X				Cantidad de solicitantes.
id_RegistroLibroespec	int			X				Especificación del libro.
id_PedidoLibro	int		X				FK	Llave foránea relacionada a la tabla PedidoLibro columna(id_Pedido Libro)

PEDIDOLIBRO								
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción
Id_PedidoLibro	Int			X		X	PK	Llave primaria. Identificador único PedidoLibro.
Fecha	Date			X				Fecha de registro de un pedido de libro.
Hora	Time			X				Hora de registro de un pedido de libro.
cve_usr	Varchar(25)			X				Es el número económico del usuario que registro un pedido.
Id_distribuidor	int			X			FK	Llave foránea. Relacionada a la tabla Distribuidor columna(id). Id del distribuidor a quien se dirige el pedido

RegistroArticulo									
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción	
Id_RegistroArticulo	Int	X		X		X	Fk PK	Llave primaria de una relación con la tabla PedidoArticulo (id) único	
Cve_espec	int	X					Fk	Llave foránea relacionada a la tabla RegistroArticulo.	
Fecha	Date			X				Fecha de registro de un artículo por un usuario.	
Hora	Varchar(25)		X					Hora de registro de un artículo por un usuario. Formato (HH:mm:ss)	
Id_Usuario	int			X			FK	Llave foránea. Id del Usuario quien realizo el registro. Tabla usuario columna(id_Usuario)	
Id_Registro Articuloespec	int			X			FK	Llave foránea. Relacionada a la tabla RegistroArticuloespec columna(id).	

SOLICITUDARTICULO									
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción	
id_SolicitudArticulo	int			X		X	PK	Llave primaria. De una solicitud de Artículo.	
Solicitantes_SolicitudArticulo	int			X				Cantidad de solicitantes.	
id_RegistroArticuloespec	int			X				Especificación del Artículo.	
id_PedidoArticulo	int		X				FK	Llave foránea relacionada a la tabla PedidoArticulo columna(id_PedidoArticulo)	

PEDIDOARTICULO									
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción	
Id_PedidoArticulo	Int			X		X	PK	Llave primaria. Identificador único PedidoArticulo.	
Fecha	Date			X				Fecha de registro de un pedido de Artículo .	
Hora	Time			X				Hora de registro de un pedido de Artículo.	
cve_usr	Varchar(25)			X				Es el número económico del usuario que registro un pedido.	

Id_distribuidor	int			X				FK	Llave foránea. Relacionada a la tabla Distribuidor columna(id). Id del distribuidor a quien se dirige el pedido
------------------------	-----	--	--	---	--	--	--	----	---

PedidoA_espec								
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción
Cve_pedidoA	int			X			Fk	Llave foránea relacionada a la tabla PedidoArticulo (id)
Cve_especA	int			X			Fk	Llave foránea relacionada a la tabla RegistroArticuloespec(id). Especificación del pedido.

RegistroArticuloespec									
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción	
Id_RegistroArticuloespec	Int			X		X	PK	Llave primaria de una especificación de artículo.	
Descripcion_RegistroAespec	Varchar(100)			X				Descripción de un artículo por ejemplo (papel albanene).	
Marca_RegistroAespec	Varchar(45)			X				Marca del artículo.	
Especificaciones_RegistroAespec	Varchar(45)			X				Detalles de un artículo.	
Cantidad_sugerida_RegistroAespec	int			X				Cantidad sugerida del artículo registrado.	

DISTRIBUIDOR									
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción	
Id_Distribuidor	int			X		X	PK	Llave primaria de un distribuidor	
Nombre_Distribuidor	Varchar(45)	X		X				Nombre de un distribuidor	

Dirección_Distribuidor	Varchar(100)			X				Dirección.
Telefono_Distribuidor	Varchar(15)			X				Telefono de un distribuidor
Celular_Distribuidor	Varchar(20)		X					Celular
Correo_Distribuidor	Varchar(60)		X					Correo de un distribuidor
Referencias_Distribuidor	Varchar(100)		X					Anotaciones de un distribuidor
Tipo	int			X				Tipo de distribuidor 1)distribuidor de libros y 0)distribuidor de articulos
Estado	int			X				Estado 1)activo y 0)inactivo

Catalogo_Distribuidor								
	Tipo	Unique	Null	Not null	Default	Autoincrement	Llave	Descripción
Id_CatalogoDistribuidor	int			X		X	Pk	Llave primaria de un catalogo
Descripcion_CatalogoD	Varchar(45)			X				Descripcion de material que surte. Ejemp PORRUA, compas metalico..etc.
Cantidad_minima_CatalogoD	int			X				Cantidad mínima que surte por descripcion.
Precio_mayor_CatalogoD	boolean			X				1)si lo maneja 0)no
Id_Distribuidor	int			X			FK	Llave foránea relacionada a la tabla Distribuidor(id).