



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

Sistema de administración y abastecimiento del almacén de la
División de Estudios de Posgrado de la Facultad de Medicina de
la UNAM.

REPORTE DE SERVICIO SOCIAL.

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA
COMPUTACIÓN**

P R E S E N T A :

ARAMIS ARIAS ROMERO



**TUTOR
M. EN I. DAVID VELÁZQUEZ PORTILLA
2017**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno.

Arias

Romero

Aramis

0445513718597

Universidad Nacional Autónoma de México

Facultad de Ciencias

Ciencias de la computación

304185320

2. Datos del tutor.

M. en I.

David

Velázquez

Portilla

3. Datos del sinodal 1

Dra.

Hanna Jadwiga

Oktaba

4. Datos del sinodal 2

Dr.

Miguel Ehécatl

Morales

Trujillo

5. Datos del sinodal 3

M. en I.

Gerardo

Avilés

Rosas

6. Datos del sinodal 4

M. en C.

Gustavo Arturo

Márquez

Flores

7. Datos del trabajo escrito

Sistema de administración y abastecimiento del almacén de la División de Estudios de Posgrado de la Facultad de Medicina. UNAM.

106 p

2017

AGRADECIMIENTOS.

Después de un largo camino recorrido, al mirar hacia atrás, me doy cuenta de todos los momentos vividos. Momentos buenos, malos y peores, que sin embargo, cada uno, a su manera, ha dejado enseñanzas y experiencias las cuales sabré aprovechar para esta nueva etapa en mi vida.

Ha pasado tiempo desde aquel primer día en que llegué a la UNAM. Recuerdo a ese Aramis desorientado, con miedo y sin saber que sería de su vida en esa etapa que estaba a punto de comenzar. Me agobiaban preguntas como: “¿Estaré haciendo lo correcto?”, “¿Realmente es aquí donde quiero estar?”, “¿Y si no puedo?”. Con el paso del tiempo esas preguntas se fueron contestando por si solas, y, sin darme cuenta, la UNAM se convirtió en mi segundo hogar, lugar donde conocí a gente y amigos maravillosos, que con el paso del tiempo, se fueron convirtiendo en parte fundamental de mi vida.

Afortunadamente, hoy estoy donde quiero estar, pero esto no hubiera sido posible sin esas personas que siempre creyeron en mí, a pesar de haber fallado más de una vez y haber caído en desesperación en aquellos momentos complicados en los cuales pensaba que jamás podría contra ciertas situaciones.

Me rodea una especie de nostalgia, alegría, incertidumbre y a la vez una mezcla de sentimientos encontrados. Estoy a un paso de cumplir mi sueño que un día lo veía completamente lejos e inalcanzable.

Hoy puedo decir que he encontrado mi vocación: “Las Ciencias de la Computación y la programación son mi vida”. Estoy muy agradecido con el destino por ponerme en el lugar y momento indicado para darme cuenta de que a partir de hoy y por el resto de mi vida haré lo que realmente me gusta y me apasiona: La programación.

Quiero agradecer a mis padres, Leonardo Arias y Ángeles Romero, por ese gran apoyo y creer en mí sin importar las situaciones o circunstancias. Nunca me dejaron de lado y siempre me apoyaron y me hicieron creer que realmente podría hacerlo. A mi hermano, Tony Arias, que siempre estuvo atento, apoyándome en todo momento y dándome consejos que fueron y serán sumamente valiosos. A Keyla, que desde la primera vez, me trató como parte de los suyos y que ha demostrado ser alguien que es capaz de sacar a una familia adelante. A Dana, Leo y Sharon, mis sobrinos que hace algunos años llegaron a mi vida y me enseñaron un nuevo significado de la palabra “Tío”. A todos ustedes, nunca podré dejar de agradecerles todo lo que hicieron por mí y esos valores que desde niño me inculcaron y que jamás olvidaré. Los amo.

A los sinodales, por tomarse el tiempo para revisar, corregir mi trabajo y aceptar formar parte del jurado para mi examen profesional.

A mi asesor, David Velázquez, por ese tiempo que invirtió en orientarme y corregir los errores de mi trabajo. Por esa paciencia, oportunidades que me ofreció y sobre todo, por la confianza que me brindó desde el primer momento en que llegué a posgrado.

A Erick Matla, por esa paciencia, orientación y disponibilidad de siempre ayudarme y echarme la mano para sacar tanto el proyecto adelante, como asuntos y dudas ajenas a este mismo.

A Miguel Morales, por esa disponibilidad, amabilidad y tiempo que invirtió en revisar mi reporte y hacerme correcciones. En verdad muchas gracias.

A mis amigos de la facultad, Pako, Enrique (Wee-man), Miguel (Patatas, te debo varios paros), Fer, Ángel (BJ), Efraín (Estopa) y todos los que me acompañaron en la facultad durante este camino. Creo que me es imposible nombrarlos a todos, pero de antemano muchas gracias.

A Jessica, una de las personas más importantes que conocí en la facultad, la cual quiero que se quede en mi vida para siempre. Gracias por todos tus consejos, por aguantarme todas mis graciosadas y por todos esos momentos alegres y difíciles que pasamos.

A pumitas, que me hizo conocer a grandes amigos: David (De los primeros y que llegaron para quedarse), Rodrigo (Dreik), Pancho (Polémico), Ariel, por supuesto que Gustavo (Gracias por todo amigo), Jazz (Gracias por todos esos consejos e infinitas pláticas), Carlos (Gracias por todos estos años amigo) y a toda la banda de pumitas que siempre está ahí, me es imposible mencionarlos a todos, pero en verdad, gracias por todo.

A mis amigos de la infancia: Hector, Agustín, Giovanni, Oscar y a la banda de la JB. A pesar de que nuestros caminos de a poco se han ido separando, la amistad ahí estará y durará por siempre.

A todas y a cada una de las personas que me brindaron su apoyo incondicional, siempre estaré infinitamente agradecido. Gracias por esta magia.

“Tener verdadero éxito en la vida es: reír mucho y muchas veces; ganar el respeto de personas inteligentes; gozar del cariño de niños; ganar el reconocimiento de personas cualificadas y saber soportar la traición de falsos amigos; apreciar la belleza; buscar lo mejor en los demás; dejar el mundo un poquito mejor de cómo lo encontraste: con un hijo sano, un jardín bonito o una persona más feliz; saber que al menos alguien ha vivido mejor gracias a ti.”

-Anónimo-

ÍNDICE

AGRADECIMIENTOS.....	1
1. PROBLEMÁTICA.....	6
1.1. <i>Introducción</i>	6
1.2. <i>Objetivos</i>	6
2. CONCEPTOS BÁSICOS.....	7
2.1. <i>Ingeniería de software</i>	7
2.2. <i>Proceso Unificado</i>	9
2.3. <i>Patrón MVC</i>	11
2.4. <i>Base de datos</i>	12
3. ANÁLISIS DE REQUERIMIENTOS.....	14
3.1. <i>Requerimientos funcionales</i>	14
3.2. <i>Diagrama general de casos de uso</i>	15
3.3. <i>Casos de uso detallados</i>	16
3.3.1. <i>Iniciar sesión</i>	16
3.3.2. <i>Solicitar material</i>	17
3.3.3. <i>Validar material</i>	19
3.3.4. <i>Registrar Material</i>	20
3.3.5. <i>Registrar material entregado</i>	21
3.3.6. <i>Cerrar sesión</i>	22
3.4. <i>Requerimientos no funcionales</i>	23
3.5. <i>Prototipos</i>	23
3.5.1. <i>Prototipos de administrador</i>	24
3.5.2. <i>Prototipos de usuario</i>	31
3.5.3. <i>Prototipos de almacenista</i>	34
4. ADMINISTRACIÓN DE LA CONFIGURACIÓN.....	38
4.1. <i>Definición de la línea base</i>	38
4.2. <i>Definición de los componentes a controlar</i>	38
4.3. <i>Creación del depósito del proyecto</i>	39
4.4. <i>Características del equipo</i>	39
4.5. <i>Definición de control de cambios</i>	40
4.5.1. <i>Procedimiento de cambios y revisiones</i>	40
4.5.2. <i>Informe del estado de la configuración</i>	40
5. DISEÑO.....	42
5.1. <i>Diagramas del administrador</i>	43
5.1.1. <i>Control</i>	43
5.1.2. <i>Modelo</i>	44

5.1.3. Estado	45
5.2. Diagramas del almacenista	46
5.2.1. Control	46
5.2.2. Modelo	47
5.2.3. Estado	48
5.3. Diagramas del Usuario.	49
5.3.1. Control	49
5.3.2. Modelo	50
5.3.4. Estado	51
5.4. Diagramas de secuencia.....	52
5.4.1. Caso de uso: Iniciar sesión.	52
5.4.2. Caso de uso: Solicitar material.	53
5.4.3. Caso de uso: Validar solicitud de material	54
5.4.4. Caso de uso: Dar de alta material.	55
5.4.5. Caso de uso: Registrar material entregado.	56
5.5. Esquema de la Base de datos.....	57
5.6. Tablas de la base de datos.....	59
6. CONSTRUCCIÓN DEL SOFTWARE.....	67
6.1. Entendimiento del diseño del software.	67
6.2. Implementación de software a partir del diseño de software.	67
7. INTEGRACIÓN.....	69
8. PRUEBAS.....	78
8.1. Entendimiento de casos de uso para la realización de pruebas.	78
8.2. Realización de pruebas unitarias.	78
8.3. Corrección de errores y defectos encontrados.	79
8.4. Realización de pruebas mediante casos	80
8.4.1. Registro de material.	80
8.4.2. Registro de marca.	81
8.4.3. Registro de una solicitud.	81
8.4.4. Acción a las solicitudes.....	81
8.4.5. Acción a las solicitudes por parte del usuario.	81
8.4.6. Consulta de solicitudes:.....	81
8.4.7. Consulta de solicitudes por estado.....	81
8.4.8. Corrección de defectos encontrados.	82

9. ENTREGA DE PRODUCTOS.....	83
9.1. <i>Especificaciones</i>	83
9.2. <i>Manual de mantenimiento</i>	83
9.3. <i>Manual de usuario</i>	83
9.4. <i>Verificación del manual por parte del usuario</i>	98
10. RESULTADOS.....	101
11. CONCLUSIONES	103
BIBLIOGRAFÍA	105
APÉNDICE	106

1. PROBLEMÁTICA.

1.1. Introducción.

En la División de Estudios de Posgrado de la Facultad de Medicina (DEP-FM), para solicitar insumos del almacén, los usuarios realizan las solicitudes por medio de un “vale”, el cual es llenado con los datos de los materiales y accesorios que requieren.

Realizar dicho proceso, conlleva una serie de problemáticas entre las cuales destacan la pérdida de tiempo, errores en el llenado de la solicitud, retraso en el abastecimiento de material, modificación de las cantidades de material solicitadas, entre otros; lo cual da como resultado un mal control administrativo del almacén.

Al haber distintas formas en las que se llena una solicitud, muchos campos de información son registrados incorrectamente y esto ocasiona que los vales se entreguen con errores y generen problemas al momento de gestionar dicho material.

Por lo anterior, es necesario un formato que automatice y haga más fácil y práctica la solicitud de material del almacén y que, a su vez, guíe al usuario para que dicho vale sea completado de forma correcta y sin contratiempos.

Por otro lado, se necesita que todos los movimientos que haya en el almacén sean registrados para llevar un control de las peticiones de los usuarios y de los departamentos a los que pertenecen, esto para hacer cortes mensuales de esas extracciones y de la misma manera se podrán saber las existencias de material que hay en el almacén.

1.2. Objetivos.

Este reporte describe un módulo de un sistema de software que tiene como objetivos:

- Automatizar el manejo y administración de la entrada y salida de material de almacén, así como las solicitudes de abastecimiento y suministro de bienes de la DEP-FM.
- Manejar de una manera segura e íntegra la información del almacén y de la solicitud del material.
- Agilizar y facilitar la dinámica de solicitud de abastecimiento de material.

2. CONCEPTOS BÁSICOS.

Para tener un mejor entendimiento del sistema a desarrollar, en este capítulo se mostrarán algunos conceptos y definiciones que ayudarán a comprender la funcionalidad del sistema en cuestión y su contexto.

En el desarrollo de cualquier sistema de información, se recomienda tener documentación actualizada y clara, ya que esto facilitará la comprensión del sistema para actualizaciones y/o para agregar nuevas funcionalidades para cubrir las necesidades del usuario final.

El objetivo de documentar el sistema será crear una guía para que el equipo técnico de programación de la Coordinación de Cómputo e Informática (CCI) de la DEP-FM, sea capaz de entenderlo, corregir defectos, añadir módulos y extensiones de manera más rápida y eficiente.

A continuación se mencionarán algunos conceptos básicos para un mejor análisis y desarrollo.

2.1. Ingeniería de software

Podemos definir a la ingeniería de software como aquella disciplina la cual se ocupa del desarrollo, implementación y el mantenimiento de algún tipo de software.

Para tener un concepto más claro de lo que es la ingeniería de software, es necesario estudiar algunas metodologías que se llevan a cabo para el desarrollo de un sistema, por tanto, la disposición de ese conocimiento es lo que permitirá el diseño y la construcción de software, lo que nos llevará a operar de un modo satisfactorio nuestro software en diversas computadoras y/o plataformas.

Entonces, la ingeniería de software implica un proceso en el cuál se produce un análisis del contexto o la idea, se diseña el proyecto, se desarrolla el software, se efectúan las pruebas para asegurar su correcto funcionamiento para finalmente poder implementarlo (Pressman, 2006).

Una vez que el software está funcionando, se realizarán tareas de mantenimiento. Generalmente, suelen aparecer errores con relación al diseño del programa, por lo cual, el proceso de mantenimiento será el que permitirá corregir dichos errores.

Para nuestro proceso de desarrollo, involucraremos las siguientes etapas que se irán definiendo con más detalle en capítulos posteriores (Sommerville, 2005):

- **Especificación de requerimientos:** A través de entrevistas y juntas con el cliente o usuario final, se recabarán todas las necesidades del negocio o

los requisitos del usuario los cuales debe cumplir el sistema de software. En este punto se reconocerá, entenderá y modelará el problema para facilitar el desarrollo en etapas posteriores.

- **Análisis.** En esta etapa a través de diagramas, se describirá el comportamiento esperado del software y su interacción con el usuario final.
- **Diseño.** Después del análisis, se procederá a representar los requisitos recabados en los pasos anteriores a través de interfaces, algoritmos y estructuras de datos diseñadas para una buena ejecución de dicho software.
- **Desarrollo.** En esta etapa se utilizarán los modelos creados durante la etapa de diseño para crear los componentes del sistema, dichos modelos se traducirán a código legible por la computadora, el cuál será escrito en el lenguaje de programación elegido por el equipo de desarrollo.
- **Implementación.** Una vez concluida la etapa de desarrollo, se procederá a la etapa de implementación. Aquí es donde todos los componentes desarrollados y codificados en la etapa de desarrollo se juntan. Cada componente se enlaza con el resto para obtener el producto, para finalmente realizar la etapa de pruebas.
- **Pruebas.** Después de haber generado el código y un sistema funcional, se pasa a la etapa de pruebas. En esta etapa se comprobará que el software tenga el funcionamiento correcto y muestre los resultados esperados. Las pruebas se realizarán tanto por el equipo de desarrollo como por el usuario final, esto con el fin de poder realizar una retroalimentación de ambas partes.
- **Liberación.** Cuando el software ha pasado la fase de pruebas de manera satisfactoria, se procederá a hacer la entrega final cliente. En esta etapa se podrá dar por hecho que el producto funciona correctamente y que cubrirá las necesidades del cliente.

2.2. Proceso Unificado

Se podrá definir el Proceso Unificado como una metodología de trabajo la cual se puede extender y ser adaptada a proyectos específicos. Dicho proceso puede ser utilizado para diferentes tipos de software, organizaciones, áreas de aplicación y diferentes tamaños de proyectos. Su finalidad es poder asegurar la producción de software que satisfaga las necesidades de los usuarios finales, dentro de un tiempo determinado y con un presupuesto predecible (Van Vliet, 2008).

El Proceso Unificado usa UML (Lenguaje de Modelado Unificado) el cual consiste en usar diagramas para visualizar, especificar, construir y documentar un sistema. También permite modelar aspectos conceptuales como procesos, funciones del sistema, expresiones de lenguajes de programación y esquemas de bases de datos.

Las características principales del Proceso Unificado es que está basado en tres conceptos clave: Dirigido por casos de uso, está centrado en la arquitectura y es iterativo e incremental.

A continuación se detalla cada una de las características (Jacobson, 2000):

- **Casos de uso:** Podemos definir a los casos de uso como las funcionalidades que tendrá nuestro sistema a desarrollar. Se basan en los requerimientos funcionales definidos en la etapa de análisis de requerimientos. Para los casos de uso, se definirán personajes que serán partícipes de las funcionalidades de cada caso de uso. A estos personajes se les denominará actores.

Con el uso de actores, podemos definir un caso de uso como una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Los diagramas de casos de uso nos servirán para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios. De la misma manera, con los diagramas de casos de uso se podrán visualizar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

- **Centrado en la arquitectura:** La arquitectura en un sistema de software se define como las diferentes vistas del sistema que está siendo construido. Tomando la analogía de un edificio en construcción, existen diversos planos que incluyen los distintos servicios que el edificio tendrá y de esta manera el arquitecto puede ver una imagen completa del edificio antes de que comience la construcción. La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles.

La arquitectura surge de las necesidades del usuario final, tal y como están reflejadas en los casos de uso. Aparte de las necesidades del usuario, también está influenciada por otros factores, como por ejemplo, la plataforma de software en la que se ejecutará, consideraciones de instalación, requerimientos no funcionales, entre otros.

- **Iterativo e incremental.** Esta característica corresponde a mantener siempre un enfoque en los cambios que se puedan producir en un proyecto. Consiste en poner en manos del usuario un sistema que se irá completando en fases sucesivas. Así el usuario puede ir checando algunas funcionalidades del sistema mientras se van completando funcionalidades nuevas para presentarlas en futuras iteraciones.

Para el proceso de desarrollo incremental, se van mostrando versiones de un subsistema funcional pequeño, el cuál se le van agregando nuevas funcionalidades con el paso de cada versión.

Sin embargo, en el desarrollo iterativo también se puede entregar un sistema completo desde el principio y a partir de aquí ir modificando las funciones ya existentes en cada nueva versión que se entregue. Tanto el enfoque iterativo como el incremental, se pueden combinar durante el desarrollo.

El desarrollo iterativo se puede considerar como un método de construcción de un producto en el cuál el ciclo de vida está compuesto por iteraciones, las cuales tienen como objetivo entregar versiones de software con correcciones y mejoras.

Cada iteración se considera un proyecto que permite al usuario tener puntos de verificación y control más rápidos, para así poder tener un proceso de pruebas continuas desde las primeras iteraciones.

Los proyectos se pueden dividir en mini-proyectos los cuales se van entregando durante cada iteración o periodo corto de tiempo. Estos incluyen un conjunto reducido de requerimientos a implementar.

El resultado de cada iteración es un sistema al cuál se le pueden aplicar pruebas para ser corregidas y entregadas en la siguiente iteración. Se pueden hacer retroalimentaciones más rápidas y asimilar los cambios de una manera más sencilla en cada entrega.

2.3. Patrón MVC

Para el desarrollo del software propuesto, se utilizará el patrón MVC (Modelo-Vista-Controlador) el cuál es utilizado para el diseño de software de una manera más organizada, tener una mayor facilidad para dar mantenimiento, poder aplicar extensiones a un sistema, reutilizar código y facilitar el desarrollo (Murat, 2015)-.

El patrón MVC (ver Figura 2) se basa en la separación de código para una mayor organización y entendimiento de un sistema. Utiliza tres tipos de capas dependiendo de su funcionalidad¹.

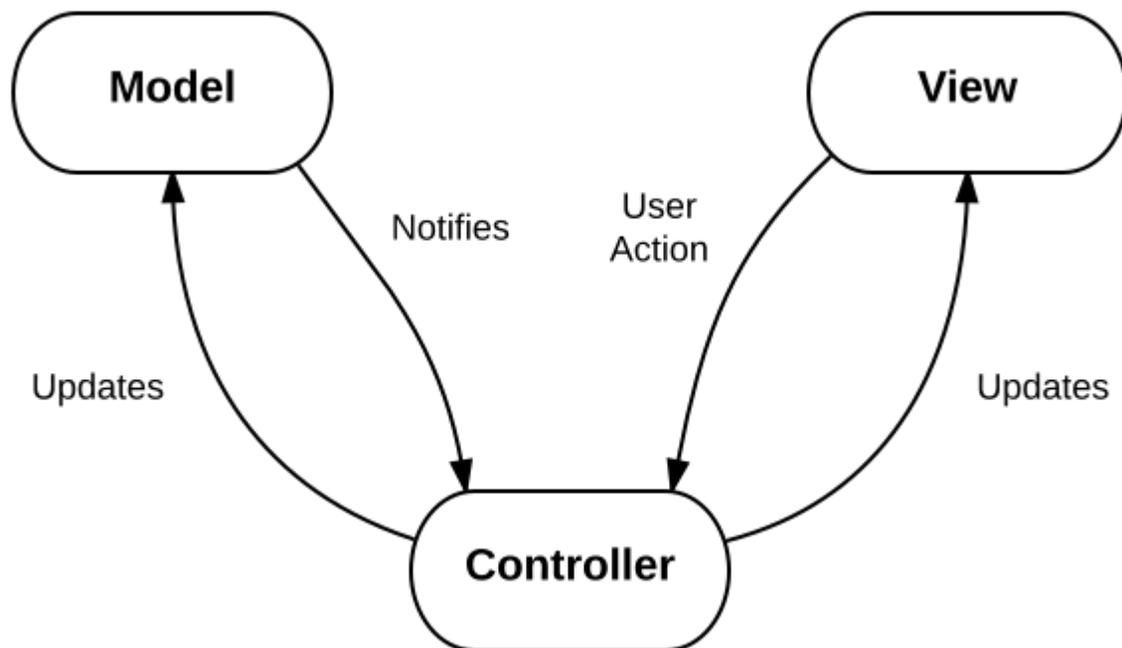


Fig. 2 – Patrón MVC (Modelo-Vista-Controlador)¹

A continuación se explica cada una de las capas (Murat, 2015):

- **Modelo:** En esta capa es donde se tiene el modelo de la base de datos en la cual se harán las operaciones correspondientes para acceder a la información almacenada, aquí se trabaja y almacenan los datos y los cambios que se hagan. Por tanto contendrá los mecanismos que permitirán acceder a la información y así poder enviar a la vista la información que haya sido solicitada.

¹ Imagen tomada de <https://en.wikipedia.org/wiki/Model-view-controller> (Consultada el 01/03/2017)

- **Vista:** Contiene el código y las interfaces gráficas mediante las cuales el usuario interactuará con el sistema y así obtener la información necesaria. Por lo general esta capa contiene código en HTML (Hyper Text Markup Language) y CSS (Cascading Style Sheets). A través de ésta el usuario trabajará con los datos almacenados en la capa del modelo.
- **Controlador:** es la capa que realiza la interacción entre la vista y el modelo, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Contiene el código necesario para responder a las acciones que se solicitan en la aplicación. Por lo tanto se puede decir que el controlador es el intermediario entre la vista y el modelo.

2.4. Base de datos

Podemos definir a las bases de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de algún negocio o empresa.

Algunas características de las bases de datos son las siguientes (Post, 2006) (Ramez, 2007):

- **Independencia física:** Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación.
- **Independencia lógica de datos:** Capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación.
- **Redundancia mínima:** Es la capacidad de eliminar las repeticiones innecesarias de los datos en los archivos de la base de datos. Con esto se evita que la misma información se almacene repetidas veces en la base de datos.
- **Integridad de los datos:** Se refiere a la manera correcta en que los datos están almacenados en la base de datos.
- **Seguridad de acceso:** Protección de los datos para accesos no autorizados.
- **Respaldo y recuperación:** Se puede realizar un respaldo de los datos con el fin de evitar pérdidas imprevistas de la información.

Además de las características de las bases de datos, podemos también clasificar los modelos de bases de datos. Un modelo de datos es la descripción de un contenedor de datos, así como sus funciones o métodos para poder acceder a estos y recuperarlos. Los modelos de datos son abstracciones que permiten la implementación de un sistema de base de datos (Silberschatz, 2010).

Algunos tipos de modelos de bases de datos son los siguientes:

- **Bases de datos jerárquicas:** como su nombre lo indica, almacenan la información de manera jerárquica, es decir, utiliza una estructura parecida a la de un árbol, donde un nodo padre puede tener varios hijos de información. Este tipo de bases son útiles cuando se maneja una gran cantidad de información permitiendo crear estructuras estables de gran rendimiento. Una desventaja de este modelo es que es difícil encontrar redundancia de datos.
- **Bases de datos relacionales:** este tipo de base de datos organiza la información a través de tablas las cuales contienen atributos. Cada tabla almacena algún atributo en común con otra tabla mediante las cuales se establecen relaciones entre tablas. Para poder manipular la información en dichas tablas, se utiliza el álgebra relacional, la cual da bases para realizar algún tipo de consulta para obtener información específica.

Para el desarrollo del sistema, se utilizarán bases de datos relacionales esto con el fin de facilitar la abstracción de la idea y organización de las entidades para poder consultar cualquier tipo de información de manera más clara y eficiente. También podremos manejar más fácil la información para evitar la duplicidad de datos y así mantener la integridad de los datos de una mejor manera (Silberschatz, 2010).

3. ANÁLISIS DE REQUERIMIENTOS

En esta etapa se realiza un estudio y análisis de las necesidades que tiene el cliente, empresa, organización o negocio. Se especifican las características deseadas del software a desarrollar (Pressman, 2006).

Tanto el cliente como el equipo de desarrollo tienen una participación activa ya que a través de juntas, retroalimentaciones y entrevistas de ambas partes, se van definiendo las características y técnicas específicas que se utilizarán a través del desarrollo del software.

De la misma manera, se describe el plan del proyecto a seguir y se programan futuras juntas y retroalimentaciones para hacer un software de mejor calidad que cubra con las necesidades del cliente.

En la fase de análisis de requerimientos se consideran los siguientes puntos a desarrollar:

- Requerimientos funcionales.
- Diagrama general de casos de uso.
- Requerimientos no funcionales.
- Prototipos.

3.1. Requerimientos funcionales

Los requerimientos funcionales del sistema que serán atendidos por el módulo construido en este proyecto, son:

1. El sistema registrará y actualizará material del almacén.
2. Registrará usuarios que tendrán acceso al material del almacén.
3. Se iniciará y cerrará sesión.
4. Se realizarán búsquedas.
5. Se harán registros de material entregado.
6. Se imprimirán vales.
7. El sistema proporcionará información acerca de los productos registrados en la base del almacén.

3.2. Diagrama general de casos de uso

A continuación se definirá el diagrama general de casos de uso, en el cuál se detalla las funcionalidades que tendrá el sistema y los tipos de usuarios que interactuarán con dicho software (actores), como se muestra en la Figura 3.1.

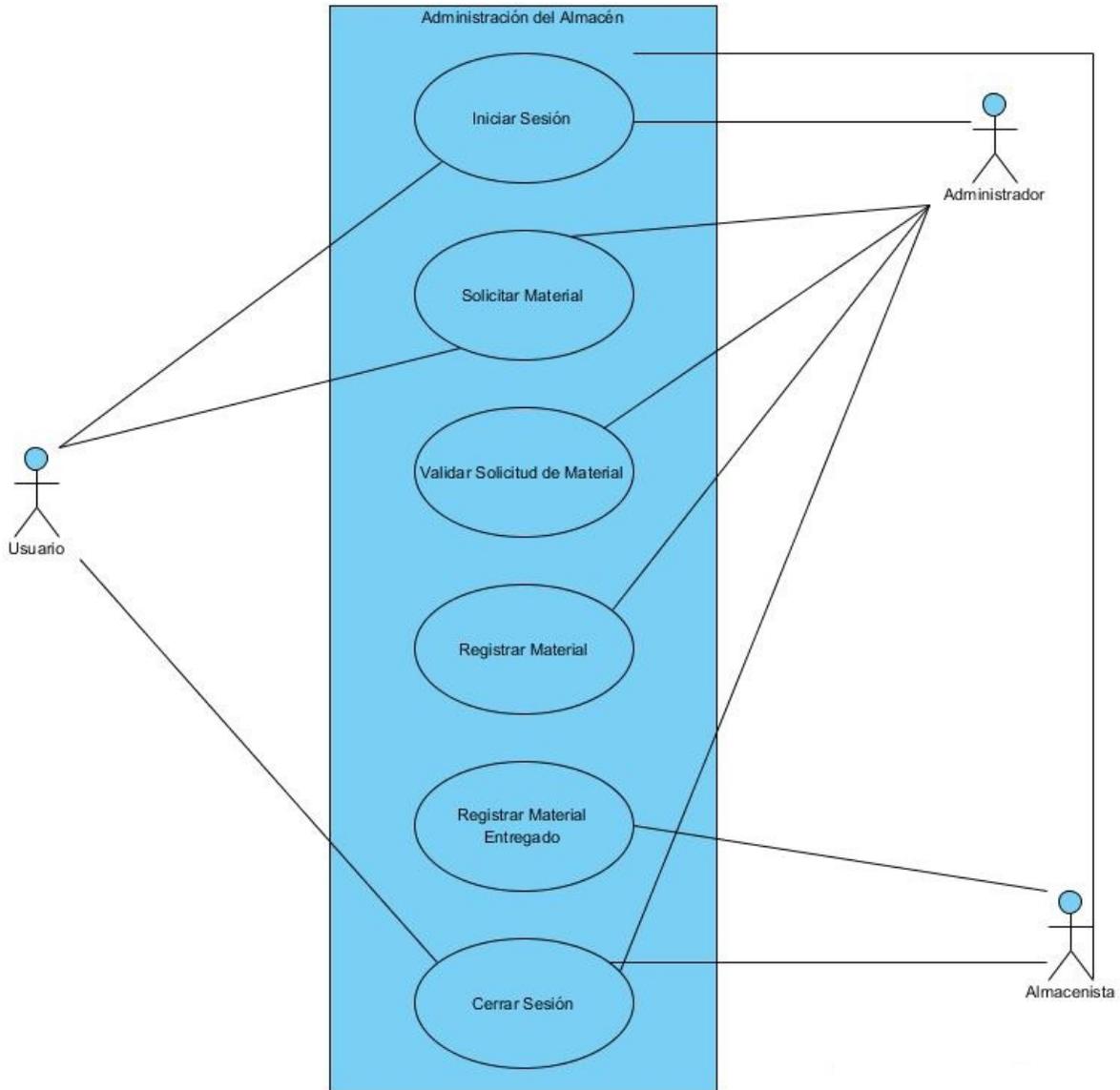


Fig. 3.1 – Diagrama general de casos de uso.

A partir de la Figura 3.1, se detallarán las funcionalidades de cada caso de uso según lo establecido en el análisis de requerimientos.

3.3. Casos de uso detallados.

En esta sección se describirán los requerimientos funcionales del sistema a través de casos de uso.

3.3.1. Iniciar sesión

Caso de uso: *Iniciar sesión.*

Actores: *Usuario, administrador, almacenista.*

Descripción: *El usuario se identifica para tener acceso a las funciones del sistema dependiendo de qué tipo de usuario sea.*

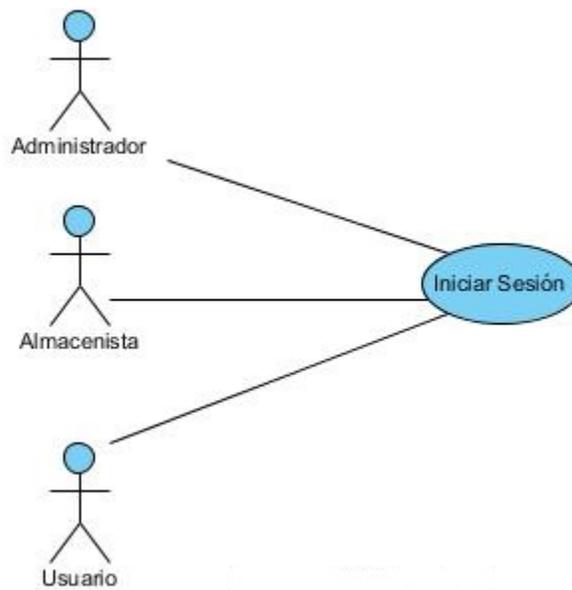


Fig. 3.2 – Caso de uso de inicio de sesión.

Precondiciones:

Debe de estar dado de alta en el sistema y haber establecido la contraseña de la aplicación para poder acceder.

Suposiciones:

Los usuarios que conocen la contraseña de entrada deberán de ser los únicos autorizados para ingresar al sistema.

Flujo normal de eventos:

1. El usuario introduce la contraseña.
2. El sistema comprueba si la contraseña introducida es correcta.
3. El sistema determinará y mostrará las vistas adecuadas de acuerdo al usuario autenticado.

Flujo alternativo de eventos:

- Si la contraseña es incorrecta, el sistema deberá mostrar un mensaje de error. Se indicará que se vuelva a intentar el acceso.

Post-condiciones:

- El sistema muestra las vistas según el usuario que haya iniciado sesión (usuario normal, administrador o almacenista) permitiendo opciones que son exclusivas de acuerdo al usuario.

3.3.2. Solicitar material

Caso de uso: Solicitar material.

Actores: Usuario.

Descripción: después de acceder al sistema, el usuario solicita material a través de un formulario.

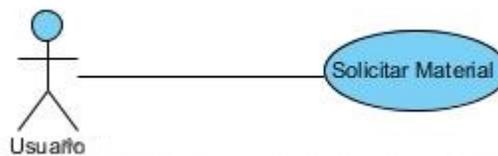


Fig. 3.2 – Caso de uso de solicitar material.

Precondiciones:

El usuario debe de haber iniciado sesión.

Suposiciones:

El material solicitado está en existencia en el almacén.

Flujo normal de eventos:

1. *El usuario consulta material en existencia.*
2. *El usuario solicita material.*
3. *Selecciona la cantidad de material.*
4. *Envía solicitud de material.*
5. *Se asigna automáticamente folio de solicitud de material*

Flujo alternativo de eventos:

1. *Selecciona material agotado.*
2. *Muestra mensaje de error.*

Post-condiciones:

1. *El sistema muestra el vale del material solicitado para confirmar envío de solicitud de material.*
2. *Se envía la solicitud de material para validación por parte del administrador.*

3.3.3. Validar material

Caso de uso: Validar solicitud de material.

Actores: Administrador.

Descripción: Después de que el usuario solicita el material, el administrador autoriza o cancela solicitud de dicho material. Si es necesario, el administrador modifica las cantidades de material solicitado.

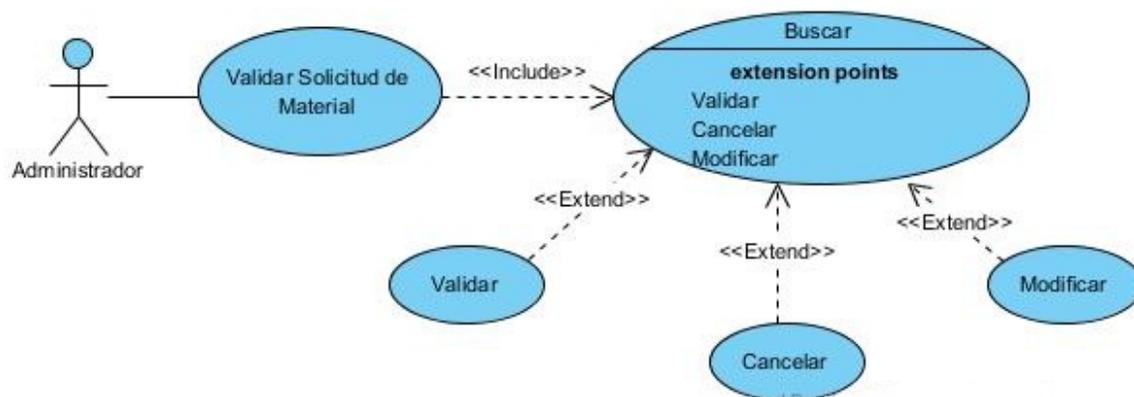


Fig. 3.3 – Caso de uso de validar o cancelar material.

Precondiciones:

El usuario previamente realizó la solicitud de material.

Suposiciones:

El material solicitado está en existencia en almacén.

Flujo normal de eventos:

1. El administrador consulta material solicitado.
2. Consulta el material disponible.
3. El administrador, si es necesario, modifica las cantidades de material solicitado.
4. Autoriza solicitud de material.
5. Muestra mensaje de confirmación y se le notifica al almacenista

Flujo alternativo de eventos:

1. El administrador cancela solicitud de material.
2. Muestra mensaje de notificación.

Post-condiciones:

El administrador imprime el vale de material.

3.3.4. Registrar Material

Caso de uso: Registrar Material.

Actores: Administrador.

Descripción: Cuando llega material nuevo al almacén o existe material que se desea actualizar, el administrador registra el material y lo pone en existencia en el sistema o si ya existe, el administrador lo puede actualizar.

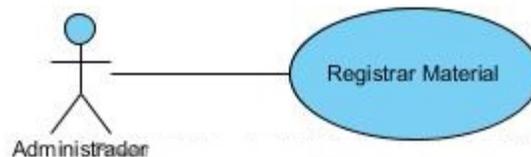


Fig. 3.4 – Detallado caso de uso registro material.

Precondiciones:

Se tienen todos los datos del material a registrar o actualizar.

Suposiciones:

El material no existe en el almacén o puede ser escaso.

Flujo normal de eventos:

1. El administrador abre la ventana de registro.
2. Introduce los datos del material a registrar.
3. El administrador valida los datos introducidos.

4. El sistema comprueba que el nombre introducido para el producto nuevo no existe en el sistema y que ningún otro campo imprescindible este vacío.
5. El sistema muestra un mensaje de registro exitoso, almacenando los datos introducidos.

Flujo alternativo de eventos:

1. *El material ya se encuentra dado de alta en el sistema.*
2. *Muestra mensaje de error.*

Post-condiciones:

El material ya se encuentra registrado en el sistema, disponible y visible para su solicitud del usuario.

3.3.5. Registrar material entregado

Caso de uso: Registrar material entregado.

Actores: Almacenerista.

Descripción: Una vez entregado el material con el vale aprobado, se hace un registro de dicho material para actualizar el nuevo número de material en existencia en almacén.

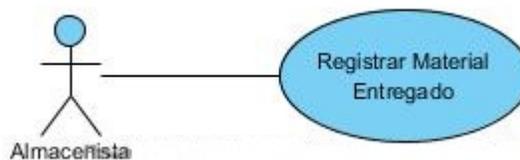


Fig. 3.5 – Detallado caso de uso registrar material entregado.

Precondiciones:

1. *La solicitud del material fue aprobada por el administrador.*
2. *Hay material en existencia.*

Suposiciones:

La solicitud fue aprobada.

Flujo normal de eventos:

1. *El almacenista abre la ventana de actualización del material.*
2. *Actualiza número de existencias en almacén del material entregado.*
3. *Muestra mensaje de actualización exitosa.*

Post-condiciones:

Se actualiza el registro de material exitosamente.
Se hace entrega del material.

3.3.6. Cerrar sesión

Caso de uso: *cerrar sesión.*

Actores: usuario, administrador y *almacenista.*

Descripción: *el usuario finaliza la sesión abierta en el sistema.*

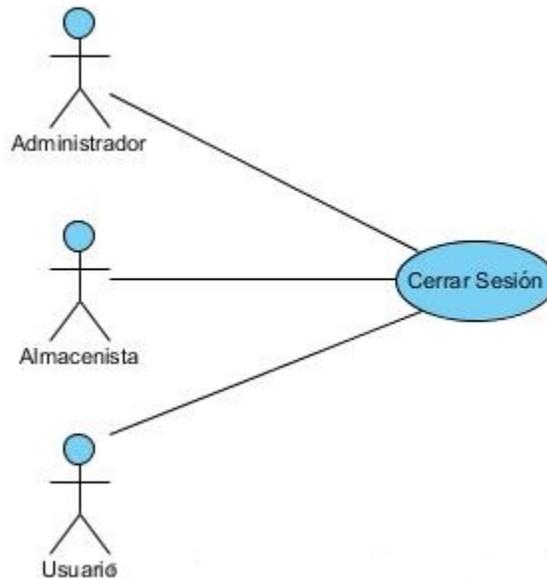


Fig. 3.6 – Detallado caso de uso cerrar sesión.

Precondiciones:

El usuario se encuentra con su sesión abierta en el sistema.

Suposiciones.

El usuario quiere cerrar su sesión o cambiar de usuario.

Flujo normal de eventos.

1. *El usuario valida el cierre de sesión.*
2. *El sistema cierra la sesión abierta.*
3. *Muestra mensaje de cierre de sesión exitoso.*

Post-condiciones:

La sesión se cierra correctamente para que otro usuario inicie sesión.

3.4. Requerimientos no funcionales.

1. El sistema manejará la información de manera organizada y eficiente.
2. Tendrá seguridad para manejar la información.
3. Agilizará y facilitará la solicitud de abastecimiento.
4. Tiempo de respuesta óptimo entre el usuario, almacenista y administrador.
5. Interfaz gráfica amigable para el usuario final.
6. Será un sistema web.
7. Estará disponible para todos los usuarios registrados de la unidad de posgrado.
8. El sistema será escalable.
9. Se le podrá dar mantenimiento.

3.5. Prototipos.

A partir de los requerimientos del capítulo anterior, podemos detallar los prototipos. Un prototipo es la representación gráfica de nuestro sistema a desarrollar, de forma que podamos transmitir mejor la idea al usuario final de lo que se quiere implementar.

A continuación se muestran los prototipos o pantallas que tendrá cada tipo de usuario que utilice el sistema.

3.5.1. Prototipos de administrador.

El usuario con permisos de administrador, podrá acceder al sistema a través de una vista de inicio de sesión, la cual se muestra en la Figura 3.7. En dicha vista realizará la consulta a la base de datos y, si los datos son correctos, el sistema direccionará a la vista que se muestra en la Figura 3.8.

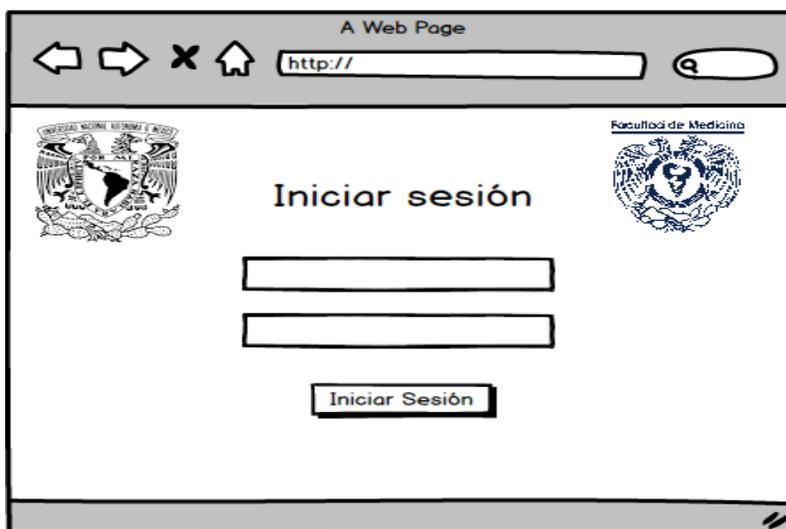


Fig. 3.7 – Prototipo de inicio de sesión para el administrador.

Una vez iniciada la sesión, se mostrará el menú principal (ver Figura 3.8) en el cuál usuario con permisos de administrador podrá realizar distintas acciones sobre el material y las solicitudes de dicho material. A continuación se muestra el prototipo del menú principal del administrador:

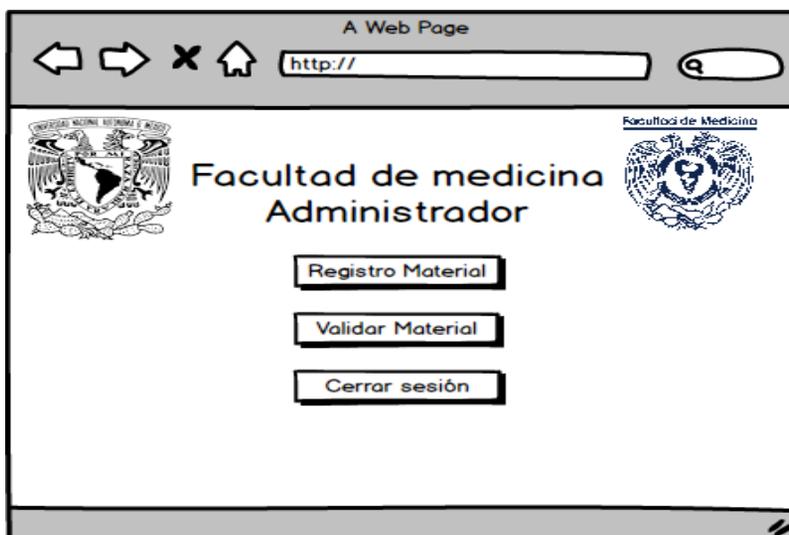


Fig. 3.8 – Prototipo menú principal administrador.

Si el usuario oprime la opción de “Registro de material”, el sistema direccionará a la vista mostrada en la Figura 3.9.

En esta vista, el usuario podrá realizar distintas acciones sobre el material, entre ellas podrá registrar, actualizar y eliminar dicho material. A continuación se muestra el prototipo de registro de material.

A Web Page
http://

Facultad de Medicina

Registro Material

Nombre

Descripción

Cantidad

Código

Cancelar Registrar

Fig. 3.9 – Registro de material.

Una vez realizada la acción sobre el material, el sistema direccionará a la vista que mostrará el aviso de confirmación según la acción que se haya hecho (Figura 3.10), en este caso mostrará una ventana anunciando que el registro fue exitoso.

A Web Page
http://

Facultad de Medicina

Facultad de medicina Administrador

Registro Exitoso

Aceptar

Fig. 3.10 – Registro de material exitoso

Otro tipo de advertencia o anuncio que puede surgir, es que ya existe material registrado en el almacén, como se muestra a continuación en la Figura 3.11:

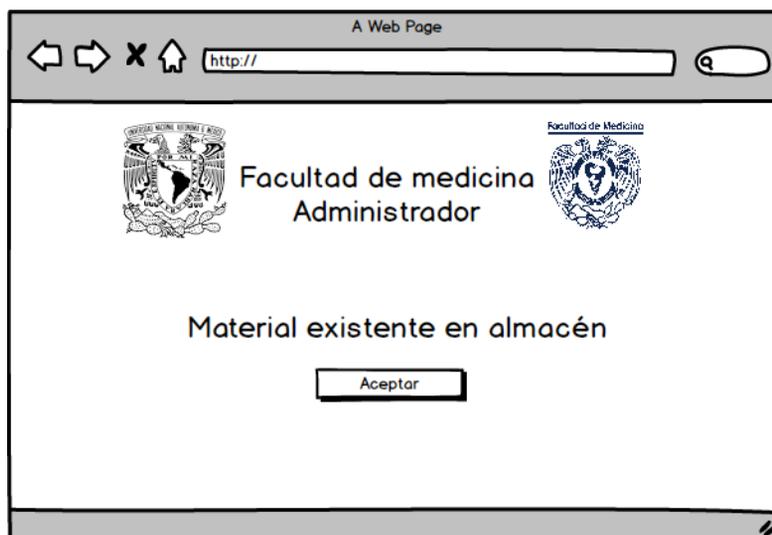


Fig. 3.11 – Notificación material existente

Si el usuario oprime la opción de “validar material” en la Figura 3.8, el sistema direccionará a la vista para buscar algún “vale” para poder validarlo o poder realizar alguna búsqueda de registros. En la Figura 3.12 se muestra la vista para poder realizar búsquedas en los registros de la base de datos.



Fig. 3.12 – Búsqueda de registros

Una vez realizada la búsqueda, el sistema mostrará una serie de resultados que coincidan con la palabra o frase a consultar. A continuación se muestra una ventana con los resultados de dicha búsqueda (ver Figura 3.13)

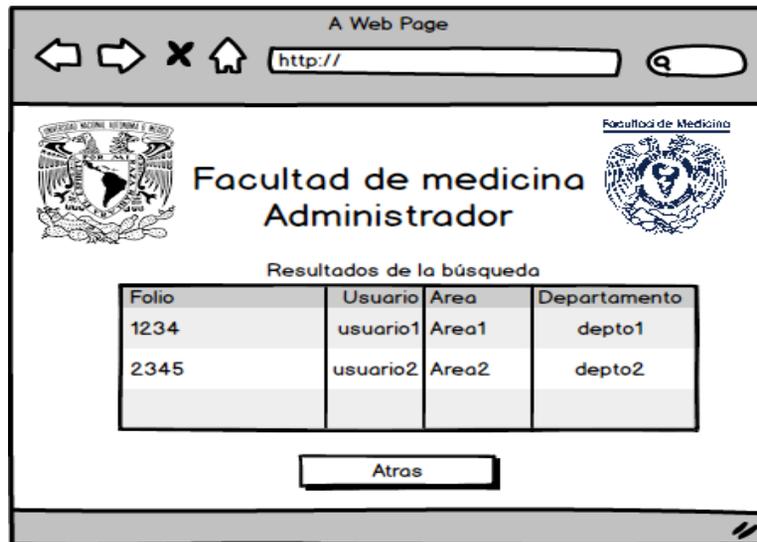


Fig. 3.13 – Resultado de la búsqueda

De la misma manera, en la búsqueda se podrán hacer diferentes tipos de consultas, entre ellas, se podrá consultar las solicitudes nuevas que aún no se han aprobado. En la Figura 3.14 se muestra una vista con este tipo de consulta.

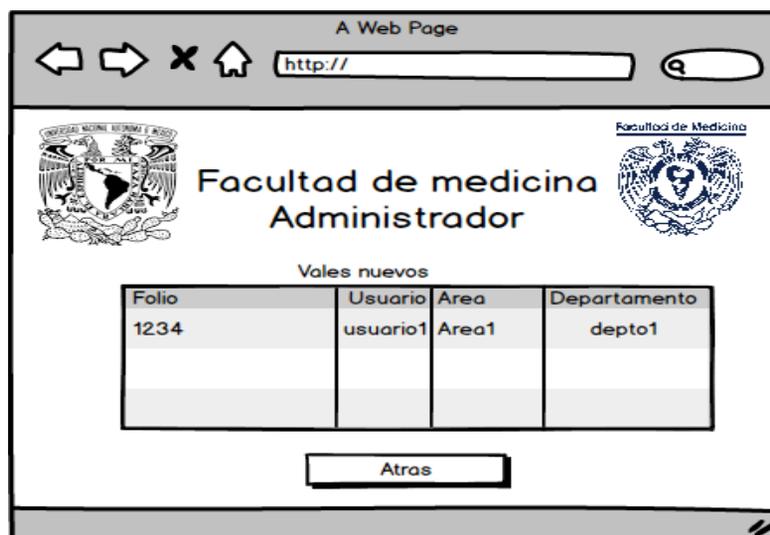


Fig. 3.14 – Vales nuevos o pendientes.

En cualquier consulta que se realice, se podrá consultar el detalle de cualquier solicitud, ya sea que esté rechazada, aprobada o en proceso. En el detalle se mostrarán los datos de la persona solicitante, el área a la que pertenece y la fecha en que fue solicitado el material, entre otros. De la misma manera se podrá consultar

en el detalle del vale, la cantidad y el tipo de productos, así como se podrán consultar y modificar las cantidades solicitadas de cada tipo de material. A continuación en la Figura 3.15, se muestra una vista para el usuario que consulta el detalle de alguna solicitud.

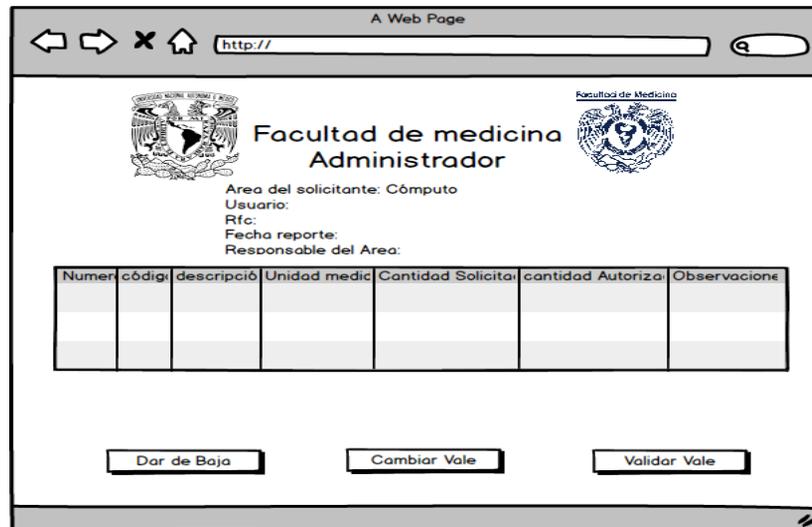


Fig. 3.15 – Detalle de vale de solicitud

Dentro del detalle de la solicitud, se podrán realizar distintas acciones. Entre ellas se podrá dar de baja, cambiar o validar alguna solicitud de material (vale). Dependiendo del tipo de acción que se haya realizado en el detalle de la solicitud, el sistema nos notificará que se ha realizado la acción correctamente. A continuación se muestra un ventana cuando se ha aprobado el vale (ver Figura 3.16)

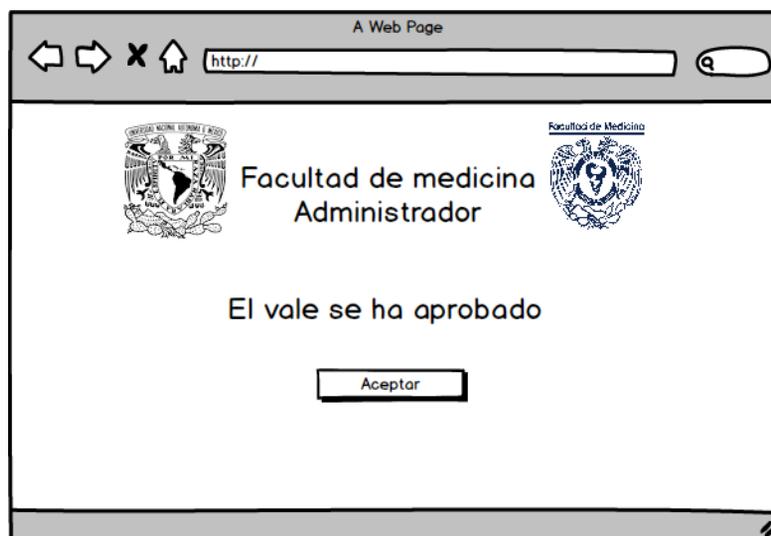


Fig. 3.16 – Notificación vale aprobado

De la misma manera, se pueden realizar otras acciones sobre el vale consultado, en este caso se muestra una vista en la cual se notifica que el vale ha sido cancelado, ver Figura 3.17.

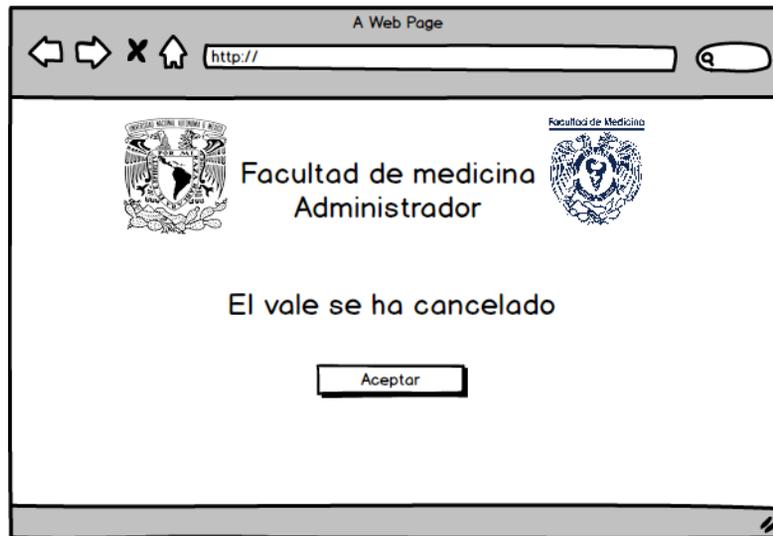


Fig. 3.17 – Notificación vale cancelado

Además, se podrán consultar los vales según el área que los ha solicitado o departamento. Se muestra un ejemplo en la Figura 3.18

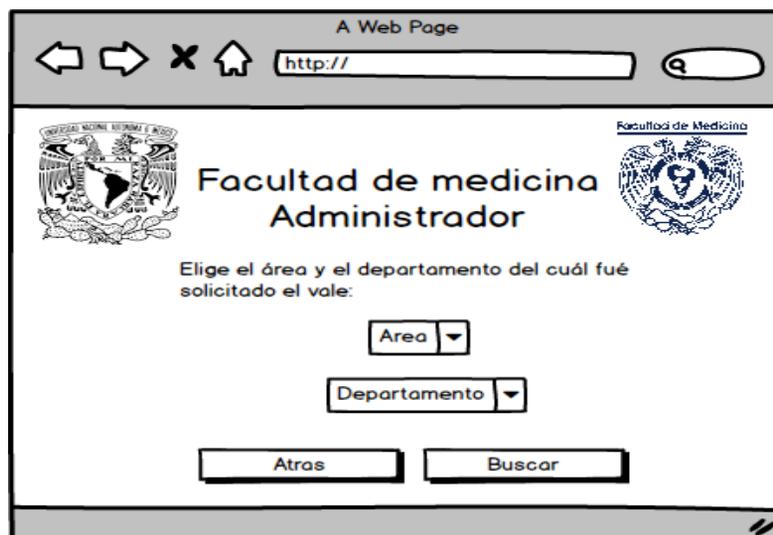


Fig. 3.18 – Tipos de consultas

Después de haber realizado las operaciones deseadas para el manejo del material y de las solicitudes, el usuario podrá cerrar la sesión de administrador. Se le notificará que la sesión ha sido cerrada correctamente (ver Figura 3.19).

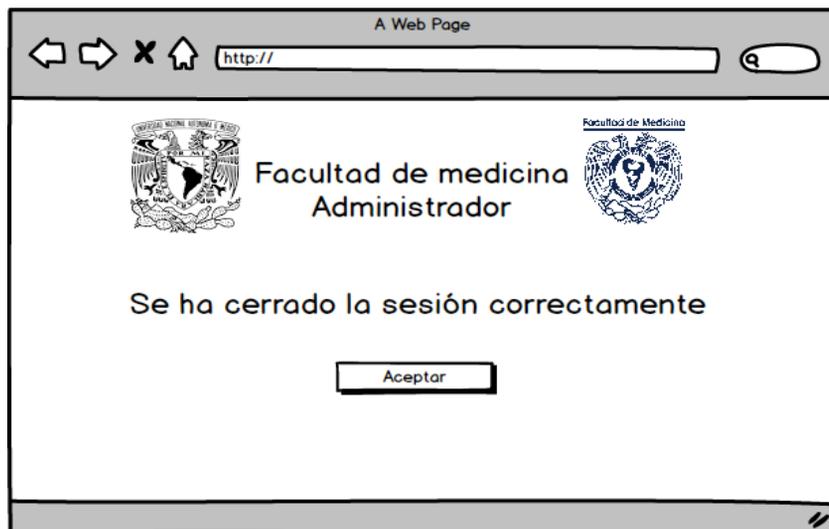


Fig. 3.19 – Sesión cerrada correctamente.

3.5.2. Prototipos de usuario.

De la misma forma que con el administrador, el actor con permisos de usuario solicitante de material, podrá acceder al sistema a través de una vista de inicio de sesión, la cual se muestra en la Figura 3.20. En dicha vista realizará la consulta a la base de datos y, si los datos son correctos, el sistema direccionará a la vista que se muestra en la Figura 3.21.



Fig. 3.20 – Prototipo de inicio de sesión para el administrador.

Cuando el usuario tiene un inicio de sesión exitoso, el sistema lo redireccionará al menú principal, en el cuál podrá cerrar sesión o solicitar algún tipo de material existente en el almacén. A continuación se muestra la vista del menú principal del usuario (ver Figura 3.21).

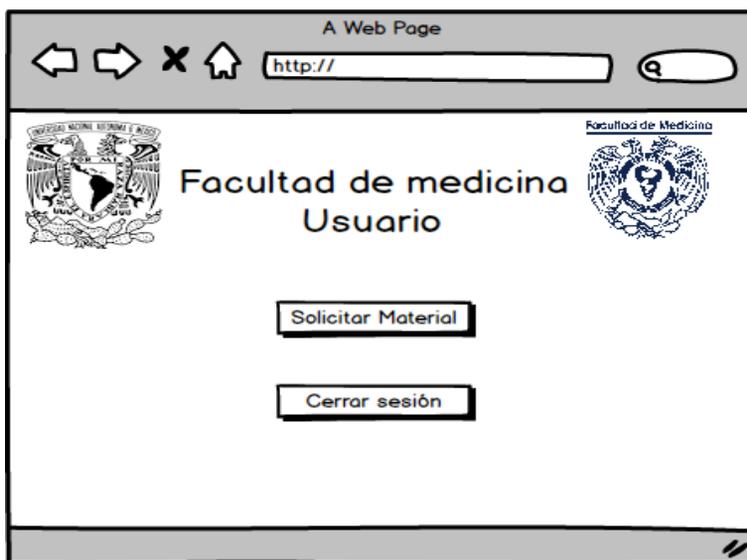


Fig. 3.21 – Menú principal usuario

Si el usuario oprime la opción “Solicitar material” de la Figura 3.21, se le mostrará una tabla en la cual se le mostrará todo el material disponible en el almacén para poder realizar su solicitud. En esta vista el usuario podrá elegir el material y la cantidad que desee de dicho material. En la vista 3.22 se muestra la manera en que el usuario podrá seleccionar el material disponible de una lista.

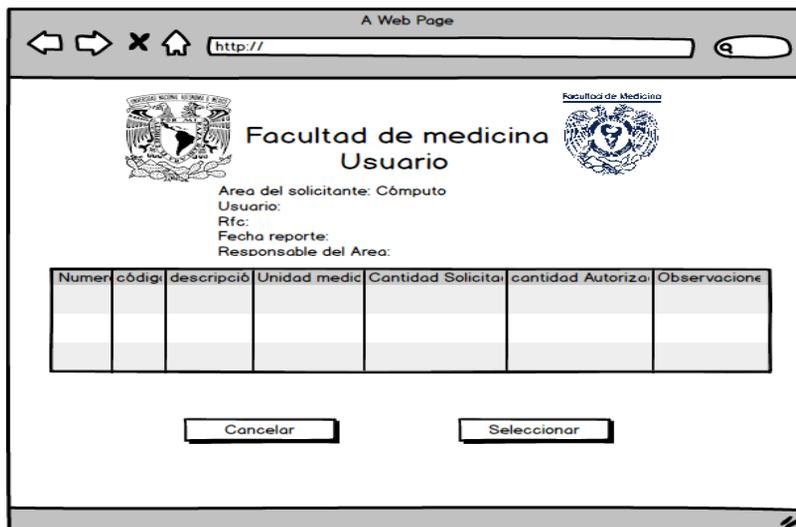


Fig. 3.22 – Selección de material

Una vez realizada la solicitud en la Figura 3.22, se direccionará a una ventana la cuál mostrará el mensaje de que la solicitud ha sido realizada. En la Figura 3.23 se muestra la ventana correspondiente al mensaje de “solicitud exitosa”.

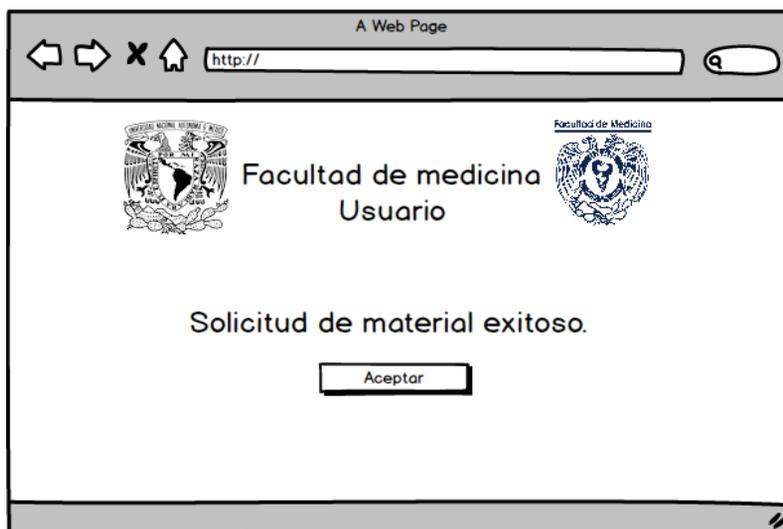


Fig. 3.23 – solicitud exitosa

Una vez que el usuario haya realizado las operaciones deseadas para las solicitudes del material, podrá cerrar la sesión. Se le notificará que la sesión ha sido cerrada correctamente. (ver Figura 3.24)

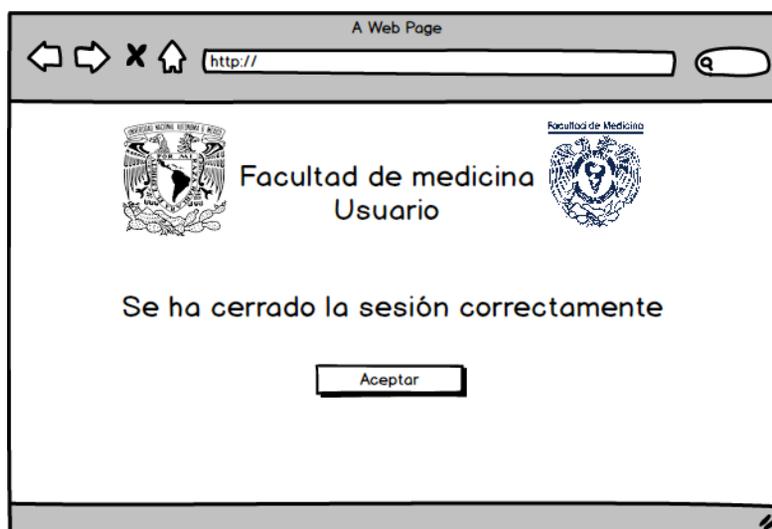


Fig. 3.24 – Sesión cerrada correctamente.

3.5.3. Prototipos de almacenista.

De la misma forma que con el administrador y el usuario solicitante de material, el actor con permisos de almacenista, podrá acceder al sistema a través de una vista de inicio de sesión, la cual se muestra en la Figura 3.25. En dicha vista realizará la consulta a la base de datos y, si los datos son correctos, el sistema direccionará a la vista que se muestra en la Figura 3.25.



Fig. 3.25 – Inicio de sesión

Una vez que el almacenista tiene un inicio de sesión exitoso, el sistema lo redireccionará al menú principal, en el cuál podrá registrar material entregado, buscar un material solicitado o cerrar sesión,. A continuación se muestra la vista del menú principal del almacenista. (ver Figura 3.26)

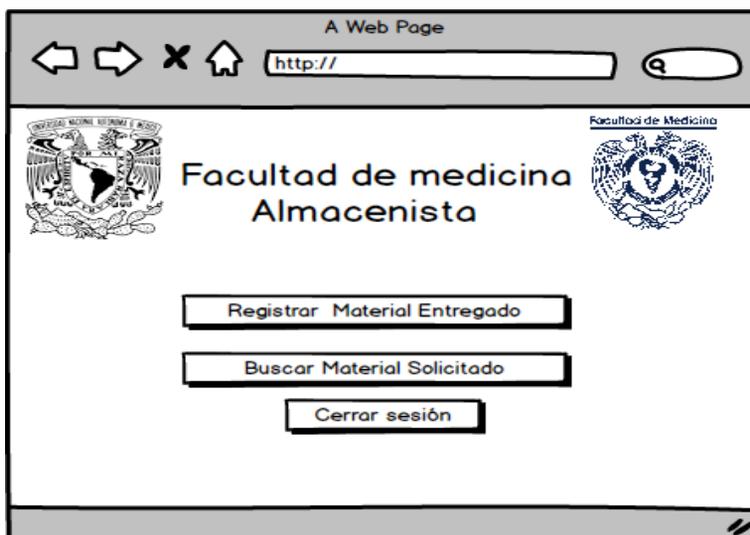


Fig. 3.26 – Menú principal almacenista

Cuando el almacenista selecciona la opción de “registrar material entregado” en el menú principal (ver Figura 3.26), se manda a la vista donde aparecen todos los vales que fueron aprobados para que se les entregue material, una vez aquí el usuario podrá ver el detalle de cada vale aprobado. A continuación se muestra la vista de los vales aprobados para poder ser despachados.

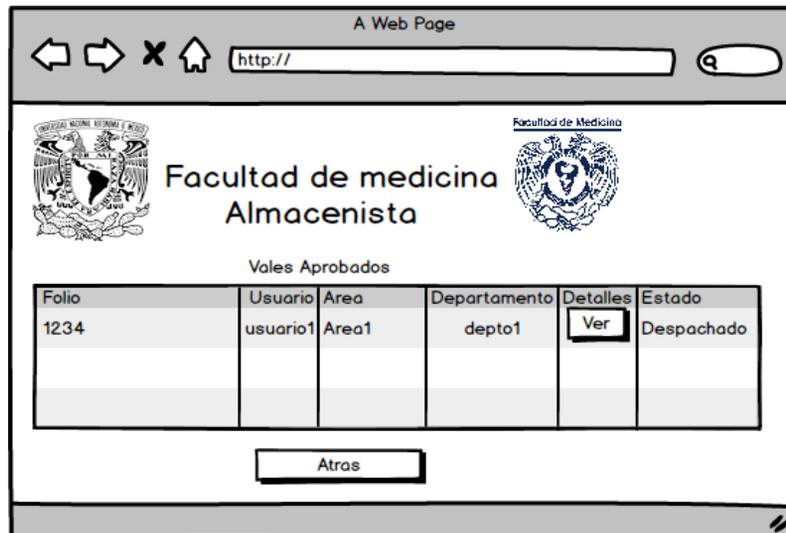


Fig. 3.27 – Lista de vales aprobados

Después de la vista de la Figura 3.37, el almacenista podrá seleccionar algún detalle de las solicitudes para poder ver que material fue autorizado para ser entregado. En este detalle le aparecerá el nombre del material, la descripción, cantidad solicitada, autorizada y algunas observaciones del administrador. Aparecerán los datos de la persona solicitante, el área y la fecha en que fue solicitado el material. A continuación se muestra la vista del detalle de la solicitud (ver Figura 3.28)

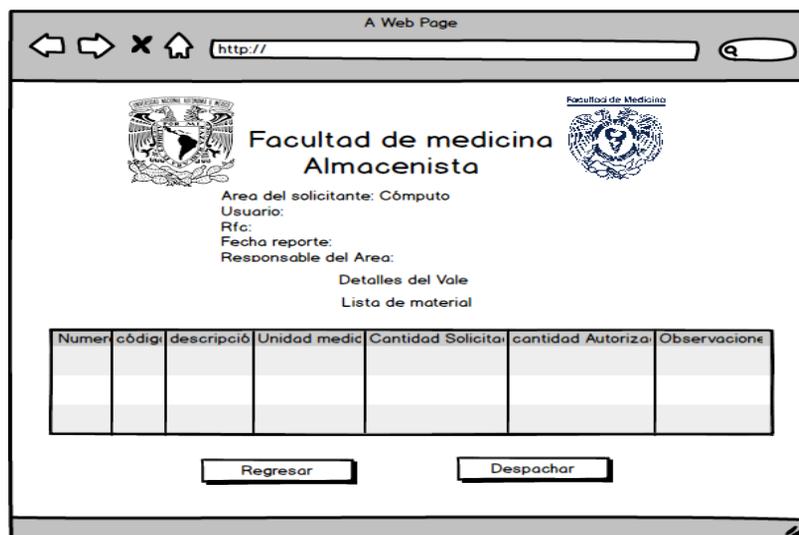


Fig. 3.28 – Detalle de la solicitud

Una vez que el material fue entregado al solicitante, el almacenista podrá poner en el sistema que la solicitud fue atendida mediante el botón “despachar” de la Figura 3.28. Una vez despachada la solicitud, el sistema informará que la solicitud ha sido despachada, como se muestra en la Figura 3.29.

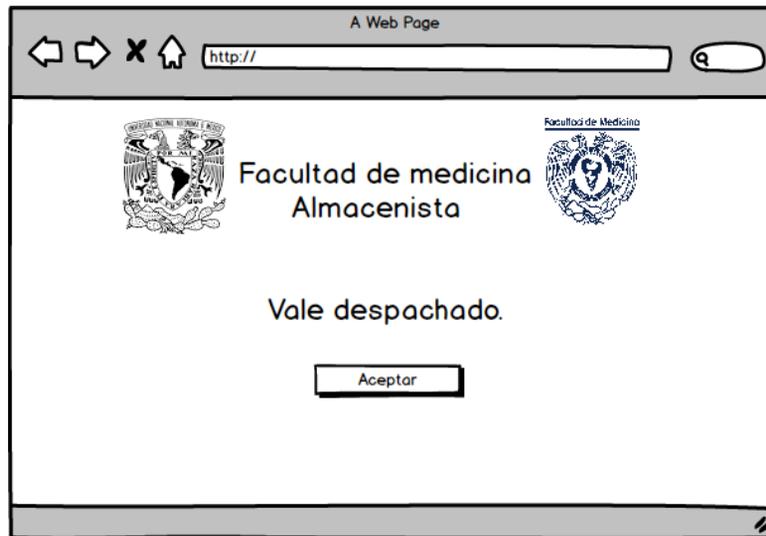


Fig. 3.29 – Notificación de vale despachado.

Una vez que nuestro vale ha sido despachado, podemos regresar al menú principal del almacenista (ver Figura 3.26), aquí también se podrán hacer búsquedas de material solicitado para una poder atender de una manera más eficiente una solicitud. Esto se hace mediante la función de “Buscar material solicitado”. Esto direccionará a la siguiente vista (ver Figura 3.30).



Fig. 3.30 – Búsqueda material solicitado

En esta vista podremos buscar cualquier material o solicitud ya sea por folio, nombre, fecha de solicitud, etc. Una vez encontrado el material o la solicitud, se mostrará una ventana con todos los detalles de la búsqueda. Dicha ventana se muestra en la Figura 3.31

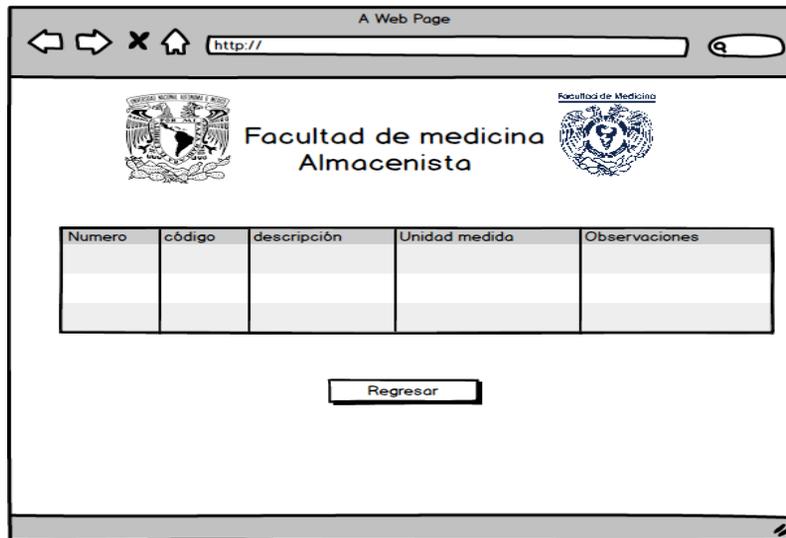


Fig. 3.31 – Detalle de la solicitud.

Una vez consultados los detalles, se podrá regresar al menú principal (ver Figura 3.26)

Una vez ya realizadas todas las acciones disponibles para el almacenista, en el menú principal podremos cerrar la sesión. Esto llevará a una vista que mostrará el aviso de que la sesión se ha cerrado correctamente, como se muestra a continuación.

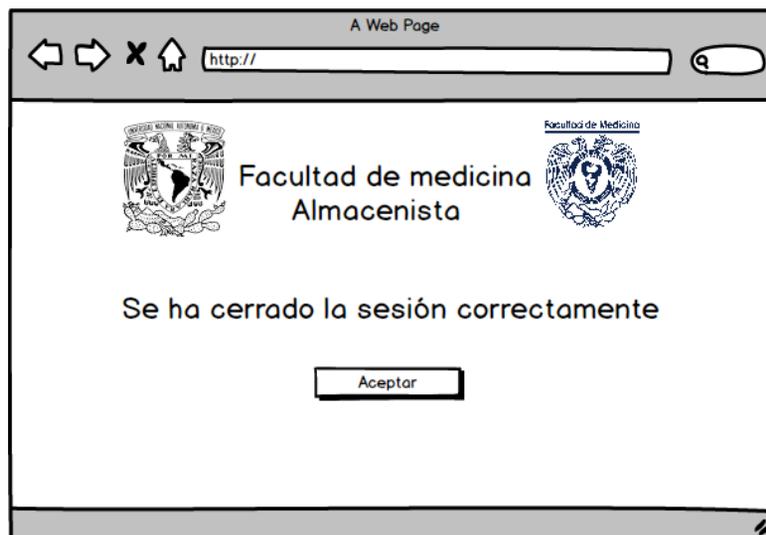


Fig. 3.32 – Sesión cerrada correctamente.

4. ADMINISTRACIÓN DE LA CONFIGURACIÓN

4.1. Definición de la línea base.

A partir de cada punto establecido como línea base, cualquier cambio posterior que se realice causará una modificación a la versión sobre la cual se está trabajando y con esto se podrá hacer una comparación entre versiones para utilizarlos en futuros cambios. Hasta el momento tendremos la línea base definida en los siguientes flujos de trabajo:

1. Administración de la configuración
2. Análisis de requerimientos
3. Análisis
4. Diseño
5. Implementación
6. Pruebas

4.2. Definición de los componentes a controlar.

Los archivos que se guardarán en el depósito del proyecto son los que se definen a continuación agrupados en los flujos de trabajo anteriormente mencionados:

- 1. Administración de la configuración**
 - 1.1. Definición de las líneas base
 - 1.2. Definición de los componentes a controlar
 - 1.3. Creación del depósito del proyecto
 - 1.4. Características del equipo
 - 1.5. Definición de control de cambios
 - 1.6. Solicitud de cambios
- 2. Análisis de requerimientos**
 - 2.1. Problemática
 - 2.2. Diagrama general de casos de uso
 - 2.3. Requerimientos funcionales
 - 2.4. Requerimientos no funcionales
 - 2.5. Prototipos
 - 2.6. Glosario de términos
- 3. Análisis**
 - 3.1. Diagramas de clase
 - 3.2. Control
 - 3.3. Integración
 - 3.4. Diagramas de estado
 - 3.5. Diagramas de secuencia

3.6. Casos de uso

4. Diseño

4.1. Descripción de la arquitectura del diseño

4.2. Diagramas de paquetes

4.3. Diagrama de distribución

4.4. Diagrama de clases del diseño detallados

4.5. Diseño de la base de datos

5. Implementación

5.1. Construcción de la interfaz

5.2. Código de clases

5.3. Pruebas unitarias

5.4. Reporte de pruebas unitarias

6. Pruebas

6.1. Reporte de pruebas de integración

6.2. Reporte de pruebas del sistema

4.3. Creación del depósito del proyecto

El depósito del proyecto tendrá la siguiente estructura:

C:\aplicaciones\sistemas\almacen\

→ administracionconfiguracion

→ analisisrequerimientos

→ análisis

→ diseño

→ implementación

→ pruebas

4.4. Características del equipo

Las características del equipo donde será alojado el proyecto son las siguientes:

- marca: Dell

- modelo: Poweredge 4900

- sistema operativo: Windows Server 2008 r2

- procesador: Xeon

- memoria RAM: 32 GB

- 2 discos duros de 1 TB cada uno

4.5. Definición de control de cambios.

4.5.1. Procedimiento de cambios y revisiones.

Cuando se solicite algún cambio o actualización del sistema, se realizará de la siguiente manera:

El responsable del proyecto realizará una solicitud de cambio en la cual se especificarán los cambios solicitados, ya sea esta petición a causa de haber encontrado algún defecto durante la revisión en el sistema o por alguna petición del usuario o el responsable del proyecto.

Forma de solicitud de cambios	
Proyecto:	fecha:
Solicitado por:	
Información del elemento	
Nombre:	
Dirección de respaldo:	
Responsable:	
Información del cambio:	
Descripción del cambio:	
Beneficio del cambio:	
Impacto del cambio:	
Estado del cambio:	
Fecha de dictamen:	
Aprobado	Rechazado

4.5.2. Informe del estado de la configuración.

A continuación se muestran la ruta de los archivos de configuración del proyecto:

Administración de la configuración:

C:\aplicaciones\sistemas\almacen\administracionconfiguracion\administracion_de_la_configuracion.pdf

Análisis de requerimientos:

C:\aplicaciones\sistemas\almacen\ analisisrequerimientos\ analisis_de_requerimientos.pdf

Diseño:

C:\aplicaciones\sistemas\almacen\diseño\

Implementación:

C:\aplicaciones\sistemas\almacen\implementación\

Pruebas:

C:\aplicaciones\sistemas\almacen\pruebas\

5. DISEÑO

Con la información obtenida en la etapa de análisis, desarrollaremos un modelo del sistema en el cuál se reflejarán las especificaciones o funcionalidades de nuestro producto final a través de diagramas. Describiremos el comportamiento esperado del software y su interacción con el usuario final.

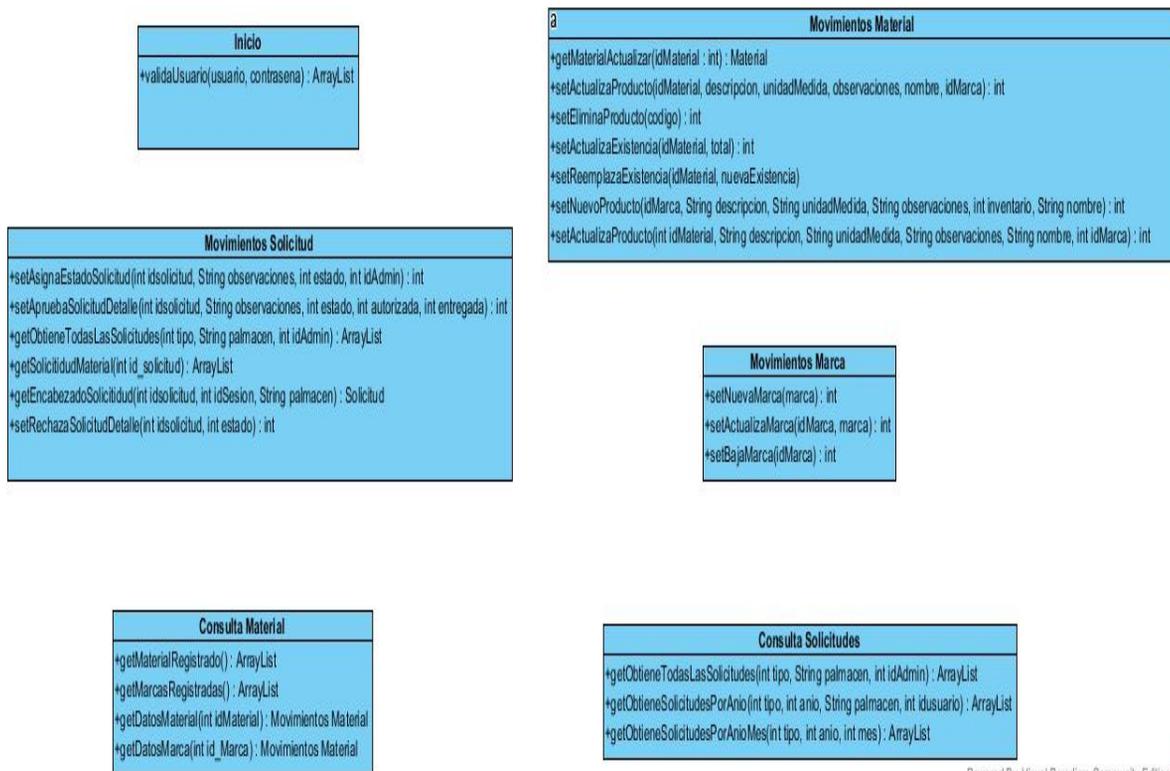
Para esto, definimos distintos tipos de esquemas y diagramas que complementan y describen de manera más específica las funcionalidades de nuestro software. Los diagramas se definirán para cada tipo de actor de acuerdo a sus funciones. Los diagramas (por rol) que se definirán, son los siguientes:

- Diagramas de control.
- Diagramas de modelo.
- Diagramas de estado.
- Diagramas de secuencia.

5.1. Diagramas del administrador

5.1.1. Control.

A continuación se muestran los diagramas de control del administrador, los cuales describen los métodos que se mandan a llamar para cada función disponible para el administrador, como por ejemplo: Alta, baja y actualización de material, registro de marcas, movimientos de una solicitud, entre otros. En la Figura 5.1 se muestra dicho diagrama.

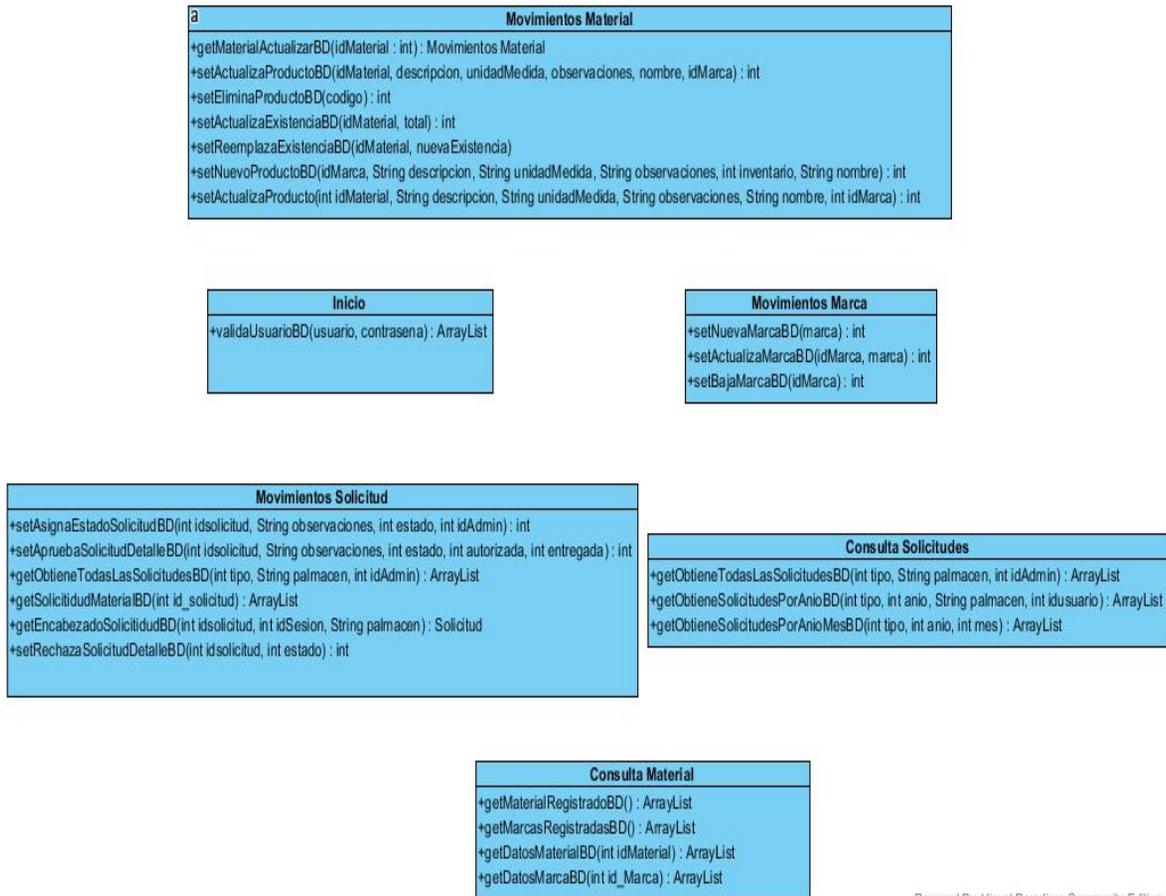


Powered By Visual Paradigm Community Edition

Fig. 5.1 – Diagrama control de administrador.

5.1.2. Modelo

En el siguiente diagrama, se muestran los métodos que se ejecutan para la interacción con la base de datos las funciones disponibles para el administrador. Aquí el administrador puede realizar inserciones, actualizaciones y borrar registros en la base de datos. En la Figura 5.2 se muestra dicho diagrama y sus funciones correspondientes.



Powered By Visual Paradigm Community Edition

Fig. 5.2 – Diagrama modelo de administrador

5.1.3. Estado

En la siguiente figura se muestra el posible flujo del administrador en el sistema, es decir, los caminos que podrá tomar para realizar distintas funciones. En la Figura 5.3 se muestra dicho flujo o posibles caminos del administrador en el sistema.

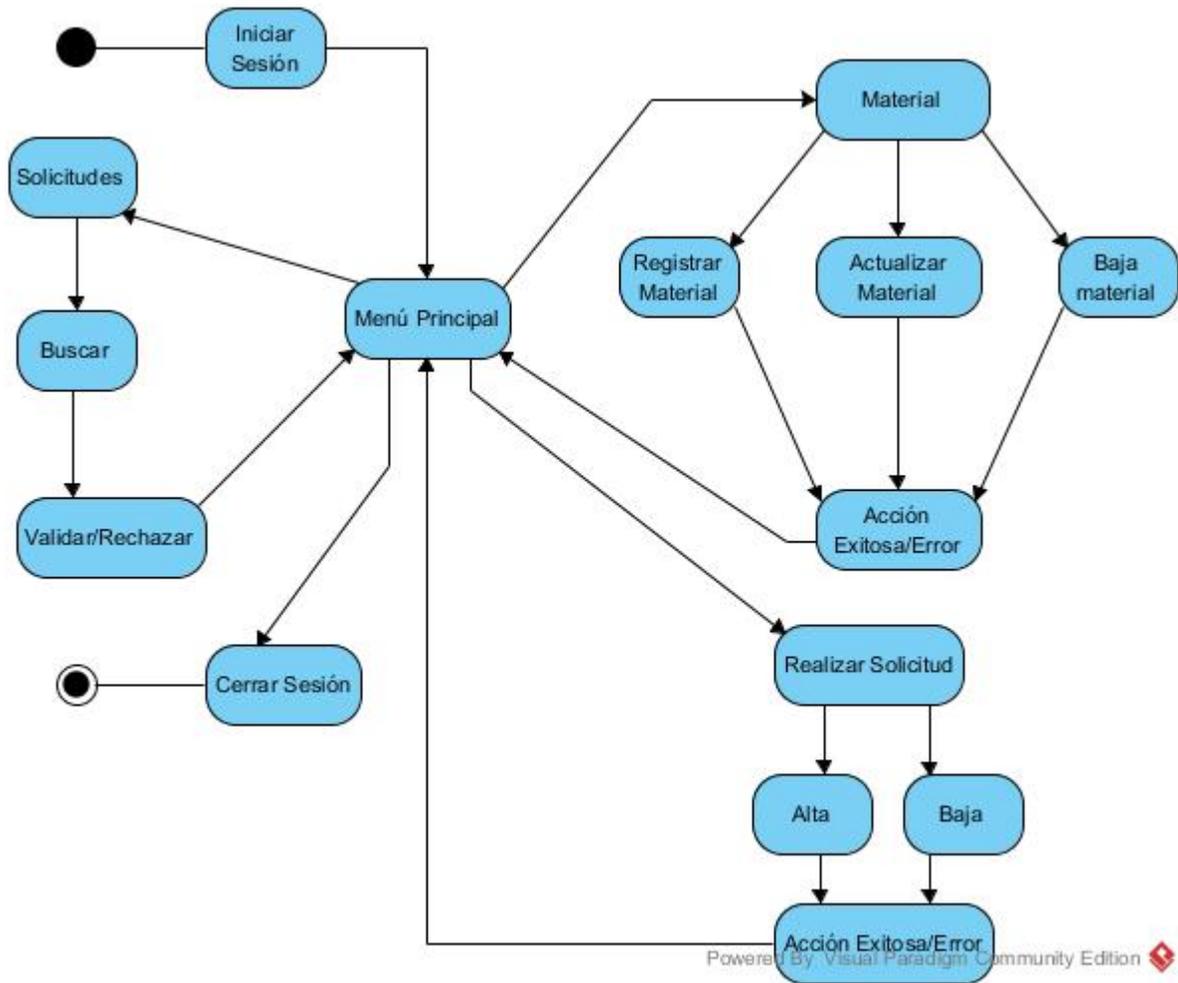


Fig. 5.3 – Diagrama estado de administrador

5.2. Diagramas del almacenista

5.2.1. Control

A continuación se muestran los diagramas de control para el almacenista, los cuales describen los métodos que se mandan a llamar para cada función disponible para el almacenista, como por ejemplo: Consulta de las solicitudes por despachar, datos de las solicitudes, entre otros. En la Figura 5.4 se muestra dicho diagrama.

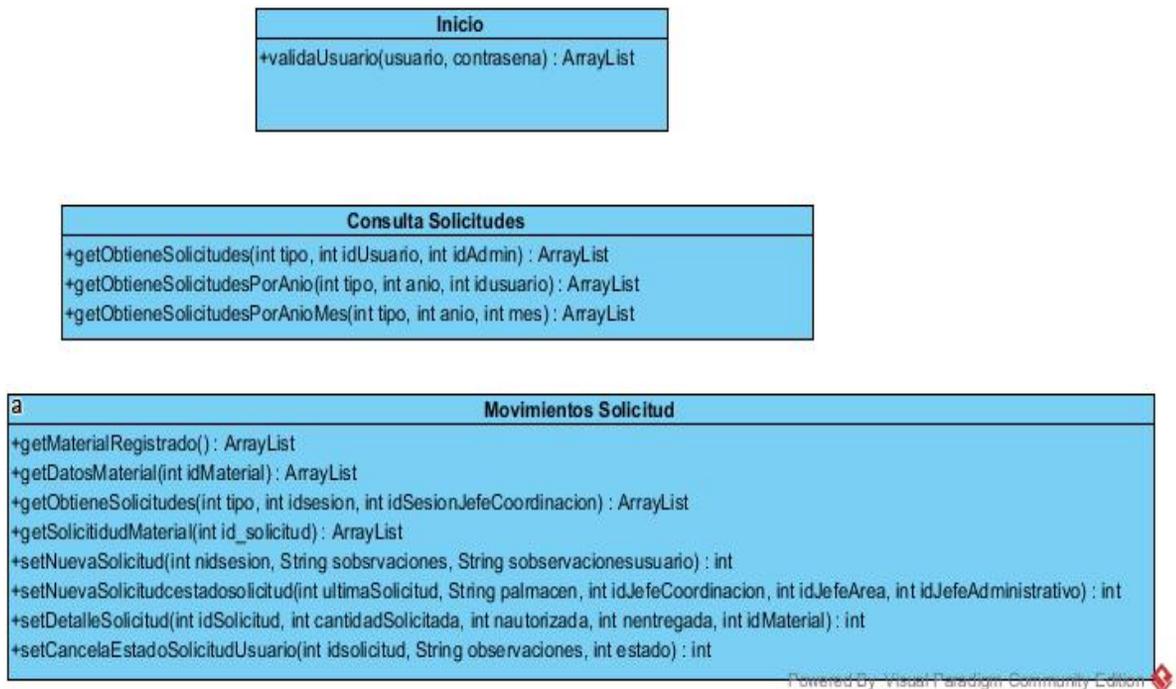


Fig. 5.4 – Diagrama control de almacenista

5.2.2. Modelo

En el siguiente diagrama, se muestran los métodos que se ejecutan para la interacción con la base de datos las funciones disponibles para el almacenista. Aquí el almacenista puede realizar búsqueda de solicitudes mediante consultas en la base de datos, actualizar solicitudes, entre otros. En la Figura 5.5 se muestra dicho diagrama y sus funciones correspondientes.

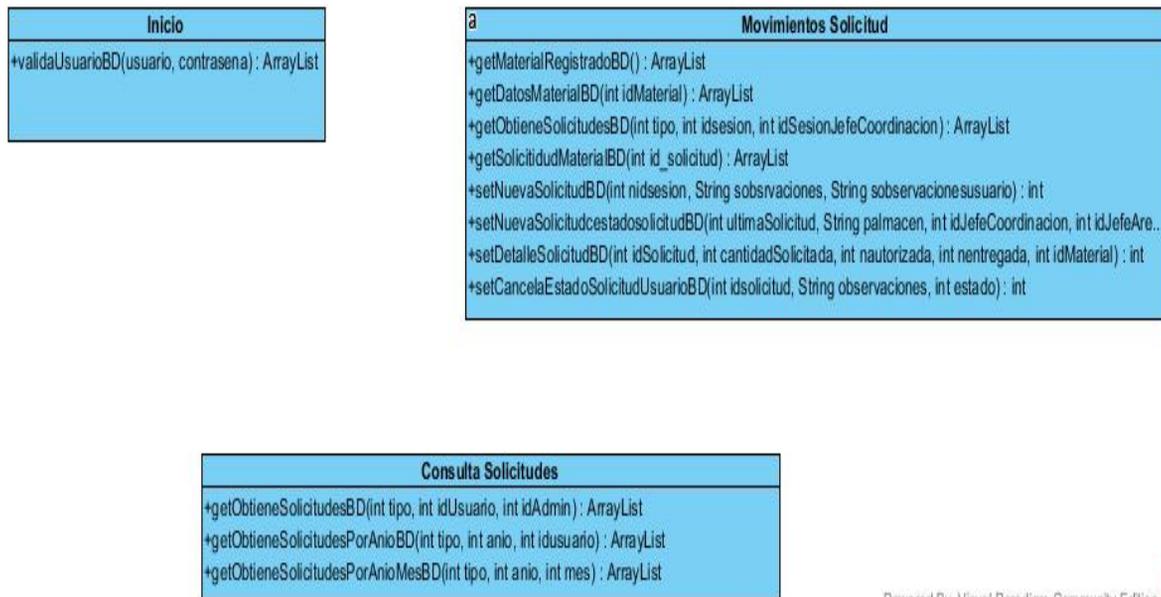


Fig. 5.5 – Diagrama integración de almacenista

5.2.3. Estado.

En la siguiente figura se muestra el posible flujo del almacenista en el sistema, es decir, los caminos que podrá tomar para realizar distintas funciones. En la Figura 5.6 se muestra dicho flujo o posibles caminos del almacenista en el sistema.

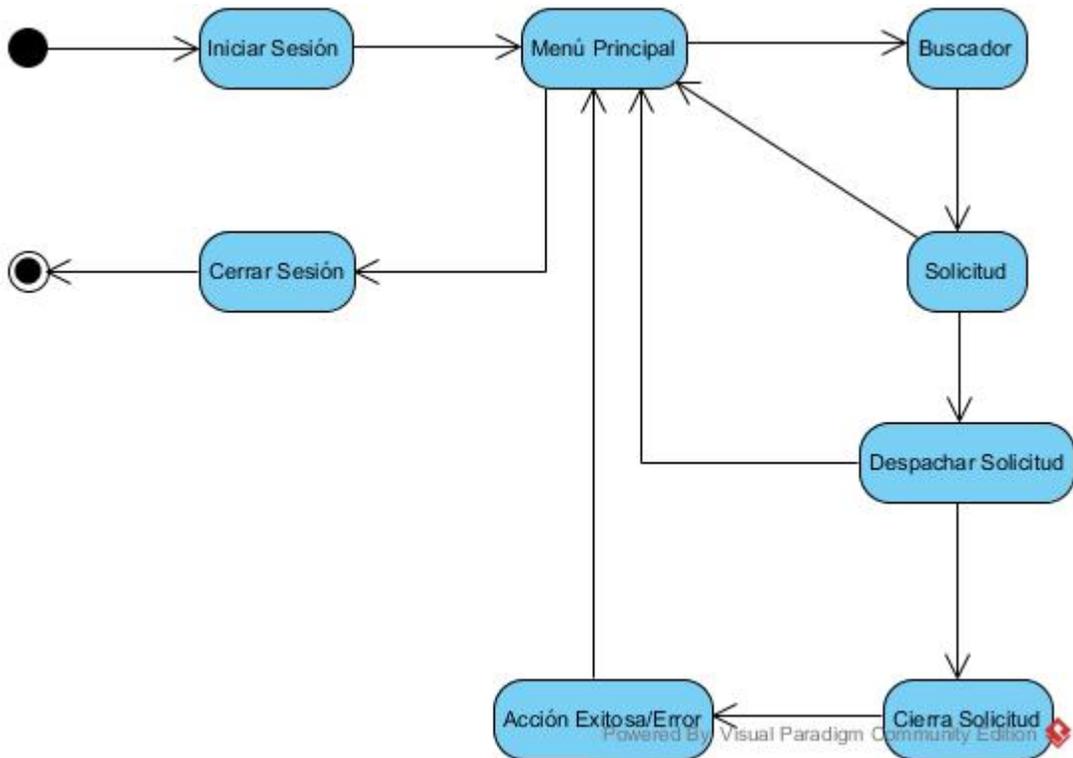


Fig. 5.6 – Diagrama estado de almacenista.

5.3. Diagramas del Usuario.

5.3.1. Control

A continuación se muestran los diagramas de control para el usuario, los cuales describen los métodos que se mandan a llamar para cada función disponible para el usuario, como por ejemplo: Consulta del material disponible, realizar solicitud, entre otros. En la Figura 5.7 se muestra dicho diagrama.

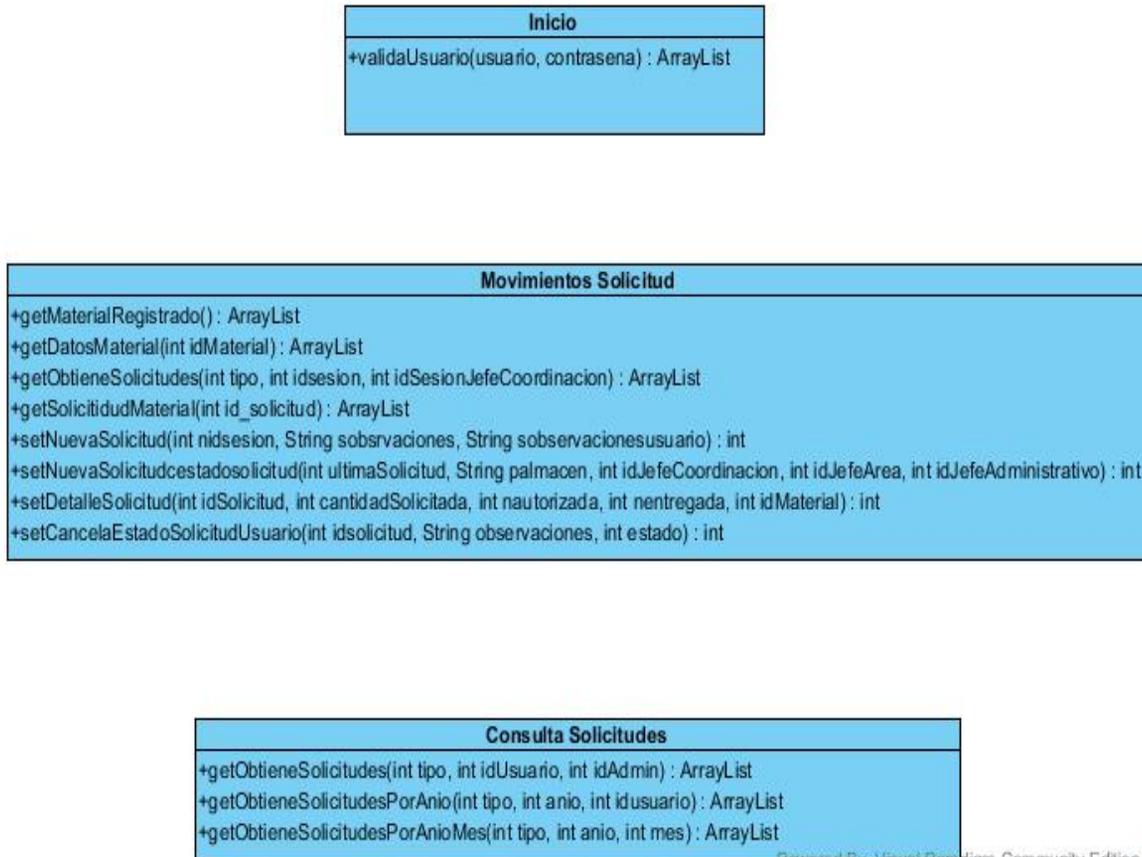


Fig. 5.7 Diagrama control de usuario

5.3.2. Modelo

En el siguiente diagrama, se muestran los métodos que se ejecutan para la interacción con la base de datos las funciones disponibles para el usuario. Aquí el usuario puede realizar búsqueda de material disponible mediante consultas en la base de datos, cancelar una solicitud realizada, entre otros. En la Figura 5.8 se muestra dicho diagrama y sus funciones correspondientes.

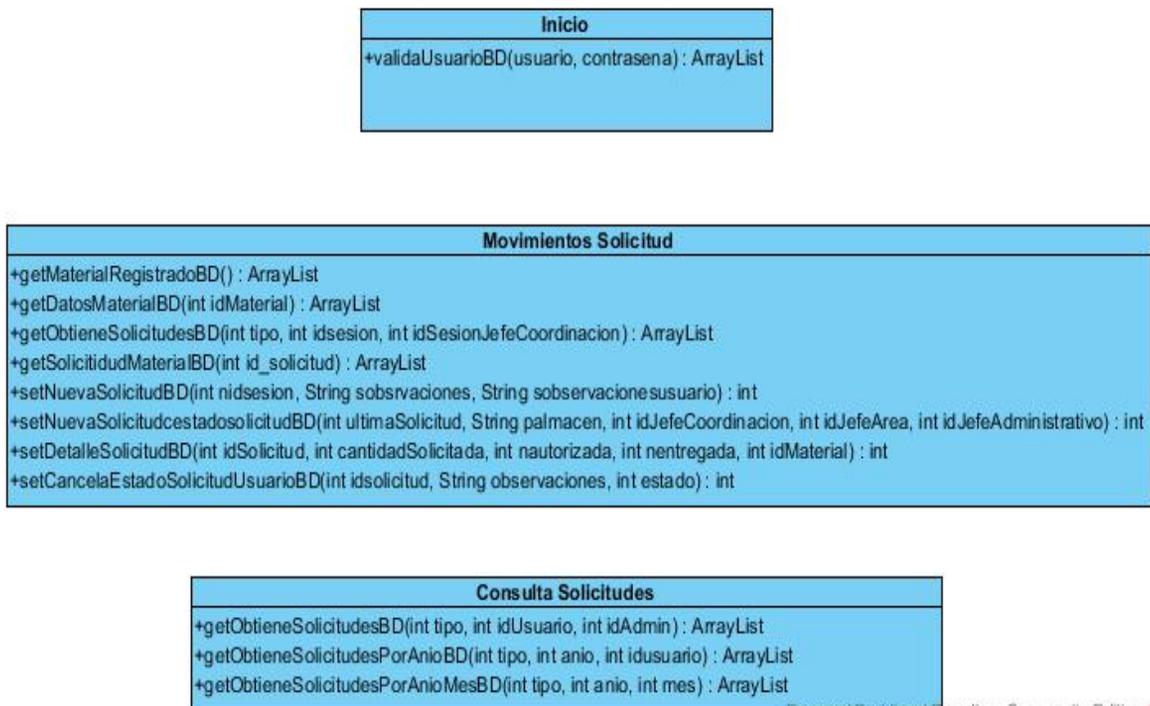


Fig. – 5.8 Diagrama modelo de usuario

5.3.4. Estado.

En la siguiente figura se muestra el posible flujo del usuario en el sistema, es decir, los caminos que podrá tomar para realizar distintas funciones. En la Figura 5.9 se muestra dicho flujo o posibles caminos del usuario en el sistema.

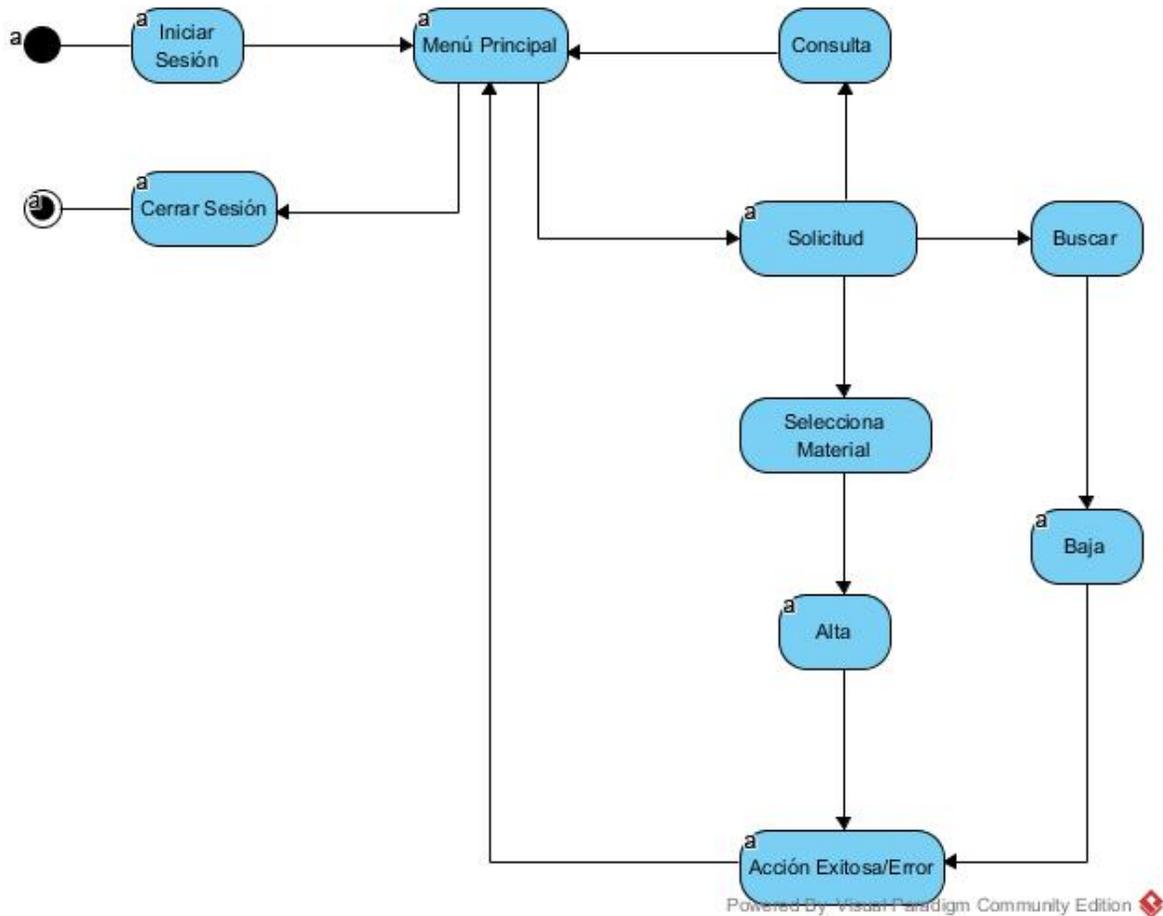


Fig. 5.9 Diagrama estado de usuario

5.4. Diagramas de secuencia

5.4.1. Caso de uso: Iniciar sesión.

En la figura 5.10 se muestra el diagrama de secuencia para el caso de uso de Iniciar Sesión. Dicho diagrama muestra el proceso mediante el cual el actor se autentica en el sistema, la interacción con la base de datos y los resultados que se muestran en pantalla dependiendo si la autenticación fue exitosa o errónea. Los actores que se modelarán en este diagrama son: Administrador, almacenista y usuario.

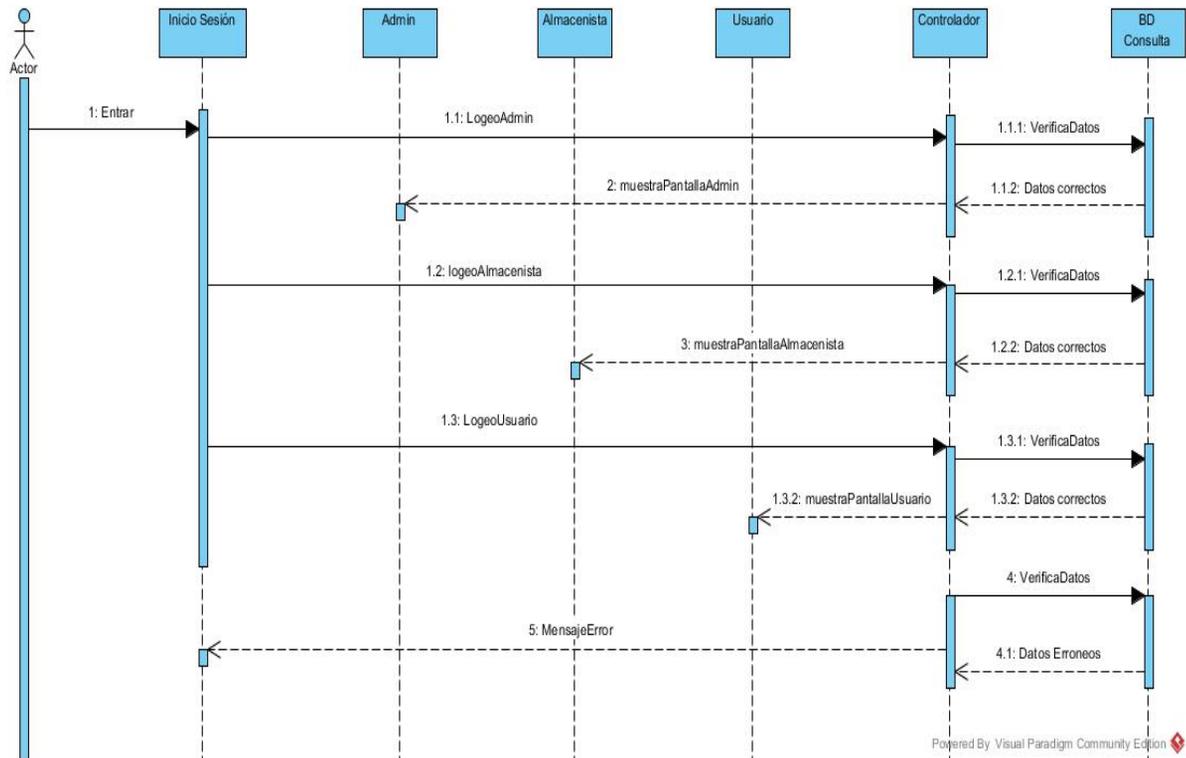


Fig. 5.10 – Diagrama de secuencia iniciar sesión

5.4.2. Caso de uso: Solicitar material.

A continuación se muestra el diagrama de secuencia para el caso de uso de Solicitar Material, el cuál se refleja la interacción del usuario con el sistema. En dicho diagrama se presenta el proceso mediante el cual se muestran las vistas correspondientes al usuario dependiendo de la acción que se haya realizado, ya sea la de solicitar material o mostrar un vale y su interacción con la base de datos. El actor que se modelará en este diagrama será el Usuario solicitante. En la Figura 5.11 se muestra el diagrama de secuencia antes mencionado.

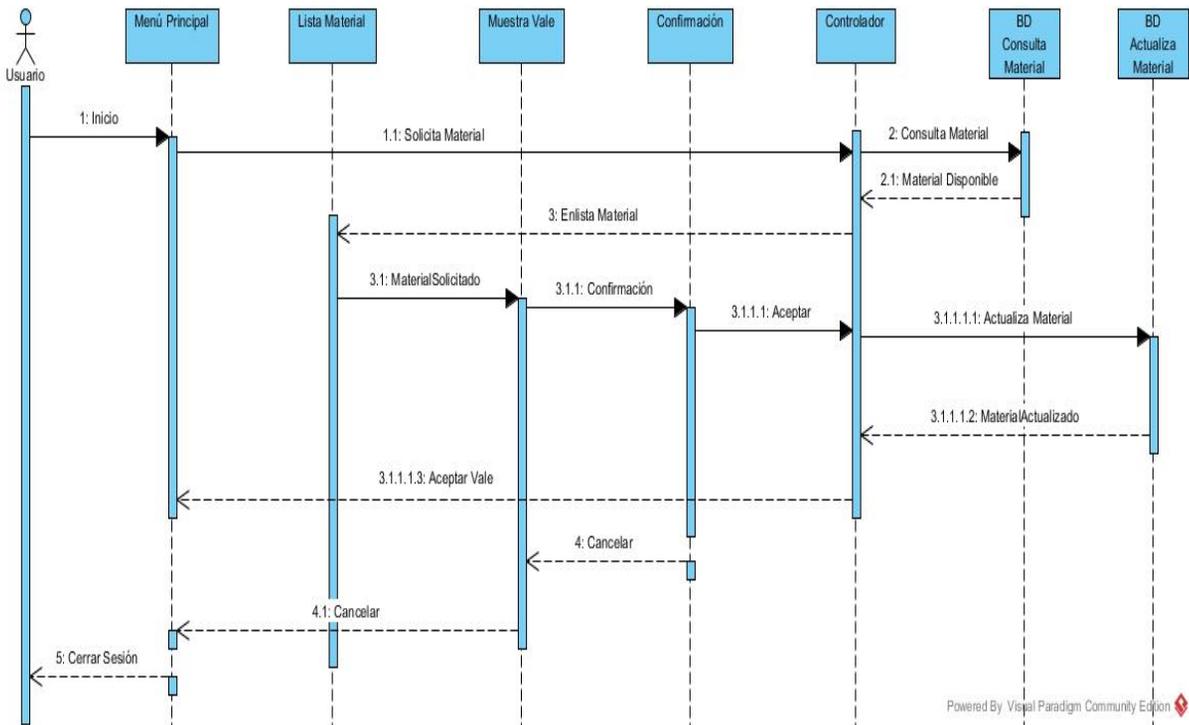


Fig. 5.11 – Diagrama de secuencia solicitar material.

5.4.3. Caso de uso: Validar solicitud de material

En la Figura 5.12 se muestra el diagrama de secuencia del caso de uso de Validar solicitud de Material. Dicho diagrama refleja el proceso del Administrador con el sistema para validar una solicitud pendiente, se muestra las vistas posibles dependiendo de la acción que se realice junto con la interacción con la base de datos. El actor que se modelará en este diagrama será el Administrador. A continuación se muestra el diagrama de secuencia antes mencionado.

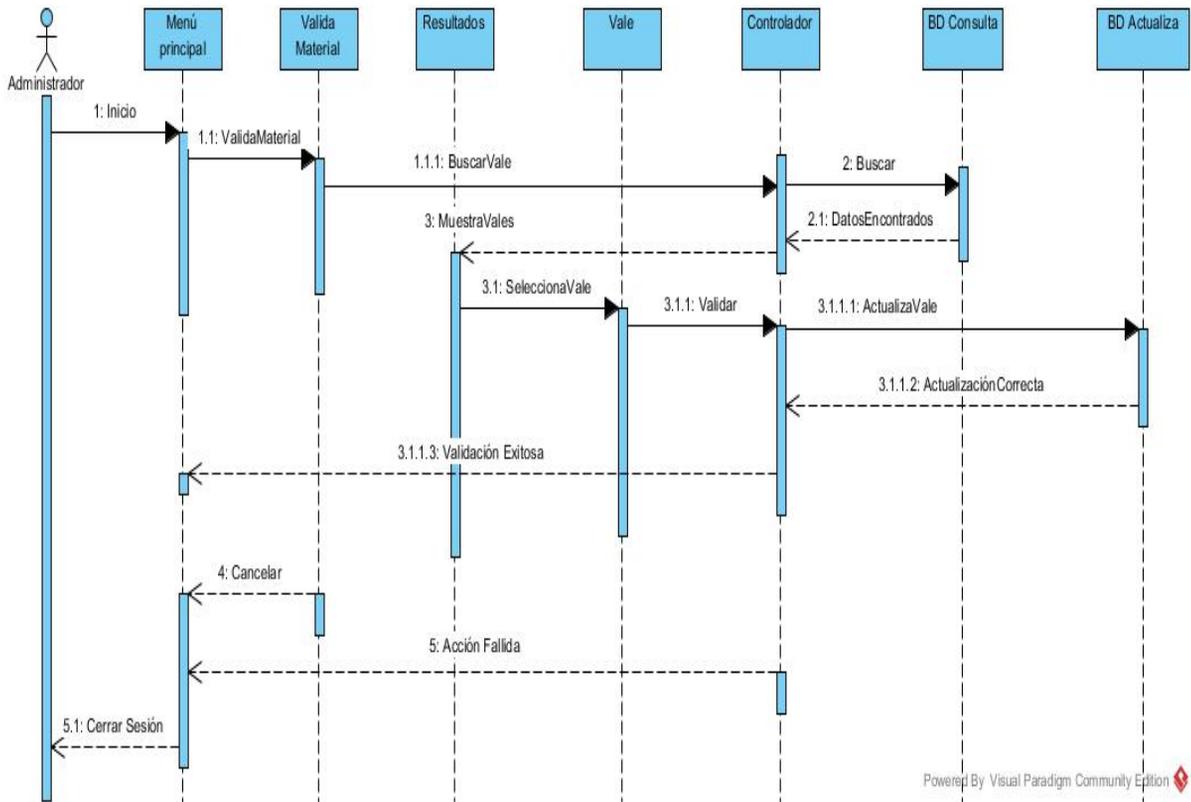


Fig. 5.12 Diagrama de secuencia validar o cancelar solicitud de material.

5.4.4. Caso de uso: Dar de alta material.

A continuación se muestra el diagrama de secuencia para el caso de uso de Alta de Material, el cuál se refleja la interacción del Administrador con el sistema para registrar un nuevo producto, así como las vistas posibles dependiendo de la acción que realice y la interacción con la base de datos para insertar registros de dicho material. El actor que se modelará en este diagrama será el Administrador. En la Figura 5.13 se muestra el diagrama de secuencia antes mencionado.

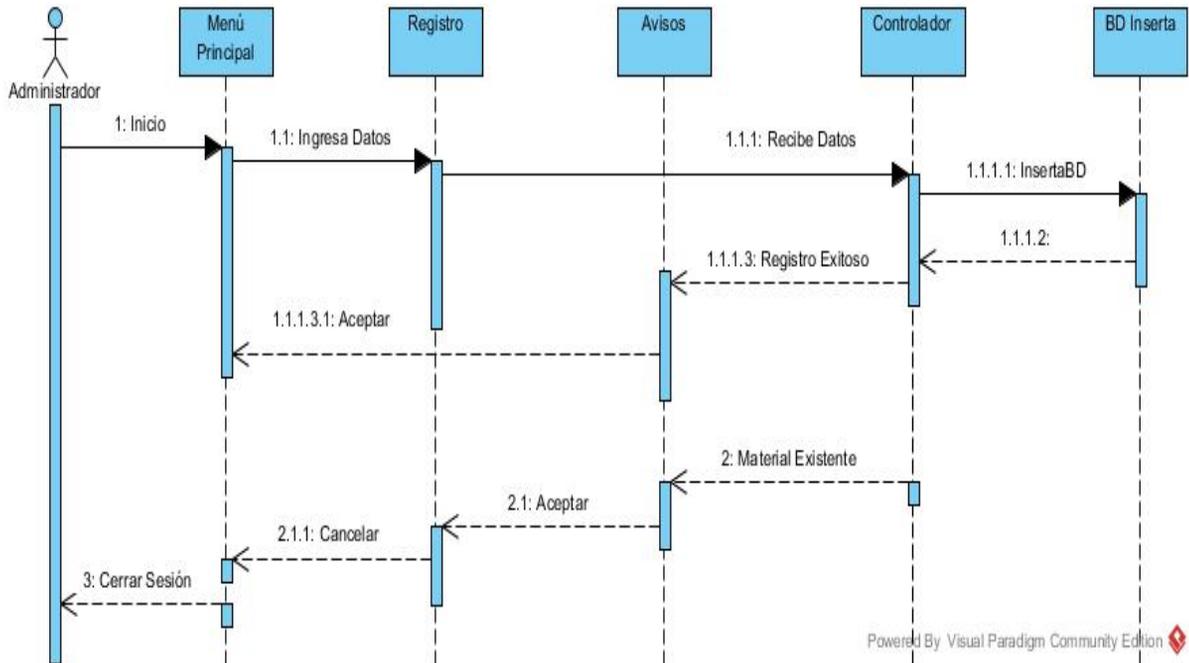


Fig. 5.13 – Diagrama de secuencia dar de alta material.

5.4.5. Caso de uso: Registrar material entregado.

Para el caso de uso de Registro de Material Entregado, se muestra a continuación su diagrama de secuencia, el cuál se refleja la interacción del usuario con el sistema, así como las vistas correspondientes. El actor que se modelará será el Almacenista. En la Figura 5.14 se muestra el diagrama de secuencia antes mencionado.

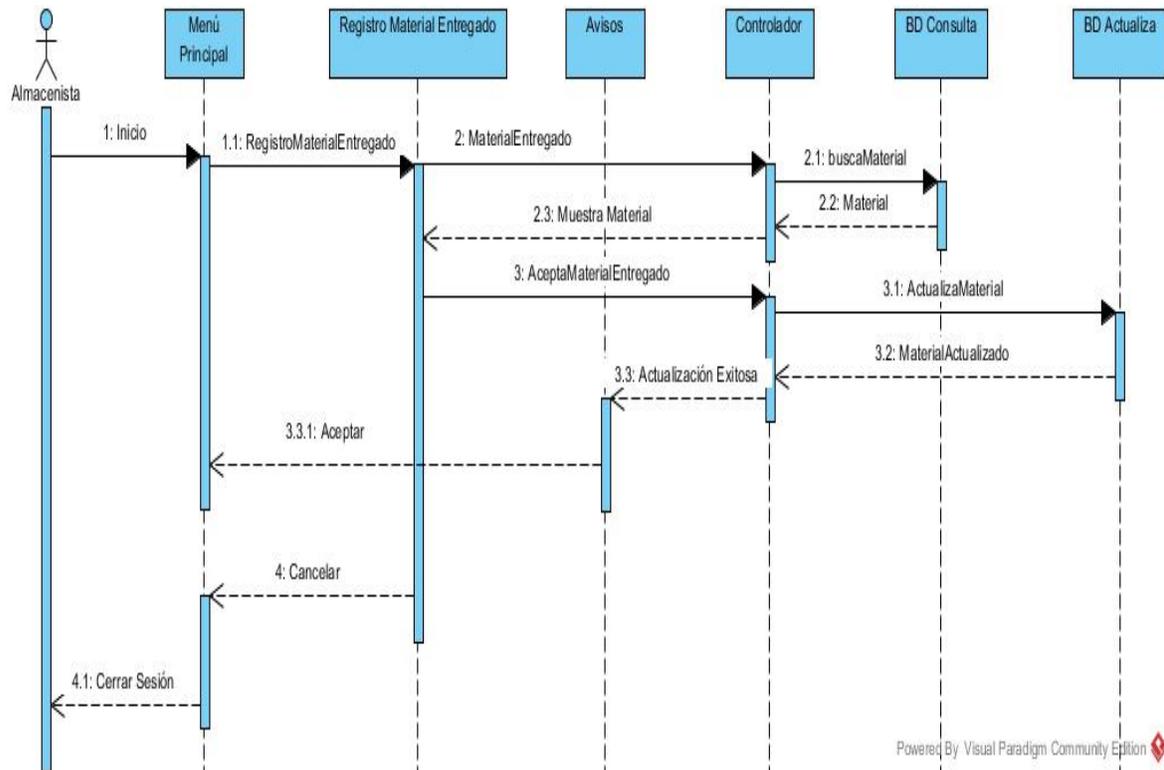


Fig. 5.14 – Diagrama de secuencia registrar material entregado.

5.5. Esquema de la Base de datos.

Para nuestra etapa de diseño, es necesario modelar la base de datos, la cual estará implementada en la capa modelo de nuestro MVC.

Como se mencionó en el capítulo 2, se usará una base de datos relacional, esto con el fin de facilitar la abstracción de la idea y organización de las entidades para poder consultar cualquier tipo de información de manera más clara y eficiente. También podremos manejar más fácil la información para evitar la redundancia de datos y así mantener la integridad de los datos de una mejor manera.

Como el módulo de administración de material tiene información que se repite, utilizaremos catálogos, los cuales nos permitirán almacenar información estática o que siempre será la misma, esto con el fin de evitar redundancias.

Después de analizar la información obtenida de los requerimientos del software, se procederá a modelar nuestra base de datos. Para esto, se utilizarán Diagramas Entidad-Relación para posteriormente implementarlo en nuestro gestor de base de datos, que en este caso es PostgreSQL.

Este diagrama nos ayudará a entender de una mejor manera los datos que se quieren utilizar para el software y como relacionarlos entre ellos. Para proceder a generar nuestro diagrama, necesitamos ubicar las entidades u objetos que comprenderán el sistema. En nuestro caso, las entidades que podemos declarar son las siguientes:

- **Solicitud:** Contendrá información referente a las solicitudes que realizan los usuarios, como por ejemplo, Observaciones de la solicitud, fecha de la solicitud, quien realiza la solicitud, entre otros. Una solicitud podrá tener muchos estados y solo podrá ser asignada a un usuario.
- **Material:** Información referente al material disponible para realizar las solicitudes, como por ejemplo, nombre del material, descripción, observaciones, existencia, etc. Un material podrá tener una sola marca pero podrá ser parte de muchas solicitudes.
- **Marca:** Contiene información de las marcas registradas en la base para asociarlas a algún material registrado. Contiene nombre de la marca. Una marca podrá pertenecer a unos tipos de material.
- **Estado_Solicitud:** Información acerca de los posibles estados que puede tener una solicitud. Contendrá el nombre del estado de la solicitud (Aprobada, Rechazada, En proceso o Cancelada por el usuario). Un estado podrá pertenecer a muchas solicitudes.

- **Sesión:** Información de los usuarios que solicitan material, como el nombre, usuario, contraseña, correo, entre otros. Un usuario podrá realizar muchas solicitudes.
- **Área:** Información del área a la que puede pertenecer un usuario o coordinación. Tendrá el nombre del área. Un Área puede contener muchas coordinaciones.
- **Coordinación:** Información de alguna coordinación registrada a la cual pertenece algún usuario, contendrá el nombre de la coordinación. Muchas coordinaciones pueden pertenecer a un Área.

Una vez que tenemos definidas nuestras entidades, necesitamos definir la manera en que la información se relacionará entre las distintas entidades, para eso utilizaremos Relaciones. Dichas relaciones permitirán definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

Por ejemplo, las solicitudes realizadas (de la entidad "Solicitud") tienen un material asignado (de la entidad "Material"). Es decir, un atributo de la entidad "*Material*" especificará a que solicitud pertenece.

Las relaciones se las representaremos en los diagramas como rombos, las cuales uniremos a las entidades.

A partir de las entidades que tenemos hasta este momento, obtendremos las siguientes relaciones para unir las entidades anteriormente mencionadas:

- **Solicitud_EstadoSolicitud:** Permitirá unir a la Entidad solicitud con la entidad EstadoSolicitud. Nos proporcionará información como la fecha y hora en que se asignó un estado a una solicitud.
- **Detalle_Solicitud:** Permitirá unir la entidad relación con la entidad Material. Contiene detalle de una solicitud realizada, como por ejemplo: Cantidad de material que se solicitó y la cantidad que se autorizó, entre otros.
- **Coordinacion_Sesión:** Permitirá unir la entidad Coordinación con la entidad Sesión. Nos proporcionará información de un usuario que pertenece a cierta coordinación.

Una vez obtenida toda la información anterior, procederemos a generar nuestro Diagrama Entidad-Relación, el cual se puede observar en la Figura 5.15:

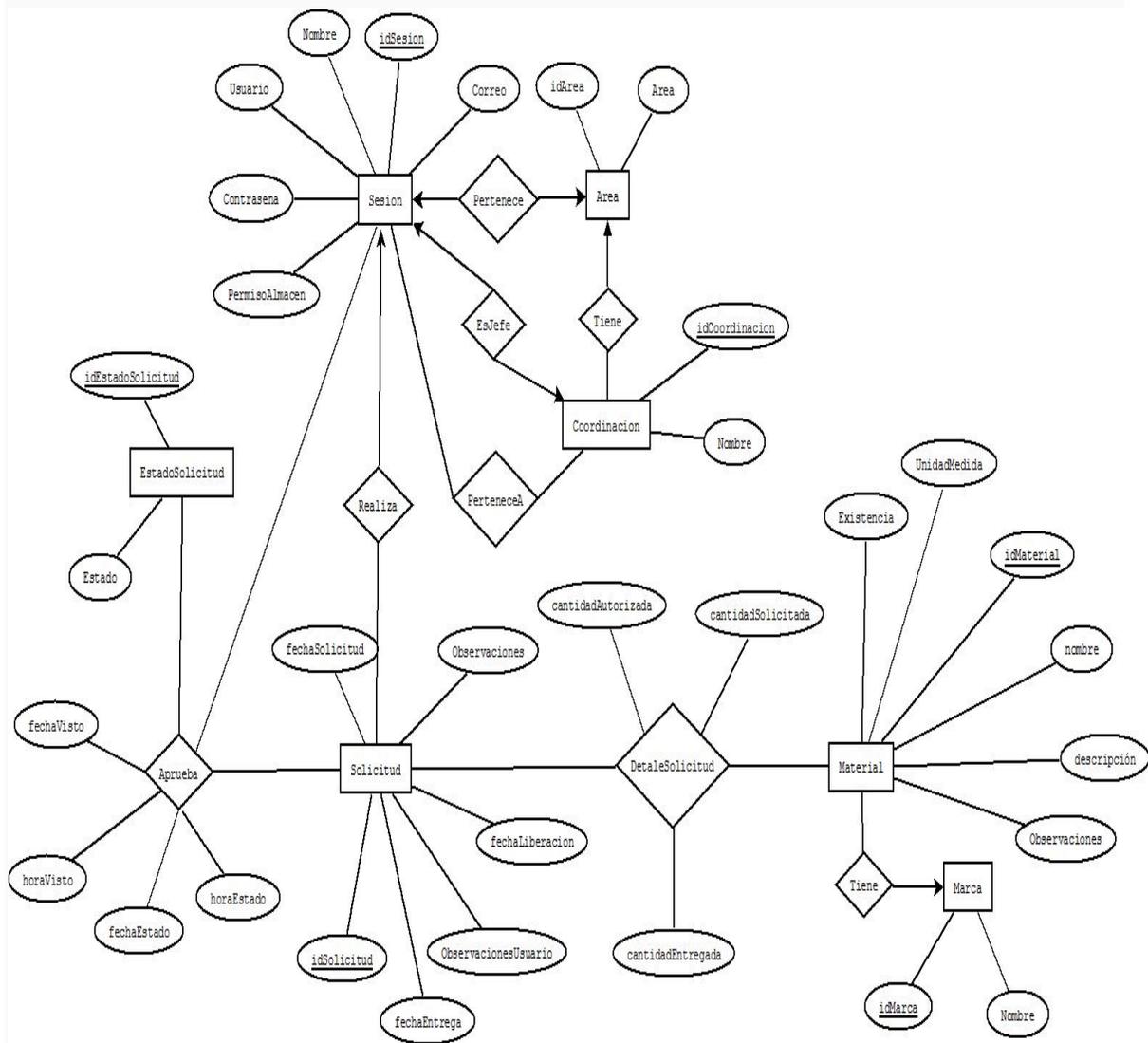


Fig. 5.15 – Modelo Entidad-Relación

5.6. Tablas de la base de datos

Ya que tenemos nuestro diagrama Entidad-Relación, procedemos a pasarlo a Modelo-Relacional. Con el modelo relacional podemos representar la lógica de nuestro modelo Entidad-Relación. La idea fundamental del Modelo Relacional se basa en el concepto de tablas, que se componen de registros y cada registro se compone de campos que son las columnas de la tabla. Una tabla es una estructura que sirve para almacenar datos tanto de tipo numérico como alfanumérico. Desde el punto de vista conceptual esto significa que cada entidad

se almacena en estructuras separadas. Así, cada entidad, tendrá una tabla pensada y diseñada para ese tipo de entidad. Cada elemento almacenado dentro de una tabla recibe el nombre de registro o fila. Una tabla se compone de campos o columnas, que son conjuntos de datos.

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia. Esto hace que sea más fácil de entender y de utilizar para cualquier usuario de la base de datos. En este modelo relacional, la información puede ser recuperada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje que utilizaremos para construir las consultas será SQL.

Basándonos en lo anterior, procederemos a generar las tablas con sus respectivos atributos:

Sesión: esta tabla contendrá los datos de los académicos que podrán hacer las solicitudes del material. Su estructura será la siguiente:

- **Sesión**

- **nIdSesion:** Llave primaria de nuestra tabla sesión. Tipo de dato: integer.
- **sUsuario:** Nombre del usuario con el cuál se podrá iniciar sesión en el sistema. Tipo de dato: varchar.
- **sContraseña:** Contraseña para poder ingresar al sistema. Tipo varchar.
- **pAlmacen:** Permisos que tendrá el usuario sobre el almacén, ya sea permiso de usuario normal, almacenista o administrador. Tipo de dato: varchar.
- **sNombre:** Nombre completo del usuario. Tipo de dato: varchar

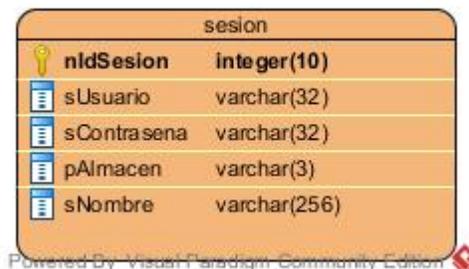


Fig. 5.16 – Tabla sesion

cArea_dep: Tabla que contendrá los datos de las áreas registradas. Para evitar redundancia de datos, esta tabla será un catálogo. Su estructura será la siguiente:

- **cArea_dep:**
 - **nIdArea:** Llave primaria de nuestra tabla cAreaDep. Tipo de dato: integer.
 - **sArea:** Nombre del área. Restricción unique para evitar redundancias. Tipo de dato: varchar.
 - **nIdSesion:** Llave foránea de la tabla sesión. Indicará que usuario de la tabla sesión es el jefe del área. Tipo de dato: integer.



Fig. 5.17 – Tabla carea_dep

ccoordinación: tabla que contendrá los datos de las coordinaciones registradas. Para evitar redundancia de datos, esta tabla será un catálogo. Su estructura será la siguiente:

- **ccoordinacion:**
 - **nIdCoordinacion:** llave primaria de la tabla coordinación. Tipo de dato: integer.
 - **sCoordinacion:** nombre de la coordinación. Restricción unique para evitar redundancias. Tipo de dato: varchar.
 - **nIdArea:** llave foránea de la tabla carea_dep. Indicará a que área pertenece la coordinación. Tipo de dato: integer.
 - **nIdSesion:** llave foránea de la tabla sesión indicará que usuario de la tabla sesión es el jefe de la coordinación. Tipo de dato: integer.



Fig. 5.18 – Tabla coordinación.

coordinaciónsesión: Tabla que contendrá la relación entre la tabla coordinación y la tabla sesión. Indicará a los usuarios que pertenecen a una cierta coordinación. Su estructura será la siguiente:

- **coordinacionSesion:**
 - **nidcoordinacionsesion:** llave primaria de la tabla coordinación:sesión. Tipo de dato: integer.
 - **nidcoordinacion:** llave foránea de la tabla coordinación. Indicará la coordinación a la que pertenece el usuario. Tipo de dato: integer.
 - **nidsesion:** llave foránea de la tabla sesión. Indicará que usuario pertenece a cierta coordinación. Tipo de dato: integer.



Fig. 5.19 – Tabla coordinacionsesion

cestadosolicitud: tabla que contendrá los datos de los estados que tendrán las solicitudes. Para evitar redundancia de datos, esta tabla será un catálogo. Los posibles estados que tendrá esta tabla son: en proceso, aprobada, rechazada y cancelada. Su estructura será la siguiente:

- **cestadosolicitud:**
 - **nidestadosolicitud:** llave primaria de la tabla nidestadosolicitud. Tipo de dato: integer.
 - **sestadosolicitud:** contendrá el nombre del estado que podrá tener la solicitud.



Fig. 5.20 – Tabla cestadosolicitud

solicitud_cestadosolicitud: tabla que contendrá la relación entre la tabla cestadosolicitud y la tabla solicitud. Esta tabla contendrá la información de las solicitudes para asignarles un estado y saber quién aprobó dicha solicitud. Su estructura es la siguiente:

- **solicitud_cestadosolicitud:**
 - **nidestadosolicitud_cestadosolicitud:** llave primaria de la tabla solicitud_cestadosolicitud. Tipo de dato: integer.
 - **nidestadosolicitud:** llave foránea de la tabla cestadosolicitud. Indicará en qué estado se encuentra la solicitud. Tipo de dato: integer.

- **nidsolicitud** llave foránea de la tabla solicitud. Indicará a que solicitud se le asignará un estado. Tipo de dato: integer.
- **nidsesion**: llave foránea de la tabla sesion. Indicará quien asignará estado a la solicitud. Tipo de dato: integer.
- **dfechavisto**: indicará la fecha en que una solicitud fue vista por el administrador. Tipo de dato: date
- **hhoravisto**: indicará la hora en que una solicitud fue vista por el administrador. Tipo de dato: time
- **dfechaestado**: indicará la fecha en que se le asignó un estado a la solicitud. Tipo de dato: date.
- **hhoraestado**: indicará la hora en que se le asignó un estado a la solicitud. Tipo de dato: time.

Solicitud_cEstadoSolicitud	
 nidSolicitud_cEstadoSolicitud	integer(10)
 <i>nIdEstadoSolicitud</i>	integer(10)
 <i>nIdSolicitud</i>	integer(10)
 <i>nIdSesion</i>	integer(10)
 dfechaVisto	date
 hhoraVisto	time(7)
 dFechaEstado	date
 hHoraEstado	time(7)

Fig. 5.21 – Tabla solicitud_cestadosolicitud

solicitud: tabla que contendrá la información de la solicitud. Su estructura es la siguiente:

- **solicitud**:
 - **nidsolicitud**: llave primaria de la tabla solicitud. Tipo de dato: integer.
 - **nidsesion**: llave foránea de la tabla sesion. Indicará quien está haciendo esa solicitud de material. Tipo de dato: integer.
 - **dsolicitud**: fecha en que fue hecha la solicitud. Tipo de dato: date
 - **dentrega**: fecha en que se entregó el material. Tipo de dato: date
 - **dliberacion**: fecha en que fue cerrada una solicitud. Tipo de dato: date
 - **observaciones**: observaciones de la solicitud por parte del administrador. Tipo de dato: varchar.
 - **observacionesusuario**: observaciones de la solicitud por parte del usuario solicitante. Tipo de dato: varchar

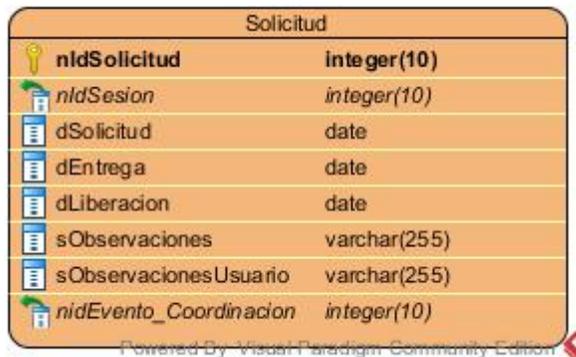


Fig. 5.22 – Tabla solicitud.

detallesolicitud: tabla que contendrá la relación entre la tabla solicitud y la tabla material. Obtiene los detalles de una solicitud realizada. Su estructura es la siguiente.

- **detallesolicitud:**
 - **niddetallesolicitud:** llave primaria de la tabla detallesolicitud. Tipo de dato. Integer.
 - **nsolicitada.** Cantidad de un material solicitado. Tipo de dato. Integer.
 - **nautorizada:** cantidad de material autorizado. Tipo de dato. Integer.
 - **nentregado:** cantidad de material entregado. Tipo de dato. Integer.
 - **nidsolicitud.** Llave foránea de la tabla solicitud. Indica a que solicitud pertenece este detalle. Tipo de dato: integer.
 - **nidmaterial.** Llave foránea de la tabla material. Indica de que material se está pidiendo en este detalle. Tipo de dato: integer.



Fig. 5.23 – Tabla detalle_solicitud.

cMarca: tabla que contendrá la información de las marcas de los productos del almacén. Para evitar redundancia de datos, esta tabla será un catálogo con la restricción unique. La estructura es la siguiente:

- **cMarca:**
 - **nIdmarca:** llave primaria de la tabla cmarca. Tipo de dato: integer.
 - **smarca:** nombre de la marca. Tipo de dato: varchar.
 - **bactivo.** Bandera que indicará si la marca fue eliminada. Tipo de dato: boolean



Fig. 5.24 – Tabla cmarca

Material: tabla que contendrá la información del material registrado en el almacén. Su estructura es la siguiente:

- **Material:**
 - **Nidmaterial:** llave primaria de la tabla material. Tipo de dato: integer.
 - **Snombre:** nombre del material. Tipo de dato: varchar.
 - **Sdescripcion.** Descripción del material. Tipo de dato: varchar.
 - **Sunidadmedida:** tipo de medida del material. Tipo de dato: varchar.
 - **Sobservaciones:** observaciones del material. Tipo de dato: varchar.
 - **Nexistencia:** cantidad de material disponible en el almacén. Tipo de dato: integer.
 - **Bactivo:** bandera que indica si un material fue eliminado. Tipo de dato: boolean.
 - **Nidmarca:** llave foránea de la tabla marca que indicará de que marca es el material. Tipo de dato: integer.



Fig. 5.25 – Tabla material.

Al unir todas las tablas con sus respectivos atributos y relacionándolas con las llaves foráneas, nos quedará el diagrama de la Figura 6.13:

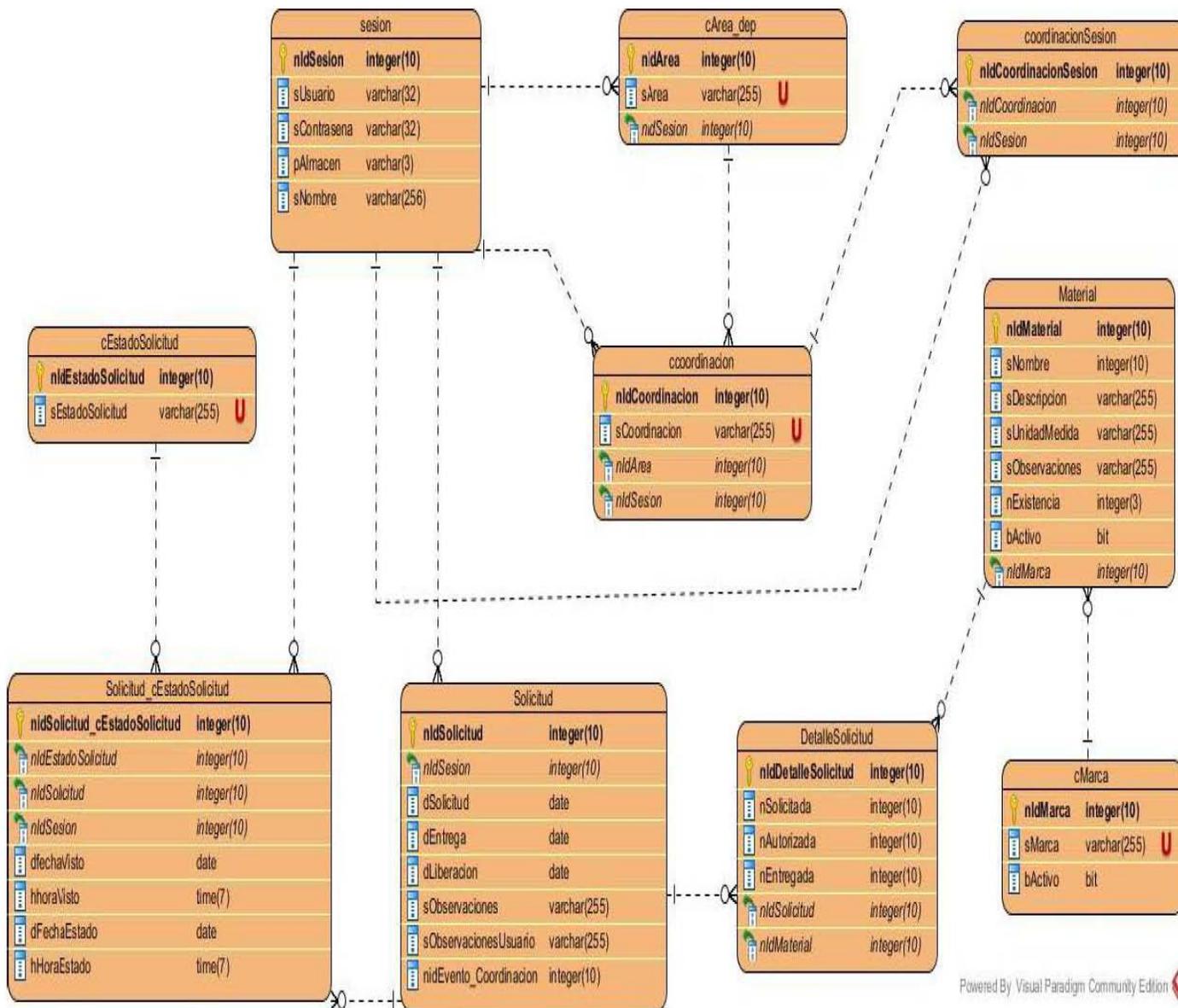


Fig. 5.26 – Diagrama de base de datos del almacén.

6. CONSTRUCCIÓN DEL SOFTWARE

En este capítulo se creará el código del software a partir del diseño del proyecto implementado en capítulos anteriores.

En este apartado se proporcionará:

- Una revisión del plan del proyecto para determinar actividades y tareas asignadas
- Se revisará el diseño del software para determinar la secuencia, construcción y los componentes del software.

6.1. Entendimiento del diseño del software.

A partir de reuniones de trabajo realizadas con el administrador del proyecto y el usuario final, se detalló la forma en que se diseñaría el sistema, la cual fue aprobada satisfactoriamente por el administrador del proyecto. En dichas juntas, se resolvieron dudas por ambas partes, tanto del usuario final como del equipo de desarrollo, se estableció una agenda reuniones con el usuario final para aclarar y/o especificar detalles de proyecto, así como para entregar avances y proponer metas para las próximas juntas.

6.2. Implementación de software a partir del diseño de software.

Para la implementación de dicho software, se utilizaron diferentes tipos de tecnologías, las cuales se detallan a continuación:

Java: Se utilizó como lenguaje principal para el manejo de los datos y variables que se muestran en las vistas para el usuario final, también se utilizó para la comunicación entre el sistema y la base de datos, ya sea para realizar consultas, actualizaciones y eliminaciones de manera dinámica. Para este caso, se utilizó la versión 1.8 java jdk.

PostgreSQL: Motor de base de datos en el cuál se implementaron las tablas y relaciones de nuestro modelo entidad relación, también se utilizó para implementar funciones que se utilizarán para insertar datos y actualizaciones de los mismos. Se utilizó la versión 9.5.

Java Server Pages (jsp): En la parte de la vista para la interacción entre el usuario final y el sistema, se utilizó dicha tecnología, la cual consiste en la combinación entre páginas web y el lenguaje de programación java para crear un contenido dinámico el cuál muestra valores de variables y funciones de java en

formato HTML. Dicho código java es ejecutado en el lado del servidor cuando la solicitud del usuario es recibida.

Netbeans IDE: Como entorno de desarrollo, se utilizó el IDE de Netbeans en su versión 8.1.

GlassFish: Como servidor de aplicación, se utilizó GlassFish en su versión 4.1.1.

Se decidió utilizar el patrón Modelo-Vista-Controlador (MVC) ya que nos proporciona mayor organización del código y facilidad para extensión, actualizaciones y mantenimiento del sistema.

Para mantener la integridad de los datos, las consultas y solicitudes de los datos se aplicaron mediante procedimientos almacenados, los cuales son llamados desde una clase definida dentro de la capa del modelo del sistema.

La parte de la lógica y funcionamiento del sistema, la cual proporciona la interacción entre la vista y el modelo, se implementó con java, esto con el fin de poder realizar consulta y actualizaciones de manera dinámica.

Se decidió utilizar estas tecnologías y versiones para poder realizar la integración al sistema SIDEPA sin ningún inconveniente, ya que dicho sistema está desarrollado bajo estas mismas.

Una vez analizado la base de datos de la Figura 7.1, se centró la atención en la tabla “Sesión”, la cual se muestra en la Figura 7.2.

sesion	
 nIdSesion	integer(10)
 sUsuario	varchar(32)
 sContrasena	varchar(32)
 pProfesores	varchar(3)
 sNombre	varchar(256)
 psepc	varchar(5)
 ppnpc	varchar(3)
 psevadem	varchar(3)
 pcesip	varchar(3)
 psaec	varchar(3)
 pJornadas	varchar(3)
 pSubcomites	varchar(3)
 pcvume	varchar(3)
 sCorreo	varchar(128)

Fig. 7.2 – Tabla sesión de base de datos SIDEp

En la Figura 7.2 podemos observar los diferentes atributos que contiene dicha tabla, pero en particular podemos observar los atributos psepc, ppnpc, psevadem, pcesip, psaec, pjornadas, psubcomites y pcvume. Estos atributos almacenan permisos que tienen los usuarios para poder acceder a los diferentes módulos del sistema. Para un mejor entendimiento, en la Figura 7.3 se muestra el sistema SIDEp en función:

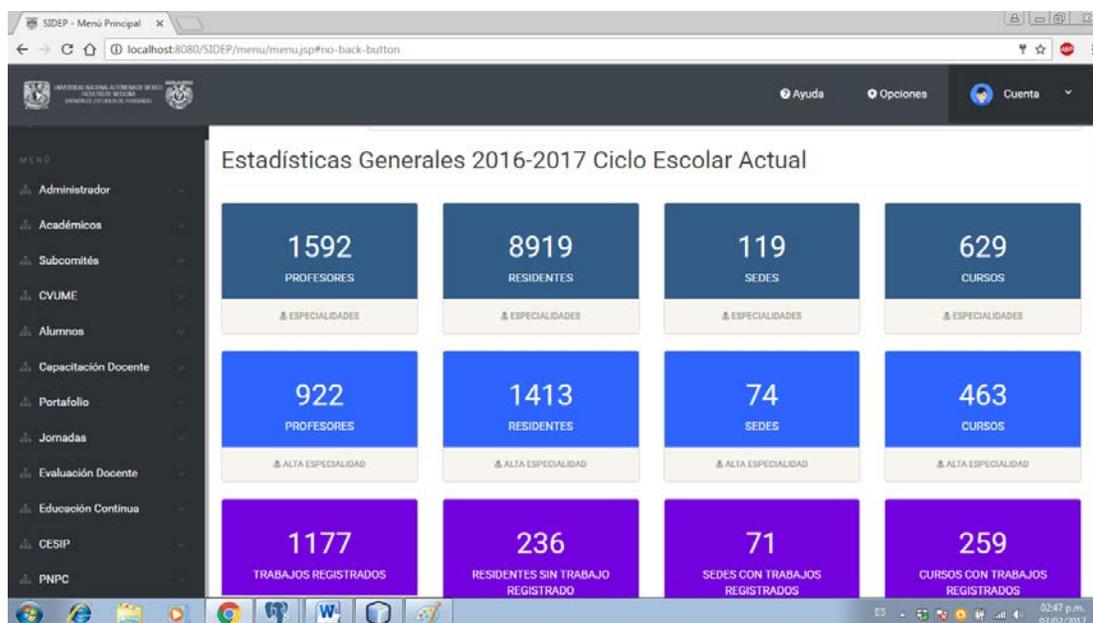


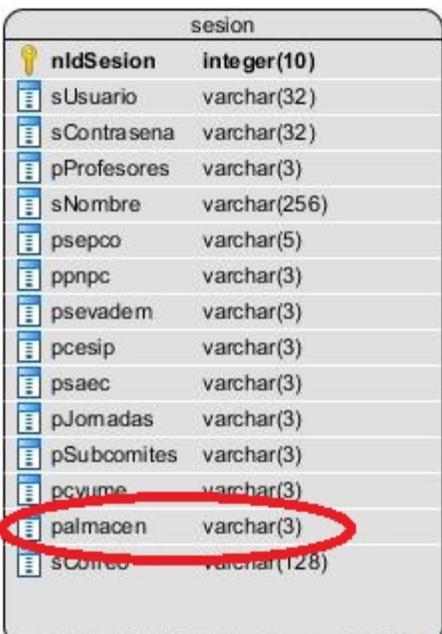
Fig. 7.3 – Sistema Sidep

En la Figura 7.3, del lado izquierdo podemos ver los diferentes módulos del sistema SIDEPE, los cuales se muestran dependiente de los permisos que tenga el usuario almacenados en la tabla “Sesion”.

Ahora bien, para poder integrar nuestro módulo de Administración de almacén, se realizó de la siguiente manera:

Analizando los tipos de permisos que tiene cada usuario en el sistema SIDEPE, se decidió agregar un permiso extra a la tabla “Sesion” el cuál lo llamamos “palmacen”. En dicho atributo se almacenarán los permisos que tiene cada usuario dentro del módulo de Almacén, es decir, dichos permisos harán visible el módulo para ciertos usuario que contenga permisos de acceso.

En la Figura 7.4 se muestra la tabla Sesión con el nuevo atributo agregado.



sesion	
 nIdSesion	integer(10)
 sUsuario	varchar(32)
 sContrasena	varchar(32)
 pProfesores	varchar(3)
 sNombre	varchar(256)
 psepco	varchar(5)
 ppnpc	varchar(3)
 psevadem	varchar(3)
 pcesip	varchar(3)
 psaec	varchar(3)
 pJomadas	varchar(3)
 pSubcomites	varchar(3)
 pcvume	varchar(3)
 palmacen	varchar(3)
 sCorteo	varchar(128)

Powered By Visual Paradigm Community Edition

Fig. 7.4 – Tabla Sesión Actualizada.

Una vez agregado el atributo para permisos de almacén, se unió el módulo de almacén a la base de datos del Sistema SIDEPE, lo cuál se hizo de la siguiente manera:

Se analizó en la base de datos del almacén, la cuál se muestra en la Figura 7.5:

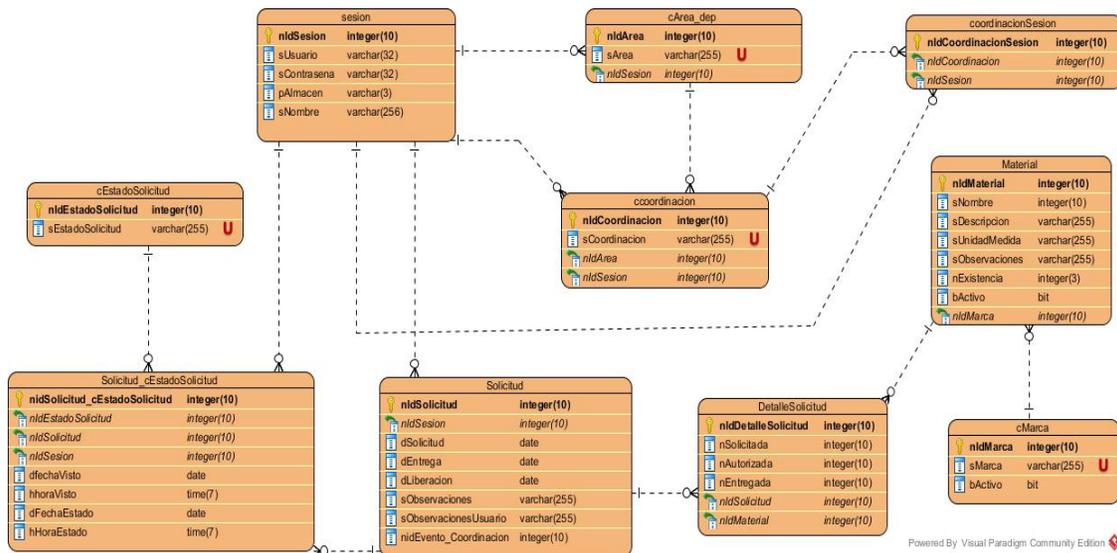


Fig. 7.5 – Diagrama de base de datos del almacén.

Se observaron que tablas dependían de la tabla Sesión, es decir, las tablas que tenían la llave foránea de la tabla sesión, en este caso las tablas afectadas serían las siguientes:

- **Carea_dep**: Se le asignó la llave foránea de la tabla sesión, esto con el fin de saber que usuario es jefe de un área.
- **Ccoordinacion**: Se le asignó la llave foránea de la tabla sesión, esto con el fin de saber que usuario es jefe de una Coordinación..
- **CoordinacionSesion**: Se le asignó la llave foránea de la tabla sesión, esto con el fin de saber de un usuario a que coordinación pertenece.
- **Solicitudo**: Se le asignó la llave foránea de la tabla sesión, esto con el fin de saber que usuario realizó alguna solicitud.
- **Solicitudo_cestadosolicitudo**: Se le asignó la llave foránea de la tabla sesión, esto con el fin de saber que jefe es el que debe de aprobar la solicitud.

Una vez asignadas todas las llaves foráneas de la tabla “Sesión” de la base de Sidep a las tablas de la base de Almacén, se obtuvo el siguiente diagrama final de la base de datos en la cuál se señala la base de datos de almacén ya integrada a la base de datos de SIDEp. A continuación se muestra la base final la Figura 7.6.

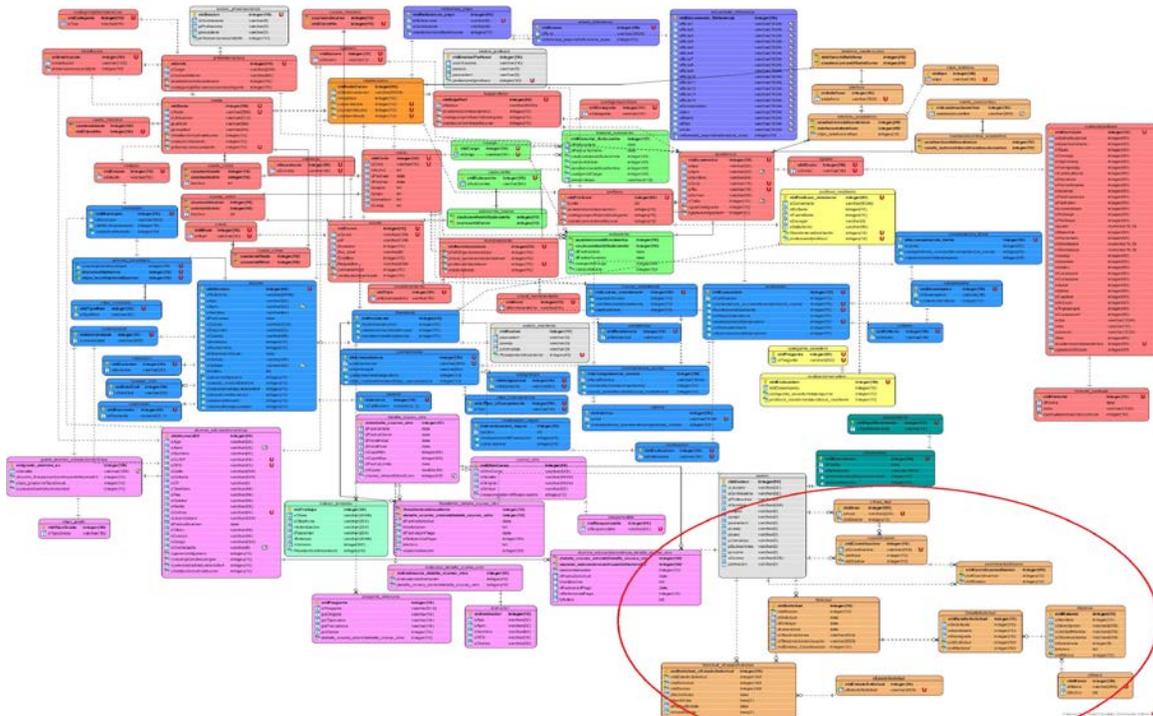


Fig. 7.6 – Base final con base de almacén integrada.

Una vez integrada la base de datos completa, buscó el archivo en el código fuente el cuál hace visibles los módulos en el sistema. El archivo se localiza en la ruta “SIDEP\src\java\control\Menu.java”.

En el archivo de Menu.java, se halla un método en especial llamado “getMenu()”. Dicho método recibe como parámetros los permisos que tiene un usuario en el sistema y a partir de esto armar y muestra el menú de acuerdo a los permisos que fueron recibidos en el método getMenu(). Para hacer visible nuestro módulo de almacén, en dicho método fue necesario agregar un atributo extra, llamado “palmacen”, el cuál nos indicaría los permisos que tiene el usuario dentro del módulo del almacén y mostrar las vistas adecuadas para el usuario. Una vez modificado el método getMenu() y agregado nuestro código de las vistas del módulo de almacén, se logró mostrar el módulo en la vista. El resultado se puede observar en la Figura 7.7.

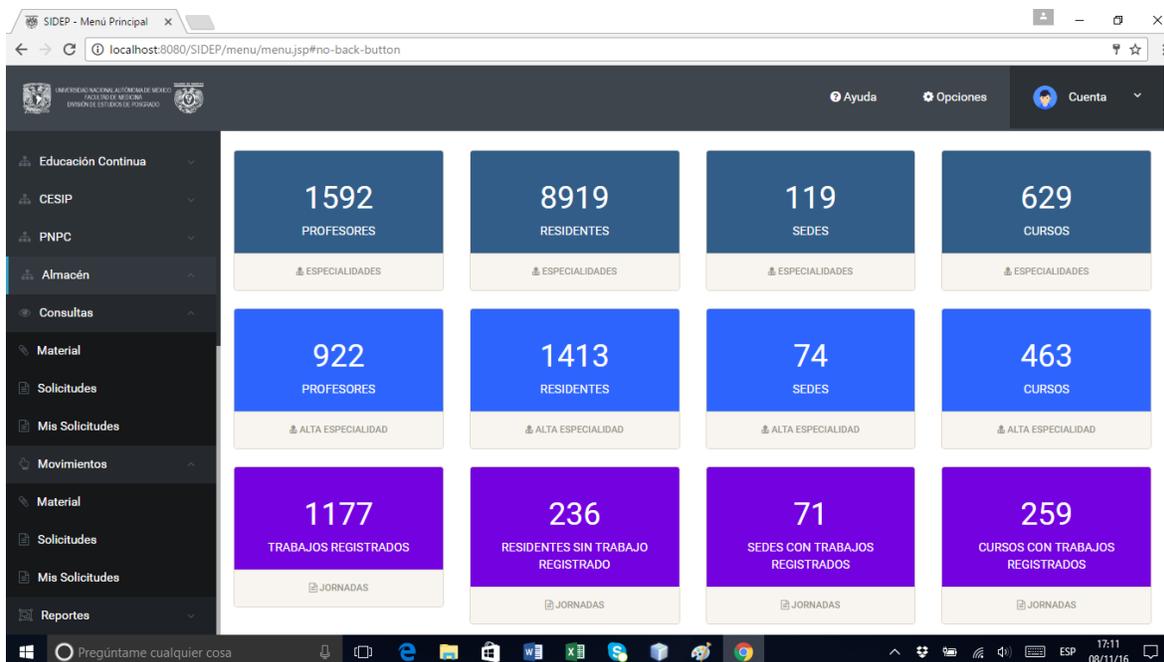


Fig. 7.7 – Módulo almacén agregado a la vista del sistema.

Una vez realizado lo anterior, se agregaron las vistas al proyecto. En primer lugar se creó una carpeta al sistema Sidep nombrada “Almacén”, la cual contiene todos los archivos para la manipulación de consultas y vistas. Esta estructura funcionará como la capa de Vista de nuestro modelo MVC. La estructura de esta carpeta es la siguiente:

- Almacén
 - Consultas: Muestra todas las consultas que se pueden hacer a la base de datos.
 - Material: Contiene archivos Ajax los cuales muestran las vistas de acuerdo a las consultas realizadas de material y un archivo material.jsp en el cuál se muestra toda la plantilla del material la cual se va modificando de acuerdo a los archivos ajax.
 - Mis solicitudes: Contiene archivos Ajax los cuales muestran las vistas de acuerdo a las consultas realizadas de para las solicitudes de un usuario, un archivo missolicitudes.jsp en el cuál se muestra toda la plantilla de las solicitudes de un usuario la cual se va modificando de acuerdo a los archivos Ajax y un archivo detalle solicitud el cual contiene los detalles de una solicitud seleccionada por el usuario.
 - Solicitudes: Contiene archivos Ajax los cuales muestran las vistas de acuerdo a las consultas realizadas de para las solicitudes de todos los usuarios Esta vista muestra todas las solicitudes de todos los usuarios, un archivo solicitudes.jsp en el cuál se muestra toda la plantilla de las solicitudes la cual se

va modificando de acuerdo a los archivos Ajax y un archivo detalle solicitud el cual contiene los detalles de una solicitud seleccionada

La estructura de la carpeta de las consultas de las solicitudes se puede observar en la Figura 7.8.

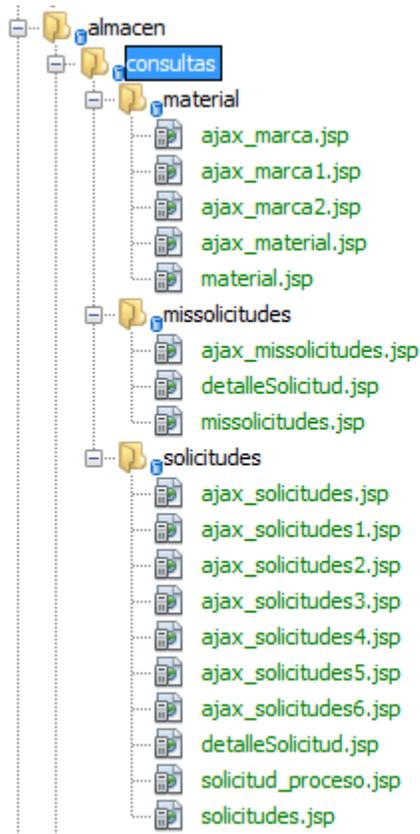


Fig. 7.8 – Estructura consultas almacén.

También contiene otra carpeta nombrada “Movimientos” dentro del directorio Almacén en la cual se guardan todas las vistas con las acciones que se pueden hacer dentro del sistema, por ejemplo Registrar material, alta de solicitudes, aprobar o rechazar, entre otros. La estructura de esta carpeta es la siguiente:

- Almacén
 - Movimientos: Muestra todos los movimientos que se pueden hacer a la base de datos.
 - Material: Contiene archivos Ajax los cuales muestran las vistas de acuerdo a las consultas realizadas de material y un archivo material.jsp en el cuál se muestra toda la plantilla del material la cual se va modificando de acuerdo a los archivos ajax. También contiene archivos nombrados como “proceso” en los

cuales se llevan a cabo todos los movimientos en la base de datos a través de las vistas, como por ejemplo alta, baja y actualización de material.

- Mis solicitudes: Contiene archivos Ajax los cuales muestran las vistas de acuerdo a las consultas realizadas de para las solicitudes de un usuario, un archivo missolicitudes.jsp en el cuál se muestra toda la plantilla de las solicitudes de un usuario la cual se va modificando de acuerdo a los archivos Ajax. También contiene archivos nombrados como “proceso” en los cuales se llevan a cabo todos los movimientos en la base de datos a través de las vistas, como por ejemplo alta y baja de una solicitud de algún usuario en específico.
- Solicitudes: Contiene archivos Ajax los cuales muestran las vistas de acuerdo a las consultas realizadas de para las solicitudes de todos los usuarios Esta vista muestra todas las solicitudes de todos los usuarios, un archivo solicitudes.jsp en el cuál se muestra toda la plantilla de las solicitudes la cual se va modificando de acuerdo a los archivos Ajax y un archivo detalle solicitud el cual contiene los detalles de una solicitud seleccionada. También contiene archivos nombrados como “proceso” en los cuales se llevan a cabo todos los movimientos en la base de datos a través de las vistas, como por ejemplo alta y baja de una solicitud de algún usuario en específico.

La estructura de la carpeta de los movimientos de las solicitudes se puede observar en la Figura 7.9.

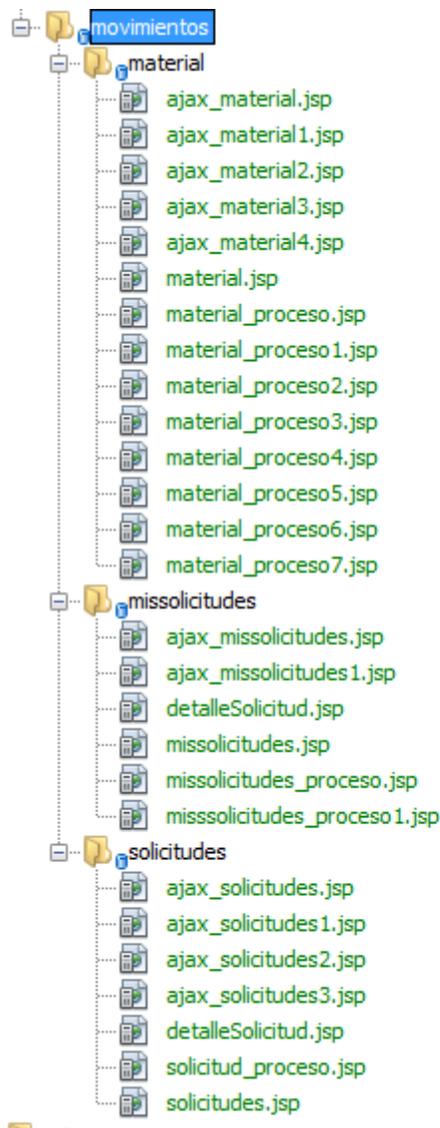


Fig. 7.9 – Estructura movimientos almacén.

Después se procedió a agregar los archivos java al proyecto, los cuales funcionan como la capa de Modelo y Controlador en el modelo MVC del proyecto. Los archivos que se agregaron fueron los siguientes:

- **Conexion.java**: Contiene todas las consultas y modificaciones directamente con la base de datos. Actúa como la capa de Modelo en nuestro proyecto MVC.
- **Almacen.java**: A través de este archivo se comunicará nuestras vistas con el modelo del proyecto. Actuará como el controlador de nuestro proyecto MVC.

Ambos archivos se agregaron al paquete “Control” del proyecto de Sidep. Con esto finalmente se tendría integrado el módulo de almacén al sistema de Sidep.

8. PRUEBAS.

Una vez realizada la integración del sistema en el apartado anterior, se procederá a realizar una serie de pruebas para asegurar el correcto funcionamiento, así como el de los datos y resultados esperados después de ejecutar alguna función dentro del software.

En este apartado se comprenderán los casos de prueba, se realizarán pruebas del usuario final introduciendo datos aleatorios en las diferentes funcionalidades del sistema y se documentarán los errores y observaciones del usuario final para futuros cambios en la versión actual del sistema.

8.1. Entendimiento de casos de uso para la realización de pruebas.

Basándonos en los casos de uso realizados y detallados en módulos anteriores y a través de juntas que se realizaron con el usuario final, se verificó que el sistema cumple con los requisitos especificados, mediante pruebas se llevaron a cabo junto con el usuario final con el fin de obtener una validación positiva del usuario y así poder continuar con el desarrollo.

En juntas posteriores que se realizaron entre el usuario final y el administrador del proyecto, se explicó de manera breve la forma en que se deberían llevar a cabo todas las pruebas para obtener resultados finales del usuario.

8.2. Realización de pruebas unitarias.

Se realizaron pruebas unitarias para cada componente del software por separado. Esto se realizó con el fin de asegurar y garantizar el correcto funcionamiento del sistema completo y de cada componente por separado con el fin de que un error en alguna parte del sistema no fuera afectar el resultado de los demás componentes. Con esto se facilitó la revisión de las tareas que se ejecutan en el sistema, tanto de manera general como individual.

Para la realización de las pruebas unitarias, se les asignó a distintos usuarios permisos para acceder al sistema con distintos privilegios y de esta manera cada usuario procedió a introducir datos aleatorios en los distintos módulos del sistema.

Se introdujeron datos de diferentes tipos, tanto datos nulos como datos aleatorios sin importar el tipo de datos solicitados en el sistema y con esto se pudo observar el comportamiento del sistema bajo distintos casos.

8.3. Corrección de errores y defectos encontrados.

Se aplicaron pruebas con datos aleatorios en todos los módulos del sistema. Algunas pruebas que se aplicaron fueron las siguientes:

- Registro de material por parte del administrador.
- Registro de una marca para los productos.
- Actualización y baja de material.
- Actualización de existencia de material.
- Solicitud de material por parte del usuario.
- Aprobar y rechazar solicitudes por parte del administrador.
- Rechazar una solicitud propia por parte del usuario.
- Distintos tipos de consultas para las solicitudes de material.

Después de aplicar dichas pruebas a cada componente del software, se detectaron y corrigieron errores y resultados erróneos o inesperados o no válidos. Con esto se logró corregir y evitar errores que pudieran afectar los resultados esperados en el funcionamiento del software.

Se fueron realizando pruebas unitarias en los distintos módulos durante su desarrollo, ya que con esto fue más sencillo detectar fallos ya que se analizaban módulos pequeños y esto permitía detectar y reparar errores con mayor facilidad.

Se probaron de manera combinada los módulos que componen *el sistema*, como por ejemplo, el módulo de registro de material se dividió en *Alta, baja y actualización*. Se probaron por separado y posteriormente se unieron para formar un solo módulo que sería utilizado por el administrador del sistema. De la misma manera, los módulos de solicitud de material, altas y bajas de solicitudes, filtros para las búsquedas, entre otros, se fueron acoplando progresivamente en conjuntos. Después de cada acoplamiento, se realizaba una prueba para verificar su correcto funcionamiento entre módulos. Cada vez que se comprobaba que el módulo funcionaba correctamente, se le suma un nuevo módulo y se vuelven a realizar pruebas.

El proceso de pruebas se fue realizando de la siguiente manera:

- **Index:** Fue el primer módulo que se construyó. Se le hicieron pruebas de inicio de sesión para posteriormente dejarlo listo para próximas integraciones.
- **Registro Material:** Este módulo se dividió en 3 sub-módulos los cuales son: Alta, baja y actualización de material. Se realizaron pruebas unitarias por

separado de estos módulos, para posteriormente integrarlos con el módulo de Menú.

- **Solicitudes:** Se construyó el módulo de solicitudes por separado. Se realizaron pruebas como realizar solicitudes y darlas de baja por parte del usuario. Una vez que todo funcionó correctamente por separado, se procedió a integrarlo con el resto de los módulos que ya estaban probados.
- **Acción sobre solicitudes:** Una vez que el módulo de solicitudes estaba terminado, se procedió a hacer el módulo para poder aprobar, rechazar o cancelar una solicitud por parte del administrador. Se realizaron pruebas para posteriormente anexar este módulo al de solicitudes que ya está integrado.

Para nuestra siguiente etapa, se diseñaron las consultas para los módulos que ya estaban integrados. Se procedió de la siguiente manera:

- **Consultas de Material:** Una vez que ya se tenía hecho el módulo de Material, se procedió a realizar un módulo para consultas del material registrado. El tipo de consultas para el material que se desarrollaron fueron las siguientes: Consulta por Marca, por nombre o por fecha. Una vez que se probó el correcto funcionamiento de este módulo de consultas, se procedió a integrarlo al módulo de Material.

Consultas de solicitudes: Para el módulo de solicitudes, se decidió trabajar por separado las consultas que se realizarían. Se procedió a desarrollar el módulo de consultas ordenadas de la siguiente manera: Solicitudes aprobadas, rechazadas, en proceso y Canceladas por el usuario. A su vez, cada consulta se le podía realizar una sub-consulta la cual permite consultar sobre año y mes. Una vez que pasó todas las pruebas, se procedió a integrarlo con el módulo de solicitudes.

8.4. Realización de pruebas mediante casos

Los casos de prueba los realizó el usuario final, probando las secciones de manera individual y realizando las acciones correspondientes, introduciendo al sistema datos de manera aleatoria para obtener distintos resultados. El usuario a partir de esto determinó si las pruebas eran satisfactorias o erróneas. Las pruebas que se realizaron fueron las siguientes:

8.4.1. Registro de material.

Se probó que el registro de material fuera exitoso con todos los valores introducidos en el sistema y que se relacionara de manera correcta con alguna marca específica y que el letrero de aviso de registro exitoso fuera el correcto y/o

adecuado. No hubo observaciones al respecto. **La prueba fue satisfactoria para el usuario.**

8.4.2. Registro de marca.

Se probó que el registro de marca fuera exitoso con todos los valores introducidos y que el letrero de aviso de registro de marca exitoso fuera el correcto y/o adecuado. No hubo observaciones al respecto. **La prueba fue satisfactoria para el usuario.**

8.4.3. Registro de una solicitud.

Se hicieron pruebas de registro de una solicitud. Se verificó que la existencia mostrada coincidiera con la existencia del material que solicitó el usuario. El registro fue exitoso con todos los valores introducidos. La observación que se hizo fue que no se mostrara la existencia total del material en almacén. **La prueba fue satisfactoria para el usuario.**

8.4.4. Acción a las solicitudes.

Se modificaron los valores de una solicitud registrada por un usuario y se le asignaron distintos estados a la solicitud (aprobada o rechazada). No se hicieron observaciones al respecto. **La prueba fue satisfactoria para el usuario.**

8.4.5. Acción a las solicitudes por parte del usuario.

El usuario canceló una solicitud registrada por él mismo. Automáticamente se añadió la observación de “cancelado por el usuario” en la solicitud. No se hicieron observaciones al respecto. **La prueba fue satisfactoria para el usuario.**

8.4.6. Consulta de solicitudes:

Se realizaron consultas a las solicitudes y se corroboró que fueran las solicitudes correspondientes a la consulta que se realizó. Se realizaron consultas por estado de solicitud, año y fecha. Se solicitó que se agregue un filtro extra para consultar por usuario. **La prueba fue satisfactoria para el usuario.**

8.4.7. Consulta de solicitudes por estado.

Se corroboró que al abrir una solicitud específica, los datos de la solicitud coincidieran con el tipo de solicitud que era. De la misma manera se verificó que las fechas mostradas en la solicitud fueran correctas y que el estado de la solicitud

indicará el estado correcto. Se hicieron observaciones de agregar la fecha en que fue vista la solicitud. **La prueba fue satisfactoria.**

8.4.8. Corrección de defectos encontrados.

Después de realizar las pruebas con el usuario, se corrigieron los defectos encontrados y peticiones del usuario para posteriormente obtener una validación. **La prueba fue satisfactoria del usuario final.**

9. ENTREGA DE PRODUCTOS.

En este apartado se comprenderá la configuración y especificaciones para el correcto funcionamiento del software, así como la documentación del manual de mantenimiento del sistema. También el usuario verificará el manual de mantenimiento que sea entendible y legible.

9.1. Especificaciones.

El sistema de administración es un sistema web el cual requiere para su correcto funcionamiento:

- Servidor Glassfish versión 4.1.1
- Navegador web con JavaScript.
- Manejador de bases de datos: PostgreSQL versión 9.5
- Entorno de desarrollo: Netbeans versión 8.1

9.2. Manual de mantenimiento.

El manual de mantenimiento está conformado por comentarios y especificaciones en el código fuente, script^{rs} para generar las tablas y la estructura de la base de datos, así como las restricciones, llaves primarias y foráneas, nombres de las tablas y procedimientos almacenados. También se incluyen algunos diagramas presentados en la documentación del proyecto. Los archivos que incluyen la documentación contienen la siguiente información:

- Generabasealmacen.sql
- Diagramas de base de datos.
- Código de proyecto.

9.3. Manual de usuario.

El manual de usuario contiene las operaciones que cada usuario puede realizar dentro del módulo dependiendo del tipo de permisos que tenga el usuario que acceda. Este manual se realizó amalgamando los módulos de la siguiente manera:

Menú principal

En el índice o ventana principal del sistema: a partir de esta ventana principal, se tiene acceso al módulo de almacén, el cual contiene todas las opciones y/o funcionalidades para el usuario final según los permisos que tenga. Los permisos pueden estar clasificados de la siguiente manera según el tipo de usuario que acceda al sistema (ver Figura 9.1):

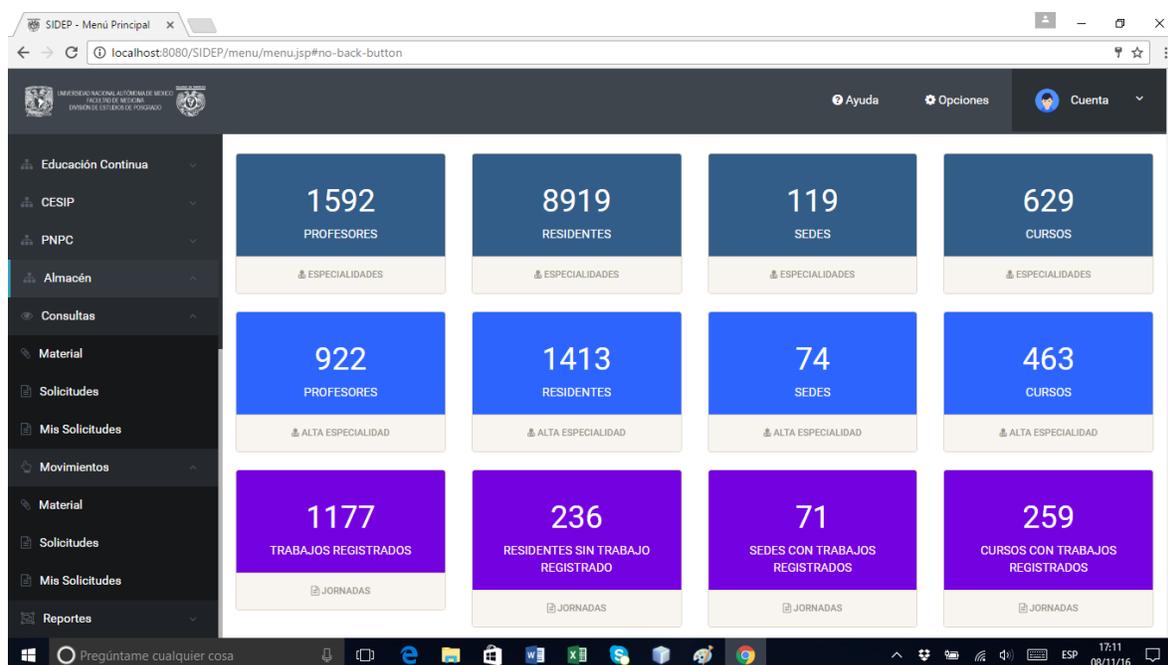


Fig. 9.1 — Menú principal

- Permisos de usuario solicitante de material: este tipo de usuario podrá consultar material, realizar solicitudes, consulta de historial de sus solicitudes por tipo (aprobadas, rechazadas o en proceso). De la misma manera podrá verificar que una solicitud ya fue revisada por el administrador.
- Permisos de administrador: podrá realizar registro, actualización y eliminación de material de la base de datos del almacén, consulta de las solicitudes realizadas por los usuarios, de la misma manera podrá consultar estas solicitudes por año, fecha y estado de la solicitud (aprobadas, rechazadas y en proceso), modificar cantidades de material solicitadas y también tendrá las funciones de usuario solicitante de material.
- Permisos de almacenista: solo tendrá permisos para modificar cantidades solicitadas y funciones de usuario solicitante de material.

Menú de consultas de material:

En este menú el usuario con cualquier tipo de permiso podrá realizar consultas de material registrado por el administrador del sistema.

En la siguiente Figura podemos observar el menú de consultas de material, el cual se puede consultar por el material de forma general y/o por marca.

Consulta de todo el material:

Almacén/ Consultas/ Material

Por Todos Marca - Material

MATERIAL

Mostrando 10 registros Búsqueda:

Nombre	Marca	Descripción	Existencia	Unidad Medida
PRODUCTO 1	MARCA 2	DESCRIPCIÓN 1	18	MEDIDA 1
PRODUCTO 2	MARCA 1	DESCRIPCIÓN 2	14	MEDIDA 2
PRODUCTO 3	MARCA 1	DESCRIPCIÓN 3	10	MEDIDA 3
PRODUCTO 4	MARCA 4	DESCRIPCIÓN 4	10	MEDIDA 4
PRODUCTO 5	MARCA 3	DESCRIPCIÓN 5	3	UNIDAD 5

Mostrando 1 a 5 de 5 registros

ANTERIOR 1 SIGUIENTE

Fig. 9.2 — Consulta todo el material

Las consultas de material también se podrán realizar por un material en específico o todos los materiales que pertenecen a una marca en particular.

Consulta por marca de material:

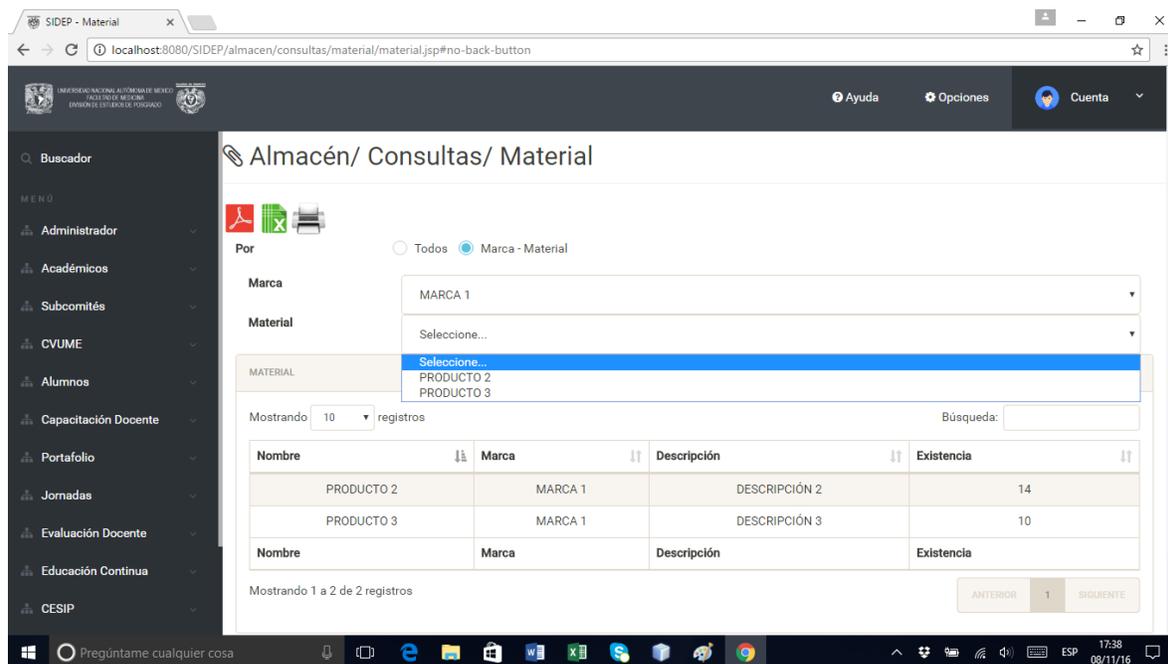


Fig. 9.3 — Consulta por marca

Las consultas de solicitudes para el usuario con permisos de administrador, se podrán realizar con los siguientes filtros:

- Todas las solicitudes
- Solicitudes aprobadas
- Solicitudes rechazadas
- Solicitudes en proceso revisadas (dichas solicitudes ya fueron revisadas por el administrador pero aún no han sido atendidas)
- Solicitudes en proceso no revisadas (dichas solicitudes no han sido revisadas por el administrador)
- Canceladas por el usuario.

En todas las consultas se podrán checar las solicitudes por año y por mes.

En la Figura 9.4, se muestra una consulta de todas las solicitudes realizadas en enero del 2017:

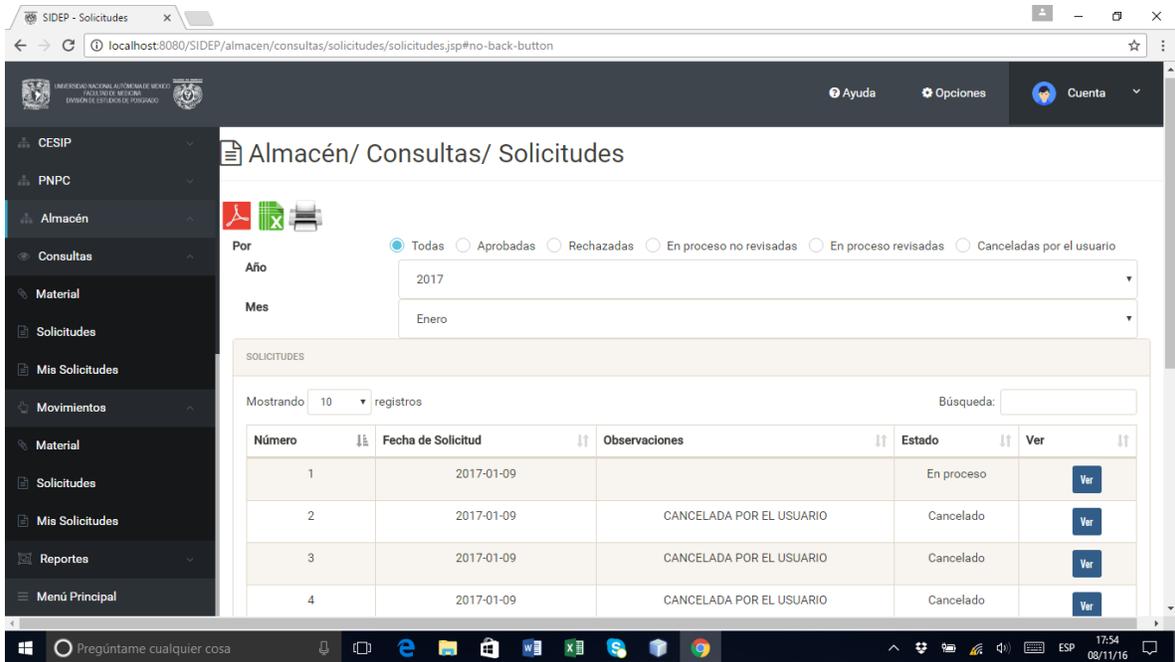


Fig. 9.4 — Consulta todas las solicitudes

Consulta de solicitudes de usuario:

El usuario solicitante podrá realizar consultas de su histórico de solicitudes. Podrá hacer consultas por los siguientes filtros: todas, aprobadas, en proceso y rechazadas (ver Figura 9.5).



Fig. 9.5 — Consulta solicitudes usuario

El usuario podrá ver el detalle de cada solicitud y podrá verificar si su solicitud ya fue atendida por el administrador. El formato y color de la solicitud dependerá del estado de la solicitud. A continuación se muestran los distintos estados en que puede estar una solicitud.

Solicitud aprobada:

En este estado, se muestra en color verde con la leyenda de “Estado: Aprobada”, como se muestra en la Figura 9.6.

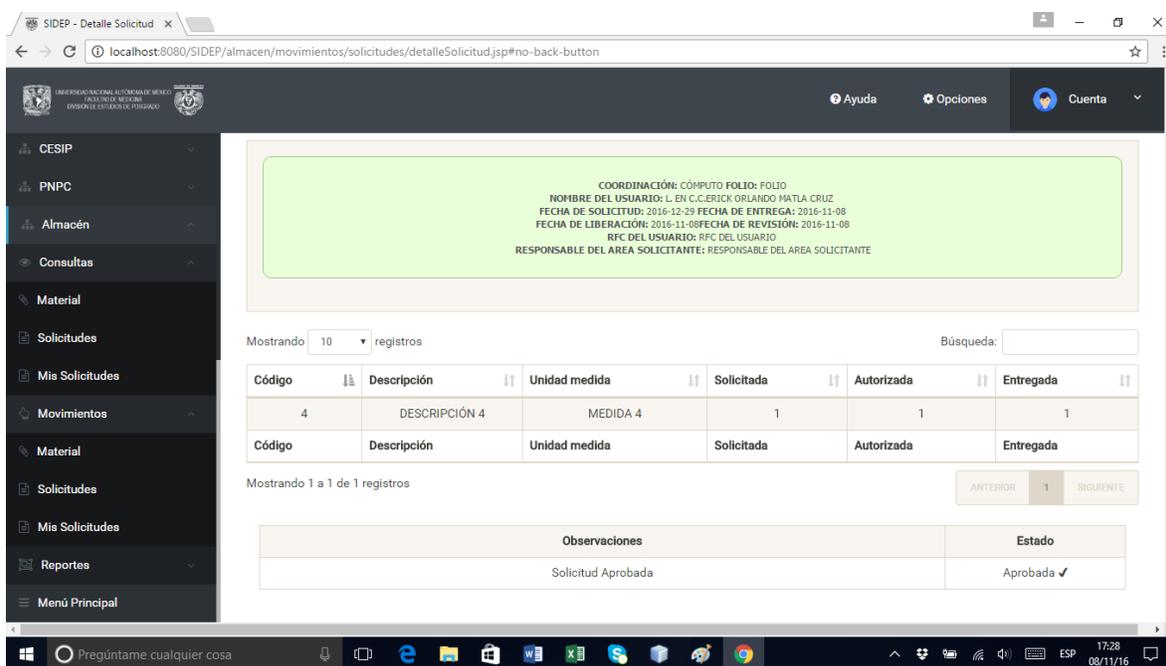


Fig. 9.6 — Solicitud aprobada

En este estado se muestra la fecha en que se revisó la solicitud, fecha de liberación y fecha de entrega. Así como las observaciones del administrador.

Solicitud en proceso:

En este estado, se muestra en color amarillo con la leyenda de “Estado: En proceso”, para este caso, el administrador tendrá la opción de aprobar o rechazar la solicitud para que el usuario vea la respuesta de su solicitud, la solicitud se muestra en la Figura 9.7.

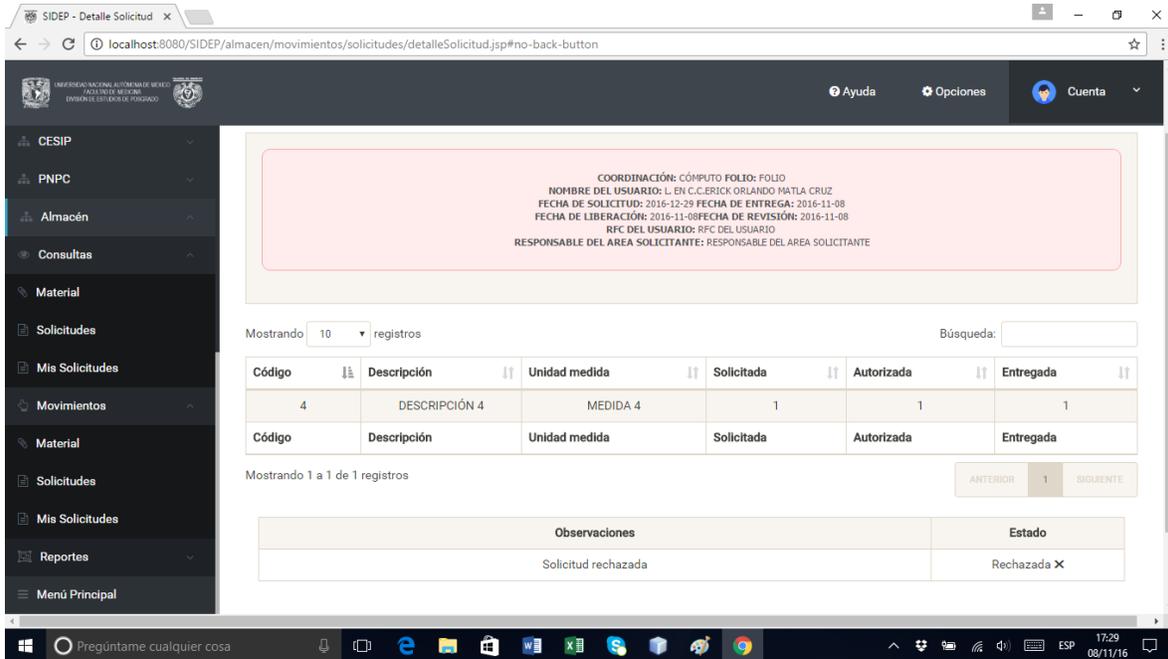
The screenshot displays the SIDEP web application interface. The browser address bar shows the URL: localhost:8080/SIDEP/almacen/movimientos/solicitudes/detalleSolicitud.jsp#no-back-button. The application header includes the SIDEP logo, navigation links for 'Ayuda', 'Opciones', and 'Cuenta', and a user profile icon. The sidebar menu on the left lists various sections: CESIP, PNPC, Almacén, Consultas, Material, Solicitudes, Mis Solicitudes, Movimientos, Material, Solicitudes, Mis Solicitudes, Reportes, and Menú Principal. The main content area features a yellow box with the following information: COORDINACIÓN: CÓMPUTO FOLIO: FOLIO; NOMBRE DEL USUARIO: L EN C.G. ERICK ORLANDO MATLA CRUZ; FECHA DE SOLICITUD: 2017-03-09; FECHA DE ENTREGA: SIN FECHA DE ENTREGA; FECHA DE LIBERACIÓN: SIN FECHA DE LIBERACIÓN; FECHA DE REVISIÓN: 2016-11-08; RFC DEL USUARIO: RFC DEL USUARIO; RESPONSABLE DEL AREA SOLICITANTE: RESPONSABLE DEL AREA SOLICITANTE. Below this, there is a table with columns: Código, Descripción, Unidad medida, Solicitada, Autorizada, and Entregada. The table shows one record with the following values: Código: 5, Descripción: DESCRIPCIÓN 3, Unidad medida: MEDIDA 3, Solicitada: 1, Autorizada: 0, Entregada: 0. The table is followed by a dropdown menu for 'Acción:' with options: Seleccione..., Aprobar, and Rechazar. The bottom of the screenshot shows the Windows taskbar with the system clock at 17:31 on 08/11/16.

Fig. 9.7 — Solicitud en proceso

En este estado se muestra la fecha en que el administrador revisó la solicitud se revisó la solicitud.

Solicitud rechazada:

Para este estado, se mostrará en color rojo la solicitud indicando que la solicitud fue rechazada, junto con la leyenda de “Estado: Rechazado”, como se muestra en la imagen 9.8.



The screenshot shows the SIDEP web application interface. The top navigation bar includes 'Ayuda', 'Opciones', and 'Cuenta'. The left sidebar contains a menu with items like 'CESIP', 'PNPC', 'Almacén', 'Consultas', 'Material', 'Solicitudes', 'Mis Solicitudes', 'Movimientos', 'Material', 'Solicitudes', 'Mis Solicitudes', 'Reportes', and 'Menú Principal'. The main content area displays a pink box with the following information:

COORDINACIÓN: CÓMPUTO FOLIO: FOLIO
NOMBRE DEL USUARIO: L EN C.C.ERICK ORLANDO MATLA CRUZ
FECHA DE SOLICITUD: 2016-12-29 FECHA DE ENTREGA: 2016-11-08
FECHA DE LIBERACIÓN: 2016-11-08 FECHA DE REVISIÓN: 2016-11-08
RFC DEL USUARIO: RFC DEL USUARIO
RESPONSABLE DEL AREA SOLICITANTE: RESPONSABLE DEL AREA SOLICITANTE

Below this, there is a search bar and a table showing request details. The table has columns: Código, Descripción, Unidad medida, Solicitada, Autorizada, and Entregada. The first row shows: 4, DESCRIPCIÓN 4, MEDIDA 4, 1, 1, 1. Below the table, it says 'Mostrando 1 a 1 de 1 registros' and 'ANTERIOR 1 SIGUIENTE'.

At the bottom, there is a table with two columns: Observaciones and Estado. The first row shows: Solicitud rechazada and Rechazada X.

Fig. 9.8 — Solicitud rechazada

En este estado se muestra la fecha en que el administrador revisó la solicitud se revisó la solicitud junto con las observaciones del administrador de porqué fue cancelada la solicitud.

Menú de movimientos:

Se podrán realizar movimientos de material y solicitudes de acuerdo a los permisos que tenga el usuario que acceda al sistema, ya sea administrador, almacenista o usuario solicitante.

En movimientos de material, solo podrán acceder usuarios con permisos de administrador y podrán realizar alta, baja y cambios de material del almacén, tanto del material como la cantidad en existencia de cualquier material registrado. Para el registro se podrá guardar material y las marcas de los materiales.

En la en la Figura 9.9 se muestra el menú para realizar movimientos de material.

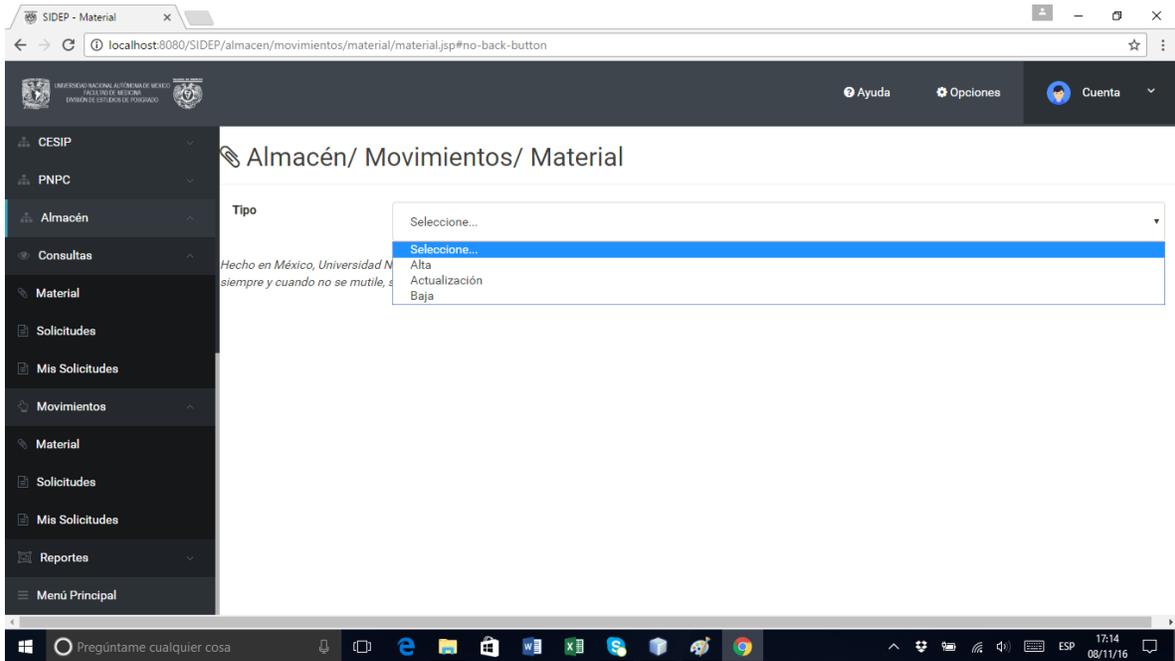


Fig. 9.9 — Movimientos material

En la en la Figura 9.10, se muestra la ventana para el administrador para realizar movimientos de alta de material:

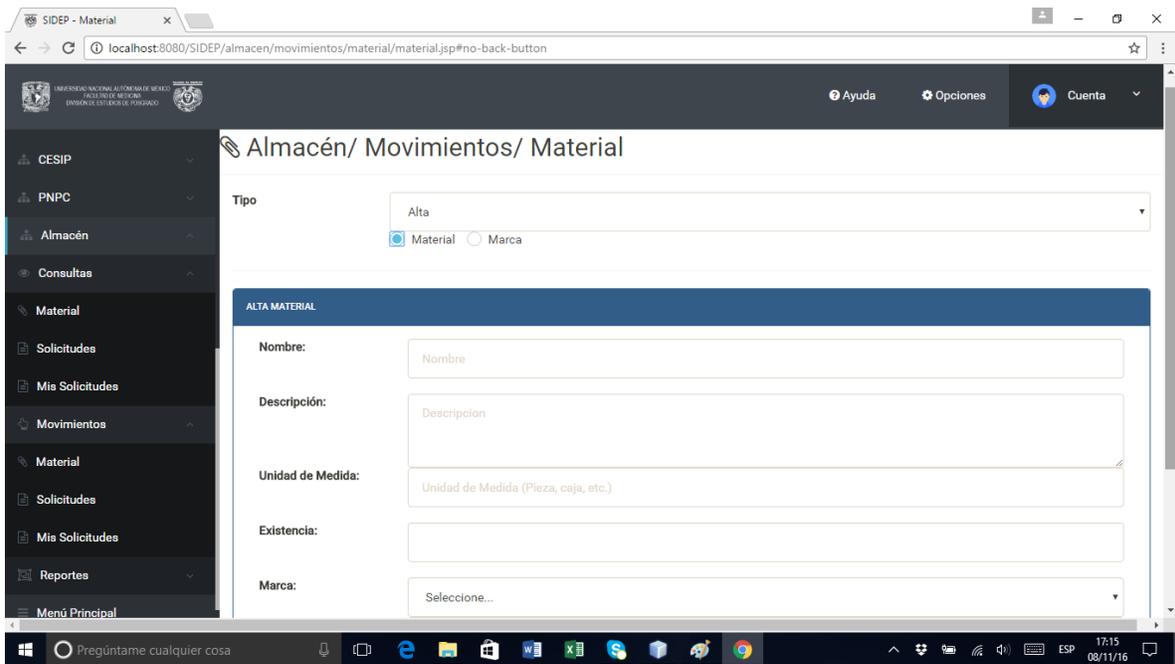


Fig. 9.10 — Alta material

En la en la Figura 9.11, se muestra la ventana para el administrador para realizar movimientos de alta de marca:

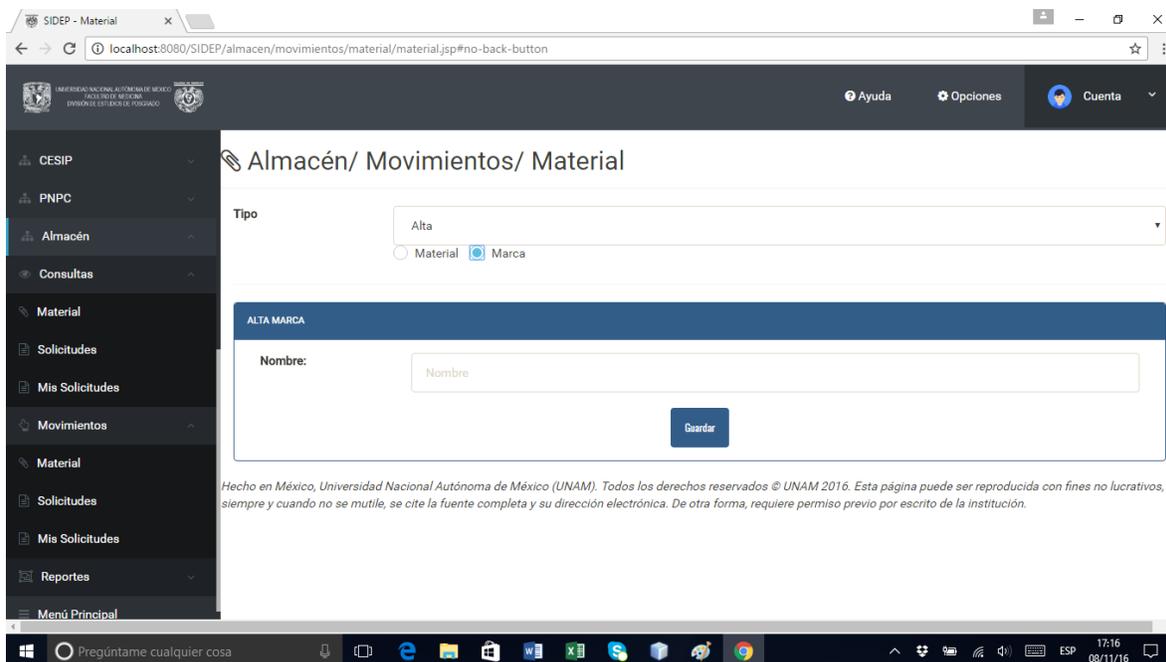


Fig. 9.11 — Alta marca

Para actualización de material, se podrá actualizar nombre, cantidad (actualizar o sumar existencia de material), marca y observaciones de dicho material.

En la Figura 9.12, se muestra la ventana para el administrador para realizar movimientos de actualización de material:

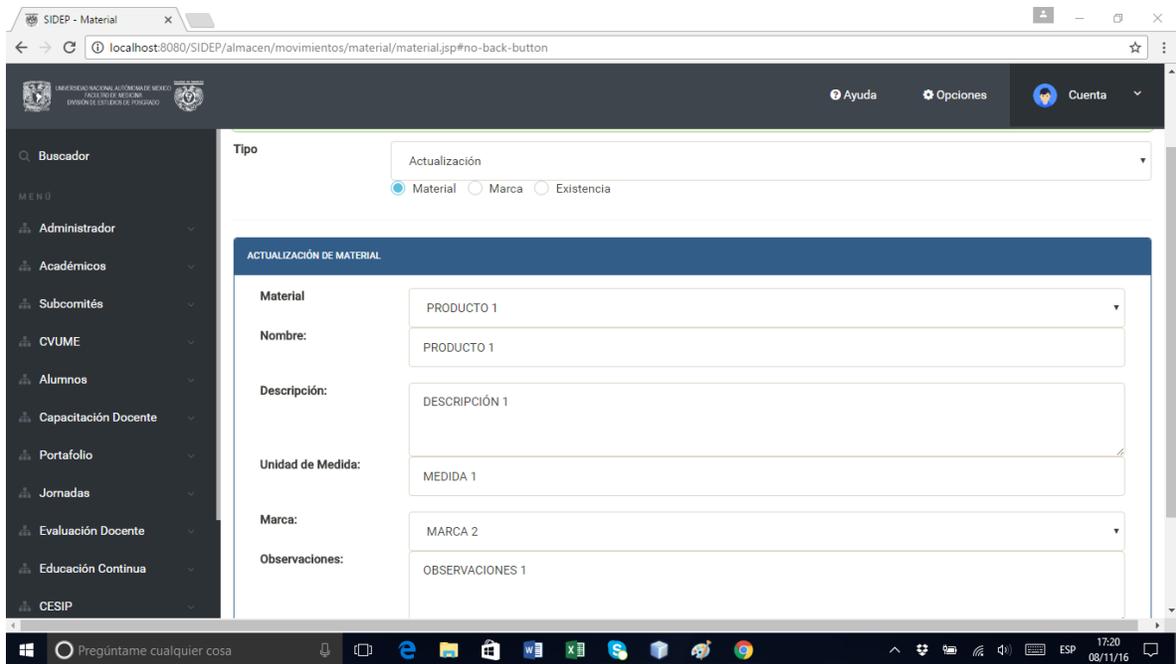


Fig. 9.12 — Actualización material

En la Figura 9.13, se muestra la ventana para el administrador para realizar movimientos de actualización de marca de material:

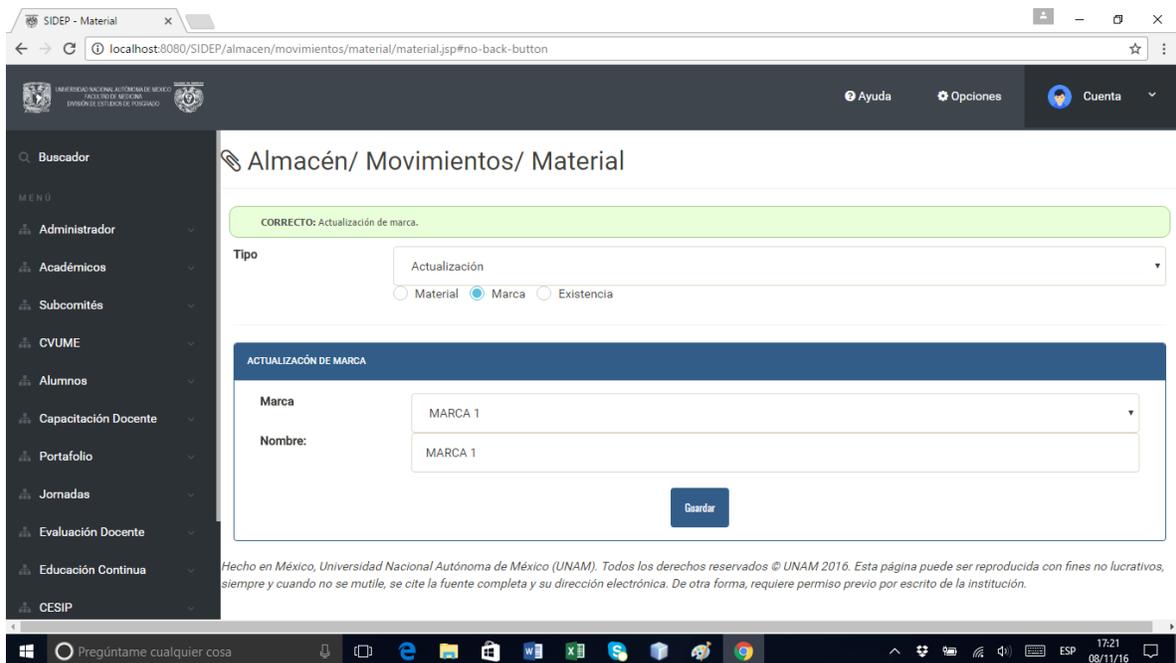


Fig. 9.13 — Actualización marca

En la Figura 9.14, se muestra la ventana para el administrador para realizar movimientos de actualización de existencia de material el cual se podrá actualizar de dos maneras la existencia de material, actualizar existencia o sumar existencia:

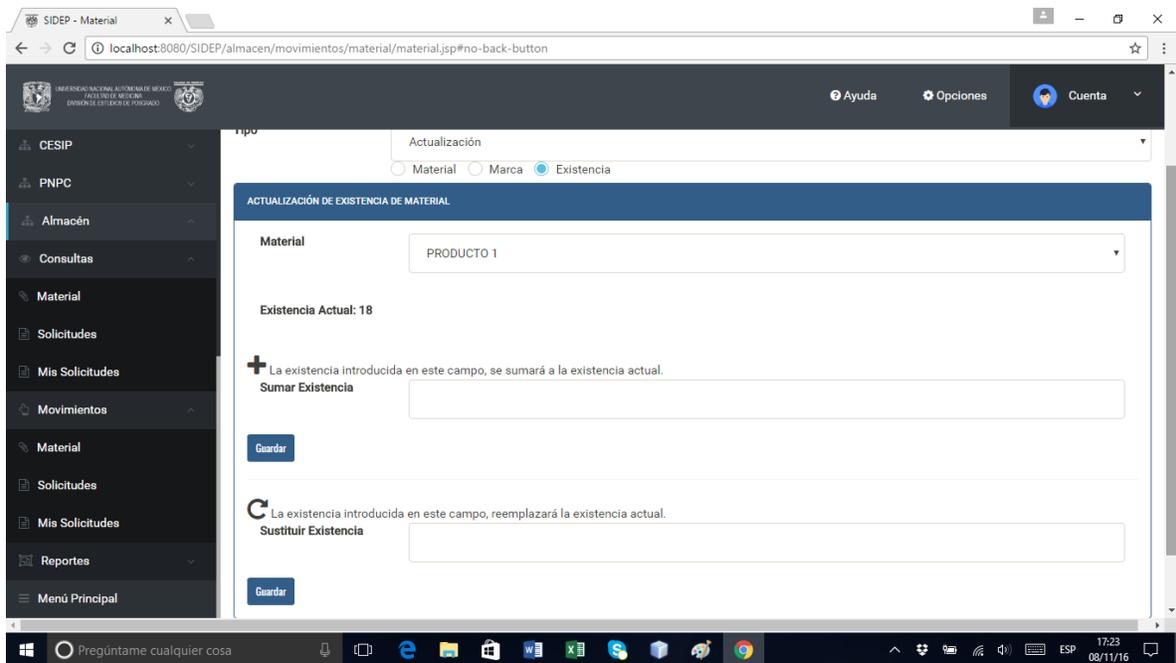


Fig. 9.14 — Actualización existencia

En baja se podrá dar de baja algún material en específico o alguna marca. Al dar de baja una marca se dará de baja de manera automática todos los materiales relacionados con dicha marca.

En la Figura 9.5, se muestra la ventana para el administrador para realizar movimientos de baja de material, cabe señalar que cuando se realiza una baja se le manda una alerta visual al usuario para que valide la eliminación:

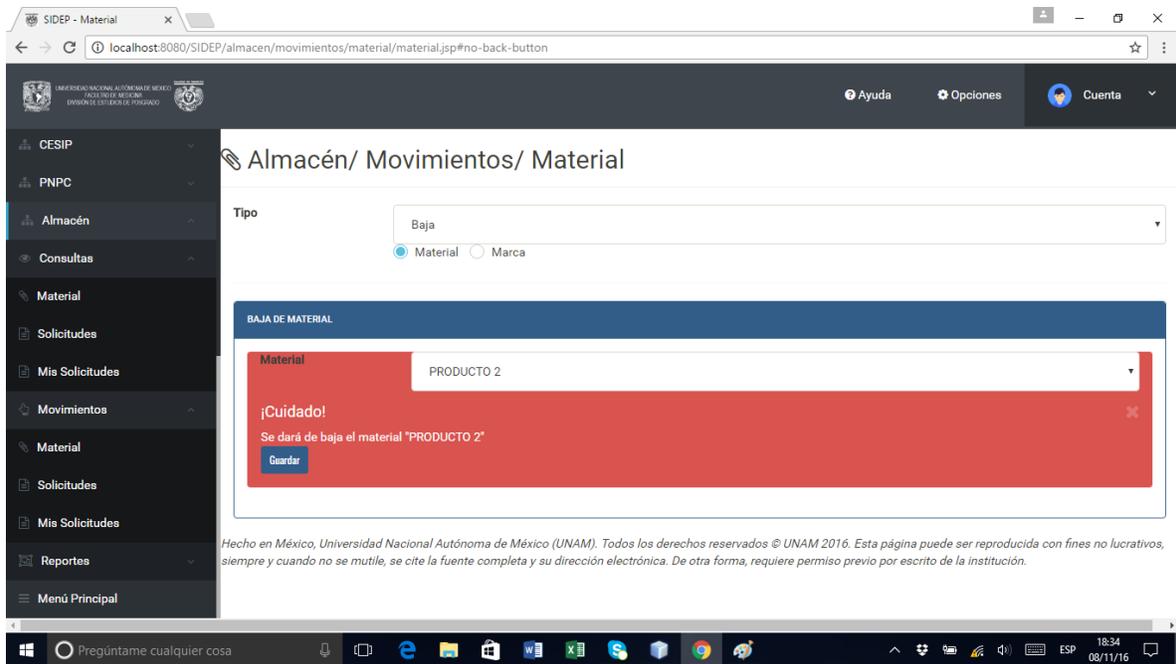


Fig. 9.15 — Baja de material

En la Figura 9.16 se muestra la ventana para el administrador para realizar movimientos de baja de marca:

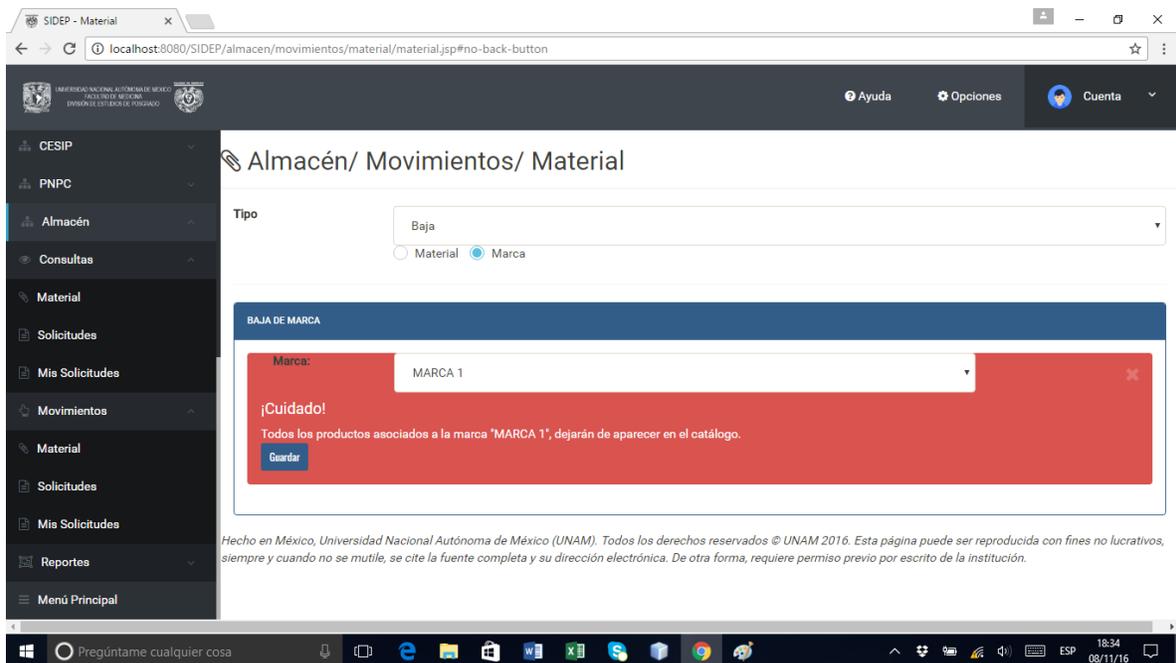


Fig. 9.16 – Baja marca

Movimientos de solicitudes:

Un usuario con permisos de usuario solicitante, podrá realizar solicitudes de material mediante una lista proporcionada por el administrador del sistema la cuál

contendrá la lista y existencia del material del almacén. De la misma manera podrá cancelar solicitudes previamente realizadas por cualquier motivo.

A continuación se muestra la ventana de solicitud de material para el usuario:

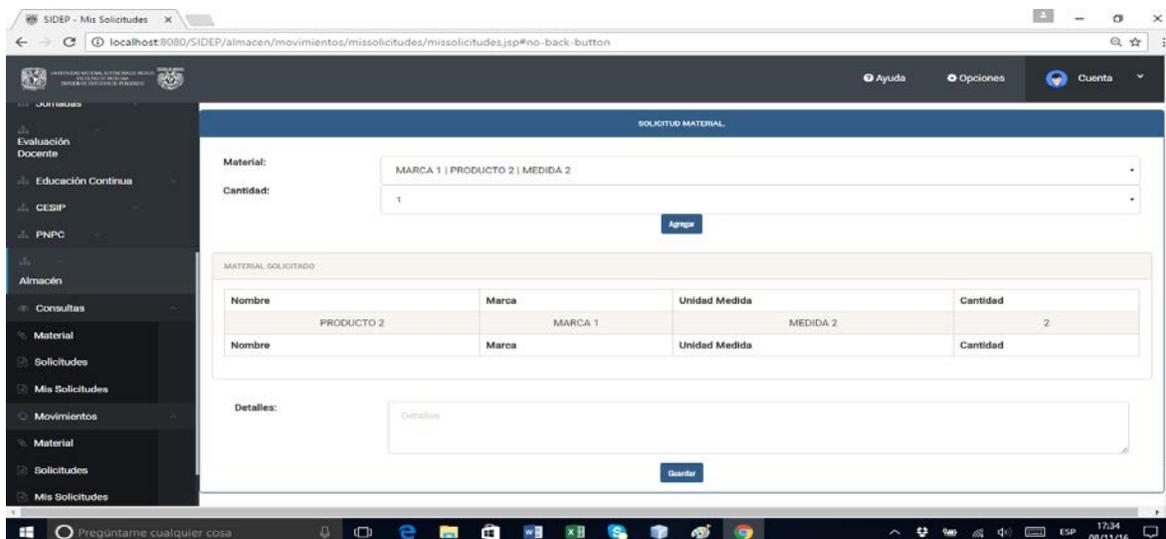


Fig. 9.17 – Solicitud material

El usuario podrá agregar observaciones de su solicitud las cuales se verán reflejadas para el administrador, el cuál determinará si se aprueba o rechaza dicha solicitud.

El administrador podrá consultar todas las solicitudes de material y podrá realizar cambios en dichas solicitudes (cantidad solicitada, autorizada y de la misma manera podrá aprobar o rechazar solicitudes).

A continuación se muestra una consulta que puede hacer el administrador para ver todas las solicitudes de todos los usuarios que han solicitado material.

Dichas solicitudes las podrá consultar por: todas, aprobadas, rechazadas, en proceso no revisadas, en proceso revisadas y canceladas por el usuario.

A su vez podrá realizar dichas consultas por filtros para obtener todas las consultas de un año y/o mes en específico.

A continuación (ver Figura 9.18) se muestra la imagen de dicha consulta del administrador.

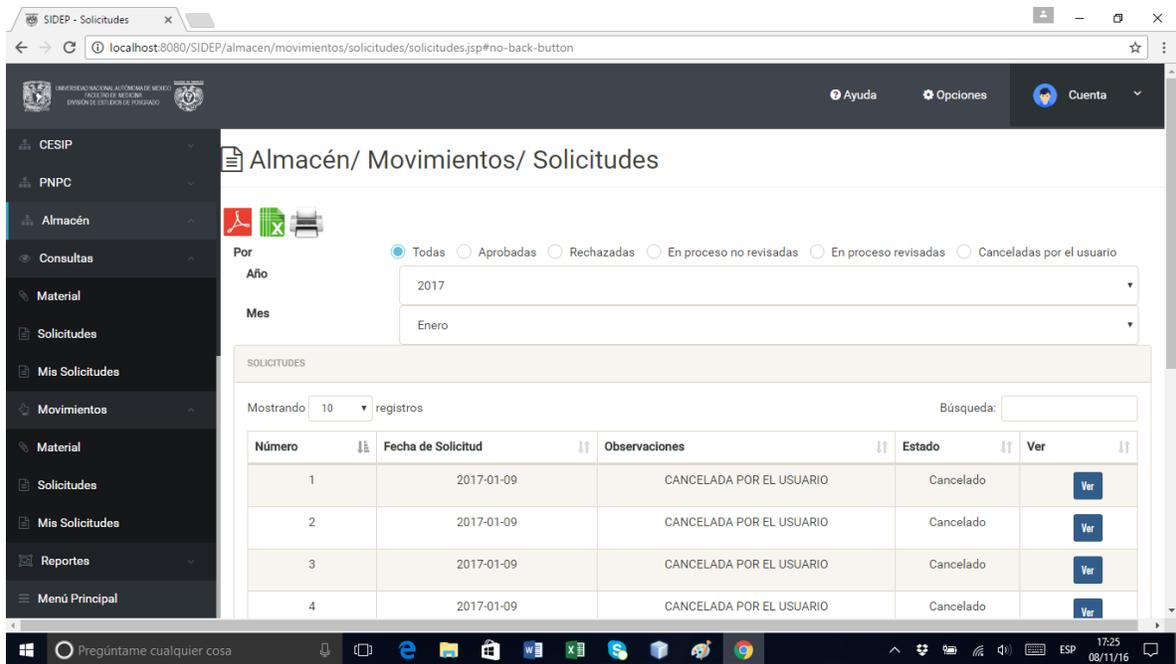


Fig. 9.18 – Consulta administrador

Al entrar en el detalle de una solicitud, el administrador podrá modificar las solicitudes por el usuario y podrá aprobar o rechazar la solicitud, también podrá añadir observaciones si es necesario (ver Figura 9.19).

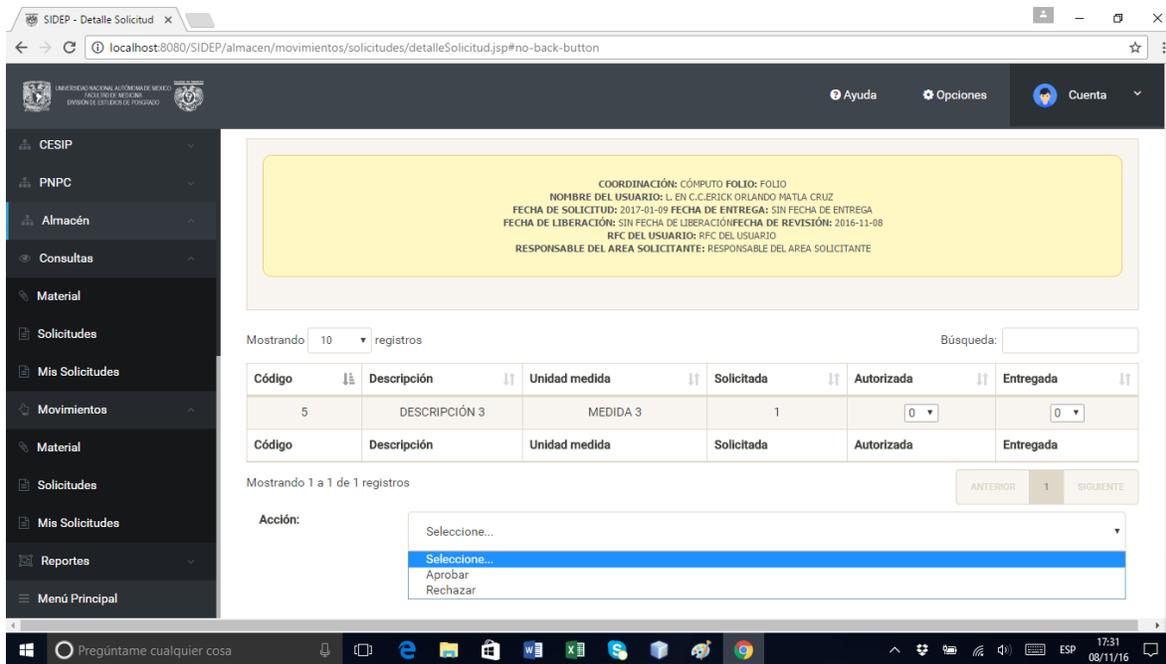


Fig. 9.19 — Detalle solicitud

9.4. Verificación del manual por parte del usuario.

Se realizó una reunión de trabajo con el usuario donde se verificó el manual de mantenimiento mencionado anteriormente. Dicho manual fue aprobado de manera satisfactoria. El manual de mantenimiento se incorporó al repositorio.

En la misma junta, se llevó a cabo de manera exitosa la entrega del sistema, obteniéndose el documento aprobatorio de satisfacción por parte del cliente, misma que avala el término y entrega satisfactoria del proyecto. Los documentos se pueden observar en las Figuras 9.1 y 9.2:



Fecha: 9 de diciembre del 2016

Entrega final del software

División de Estudios de Posgrado de la Facultad de Medicina.

De una parte, Dr. Rogelio Chavolla Magaña, Secretario Académico de la División de Estudios de Posgrado de la Facultad de Medicina de la Universidad Nacional Autónoma de México, en adelante "El Cliente".

De otra parte, L. en C. Erick Orlando Matla Cruz, Ingeniería de Software y base de datos de la División de Estudios de Posgrado de la Facultad de Medicina de la Universidad Nacional Autónoma de México, en adelante "Administrador de Proyecto".

De otra parte, Aramis Arias Romero, prestador de servicio social con número de cuenta 30418532-0 y proveniente de la Facultad de Ciencias de la Universidad Nacional Autónoma de México, en adelante el "Equipo de trabajo".

En este acto, el equipo de trabajo entrega el módulo con todos sus documentos:

- Código fuente del software y sus pruebas, arquitectura necesaria para su correcto funcionamiento, entorno de control de versiones y el resto de la documentación.
- Manual de Usuario, con toda la información necesaria para que los usuarios utilicen correctamente la aplicación, desglosado por los diferentes perfiles del usuario.
- Guía de instalación, con la información necesaria para la implementación de la aplicación.

El **Cliente** recibe el módulo "Administración y abastecimiento del almacén de la División de Estudios de Posgrado de la Facultad de Medicina. UNAM." acoplado al sistema Sidep con toda su documentación.

El **Equipo de trabajo**, el Administrador de proyecto y cliente dan por realizada la entrega del software pactado.

Fig. 9.1 – Documento final de entrega de producto.



UNIVERSIDAD NACIONAL
AVÁNEMA DE
MÉXICO

Fecha: 9 de diciembre del 2016

Entrega final del software

Este documento se firma por duplicado y a un solo efecto.

De conformidad de las tres partes:

El Cliente

Dr. Rogelio Chavolla Magaña

El Administrador del proyecto

L. en C.C. Erick Orlando Matla Cruz

El Equipo de Trabajo

Aramis Arias Romero.

Fig. 9.2 – Documento final de entrega de producto.

10. RESULTADOS

Después del diseño y proceso del software que se llevó a cabo, se obtuvo de manera satisfactoria el sistema nombrado “**Sistema de administración y abastecimiento del almacén de la División de Estudios de Posgrado de la Facultad de Medicina**” v.1.0., el cuál fue montado en un servidor de la unidad de posgrado.

El sistema montado en el servidor mencionado anteriormente, se puede verificar en la siguiente dirección:

<http://132.247.127.131:8080/almacen/>

El sistema está diseñado para futuras correcciones y extensión de funcionalidades.

Por ser la primera versión del sistema, se podrán aplicar extensiones y modificaciones tanto para la base de datos como en las funcionalidades del propio sistema para el usuario final.

Se podrán agregar tablas a la base de datos, llaves foráneas y primarias así como también funciones y métodos extras que mejoren la interacción del usuario con el sistema.

El sistema es compatible con distintos tipos de navegadores, como por ejemplo Google Chrome, Internet Explorer, entre otros.

El módulo de movimientos tanto para el usuario, como para el administrador, mantiene la integridad de los datos al permitir realizar cambios en la base de datos de manera dinámica y eficiente. Tales movimientos que puede realizar el usuario son altas, bajas y actualizaciones de datos del material correspondiente, así como movimientos en las solicitudes.

Se podrá recuperar de manera eficiente datos de la base a través de consultas dinámicas mediante procedimientos almacenados creados en la base de datos.

Al utilizar el patrón de arquitectura **Modelo Vista Controlador** (MVC) para el desarrollo del sistema, nos facilitará la actualización y extensión del sistema ya que se encuentra de manera más organizada y separadas todas las funcionalidades del sistema, con lo cual será más fácil cubrir futuras necesidades del cliente.

Por ultimo hay que señalar que el sistema está en funcionamiento y operando de manera satisfactoria y ha impactado de manera positiva a más de 100 usuarios que son los que laboran en la división de estudios de posgrado de la facultad de

medicina, optimizando tiempos y haciendo más eficiente la solicitud y entrega de material del almacén, también ha impactado de manera positiva en la parte administrativa de dicha división, ya que se puede llevar un seguimiento más preciso de las entradas y salidas del almacén y así elaborar reportes más precisos para sus auditorías.

11. CONCLUSIONES

El presente desarrollo tuvo como objetivo automatizar el manejo y administración de la entrada y salida de material de almacén, así como las solicitudes de abastecimiento y suministro de bienes de la DEP-FM, manejar de una manera segura e íntegra la información del almacén y de la solicitud del material del mismo mediante el sistema de gestión desarrollado a lo largo de este reporte.

Con la conclusión de este desarrollo, el usuario administra de manera más organizada y controlada las solicitudes de material que realiza en la institución, así mismo, podrá llevar un control del material que solicita y consultarlo siempre que lo desee.

Durante el desarrollo de este proyecto, se pudo observar la importancia que implica llevar una buena documentación y metodología para desarrollar cualquier tipo de software, ya que con esto se tiene de una manera organizada el código y las funcionalidades del sistema y se facilitará futuras actualizaciones y/o, mantenimiento del sistema. De la misma manera, se provee la abstracción de la lógica del negocio para un programador, para un equipo de trabajo o para un cliente y así maximizar los beneficios, futuros requerimientos y cubrir las necesidades del cliente. Todo esto se resume en haber adoptado buenas prácticas en cada una de las etapas del desarrollo.

Además, el trabajo permite observar que la aplicación de un a algún sistema permite una mejor organización y entendimiento del problema, y así al culminar el proyecto obtener un producto de mejor calidad. De la misma manera, en el patrón de arquitectura utilizado, se involucró al cliente (desde el principio) en el proyecto mediante juntas y así retroalimentaciones con el fin de cubrir todas las necesidades especificadas y detectadas desde el principio y las fueron surgiendo en el transcurso del desarrollo del mismo.

Para llegar a la culminación del sistema, se plantearon los objetivos, se hicieron análisis de requerimientos, diagramas de casos de usos generales y detallados, reuniones con el cliente. Con lo cual llevamos un desarrollo sano, ordenado según las bases de la administración de proyectos junto con las buenas prácticas de la ingeniería de software.

Como desarrollador del sistema puedo decir que se cumplió el objetivo planteado desde el inicio, que era la automatización y organización del material del almacén, el cual se hacía de manera manual por parte de los administradores y personal de la división de estudios de posgrado de la Facultad de Medicina y esto hacía que el proceso se hiciera de una manera lenta y muy ineficiente.

De manera personal obtuve contacto con nuevas tecnologías que ampliaron mi espectro de conocimientos, además de involucrarme en un ámbito de desarrollo que era completamente ajeno a mí, como son las metodologías existentes para

desarrollo de software, herramientas y lenguajes de programación, entre otros, y, dándome cuenta del gran campo de aplicación que actualmente tiene las ciencias de la computación y la programación.

BIBLIOGRAFÍA

- (Silberschatz, 2010). Abraham Silberschatz, Henry F. Sirbersrchatz, S. Sudarshan. Database System Concepts, Mcgraw-Hill education, 2010.
- (Post, 2006). Gerald v. Post, Sistemas de administración de base de datos, Mcgraw-Hill 2006
- (Sommerville, 2005). Sommerville, Ian. Ingeniería del software. Addison-Wesley, 2005
- (Van Vliet, 2008). Hans Van Vliet, Software engineering: principles and practice, 3rd edition, John Wiley & Sons, 2008
- (Jacobson, 2000). Ivar Jacobson, Grady Booch, James Rumbaugh. El Proceso Unificado del desarrollo del software. Addison-Wesley.
- (García, 2009). Hector García-Molina. Database System: The complete book, Pearson Education India, 2009.
- (Ramez, 2007). Ramez Elmasri, Shamkant B. Navathe. Fundamentos de sistemas de bases de datos. Pearson Educación, 2007
- (Pressman, 2006). Roger S. Pressman. Ingeniería del software: un enfoque práctico. McGraw-Hill, 2006
- (Murat, 2015). Murat Yener, Alex Theedom. Model View Controller Pattern. John Wiley & Sons, 2015

APÉNDICE

Almacenista: Persona encargada de entregar productos autorizados por el administrador.

Usuario: Personal inscrito la unidad de posgrado que usará los productos del almacén.

Administrador: Persona responsable de optimizar y controlar los recursos existentes del almacén.

Almacén: Local que sirve para depositar o guardar gran cantidad de artículos, productos o mercancías para su posterior venta, uso o distribución.

Seguridad: Confianza de manejo de la información contenida en el sistema.

Abastecimiento: Actividades que posibilitan la identificación y la compra de los bienes y de los servicios que el almacén necesita para funcionar.

Interfaz gráfica: Forma de comunicación entre un programa y el usuario.

Escalable: (Escalabilidad) Propiedad de aumentar la capacidad de trabajo o de tamaño de un sistema sin comprometer el funcionamiento y calidad normales del mismo.

Mantenimiento: Conservación del sistema en buen estado o en una situación determinada para evitar su degradación o funcionamiento incorrecto.