

Universidad de las Ciencias Informáticas

Facultad 6



Gestor de expedientes técnicos de medios informáticos

Trabajo de Diploma para optar por el título de Ingeniero Informático

Autora: Yohana Castillo Martínez

Tutor: Ing. Harold Williams Guerra Carrancá

La Habana, Julio de 2016

“Año 58 de la Revolución”



“Nada en la vida debe ser temido, solamente comprendido. Ahora es el momento de comprender más, para temer menos”.

Marie Curie

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis que tiene por título: Gestor de expedientes técnicos de medios informáticos y se reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yohana Castillo Martínez

Firma del Autor

Harold Williams Guerra Carrancá

Firma del Tutor

DATOS DE CONTACTO

Autor:

Yohana Castillo Martínez

Correo: yohana@uci.cu

Tutor:

Harold Williams Guerra Carrancá

Correo: haroldg@aica.cu

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Función: Administrador de redes, Especialista de soporte y mantenimiento de los laboratorios farmacéuticos AICA.

Graduado en la Universidad de la Ciencias Informáticas (UCI) por curso para trabajadores. Ejerció como técnico de laboratorios en ese centro durante 10 años.

Estudios Realizados: Curso de reparación de UPS, de Programación web Avanzada, de Programación orientada a servicio SOAP y básico de Linux, avanzado de Linux.

Conocimientos avanzados: Java, PHP, J2EE, JQUERY, EXTJS, PRIMEFACES, Ensamblador, SQL, BASH, Electrónica.

Otras habilidades:

- ✓ Instrucción y capacitación de cursos Informáticos.
- ✓ Administración de Redes
- ✓ Diseño WEB

AGRADECIMIENTOS

Como se ha hecho de toda la vida pues me toca agradecer a muchas personas que de alguna manera u otra han hecho que este momento se convierta en realidad;

- *A mis adorados y hermosos padres les doy gracias por ser incondicionales por seguirme siempre a donde vaya, por escuchar Hip Hop y No Reguetón, Gracias por eso.*
- *A mis abuelos por darme tanto cariño y por jugar un papel tan importante en mi vida guiándome siempre y malcriándome, como nadie sabe hacerlo.*
- *A mi chuki por ser tan dedicado, por tener tanta paciencia, porque siempre sacas lo mejor de mí y me haces reír todo el tiempo, no importa lo mal que la esté pasando. “Te AMO”*
- *A mi tío por apoyarme en las buenas y en las malas por cuidar siempre de mí y de la familia ahh y de los pajaritos.*
- *A Ayme que ha sido mi otra mitad, mi amiga, mi hermana y compañera de viaje de toda la vida durante tantos años, gracias por estar siempre.*
- *A Idania por ser una segunda madre para mí y apoyarme en todo momento.*
- *A Yiliana por no permitir que mi Introducción pareciera el guión de un capítulo del Narrador de Cuentos, por apoyarme en los momentos más difíciles con sus consejos y su optimismo, gracias por ser mi amiga.*
- *A VICO por ser incondicional para mí y para nuestra familia.*
- *A Yanelis la mama de los pollitos como ella dice, ha sido lo mejor que nos ha pasado en estos 6 años de estudio y sacrificio y me atrevo a hablar por todos porque sé que*

también están de acuerdo, nos enseñó a perseverar, a salir adelante y no dejarnos caer ante ninguna adversidad, gracias porque eres un ejemplo a seguir como profesora y como ser humano.

- *Y los que no podían faltar y que sin ellos habrían sido 6 años esquemáticos, fríos y de dolor de cabeza, mi grupo 6601 o 6602 siempre se me olvida el número. Karen gracias por ser una increíble amiga, mi compañera inseparable hasta que llegó Carolina y nos rompió el Dúo para siempre, “Te Quiero”. Roberto me encanto compartir contigo todas las experiencias que vivimos en clases, tus bromas y tener que explicarte los ejercicios más de 3 veces, ponías a prueba mi paciencia. Ernesto gracias a ti me superé programando fuiste mi ejemplo a seguir, aunque yo sé que tu ejemplo a seguir es Harold, pero bueno no me pongo brava por eso,” LO ENTIENDO”. Arlen gracias por ser tan servicial por abogar siempre por el bien del grupo y aguantar a Roberto todo este tiempo. A Yaima por tener tanta paciencia con nosotros por imprimirnos las clases y por ser una excelente amiga y compañera de aula. A Andry por enseñarnos que no todo lo que estudiábamos era suficiente, siempre había algo más. A Frank no olvido nunca cuantas veces tuve que ir a tu casa para que me ayudaras con I.O y de casualidad siempre me cogía la hora de almuerzo, gracias por compartirlo conmigo. A Suleiny que gracias a ella siempre sabíamos si había Wii-Fi. A la más centrada, madura y cuerda del aula Irasay por enseñarnos a ser más responsables, organizados y estudiosos. GRACIAS a todos.*

- *A Dafne que siempre se preocupa por nuestro bienestar, a zobeida, a Saily que dentro de su locura siempre tuvo tiempo para preocuparse por mí, gracias por eso, a yadriel, a portillo, a yordanis, a yadian mi hermanito más chiquito, Jorge, Arian “El Flaco”, gracias por cubrirme, los quiero.*

- *Por último y no menos importante a Maya y a Sura que son una gran inspiración para mí y le dan color a cada uno de mis días.*

DEDICATORIA

 todos los que han hecho que cada día cuente.

RESUMEN

Los laboratorios farmacéuticos AICA, tienen como misión brindar salud y calidad de vida a la población mediante la producción de medicamentos genéricos de avanzada. Esta institución dispone de tecnologías que complementan al proceso productivo, entre ellas se encuentran presente los medios informáticos. El control de los mismos se torna muy difícil, debido a que son muchos los equipos de cómputos que no poseen la misma locación ni comparten las mismas características. Actualmente el proceso de gestión de los medios informáticos se efectúa de forma manual, trayendo como consecuencia que se dificulte la búsqueda de información, se presenten errores de escrituras y algunos datos sean redundantes. Para darle solución al problema se desarrolló un gestor de expedientes técnicos de medios informáticos que brinda la posibilidad de gestionar la información de manera segura, persistente y dejar un menor margen de error. Este gestor está desarrollado usando la edición empresarial de Java (JEE6), que incluye JSF, Primeface y Bean, utiliza como gestor de base de datos PostgreSQL y está guiado por la metodología ágil Open Up, la cual permite la generación de artefactos necesarios para el proceso de reparación y mantenimiento del gestor.

PALABRAS CLAVE: Gestor, Medios informáticos, Control.

SUMMARY

AICA pharmaceutical laboratories, whose mission is to provide health and life quality of the population by producing generic drugs advanced. This institution has technologies that complement the production process, including computer media are present. The control them becomes very difficult because they do not have the same location or share the same characteristics. Currently the process of management of information technology is done manually, consequently resulting in the search for information is difficult, errors scriptures and the use of data submitted are redundant. For solving the problem a manager of technical files of information technology that provides the ability to manage information securely, persistently and leave a smaller margin of error was developed. This manager is developed using the enterprise edition of Java (JEE6) including JSF, PrimeFaces and Bean, used as manager PostgreSQL database and is guided by the agile methodology Open Up, which enables the generation of devices necessary for the process of repair and maintenance manager.

KEYWORDS: Manager, Computer Media, Control.

ÍNDICE

Datos de Contacto	III
Introducción	1
Capítulo 1: Fundamento teórico.....	5
Introducción	5
1.1 Conceptos fundamentales.....	5
Sistema de Gestión de Información.....	5
Sistemas ERP	6
PKI.....	6
Firma Digital.....	7
Aplicaciones informáticas	8
1.2 Estudio de soluciones existentes	9
OCS Inventory.....	9
OCS+GLPI.....	9
VERSAT-ZARASOLA.....	9
RODAS XXI	10
Exact Globe	10
CEDRUX.....	10
1.3 Herramientas, Tecnologías y Lenguajes	11
Metodologías de desarrollo de software	11
Lenguaje de modelado.....	12
Herramienta CASE (Computer-Aided Software Engineering)	13
Sistemas gestores de base de datos (SGBD)	14
Lenguajes de programación seleccionados.....	16
Otras tecnologías y lenguajes usados.....	18
IDE (Integrated Development Environment)	19
Librerías o Framework usados	20
Servidor de Aplicaciones.....	25
Máquina virtual JVM.....	25
Conclusiones parciales	26
Capítulo 2: Características del sistema	27
Introducción	27
2.1 Descripción del Sistema Propuesto	27
2.2 Modelo Conceptual.....	27
2.3 Descripción de las entidades	28
2.4 Especificación de los requisitos del sistema	29

Requisitos funcionales	29
Requisitos no funcionales	30
2.5 Diagrama de Casos de Uso	33
2.6 Descripción textual de los casos de uso.....	34
2.7 Patrones de casos de usos	39
Conclusiones Parciales	40
Capítulo 3: Diseño del sistema.....	41
Introducción	41
3.1 Patrones.....	41
3.1.1 Patrones Arquitectónicos	41
3.1.2 Patrones de diseño	42
Patrones de diseño GRASP	42
Patrones de diseño JEE6.....	45
3.2 Diagrama de clases del diseño	49
3.3 Diagrama de secuencias.....	50
3.4 Diseño de base de datos.....	50
3.5 Diagrama de despliegue	51
Conclusiones parciales	52
Capítulo 4: Implementación y Resultados esperados.....	53
Introducción.....	53
4.1 Diagrama de Componentes.....	53
4.2 Estándares de Codificación	55
4.3 Pruebas.....	56
4.4 Análisis de los resultados esperados.....	59
Prueba de carga.....	60
Prueba de rendimiento	60
Conclusiones Parciales	63
CONCLUSIONES	64
RECOMENDACIONES	65
REFERENCIAS BIBLIOGRÁFICAS	66
ANEXOS.....	68

INTRODUCCIÓN

Con el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha ido evolucionando el pensar del hombre y con ello se ha logrado implantar la utilización de la información como un recurso estratégico de gran valor para el buen desempeño de las organizaciones. La información, elemento del cual se puede extraer conocimiento y satisfacer las necesidades de personas e instituciones, adquiere una importancia significativa para el desarrollo, equilibrio y adaptabilidad en cualquier sector del mundo. Cada día son más las empresas e instituciones las que dedican su esfuerzo a conseguir un buen sistema de información para la automatización de los procesos administrativos, docentes, asistenciales e investigativos (Entonado 2001).

Cuba, como nación, no está ajena a la toma de decisiones acertadas como alternativa para poder fortalecer su economía, por lo que también se ha sumergido en todo este mundo del avance tecnológico, ya que se ha propuesto informatizar toda la sociedad empresarial con la construcción de diferentes sistemas informáticos.

Un ejemplo de ello es la creación de los sistemas de Planificación de Recursos Empresariales (ERPs). Sistema integral de gestión empresarial que está diseñado para modelar e informatizar la mayoría los procesos de las empresas entre los que se encuentran, procesos de finanzas, comerciales, logísticos y de producción, donde su misión es facilitar la planificación de todos los recursos de una institución. Los ERPs proporcionan la posibilidad de estructurar todo el proceso productivo y de gestión sobre la base de los flujos de información para aprovechar todos los conocimientos que los datos aportan. Aunque por las peculiaridades de este, no cubre todas las necesidades de las industrias atípicas, como es el caso de la industria farmacéutica, debido a que varios de los módulos con los que cuentan estos sistemas, no se adaptan a las características del modelo de negocio que posee la empresa (Agüero 2012).

La industria farmacéutica cubana tiene la responsabilidad de garantizar la disponibilidad de medicamentos para el consumo del pueblo y de otros países que lo necesiten. Es importante mencionar que dentro de estas empresas se encuentran los laboratorios farmacéuticos AICA con un historial de más de 10 años, los cuales están enfocados a la producción de medicamentos genéricos de avanzada, con destino al mejoramiento de la salud del pueblo y garantizando la excelencia, responsabilidad y sostenibilidad, que consolida su imagen de confianza frente al mercado y la sociedad. Para brindar un servicio de excelencia, dicha empresa posee tecnologías que complementan el proceso productivo.

Hoy en día en los laboratorios AICA existen varios procesos que no se encuentran informatizados, como son, la gestión de la calidad, la orden de producción y el control de los medios informáticos. Este

Introducción

último tiene un gran impacto en la industria farmacéutica, ya que sin los mismos el proceso productivo sería un fracaso total.

La labor realizada por los especialistas informáticos que trabajan en la entidad cada día se hace más compleja debido a que tienen que monitorear manualmente un gran número de fuentes de información generada por los expedientes técnicos, los cuales se gestionan de forma manual, provocando esto algunas deficiencias en la gestión y el control de la información, encontrada en diferentes áreas.

Actualmente, al no contar con las herramientas que posibiliten el procesamiento de la información, se presentan algunos problemas que atentan contra el correcto funcionamiento de los laboratorios farmacéuticos AICA:

- ✓ La inexistencia de un sistema de gestión para el control de la información.
- ✓ Los registros no se llenan con la calidad requerida debido a que los especialistas se ven sobrecargados de tareas y hay detalles que pasan desapercibidos.
- ✓ Se obvia información importante, debido a que el proceso de gestión se realiza manualmente produciéndose aglomeración de la documentación, lo que provoca problemas en la toma de decisiones, a la hora de realizar la compra de equipos o la defectación de los mismos.
- ✓ El control de la disponibilidad tecnológica posee un nivel de calidad muy bajo, por lo que el proceso de análisis de la documentación es lento y no llega la misma en el momento oportuno y con la actualización requerida.
- ✓ Los expedientes del equipamiento informático, así como la mayoría de la documentación se encuentra recogida en una fuente poco fiable, trayendo esto como consecuencia que el proceso de búsqueda de la información se ralentice.
- ✓ Se está viendo afectado el tiempo que se necesita para hacer cada una de estas tareas, debido a la cantidad de información que requieren las mismas.

A partir del análisis de la problemática expuesta anteriormente surge el siguiente **problema a resolver**: ¿Cómo contribuir al proceso de gestión de los expedientes técnicos en los laboratorios farmacéuticos de AICA?

Para ello se identificó como **objeto de estudio**: El proceso de gestión de la información.

Enmarcado en el siguiente **campo de acción**: El proceso de gestión de la información de los expedientes técnicos de medios informáticos.

Para dar solución a la problemática se plantea como **objetivo general**: Desarrollar un sistema informático que permita contribuir a la gestión de la información de los expedientes técnicos en los laboratorios farmacéuticos de AICA.

Introducción

Se define como **idea a defender**: Con el desarrollo de un sistema informático se contribuirá a la gestión de la información de los expedientes técnicos de los medios informáticos en los laboratorios farmacéuticos AICA.

Las **Tareas de la investigación son**:

- ✓ Elaboración del marco teórico referencial del sistema de gestión de los expedientes técnicos de los laboratorios farmacéuticos AICA.
- ✓ Selección de las herramientas y tecnologías de desarrollo a utilizar en la elaboración del sistema de gestión de los expedientes técnicos.
- ✓ Elaboración del diseño para el sistema de gestión de los expedientes técnicos con el objetivo de facilitar la implementación del sistema informático.
- ✓ Implementación del sistema de gestión de los expedientes técnicos para el cumplimiento de los requisitos funcionales.
- ✓ Realización de pruebas del software para garantizar el buen desempeño del sistema de gestión de los expedientes técnicos.

Para obtener los conocimientos necesarios que hagan posible el cumplimiento del objetivo trazado, se lleva a cabo una investigación en la que se utilizan algunos de los métodos científicos existentes, tanto teóricos como empíricos.

Teóricos:

- ✓ **Histórico lógico**: Se aplicó para constatar teóricamente cómo han evolucionado los sistemas de información, lo que permitió conocer cómo funcionan las tendencias más recientes en la gestión y control de la información y realizar un análisis real para identificar los problemas existentes dentro del proceso actual de la gestión de los medios informáticos.
- ✓ **Analítico-Sintético**: Permitted al autor realizar un estudio sobre los elementos más importantes relacionados con el proceso de gestión de la información y posteriormente sintetizar la información necesaria para la gestión y el control de los medios informáticos.
- ✓ **Modelación**: Se utilizó para representar por medio de diagramas el proceso de gestión y control de información de los medios informáticos. Lo que permitirá una mayor comprensión de los procesos y lineamientos de desarrollo a seguir en cada una de las fases, a partir de su utilización para el modelado del sistema, usando el Proceso Unificado de Desarrollo (OpenUp) y el Lenguaje Unificado de Modelación (UML).

Empíricos:

- ✓ **Observación**: Facilita conocer el panorama real de una situación mediante la percepción directa. Con la utilización de este método se determinaron los rasgos imprescindibles para el desarrollo del sistema informático.

Introducción

- ✓ **Entrevista:** Se realizaron entrevistas a los usuarios del sistema informático, para garantizar que los errores y deficiencias que están presentes en el proceso de gestión actual se erradiquen y se mejoren los servicios existentes.

Aporte práctico

Con el sistema informático para la gestión de la información de los expedientes técnicos de medios informáticos en los laboratorios AICA, el conjunto de trabajadores y especialistas de la empresa, obtendrá una nueva herramienta para su desempeño diario que contribuye al proceso de gestión de la información de los expedientes técnicos con que cuenta actualmente la empresa.

Estructura del documento

El documento está conformado por: Resumen, Introducción, cuatro Capítulos, Recomendaciones, Glosario de siglas y términos, Referencias bibliográficas, Bibliografía y Anexos.

En el **Capítulo 1** (Fundamento teórico), se hace una descripción de los conceptos relacionados con los sistemas de información, las tendencias, técnicas, tecnologías, metodologías y el análisis de algunas soluciones existentes a nivel internacional y nacional que son de interés por tener estrechos lazos con el alcance de esta investigación.

En el **Capítulo 2** (Análisis de la aplicación), describe las características del sistema informático a desarrollar. Se realizan las especificaciones de las funcionalidades del sistema.

En el **Capítulo 3** (Diseño de la aplicación), se presentan los diagramas de clases del diseño, diagramas de interacción y diagrama de despliegue de la aplicación y se especifica la arquitectura que tendrá el sistema propuesto.

En el **Capítulo 4** (Implementación y Resultados esperados), se presenta la interacción entre los diferentes componentes de software y hardware del sistema mediante el diagrama de despliegue, se implementan los métodos funciones y librerías, se muestran los resultados obtenidos, se realizan un conjunto de pruebas funcionales al sistema y se aplica un método de experto para validar la propuesta presentada por el autor.

CAPÍTULO 1: FUNDAMENTO TEÓRICO.

Introducción

En este capítulo se muestran los conceptos básicos relacionados con la gestión de la información y los diferentes recursos que permite aprovechar un sistema con estas características, también se lleva a cabo el análisis y estudio de la selección de herramientas, tecnologías y lenguajes de programación con el objetivo de sustentar su utilización en el desarrollo de la solución que se propone.

1.1 Conceptos fundamentales

Gestión de la Información

La finalidad de la gestión de la información es ofrecer mecanismos que permitan a la organización adquirir, producir y transmitir, al menor coste posible datos e informaciones con calidad, exactitud y actualidad suficientes para servir a los objetivos de la organización. En términos perfectamente entendibles sería conseguir la información adecuada para la persona que lo necesita, en el momento que lo necesita, al mejor precio posible para tomar la mejor de las decisiones.

En el momento actual parece indiscutible que el éxito de la empresa no dependerá únicamente como maneje sus activos materiales, sino también de la gestión de los recursos de información. La importancia de este recurso es tal que algunos autores estiman que las organizaciones deben ser consideradas como sistemas de información (Carnicero 2011).

Es frecuente confundir un sistema de información con la tecnología que lo soporta. Las Tecnologías de la información han supuesto una auténtica revolución en la capacidad de manejo de los recursos de información, permitiendo un rápido y eficiente proceso de adquisición, enriquecimiento y acceso a la misma, aunque nunca hay que olvidar que un sistema de gestión de información va más allá de las propias herramientas utilizadas.

Las funciones de la Gestión de información abarcan:

- ✓ Determinar las necesidades de información en correspondencia a sus funciones y actividades.
- ✓ Mejora de los canales de comunicación y acceso a la información.
- ✓ Mejora de los procesos informativos.
- ✓ Empleo eficiente de los recursos.

Sistema de Gestión de Información

Es aquél conjunto ordenado de elementos que permiten manipular toda aquella información necesaria para implementar aspectos específicos de la toma de decisiones. Todo sistema de información, surge

Capítulo 1: Fundamento Teórico

de la necesidad de la información que experimenta una organización para implementar un conjunto específico de toma de decisiones.

Algunos autores conceptualizan los sistemas de gestión como un sistema integrado y automatizado para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización.

Un sistema de gestión de información permite la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. El sistema de gestión de información aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos, desde la cúspide estratégica hasta la base operativa (Herrando 2011).

Sistemas ERP

Los sistemas de tipo ERPs (Planificación de Recursos Empresariales) son un sistema integral de gestión empresarial que está diseñado para modelar y automatizar la mayoría de procesos en las empresas, entre los se encuentran, área de finanzas, comercial, logística y producción. Su misión es facilitar la planificación de todos los recursos de la empresa. Estos recursos pueden tener en cuenta, activos tangibles, financieros, pagos y recursos humanos. Los ERP se utilizan para mejorar la manera en la que el flujo de información se comunica entre los departamentos considerados por estar dentro de los márgenes de la empresa o institución (Badenes 2012).

Además, este flujo de información se utiliza de una manera diferente para generar reportes con respecto al estado en que se encuentra la empresa. Lo más destacable de un ERP es que unifica y ordena toda la información de la empresa en un solo lugar, de este modo cualquier suceso queda a la vista de forma inmediata, posibilitando la toma de decisiones de forma más rápida y segura, acortando los ciclos productivos. Con un ERP la empresa tendrá bajo control e incrementará la calidad de los servicios y productos. La implantación de un ERP conlleva la eliminación de barreras inter departamentales, la información fluye por toda la empresa eliminando la improvisación por falta de información.

PKI

El acrónimo PKI corresponde con la traducción al inglés de Infraestructuras de Clave Pública. Estas infraestructuras se basan en criptografía de clave asimétrica, de modo que en todas las operaciones que se llevan a cabo en una interacción intervienen una pareja de claves, una de ellas pública, es decir, que puede ser conocida por todo el mundo y la otra privada, que sólo el propietario de la clave debe conocer. Los usuarios distribuyen su clave pública para poder llevar a cabo operaciones con otros

Capítulo 1: Fundamento Teórico

usuarios, pero en caso de que “pierda” la clave privada, la seguridad de su identidad y de sus datos puede verse seriamente comprometida. Con este contexto, cualquier mensaje puede ser cifrado usando la clave pública que ha sido distribuida por el destinatario y de este modo el contenido del mensaje sólo podrá ser descifrado por el poseedor del certificado que contiene la clave privada correspondiente.

Un sistema PKI permite la creación, almacenamiento y distribución de certificados digitales que son utilizados en este proyecto. Una estructura PKI consiste en una autoridad de certificación (CA), encargada de emitir y verificar los certificados; una autoridad de registro (RA), que verifica la identidad de los usuarios; y un sistema de manejo de certificados.

Firma Digital

La firma electrónica surge de la necesidad de las organizaciones de reducir sus costos e incrementar la seguridad de sus procesos internos, a través del uso de medios electrónicos que permita agilizar los procesos, reducir los tiempos y evitar el uso de papel. Con la entrada en vigor de la Ley de la Firma Electrónica Certificada del Estado de Guerrero se regula la aplicación del uso de la Firma Electrónica Certificada en los actos, procedimientos y trámites que se lleven a cabo entre las dependencias, entidades o cualquier órgano de los sujetos señalados en esta Ley, así como entre éstos y los particulares. Además, otorga el mismo valor jurídico a la Firma Electrónica Certificada que a la Firma Autógrafa y regula el procedimiento de certificación de la Firma Electrónica Certificada y los servicios conexos (Cárdenas 2008).

En el marco de dicha ley, se establece a la Subsecretaría de Asuntos Jurídicos y Recursos Humanos como la Autoridad Certificadora facultada para autorizar, revocar, suspender o extinguir los certificados de Firma Electrónica Certificada.

Implementación de la firma digital

La firma electrónica permite garantizar la identidad de la persona que realiza una gestión, así como la integridad del contenido de los mensajes que envía. Por este motivo, los usuarios que dispongan de la firma electrónica pueden consultar información de carácter personal, realizar trámites o acceder a servicios que impliquen una certificación de identidad del solicitante. Las aplicaciones más relevantes en el uso de firma electrónica, son aquellas que permiten la generación de documentos electrónicos que tengan validez jurídica, comprobable en términos del aseguramiento de la identidad de quienes firman o emiten dichos documentos. Dado que la comprobación de identidad se realiza a través de medios electrónicos, los documentos firmados electrónicamente tienen sustento y validez jurídica únicamente cuando se verifiquen a través de los medios diseñados para dicho efecto.

La implementación de la firma digital con respecto a la firma manuscrita trae como ventajas:

Capítulo 1: Fundamento Teórico

- ✓ Mayor seguridad e integridad de los documentos. El contenido del documento electrónico firmado no puede ser alterado, por lo que se garantiza la autenticación del mismo y la identidad del firmante.
- ✓ Se garantiza la confidencialidad, el contenido del mensaje solo será conocido por quienes estén autorizados a ello.
- ✓ Eliminación del papel, lo que implica una disminución del almacenamiento de datos (espacio físico) y reducción de gastos en los procedimientos de administración de archivos.
- ✓ Se evitan desplazamientos y traslados.
- ✓ Disminución del tiempo en la ejecución de procesos (se evitan colas y se reducen los procedimientos manuales).
- ✓ Aumento de la productividad y competitividad de la Empresa.

Certificados Digitales

El certificado digital es el documento que un usuario participante en un sistema PKI, recibe por parte de una autoridad de certificación (CA), para identificarle con su firma gracias a las claves que contiene. Este mecanismo permite la seguridad de los servicios que se ofrecen y es parte fundamental en este proyecto.

El certificado digital vincula la identidad de un sujeto con sus claves, gracias a que existe el apoyo de una CA, que es un tercero en el que las demás entidades que interactúan con el sujeto confían. El certificado de usuario, que contiene su clave pública pero no la privada, se considera que está compuesto por información no sensible, que el usuario del certificado puede distribuir libremente (Agüero 2012).

Aplicaciones informáticas

Una aplicación informática no es más que un tipo de software (programa de computadora) utilizado para desarrollar una operación o tarea específica, este se clasifica según el principio básico de su funcionamiento en Web, Escritorio (*Desktop*), entre otras.

Aplicación Web

En la Ingeniería de Software se denomina aplicación web, al programa que se codifica en un lenguaje soportado por los navegadores web, en la que se confía la ejecución al navegador. Estas aplicaciones son populares debido a lo práctico del navegador como cliente ligero, a la independencia del Sistema Operativo y a la facilidad para actualizarse y mantenerse.

1.2 Estudio de soluciones existentes

OCS Inventory

Es un software libre que permite a los usuarios administrar el inventario de sus activos de TI. OCS-NG recopila información sobre el hardware y software de equipos que hay en la red, que ejecutan el programa de cliente OCS (agente OCS de inventario).

OCS puede utilizarse para visualizar el inventario a través de una interfaz web. Además, OCS comprende la posibilidad de implementación de aplicaciones en los equipos de acuerdo a criterios de búsqueda. También tiene muchas opciones como escanear la red por medio del IPDiscovery, o instalar aplicaciones remotamente creando Builds (Cohn 2015).

OCS+GLPI

GLPI es una aplicación web de software libre distribuido bajo licencia GPL, que facilita la administración de recursos informáticos, donde sus principales funcionalidades están articuladas sobre dos ejes.

Una excelente idea es integrar GLPI y OCS, ya que juntando el HELP DESK de GLPI más la posibilidad de hacer un inventario de hardware y software totalmente actualizado y automático de OCS. Se puede realizar un seguimiento de todo lo que se tenga inventariado y con ello conseguir estadísticas de falla, seguimiento del manejo del equipamiento de usuario, entre otros. Estas dos herramientas ya son capaces de trabajar en conjunto y además los equipos de desarrollo de ambos proyectos se han propuesto a corto plazo una fuerte integración de funcionalidades.

VERSAT-ZARASOLA

Es un sistema integrado de gestión económica, diseñado por un grupo villaclareño de especialistas económicos e informáticos pertenecientes a la Empresa de Servicios Técnicos Industriales del Grupo Azucarero nacional (TEICO). Se desarrolló para ser utilizado por el sector empresarial cubano, se adecua a las características de cada entidad, es configurable en el momento de su instalación y tiene como objetivo fundamental ofrecerles a los usuarios la posibilidad de contar con un instrumento seguro, rápido, eficaz y de fácil manejo para la Planificación, Control y el Análisis de la Gestión Económica. Es el sistema integral más distribuido en Cuba, con presencia en más de 8000 entidades de 29 organismos, según datos brindados por Miguel Cabrera González, uno de sus principales artífices. Poseen este sistema para su gestión, fundamentalmente las entidades del Sector Público, Empresas Mixtas y Productores Agropecuarios (UBPC y CPA). En la actualidad los especialistas que desarrollaron este sistema se encuentran trabajando para migrar la aplicación hacia una plataforma libre (Santos 2012).

RODAS XXI

Es un software propietario desarrollado por la Empresa cubana de Tecnologías de la Información y Servicios Telemáticos Avanzados. CITMATEL cuenta con más de 600 clientes en el territorio nacional, tiene ocho módulos: Contabilidad General, Activos Fijos, Recursos Humanos, Nóminas, Facturación, Inventario, Finanzas y Gestión de Cobros, también denominado Telecobranzas. Ya se encuentra en desarrollo el de Costos y entre las perspectivas están los módulos de Gestión de Suministros y de Gestión de Clientes (Rodas 2016).

Exact Globe

Es una solución completa de *back office* o ERP que cubre todas las funcionalidades en las áreas de contabilidad, distribución, logística, servicios, RR.HH, nóminas, proyectos y fabricación. Diseñado para cualquier tipo de empresa y capaz de soportar regímenes legales y comerciales en más de 20 legislaciones diferentes. Se encuentra traducido a más de 25 idiomas por lo que resulta altamente recomendable para empresas internacionales.

CEDRUX

Conocido también como ERP cubano, actualmente está siendo desarrollado por un equipo multidisciplinario compuesto por especialistas de diferentes entidades del país y la Universidad de las Ciencias Informáticas.

En su proyección inicial se muestra como un paquete de soluciones integrales de gestión, basado en los principios de independencia tecnológica y en la inclusión de las funcionalidades generales presentes en los procesos característicos de la economía cubana. Se diseña partiendo de las necesidades básicas para las empresas nacionales, los requerimientos gubernamentales pertinentes y las funcionalidades presentes en sistemas similares a nivel mundial. En su implementación inicial se desarrolló para las áreas de contabilidad, costos y procesos, capital humano, planificación financiera y logística, todo construido dentro de una sola plataforma de negocios (Fuentes 2010).

CEDRUX si se implementa como se desea, garantiza la independencia tecnológica de nuestro país, lo cual da cumplimiento a la política de informatización de la sociedad del estado cubano. En estos momentos esta aplicación se ensaya en algunas entidades de características diferentes, entre las cuales se encuentran el Hospital Naval y la Empresa de Envases Rafael Trejo, ambas en La Habana, aún no se encuentra listo para su distribución total al resto de las empresas cubanas.

A partir de las investigaciones realizadas y los datos recopilados, los ERP más usados en el sector empresarial cubano son el Exact Globe entre los ERP extranjeros y el VERSAT Sarasola entre los de producción nacional.

Estos software han sido capaces de brindarle robustez a las empresas cubanas en la gestión de los sistemas integrales, pero se tiene que trabajar mucho en sus potencialidades para ser capaz de poder adaptarlos al modelo de negocio de la industria farmacéutica, ya que algunos de ellos son propietarios y otros aún se encuentran en periodo de prueba, sin mencionar que no es rentable el hacer uso de uno de estos sistemas, debido a que no todos los módulos que poseen se adaptan a los procesos de gestión que se realizan en AICA, como es el caso de la gestión de la calidad y la orden de producción, por lo que no es viable el despliegue de uno de estos, solo para utilizar alguno de sus módulos, debido a que llevaría un costo en software, implantación y capacitación del personal, lo que provocaría que la empresa se afecte monetariamente. Por esta razón se decide llevar a cabo el desarrollo de un sistema informático capaz de gestionar los expedientes técnicos de los medios informáticos de los laboratorios farmacéuticos AICA.

1.3 Herramientas, Tecnologías y Lenguajes

Con la evolución constante de la informática se han perfeccionado las herramientas y tecnologías para el desarrollo de software. Justificar la utilización de cada una de estas ha llevado un estudio detallado de las tendencias actuales y las herramientas que se hacen necesarias para el desarrollo de sistemas de inventarios que utilizan tecnología Web. A continuación, se muestra la selección de las herramientas, metodología y tecnologías para el desarrollo de la solución que se propone.

Metodologías de desarrollo de software

La creciente informatización de los procesos productivos y sociales ha traído consigo que las organizaciones y empresas requieran cada vez más de software confiable y de alta calidad. Es por ello que en los últimos años se han venido publicando estándares, notaciones y metodologías que establecen buenas prácticas para los procesos de desarrollo de software (Cataldi 2000).

RUP (*Rational Unified Process*)

El Proceso Unificado Racional (*Rational Unified Process*), habitualmente resumido como RUP, es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Divide el proceso en nueve flujos de trabajo y cuatro fases, es iterativo incremental, dirigido por casos de uso y centrado en la arquitectura. Está enfocado en los riesgos y es utilizado principalmente en grandes proyectos ya que al ser una metodología pesada genera mucha documentación y alarga el tiempo de producción (Kruchten 2009).

Open UP (*Open Unified Process*)

Open UP posee una gran adaptabilidad a las necesidades de un proyecto en específico y está basada en la potente metodología pesada RUP. Open Up es un proceso unificado ágil y liviano, que aplica un enfoque iterativo e incremental dentro de un ciclo de vida estructurado y contiene un conjunto mínimo de prácticas que ayuda al equipo a ser más efectivo desarrollando software. Abraza una filosofía pragmática y ágil de desarrollo, que se enfoca en la naturaleza colaborativa del desarrollo de software (Gimson 2012).

XP (*Extreme Programming*)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo, con un equipo pequeño y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. XP pone la comprobación como el fundamento del desarrollo, con cada programador escribiendo pruebas cuando escriben su código de producción. Las pruebas se integran en el proceso de integración continua y construcción lo que rinde una plataforma altamente estable para el desarrollo futuro. Se basa en la retroalimentación entre el cliente y el equipo de desarrollo, buena comunicación entre los participantes y simplicidad en las soluciones implementadas. Se preocupa por el aprendizaje de los desarrolladores y propicia un buen clima de trabajo (Fros 2001).

Fundamentos de la metodología seleccionada

Debido a las características que tiene el sistema informático se selecciona la metodología Open UP por las siguientes razones:

- ✓ Posee una gran adaptabilidad a las necesidades de un proyecto en específico.
- ✓ Está dirigida por casos de uso, es iterativa incremental y centrada en la arquitectura.
- ✓ Es extensible.
- ✓ Está pensada para proyectos pequeño.

Lenguaje de modelado

El lenguaje de modelado es un conjunto de signos o modelos que permiten transformar el lenguaje común a un lenguaje más técnico, con el objetivo de obtener una mejor comprensión, visualización y descripción del sistema a desarrollar. Se utiliza para la creación de diagramas de casos de uso, de clases del diseño y modelos de datos.

UML (*Unified Modeling Language*) 2.0

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura

Capítulo 1: Fundamento Teórico

decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas.

Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. Incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. La especificación de UML no define un proceso estándar, pero está considerado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos (Booch 2007).

Herramienta CASE (Computer-Aided Software Engineering)

Las herramientas CASE son una tecnología para automatizar el desarrollo y mantenimiento del software, combinando herramientas de software y metodologías. Estas herramientas deben constituir un conjunto integrado que automatice todas las partes del ciclo de vida y por tanto ahorren trabajo. El uso de Herramientas CASE para la modelación UML proporciona mayor rapidez y entendimiento en el desarrollo de software. Este tipo de aplicaciones se han convertido en ayuda y apoyo imprescindible para los desarrolladores (Cárdenas 2003).

Algunas de estas herramientas son:

Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Además, cuenta con una potente funcionalidad para la creación de interfaces de usuarios de las aplicaciones, dándole una peculiaridad sobre el resto de las herramientas de este tipo (Tsang 2005).

Rational Rose

Rational Rose 2000 provee el modelado basado en UML (*Unified Modeling Language*) para el diseño de aplicaciones basadas en componentes. El UML, del cual fue pionero Rational y fue adoptado oficialmente como una norma por el OMG (*Object Management Group*), es el lenguaje estándar de la industria para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Rational Rose 2000 puede hacer ingeniería en reversa de componentes COM, ActiveX y JavaBeans,

Capítulo 1: Fundamento Teórico

para derivar las interfaces y determinar las interrelaciones de todos los componentes dentro de un modelo. Rose 2000 en su versión

Enterprise provee la capacidad para mezclar y casar múltiples idiomas, como C++, Visual Básico y Java, dentro del mismo modelo (Quatrani 2002).

A pesar de ser una buena herramienta presenta inconvenientes, necesita de mucha memoria para poder ser manejado de forma rápida y eficiente. Si no se dispone de un buen rendimiento, presenta dificultades a la hora de editar los diagramas y trabajar con ellos y tiene licencia privativa.

Fundamentación de la Herramienta CASE seleccionada.

La utilización de la herramienta CASE Rational Rose Enterprise sería una solución viable por su gran cantidad de funcionalidades y su fácil uso para cualquier usuario, pero presenta desventajas importantes con respecto a Visual Paradigm:

- ✓ Es un producto privativo por lo que tiene algunos problemas con la licencia.
- ✓ No tiene versiones para Linux.

Por lo que para el desarrollo de la solución que se propone se seleccionó el Visual Paradigm ya que presenta grandes prestaciones para el modelado de sistemas. Apoyándose también en el alto número de ventajas proporcionadas por el mismo entre las cuales se encuentran su fácil uso, es amigable, genera código y posee editor de figuras. Es una herramienta multiplataforma, característica que inclina la balanza a su favor al compararlo con el Rational Rose.

Sistemas gestores de base de datos (SGBD)

Un Sistema de Gestión de Base de Datos (SGBD) es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico. actualmente existe una amplia gama de SGBD con características propias, no obstante, todos deben tener en cuenta los siguientes aspectos de manera general: abstracción de la información, la independencia de los datos, redundancia mínima, consistencia en los datos, seguridad, integridad, respaldo y recuperación, tiempo de respuesta y control de concurrencia (Marcos 1999).

Oracle

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales. Para desarrollar en Oracle se utiliza PL/SQL un lenguaje de 5ta generación, bastante potente para tratar y gestionar la base de datos, por norma general se suele utilizar SQL al crear un formulario.

Capítulo 1: Fundamento Teórico

Oracle es considerado como uno de los sistemas de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma. Oracle puede ejecutarse en todas las plataformas, soporta tanto funciones como procedimientos almacenados con una integridad referencial declarativa bastante potente (Legizamon 2014).

Esta herramienta ha sido diseñada para controlar y gestionar grandes volúmenes de datos en un único repositorio. Es posible lógicamente atacar a la base de datos a través del SQL plus incorporado en el paquete de programas Oracle para poder realizar consultas, utilizando el lenguaje SQL. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

PostgreSQL 9.2.4.1

Es un Sistema Gestor de Base de Datos que comenzó como un proyecto en la Universidad Berkeley de California. Este proyecto sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. Está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. Puede funcionar en múltiples plataformas y a partir de la versión 8.0 también en Windows de forma nativa. Para las versiones anteriores existen versiones binarias para este sistema operativo, pero no tienen respaldo oficial (Carrancá 2014).

Sus ventajas más notables son:

- ✓ Aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas rutinas y reglas.
- ✓ Soporta operadores definidos por el usuario, funciones, métodos de acceso y tipo de datos.
- ✓ Soporta el núcleo SQL99 e incluye características avanzadas tales como las uniones (joins) de SQL92.
- ✓ Soporta la integridad referencial, que se utiliza para asegurar la validez de los datos de base de datos.
- ✓ La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS15 PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++ y Pike.
- ✓ Usa una arquitectura proceso-por-usuario, cliente/servidor PostgreSQL.

MySQL

MySQL o motor MySQL es un gestor de base de datos sencillo de usar y muy rápido, constituye uno de los motores de base de datos más usados en Internet, es un programa interactivo que permite conectarse a un servidor MySQL, ejecutar algunas consultas y ver los resultados, puede ser usado

Capítulo 1: Fundamento Teórico

también en modo Batch: es decir, se pueden colocar toda una serie de consultas en un archivo y posteriormente decirle a MySQL que las ejecute (Sánchez 2004).

Puede considerarse un software gratis cuando su uso es para aplicaciones no comerciales ya que muchas de las herramientas que existen para gestionar sus bases de datos son pagadas, como por ejemplo SQLYog, SQL Studio for MySQL o SQL-Front. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo, así como para integrarse en software para ser distribuido; soporta varios lenguajes de programación como C, C++, java, Perl, PHP, Python.

Fundamentación del SGBD seleccionado

Luego de haber hecho un análisis previo de los tres Sistemas Gestores de Bases de Datos (MYSQL, Oracle y PostgreSQL), se llegó a la conclusión de que el más idóneo para utilizar en este proyecto sería el PostgreSQL. Debido a que soporta una mayor cantidad de peticiones simultáneas de manera correcta, implementa el uso de subconsultas y transacciones, lo que proporciona una mayor eficacia a su funcionamiento, así como la capacidad de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel. Una de las desventajas que presenta el Oracle es que es un software propietario y su licencia es excesivamente cara. Mientras que el PostgreSQL está disponible sin costo alguno.

Lenguajes de programación seleccionados

Un lenguaje de programación es un idioma artificial, compuesto por un conjunto de símbolos, reglas semánticas y sintácticas que permiten la comunicación entre una persona y el ordenador (Rodríguez 2013).

Pueden clasificarse en dos grupos principales:

- ✓ Lenguajes de bajo nivel: Se acercan al funcionamiento del computador. El lenguaje representativo es el código máquina. Las instrucciones en este lenguaje están formadas por cadenas binarias (0 y 1). Puede citarse el lenguaje ensamblador el cual trabaja directamente con los registros de la computadora.
- ✓ Lenguajes de alto nivel: Son fáciles de aprender porque están formados por elementos del lenguajes y al usarlos puede dar la sensación de que las computadoras los comprenden.

Programación Web

PHP

PHP es un robusto lenguaje de programación del lado del servidor. Fue diseñado por Rasmus Lerdorf en 1994 como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar

Capítulo 1: Fundamento Teórico

formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje (Vaswani 2010).

Dado que es un lenguaje abierto dando la posibilidad de modificar el código fuente y añadir nuevas funcionalidades ha tenido una rápida evolución y desarrollo. Actualmente se encuentra en su versión PHP5. Una de sus grandes potencialidades es su soporte para una gran cantidad de bases de datos. De las cuales se pueden mencionar InterBase, MySQL, Oracle y PostgreSQL. PHP también ofrece la integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en pdf (*Portable Document Format*) hasta analizar código XML.

Estas son algunas de las ventajas que presenta este lenguaje:

- ✓ Es un lenguaje multiplataforma.
- ✓ Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos.

Las desventajas más comunes que posee este lenguaje son:

- ✓ **Poca modularización:** Este lenguaje debe hacer uso de Framework para poder separar por capas la aplicación.
- ✓ **Demora en el mantenimiento:** Al poseer este lenguaje un pobre nivel de estructuración se hace más complejo y lento el mantenimiento del sistema.

Java

Java es una tecnología orientada al desarrollo de software con la cual se puede realizar cualquier tipo de programa, nace a inicio de los años noventa, creado por la empresa Sun Microsystems como un lenguaje ideado en sus comienzos para programar electrodomésticos, en sus primeras versiones, se llamó OAK, fue diseñado como un lenguaje orientado a objetos desde el principio, para crear software altamente fiable.

Con el uso de Java se facilita la creación de programas modulares y códigos reutilizables. Este lenguaje se ha convertido en una de las elecciones para ofrecer soluciones en todo el mundo. Es compilado, en la medida en que su código fuente se transforma en una especie de código máquina (*bytecodes*), semejantes a las instrucciones de ensamblador; Por otra parte, es interpretado, ya que los

Capítulo 1: Fundamento Teórico

bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (*run-time*) (Martínez 2008).

JEE6

Java Platform, Enterprise Edition (Java EE) 6 es el estándar de la industria para la informática empresarial de Java. Permite utilizar el nuevo y ligero perfil web Java EE 6 para crear aplicaciones Web de próxima generación y todo el poder de la plataforma para aplicaciones empresariales. Los desarrolladores se beneficiarán de las mejoras de la productividad con más anotaciones, más POJO's, la simplificación de empaquetado y menos configuración de XML (Yerovi 2013).

Las mejoras que incluye JEE 6 son, la inclusión de perfiles, los cuales permiten utilizar ciertos fragmentos de toda la especificación (Perfil Completo y Perfil Web). Actualización de JAX-RS y JAX-WS, JPA 2.0 con nuevas librerías, *Servlets* 3.0 los cuales incluyen varios POJO's, EJB (Enterprise Java Beans) 3.1; entre los más importantes.

Arquitectura aplicaciones JEE

El modelo JEE comienza con el lenguaje de programación Java y la JVM y soporta aplicaciones que implementan servicios empresariales. Los requerimientos de los clientes son tratados en la capa media del modelo, la cual corre en un hardware dedicado, tiene acceso a los recursos empresariales y divide el trabajo en dos partes: la lógica del negocio y la presentación.

Fundamentos del lenguaje web utilizado

Como lenguaje utilizado para el desarrollo de la aplicación Web se elige a la plataforma JEE6, ya que posee herramientas suficientes y potentes como son los *Servlets*. Presenta nuevas características para componentes *Enterprise JavaBeans*, *Servlets* y componentes *JavaServer Faces*, usadas para crear una aplicación bien estructurada, con la calidad requerida y lista para su futuro mantenimiento. Esta plataforma al utilizar el lenguaje Java está orientada a objeto, es libre y posee una abundante documentación.

Otras tecnologías y lenguajes usados

HTML5

HTML5 (*Hyper Text Markup Language*) Lenguaje de Marcas de Hipertexto, es uno de los lenguajes de marcado más utilizado para la construcción de páginas web. Es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la síntesis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto imágenes y otros medios soportados.

Capítulo 1: Fundamento Teórico

Su versión 5 brinda más facilidades que sus antecesores, lo que permite el desarrollo de la aplicación con una mayor vistosidad, calidad e integración, este es soportado por las versiones actuales de los navegadores Mozilla Firefox, Chrome, Chromiun, Safari e Internet Explorer (Sharp 2011).

CCS 3 (Cascading Style Sheets)

Este lenguaje se usa para organizar y controlar la presentación y aspecto de una página Web. Es principalmente utilizado por los diseñadores y programadores de aplicaciones para elegir la multitud de opciones de presentación que este brinda, como son los colores, tipos y tamaños de letra. Se seleccionó la versión 3 de este lenguaje por que trae muchas características nuevas que enriquecen la apariencia de las páginas web, entre los que se encuentran: los bordes redondeados, textos sombreados, sombreados de cajas de texto, múltiples fondos, rotación de textos, animaciones y transiciones.

“Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes” (Rodríguez 2013).

XML

El "*Extensible Markup Language*" (XML) se ha convertido en el formato estándar de facto para la representación de datos en Internet. Los datos XML se pueden procesar e interpretar en cualquier plataforma, desde un dispositivo de mano a una unidad central. Puede estar estructurado, ser usados como contenedores de datos, es también sencillo de utilizar, posee un formato estándar y es muy flexible (Sniper 2003).

IDE (Integrated Development Environment)

Un Entorno de Desarrollo Integrado es un conjunto de herramientas de desarrollo de software para programadores. Generalmente están compuestos por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de usuario. Son creados para el desarrollo de aplicaciones en un sólo lenguaje de programación, sin embargo, hay algunos en los que se puede desarrollar en más de un lenguaje. Cuando el lenguaje es Orientado a Objetos, incluyen navegador de clases, inspector de objetos y diagrama de jerarquía de clases (Carrancá 2014).

Netbeans 8.0

Es un entorno Integrado de Desarrollo gratuito, de código abierto para desarrolladores de software. Contiene herramientas para crear aplicaciones profesionales para el escritorio, la empresa, la web y equipos móviles con el lenguaje Java, C/C++ y Ruby. Es fácil de instalar y de usar y se ejecuta en varias plataformas incluyendo Windows, Linux y Mac OS X y Solaris. La versión de Netbeans IDE 8.0,

Capítulo 1: Fundamento Teórico

proporciona mejoras en el rendimiento, menor consumo de memoria y mejor respuesta cuando se trabajan con proyectos grandes. Algunas de las tecnologías y lenguajes soportados por esta versión son: Ajax, C/C++, Databases, Debugger, Desktop, Editor, Groovy, GUI Builder, Java EE, Java FX, Java ME, Java SE, JavaScript, Mobile, PHP, Profiler, Python, Refactor, REST, Rich Client Platform, Ruby, SOA, SOAP, UML, Web, WSDL, XML, Hibernate, Spring y Struts.

Visual Studio.net

Es un entorno integrado de programación para sistemas Windows, que soporta los lenguajes de programación Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic.NET. Los programas desarrollados en .NET no se compilan en lenguaje máquina como C++, sino que se compilan en un lenguaje intermedio llamado *Microsoft Intermediate Language (MSIL)* como se conoce por sus siglas en inglés.

Este lenguaje incluye tipos genéricos, similares en muchos aspectos a las plantillas de C++. Con esto se consigue encontrar muchos más errores en la compilación en vez de en tiempo de ejecución, incitando a usar comprobaciones estrictas en áreas donde antes no era posible. Incluye un diseñador de implantación, que permite que el diseño de la aplicación sea validado antes de su implantación. También se incluye un entorno para publicación web y Tests de carga para comprobar el rendimiento de los programas bajo varias condiciones de carga.

Fundamentos del IDE seleccionado

Se selecciona como IDE de desarrollo el Netbeans en su versión 8.0 ya que:

- ✓ Es un entorno Integrado de Desarrollo gratuito.
- ✓ Contiene herramientas para crear aplicaciones profesionales Web y Desktop.
- ✓ Permite la edición, depuración y compilación del código Java.
- ✓ Permite el rápido desarrollo de un interfaz visual haciendo usos de componentes.
- ✓ Soporta Java2, HTML5, CCS3 y JSP2.1.
- ✓ Se ejecuta en varias plataformas.
- ✓ Es fácil de usar y de instalar.

Librerías o Framework usados

El término Framework es muy utilizado en el campo de la informática, no es más que un esquema para el desarrollo y/o la implementación de una aplicación. Proporciona una estructura al código fuente, forzando al desarrollador a crear código legible y más fácil de mantener, simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes y facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas (Castillo 2011).

Capítulo 1: Fundamento Teórico

JSF 2.0

JSF es un *framework* para aplicaciones Web en Java de Sun Microsystems. Está orientado a la interfaz gráfica de usuario (GUI), facilitando el desarrollo de estas y que, sin embargo, realiza una separación entre comportamiento y presentación, además de proporcionar su propio *servlet* como controlador, brinda un modelo basado en componentes y dirigido por eventos para el desarrollo de aplicaciones Web, que es similar al modelo usado en aplicaciones GUI *standalone* (Bautista 2010).

JSF incluye las siguientes características principales:

- ✓ Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entradas, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- ✓ Un conjunto de componentes por defecto para la interfaz de usuario.
- ✓ Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- ✓ Un modelo de eventos en el lado del servidor.
- ✓ Administración de estados.
- ✓ Beans administrados.
- ✓ JSF es una especificación desarrollada por la Java Community Process.

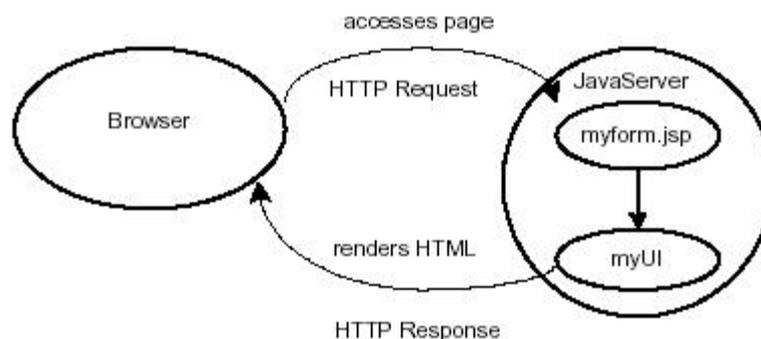


Figura No 1.1 Flujo de una petición Request en JSF.

JPA: La API de persistencia desarrollada para la plataforma Java EE es una colección de clases y métodos almacenados de forma persistente con un sin número de información contenida dentro de una base de datos la cual es proporcionada por la corporación Oracle. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional) y permitir usar objetos regulares (conocidos como POJOs) (Point 2014).

Capítulo 1: Fundamento Teórico

Los 4 tipos de interfaces de las que se compone JPA son:

- ✓ **javax.persistence.Persistence**: Contiene métodos estáticos de ayuda para obtener una instancia de Entity Manager Factory de una forma independiente al vendedor de la implementación de JPA. Una clase de inicialización que va proporcionar un método estático para la creación de una Entity Manager Factory.
- ✓ **javax.persistence.Entity.Manager.Factory**: La clase `javax.persistence.Entity.Manager.Factory` ayuda a crear objetos de EntityManager utilizando el patrón de diseño del Factory. Este objeto en tiempo de ejecución representa una unidad de persistencia particular. Generalmente va a ser manejado como un Singleton y proporciona métodos para la creación de instancias EntityManager.
- ✓ **javax.persistence.Entity.Manager.Factory**: La clase `javax.persistence.Entity` es una anotación Java que se coloca a nivel de clases Java serializables y que cada objeto de una de estas clases anotadas representa un registro de una base de datos.
- ✓ **javax.persistence.EntityManager**: Es la interfaz principal de JPA utilizada para la persistencia de las aplicaciones. Cada Entity Manager puede realizar operaciones CRUD (*Create, Read, Update, Delete*) sobre un conjunto de objetos persistentes. Es un objeto único, no compartido que representa una unidad de trabajo particular para el acceso a datos. Proporciona métodos para gestionar el ciclo de vida de las instancias entidad y para crear instancias Query.
- ✓ **javax.persistence.Query**: La interface `javax.persistence.Query` está implementada por cada vendedor de JPA para encontrar objetos persistentes manejando cierto criterio de búsqueda. JPA estandariza el soporte para consultas utilizando Java Persistence Query Language (*JPQL*) y Structured Query Language (*SQL*). Se puede obtener una instancia de *Query* desde una instancia de un Entity Manager.
- ✓ **javax.persistence.EntityTransaction**: Cada instancia de Entity Manager tiene una relación de uno a uno con una instancia de `javax.persistence.EntityTransaction`, permite operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.

PrimeFaces 5.3

PrimeFaces es una librería open Source (Licencia Apache V2) para Java Server Faces, el objetivo principal de esta es ofrecer un conjunto de componentes ricos para facilitar la creación de aplicaciones web. PrimeFaces se divide principalmente en tres módulos: El primero es el set de componentes para las interfaces de usuario, el segundo módulo llamado Optimus utiliza Guice para poder crear managed Beans utilizando anotaciones, simplificar la navegación entre páginas e integrar PrimeFaces con JPA,

Capítulo 1: Fundamento Teórico

transacciones y más. Y un tercer módulo llamado FacesTrace permite monitorear aplicaciones JSF. Lo bueno es que los tres módulos son totalmente independientes lo que permite crear cualquier tipo de combinación de *frameworks* (Çivici 2014).

Ventajas que posee PrimeFaces con respecto a otras bibliotecas de componentes JSF

- ✓ Un amplio conjunto de componentes de Interfaz de usuario entre los que se encuentran (DataTable, AutoComplete y HtmlEditor).
- ✓ No se requiere configuraciones extras, ni son necesarias dependencias.
- ✓ Ajax incorporado.
- ✓ Conjunto de más de 25 temas integrados.
- ✓ Excelente documentación con ejemplos de código y videos tutoriales.

Bootstrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “*responsive design*” o diseño adaptativo.

El beneficio de usar *responsive design* en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda. Es un módulo de CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla.

JQuery 2.0.3

Es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. Es la biblioteca de JavaScript más utilizada, es un software libre y de código abierto que posee un doble licenciamiento bajo la Licencia MIT y la Licencia Publica General GPL v2, permitiendo su uso en proyectos libres y privativos. Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio (Carrancá 2014).

IText

Es una biblioteca *Open Source* para crear y manipular archivos PDF, RTF y HTML en Java. Fue escrita por Bruno Lowagie y Paulo Soares, está distribuida bajo la *Affero General Public License*.

Capítulo 1: Fundamento Teórico

Jsingpdf

Jsingpdf es un software gratuito que te permite firmar digitalmente documentos PDF, de gran utilidad para mostrar la autenticidad de un documento que se envió o recibió.

SOAP

Es un paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más complejos y completos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una serie de protocolos de servicios web ofreciendo un *framework* de mensajería básica en el cual los servicios web se pueden construir. Este protocolo está basado en XML y se conforma de tres partes:

- ✓ Sobre el cual define qué hay en el mensaje y cómo procesarlo.
- ✓ Conjunto de reglas de codificación para expresar instancias de tipos de datos.
- ✓ La Convención para representar llamadas a procedimientos y respuestas.

El protocolo SOAP tiene tres características principales:

- ✓ **Extensibilidad:** Seguridad y WS-routing son extensiones aplicadas en el desarrollo.
- ✓ **Neutralidad:** Puede ser utilizado sobre cualquier protocolo de transporte como HTTP, SMTP, TCP o JMS.
- ✓ **Independencia:** Permite cualquier modelo de programación.

Algunas de las Ventajas de SOAP son:

- ✓ **No está asociado con ningún lenguaje:** Los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista, pero los desarrolladores responsables de mantener antiguas aflicciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación, como en Java y la plataforma como Microsoft .Net.
- ✓ **No se encuentra fuertemente asociado a ningún protocolo de transporte:** La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- ✓ **No está atado a ninguna infraestructura de objeto distribuido:** La mayoría de los sistemas de objetos distribuidos se pueden extender y ya lo están alguno de ellos para que admitan SOAP.

Capítulo 1: Fundamento Teórico

- ✓ **Aprovecha los estándares existentes en la industria:** Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y como ya se ha mencionado SOAP no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.
- ✓ **Permite la interoperabilidad entre múltiples entornos:** SOAP se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dichos estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas. Por ejemplo, una aplicación de escritorio que se ejecute en una PC puede comunicarse con una aplicación del *back-end* ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP.

Servidor de Aplicaciones

GLASSFISH

GlassFish es la implementación de referencia de Java EE y como tal soporta Enterprise JavaBeans, JPA, *JavaServer Faces*, JMS, RMI, *Java Server Pages*, *servlets*, además de otros. Esto permite a los desarrolladores crear aplicaciones empresariales que son portables y escalables, además de su integración con las más anticuadas tecnologías de este campo. GlassFish permite que componentes opcionales puedan ser instalados para servicios adicionales. Construido sobre un kernel modular y alimentado por OSGi, GlassFish se ejecuta en la parte superior de la implementación de Apache Felix. Este también se ejecuta con los tiempos de ejecución Equinox OSGi o Knopflerfish OSGi. HK2 abstrae el sistema de módulos de OSGi para aportar componentes, que también se pueden ver como los servicios. Estos servicios pueden ser descubiertos y se inyectan en tiempo de ejecución.

Máquina virtual JVM

La máquina virtual de Java se denomina al procesador o entorno virtual que se utiliza para interpretar los *bytecodes* de los binarios de Java, ya que este se desarrolló para correr en cualquier plataforma sin recompilar los binarios. Esta máquina virtual ha tenido la característica de ser un entorno de ejecución pesado en términos de recursos del procesador y memoria, que, por medio de una administración rigurosa del sistema operativo, estos podrían llegar a ser insuficientes y las aplicaciones ejecutarse de manera muy lenta. Esto no es cierto en la actualidad, ya que existen alternativas a la JVM provista por Sun Microsystems que permiten una velocidad comparable a una aplicación compilada en C++ nativa en la arquitectura (Perez 2008).

Capítulo 1: Fundamento Teórico

JDK8

El Kit de desarrollo conocido como **JDK** (*Java Development Kit*) provee de un compilador, un mecanismo para comprimir un proyecto en un solo archivo de tipo JAR (que es compatible con ZIP) y un entorno de ejecución para los binarios.

JRE8

El **JRE** (*Java Runtime Environment*) es el entorno de ejecución para las aplicaciones desarrolladas en JAVA.

Conclusiones parciales

Durante el desarrollo de este capítulo se tuvieron en cuenta conceptos de gran importancia para la evolución de la investigación:

- ✓ Se realizó una valoración de las soluciones existentes tanto a nivel mundial como nacional, las cuales poseen algunas desventajas y no se adaptan al modelo de gestión que posee la industria farmacéutica, decidiéndose desarrollar un sistema informático que permita gestionar la información de los expedientes técnicos de los medios informáticos en los laboratorios farmacéuticos AICA.
- ✓ Se decidió desarrollar una aplicación informática que permita la gestión de información de los expedientes técnicos de medios informáticos y la utilización de la firma digital de documentos.
- ✓ Se utilizó Java por ser un lenguaje multiplataforma y el *framework* JSF debido a que presenta un conjunto de características que permiten manipular de una manera fácil y simple la interfaz visual del sistema. Se seleccionó OpenUp como metodología de desarrollo la cual permitió debido a su gran adaptabilidad generar los artefactos necesarios para guiar al desarrollo de la aplicación informática.
- ✓ Se realizó un estudio sobre la implementación de la firma digital, la cual se va integrar al sistema informático para brindarle una mayor seguridad e integridad a los documentos que sean generados por los usuarios de la aplicación.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

Para presentar la solución que se propone y que se comprenda correctamente lo que se desea transmitir al lector, es necesario conocer las características del negocio. Para lograr un mejor entendimiento del negocio se presenta un modelo conceptual, así como los conceptos enmarcados en el entorno del problema describiéndose detalladamente cada una de las entidades identificadas. Se hará una especificación de los requerimientos funcionales y no funcionales que responden a la necesidad del sistema, la especificación de cada uno de los casos de usos que se identificaron, la descripción de los actores y los patrones de casos de uso.

2.1 Descripción del Sistema Propuesto

Una vez realizado el estudio basado en la problemática existente se propone desarrollar un sistema que logre informatizar el proceso actual que se lleva a cabo en la gestión de los expedientes técnicos, logrando así un mejor funcionamiento del grupo de especialistas. Este permitirá realizar búsquedas avanzadas, gestionar y almacenar toda la información necesaria en una base de datos. El sistema debe generar documentos PDF (Incidencias y Expedientes técnicos), que serán firmados digitalmente y cargar un archivo con extensión txt que contenga las propiedades del equipamiento informático con un peso a lo máximo de 50 byte. Los archivos generados por la aplicación deben ser almacenados en el servidor con el fin de tener evidencia documental de este proceso. Es importante mencionar que Cuba actualmente se encuentra en el proceso de certificación de la firma digital, aprovechando así la utilización de esta tecnología, la cual será una novedad en la industria farmacéutica.

2.2 Modelo Conceptual

Un Modelo Conceptual es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. En el siguiente modelo se evidencia cómo funciona el entorno en el cual está enmarcada la aplicación y representa los conceptos fundamentales del mundo.

Capítulo 2: Características del Sistema

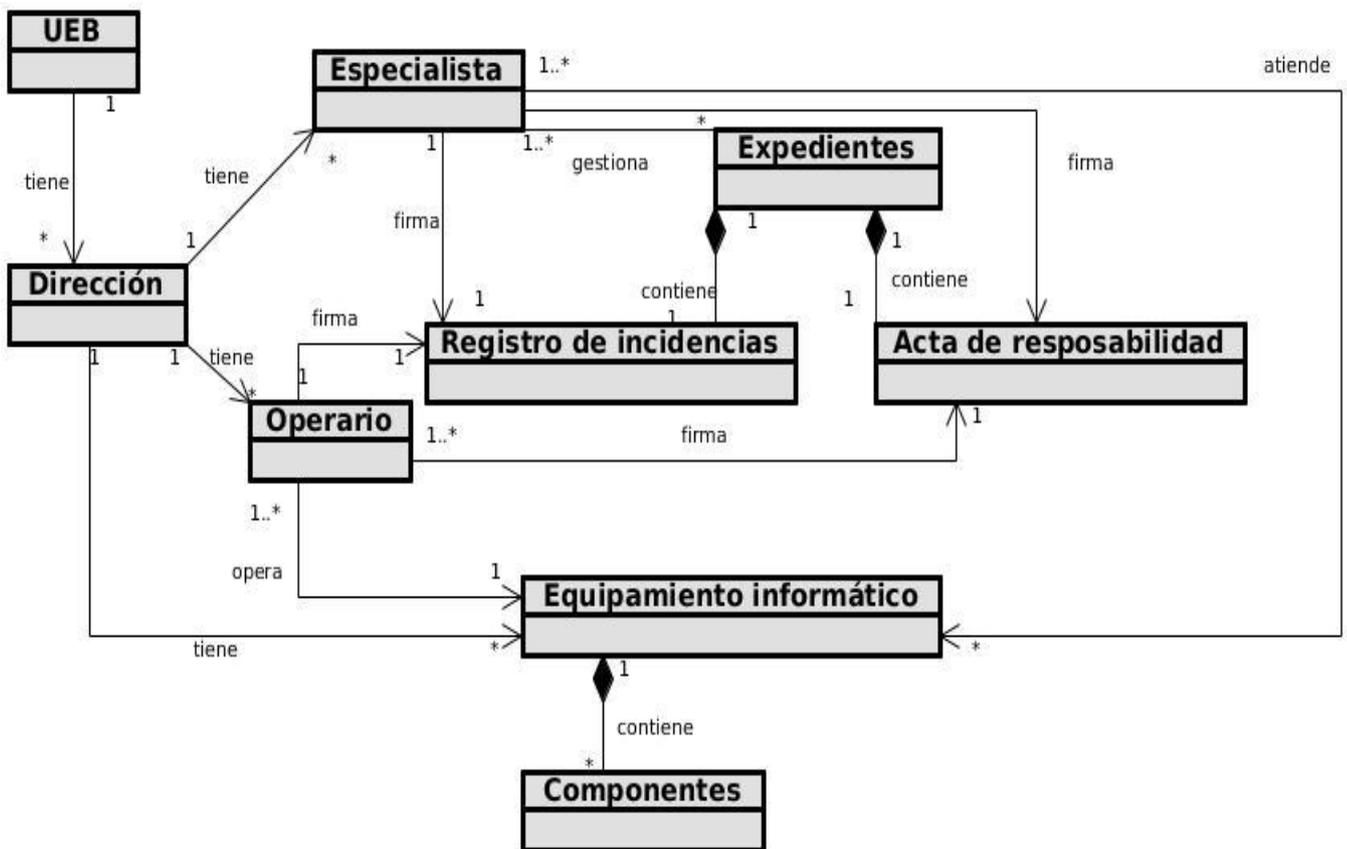


Figura No.2.1 Diagrama de Modelo Conceptual.

2.3 Descripción de las entidades

En la siguiente tabla se muestra la descripción de cada entidad usada en el diagrama de modelo de dominio.

Tabla No.2.1 Descripción de las entidades del dominio.

Entidades	Descripción
UEB	Entidad que representa los laboratorios que componen AICA.
Dirección	Entidad que representa las áreas que componen cada UEB.
Especialista	Entidad que representa el responsable de brindar el servicio de soporte y mantenimiento a los equipos informáticos.
Operario	Entidad que representa al usuario que trabaja con el equipo informático.
Expedientes	Entidad que representa el documento que contiene las características de cada equipo informático.
Registro de Incidencia	Es la entidad que representa los cambios que inciden sobre el equipamiento informático.
Acta de Responsabilidad	La entidad representa el documento que contiene la responsabilidad material.
Equipamiento Informático	Entidad que representa el equipo informático (Laptop, PC, Equipo tecnológico).
Componentes	Entidad que representa los elementos de hardware del equipamiento informático.

2.4 Especificación de los requisitos del sistema

Los requisitos no son más que la descripción de las necesidades o deseos de un producto, ya que determinan lo que hará el sistema y definen las restricciones de su operación e implementación.

Requisitos funcionales

Los requisitos funcionales establecen las capacidades y condiciones que el sistema debe cumplir, por lo que este sistema debe:

✓ RF1. Autenticar.

Gestionar usuarios.

✓ RF2. Insertar usuario.

✓ RF3. Eliminar usuario.

✓ RF4. Modificar usuario.

✓ RF5. Listar usuario.

✓ RF6. Buscar usuario.

Gestionar expediente.

✓ RF7. Insertar expediente.

✓ RF8. Eliminar expediente.

✓ RF9. Modificar expediente.

✓ RF10. Listar expediente.

✓ RF11. Buscar expediente.

Gestionar componentes del expediente.

✓ RF12. Insertar componente.

✓ RF13. Eliminar componente.

✓ RF14. Editar componente.

✓ RF15. Listar componente.

✓ RF16. Buscar componente.

Gestionar incidencias.

✓ RF17. Insertar incidencia.

✓ RF18. Editar incidencia

✓ RF19. Listar incidencia.

✓ RF20. Buscar incidencia.

✓ RF21. Mostrar incidencias.

✓ RF22. Crear archivo de incidencia.

✓ RF23. Firmar incidencia.

✓ RF24. Crear archivo de expediente.

✓ RF25. Firmar expediente.

Capítulo 2: Características del Sistema

- ✓ RF26. Firmar incidencia operario.
- ✓ RF27. Firmar expediente operario.
- ✓ RF28. Mostrar expedientes.
- ✓ RF29. Subir archivo de características del sistema.

Requisitos no funcionales

Los requisitos no funcionales son las exigencias de cualidades que se le imponen al proyecto, estos requisitos tienen que ver con los requerimientos de hardware y software, usabilidad, la apariencia y los requerimientos de implementación.

Requerimiento de usabilidad

- ✓ **RNF.1** El sistema informático está destinado a la gestión de expedientes técnicos de los medios informáticos de los laboratorios farmacéuticos AICA, el cual va a constar de un filtro de búsqueda por campos que le permite al usuario que el proceso de búsqueda de la información se efectúe de manera eficiente y un sistema de notificación que muestra al usuario el resultado final de cada funcionalidad.

Requerimiento mínimo de software

Aplicación Web cliente (AWC)

- ✓ **RNF.2** La AWC puede correr en cualquier sistema operativo.
- ✓ **RNF.3** Para que la AWC funcione correctamente debe estar instalado el navegador web Mozilla Firefox 42.0 o superior o el Google Chrome 36.0 o superior.

Aplicación Web Servidor (Sharp 2011)

- ✓ **RNF.4** La AWS debe ser instalada en el sistema operativo Linux.
- ✓ **RNF.5** Para poder desplegar la AWS tiene que estar instalado el servidor de aplicaciones GlassFish 4.0.
- ✓ **RNF.6** Para que la AWS funcione debe estar instalado el entorno de ejecución de aplicaciones Java JRE8.
- ✓ **RNF.7** Para poder desplegar la AWS tiene que tener instalado el SGBD PostgreSQL9.2.4.1.
- ✓ **RNF.8** Para poder desplegar la AWS debe tener la base de datos "Expedientes_tecnicos" cargada en el SGBD.
- ✓ **RNF.9** Para poder ejecutar desde otros lugares la AWS deben estar configurado el servidor de aplicaciones y el SGBD para permitir conexiones externas.
- ✓ **RNF.10** Para el correcto funcionamiento de la aplicación debe estar creada la estructura de archivo como se especifica en el diagrama de componentes.

Capítulo 2: Características del Sistema

Requerimiento mínimo de hardware

Aplicación Web Cliente (AWC).

- ✓ **RNF.11** Para ejecutar la AWC el ordenador debe poseer un puerto de red de 10Mbit.
- ✓ **RNF.12** Para ejecutar la AWC el ordenador debe tener instalado 512MByte de RAM.
- ✓ **RNF.13** Para ejecutar la AWC el ordenador debe tener instalado como mínimo un microprocesador Intel Celeron a 2.7.

Aplicación Web Servidor (Sharp 2011)

- ✓ **RNF.14** Para desplegar la AWC el ordenador necesita tener un puerto de red de 10Mbit.
- ✓ **RNF.15** Para desplegar la AWC el ordenador necesita tener instalado como mínimo 4GByte de RAM.
- ✓ **RNF.16** Para desplegar la AWC el ordenador necesita tener instalado como mínimo un disco duro de 160GByte.
- ✓ **RNF.17** Para desplegar la AWC el ordenador necesita tener instalado como mínimo un microprocesador Intel Dual Core 2.3 Hz.

Requerimiento de interfaz

- ✓ **RNF.18** El sistema gestor de expedientes técnicos posee una interfaz agradable a la vista del usuario, cuenta con un diseño intuitivo, presenta los colores de la entidad y brinda una gran facilidad de comprensión e integración.

Cuestiones legales

- ✓ **RNF.19** La aplicación web se rige por las normas y leyes establecidas por el país y la entidad, cumpliendo además con las regulaciones que brinda CFR Part 11 dictadas por la Agencia de Alimentos y Medicamentos (*FDA*) y es desarrollada utilizando herramientas libres bajo la licencia GNU/GPL.

Requerimiento de seguridad

- ✓ **RNF.20** Confidencialidad: Fueron definidas un grupo de políticas de seguridad que prevén el uso inadecuado del sistema. La aplicación posee un sistema de autenticación donde existen los roles de especialista y operario, automáticamente todo usuario que se autentique contra el LDAP se le asigna el rol de operario. Solo el especialista puede cambiar el rol de un usuario.
- ✓ **RNF.21** Integridad: La contraseña de los usuarios del dominio es manipulada con minucioso cuidado por el alto grado de confiabilidad que representa la misma dentro de los laboratorios

Capítulo 2: Características del Sistema

AICA, utilizando para este fin el servidor LDAP que permite autenticar el usuario contra el dominio, evitando que la contraseña sea almacenada en la BD de la aplicación web.

Portabilidad

- ✓ **RNF.22** Como el sistema gestor de expedientes técnicos es una aplicación web, se puede ejecutar desde cualquier dispositivo conectado a la red de la entidad y es multiplataforma.

Rendimiento

- ✓ **RNF.23.** La funcionalidad Crear Incidencia debe brindar un tiempo de respuesta a lo máximo de 2.5s para un total de 1126 peticiones con 50 hilos.
- ✓ **RNF.24:** La funcionalidad para crear cualquier componente debe brindar un tiempo de respuesta a lo máximo de 1.3s para un total de 1089 peticiones con 50 hilos.
- ✓ **RNF.25:** La funcionalidad para borrar cualquier componente debe brindar un tiempo de respuesta a lo máximo de 1s para un total de 1051 peticiones con 50 hilos.
- ✓ **RNF.26:** La funcionalidad Mostrar Expediente debe brindar un tiempo medio de respuesta a lo máximo de 1.6s para un total de 1200 peticiones con 50 hilos.
- ✓ **RNF.27:** La funcionalidad Listar Incidencias debe brindar un tiempo medio de respuesta a lo máximo de 0.609s para un total de 1010 peticiones con 50 hilos.

Capítulo 2: Características del Sistema

2.5 Diagrama de Casos de Uso

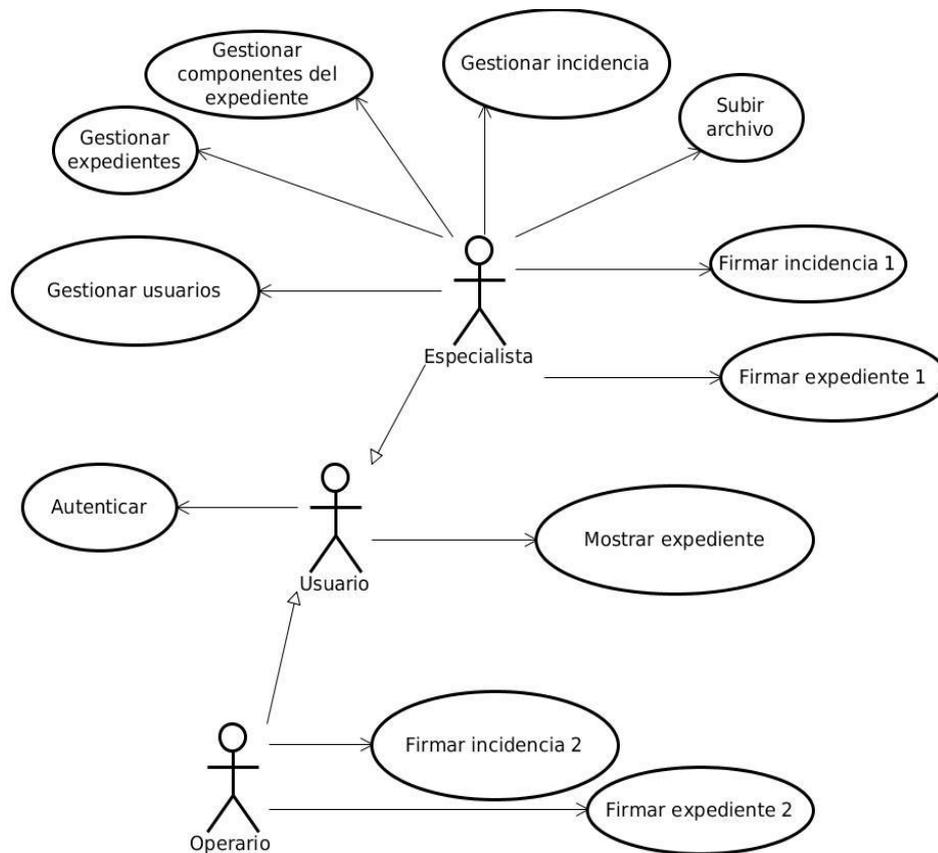


Fig. No.2.2 Diagrama de Casos de Uso de la aplicación web.

Actores y Casos de Uso del sistema

Tabla No.2.2 Actores que interactúan con la aplicación web

Actores	Descripción
Usuario	Este actor es el que se autentica en el sistema y es común para operario y especialista.
Operario	Este actor puede observar, firmar el archivo de expediente técnico y el archivo de incidencia creado por el especialista.
Especialista	Este actor puede gestionar los expedientes técnicos, los usuarios y las incidencias, además de firmar los documentos que él pueda generar.

Tabla No.2.3 Casos de uso que interactúan con los actores

No	Caso de Uso	Prioridad	Descripción
1	Mostrar expediente	Alta	Se encarga de mostrar las características que posee el equipamiento informático.
2	Autenticar	Alta	Se encarga de darle seguridad a la aplicación y de brindarle permisos al usuario en dependencia de su rol.

Capítulo 2: Características del Sistema

3	Gestionar incidencia	Alta	Permite al especialista insertar, listar y buscar incidencias de un equipamiento informático.
4	Firmar incidencia 1	Alta	Permite al especialista crear un archivo de incidencia y firmarlo digitalmente.
5	Firmar expediente 1	Alta	Permite al especialista crear un archivo de expediente técnico y firmarlo digitalmente.
6	Firmar incidencia 2	Alta	Permite al operario firmar un archivo de incidencia creado por el especialista.
7	Firmar expediente 2	Alta	Permite al operario firmar el archivo de expediente técnico creado por el especialista.
8	Subir archivo	Media	Funcionalidad que permite subir un archivo txt, con las propiedades del equipamiento informático con un peso a lo máximo de 50 byte.
9	Gestionar usuario	Media	Funcionalidad que permite insertar, listar, buscar, eliminar y modificar los usuarios que accederán a la aplicación, así como definir para cada uno los roles o permisos que tendrán dentro de la misma.
10	Gestionar expediente	Alta	Funcionalidad que permite insertar, modificar, eliminar, listar y buscar un expediente técnico.
11	Gestionar componentes del expediente	Alta	Funcionalidad que permite insertar, modificar, eliminar, listar y buscar el componente del expediente técnico.

2.6 Descripción textual de los casos de uso

Casos de uso	Gestionar incidencia
Actores	Especialista
Resumen	Es iniciado cuando el especialista realiza alguna de las operaciones: Insertar, Modificar, Mostrar, Listar, buscar, crear archivo de incidencia y firmar archivo de incidencia. Este caso de uso se encarga de gestionar las incidencias de un equipo.
Precondición	
Referencia	Caso de uso No 3 del sistema
Prioridad	Crítico
Sección "Insertar incidencia"	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El usuario selecciona el expediente que le va a crear la incidencia. 2. El usuario presiona el ítem incidencias del menú de la aplicación. 4. El usuario presiona el botón Crear 	<ol style="list-style-type: none"> 3. El sistema muestra el listado de incidencias y los botones crear, editar, mostrar y los campos de búsqueda. 5. El sistema muestra una caja de dialogo con el formulario para introducir una nueva

Capítulo 2: Características del Sistema

6. El usuario llena los datos del formulario y presiona el botón salvar.	incidencia. 7. El sistema valida los datos. 8. El sistema crea el archivo de incidencia. 9. El sistema firma el archivo de incidencia con la firma del especialista. 10. El sistema cierra la caja de dialogo y envía una notificación de que la inserción ha sido satisfactoria. 11. El sistema actualiza el listado de incidencia.
Flujos alternos Sección “Insertar incidencia”	
Acción de actor	Respuesta del sistema
	8.1 Si los campos están vacíos o los valores no son válidos el sistema muestra un mensaje de error.
Sección “Modificar incidencia”	
Acción de actor	Respuesta del sistema
1. El especialista presiona el ítem incidencias del menú de la aplicación. 3. El especialista selecciona la incidencia que se va a modificar. 4. El especialista presiona el botón editar 6. El usuario llena los datos del formulario y presiona el botón salvar .	2. El sistema muestra el listado de incidencias y los botones crear , editar y mostrar y los campos de búsqueda. 5. El sistema muestra una caja de dialogo con el formulario para modificar la incidencia. 7. El sistema valida los datos. 8. El sistema cierra la caja de dialogo y envía una notificación de que la modificación ha sido satisfactoria. 9. El sistema actualiza el listado de incidencia
Flujos alternos Sección “Modificar incidencia”	
Acción de actor	Respuesta del sistema
	8.2 Si los campos no son válidos o los valores no son válidos el sistema manda un mensaje de error.
Sección “Mostrar incidencia”	
Acción de actor	Respuesta del sistema
1. El especialista presiona el ítem incidencias del menú de la aplicación. 3. El especialista selecciona la incidencia que se va a mostrar. 4. El especialista presiona el botón mostrar	2. El sistema muestra el listado de incidencias y los botones crear , editar y mostrar y los campos de búsqueda. 5. El sistema muestra una caja de dialogo con la información de la incidencia.
Sección “Listar Incidencia”	
Acción de actor	Respuesta del sistema
1. El especialista presiona el ítem incidencias del	2. El sistema muestra el listado de incidencias y

Capítulo 2: Características del Sistema

menú de la aplicación.	los botones crear , editar y mostrar y los campos de búsqueda.
Flujos alternos Sección “Listar incidencia”	
Acción de actor	Respuesta del sistema
	2.1 Si no existen incidencias la lista se muestra vacía.
Sección “Buscar incidencia”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El especialista presiona el ítem incidencias del menú de la aplicación. 2. El especialista busca la incidencia creada por los campos de búsqueda que desee. 	<ol style="list-style-type: none"> 3. El sistema muestra el listado de incidencias y los botones crear, editar y mostrar y los campos de búsqueda. 4. El sistema va actualizando la lista según la búsqueda.

Casos de uso	Gestionar expedientes
Actores	Especialista
Resumen	Es iniciado cuando el especialista realiza alguna de las operaciones: Insertar, Modificar, Mostrar, Listar, buscar, crear archivo de expediente y firmar archivo de expediente. Este caso de uso se encarga gestionar los expedientes de los equipos.
Precondición	
Referencia	Caso de uso No 9 del sistema
Prioridad	Crítico
Sección “Insertar expedientes”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 2. El usuario presiona el botón crear. 4. El usuario llena los datos del formulario y presiona el botón salvar. 	<ol style="list-style-type: none"> 1. El sistema muestra el listado de expedientes y los botones crear, mostrar, editar, firmar, eliminar y los campos de búsqueda. 3. El sistema muestra una caja de dialogo con el formulario para introducir un nuevo expediente. 5. El sistema valida los datos. 6. El sistema cierra la caja de dialogo y envía una notificación de que la inserción ha sido satisfactoria. 7. El sistema actualiza el listado de expedientes.
Flujos alternos Sección “Insertar expedientes”	
Acción de actor	Respuesta del sistema
	5.1 Si los campos están vacíos o los valores no son válidos el sistema manda un mensaje de error.
Sección “Modificar expedientes”	
Acción de actor	Respuesta del sistema

Capítulo 2: Características del Sistema

<ol style="list-style-type: none"> 2. El especialista selecciona el expediente que se va a modificar. 3. El especialista presiona el botón modificar. 5. El usuario llena los datos del formulario y presiona el botón salvar. 	<ol style="list-style-type: none"> 1. El sistema muestra el listado de expedientes y los botones crear, mostrar editar, firmar, eliminar y los campos de búsqueda. 4. El sistema muestra una caja de dialogo con el formulario para modificar el expediente. 6. El sistema valida los datos. 7. El sistema cierra la caja de dialogo y envía una notificación de que la modificación ha sido satisfactoria. 8. El sistema actualiza el listado de expedientes.
Flujos alternos Sección “Modificar expedientes”	
Acción de actor	Respuesta del sistema
	6.1 Si los campos no son válidos o los valores no son válidos el sistema manda un mensaje de error.
Sección “Mostrar expedientes”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 2. El especialista selecciona el expediente que se va a mostrar. 3. El especialista presiona el botón mostrar 	<ol style="list-style-type: none"> 1. El sistema muestra el listado de expedientes y los botones crear, mostrar editar, firmar, eliminar y los campos de búsqueda. 4. El sistema muestra la información del expediente con sus componentes.
Sección “Listar expedientes”	
Acción de actor	Respuesta del sistema
	1. El sistema muestra el listado de expedientes y los botones crear, mostrar editar, firmar, eliminar y los campos de búsqueda.
Flujos alternos Sección “Listar expedientes”	
Acción de actor	Respuesta del sistema
	1.1 Si no existen expedientes se muestra la lista vacía.
Sección “Buscar expedientes”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 2. El especialista busca el expediente creado por los campos de búsqueda que desee. 	<ol style="list-style-type: none"> 1. El sistema muestra el listado de expedientes y los botones crear, mostrar editar, firmar, eliminar y los campos de búsqueda. 3. El sistema va actualizando la lista según la búsqueda.
Sección “Eliminar expedientes”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 2. El especialista selecciona el expediente que va a eliminar. 3. El especialista presiona el botón eliminar. 	<ol style="list-style-type: none"> 1. El sistema muestra el listado de expedientes y los botones crear, mostrar editar, firmar, eliminar y los campos de búsqueda. 4. El sistema elimina el expediente con sus componentes. 5. El sistema envía una notificación de que se eliminado el expediente satisfactoriamente.
Sección “Firmar expedientes”	

Capítulo 2: Características del Sistema

Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 2. El especialista selecciona el expediente que va a firmar. 3. El especialista presiona el botón firmar. 	<ol style="list-style-type: none"> 1. El sistema muestra el listado de expedientes y los botones crear, mostrar editar, firmar, eliminar y los campos de búsqueda. 4. El sistema muestra una caja de dialogo con el formulario para insertar la contraseña de la firma digital. 5. El sistema valida la contraseña de la firma digital. 6. El sistema crea el archivo del expediente técnico. 7. El sistema firma el archivo de expediente técnico. 8. El sistema envía una notificación de que la firma fue creada satisfactoriamente.
Flujos alternos Sección “Firmar expedientes”	
Acción de actor	Respuesta del sistema
	5.1 Si la contraseña de la firma no es válida envía una notificación de error y el sistema no firma.

Casos de uso	Gestionar usuario
Actores	Especialista
Resumen	Es iniciado cuando el especialista realiza alguna de las operaciones: Insertar, Modificar, Eliminar o Listar. La aplicación muestra una interfaz con el listado de usuarios y las posibles acciones que puedes realizar. El caso de uso finaliza cuando se ejecuta la acción solicitada.
Precondición	El usuario tiene que estar autenticado como especialista.
Referencia	Caso de uso No 8 de la aplicación desktop
Prioridad	Critico
Flujo normal de eventos	
Acción de actor	Respuesta del sistema
1. El especialista selecciona el ítem “Usuarios” en el submenú Gestión de Usuario.	2. El sistema muestra el listado de usuarios y los botones crear, mostrar, editar y borrar .
Sección “Crear usuario”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El especialista oprime el botón crear. 3. El especialista introduce los datos del usuario 	2. El sistema muestra una caja de dialogo con el formulario para introducir un nuevo usuario.

Capítulo 2: Características del Sistema

y oprime el botón salvar.	<ol style="list-style-type: none"> 4. El sistema valida los datos. 5. El sistema cierra la caja de diálogo y envía una notificación de que la inserción ha sido satisfactoria.
Flujo alternativo Sección "Crear usuario"	
Acción de actor	Respuesta del sistema
	<ol style="list-style-type: none"> 4.1 Si los campos están vacíos o los valores no son válidos el sistema manda un mensaje de error.
Sección "Mostrar usuario"	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 4. El especialista selecciona el usuario que desea mostrar de la lista y oprime el botón mostrar. 5. El especialista oprime el botón cerrar para cerrar la caja de diálogo mostrar usuario. 	<ol style="list-style-type: none"> 6. El sistema muestra una caja de diálogo con los datos del usuario seleccionado. 7. El sistema cierra la caja de diálogo.
Sección "Editar usuario"	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El especialista selecciona el usuario que desea editar de la lista y oprime el botón editar. 2. El especialista introduce el dato del usuario que desea modificar y oprime el botón salvar. 	<ol style="list-style-type: none"> 3. El sistema muestra una caja de diálogo con el formulario para editar el usuario seleccionado. 4. El sistema valida los datos. 5. El sistema cierra la caja de diálogo y envía una notificación de que la edición ha sido satisfactoria.
Flujo alternativo Sección "Editar usuario"	
Acción de actor	Respuesta del sistema
	<ol style="list-style-type: none"> 4.1 Si los campos están vacíos o los valores no son válidos el sistema manda un mensaje de error.
Sección "Eliminar usuario"	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El especialista selecciona el usuario que desea eliminar de la lista y oprime el botón eliminar. 	<ol style="list-style-type: none"> 2. El sistema elimina el usuario y muestra la lista actualizada.

2.7 Patrones de casos de usos

Un patrón se define como una solución probada con éxito que aparece una y otra vez ante determinado tipo de problema en un contexto dado. Los patrones se definen por un nombre, un problema, una solución y las consecuencias de su aplicación; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias y considera los puntos fuertes y compromisos.

Capítulo 2: Características del Sistema

Extensión: Este patrón se utiliza en la aplicación web cuando el usuario acceda a la aplicación web autenticándose contra el LDAP, donde se tiene que consumir el servicio web para que se pueda ejecutar la funcionalidad autenticar.

Inclusión: Este patrón se utiliza en la aplicación web en el caso de uso casos de usos Gestionar usuario, ya que no es necesario que siempre se realicen todas las funcionalidades.

Múltiples Actores: Este patrón se utiliza en la aplicación web, ya que los usuarios pueden tener roles comunes.

CRUD (Creating, Reading, Updating, Deleting): Este patrón se utiliza en la aplicación web, ya que los casos de uso insertar expedientes, actualizar expedientes, mostrar expedientes y borrar expedientes conforman el caso de uso gestionar expedientes.

CRUD Parcial: Este patrón se utiliza en la aplicación web, debido a que el caso de uso gestionar incidencia está compuesto por las funcionalidades crear incidencia, mostrar y editar incidencia.

Conclusiones Parciales

- ✓ Se identificaron diversos requisitos con los que el sistema debe cumplir para lograr que funcione de manera eficiente.
- ✓ Se hizo uso de diferentes patrones que condicionaron el comportamiento de la aplicación y proporcionaron diversas soluciones para fomentar la reutilización y extensibilidad del código.
- ✓ Para garantizar la seguridad del sistema se definió que solo el especialista puede cambiar el rol que posea el usuario.

CAPÍTULO 3: DISEÑO DEL SISTEMA

Introducción

Este capítulo está destinado a describir las fases de análisis y diseño que propone la metodología Open Up; Así como los artefactos que exige la misma para esta fase, como, por ejemplo, diagrama de clases, diagrama entidad relación y diagrama de despliegue. Se establece el uso de diversos patrones que permitieron estructurar de manera correcta el sistema.

3.1 Patrones

Es una descripción de un problema y la solución, a la que se da un nombre y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias y considera los puntos fuertes y compromisos.

3.1.1 Patrones Arquitectónicos

Centrados en la arquitectura del sistema. Definen una estructura fundamental sobre la organización del sistema. Proveen un conjunto predefinido de subsistemas, cuáles son sus responsabilidades y como se interrelacionan.

Arquitectura Cliente/Servidor

En la aplicación informática se hace uso de la arquitectura Cliente/Servidor ya que todos los clientes están conectados a través del uso de la red. El servidor debe ser lo suficientemente potente debido a que todas las gestiones que se realizan se concentran en él, mientras que el cliente puede cumplir con especificaciones más ligeras debido a que solo va a realizar peticiones, esperar y recibir respuesta del servidor. En dicha aplicación se utilizó como servidor de aplicaciones GlassFish, servidor de BD PostgreSQL y como cliente los navegadores web. La centralización del control, la escalabilidad, el fácil mantenimiento y la seguridad son algunas de las ventajas que provee este modelo y que se decidió aprovechar para beneficio del sistema propuesto.

Modelo Vista Controlador (Modelo 2)

Modelo-Vista-Controlador (MVC) es una arquitectura de diseño de software que separa los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos, en este caso se utilizó el *framework* JavaServer Faces que , donde el Modelo va a constar de objetos java entre los que se encuentran los EJB y JavaBeans, la Vista contiene las páginas JSF , el código que provee datos dinámicos a las misma y el Controlador se implementa siempre como un *servlet* FrontController llamado FacesServlet el cual recibe peticiones y ejecuta un conjunto de pasos para servir una respuesta, un conjunto de *Action Event* y archivos de configuración. Este modelo permitió normalizar y estandarizar el desarrollo del Software, además de brindar una API muy bien definida; que cualquiera que

decida utilizarla, podrá reemplazar el modelo, la vista o el controlador, sin demasiada dificultad, mientras que la conexión entre el modelo y sus vistas es dinámica y se produce en tiempo de ejecución, no en tiempo de compilación.

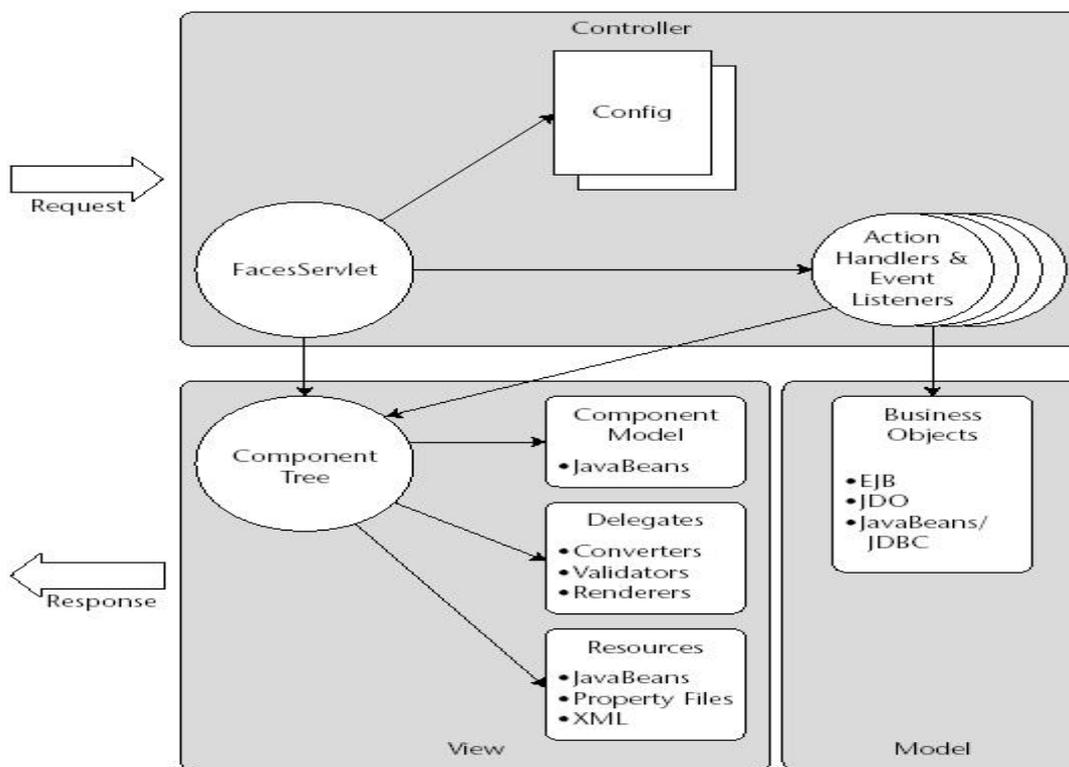


Fig. No 3.1 Arquitectura Modelo Vista Controlador (Modelo 2).

3.1.2 Patrones de diseño

Relacionados con el diseño de los objetos y *frameworks* de pequeña y mediana escala. Aplicables al diseño de una solución para conectar los elementos de gran escala que se definen mediante los patrones de arquitectura y durante el trabajo de diseño detallado para cualquier aspecto del diseño local. También se conocen como patrones de micro-arquitectura.

Patrones de diseño GRASP

Los patrones de diseño son descripciones de clases y objetos relacionados, que están particularizados para resolver un problema de diseño general en un determinado contexto (LARMAN 1999).

Experto: Este patrón es muy utilizado para asignar responsabilidades. Una clase contiene la información necesaria para llevar a cabo sus funcionalidades. Permite conservar el encapsulamiento, debido a que los objetos se valen de su propia información para cumplir lo que se le pide. Proporciona que las clases cuenten con la funcionalidad requerida, brindando así una alta cohesión.

Se aplica en el sistema en clases como **ComputadoraController.java**, **MotherboardController.java**, **UsuariosController.java**.

Capítulo 3: Diseño del Sistema

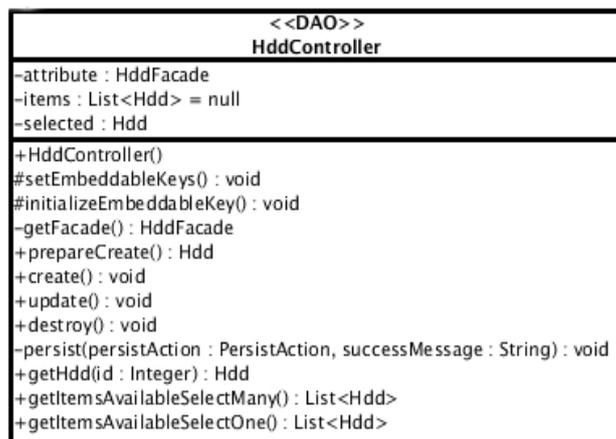


Fig. No 3.2 Ejemplo de clase que utiliza el Patrón Experto

Creador: Este patrón establece la creación de instancias entre clases. Brinda un soporte a bajo acoplamiento. Guía la asignación de responsabilidades relacionada con la creación de objetos, tarea frecuente en la programación orientada a objeto.

Se aplica en el sistema en clases como **CreatePdf.java**, **DireccionController.java**.

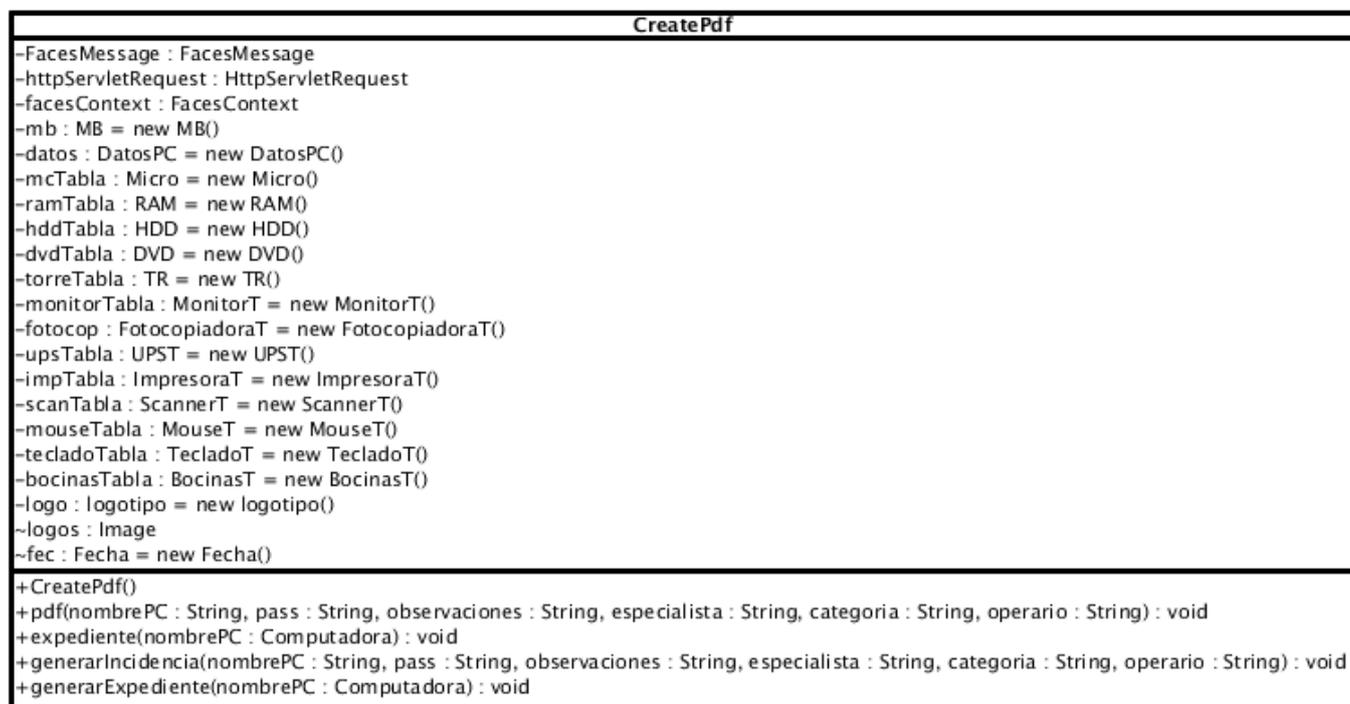


Fig. No 3.3 Ejemplo de clase que utiliza el Patrón Creador

Alta Cohesión: Este patrón se encarga de guiar el diseño. Permite que las clases del diseño realicen las funcionalidades necesarias para cumplir con su responsabilidad. Mejora la claridad y facilidad para entender el diseño. Busca soluciones para asignar los métodos a las clases de forma coherente, completa y

Capítulo 3: Diseño del Sistema

relacionada, permitiendo el cambio, poniendo toda la información que se necesita controlar a la vista en el mismo fichero. Fomenta la reutilización.

Se aplica en el sistema en clases como **Bocinas.java**, **Cpu.java**, **Fecha.java**.

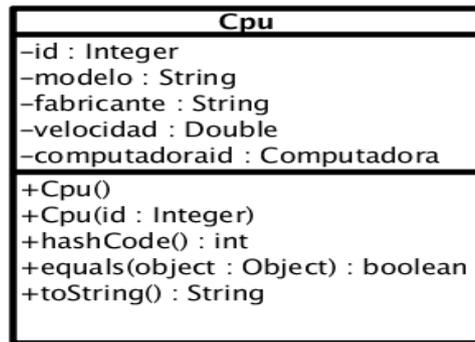


Fig. No 3.4 Ejemplo de clase que utiliza el Patrón Alta Cohesión.

Bajo acoplamiento: Es imprescindible utilizarlo cuando se vaya a diseñar. Permite la reutilización y el diseño de clases más independientes, minimiza el impacto de los cambios.

Se aplica en el sistema en clases como **CpuFacade.java**, **UsuarioFacade.java**.

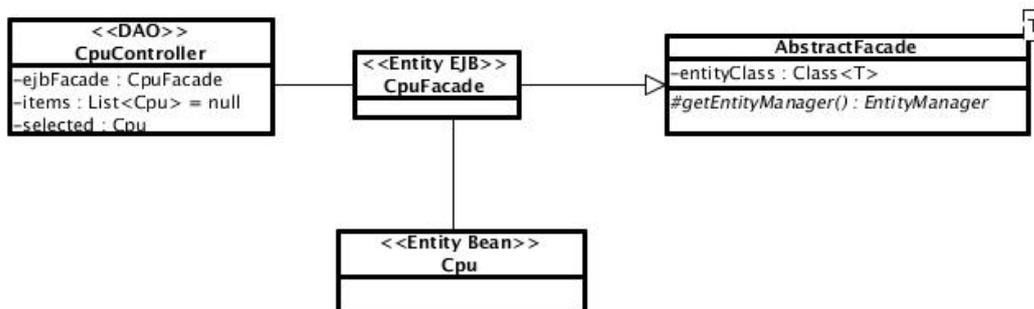


Fig. No 3.5 Ejemplo de clase que utiliza el Patrón Bajo Acoplamiento.

Controlador: Asigna la responsabilidad de gestionar un mensaje de un evento del sistema a una clase que represente una de estas dos opciones:

- ✓ Representa el sistema global, dispositivo o subsistema (controlador de fachada).
- ✓ Representa un escenario de caso de uso en el que tiene lugar el evento del sistema (controlador de caso de uso o de sesión).

Se aplica en el sistema en clases como **JsUtil.java**.

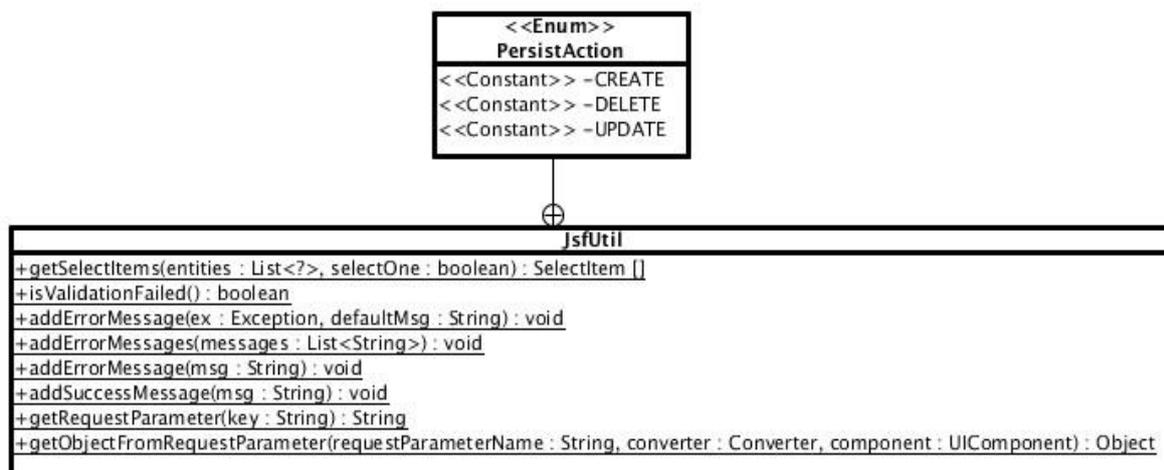


Fig. No 3.6 Ejemplo de clase que utiliza el Patrón Controlador

Patrones de diseño JEE6

“Los patrones de diseño JEE contienen las mejores soluciones para ayudar a los desarrolladores a diseñar y construir aplicaciones en la plataforma JEE”.

La mayoría de los proyectos Java EE 6 continúan utilizando patrones J2EE, pero esta plataforma fue intrusiva, la separación de la infraestructura técnica y la lógica de negocio, necesitaba la introducción obligatoria de patrones, que eran en su mayoría soluciones para las deficiencias de J2EE. La funcionalidad de esta nueva versión de la plataforma Empresarial de Java, demanda un replanteamiento de los patrones de diseño utilizados en J2EE, así como la eliminación de código superfluo requerido para la implementación de aplicaciones en dicha plataforma.

Patrones de la capa de presentación.

La capa de presentación incluye *Servlets*, *JSPs* y otros elementos que generan interfaz de usuario del lado del servidor.

Front Controller

El patrón *FrontController* acepta todas las peticiones de un cliente y las direcciona a los manejadores apropiados.

Solución: El patrón *FrontController* centraliza la lógica de control impidiendo su duplicación, administra la invocación de los servicios de seguridad, como la autenticación, controla la elección de una vista apropiada y el manejo de errores.

Estrategia: *Servlet Front*, *JSP Front*, *Base Front*.

Beneficios: Promueve el particionamiento transparente de la aplicación, fomenta la reutilización y provee un punto de obstrucción para los intentos de acceso ilícitos en la aplicación Web.

En la Fig. 3.7 se muestra la configuración del *FacesServlet* de la aplicación Web, contenido en el archivo web.xml.

```
<context-param>
  <param-name>javax.faces.PROJECT_STAGE</param-name>
  <param-value>Development</param-value>
</context-param>
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<welcome-file-list>
  <welcome-file>faces/index.xhtml</welcome-file>
</welcome-file-list>
<context-param>
  <param-name>primefaces.THEME</param-name>
  <param-value>cupertino</param-value>
</context-param>
<context-param>
  <param-name>primefaces.FONT_AWESOME</param-name>
  <param-value>>true</param-value>
</context-param>
```

Fig. No 3.7 Fragmento del archivo de configuración web.xml.

Composite View

Solución: Cada componente de la plantilla se puede incluir forma dinámica y el diseño de la página es administrada independientemente del contenido.

Estrategia: JavaBean View Management, Standard Tag View Management, Custom Tag View Management, Transformer View Management.

Beneficios: Las modificaciones en el diseño de una página se gestionan fácilmente porque los cambios están centralizados y es posible reutilizar fragmentos de una plantilla, en varias vistas y re-decorar estos fragmentos con información diferente.

Capítulo 3: Diseño del Sistema

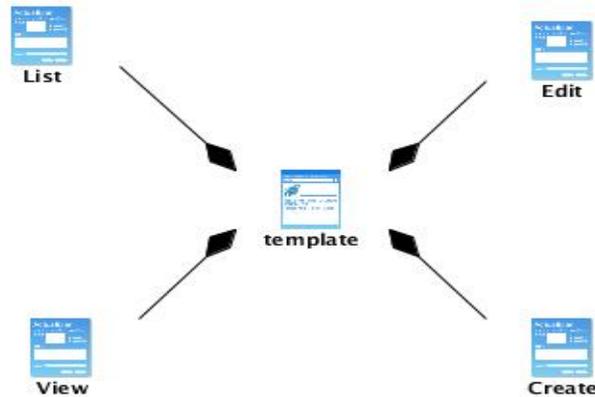


Fig. No 3.8 Ejemplo del Patrón Composición.

Patrones de la capa de negocio.

La capa de negocio expone la lógica necesaria para que el usuario a través de la interfaz, interactúe con las funcionalidades de la aplicación.

La plataforma Java EE define el uso de componentes de negocio EJB para abstraer la lógica de negocio de otros problemas generales de las aplicaciones como la concurrencia, transacciones, persistencia y seguridad.

Service Facade (Application Service).

El Service Facade es la versión mejorada del patrón Application Service. Es un Bean de sesión sin estado con una interfaz local de negocio que debe proporcionarse remotamente sólo si se va a utilizar fuera de la Máquina Virtual de Java (JVM) y no se inyecta en un *Servlet*, Bean de respaldo, u otro componente web.

Solución: Su responsabilidad principal continúa siendo la composición de servicios independientes y reutilizables, pero en casos sencillos, se puede realizar directamente la lógica de negocio sin delegación a otros participantes y está diseñado para coordinar DAOs y servicios independientes, para asegurarse de que tales servicios puedan ser reutilizados en otros contextos.

Estrategia: Fachada Crud, Fachada SOA, Fachada Ligera Asíncrona.

Beneficios: Asegura la consistencia, es importante para la comunicación remota, toda la comunicación se encamina a través de la fachada. Por lo que es un lugar excelente para la implementación de inquietudes transversales como monitoreo, auditoría, o requisito previo de control.

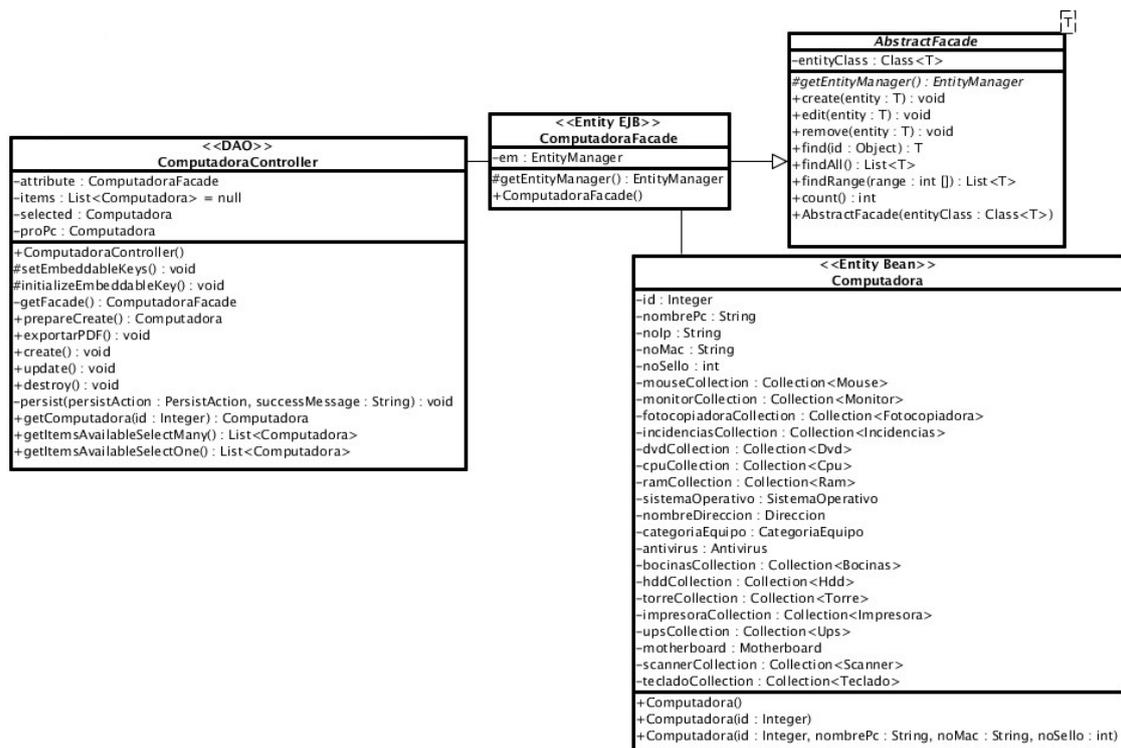


Fig. No 3.9 Ejemplo del Patrón Service Facade.

Patrones de la capa de integración.

La capa de integración agrupa componentes de conexión con otros sistemas, conectores con servicios y servicios de mensajería.

Data Access Object (DAO).

Este patrón permite separar el acceso a datos de la lógica del negocio. Presenta una interfaz común entre las aplicaciones y uno o más dispositivos de almacenamiento de datos y de recuperación. Cuenta con diversas fuentes de datos, oculta los detalles de la implementación y encapsula la forma de acceder a la fuente de datos. El software cliente se centra en el acceso a los datos que necesita, olvidándose de cómo realiza el acceso o cual es la fuente de almacenamiento. Un cambio en el origen de los datos no afecta la capa superior siempre que se implemente correctamente la interfaz.

Solución: Proporciona un adecuado acceso al almacén de datos y es el encargado de la encapsulación de APIs y tecnologías patentadas. Su principal responsabilidad es la gestión de los recursos y la conversión de datos en objetos de dominio de nivel superior.

Estrategia: Generic DAO, Domain specific DAO, Attached result DAO, Detached result DAO, Back-end Integration DAO, Abstract DAO.

Beneficios: Minimiza la duplicación de código y mejora la capacidad de mantenimiento.

Capítulo 3: Diseño del Sistema

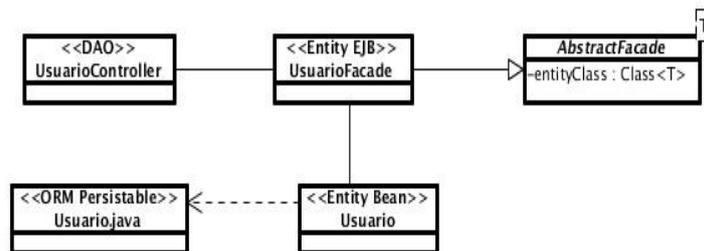


Fig. No 3.11 Ejemplo del Patrón DAO.

3.2 Diagrama de clases del diseño

Un Diagrama de Clases de Diseño muestra definiciones de entidades software más que conceptos del mundo real. Estos diagramas describen gráficamente las especificaciones de las clases de software y las interfaces y son utilizados durante el proceso de análisis y diseño del sistema. En la figura 3.2 se muestra el diagrama de clase de diseño correspondiente al caso de uso Autenticar usuario. En él se establecen las clases que se utilizan para lograr la autenticación de los usuarios, entre las que se encuentran la clase login. Se modela la clase entidad Usuarios como clases de acceso a datos, el formulario que se le mostrara al cliente y las relaciones entre los elementos.

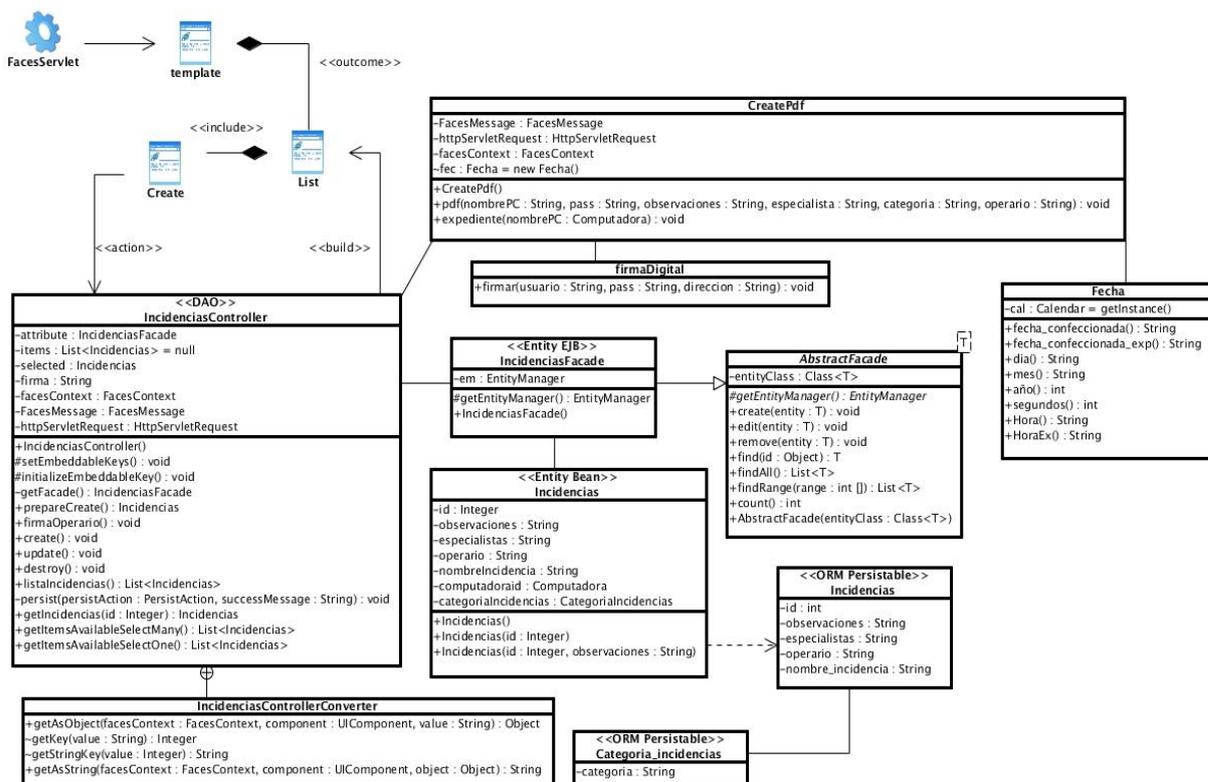


Fig. No 3.12 Diagrama de clase del diseño del caso de uso Firmar incidencia 1.

3.3 Diagrama de secuencias

El diagrama de secuencia de UML muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo. Estos están compuestos por objetos con su línea de vida, mensajes intercambiados entre objetos de una secuencia ordenada y una Línea de vida activa, se utilizan con frecuencia para validar los casos de uso. En la figura 3.13 se muestra el diagrama de secuencias correspondiente al caso de uso Firmar.

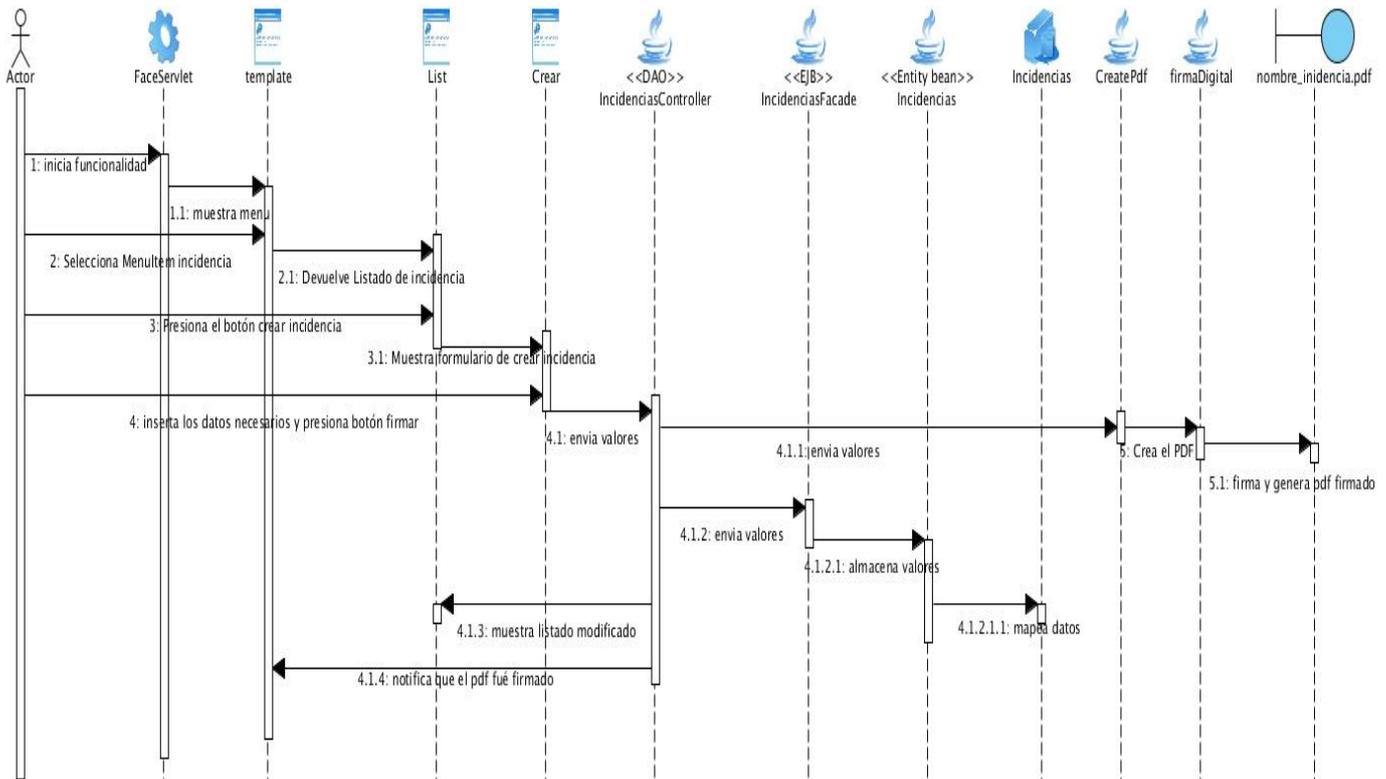


Fig. No 3.13 Diagrama de secuencias del caso de uso Firmar Incidencia 1.

3.4 Diseño de base de datos

El diseño de una base de datos es de suma importancia ya que de ello dependerá los datos estén correctamente actualizados y la información siempre sea exacta. Si se hace un buen diseño de base de datos se podrá obtener reportes efectivos y eficientes. Permitirá definir la estructura de los datos que debe tener la base de datos del sistema.

En la figura 3.14 se muestra el diagrama de donde parte el diseño de la BD a implementar.

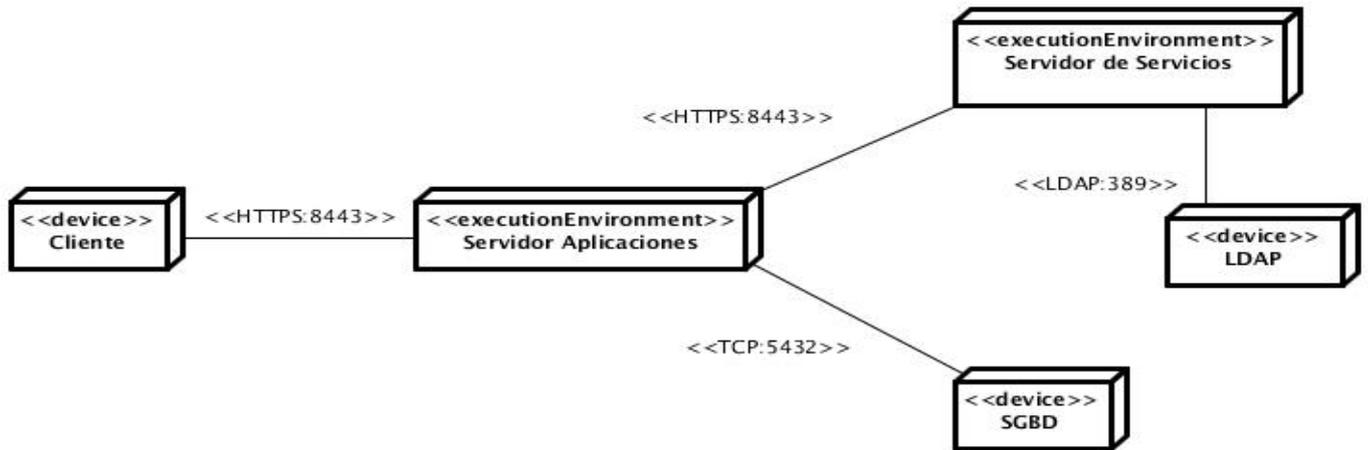


Fig. No 3.15 Diagrama de despliegue.

Nodo: Un nodo es un elemento de hardware o software. Se representa en forma de caja en tres dimensiones. **Descripción de los nodos**

Cliente: Representa los ordenadores que van permitirle a los usuarios acceder a la aplicación web.

Servidor Aplicaciones: Representa el servidor GlassFish donde se encuentra desplegada la aplicación

Servidor de Servicios: Representa el servidor donde se encuentra desplegado los servicios utilizados.

LDAP: Representa el servidor donde se encuentra instalado el directorio active con el servicio LDAP.

SGBD: Representa el servidor donde se encuentra instalado el gestor de base de datos PostgreSQL con la BD del sistema.

Conclusiones parciales

- ✓ Para lograr una mayor organización se hizo uso de los patrones, Cliente-Servidor y MVC (Modelo 2) con el *framework* JavaServer Faces, lo que permitió normalizar y estandarizar el desarrollo del Software.
- ✓ La inclusión de los patrones *JEE6* facilitó el uso de soluciones para desarrollar, diseñar y construir la aplicación en la plataforma *JEE*, la cual brinda soluciones para las deficiencias de *J2EE*.
- ✓ Los usuarios se autentican en la aplicación a través de un servicio web que les permite acceder al *LDAP*, el cual les brinda la información de los usuarios de AICA.
- ✓ La conexión a la aplicación web se efectúa mediante el puerto 8443 del protocolo *HTTPS* para lograr una mayor seguridad.

Capítulo 4: Implementación y Resultados esperados

Introducción

Al concluir el diseño del sistema, se realiza la descripción de implementación en términos de componentes y los estándares de codificación del Gestor de Expedientes Técnicos; además de las pruebas de caja negra realizadas con el propósito de comprobar y documentar cómo se comporta este sistema cuándo se miden sus principales funcionalidades, de acuerdo a las especificaciones descritas en los casos de uso del sistema.

4.1 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Estos diagramas muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente y demás. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

A continuación, en la figura 4.1, se muestra el diagrama de componentes que representa la vista estática del sistema, así como todos los paquetes y las relaciones de dependencias que se establecen entre los componentes y en la figura 4.2 el diagrama de componentes del componente GET.

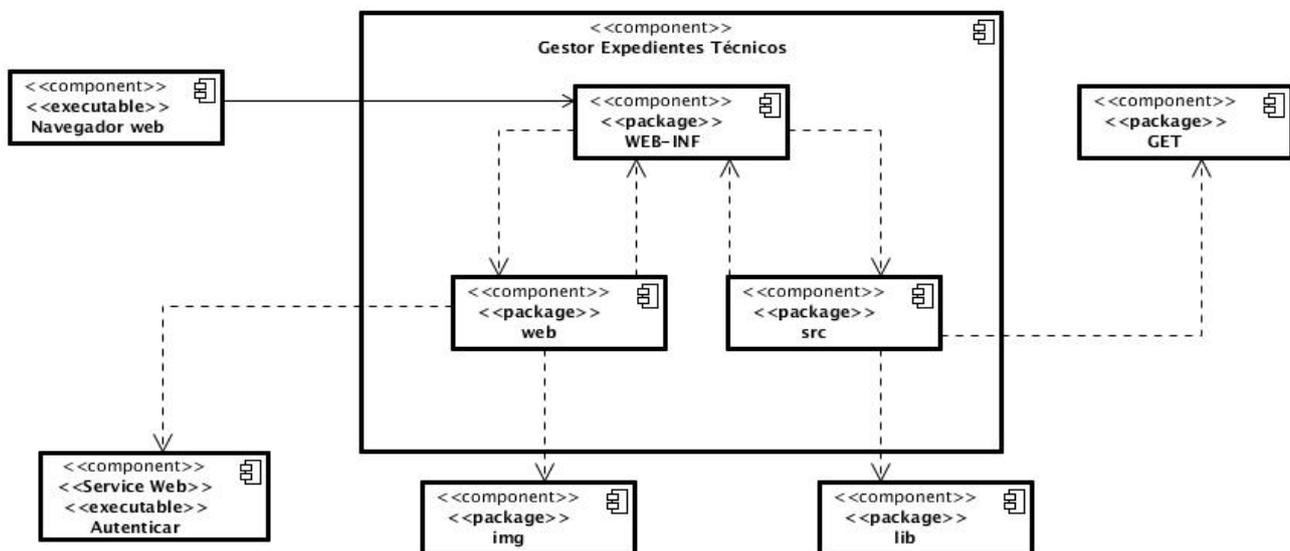


Fig. No.4.1 Diagrama de componentes de la aplicación web.

Descripción de los componentes del diagrama de componente de la aplicación web.

- ✓ **Navegador web:** Este componente indica que se va a acceder a la aplicación mediante un navegador web.
- ✓ **Gestor Expedientes Técnicos:** Contiene a los paquetes de la aplicación.

Capítulo 4: Implementación y Resultados Esperados

- ✓ **WEB-INF:** Contiene los archivos de configuración del sistema y encargados de controlar todos los eventos de la aplicación, además de las librerías.
- ✓ **web:** Contiene las vistas de la aplicación.
- ✓ **src:** Contiene los paquetes **JAVA**.
- ✓ **Autenticar:** Este componente representa el servicio web que autentica contra el LDAP.
- ✓ **img:** Este componente contiene el logo de la entidad.
- ✓ **lib:** Este componente contiene todas las librerías usadas por el sistema.
- ✓ **GET:** Este componente contiene la estructura necesaria para el almacenamiento de la documentación generada por la aplicación (Expedientes Técnicos e Incidencias).

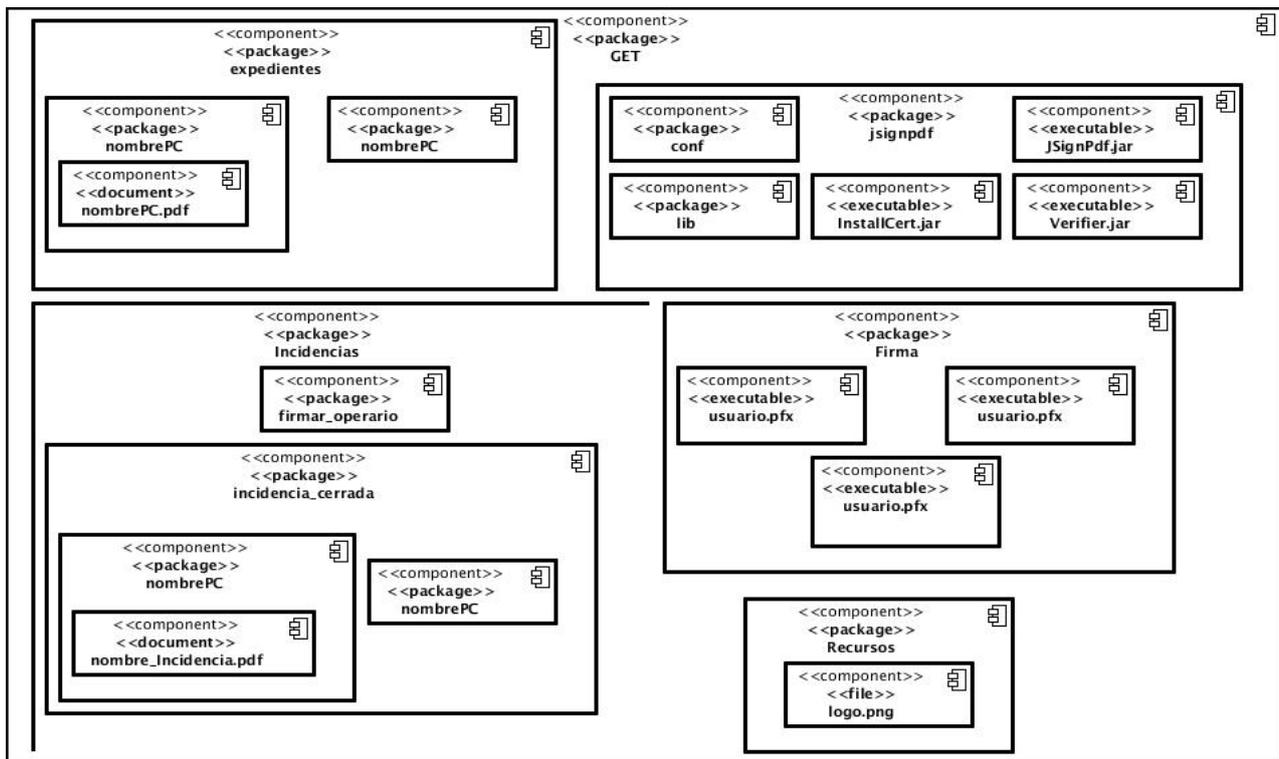


Fig. No.4.2 Diagrama de componentes del componente GET.

Descripción de los componentes del diagrama de componente GET.

- ✓ **GET:** Este componente contiene la estructura necesaria para el almacenamiento de la documentación generada por la aplicación (Expedientes Técnicos e Incidencias).
- ✓ **expedientes:** Contiene los expedientes técnicos generados y firmados por la aplicación.
- ✓ **incidencias:** Contiene las incidencias de cada equipo generadas por la aplicación.
- ✓ **Firma:** Contiene los archivos de firma digital de cada usuario.
- ✓ **nombrePC:** Este componente representa una carpeta con el nombre del equipo.
- ✓ **nombrePC.pdf:** Este componente representa el expediente generado y firmado por la aplicación.
- ✓ **firmar_operario:** Este componente representa la carpeta donde se almacenan las incidencias que han sido firmadas por los especialistas y están en espera de ser firmadas por el operario.

Capítulo 4: Implementación y Resultados Esperados

- ✓ **Incidencia_cerrada:** Este componente representa la carpeta donde se almacena las incidencias que se han firmado por el operario, estas incidencias han sido eliminadas del paquete **firmar_operario**.
- ✓ **nombre_Incidencia.pdf:** Este componente representa la incidencia creada por el sistema.
- ✓ **jsingpdf:** Este componente representa la carpeta donde se encuentran las librerías utilizadas para realizar la firma digital.
- ✓ **JsingPdf.jar:** Este componente representa la librería utilizada para realizar la firma digital.
- ✓ **usuario.pfx:** Este componente representa el archivo de firma digital.
- ✓ **Recursos:** Este componente representa la carpeta donde se guarda las imágenes de generadas en la documentación.
- ✓ **Logo.png:** Este componente contiene el logo de la entidad que aparece en los expedientes y en las incidencias.

4.2 Estándares de Codificación

Un estándar de codificación, son reglas que se siguen para la escritura del código fuente. Para una mejor comprensión del código, se definieron una serie de estándares basados en diversas reglas.

Estilos para la capitalización

Clases: Para todo nombre de clase, la primera letra debe de ser mayúscula, si son varias palabras se debe de intercalar entre mayúsculas y minúsculas, este mecanismo de nombre es llamado **CamelCase**. Por ejemplo: **MotherboardFacade**, **ImpresoraController** y **HddFacade**.

Métodos: Para los métodos de clases, la primera letra debe ser minúscula, si son varias palabras se debe de intercalar entre minúsculas y mayúsculas, para el caso de los métodos de clases aplica el mecanismo del **CamelCase**. Por ejemplo, **getEntityManager**, **getItemsAvailableSelectMany** y **getEspecialistas**.

Variables: Para las variables, se aplica el caso de los métodos, donde la primera letra es minúscula y las demás se deben de guiar por el mecanismo de **CamelCase**, deben ser cortos y descriptivos en sí mismo. Por ejemplo: **ejbFacade**, **items** y **selected**.

```
@Named("computadoraController")
@SessionScoped
public class ComputadoraController implements Serializable {
```

Fig. No.4.3 Convención CamelCase para el nombre de las clases.

```
public String getEspecialistas() { ...3 lines }  
  
public void setEspecialistas(String especialistas) { ...3 lines }
```

Fig. No.4.4 Convención CamelCase para el nombre de los métodos.

```
@EJB  
private beans.HddFacade ejbFacade;  
private List<Hdd> items = null;  
private Hdd selected;
```

Fig. No.4.5 Convención CamelCase para el nombre de las variables.

Reglas de codificación

- ✓ No usar caracteres simples para el nombre de las variables i, n, s, entre otros. Una excepción de esta regla son las variables dentro de los ciclos.
- ✓ No usar nombres de variables que coincidan con palabras reservadas.
- ✓ Los comentarios deben estar en el mismo nivel del código.
- ✓ Las llaves se deben poner al mismo nivel del código que las contiene.
- ✓ Usar una línea en blanco para separar agrupaciones lógicas del código.
- ✓ Debe dejarse una y solo una línea en blanco entre cada método dentro de las clases.
- ✓ Las llaves deben ser utilizadas sobre líneas separadas y no sobre la misma línea como en *if*.

4.3 Pruebas

Las pruebas de software consisten en la dinámica de la verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general infinitas ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo (Sandoval 2011).

La prueba de software es un elemento crítico para la garantía del correcto funcionamiento del software. Entre sus objetivos están:

- ✓ Detectar defectos en el software.
- ✓ Verificar la integración adecuada de los componentes.
- ✓ Verificar que todos los requisitos se han implementado correctamente.
- ✓ Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- ✓ Diseñar casos de prueba que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Capítulo 4: Implementación y Resultados Esperados

A continuación, se muestran las pruebas de caja negra que se realizaron sobre la interfaz del sistema a desplegar.

Prueba de Caja Negra

Se decide realizar este método de prueba, debido a que se conoce la función específica por la que se construyó este software, por cual se aplica este tipo de prueba que demuestra que cada función es plenamente operacional, mientras se buscan los errores de cada función.

La técnica utilizada en la presente investigación se denomina Prueba de Partición Equivalente, la cual divide el dominio de entrada de un programa en clases de datos, a partir de las cuales derivan los casos de prueba. Cada una de estas clases de equivalencia representa a un conjunto de estados válidos o inválidos para las condiciones de entrada.

Casos de prueba

Un caso de prueba es una serie de pruebas de entrada, condiciones de ejecución y resultados esperados desarrollados para un objetivo en particular, tal como ejecutar una ruta particular de un programa o verificar el cumplimiento con un requerimiento en específico.

Para desarrollar un software de calidad y libre de errores, el plan de pruebas y los casos de prueba son muy importantes. El gestor de expedientes técnicos se diseña para lograr un mayor control de los recursos y la información por lo que se debe identificar el dominio y sus características a probar, lo mismo que el tipo de pruebas a realizar. Un caso de prueba bien diseñado tiene gran posibilidad de llegar a resultados más fiables y eficientes, mejorar el rendimiento del sistema y reducir los costos en tres categorías: productividad (menos tiempo para escribir y mantener los casos), capacidad de prueba (menos tiempo para ejecutarlos) programar la fiabilidad (estimaciones más fiables y efectivas) (O. 2010).

Los casos de prueba de la caja negra pretenden demostrar que:

- ✓ Las funciones del software son operativas.
- ✓ La entrada se acepta de forma correcta.
- ✓ Se produce una salida correcta.
- ✓ La integridad de la información externa se mantiene.

Autenticación				
Escenario	Entrada	Resultados Esperados	Resultados Obtenidos	Condiciones
Campos vacíos	El actor deja vacío los campos y presiona el botón aceptar de la página de autenticación	Debe salir un mensaje de error de campo vacío	El sistema muestra el mensaje de error satisfactoriamente	El campo debe estar vacío
No existe	El actor introduce usuario y	Debe salir el	El sistema muestra	Usuario no existe

Capítulo 4: Implementación y Resultados Esperados

usuario en base de datos	contraseña y presiona el botón entrar	mensaje de error que diga que el usuario no existe en la base de datos del sistema	el mensaje satisfactoriamente	en la base de datos del sistema
Usuario o Contraseña incorrecto	El actor introduce usuario y contraseña y presiona el botón entrar	Debe salir la notificación "El usuario o la contraseña son incorrectos"	El sistema muestra el mensaje satisfactoriamente	El usuario o la contraseña deben de estar incorrectos
El usuario se autentica satisfactoriamente	El actor introduce usuario y contraseña y presiona el botón entrar	Debe salir la notificación "Usted se autenticado como Especialista u Operario"	El sistema muestra la notificación satisfactoriamente	El usuario tiene que autenticarse

Crear nuevo expediente				
Escenario	Entrada	Resultados Esperados	Resultados Obtenidos	Condiciones
Campos vacíos	El actor deja campos vacíos y presiona el botón salvar del formulario de crear expediente	Debe salir un mensaje de error de campo vacío	El sistema muestra el mensaje de error satisfactoriamente	El campo debe estar vacío
Campo Dirección IP incorrecto	El actor introduce incorrectamente la dirección IP	Debe salir el mensaje de error "Por favor introduzca correctamente la dirección IP"	El sistema muestra el mensaje de error satisfactoriamente	El campo Dirección IP debe ser incorrecto
Campo Dirección MAC incorrecto	El actor introduce incorrectamente la dirección MAC	Debe salir el mensaje de error "Por favor introduzca correctamente la Dirección MAC"	El sistema muestra el mensaje de error satisfactoriamente	El campo Dirección MAC debe ser incorrecto
Campo Número de sello es incorrecto	El actor introduce un dato no entero en el campo Número de sello.	Debe salir el mensaje de error "El campo Número de sello debe tener dígitos solamente"	El sistema muestra mensaje de error satisfactoriamente	El campo Número de sello debe estar formado por caracteres no numéricos
Todos los campos fueron bien llenados	El actor llena los campos con valores correctos.	Debe salir el mensaje "El expediente fue exitosamente creado"	El sistema muestra el mensaje satisfactoriamente	Los campos deben ser llenados correctamente
Datos duplicados	El actor llena los campos con información existente en la base de datos.	Debe el mensaje de error "El expediente ya existe, introduzca uno nuevo".	El sistema muestra el mensaje de error satisfactoriamente	El valor de los campos debe encontrarse en la base de datos

Crear nueva incidencia

Capítulo 4: Implementación y Resultados Esperados

Escenario	Entrada	Resultados Esperados	Resultados Obtenidos	Condiciones
Campos vacíos	El actor no llena algún campo del formulario	Debe salir el mensaje de error de campo vacío	El sistema muestra el mensaje de error satisfactoriamente	El campo debe estar vacío
Contraseña o firma no válida	El actor introduce los datos e introduce mal la contraseña de la firma digital	Debe salir el mensaje de error "No se pudo firmar la Incidencia, compruebe si la contraseña o la firma es correcta "	El sistema muestra el mensaje de error satisfactoriamente	La contraseña o la firma debe ser incorrecta
Contraseña válida de la firma	El actor introduce los datos e introduce bien la contraseña de la firma digital	Debe salir el mensaje "La Incidencia fue exitosamente creada."	El sistema muestra mensaje satisfactoriamente	La contraseña debe ser correcta

Crear nuevo usuario				
Escenario	Entrada	Resultados Esperados	Resultados Obtenidos	Condiciones
Campos vacíos	El actor no llena algún campo del formulario	Debe salir el mensaje de error de campo vacío	El sistema muestra el mensaje de error satisfactoriamente	El campo debe estar vacío
Campo Correo es incorrecto	El actor introduce incorrectamente la dirección de correo	Debe salir el mensaje de error "Inserte correctamente la dirección de correo"	El sistema muestra el mensaje de error satisfactoriamente	El campo Correo debe tener un valor incorrecto
Todos los campos fueron bien llenados	El actor llena los campos con valores correctos	Debe salir el mensaje "El usuario fue exitosamente creado"	El sistema muestra el mensaje satisfactoriamente	Los campos deben ser llenados correctamente
Datos duplicados	El actor llena los campos con información existente en la base de datos.	Debe salir el mensaje de error "El usuario ya existe, introduzca uno nuevo"	El sistema muestra el mensaje de error satisfactoriamente	El valor de los campos debe encontrarse en la base de datos

4.4 Análisis de los resultados esperados

Las pruebas anteriores se les realizaron a los casos de uso del sistema, definiéndose para cada uno, diferentes tipos de escenarios. Inicialmente fueron encontradas en la aplicación web, de 110 funcionalidades probadas, un total de 6 no conformidades, en la segunda iteración de pruebas se encontraron 3 no conformidades de 110 y ya en la tercera iteración fueron obtenidos resultados óptimos al no encontrar no conformidades. Con estas pruebas se demostró que el sistema gestor expedientes técnicos es funcional y cumple con los requisitos necesarios para ser desplegado. Posteriormente se muestran las no conformidades encontradas y sus soluciones para cada iteración.

Primera iteración

Capítulo 4: Implementación y Resultados Esperados

No conformidad	Solución
<ul style="list-style-type: none"> ✓ La aplicación no tiene sistema de autenticación. ✓ La aplicación no salva correctamente la documentación. ✓ La aplicación no posee gestión de usuario. ✓ La aplicación tiene las validaciones en el idioma inglés. ✓ La aplicación no separa correctamente los expedientes por direcciones. ✓ La aplicación posee error de persistencia a la hora de editar los campos. 	<ol style="list-style-type: none"> 1. Se creó el sistema de autenticación haciendo uso del LDAP y de las sesiones. 2. Se le dio los permisos de escritura a la carpeta GET. 3. Se agregó el modulo gestión de usuario a la aplicación. 4. Se tradujeron todos los valores de las variables contenidas en el archivo Bundle. 5. Se crea en el gestor de expediente una estructura que es capaz de separar las direcciones por carpetas y dentro de ellas se guardaran sus expedientes. 6. Se añadieron id de tipo serial a las tablas requeridas.
Segunda iteración	
No conformidad	Solución
<ul style="list-style-type: none"> ✓ La validación del IP no funciona correctamente, reconoce números superiores a 255. ✓ Cuando la aplicación no está conectada al LDAP no se puede autenticar ningún usuario. ✓ Los campos de texto y los Combo Box no poseen el mismo tamaño 	<ol style="list-style-type: none"> 1. Se corrigió el patrón para validar la dirección IP. 2. Se validó la aplicación para que, si no está funcionando el servicio LDAP, busque el usuario dentro de la tabla, si está autentica si no muestre un mensaje de error. 3. Esto sucede cuando la aplicación no es iniciada con el navegador Mozilla Firefox, se recomienda usar este navegador.
Tercera iteración	
No conformidad	Solución
No existe	No existe

Prueba de carga

Es una prueba que somete al sistema a una carga de trabajo concreta y estable durante un tiempo relativamente corto. El nivel de carga debe ser alto y continuo para verificar que el sistema soporta esa carga sin pérdidas de servicio y con un tiempo de respuesta estable y sin degradaciones. Se suelen hacer con varios niveles de carga (100%, 150% y 200%, por ejemplo) para conocer el comportamiento del sistema si la carga real llegara a esos niveles (Lingam 2004).

Prueba de rendimiento

Las pruebas de rendimiento consisten en someter al sistema a altas cargas de trabajo mediante la simulación de la actividad de los usuarios reales en el sistema. Esta simulación tendrá que ser lo más fidedigna posible para que los resultados sean relevantes y al analizar la información obtenida se alcancen conclusiones acertadas y precisas.

Capítulo 4: Implementación y Resultados Esperados

Para realizar las pruebas se utilizó la herramienta JMeter, la cual permite la realización de pruebas unitarias y soporta aserciones para asegurarse que los datos recibidos son correctos, por cookies de hilos, configuración de variables y una variedad de reportes.

Se realizaron un total de 3 pruebas de 50, 100 y 300 hilos cada una, las cuales simulan 50, 100 y 300 accesos de usuarios respectivamente. Se definió también el enlace de la lista de incidencia del gestor de expediente técnico al cual se simuló el acceso aleatorio y a partir de ahí, se recolectaron los datos necesarios para su interpretación.

✓ La primera prueba se realizó el día 13-6-2016 las 13:28 hs, se configuraron 50 hilos cada 1 s.

Los valores obtenidos por los componentes gráficos, se muestran a continuación:

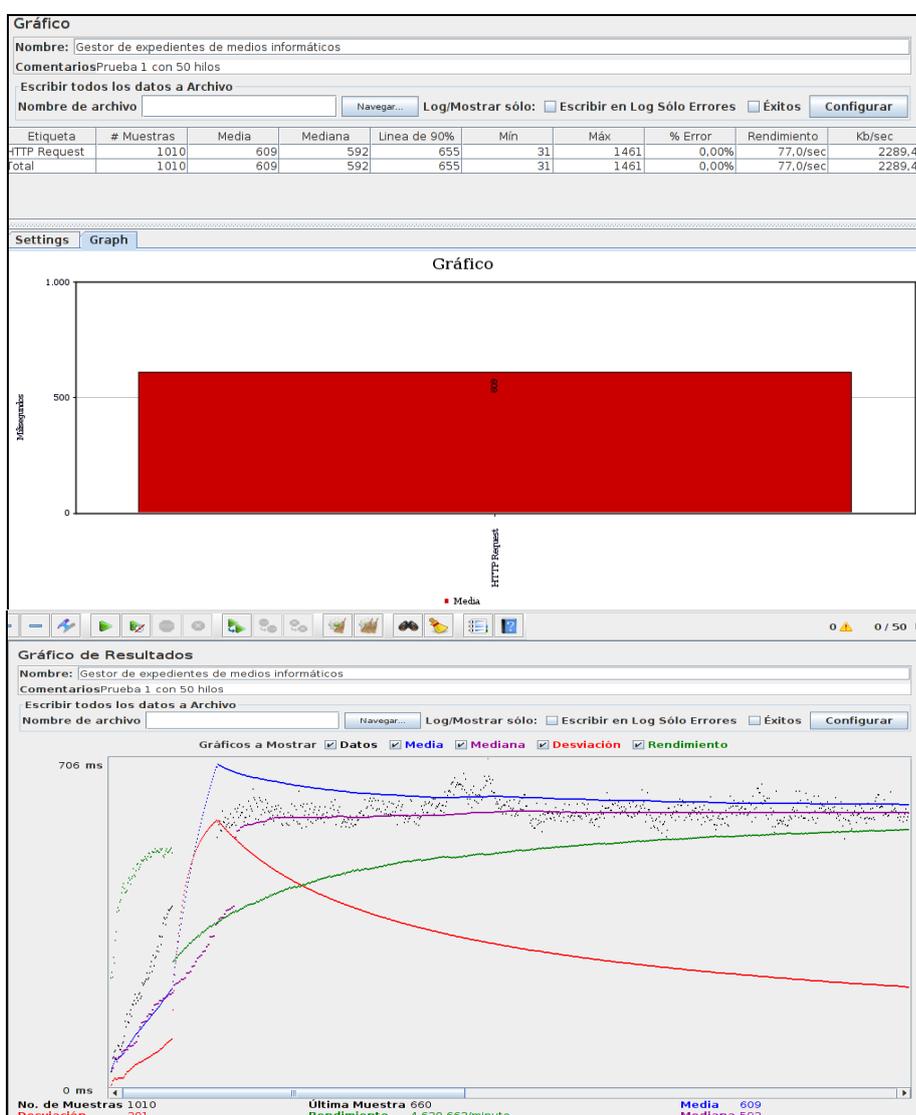


Fig. No.4.6 Resultados obtenidos en la primera prueba para 50 usuarios.

Como se muestra en las gráficas anteriores, el tiempo medio para acceder al listado de incidencias del gestor de expedientes técnicos de medios informáticos es de 0.609 s, realizándose un total de 1010 peticiones al servidor, con 50 hilos.

Capítulo 4: Implementación y Resultados Esperados

✓ La segunda prueba se realizó el día 13-6-2016 las 13:40 hs, se configuraron 100 hilos cada 1 s. Los valores obtenidos por los componentes gráficos, se muestran a continuación:

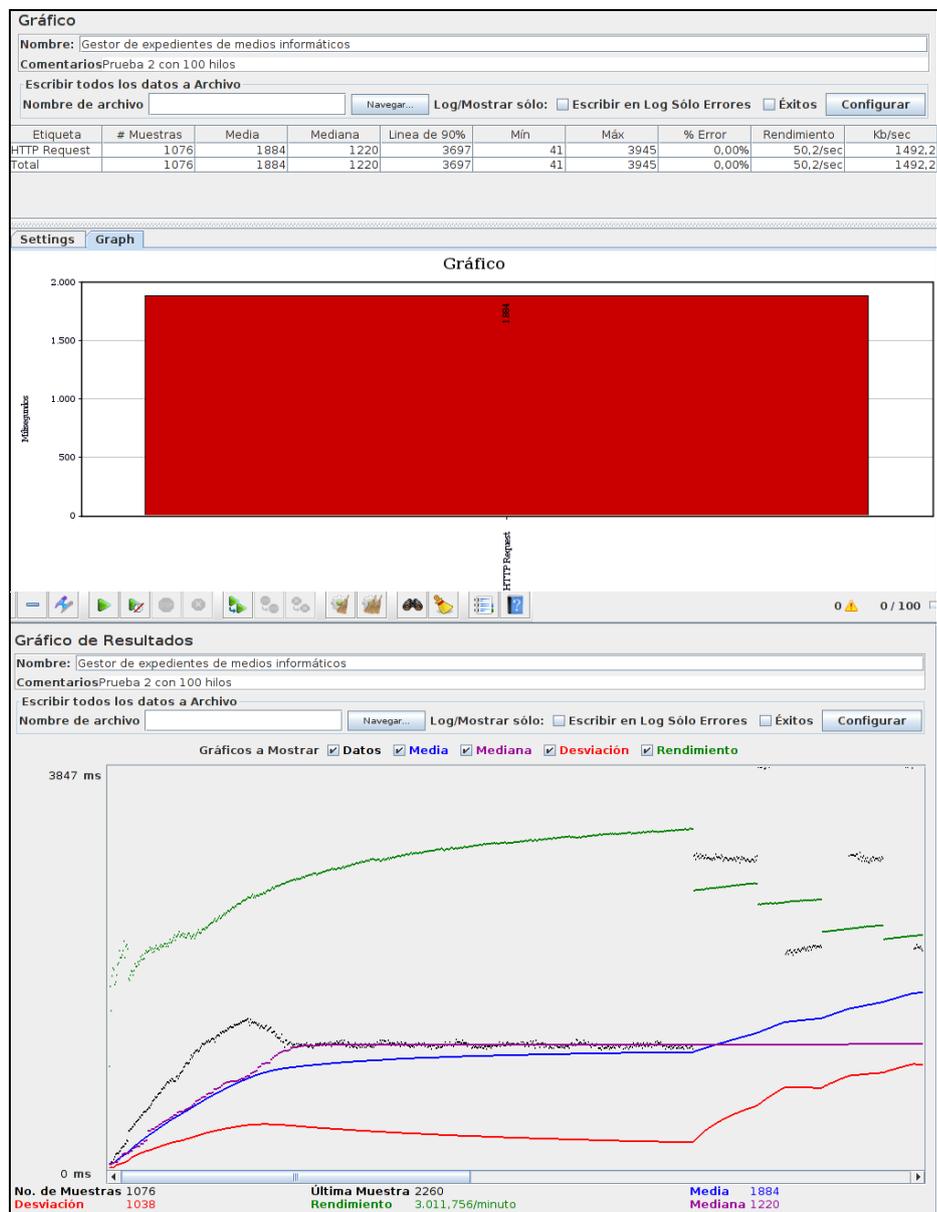


Fig. No.4.7 Resultados obtenidos en la segunda prueba para 100 usuarios.

En esta prueba, el tiempo medio para acceder al listado de incidencias del gestor de expedientes técnicos de medios informáticos es de 1.884 s, realizándose un total de 1076 peticiones al servidor, con 100 hilos.

✓ La tercera prueba se realizó el día 13-6-2016 las 14:05 hs, se configuraron 300 hilos cada 1 s. Los valores obtenidos por los componentes gráficos, se muestran a continuación.

Capítulo 4: Implementación y Resultados Esperados

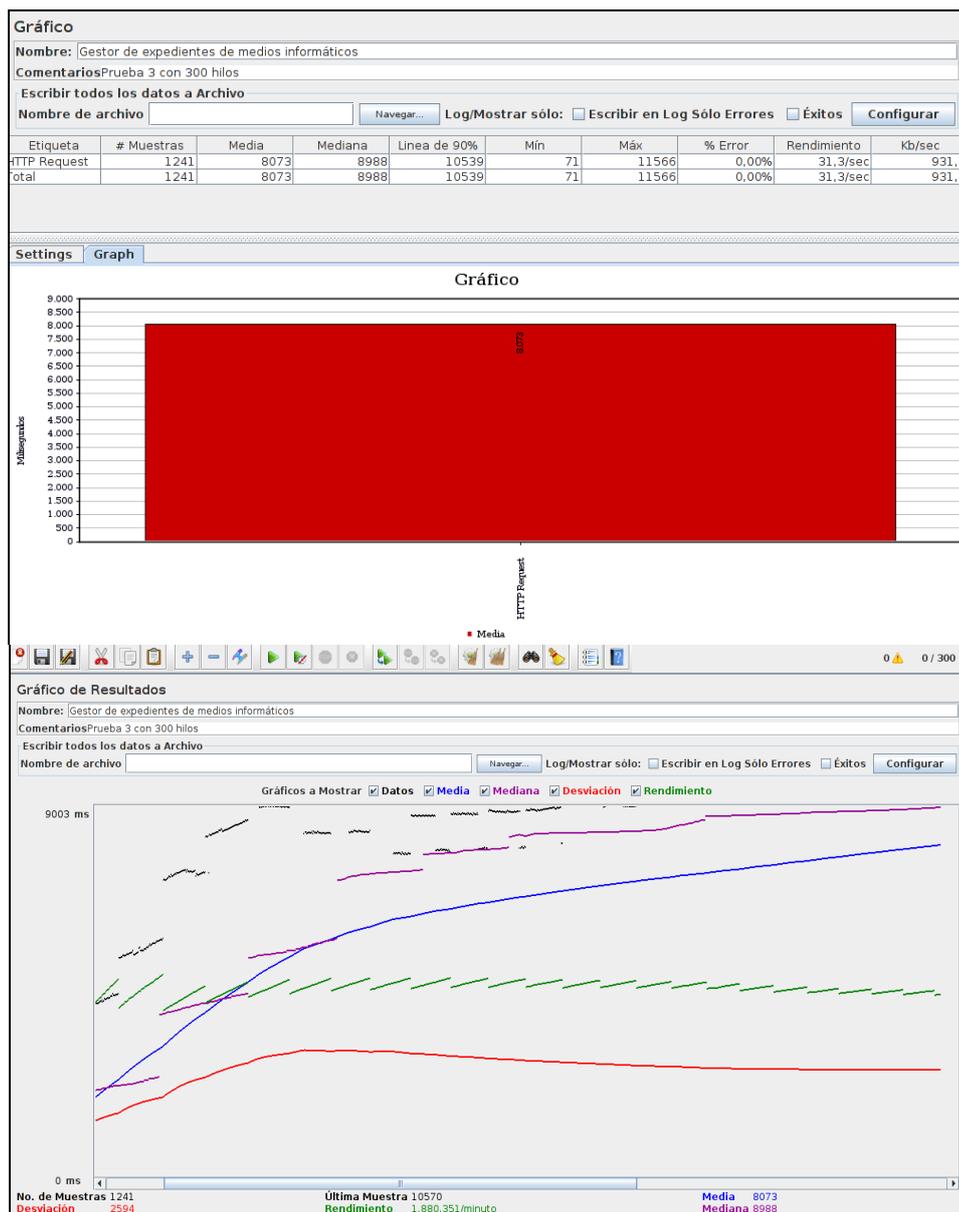


Fig. No.4.8 Resultados obtenidos en la tercera prueba para 300 usuarios.

En esta prueba, el tiempo medio para acceder al listado de incidencias del gestor de expedientes técnicos de medios informáticos es de 8.073 s, realizándose un total de 1241 peticiones al servidor, con 300 hilos.

Conclusiones Parciales

- ✓ Se establecieron escenarios de prueba a la interfaz de usuario con el objetivo de comprobar el correcto funcionamiento de la aplicación.
- ✓ El sistema puede ser desplegado satisfactoriamente, ya que no se encontraron no conformidades en la última iteración realizada.
- ✓ Se utilizaron los casos de pruebas para lograr que el comportamiento del software fuera más fiable y eficiente.

Conclusiones

CONCLUSIONES

Una vez finalizado el proceso de desarrollo se ha podido cumplir a cabalidad con los objetivos propuestos inicialmente de construir una herramienta para la gestión de los expedientes técnicos en los laboratorios farmacéuticos AICA, permitiendo así que:

- ✓ El control y la gestión de la información se realice con la calidad que exige este proceso para contribuir a brindar un servicio de excelencia.
- ✓ El estudio efectuado acerca de las principales metodologías de desarrollo de software existentes y las ventajas de cada una permitió concluir que Open Up es la metodología que más se ajusta a las necesidades del proyecto debido a su gran adaptabilidad.
- ✓ El sistema construido cumple con la especificación de requerimientos iniciales, además cumple con los estándares propios de aplicaciones informáticas como son (interfaces amigables e intuitivas, comprensibles, confiable, portable, robusto y escalable).
- ✓ Como novedad se tiene la implementación de la firma digital donde se ha logrado un nivel de seguridad satisfactorio en el ecosistema de PKI implementado.

Recomendaciones

RECOMENDACIONES

- ✓ Continuar investigando sobre los ERPs que existen en la actualidad para lograr adaptarlos al proceso de negocio que se realiza actualmente en la industria farmacéutica.
- ✓ Agregar nuevas funcionalidades al sistema, como es el caso de la disponibilidad tecnológica y sistemas de reporte para tener un mejor control de los medios informáticos.

REFERENCIAS BIBLIOGRÁFICAS

1. Agüero, I. G. (2012). Ecosistema para la Creación de Firma Digital Avanzada en Movilidad y Autenticación Mediante Elemento Seguro. Departamento de Ciencia de la Computación e Inteligencia Artificial, Facultad de Informática, Universidad del País Vasco: 60.
2. Badenes, R. O. (2012). Sistemas Integrados de Gestión Empresarial. Evolución Histórica y tendencias de futuro. C. Consejería de Educación, Tecnología. Valencia.
3. Bautista, R. C. (2010). Sistema de gestión de salas y de dispositivos de red basado en la tecnología de Java Server Faces. Departamento de Computación, Electrónica y Mecatrónica. Cholula, Puebla.
4. Booch, J. R. I. J. G. (2007). El Lenguaje Unificado de Modelado A.-. Wesley.
5. Cárdenas, A. H. G. S. A. (2003). Una Herramienta Cases para el Diseño y la Generación de la Estructura Estática de La Base de datos. Revista Investigación Operacional. Centro de Estudios de Ingeniería y Sistemas (CEIS). 24, No. 2: 11.
6. Cárdenas, E. R. (2008). Aproximación Jurídica a la Firma Digital y a los Prestadores de de Servicios de Certificación Digital en la Comunidad Andina de Naciones. F. U. E. d. I. C. d. C. d. Bogotá. Bogotá, Colombia.
7. Carnicero, P. (2011). La comunicación y la gestión de la información en las instituciones educativas. W. K. Educación.
8. Carrancá, H. W. G. (2014). Sistema Informático para el diagnóstico de la calidad de servicio del sistema de televisión de la UCI. Facultad 6. Havana, Universidad de las Ciencias Informáticas.
9. Castillo, R. P. (2011). "Gestión de reportes para la Plataforma PTARTV." 75.
10. Cataldi, Z. (2000). Metodología de Diseño, desarrollo y evaluación de software educativo, Facultad de Informática. UNLP. Ingeniera: 75.
11. Çivici (2014) "PRIMEFACES USER GUIDE 5.1."
12. Cohn, J. R. R. (2015). OCS Inventory. M. Book on Demand, Transmedia Holding.
13. Entonado, F. B. (2001). Sociedad de la información y la Educación. C. y. T. Consejería de Educación. Badajoz.
14. Fros, A. E. P. A. G. (2001). Problemas y Soluciones en la Implementación de Extreme Programming: 91.
15. Fuentes, V. D. Q. C. M. M. C. L. A. G. G. A. F. (2010). "CedruX. Solución para sistemas de control logístico." Convención Científica de Ingeniería y Arquitectura: 9.
16. Gimson, L. (2012). Metodologías ágiles y desarrollo basado en conocimiento. Facultad de Informática, Universidad Nacional de La Plata. Licenciada: 106.
17. Herrando, R. L. A. C. D. C. J. G. (2011). Introducción a la Gestión de Sistemas de Información en la empresa, Universidad Jaume I. Servei de Publicación y Comunicació.
18. Kruchten, P. (2009). The Rational Unified Process An Introduction.

Referencias Bibliográficas

19. Legizamon, J. G. (2014). "Sistema Gestor de Base de Datos ORACLE." from www.prezy.com.
20. Lingam, C. M. (2004). "Performance Testing and Tuning."
21. Marcos, M. P. A. d. M. E. (1999). Diseño de Bases de Datos Relacionales: 12.
22. Martínez, L. J. A. I. Z. (2008). Estructura de Datos en Java. J. L. García. Aravaca, Madrid.
23. O., J. L. A. (2010) "LOS CASOS DE PRUEBA EN LA PRUEBA DEL SOFTWARE." Revista Digital Lámpsakos No. 3, 8.
24. Perez, G. (2008). "Aprendiendo Java y Programación Orientada a Objetos."
25. Point, T. (2014) "JAVA PERSISTENCE API (JPA)." 113.
26. Quatrani, T. (2002). Visual Modeling with Rational Rose 2002 and UML.
27. Rodas, S. O. d. (2016). "Rodas XXI."
28. Rodríguez, A. (2013). "Lenguaje de Programación." from www.prezy.com.
29. Rodríguez, Y. O. (2013). Módulo de análisis de redes sociales para el Sistema de Vigilancia Tecnológica del grupo Gestión de la Información y el Conocimiento. Facultad 4. Havana, Cuba, Universidad de las Ciencias Informáticas: 80.
30. Sánchez, J. (2004) "MySQL Guía Rápida." 23.
31. Sandoval, G. R. (2011). Fundamentos de Pruebas de Software.
32. Santos, N. G. (2012). "Versat - Zarasola." from www.juventudrebelde.cu.
33. Sharp, B. L. R. (2011). Introducing HTML5. M. J. Nolan. Berkeley, CA 94710.
34. Sniper (2003). Manual de Programacion : Java, JSP y XML.
35. Tsang, C. H. K. (2005). Object-Oriented Technology. Pittsford, NY, Castle Rock.
36. Vaswani, V. (2010). Fundamentos de PHP. M. A. L. Ponce.
37. Yerovi, M. d. L. A. (2013). ESTUDIO DE PATRONES DE DISEÑO EN PLATAFORMA JAVA ENTERPRISE EDITION VERSIÓN 6 PARA EL DESARROLLO DE APLICACIONES WEB. FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS. Ibarra, Ecuador, UNIVERSIDAD TÉCNICA DEL NORTE: 224.

Anexos

ANEXOS

Anexo 1 Entrevista a especialista del grupo de informática de los laboratorios farmacéuticos AICA

Soy estudiante que cursa la carrera de Ingeniería en Ciencias Informáticas que se estudia en la UCI, realizo un proyecto de investigación en el que usted forma parte de la población analizada. Considero que tiene información importante para el desarrollo de este trabajo por lo que solicito su colaboración y garantizo el carácter confidencial de sus respuestas.

Gracias por adelantado.

DE NATA

Objetivo: Desarrollar una aplicación informática que permita contribuir a la gestión de los expedientes técnicos en los laboratorios farmacéuticos.

1. ¿Cómo se gestionan los expedientes técnicos en los laboratorios farmacéuticos en AICA?
 Formato Duro. Digital. Otros.
Especificar para otros.

2. Explique brevemente el proceso de gestión de los expedientes técnicos en los laboratorios farmacéuticos AICA.
Cuando se le hace algún cambio a un equipo informático se crea una nueva hoja del expediente, esta es firmada por el especialista y por los operarios responsable del equipo, y se añade en la hoja de incidencias el cambio que sufrió el equipo. Esta hoja de incidencias es firmada por el especialista y el operario.

3. De que forman se obtienen los datos de los equipos de cómputos para después gestionar la información recogida.
 Sistema Informático. Formato Duro. Otros. Especificar para otros.

4. Como se gestionan las incidencias ocurridas con los equipos de cómputos, explíquela brevemente.
Una vez ocurrido un hecho con el equipo el especialista describe lo ocurrido en una hoja de incidencias, esta hoja es archivada en el expediente y tiene que ser firmada por el especialista y el operario.

5. ¿Cómo se garantiza la veracidad de la información que se gestiona con respecto a los expedientes técnicos?
no siempre se puede saber si la información que se encuentra en los expedientes es verdadera ya que existen varios especialistas que gestionan los expedientes y a veces no lo actualizan cuando arreglan o dan mantenimiento a un equipo

6. ¿Cree usted que el proceso de gestión actual se realiza de una forma eficiente? En ambos casos justifique su respuesta.
no, ya que se acumula mucho papel, buscar la información que piden los superiores se hace muy difícil, no existe veracidad en la información, no se tiene, y los informes no quedan con la calidad requerida

7. Según su experiencia, ¿Cree que la fuente de información donde se encuentra recogida la información de los expedientes técnicos sea una fuente confiable?
 Si. No. Especifique para cualquiera de los dos casos.
ya que la fuente es el papel, esto implica pérdida de la documentación, trapapeleo, muchas veces se dejan los papeles a donde se llevan a firmar por los usuarios etc

Anexo 2 Guía de observación.

Soy estudiante que cursa la carrera de Ingeniería en Ciencias Informáticas que se estudia en la UCI, realizo un proyecto de investigación en el que usted forma parte de la población analizada. Considero que tiene información importante para el desarrollo de este trabajo por lo que solicito su colaboración para observar los principales elementos que se encuentran en el proceso de gestión de expedientes técnicos de los laboratorios farmacéuticos AICA.

Gracias por adelantado.

Objetivo: Conocer los elementos que se encuentran presente en el proceso de gestión y obtención de la información de los expedientes técnicos.

1. ¿Cómo se gestiona los expedientes técnicos en los laboratorios farmacéuticos en AICA?
 Formato Duro Digital Otros.
 Especificar para otros.

2. Explique brevemente el proceso de gestión de los expedientes técnicos en los laboratorios farmacéuticos AICA.
Se crea el expediente técnico de forma manual con los antecedentes del equipo, una vez creado este expediente por el especialista posteriormente la firma y análisis es hecho por el operador responsable del equipo. Al final a este expediente se encuentra el acta de responsabilidad y el trámite de incidencia el cual es hecho y firmado por el especialista y por el operador.

3. De que forman se obtienen los datos de los equipos de cómputos para después gestionar la información recogida.
 Sistema Informático Formato Duro Otros. Especificar para otros.

4. Como se gestionan las incidencias ocurridas con los equipos de cómputos, explique la brevemente.
Se recibe la incidencia del equipo y una vez terminada es firmada por el especialista y después por el operador, luego se archiva en el expediente técnico.

5. ¿Cómo se garantiza la veracidad de la información que se gestiona con respecto a los expedientes técnicos?
No se puede tener la certeza de que la información que se encuentra recogida en los expedientes técnicos sea verdadera debido a que en muchas de las ocasiones la información no se actualiza en el momento requerido.

6. ¿Cree usted que el proceso de gestión actual se realiza de una forma eficiente? En ambos casos justifique su respuesta.
No, debido a que el proceso se realiza de forma manual trayendo esto como consecuencia que en muchas ocasiones existen errores de escritura y que la gestión de la información de los expedientes se ralentice.

7. Según su experiencia, ¿Cree que la fuente de información donde se encuentra recogida la información de los expedientes técnicos sea una fuente confiable?
 Si No Especifique para cualquiera de los dos casos.
La documentación generada por los expedientes técnicos es abundante y al estar recogida en papeles, el proceso de la búsqueda de esta información se hace lento y bastante complejo mantener organización y control de la información.