

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 6



TÍTULO: SISTEMA DE GESTIÓN PARA LA INGENIERIA CLÍNICA Y
ELECTROMEDICINA: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL MÓDULO
PARA LA GESTIÓN DE PIEZAS DE REPUESTO.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS.

Autor: Milton Fuentes Beltrán.

Tutores: Ing. Dennys J. Hernández Peña.

Ing. Asnay Guirola González.

Ing. Reynier García Vistorte.

Ing. Daulemys Rigo Portillo.

Ciudad de la Habana, Cuba

“Año 50 de la Revolución”

"Si se siembra una semilla con fe y se cuida con perseverancia, solo será cuestión de recoger sus frutos."

Tomás Carlyle.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

Milton Fuentes Beltrán (Autor)

Ing. Dennys J. Hernández Peña (Tutor)

Ing. Asnay Guirola González (Tutor)

Ing. Reynier García Vistorte (Tutor)

Ing. Daulemys Rigo Portillo (Tutor)

DATOS DE CONTACTOS

Nombre: Dennys Javier Hernández Peña

Título universitario: Ingeniero Informático

Año de graduado: 2005

Correo: dhernandezp@uci.cu

Nombre: Asnay Guirola González

Título universitario: Ingeniero en Ciencias Informáticas

Año de graduado: 2007

Correo: aguirola@uci.cu

Nombre: Daulemys Rigo Portillo

Título universitario: Ingeniero en Ciencias Informáticas

Año de graduado: 2007

Correo: drigo@uci.cu

Nombre: Reynier García Vistorte

Título universitario: Ingeniero Industrial

Año de graduado: 2005

Correo: reyniergv@uci.cu

Agradecimientos

A mi familia por haber estado siempre a mi lado y brindarme apoyo cuando lo he necesitado.

A las amistades que diariamente se han preocupado por mi y que han compartido conmigo los momentos difíciles que han pasado.

A los compañeros de proyecto con los que he compartido este último período y que me han ayudado.

A mis tutores por haber sabido guiarme en el trabajo cuando se ha hecho difícil y por la seriedad y disposición con que asumieron la tarea.

A la persona que aportó una parte significativa de este trabajo y que permitió en gran medida obtener este resultado aun sin poder estar presente en la última parte.

Gracias sinceras a todos.

Dedicatoria

Dedico este trabajo a mi mamá la persona más importante en mi vida, que ha sabido entenderme, que me ha transmitido su espíritu y fuerza de voluntad, que ha formado mi carácter y me ha enseñado a salir adelante.

RESUMEN

El Sistema Nacional de Electromedicina (SNE) es el encargado de garantizar la sostenibilidad de la tecnología médica mediante el correcto asesoramiento de las adquisiciones de equipos y piezas, y la prestación de servicios técnicos de excelencia durante el tiempo de explotación de cada tecnología.

Para contribuir con tal misión se desarrolla el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM). Dicho sistema en su versión inicial permite llevar el control de los equipos médicos del país y gestionar los servicios realizados sobre los mismos.

Con el desarrollo de este trabajo se permite gestionar las piezas de repuesto necesarias para el correcto funcionamiento de los equipos, para ello se gestionan los tipos de piezas para cada uno de sus modelos , se controlan las piezas existentes de cada tipo, se gestionan las solicitudes de las misma así como los pedidos, despachos y movimientos que se realicen.

Con el desarrollo de este trabajo se le agrega una funcionalidad al sistema que es esencial para el SNE.

Palabras claves:

Equipo

Pieza

Gestión

Repuesto

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 TENDENCIAS ACTUALES.	5
1.2 ANTECEDENTES Y SISTEMAS EXISTENTES.....	9
1.3 PROCESO DE GESTIÓN DE LAS PIEZAS DE REPUESTO EN EL SNE	10
1.4 FUNDAMENTACIÓN DE LA METODOLOGÍA EMPLEADA.....	10
1.5 TECNOLOGÍAS UTILIZADAS PARA EL DESARROLLO DEL SISTEMA.	14
1.6 ARQUITECTURA DE SOFTWARE EMPLEADA EN EL SNS.....	19
1.7 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DEL SISTEMA.....	19
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	21
2.1 OBJETO DE ESTUDIO.....	21
2.2 OBJETO DE AUTOMATIZACIÓN.	22
2.3 INFORMACIÓN QUE SE MANEJA.	22
2.4 PROPUESTA DEL SISTEMA.	23
2.5 MODELO DEL NEGOCIO.....	24
2.6 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	30
2.7 DEFINICIÓN DE LOS CASOS DE USO.	33
2.8 DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.	35
CAPÍTULO 3: ANALISIS Y DISEÑO DEL SISTEMA	66
3.1 DIAGRAMAS DE CLASES.	66
3.2 DIAGRAMAS DE INTERACCIÓN DEL DISEÑO.....	70
3.3 DESCRIPCIÓN DE LAS CLASES.....	73
3.4 PATRONES DE DISEÑO EMPLEADOS.....	82
3.5 ARQUITECTURA EMPLEADA.....	83
3.6 DISEÑO DE LA BASE DE DATOS.	84
3.7 DIAGRAMA DE DESPLIEGUE.	85
3.8 DEFINICIONES DE DISEÑO QUE SE APLIQUEN.....	86
3.9 TRATAMIENTO DE ERRORES.	87
3.10 SEGURIDAD.....	87
CAPÍTULO 4: IMPLEMENTACIÓN	88
4.1. MODELOS DE COMPONENTES.	88
CONCLUSIONES	95
RECOMENDACIONES	96
REFERENCIA BIBLIOGRÁFICA	97
BIBLIOGRAFÍA	99
GLOSARIO DE TÉRMINOS	101

ÍNDICE DE FIGURAS

Figura 1 Fases e Iteraciones de la Metodología RUP	12
Figura 2 Diagrama de casos de uso del negocio.....	25
Figura 3 Diagrama de actividad del caso de uso del negocio solicitar piezas.	28
Figura 4 Diagrama de actividades del caso de uso del negocio recibir piezas....	29
Figura 5 Diagramas de clases del modelo de objetos.....	29
Figura 6 Diagrama de actores del sistema.....	34
Figura 7 Diagrama de casos de uso del sistema.	35
Figura 8 Diagramas de clases caso de uso gestionar importadores.	66
Figura 9 Diagramas de clases caso de uso gestionar unidades de medida.....	67
Figura 10 Diagrama de clases caso de uso gestionar tipos de piezas.	68
Figura 11 Diagrama de clases caso de uso gestionar piezas.....	69
Figura 12 Diagrama de secuencia caso de uso gestionar importadores.....	70
Figura 13 Diagrama de secuencia del caso de uso gestionar unidad de medida.	71
Figura 14 Diagrama de secuencia del caso de uso gestionar tipo de piezas, escenario: adicionar tipo de pieza.....	72
Figura 15 Ejemplo del Patrón Experto.....	82
Figura 16 Modelo de datos.....	84
Figura 17 Distribución física del sistema.	85
Figura 18 Diagrama de despliegue	86
Figura 19 Modelo de componentes caso de uso gestionar tipos de piezas	88
Figura 20 Modelo de componentes caso de uso gestionar piezas	89
Figura 21 Modelo de componentes caso de uso gestionar movimientos de piezas	90
Figura 22 Modelo de componentes paquete crear_movimineto_piezas.....	91
Figura 23 Modelo de componentes visualizar_moviminetos_creados	92
Figura 24 Modelo de componentes visualizar_moviminetos_asignados	93
Figura 25 Modelo de componentes editar_moviminetos	94

ÍNDICE DE TABLAS

Tabla 1 Justificación de los actores del negocio.	24
Tabla 2 Justificación de los trabajadores del negocio.....	25
Tabla 3 Descripción textual del caso de uso del negocio solicitar piezas.	25
Tabla 4 Descripción textual del caso de uso del negocio recibir piezas.	26
Tabla 5 Justificación de los actores del sistema.....	33
Tabla 6 Descripción del caso de uso autenticarse.....	35
Tabla 7 Descripción del caso de uso gestionar unidad medida.	36
Tabla 8 Descripción del caso de uso gestionar importador.	38
Tabla 9 Descripción del caso de uso gestionar solicitudes de piezas.....	41
Tabla 10 Descripción del caso de uso gestionar despachos de piezas.....	44
Tabla 11 Descripción del caso de uso gestionar tipos de piezas.....	47
Tabla 12 Descripción del caso de uso gestionar piezas.	50
Tabla 13 Descripción del caso de uso actualizar inventario de piezas del almacén.	54
Tabla 14 Descripción del caso de uso gestionar pedidos de piezas.	57
Tabla 15 Descripción del caso de uso gestionar movimientos de piezas.	61
Tabla 16 Descripción de la clase CI_adicionar_tipo_pieza	73
Tabla 17 Descripción de la clase CI_adicionar_pieza	74
Tabla 18 Descripción de la clase CCadicionar_tipo_pieza	75
Tabla 19 Descripción de la clase CCadicionar_pieza.....	75
Tabla 20 Descripción de la clase area_salud	76
Tabla 21 Descripción de la clase baja	76
Tabla 22 Descripción de la clase denominación	76
Tabla 23 Descripción de la clase despacho.....	77
Tabla 24 Descripción de la clase equipo.....	77
Tabla 25 Descripción de la clase especialidad	78
Tabla 26 Descripción de la clase especialista	78
Tabla 27 Descripción de la clase importador.....	80
Tabla 28 Descripción de la clase movimiento	80
Tabla 29 Descripción de la clase pedido.....	80
Tabla 30 Descripción de la clase pieza	81

INTRODUCCIÓN

Tras el triunfo revolucionario, Cuba se propuso un camino de desarrollo que pudiera solucionar por igual las necesidades materiales básicas y las espirituales de su población, sobre la base de una distribución más justa y equitativa de la riqueza. **(1)**

El 1 de agosto de 1961 se crea el Ministerio de Salud Pública (MINSAP) y se planifica la construcción de nuevos Hospitales, Policlínicos y Pequeñas unidades de salud que se dotaron con las nuevas tecnologías. Para lograr el óptimo funcionamiento de los equipos médicos era necesario formar talleres especializados en el mantenimiento de los mismos. Para la fecha el país contaba con pocos profesionales en el sector, estos trabajaron en pequeñas fábricas privadas que producían equipos médicos antes de 1959, a los que el estado reunió para formar los primeros talleres que ya para el año 62 prestaban algunos servicios.

A partir de 1980, Cuba por una idea planteada por Fidel Castro Ruz propone convertirse en una potencia médica, para ello se introdujo un inmenso volumen de tecnología. Hasta ese momento los talleres se encontraban dentro de las unidades de salud en pequeños locales, por lo que se definió un nuevo programa para la Electromedicina: se creó un Sistema Nacional de Electromedicina (SNE) que incluyó, la creación de una Dirección Nacional y talleres especializados.

En 1983 comenzó la construcción de los Centros Provinciales de Electromedicina (CPE) y Organizaciones Básicas de Electromedicina (OBE) y un centro Nacional. Por lo que Electromedicina se proyectó con una nueva misión: garantizar la sostenibilidad de la tecnología médica mediante el correcto asesoramiento de las adquisiciones de equipos y piezas, y la prestación de servicios técnicos de excelencia durante el tiempo de explotación de cada tecnología.

Con el desarrollo de las tecnologías en el campo de la informática y las comunicaciones, el MINSAP impulsado por la dirección del país se ha propuesto informatizar su sector para facilitar el intercambio de los especialistas del país y foráneos por medio de redes digitales. El 28 de junio de 2006 Electromedicina formó el Grupo de Automatización y Desarrollo de Electromedicina (GADE) con la misión de: Apoyar la Gestión Informativa, Investigativa y Administrativa del SNE, utilizando las herramientas que ofrecen las tecnologías de la informática y de telecomunicaciones. Ello se logra mediante el diseño,

desarrollo y la prestación de servicios de informática a toda la red de Electromedicina, fomentando su aprovechamiento adecuado y velando por su correcta implementación contribuyendo así al desarrollo sostenible de la política del MINSAP en materia de informatización.

En noviembre de 2006, comenzó el trabajo de diseño y programación de un sistema automatizado que permitiría realizar las gestiones específicas de las actividades de Electromedicina e integrar las informaciones de carácter general del sistema nacional de salud, al cual se le denominó: Sistema de Gestión para Ingeniería Clínica Electromedicina (SIGEM) con prestaciones que van desde el Aseguramiento Técnico, el Inventario hasta la Gestión de la Estadística Técnica. Para el desarrollo y la implementación del sistema participan diferentes entidades tales como: ETECSA, COPEXTEL, SEGURIMATICA y la Universidad de las Ciencias Informáticas (UCI).

Al observar las particularidades del SNE se pudo detectar que el sistema de gestión desarrollado no permite gestionar las piezas de repuesto, por ello las personas encargadas de este proceso presentan algunos problemas que son generados por el hecho de que este se realiza actualmente de forma manual, generando los problemas siguientes:

- Las personas encargadas de la realización de la misma no conocen la disponibilidad de piezas que existe.
- Las solicitudes y los pedidos de piezas son imprecisos al no usar correctamente los nomencladores (nombres) de las piezas.
- La entrega de un parte determinado pedido por la empresa a estas personas tiende a demorarse, por ser un proceso lento.
- Esto provoca que halla piezas que por el tiempo que pasan en los almacenes se rompan sin ser usadas.
- Como resultado de este engorroso proceso hay equipos que no se pueden reparar en tiempo.

Es por esto que se plantea como **problema científico**:

¿Cómo contribuir a la gestión de la información de las Piezas de Repuesto en Electromedicina?

Objeto de estudio:

Proceso de gestión de la información.

Campo de acción:

Proceso de Gestión de la información de las piezas de Repuesto en Electromedicina.

Objetivo general:

Desarrollar el análisis, diseño e implementación de un módulo para la gestión de las piezas de repuesto en Electromedicina.

Objetivos específicos:

- Definir funcionalidades del módulo para la gestión de la información de las piezas de repuesto.
- Diseñar un módulo para la gestión de la información de las piezas de repuesto.
- Diseñar una base de datos para almacenar información de las piezas de repuesto.
- Implementar un módulo para la gestión de la información de las piezas de repuesto.

Para asegurar el cumplimiento de los objetivos trazados se proponen las siguientes **tareas:**

- Análisis del funcionamiento del CNE.
- Realización de entrevistas al personal del CNE.
- Definición del conjunto de casos de uso que determinan las funcionalidades del sistema.
- Diseño de los prototipos de interfaz.
- Especificación de los casos de uso definidos.
- Definición del conjunto de clases de interfaz, de control y de entidad.
- Determinación de las clases de diseño.
- Realización de diagramas de clases del diseño.
- Determinación de los componentes del sistema.
- Realización de los diagramas de componentes.
- Realización del diagrama de despliegue.

El presente documento cuenta con un Resumen, varios capítulos que constituyen el cuerpo del trabajo, Conclusiones, Recomendaciones, Bibliografía y Referencias bibliográficas. Los capítulos son:

Capítulo 1: Fundamentación teórica.

En este capítulo abordaremos temas de interés como son: el proceso de gestión en las empresas, sistemas de gestión, abordando además los ERP como sistemas avanzados en el mundo y en Cuba. Brindaremos una panorámica relacionado al proceso actual de control y gestión de las piezas de repuesto en Electromedicina. Se establecen las

técnicas, tecnologías y metodologías que se utilizan para la dar solución al problema planteado.

Capítulo 2: Modelo del negocio.

En este capítulo se expone una definición del objeto de estudio del problema, se plantean los objetivos estratégicos de la organización y procesos de negocio que los soportan. Se realiza una descripción de los procesos que serán objeto de automatización así como una descripción general de la propuesta de sistema, como debe funcionar. Definición del Modelo de negocio. Especificación de los requisitos del sistema y casos de uso existentes.

Capítulo 3. Análisis y diseño del sistema.

Este capítulo abarca el desarrollo de la fase de análisis y diseño del sistema, englobando todo lo referente al modelado del diseño específicamente. En el mismo figuran los diagramas de clases del diseño de los casos de uso definidos en el capítulo anterior, así como los diagramas de secuencia correspondientes a dichos casos de uso. Además se representa el modelo de datos con la descripción de sus tablas, el diagrama de clases persistentes y el modelo de despliegue del sistema.

Capítulo 4. Implementación.

Este capítulo abarca detalles de la fase de elaboración para la implementación del sistema. Se realiza el modelo de implementación con sus respectivos diagramas de componentes pertenecientes a los casos de uso definidos en las iteraciones anteriores.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Introducción

En este capítulo abordaremos temas de interés como son: las tendencias actuales, en el mundo, en Cuba y en el Sistema Nacional de Salud; los antecedentes y sistemas existentes; la situación actual en el proceso de gestión de las piezas de repuesto en el Sistema Nacional de Electromedicina, la metodología y las tecnologías empleadas para el desarrollo del sistema, las herramientas utilizadas y la plataforma en la que se desarrolla

1.1 Tendencias actuales.

Principios de gestión.

La gestión tiene sus inicios en la antigüedad, el hombre se ha acostumbrado a vivir en sociedad por lo que tuvo que crear un orden económico, social y político para satisfacer sus necesidades. Por lo que tuvo que desarrollar métodos que le permitieran una mejor adquisición, asignación de recursos físicos y esfuerzos humanos para alcanzar ciertas metas. Surge así la gestión que facilita la satisfacción de tales necesidades a las personas.

Los principios de gestión que existían en la antigüedad han evolucionado y se han tornado indispensables para cualquier actividad humana. Cada día la sociedad enfrenta cambios más frecuentes y complejos que demandan una mayor eficiencia y optimización de los procesos. La filosofía de la gestión vincula los aspectos relativos al establecimiento de metas y objetivos con los aspectos relativos a la coordinación para alcanzar vías efectivas en la que la organización pueda manejar sus asuntos.

Proceso de Gestión

“Para dirigir y hacer funcionar una organización satisfactoriamente requiere gestionar de una manera sistemática y visible. El éxito puede derivarse de implementar y mantener un sistema de gestión que sea diseñado para mejorar continuamente las prestaciones por medio de identificar las necesidades de todas las partes interesadas. Al gestionar una organización incluye también gestionar la calidad, entre otras disciplinas de gestión.” (2)

Toda actividad de gestión exige un orden, un método y debe ser concebida y ejecutada a partir de determinada lógica. Aun aquellas actividades que supuestamente se desarrollan en forma espontánea, responden a determinados mecanismos que se han fijado con anticipación, por lo que en cierto sentido pasan igualmente por este proceso. Muchas veces estas actividades lógicas forman parte de un ciclo, que al repetirse en función de un mejoramiento, pasan en su siguiente ejecución a un nivel superior, respondiendo a la llamada espiral de desarrollo.

Dentro de este proceso, se mueven diferentes componentes conocidos como funciones de la gestión estas son:

Planificar: es el proceso de establecer objetivos con el fin de alcanzar determinados resultados, así como, identificar las acciones necesarias para alcanzarlos. Dentro de este concepto se contemplan un conjunto de decisiones o una selección de alternativas para el logro de tales resultados. Esta función se registra en diversos tipos de documentos: planes, programas, pronósticos y políticas.

Organizar: es el proceso de dividir el trabajo a realizar y de coordinar el logro de resultados que tienen un propósito común. Organizar es la química de la organización donde se mezclan todos los elementos que interactúan entre sí a fin de obtener los resultados esperados. Es el acto de combinar habilidades, posibilidades técnicas, experiencias, recursos y todos los elementos que podrían convertirse en resultados.

Dirigir: es el proceso de conducir y coordinar los esfuerzos laborales de las personas que integran una organización, ayudándolos a desarrollar tareas relevantes dentro de ella. La dirección es la función mediante la cual se ponen en marcha las actividades programadas. Comprende el compromiso de alcanzar un objetivo mediante el liderazgo de un grupo. La dirección ejerce una influencia notable en las personas para que trabajen voluntaria y entusiastamente para el logro de las metas colectivas de equipos y de la organización en su conjunto.

Controlar: es el proceso de supervisar las actividades y resultados, comparándolos con los objetivos y tomando las acciones correctivas, si son necesarias. Para ello se compara el desempeño con metas y planes, se muestran las desviaciones y al emprender medidas para corregir las desviaciones, se ayuda a asegurar el logro de los planes. Esta función comprende el establecimiento de normas de desempeño como base para la medida de los resultados, investigación, análisis, diseño,

implantación y operación de los sistemas de información, registros contables y estadísticos, auditorías, inspecciones, controles y otros métodos de verificación directa.

Gestión de la Información.

En el libro Gestión de información de las organizaciones la profesora Ponjuán hace mención de lo dicho por Woodman sobre la gestión de información:

“Es todo lo relacionado con la obtención de la información adecuada, en la forma correcta, para la persona indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta.” (3)

Los Objetivos de la gestión de información son:

- Maximizar el valor y los beneficios derivados del uso de la información.
- Minimizar el costo de adquisición, procesamiento y uso de la información
- Determinar responsabilidades para el uso efectivo, eficiente y económico de la información
- Asegurar un suministro continuo de la información.

En el libro Gestión de información de las organizaciones la profesora Ponjuán hace mención de lo dicho por, Schneyman en 1985 sobre este tema.

“...es la gestión (planeamiento, organización, operaciones y control) de los recursos (humanos y físicos) que tienen que ver con el apoyo a sistemas (desarrollo, mejoría y mantenimiento) y servicios (procesamiento, transformación, distribución, almacenamiento y recuperación) de la información (datos, textos, voz e imagen) para una empresa.” (4)

Ante situaciones de esta naturaleza, la gestión de la información se convierte en un aspecto estratégico para las organizaciones que se insertan en el actual entorno y asumen las nuevas tecnologías de información y comunicación. Muchas organizaciones recurren a la implementación de coherentes sistemas de gestión de información, espacios e infraestructuras para disponer de su propia información, compartir sus recursos y poseer canales de comunicación rápidos y eficientes, que colaboren con el desarrollo del trabajo y la toma de decisiones. Esto junto al desarrollo acelerado de la tecnología, proponen novedosas soluciones para potenciar valores a los denominados recursos intangibles, mejorar estrategias de administración y elevar niveles de eficiencia y eficacia.

Las personas juegan determinados roles y ocupan determinadas posiciones en su interacción con los recursos de información de las organizaciones como es el caso del usuario, del procesador, del suministrador y del gerente de información.

1.1.1. Tendencias en el mundo.

En el mundo moderno el desarrollo de las tecnologías de la información y las comunicaciones se ha evidenciado en todo los sectores.

Las empresas, independientemente de su tamaño, enfrentan demandas respecto a rentabilidad, calidad, tecnología y desarrollo sustentable. Un sistema de gestión eficiente, diseñado a la medida de sus procesos comerciales, puede ayudarle a centrar, organizar y sistematizar los procesos para la gestión y mejora de sus servicios. Ejemplo de sistemas de gestión se encuentran los ERP.

Los ERP (Enterprise Resource Planning).

Enterprise Resource Planning es una herramienta de estrategia, que equipa a la empresa con las capacidades y recursos necesarios para integrar y sincronizar las funciones aisladas en un proceso continuo de negocios en miras de ganar una competitividad en el turbulento ambiente de los negocios. (5)

El ERP es un sistema integral de gestión empresarial que está diseñado para modelar y automatizar la mayoría de procesos en la empresa (área de finanzas, comercial, logística, producción, etc.). Su misión es facilitar la planificación de todos los recursos de la empresa.

La importancia del impacto del ERP en los procesos cotidianos de la organización y la inversión que la misma debe hacer en términos económicos, hacen que el proceso de selección de la herramienta sea un tema delicado. Se debe tener en cuenta también que no es una tarea que se haga frecuentemente y que se espera un determinado retorno de la inversión en términos monetarios y de tiempo de uso. (6)

Se ha extendido de manera notable la implantación y uso de software ERP, no solo entre las empresas multinacionales necesitadas de un sistema complejo y homogéneo para los diversos países, sino también entre las empresas de todos los ámbitos sin necesidad de consolidación informativa y operativa. Pero como suele ocurrir también aquí puede decirse que no existe una solución única que sea óptima para todo tipo de empresas.

1.1.2. Tendencias en Cuba

En el mundo no existe ningún ERP OpenSource y que se adapte a la estructura de

Cuba que hasta la fecha no ha desarrollado ninguno propio, actualmente se trabaja en la Universidad de las Ciencias Informáticas en la implementación de un ERP cubano. En Cuba aun no se ha desarrollado ningún ERP, pero si existen empresas que utilizan sistemas que les facilitan algunos servicios, ejemplo de estos sistemas tenemos: Versat Sarasolo que es utilizado por TEICOS en Santa Clara que estos momentos esta siendo adoptado por distintas instituciones por su alta calidad como es el SNS, Rodas XXI desarrollado por CIMATEL, SISCONT desarrollado por Tecnomática en el MINBAS y Condor desarrollado para Cubalse.

1.1.3. Tendencias y tecnologías en el SNS.

Se ha definido por el MINSAP un grupo de premisas y requisitos que incorporan los últimos adelantos en el área de las Tecnologías de la Información y las Comunicaciones(TIC) y que garantizan la plataforma de integración de las aplicaciones, la compatibilidad y sostenibilidad de los productos a desarrollar, tales como: empleo de tecnologías basadas en Internet utilizando una plataforma LAMP (Linux, Apache, MySql, Php) documentación de todo el proceso productivo, requisitos de seguridad del software, independencia de la base de datos, desarrollo en multiplataforma y empleo de estándares internacionales para los productos relacionados con la salud. El soporte de infraestructura en todos los aspectos mencionados es la Red Telemática de la Salud Cubana (INFOMED).

Existe la empresa del Ministerio de la Informática y las Comunicaciones (MIC) encargada del desarrollo de tecnologías informáticas y de Programación para el SNS (SOFTTEL), la cual estableció la arquitectura de software, fija para todo el sistema nacional de salud.

1.2 Antecedentes y sistemas existentes.

En el mundo son varias las empresas que se dedican a producir sistemas ERP por su alta demanda ejemplo de esto es "DATAHOUSE COMPANY SOFTWARE Y SOLUCIONES EMPRESARIALES" (7) empresa que propone una serie de productos para las empresas: software ERP para la gestión comercial "Isis", Software MRP para la gestión Industrial, software para estaciones de servicios entre otros.

Otro ejemplo es la Empresa AdPyme Iberia cuya misión es proveer a la pequeña y mediana empresa de la tecnología y organización necesarias para competir al más alto nivel y a un coste razonable uno de sus productos es el erp"movex".

"El sistema de administración Duet hace posible un acceso homogéneo a los datos y procesos de negocios de SAP ERP y otras aplicaciones, a través de Microsoft Office, revolucionando así la manera en la que los trabajadores del área de la información

interactúan con las soluciones empresariales. El software de gestión "Duet" es el resultado de una colaboración sin antecedentes entre SAP y Microsoft, y constituye el primer producto conjunto que ha sido creado y soportado por los dos líderes de la industria". (8)

En el mundo existen muchos sistemas de gestión pero ninguno se adapta a las características requeridas para el SNE.

1.3 Proceso de gestión de las piezas de repuesto en el SNE

El Sistema Nacional de Electromedicina es la institución encargada de velar por el correcto funcionamiento de los equipos médicos del país, para ello cuenta con centros en todas las provincias y municipios a los que debe suministrar las piezas necesarias para el mantenimiento de los equipos que sufren roturas.

El SNE cuenta con servicio de correo y telefonía en la mayoría de sus centros, vías mediante se hace la solicitud de piezas para determinado mantenimiento, para este proceso se debe llevar el control de una serie de parámetros.

Los planificadores de los centros deben hacer los pedidos de piezas por trimestre brindando los datos correctamente.

Para hacer los movimientos de piezas es necesario controlar hacia que centro salió la pieza.

Para hacer las solicitudes se necesita además de los datos de la pieza que se solicite, el equipo que se va a atender.

Para los despachos es necesario controlar el equipo a al que se le pone la pieza.

1.4 Fundamentación de la metodología empleada

Los arquitectos de Software, deben tener un plano en que apoyarse. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo.

En este caso el proyecto definió la metodología a utilizar por existir una precedencia en el trabajo que se desarrolla, por lo que se asume la metodología RUP que es la que

anteriormente se utilizó.

Metodología RUP

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transición:** El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo:

- **Ingeniería de Negocios:** Entendiendo las necesidades del negocio.
- **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte:

- **Configuración y administración del cambio:** Guardando todas las versiones del proyecto.
- **Administrando el proyecto:** Administrando horarios y recursos.

- **Ambiente:** Administrando el ambiente de desarrollo.
- **Distribución:** Hacer todo lo necesario para la salida del proyecto

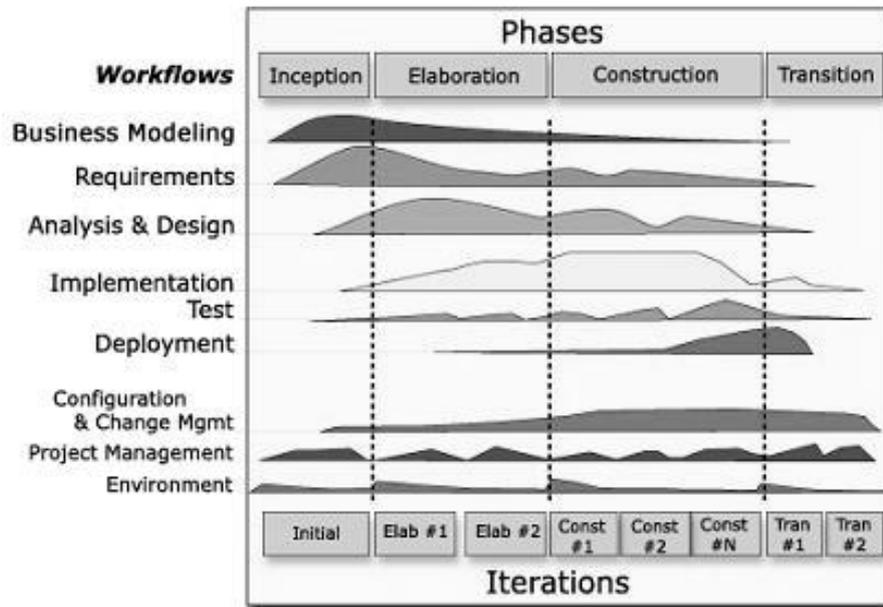


Figura 1 Fases e Iteraciones de la Metodología RUP

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos de RUP son:

- **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores:** Vienen hacer las personas o entes involucrados en cada proceso.
- **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Lenguaje de Unificado de Modelado

A mediados de los noventa existían muchos métodos de análisis y diseño OO lo que suponía que los mismos conceptos tenían distinta notación según el método de que se tratara. Ante esta situación de confusión, en 1994 Booch, Rumbaugh y Jacobson decidieron unificar sus métodos dando lugar a UML. Esta unificación fue promovida por el OMG de tal manera que UML se convirtió en la notación estándar para la descripción de métodos software. Según su definición, UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, desde una perspectiva OO. **(9)**

UML prescribe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí. Ahora con UML, diseñadores diferentes modelando sistemas diferentes pueden sobradamente entender cada uno los diseños de los otros.

UML tiene un uso limitado en otro tipo de cuestiones de programación. UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema. UML presenta los siguientes tipos de diagramas:

- **Diagrama de casos de uso que muestra a los actores** (otros usuarios del sistema), los casos de uso (las situaciones que se producen cuando utilizan el sistema) y sus relaciones.
- **Diagrama de clases** que muestra las clases y la relaciones entre ellas.
- **Diagrama de secuencia** muestra los objetos y sus múltiples relaciones entre ellos.
- **Diagrama de colaboración** que muestra objetos y sus relaciones, destacando los objetos que participan en el intercambio de mensajes.
- **Diagrama de estado muestra estados**, cambios de estado y eventos en un objeto o en parte del sistema.

- **Diagrama de actividad** que muestra actividades, así como los cambios de una a otra actividad junto con los eventos que ocurren en ciertas partes del sistema.
- **Diagrama de componentes** que muestra los componentes de mayor nivel de la programación.
- **Diagrama de implementación** que muestra las instancias de los componentes y sus relaciones.

1.5 Tecnologías utilizadas para el desarrollo del sistema.

Para el desarrollo del sistema es necesario utilizar diferentes tecnologías, las mismas fueron definidas por Softel para el MINSAP que incluye la utilización de la plataforma “LAMP” por sus siglas Linux, servidor Web Apache, MySQL y php y que al ser el sistema desarrollado para Electromedicina es asumida.

1.5.1. Tecnología del lado del cliente.

HTML (HyperText Markup Language)

Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces (hyperlinks)* que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones multimedia* (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape). **(10)**

CSS (Cascading Style Sheets)

Las hojas de estilo facilitan muchísimo la creación de páginas. Gracias a estas podemos definir un estilo específico para cada atributo sin tener que modificar las etiquetas dentro de la página HTML. Por ejemplo, se puede utilizar un solo archivo .CSS para manejar los estilos de todo un sitio. Las hojas de estilo se utilizan para dar estilo a documentos HTML y XML separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. Las hojas de estilo permiten a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier

cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

Java Script.

Es una tecnología del lado del cliente que es soportado tanto por el Internet Explorer de Microsoft, como por el Mozilla FireFox, lo que le confiere la característica de ser multiplataforma. Está diseñado para manejar la apariencia de la ventana en el navegador y manipular los eventos, para lo cual usa un conjunto de objetos.

- **Ventajas**

Fácil de aprender, rápida y potente: permite realizar ciertas funciones rápidas en una página Web solo con crear el código y cargarlo sin necesidad de crear una máquina virtual para compilar su código. Es un lenguaje de alto nivel, siendo capaz de aprovechar las propiedades de los exploradores Web incluso puede realizar algunas acciones sobre el sistema en que se está ejecutando.

Usabilidad: Java Script es uno de los lenguajes que más se utiliza en la Web, donde están publicadas millones de páginas que lo usan. Reducción de la carga del servidor: Java Script al contar con la habilidad de ejecutarse en el cliente y tener tantas funcionalidades, se ha podido ganar la atención de la mayoría de los desarrolladores Web pues ayuda a reducir la carga de trabajo del servidor, ejemplo de esto son las validaciones que posibilita. Con Java Script se evita ese proceso de mandar información de un lado a otro haciendo uso excesivo de la red ya que él valida antes de enviarse la información al servidor. **(11)**

1.5.2. Plataforma Lamp.

Linux

GNU/Linux implementa la mayor parte de las características que se encuentran en otras implementaciones de UNIX, más algunas otras que no son habituales. GNU/Linux es un sistema operativo completo con multitarea y multiusuario (como cualquier otra versión de UNIX). Esto significa que pueden trabajar varios usuarios simultáneamente en él, y que cada uno de ellos puede tener varios programas en

ejecución.

El sistema GNU/Linux es compatible con ciertos estándares de UNIX a nivel de código fuente, incluyendo el IEEE POSIX.1, System V y BSD. Fue desarrollado buscando la portabilidad de los archivos fuentes. Y esto es cierto, encontraremos que casi todo el software gratuito desarrollado para UNIX se compila en GNU/Linux sin problemas. Y todo lo que se hace para GNU/Linux (código del núcleo, drivers, librerías y programas de usuario) es de libre distribución. (12)

PHP

Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma. PHP es el acrónimo de Hypertext Preprocessor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

- **Ventajas adicionales de PHP**

Entre las ventajas podemos citar:

- Muy sencillo de aprender.
- Similar en sintaxis a C y a PERL
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.

- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son validos se hace en el servidor y no en el cliente (como se hace con JavaScript) de forma que se puede evitar que chequear que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía HTTP.
- PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.

Servidor de Base de Datos. MySQL.

MySQL es un servidor de base de datos que en los últimos años ha tenido un crecimiento vertiginoso. Es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL, esto significa que también todos pueden contribuir con ideas, elementos, mejoras o sugerir optimizaciones. Y así es que MySQL ha pasado de ser una pequeña base de datos a una completa herramienta. Su rápido desarrollo se debe en gran medida a la contribución de mucha gente al proyecto, así como la dedicación del equipo de MySQL.

- **Características de MySQL**

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

Servidor Web Apache.

El Servidor Web es el programa encargado de gestionar las peticiones de los usuarios que visitan su página Web. Es el programa encargado de mostrar sus páginas cuando un visitante realiza una petición en su dominio. Teniendo en cuenta que el 80% de las operaciones que realizara un usuario en el servidor es visionar paginas Web, la elección del programa encargado para tal cometido es clave para el buen funcionamiento general del servidor.

Apache es considerado el Servidor Web por excelencia, no solo por su gran aceptación, puesto que casi el 70% de los servidores Internet confía en el, sino porque desde su nacimiento ha demostrado con creces su estabilidad, solidez y su mayor rendimiento ante sus competidores.

Entre las ventajas de este podemos encontrar:

Fiabilidad: Alrededor del 90% de los servidores con más alta disponibilidad funcionan bajo servidores Apache.

Software Libre: Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.

Extensibilidad: se pueden añadir módulos para ampliar las ya de por si amplias capacidades de Apache. **(13)**

También dispone de una amplia variedad de módulos, que permiten desde generar contenido dinámico (con PHP, Java, Perl, Python,...), monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, crear servidores virtuales por IP o por nombre (varias direcciones Web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos módulos están disponibles junto con su código fuente, por lo cual pueden ser incluso modificado por cualquier persona con conocimientos de programación.

Como conclusión se puede ver que todo lo anteriormente expuesto no es más que una breve justificación a través de las ventajas del PHP como tecnología al lado del servidor, del Java Script como tecnología al lado del cliente, del My SQL como servidor de base de datos y el Apache como servidor de red, para la utilización de estos en el desarrollo del portal digital.

1.6 Arquitectura de Software empleada en el SNS.

Arquitectura de tres niveles:

La arquitectura de tres niveles es la generalización de la arquitectura cliente-servidor donde la carga se divide en tres partes con un reparto claro de funciones: una capa para la presentación, otra para el cálculo (capa de negocio) y otra para el almacenamiento (capa de datos). Una capa solamente tiene relación con la siguiente.

Debido a todas las características de arquitectura descritas anteriormente para que pueda ser ejecutado cualquier componente de desarrollado para el SNS en sus servidores centrales se torna necesario cumplir con los requerimientos de MySQL y desplegarse sobre servidor Linux. Estos requisitos propuestos exigen a cualquier aplicación que se ejecute sobre esta red la utilización de PHP como tecnología del lado del servidor, Java Script como tecnología del lado del cliente. **(14)**

1.7 Herramientas utilizadas para el desarrollo del sistema

Visual Paradigm.

A pesar de ser el Rational Rose la herramienta case más utilizada mundialmente en la actualidad, hemos seleccionado para la realización del presente trabajo de diploma el Visual Paradigm, ya que es una herramienta case multiplataforma, que utiliza UML como lenguaje de modelado. Es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario. Su notación es muy parecida a la estándar, permite configurar las líneas de redacción, el modelado de base de datos, el modelado de requerimientos, el modelado del proceso de negocio, la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP además de permitir la integración con herramientas de desarrollo (IDE's). Corrige sintaxis en tiempo de edición, además de contar con plugin para el soporte de las extensiones UML 2.0. y aunque no es una herramienta libre la UCI tiene una licencia para utilizarla.

Zend Studio.

Zend Studio es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE) disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para ciclo de desarrollo de aplicaciones PHP. A través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos. (15)

DB Designer 4 ForK.

DBDesigner 4 ForK es un entorno visual integrado para el diseño, modelado, creación y mantenimiento de bases de datos relacionales, está disponible tanto para sistemas GNU/Linux como para Microsoft Windows 2000/XP, fue creado para funcionar íntegramente con MySQL, aunque también puede ser capaz de soportar otros sistemas por medio de la inclusión de plugins.

Conclusiones

Con el estudio de la situación actual a nivel mundial y nacional de los sistemas de gestión, ha brindado gran apoyo a la realización del sistema informático para la Electromedicina en Cuba, principalmente en el módulo para la gestión de piezas de repuesto, esto es algo que se desarrolla en el país por primera vez y la metodología de trabajo es muy distinta a la de todos los países del mundo por lo que el sistema informático será el único de su tipo. Como resultado del capítulo se abordó sobre la metodología a emplear, las herramientas y las tecnologías.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

En el capítulo se abordan las características fundamentales y básicas del sistema a desarrollar. Se explica el objetivo y el problema de la investigación, en cómo lograr la gestión de las piezas de repuesto, para llevar a cabo el seguimiento de los equipos médicos. Se tiene en cuenta los requerimientos que plantea el cliente y en correspondencia con ello, se plantea la propuesta del sistema. Finalmente se definen los casos de uso, los cuales serán el hilo conductor del presente trabajo.

2.1 Objeto de estudio

Problema y situación problemática

En Cuba existe una gran cantidad de equipos médicos los cuales están distribuidos por todo el país, Electromedicina es la entidad que se dedica a darles mantenimiento y asesoramiento técnico así como su instalación, como es de suponer la cantidad de información que se maneja es enorme por lo que el trabajo es engorroso si se hace manualmente en algún formato duro.

La principal tarea de Electromedicina es velar que los equipos que se rompen en todas las unidades de salud del país sean reparados con la mayor brevedad posible, darle mantenimiento sistemático para evitar roturas (mantenimiento preventivo). En la actualidad no se conocen exactamente la cantidad de equipos instalados en el país, en una ocasión se hizo una encuesta para tratar de conocer la existencia de los equipos así como: la marca, modelo, fabricante y número de serie pero arrojó datos muy imprecisos por no contar con un sistema que le permitiera almacenar la información para hacerle consultas y lograr establecer estándares para los nombres de los equipos, que en muchas ocasiones un mismo equipo era llamado de distintas formas. Toda esta información tan imprecisa no permite hacer una planificación de piezas de repuesto para todos los equipos.

Luego de erradicar el problema que existía, al desarrollar un sistema informático que permite llevar el control de los equipos médicos, se mantiene la dificultad de que estos no son reparados con prontitud porque la adquisición de las piezas necesarias para su reparación aun no cuenta con un sistema informático y se sigue haciendo por las vías

tradicionales: por teléfono, correo y en el mejor de los casos correo electrónico, pero que aun siguen siendo imprecisos los datos, así surge el problema referido a:

¿Cómo contribuir a la gestión de la información de las Piezas de Repuesto en Electromedicina?

2.2 Objeto de Automatización.

Con el sistema se pretende automatizar los siguientes procesos:

➤ **Crear solicitud de piezas.**

En este proceso los operadores, crearán una solicitud de piezas de repuesto cada vez que necesiten para reparar algún equipo. Pueden hacerlo desde su institución contando con una aplicación web. Para ello deberán recoger los datos la orden de servicio que está atendiendo el técnico y los datos de las piezas solicitadas.

➤ **Crear despachos de piezas**

En este proceso los almaceneros contarán con un sistema que les permita crear los despachos de las piezas. Pueden hacerlo desde su institución contando con una aplicación web. Para ello deberán recoger los datos de la orden de servicio que está atendiendo el técnico y verificar si en el almacén existen piezas de las solicitadas.

➤ **Crear movimientos de piezas.**

En este proceso los planificadores contarán con un sistema que les permita crear los movimientos de piezas hacia unidades de Electromedicina que las necesitan para realizar reposiciones. Pueden hacerlo desde su institución contando con una aplicación web. Para ello el sistema visualizara la existencia de piezas en el almacén subordinado que seleccione.

➤ **Crear pedidos de piezas.**

Este proceso consiste en hacer la planificación de la necesidad de piezas de repuesto. Pueden hacerlo desde su institución contando con una aplicación web. Para ello creará el pedido de piezas por trimestre.

2.3 Información que se maneja.

Se utiliza para la realización del sistema la información referente a:

- Datos de las piezas (tipo de pieza y pieza).
- Datos de las planificaciones (pedido y movimiento)
- Datos de los despachos (solicitud y despacho).

Los datos necesarios de los tipos de piezas son:

Del equipo al que pertenece; especialidad, denominación, tipo de equipo, marca, modelo y serie, y propios del tipo de pieza; grupo, descripción, referencia y código cne.

Los datos necesarios de las piezas son:

Tipo de pieza, importador, código del importador, número de contrato, fecha de entrada, costo en moneda nacional y costo en divisas.

Los datos necesarios de los pedidos son:

Provincia, municipio, unidad de Electromedicina, tipo de pieza, cantidad del primer trimestre, cantidad del segundo trimestre, cantidad del tercer trimestre, cantidad del cuarto trimestre y unidad de medida.

Los datos necesarios de los movimientos son:

Provincia, municipio, unidad de Electromedicina que crea el movimiento, pieza que se va a mover, cantidad y unidad de medida de la misma, y provincia, municipio y unidad de Electromedicina hacia donde se va a mover la pieza.

Los datos necesarios de la solicitud son:

Orden de servicio (código), localidad, área de salud, unidad de salud, tipo de pieza, cantidad y unidad de medida.

Los datos necesarios de los despachos son:

Solicitud, pieza, cantidad y unidad de medida.

2.4 Propuesta del sistema.

Con el fin de desarrollar el sistema para la gestión de las piezas de repuesto se propone introducir los datos de las piezas por medio de formularios Web, validando los campos mediante Java script y guardando la información en una base de datos desarrollada en MySQL y utilizaremos PHP como lenguaje de programación para intermediar entre los formularios y la base de datos.

El sistema permitirá a los operadores crear solicitudes de piezas provinciales podrán hacer su solicitud de piezas o material gastable de la provincia a la que pertenece y al igual que los planificadores municipales no tienen acceso a los niveles superiores, los planificadores nacionales harán las solicitudes de piezas o material gastable (la solicitud de piezas o material gastable en cada nivel tiene sus

particularidades), además todos los planificadores tienen la opción de asignar piezas o material gastable, verificar existencia y hacer su planificación de piezas o material gastable cada cual a su nivel y a la unidad de servicio que pertenezca.

2.5 Modelo del Negocio.

Descripción general de los procesos de negocio propuestos.

El mantenimiento (específicamente la gestión de las piezas de repuestos) de los equipos médicos, es una actividad que se lleva a cabo por el personal de Electromedicina, perteneciente al MINSAP. El proceso comienza cuando el Técnico solicita una o varias piezas de repuesto, para la orden de servicio que está atendiendo. El caso de uso finaliza cuando el operador emite la orden de despacho y se la entrega al Técnico.

De esta forma la información se maneja de una forma más segura y precisa, y que no solo eso, sino que la gestión de las piezas se realiza más rápido y al igual la reparación de los equipos médicos, ya sea en dependencia de lo que necesiten, mantenimiento o reparación, en caso de ser mantenimiento si es preventivo, en caso de ser reparación, si es rotura o falta de material tecnológico, cuando sea uno de estos casos y no se encuentre alguna pieza o material gastable en la provincia o el municipio, se hace la solicitud al Centro Nacional, dándole una rápida respuesta y solución al problema.

Así de esta forma no se pierde la información, garantizando que llegue segura a su destino, y en caso de que se pierda, hay posibilidad de recuperarla.

2Representación de los casos de uso del negocio.

Tabla 1 Justificación de los actores del negocio.

Actor	Descripción
Técnico	Son los técnicos de los centros de Electromedicina.

Tabla 2 Justificación de los trabajadores del negocio.

Trabajador	Descripción
Operador.	Es el encargado de recoger los datos presentados por los técnicos en los talleres y actualiza el sistema.
Almacenero.	Es el encargado de atender las solicitudes de piezas.

Diagrama de casos de uso del negocio.

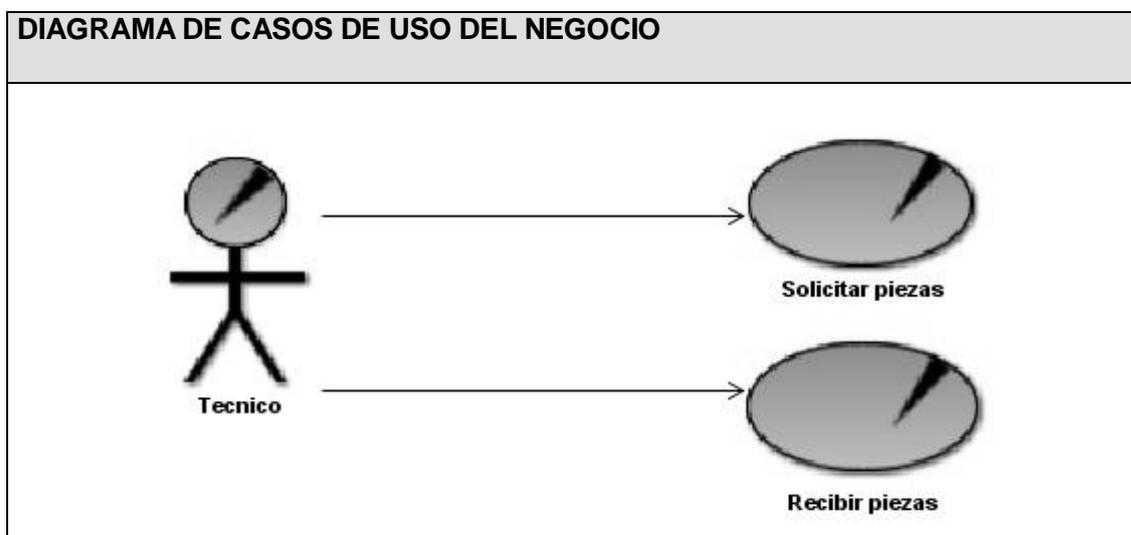


Figura 2 Diagrama de casos de uso del negocio.

Descripción textual de los Casos de Uso del Negocio.

Tabla 3 Descripción textual del caso de uso del negocio solicitar piezas.

Caso de Uso:	Solicitar piezas.
Actores:	Técnico
Trabajadores:	Operador.
Resumen:	El caso de uso se inicia, cuando el Técnico llega a la unidad de Electromedicina, presenta la orden de servicio que está atendiendo y solicita una o varias piezas. El operador crea la solicitud, la imprime y se la da al Técnico.
Precondiciones:	Debe existir una orden de servicio.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El Técnico le muestra la orden de servicio que está atendiendo.	2. El Operador recibe los datos de la orden de servicio. 3. El Operador introduce los datos de la orden de servicio. 4. El Operador pide los datos de las piezas.
5. El Técnico muestra los datos de las piezas que solicita; del equipo: especialidad, denominación, tipo de equipo, marca y modelo; de la pieza: grupo y tipo de pieza.	6. El Operador introduce los datos de las piezas en el sistema. 7. El Operador crea solicitud de piezas. 8. El Operador imprime la solicitud.
9. El técnico recibe la solicitud.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Poscondiciones	El Operador archiva las solicitudes de piezas.

Tabla 4 Descripción textual del caso de uso del negocio recibir piezas.

Caso de Uso:	Recibir piezas.
Actores:	Técnico
Trabajadores:	Almacenero.
Resumen:	El caso de uso se inicia, cuando el Técnico llega al almacén, entrega la orden de despacho al almacenero y este le entrega las piezas. Finaliza el caso de uso.
Precondiciones:	Debe existir piezas en el almacén
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio

1. El Técnico entrega la solicitud de piezas.	2. El Almacenero recibe la solicitud de piezas. 3. El Almacenero verifica existencia de piezas 4. Si hay piezas de las solicitadas en el almacén el Almacenero crea el despacho. 5. El Almacenero entrega las piezas.
6. El técnico recibe las piezas.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	3.1. Si no hay piezas en el almacén el almacenero devuelve la solicitud al técnico.
3.2. El Técnico recibe la solicitud.	
Poscondiciones	El Almacenero archiva los despachos.

Diagramas de actividades.

Diagrama de actividad del caso de uso del negocio solicitar piezas.

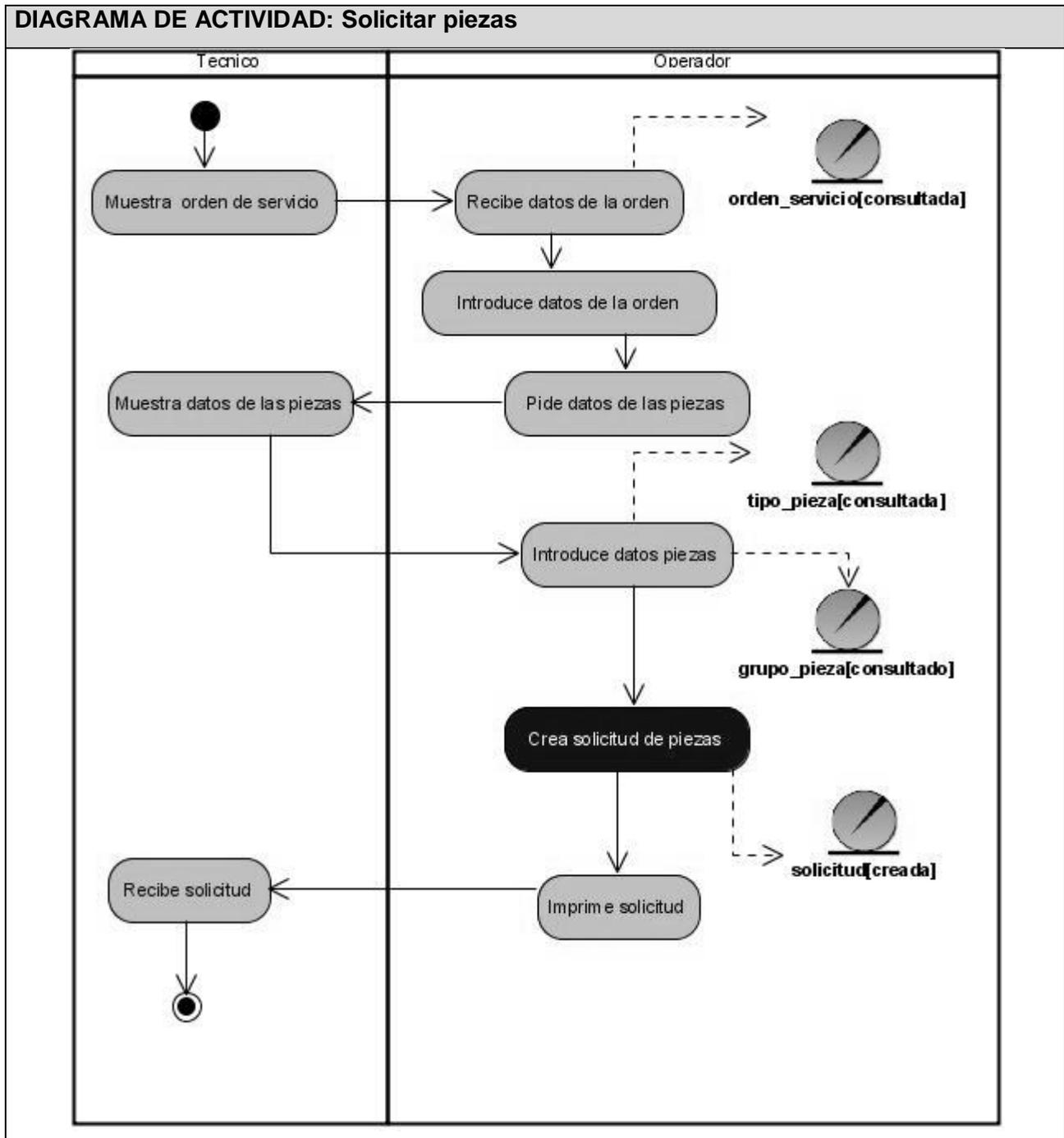


Figura 3 Diagrama de actividad del caso de uso del negocio solicitar piezas.

Diagrama de actividades del caso de uso del negocio recibir piezas.

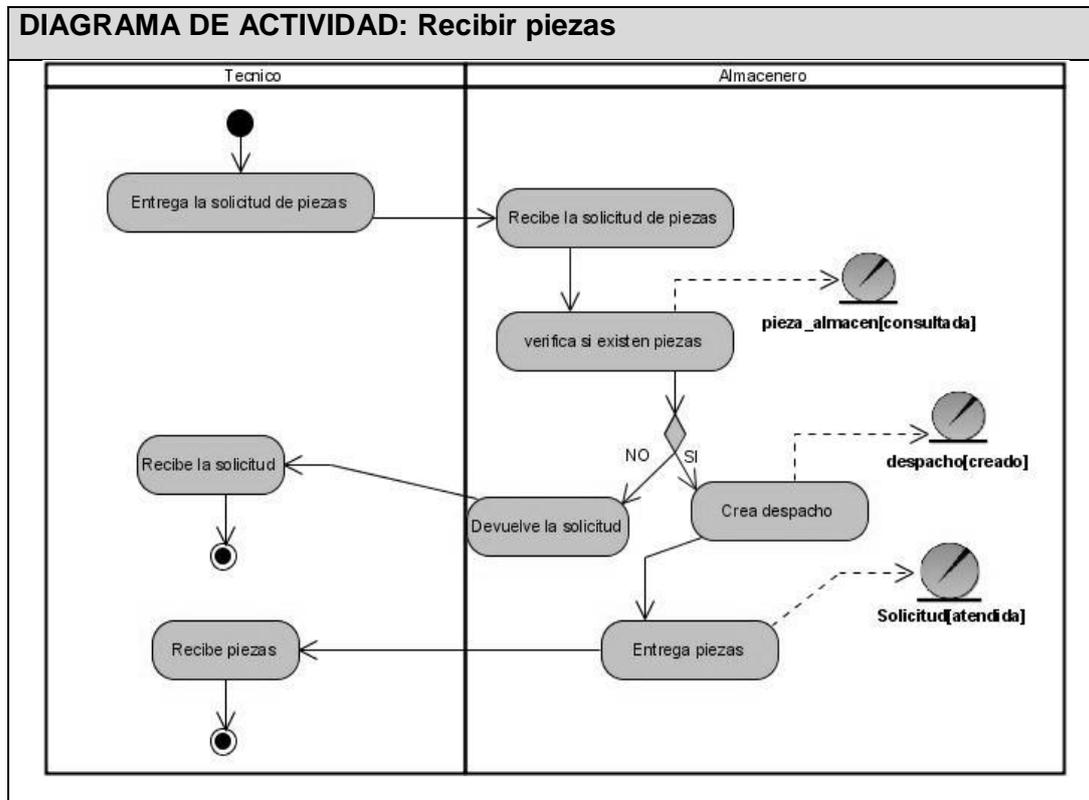


Figura 4 Diagrama de actividades del caso de uso del negocio recibir piezas.

Modelo de objetos.

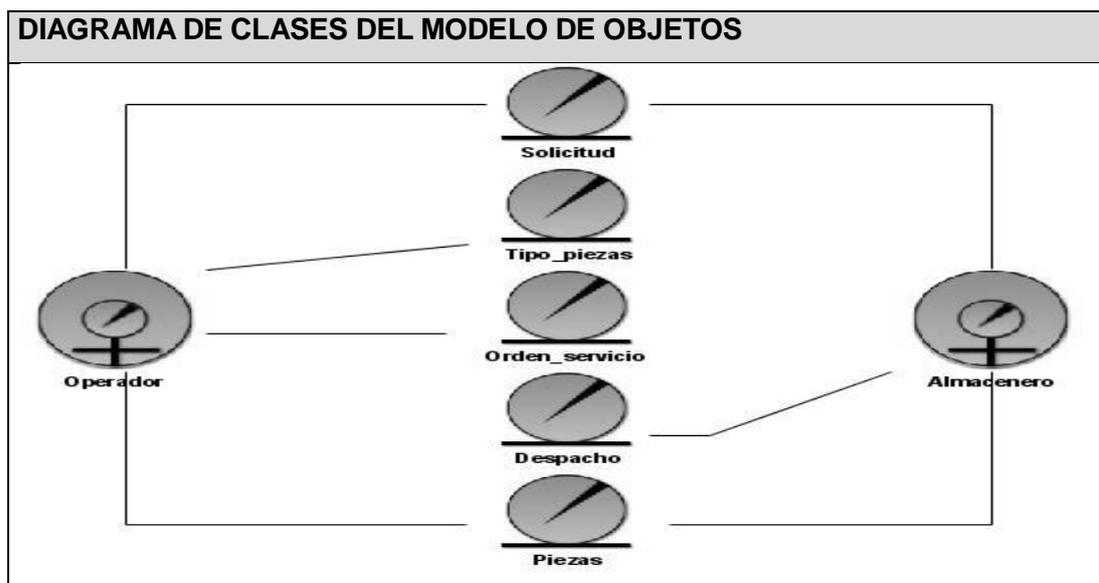


Figura 5 Diagramas de clases del modelo de objetos.

2.6 Especificación de los requisitos de software.

Requerimientos funcionales.

1. Autenticar usuario (nombre de usuario y contraseña).

RF 1 Autentica usuario

2. Gestionar unidad de medida.

RF 2.1 Adicionar unidad de medida.

RF 2.2 Editar unidad de medida.

RF 2.3 Eliminar unidad de medida.

RF 2.4 Visualizar unidades de medida.

3. Gestionar importador.

RF 3.1 Adicionar importador.

RF 3.2 Editar importador.

RF 3.3 Eliminar importador.

RF 3.4 Visualizar importadores.

4. Gestionar solicitudes de piezas.

RF 4.1 Crear solicitud de piezas.

RF 4.2 Editar solicitud de piezas.

RF 4.3 Eliminar solicitud de piezas.

RF 4.4 Visualizar solicitudes de piezas.

5. Gestionar despachos de piezas.

RF 5.1 Crear despacho de piezas.

RF 5.2 Editar despacho de piezas.

RF 5.3 Eliminar despacho de piezas.

RF 5.4 Visualizar despachos.

6. Gestionar tipos de piezas.

RF 6.1 Adicionar tipo de pieza.

RF 6.2 Editar tipo de pieza.

RF 6.3 Eliminar tipo de pieza.

RF 6.4 Visualizar tipos de piezas.

7. Gestionar piezas.

RF 7.1 Adicionar pieza.

RF 7.2 Editar pieza.

RF 7.3 Eliminar pieza.

RF 7.4 Visualizar piezas.

8. Actualizar inventario de piezas del almacén.

RF 8.1 Adicionar pieza al almacén.

RF 8.2 Editar pieza del almacén.

RF 8.3 Visualizar piezas del almacén.

9. Gestionar pedidos de piezas.

RF 9.1 Crear pedido de piezas.

RF 9.2 Editar pedido de piezas.

RF 9.3 Eliminar pedido de piezas.

RF 9.4 Visualizar pedidos de piezas.

10. Gestionar movimientos de piezas.

RF 10.1 Crear movimiento de piezas.

RF 10.2 Editar movimiento de piezas.

RF 10.3 Eliminar movimiento de piezas.

RF 10.4 Visualizar movimientos creados.

RF 10.5 Visualizar movimientos asignados.

Requerimientos no funcionales

Usabilidad:

El sistema puede ser usado por cualquier tipo de personas que posea conocimientos básicos en el manejo de la computadora. Debe ser capaz de ofrecer facilidades de uso para un buen entendimiento y aceptación del producto por los usuarios finales. Debe ser sencillo a la vista de los usuarios.

Rendimiento:

Al estar concebida para un ambiente cliente/servidor, se trata de garantizar la rapidez de respuesta del sistema ante las solicitudes de los usuarios, al igual que la velocidad de procesamiento de la información. Para lo cual se realiza la validación de los datos y la manipulación de eventos en el cliente y en el servidor aquellas que por cuestiones de seguridad, o de acceso a los datos lo requieran. Lográndose así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento, y un mayor aprovechamiento de los recursos.

Apariencia o interfaz externa.

La interfaz externa (vista que se le presentará al usuario que utilice el sistema), como intermediaria entre el software y el usuario debe ofrecer facilidades de entendimiento y sencillez al realizar las operaciones que en ella se presentan.

Soporte:

Se requiere de la instalación de un servidor Web Apache 1.3 o superior con funcionalidades relacionadas con el manejo de la Base de Datos con un gestor MySQL y el servicio de interpretación de códigos con PHP 4 o superior.

Portabilidad:

El sistema será multiplataforma (Linux-Windows) debido a que será capaz de ejecutarse sobre diferentes sistemas operativos sin importar sus versiones, y sin necesidad de modificar su código fuente.

Seguridad:

El sistema contará con una política de seguridad que permita a cada cual acceder a la información que le es permitida de acuerdo al rol que cumple dentro del sistema.

Confiabilidad:

El sistema debe ser confiable y preciso en la información que le suministra al usuario para evitar cualquier tipo de error.

Legales:

El sistema estará desarrollado en base a las políticas del software libre, que fueron ajustadas al sistema nacional de salud como: uso de servidores GNU/Linux, gestor de base de datos MySQL, lenguaje del lado del servidor PHP, que son herramientas libres además del uso de Visual Paradigm que no es libre pero la Universidad de las Ciencias Informáticas posee licencia para su uso.

Software:

Se usará como gestor de bases de datos MySQL Server, garantizando que el sistema sea multiplataforma, es decir, que pueda desplegarse sobre versiones de Windows 98,2000 o superior, o Linux sin dificultad, además de garantizar así, el ajustarse a las políticas que plantea el Sistema Nacional de Salud en cuanto al uso del software libre para sus aplicaciones. Para la navegación del sistema se realizará con el uso de navegadores: Internet-Explorer, Mozilla, Opera, Netscape u otro compatible.

Además se requiere de la instalación previa del paquete LAMP, para el hostéo de la aplicación.

Hardware

Requerimientos mínimos para el servidor:

- Microprocesador superior a 2.0 GHz
- 512 MB RAM o superior.
- 100 GB de Disco Duro como mínimo.

Requerimientos mínimos para la conexión del cliente:

- Microprocesador superior a 2.0 GHz
- 256 MB RAM o superior.
- MODEM o red con TCP-IP para conexión al servidor

2.7 Definición de los casos de uso.

Tabla 5 Justificación de los actores del sistema.

Actor	Justificación
Usuarios del Sistema	Este actor se ha definido para expresar la inicialización de las acciones que pueden ser realizadas por todos los usuarios: autenticación al sistema.
Planificador	Este actor modela a las personas encargadas de: realizar los pedidos de piezas y gestionar movimientos de piezas.
Operador	Este actor modela a las personas encargadas de: gestionar las solicitudes de piezas, gestionar tipos de pieza, gestionar piezas, gestionar unidades de

	medida y gestionar importadores
Almacenero	Este actor modela a las personas encargadas de: actualizar inventario de piezas del almacén y gestionar despachos.

Diagrama de actores del sistema.

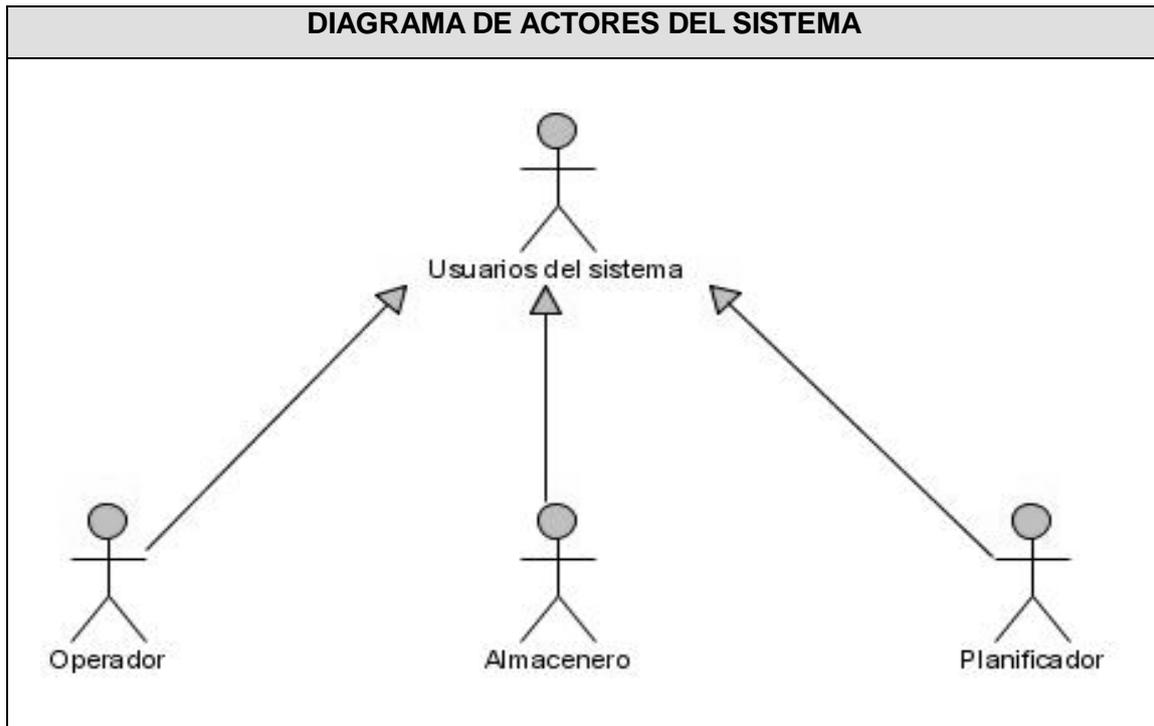


Figura 6 Diagrama de actores del sistema.

Diagrama de casos de uso del sistema.

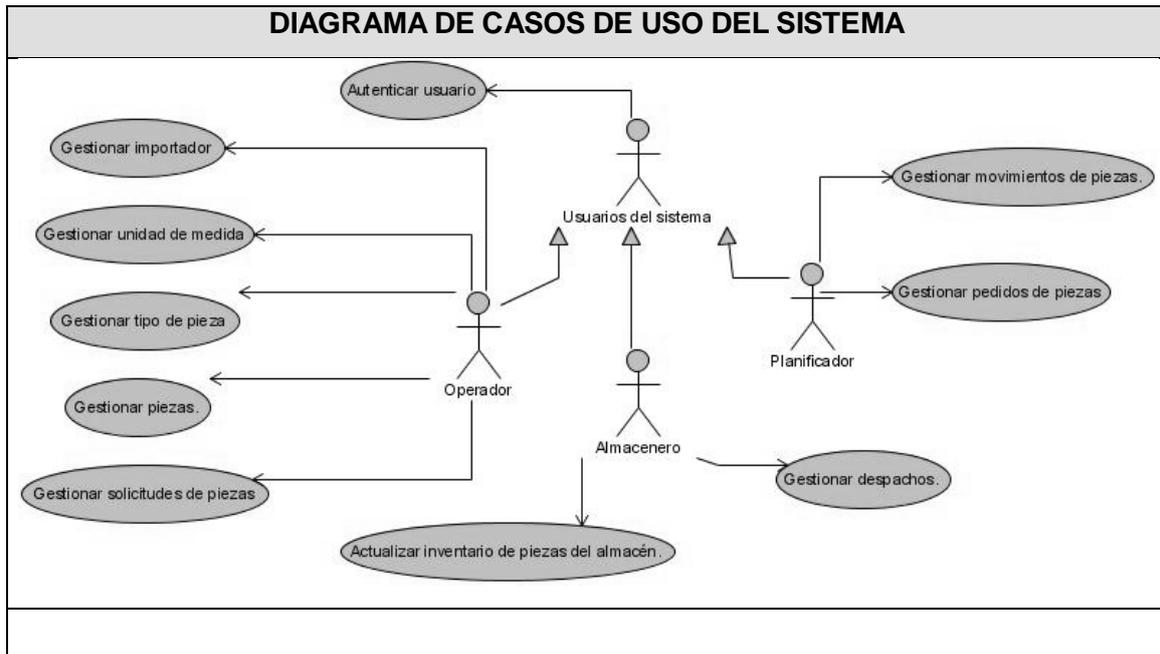


Figura 7 Diagrama de casos de uso del sistema.

2.8 Descripción de los Casos de Uso del Sistema.

Tabla 6 Descripción del caso de uso autenticarse.

Caso de Uso:	Autenticarse
Actores:	Usuario del sistema (Inicia)
Resumen:	El caso de uso se inicia cuando cualquier usuario del sistema teclea su usuario y contraseña, estos se verifican y finaliza dándole los permisos y privilegios por cada nivel, y habilitándole la entrada. Finaliza el caso de uso.
Referencias	RF-1
Prioridad	Crítica.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El usuario del sistema introduce usuario y contraseña.	2 El sistema encripta la contraseña. Busca el usuario y compara la contraseña. En caso de ser correcto se le asignan los permisos.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1. En caso de no existir se envía un mensaje de aviso. Retornar a 1
Poscondiciones	Se habilitan las funcionalidades según lo privilegios.

Tabla 7 Descripción del caso de uso gestionar unidad medida.

Caso de Uso:	Gestionar unidad de medida
Actores:	Operador(Inicia)
Resumen:	El caso de uso se inicia cuando el actor necesita adicionar, editar, eliminar una unidad de medida o visualizar unidades de medida. El sistema muestra una interfaz con un listado de las unidades de medida existentes y las operaciones que se pueden realizar sobre las mismas. Si introduce una unidad de medida, el sistema la almacena en la base de datos. Si selecciona editar alguna, el sistema muestra una interfaz con el valor de la seleccionada, solicita el valor que va a tomar y la almacena en la base de datos. Si selecciona eliminar, el sistema la elimina y actualiza la base de datos. Finaliza el caso de uso.
Referencias	RF-2.1., RF-2.2., RF-2.3., RF-2.4.
Prioridad	Crítica.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El operador selecciona gestionar unidad de medida.	2. El sistema muestra la interfaz con un listado de las unidades de medida y las opciones que se pueden realizar sobre ellas: a) Adicionar unidad de medida (Ir a sección Adicionar unidad de medida) b) Editar unidad de medida (Ir a sección Editar unidad de medida). c) Eliminar unidad de medida (Ir a sección Eliminar unidad de medida).

Acción del Actor		Respuesta del Sistema
1. El Operador inserta la unidad de medida.		
2. El Operador presiona el botón "Adicionar"		3. El sistema verifica que el campo no este vacio.
		4. En caso de estar correcto, el sistema almacena la unidad de medida en la base de datos.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		4.1. En caso de no estar correctos, el sistema muestra un mensaje y señala donde ocurrió el error.
Sección "Editar unidad de medida"		
Acción del Actor		Respuesta del Sistema
1. El Operador selecciona una unidad de medita, presionando el botón "Editar" correspondiente.		2. El sistema muestra en una interfaz con la unidad de medida seleccionada.
3. El Operador introduce el nuevo valor de la unidad de medida.		
4. El Operador presiona el botón "Editar".		5. El sistema verifica que el campo no este vacio.

	6. En caso de estar correcto, el sistema almacena la unidad de medida en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1. En caso de no estar correctos, el sistema muestra un mensaje y señala donde ocurrió el error.
Sección "Eliminar unidad de medida"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona una unidad de medida, presionando el botón "Eliminar" correspondiente.	2. El sistema comprueba que la unidad de medida seleccionada no este siendo utilizada.
	3. En caso de no estar siendo utilizada, el sistema elimina la unidad de medida y actualiza la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1. En caso de estar siendo utilizada el sistema no puede eliminar la unidad de medida y retorna a 1.

Tabla 8 Descripción del caso de uso gestionar importador.

Caso de Uso:	Gestionar importador
Actores:	Operador(Inicia)
Resumen:	El caso de uso se inicia cuando el actor necesita adicionar, editar, eliminar un importador o visualizar importadores. El sistema muestra una interfaz con un listado de los importadores existentes y las operaciones que se pueden realizar sobre los mismos. Si introduce un importador, el sistema lo almacena en la base de datos. Si selecciona editar alguno, el sistema muestra una interfaz con el valor del seleccionado, solicita el valor que va a tomar y la almacena en la base de datos. Si selecciona eliminar, el sistema lo elimina de la base de datos. Finaliza el caso de uso.

Referencias	RF-3.1., RF-3.2., RF-3.3., RF-3.4.	
Prioridad	Crítica.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El operador selecciona gestionar importador.	2. El sistema muestra la interfaz con un listado de los importadores y las opciones que se pueden realizar sobre ellos: a) Adicionar importador (Ir a sección Adicionar importador) b) Editar importador (Ir a sección Editar importador). c) Eliminar importador (Ir a sección Eliminar importador).	
		
Sección "Adicionar importador"		
Acción del Actor	Respuesta del Sistema	
1. El Operador inserta el importador.		
2. El Operador presiona el botón "Adicionar".	3. El sistema verifica que el campo no este vacio.	
	4. En caso de estar correcto, el sistema almacena el importador en la base de datos.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	4.1. En caso de estar vacio el campo, el sistema muestra un mensaje y señala donde ocurrió	

	el error.
Sección "Editar unidad de medida"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona un importador, presionando el botón "Editar" correspondiente.	2. El sistema muestra en una interfaz el importador seleccionado.
3. El Operador introduce el nuevo valor del importador.	
4. El Operador presiona el botón "Editar".	5. El sistema verifica que el campo no este vacio.
	6. En caso de no estar vacio, el sistema lo almacena en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1. En caso de estar vacio el campo el sistema muestra un mensaje y señala donde ocurrió el error.
Sección "Eliminar importador"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona un importador, presionando el botón "Eliminar" correspondiente.	2. El sistema comprueba que el importador no este siendo utilizado.
	3. Si no esta siendo utilizado el sistema elimina el importador de la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1. Si se esta utilizando, el sistema no elimina el importador y retorna a 1

Tabla 9 Descripción del caso de uso gestionar solicitudes de piezas.

Caso de Uso:	Gestionar solicitudes de piezas	
Actores:	Operador (Inicio)	
Resumen:	El caso inicia cuando el Operador necesita crear una solicitud editar, eliminar una solicitud o visualizar solicitudes de piezas. Si selecciona crear solicitud de piezas, el sistema muestra las órdenes de servicio que no se han atendido, permitiéndole seleccionar una, el sistema muestra los tipos de piezas, el actor selecciona los necesitados, guarda la solicitud y actualiza la base de datos Si selecciona visualizar solicitudes el sistema muestra las solicitudes de la orden servicio que seleccione permitiendo editar o eliminar una solicitud. Si selecciona editar solicitud el sistema muestra una interfaz con los datos de solicitud seleccionada, edita los campos y guarda los cambios. Si selecciona eliminar, el sistema la elimina y actualiza la base de datos. Finaliza el caso de uso.	
Precondiciones:	Debe existir al menos una orden de servicio.	
Referencias	RF-4.1., RF-4.2., RF-4.3., RF-4.4.	
Prioridad	Crítica.	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El Operador selecciona el grupo de acciones "Gestionar solicitud de piezas" en el menú.	2. El sistema despliega en el menú las opciones referentes a las operaciones sobre las solicitudes. a. Crear solicitud (Ir a sección Crear solicitud). b. Visualizar solicitudes (Ir a

	sección Visualizar solicitudes).
Sección "Crear solicitud"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona la opción Crear tipo de solicitud del menú.	2. El sistema muestra la interfaz correspondiente a crear solicitud.
3. El Operador introduce los datos: Código de la orden de servicio; de la pieza: grupo de la pieza, descripción, referencia, cantidad solicitada y unidad de medida.	
4. El Operador presionar el botón "Crear".	5. El sistema verifica que los datos introducidos estén correctos.
	6. En caso de estar correctos los datos, almacena la información en la base de datos y actualiza las solicitudes en el servidor de la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1. En caso de no estar correctos los datos, el sistema muestra un mensaje y señala donde ocurrió el error.
Sección "Visualizar solicitudes"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona la opción Visualizar solicitudes del menú.	2. El sistema muestra la interfaz correspondiente.
3. El Operador introduce el código de la orden	4. El sistema muestra las solicitudes

de servicio.	correspondientes
	<p>5. El sistema muestra las opciones que se pueden realizar sobre las solicitudes.</p> <p>a. Eliminar solicitud (Ir a sección eliminar solicitud)</p> <p>b. Editar solicitud (Ir a sección editar solicitud).</p>
Sección "Eliminar solicitud"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona la solicitud a eliminar.	
2. El Operador presiona el botón "Eliminar".	3. El sistema elimina la solicitud de la base de datos.
Sección "Editar solicitud"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona la solicitud que desea modificar.	
2. El Operador presiona el botón "Editar".	3. El sistema muestra la solicitud con todos sus datos.
4. El Operador introduce los nuevos datos y presiona el botón aceptar.	5. El sistema verifica que los datos introducidos sean correctos y no hallan campos vacíos.
	6. Si los datos introducidos son correctos el sistema actualiza los datos de las solicitudes en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5. 1. Si los datos introducidos no son correctos el sistema muestra un mensaje con el error correspondiente y señala el campo incorrecto.

Tabla 10 Descripción del caso de uso gestionar despachos de piezas.

Caso de Uso:	Gestionar despachos de piezas	
Actores:	Almacenero(Inicia)	
Resumen:	El caso inicia cuando el actor necesita crear, editar, eliminar despacho de piezas o visualizar despachos. Si selecciona despachar piezas, el sistema muestra las solicitudes correspondiente a la orden de servicio que seleccione, permitiendo escoger la solicitud con la que creará el despacho, adiciona las piezas al despacho, lo crea y actualiza la base de datos. Si selecciona visualizar despachos el sistema muestra los despachos realizados en la unidad de Electromedicina que seleccione, permitiendo editar o eliminar el despacho que seleccione. Si selecciona editar despacho el sistema muestra una interfaz con los datos del despacho seleccionado, guarda los cambios y actualiza la base de datos. Si selecciona eliminar el despacho, lo elimina y actualiza la base de datos. Finaliza el caso de uso.	
Precondiciones:	Debe existir alguna solicitud de piezas y piezas de las solicitadas.	
Referencias	RF-5.1., RF-5.2., RF-5.3., RF-5.4.	
Prioridad	Crítica.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Almacenero selecciona el grupo de acciones "Gestionar despachos de piezas" en el menú.	2. El sistema despliega en el menú las opciones referentes a las operaciones sobre los despachos. a) Crear despacho de piezas (Ir a sección Crear despacho de piezas). b) Visualizar despachos (Ir a sección Visualizar despachos).	
Sección "Crear despacho de piezas"		
Acción del Actor	Respuesta del Sistema	
1. El Almacenero selecciona Despachar	2. El sistema muestra la interfaz	

piezas en el menú.	correspondiente.
3. El Almacenero introduce el código de la orden servicio.	4. El sistema muestra las solicitudes de la orden de servicio.
5. El Almacenero selecciona la solicitud deseada.	6. El sistema muestra un listado de piezas que pueden cubrir esa solicitud.
7. El almacenero selecciona; del centro: la provincia, el municipio, la unidad de Electromedicina; de la pieza: la pieza que desee despachar, la cantidad y la unidad de medida.	8. El sistema verifica que los datos sean correctos y que no hallan campos vacíos.
	9. Si los datos son correctos el sistema crea el despacho y lo guarda en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	9.1. Si los datos no son correctos, el sistema muestra un mensaje con el error correspondiente y señala donde ocurrió.
Sección "Visualizar despachos"	
Acción del Actor	Respuesta del Sistema
1. El Almacenero selecciona en el menú Visualizar despachos.	2. El sistema muestra la interfaz correspondiente.
3. El almacenero introduce los datos requeridos: provincia, municipio y unidad de Electromedicina.	4. El sistema muestra los despachos correspondientes.
	5. El sistema muestra las acciones que pueden realizarse sobre los despachos.

	<p>a) Eliminar despacho (Ir a sección Eliminar despacho).</p> <p>b) Editar despacho (Ir a sección Editar despacho).</p>
Sección “Eliminar despacho”	
Acción del Actor	Respuesta del Sistema
1. El Almacenero selecciona un despacho y presiona el botón “Eliminar”.	2. El sistema elimina el despacho de la base de datos.
Sección “Editar despacho”	
Acción del Actor	Respuesta del Sistema
1. El Almacenero selecciona un despacho y presiona el botón “Editar”.	2. El sistema muestra una interfaz con los datos del despacho seleccionado.
3. El Almacenero introduce los datos que desea cambiar que pueden ser: pieza, cantidad y/o unidad de medida y presiona el botón “Editar”.	4. El sistema comprueba que los datos sean correctos.
	5. Si los datos introducidos son correctos el sistema guarda los cambios en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1. Si los datos introducidos no son correctos, el sistema muestra un mensaje y señala donde ocurrió el error.

Tabla 11 Descripción del caso de uso gestionar tipos de piezas.

Caso de Uso:	Gestionar tipos de piezas	
Actores:	Operador(Inicia)	
Resumen:	El caso de uso inicia cuando el Operador necesita adicionar, editar, eliminar un tipo de pieza o visualizar tipos de piezas. Si selecciona adicionar tipo de pieza el sistema muestra el listado de los modelos de equipos permitiéndole seleccionar uno y adicionar tipo de pieza guardándola en el sistema. Si selecciona visualizar tipo de pieza, el sistema muestra un listado de los tipos de piezas permitiéndole eliminar o editar nuevamente. Si selecciona una y presiona eliminar tipo de pieza, el sistema elimina el tipo de pieza y actualiza la base de datos y si presiona editar el sistema muestra una interfaz con los datos del tipo de pieza seleccionado, realiza los cambios y se actualiza la base de datos. Finaliza el caso de uso.	
Precondiciones:	Debe existir el modelo.	
Referencias	RF-6.1., RF-6.2., RF-6.3., RF-6.4.	
Prioridad	Crítica.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Operador selecciona el grupo de acciones "Gestionar tipo de piezas" en el menú	2. El sistema despliega en el menú las opciones referentes a las operaciones sobre los tipos de piezas. a. Adicionar tipo de pieza (Ir a sección Adicionar tipo de pieza). b. Visualizar tipos de piezas (Ir a sección Visualizar tipos de piezas).	
Sección "Adicionar tipo de pieza"		
Acción del Actor	Respuesta del Sistema	
1. El Operador selecciona la opción Adicionar tipo de pieza en el menú.	2. El sistema muestra la interfaz correspondiente.	
3. El Operador introduce los		

<p>datos solicitados; del equipo: especialidad, denominación, tipo de equipo, marca y modelo; del tipo de pieza: grupo, unidad de medida, descripción, referencia y código cne.</p>	
<p>4. El Operador presiona el botón "Adicionar".</p>	<p>5. El sistema verifica que los datos sean correctos.</p>
	<p>6. Si los datos son correctos el sistema almacena el tipo de pieza en la base de datos.</p>
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>4.2. Si los datos introducidos no son correctos, el sistema muestra un mensaje y señala donde ocurrió el error.</p>
	<div style="border: 1px solid gray; padding: 5px;"> <div style="text-align: center;">  ADICIONAR TIPO DE PIEZA </div> <hr/> <p>Datos del Equipo</p> <p>Especialidad <input type="text" value="Seleccione..."/></p> <p>Denominación <input type="text"/></p> <p>Tipo de Equipo <input type="text"/></p> <p>Marca <input type="text"/></p> <p>Modelo <input type="text"/></p> <hr/> <p>Datos del tipo de Pieza</p> <p>Grupo <input type="text"/></p> <p>Descripción <input type="text"/></p> <p>Referencia <input type="text"/></p> <p>Código del CNE <input type="text"/></p> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="Adicionar"/> </div> </div>
<p>Sección "Visualizar tipos de piezas"</p>	

Acción del Actor	Respuesta del Sistema
1. El Operador selecciona la opción "Visualizar tipos de piezas" en el menú.	2. El sistema muestra la interfaz correspondiente.
3. El Operador introduce los datos solicitados; del equipo: especialidad, denominación, tipo de equipo, marca y modelo.	4. El sistema muestra los tipos de piezas correspondientes.
	5. El sistema muestra las opciones que se pueden realizar sobre los tipos de piezas. <ul style="list-style-type: none"> a. Eliminar tipo de pieza (Ir a sección Eliminar tipo de pieza) b. Editar tipo de pieza (Ir a sección Editar tipo de pieza).
Sección "Eliminar tipo de pieza"	
Acción del Actor	Respuesta del Sistema
1. El operador selecciona el tipo de pieza que desea eliminar y presiona el botón eliminar.	2. El sistema elimina el tipo de pieza seleccionado y actualiza la base de datos.
Sección "Editar tipo de pieza"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona el tipo de pieza que desea modificar y presiona el botón "Editar".	2. El sistema muestra una interfaz con todos los datos del tipo de pieza.
3. El Operador introduce los cambios del tipo de pieza que pueden ser: grupo, descripción, referencia y/o código cne y presiona el botón "Editar".	4. El sistema verifica que los datos introducidos sean correctos.
	5. Si los datos introducidos son correctos y sistema guarda los cambios en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema

	4.1. Si los datos introducidos no son correctos el sistema muestra un mensaje y señala donde ocurrió error.
--	---

Tabla 12 Descripción del caso de uso gestionar piezas.

Caso de Uso:	Gestionar piezas
Actores:	Operador(inicia)
Resumen:	El caso de uso inicia cuando el Operador necesita adicionar, editar, eliminar una pieza o visualizar piezas. Si selecciona adicionar pieza el sistema muestra el listado de los tipos de piezas permitiéndole seleccionar uno y adicionar la pieza guardándola en el sistema. Si selecciona visualizar piezas, el sistema muestra un listado de las piezas permitiéndole editar o eliminar. Si selecciona una y presiona editar, el sistema muestra una interfaz con los datos de la pieza y el sistema guarda los cambios en la base de datos. Si selecciona una eliminarla, el sistema la elimina y actualiza la base de datos. Finaliza el caso de uso.
Precondiciones:	Debe existir tipo de pieza
Referencias	RF-7.1., RF-7.2., RF-7.3., RF-7.4.
Prioridad	Critica.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona el grupo de acciones "Gestionar piezas" en el menú.	2. El sistema despliega en el menú las opciones referentes a las operaciones sobre las piezas. <ul style="list-style-type: none"> a. Adicionar pieza (Ir a sección Adicionar pieza). b. Visualizar piezas (Ir a sección Visualizar piezas).
Sección "Adicionar pieza"	

Acción del Actor	Respuesta del Negocio
1. El Operador selecciona la opción Adicionar pieza en el menú.	2. El sistema muestra la interfaz correspondiente.
3. El Operador introduce los datos; del equipo: especialidad, denominación, tipo de equipo, marca y modelo; del tipo de pieza: grupo, descripción y referencia; de la pieza: importador, código del importador, costo en moneda nacional, costo en divisas, numero de contrato y fecha de entrada.	4. El sistema verifica que los datos introducidos sean correctos.
5. El Operador presiona el botón adicionar.	6. Si los datos son correctos, el sistema adiciona la pieza a la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1. Si los datos no son correctos, el sistema muestra un mensaje y señala el campo incorrecto.

Sección "Visualizar piezas"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona la opción "Visualizar piezas" en el menú.	2. El sistema muestra la interfaz correspondiente. solicita del equipo: especialidad, denominación, tipo de equipo, marca y modelo; del tipo de pieza: descripción y referencia.
3. El Operador introduce los datos; solicita del equipo: especialidad, denominación, tipo de equipo, marca y modelo; del tipo de pieza: grupo y tipo de pieza.	4. El sistema muestra la interfaz con los datos de las piezas correspondientes
	5. El sistema muestra las opciones que se pueden realizar sobre las piezas. <ol style="list-style-type: none"> a. Eliminar pieza (Ir a sección Eliminar pieza).

 **ADICIONAR PIEZA**

Datos del Equipo

Especialidad	Seleccione... ▾
Denominación	
Tipo de Equipo	
Marca	
Modelo	

Datos de la Pieza

Grupo	
Tipo de Pieza	
Importador	
Código del Importador	
Número de Contrato	
Fecha de Entrada	
Costo en MN	
Costo en Divisas	

	b. Editar pieza (Ir a sección Editar pieza).
Sección "Eliminar pieza"	
Acción del Actor	Respuesta del Sistema
1. El operador selecciona la pieza que desea eliminar y presiona el botón "Eliminar".	2. El sistema elimina la pieza seleccionada y actualiza la base de datos.
Sección "Editar piezas"	
Acción del Actor	Respuesta del Sistema
1. El Operador selecciona la pieza y presiona el botón "Editar".	2. El sistema muestra una interfaz con todos los datos de la pieza.
3. El Operador introduce los campos que puede cambiar: importador, código del importador, costo en moneda nacional, costo en divisas, numero de contrato y fecha de entrada	
4. El Operador presiona el botón "Editar".	5. El sistema comprueba que no halla campos vacios y que los datos sean correctos.
	6. Si los datos introducidos son correctos el sistema actualiza los datos de la pieza en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.2. Si los datos introducidos no son correctos el sistema muestra un mensaje y señala el parámetro incorrecto.

Tabla 13 Descripción del caso de uso actualizar inventario de piezas del almacén.

Caso de Uso:	Actualizar inventario de piezas del almacén.	
Actores:	Almacenero(Inicia)	
Resumen:	El caso de uso inicia cuando el actor necesita adicionar piezas al almacén, editar piezas del almacén o visualizar piezas del almacén. Si selecciona la opción adicionar piezas al almacén, el sistema solicita los datos referentes a la pieza y la cantidad. Si selecciona visualizar el sistema muestra un listado de las piezas que tiene en el almacén, permitiendo editar una pieza del almacén. Si selecciona una y presiona editar, el sistema muestra una interfaz con los datos de la pieza seleccionada, lo edita y guarda los cambios en la base de datos. Finaliza el caso de uso.	
Precondiciones:	Tiene que existir piezas.	
Referencias	RF-8.1., RF-8.2., RF-8.3.	
Prioridad	Crítica.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El Almacenero selecciona el grupo de acciones “Actualizar inventario de piezas del almacén” en el menú.	2. El sistema despliega un menú con las acciones que pueden realizarse sobre el inventario de piezas del almacén. a. Adicionar piezas al almacén (Ir a sección Adicionar piezas al almacén). b. Visualizar piezas del almacén (Ir a sección Visualizar piezas del almacén).	
Sección “Adicionar piezas al almacén”		
Acción del Actor	Respuesta del Negocio	
1. El Almacenero selecciona la opción “Adicionar piezas al almacén” en el menú.	2. El sistema muestra la interfaz correspondiente.	
3. El Almacenero introduce los datos; del equipo especialidad, denominación, tipo de equipo,	4. El sistema comprueba que no existan campos vacíos e incorrectos.	

marca, modelo; de la pieza: grupo, tipo de pieza, importador, numero de contrato; para el almacén: la cantidad, unidad de medida, provincia, municipio y la unidad de Electromedicina.	
	5. Si los datos son correctos, el sistema guarda los datos y actualiza la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	5.1 Si los datos no son correctos, el sistema muestra un mensaje y señala los datos incorrectos.
Sección “Visualizar inventario de piezas del almacén”	
Acción del Actor	Respuesta del Negocio
1. El Almacenero selecciona la opción “Visualizar inventario de piezas del almacén” en el menú.	2. El sistema muestra la interfaz correspondiente.
3. El Almacenero selecciona los datos; del centro: provincia, municipio, unidad de Electromedicina, requeridos: del equipo: especialidad, denominación, tipo de equipo, marca y modelo.	4. El sistema muestra las piezas que se encuentra en el almacén.
5.	6. El sistema muestra un menú con las opciones que pueden realizarse sobre el inventario de piezas. a) Editar piezas del almacén (Ir a sección Editar piezas del almacén).
Sección “Editar piezas del almacén”	
Acción del Actor	Respuesta del Negocio

<p>1. El Almacenero selecciona la pieza que desea editar y presiona el botón "Editar".</p>	<p>2. El sistema muestra una interfaz con los datos de la pieza correspondiente.</p>
<p>3. El Almacenero introduce los datos que desea modificar que pueden ser: cantidad y unidad de medida y presiona el botón "Editar".</p>	<p>4. El sistema comprueba que no existan campos vacios y que los valores sean correctos.</p>
	<p>5. Si la cantidad introducida es igual cero (0), el sistema elimina la pieza del almacén y actualiza la base de datos.</p>
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Negocio</p>
	<p>5.2 Si la cantidad nos es cero (0) guarda los cambios en la base de datos y actualiza las piezas del almacén.</p>
	<p>5.3 Si los valores no son correctos el sistema muestra un mensaje y señala donde ocurrió el error.</p>

Tabla 14 Descripción del caso de uso gestionar pedidos de piezas.

Caso de Uso:	Gestionar pedidos de piezas
Actores:	Planificador(Inicia)
Resumen:	El caso de uso inicia cuando el actor necesita crear, editar, eliminar un pedido de piezas o visualizar pedidos. Si el actor selecciona crear pedido, el sistema solicitará los datos requeridos, lo crea y actualiza la base de datos. Si selecciona visualizar pedidos el sistema solicita la unidad de Electromedicina, el año y el modelo de equipo y mostrará un listado con los pedidos correspondientes, permitiendo editar o eliminar un pedido. Si selecciona uno y presiona eliminar, el sistema lo elimina y actualiza la base de datos y si presiona editar el sistema muestra una interfaz con los datos del pedido seleccionado, lo edita y guarda los cambios en la base de datos. Finaliza el caso de uso.
Precondiciones:	
Referencias	RF-9.1., RF-9.2., RF-9.3., RF-9.4.
Prioridad	Crítica.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona el grupo de acciones “Gestionar pedidos de piezas” en el menú.	2. El sistema despliega en el menú las opciones referentes a las operaciones sobre los pedidos de piezas. <ul style="list-style-type: none"> a. Crear pedido de piezas (Ir a sección Crear pedido de piezas). b. Visualizar pedidos de piezas (Ir a sección Visualizar pedidos de piezas).
Sección “Crear pedido de piezas”	
Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona la opción Crear pedido de piezas del menú.	2. El sistema muestra la interfaz correspondiente

<p>3. El Planificador introduce los datos; del centro: provincia, municipio y unidad de Electromedicina; del equipo: especialidad, denominación, tipo de equipo, marca y modelo; de la pieza: grupo, tipo de pieza, cantidad por trimestre y unidad de medida y presiona el botón "Crear".</p>	<p>4. El sistema verifica que los datos introducidos sean correctos.</p>
	<p>5. Si los datos introducidos son correctos el sistema guarda el pedido de piezas en la base de datos.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>6.3. Si los datos no son correctos, el sistema muestra un mensaje y señala los campos incorrectos.</p>

 CREAR PEDIDO DE PIEZAS			
Datos del Centro			
Provincia	Seleccione... ▼		
Municipio			
Centro			
Año	Seleccione... ▼		
Datos del Equipo			
Especialidad			
Denominación			
Tipo de Equipo			
Marca			
Modelo			
Datos de la Pieza			
Grupo			
Tipos de Piezas			
Unidad de Medida			
Cantidad por Trimestre			
Cant_1ro	Cant_2do	Cant_3ro	Cant_4to
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Guardar"/>			

Sección “Visualizar pedidos de piezas”

Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona la opción “Visualizar pedidos de piezas” en el menú.	2. El sistema muestra una interfaz con los datos requeridos.
3. El Planificador introduce los datos solicitados.	4. El sistema muestra un listado con los pedidos de piezas correspondientes y las opciones que se pueden realizar sobre los mismos. <ul style="list-style-type: none"> a. Eliminar pedido (Ir a sección Eliminar pedido). b. Editar pedido de piezas (Ir a sección Editar pedido de piezas).

Sección “Eliminar pedido de piezas”

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

1. El Planificador selecciona un pedido y presiona el botón "Eliminar".	2. El sistema elimina el pedido de la base de datos.
Sección "Editar pedido de piezas"	
Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona un pedido y presiona el botón "Editar".	2. El sistema muestra una interfaz con los datos del pedido seleccionado.
3. El Planificador introduce los datos del pedido: cantidad por trimestre y unidad de medida y presiona el botón "Editar".	4. El sistema verifica que no halla campos vacíos y que estén correctos los parámetros.
	5. Si los datos introducidos son correctos el sistema guarda los cambios en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5.1 Si los datos no son correcto el sistema muestra un mensaje y señala los parámetros incorrectos.

Tabla 15 Descripción del caso de uso gestionar movimientos de piezas.

Caso de Uso:	Gestionar movimientos de piezas	
Actores:	Planificador(Inicia)	
Resumen:	<p>El caso de uso inicia cuando el actor necesita crear, eliminar, editar movimiento de piezas o visualizar movimientos de piezas. Si selecciona crear movimiento de piezas, el sistema solicitará la unidad de Electromedicina (origen) donde va a buscar la pieza, mostrando el listado de las piezas que tiene en el almacén, permitiendo seleccionar la pieza que va a ser movida, la cantidad de la misma, unidad de medida y la unidad de Electromedicina a la que va a ser movida. Si selecciona visualizar movimientos creados, el sistema solicita unidad de Electromedicina (origen) y muestra los movimientos realizados, permitiendo seleccionar un movimiento que puede ser eliminado y modificado. Finaliza el caso de uso. Si selecciona uno y presiona eliminar, el sistema lo elimina y actualiza la base de datos y si presiona editar el sistema muestra una interfaz con los datos del pedido seleccionado, lo edita y guarda los cambios en la base de datos. Si selecciona visualizar movimientos asignados el sistema muestra un listado los movimientos asignados. Finaliza el caso de uso.</p>	
Precondiciones:	Tiene que existir unidades de Electromedicina y piezas en el almacén.	
Referencias	RF-10.1., RF-10.2., RF-10.3., RF-10.4., RF-10.5.	
Prioridad	Crítica.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Planificador selecciona el grupo de acciones “Gestionar movimientos de piezas” en el menú.	2. El sistema despliega en el menú las opciones referentes a las operaciones sobre los movimientos de piezas. <ul style="list-style-type: none"> a) Crear movimiento de piezas (Ir a sección mover pieza). b) Visualizar movimientos creados (Ir a sección visualizar movimientos creados). 	

	c) Visualizar movimientos asignados (Ir a sección Visualizar movimientos asignados).
Sección "Crear movimiento de piezas"	
Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona la opción Crear movimiento de piezas del menú.	2. El sistema muestra la interfaz correspondiente a crear movimiento
3. El Planificador selecciona; del centro: provincia, municipio y unidad de Electromedicina (origen); del equipo: especialidad, denominación, tipo de equipo, marca y modelo.	4. El sistema muestra las piezas que tiene en el almacén.
5. El Planificador selecciona la pieza que va a mover, introduce la cantidad y la unidad de Electromedicina hacia donde será movida. Presiona el botón "Crear".	6. El sistema verifica que los datos introducidos sean correctos y no halla campos vacíos.
	7. Si los datos introducidos son correctos se guardan los datos en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	7.1. Si los datos introducidos no son correctos, el sistema muestra un mensaje y señala los datos incorrectos.

Sección "Visualizar movimientos creados"	
Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona la opción "Visualizar movimientos creados" en el menú.	2. El sistema el sistema muestra la interfaz correspondiente.
3. El Planificador introduce los datos requeridos: provincia, municipio: unidad de Electromedicina (origen).	4. El sistema muestra los movimientos de piezas.

 **CREAR MOVIMIENTO PIEZAS**

Unidad de Origen

Provincia

Municipio

Centro

Datos del Equipo

Especialidad

Denominación

Tipo de Equipo

Marca

Modelo

Datos de la Pieza

Grupo

Tipo de pieza

Importador

No. Contrato

Cantidad

Unidad Destino

Provincia

Municipio

Centro

	<p>5. El sistema muestra un menú con las opciones que pueden realizarse sobre los movimientos de piezas.</p> <p>a) Eliminar movimiento de piezas (Ir a sección Eliminar movimiento de piezas).</p> <p>b) Editar movimiento de piezas (Ir a sección Editar movimiento de piezas).</p>
Sección “Eliminar movimiento”	
Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona el movimiento que desea eliminar y presiona el botón “Eliminar”.	2. El sistema elimina el movimiento de la base de datos.
Sección “Editar movimiento”	
Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona un movimiento y presiona el botón “Editar”.	2. El sistema muestra una interfaz con todos los datos del movimiento de pieza seleccionado.
3. El Planificador introduce los cambios en los datos solicitados: cantidad y unidad y presiona el botón Editar.	4. El sistema comprueba que los parámetros sean correctos y no halla campos vacíos.
	5. Si los datos introducidos son correctos, el sistema guarda los datos del movimiento de piezas en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5.1. Si los datos introducidos no son correctos, el sistema muestra un mensaje y señala los datos incorrectos.
Sección “Visualizar movimientos asignados”	

Acción del Actor	Respuesta del Sistema
1. El Planificador selecciona la opción "Visualizar movimientos asignados" en el menú.	2. El sistema el sistema muestra la interfaz correspondiente.
3. El Planificador introduce los datos requeridos: provincia, municipio unidad de Electromedicina (destino).	4. El sistema muestra los movimientos de piezas asignados.

Conclusiones:

En el capítulo se profundizó en el objeto de estudio y la situación problemática, se realizó la modelación del negocio con sus diagramas y descripción de los casos uso del negocio (solicitar piezas, recibir piezas), se recogieron 36 requerimientos funcionales que se agruparon en 10 casos de uso, se definieron los casos de uso mediante los diagramas y se realizó la descripción de los mismos. Con la realización de este capítulo se evidencia que el sistema propone mejoras muy necesarias para la gestión de las piezas de repuesto en Electromedicina.

CAPÍTULO 3: ANALISIS Y DISEÑO DEL SISTEMA

Introducción.

Este capítulo abarca el desarrollo de la fase de análisis y diseño del sistema, englobando todo lo referente al modelado del diseño específicamente. En el mismo figuran los diagramas de clases del diseño de los casos de uso definidos en el capítulo anterior, así como los diagramas de secuencia correspondientes a dichos casos de uso. Además se representa el modelo de datos con la descripción de sus tablas, el diagrama de clases persistentes y el modelo de despliegue del sistema.

3.1 Diagramas de Clases.

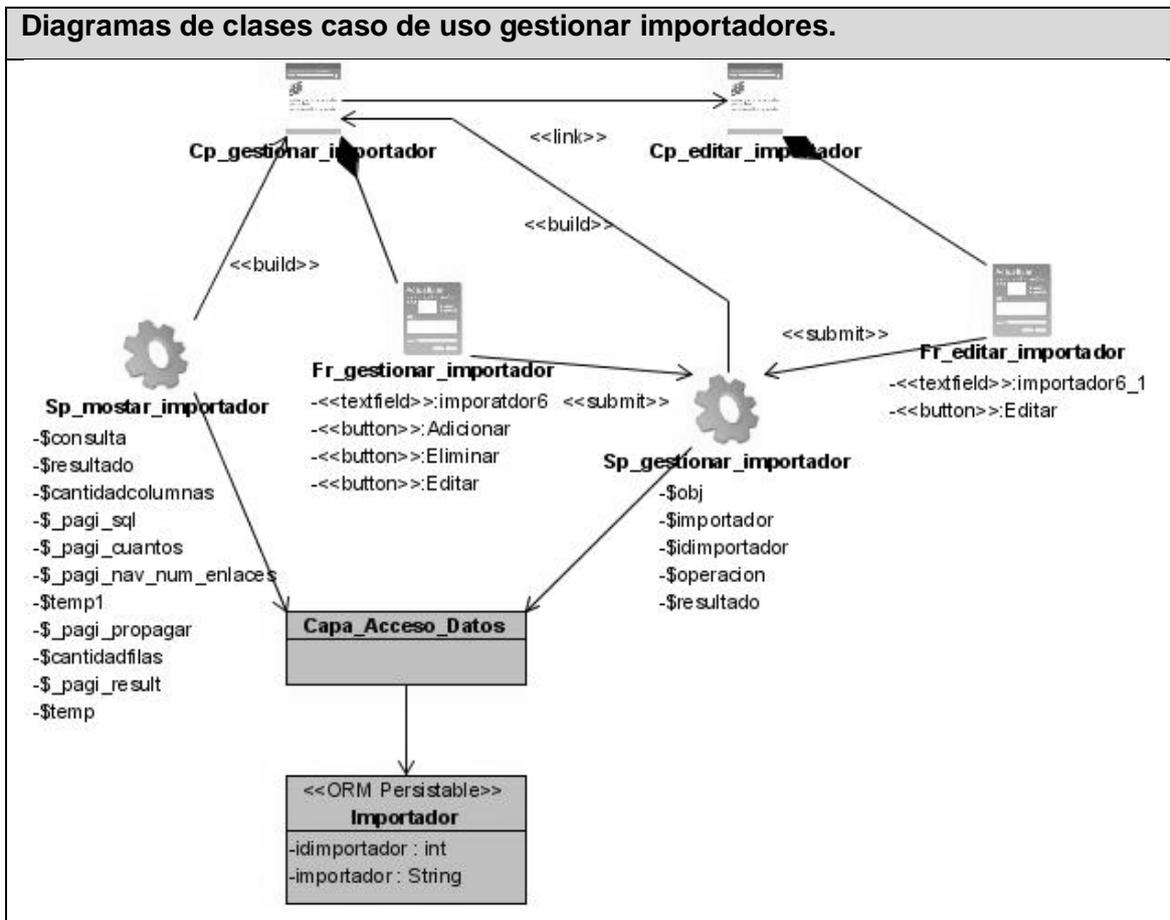


Figura 8 Diagramas de clases caso de uso gestionar importadores.

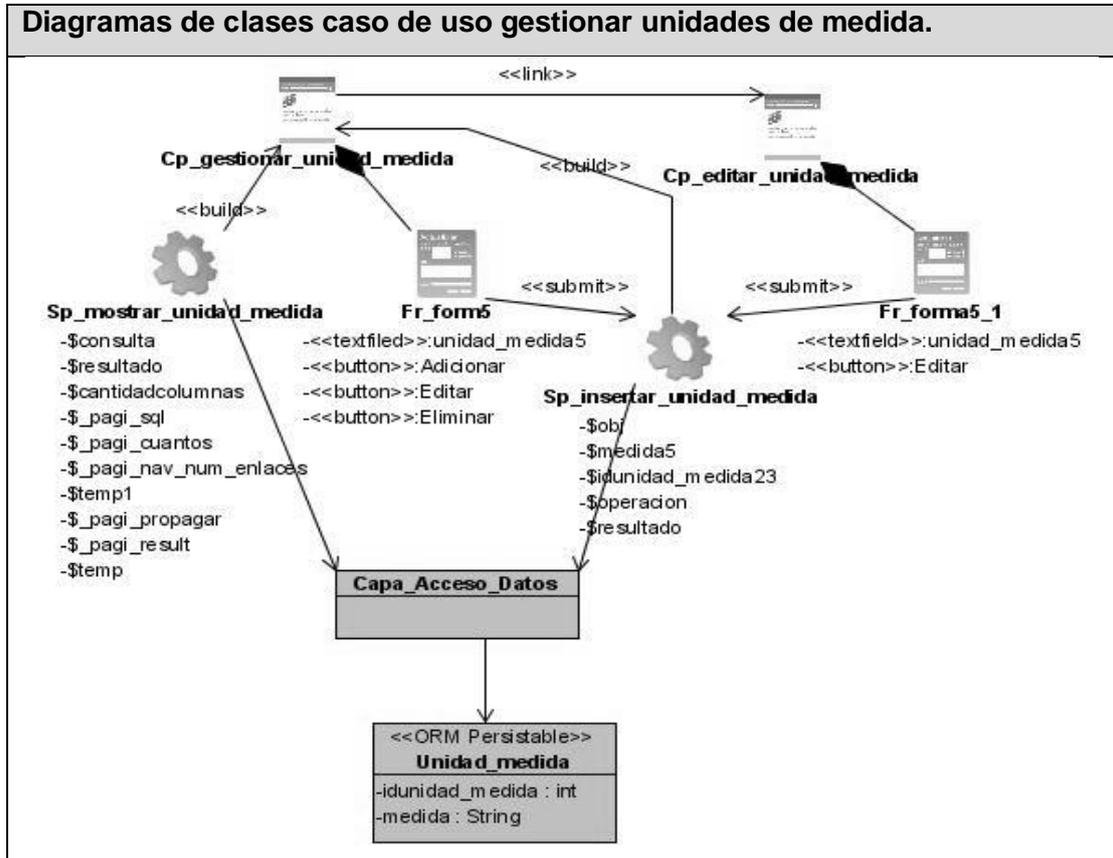


Figura 9 Diagramas de clases caso de uso gestionar unidades de medida.

Diagrama de clases caso de uso gestionar tipos de piezas.

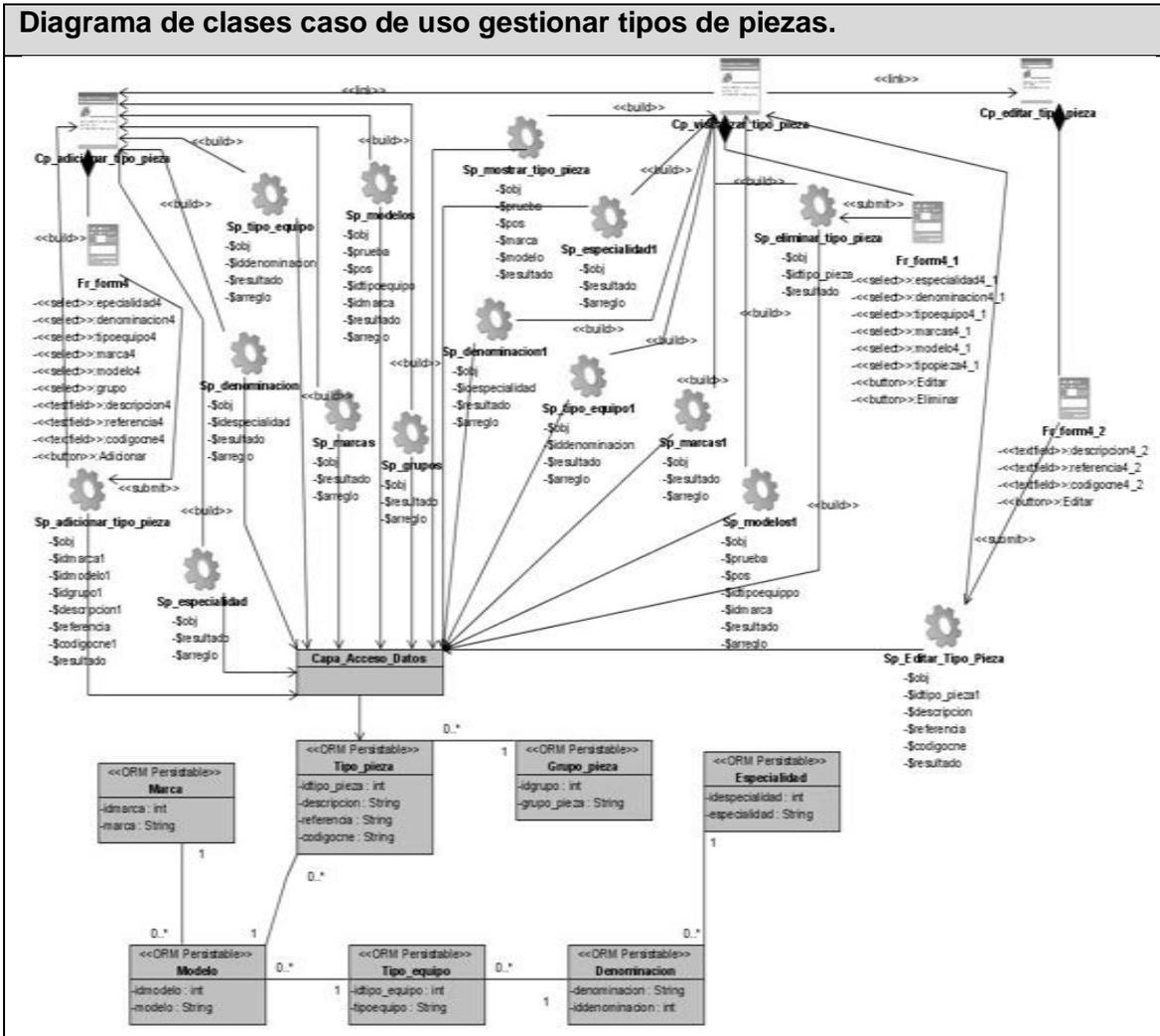


Figura 10 Diagrama de clases caso de uso gestionar tipos de piezas.

3.2 Diagramas de interacción del diseño.

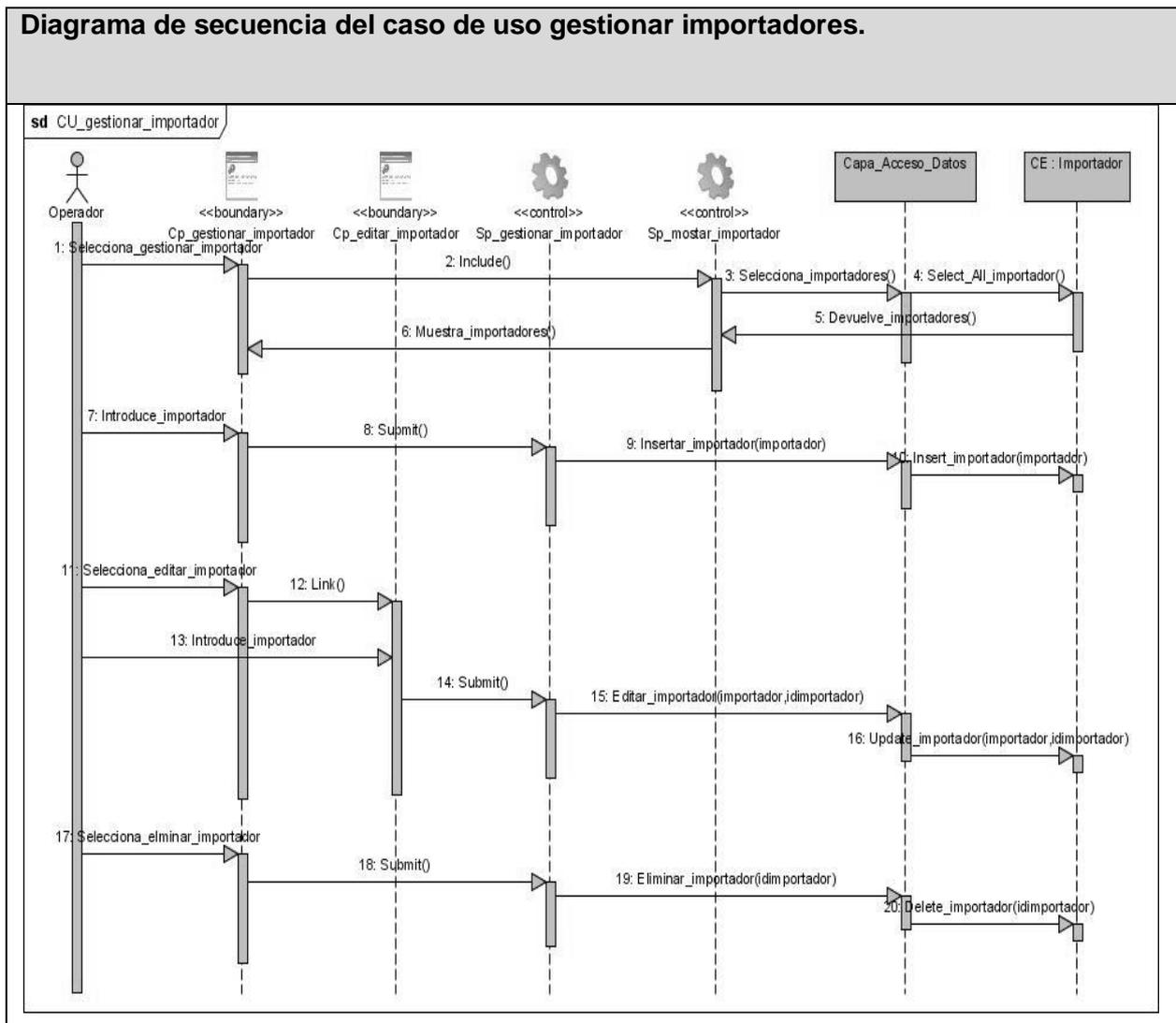


Figura 12 Diagrama de secuencia caso de uso gestionar importadores.

Diagrama de secuencia del caso de uso gestionar unidad de medida.

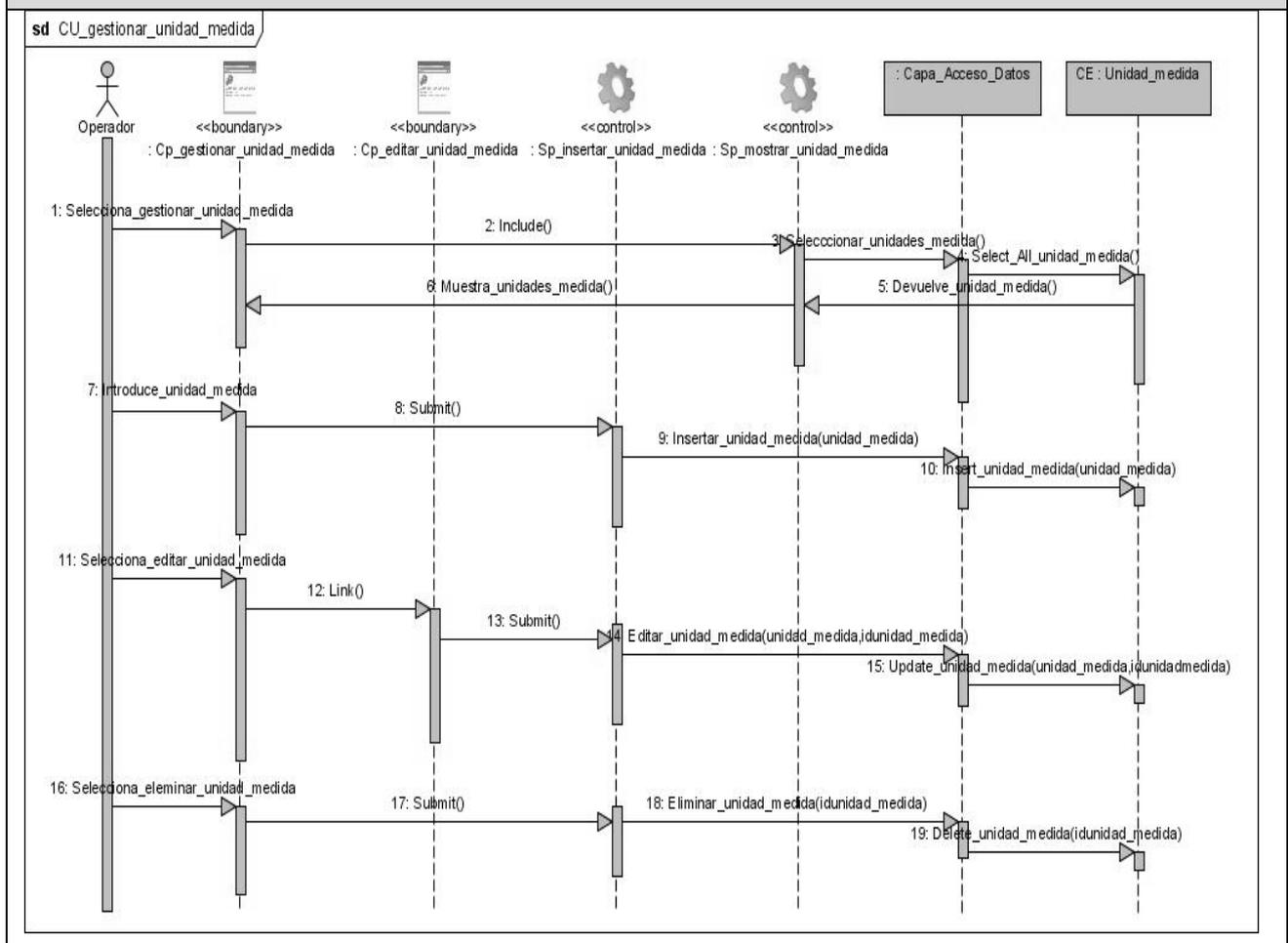


Figura 13 Diagrama de secuencia del caso de uso gestionar unidad de medida.

Diagrama de secuencia del caso de uso gestionar tipo de piezas, escenario: adicionar tipo de pieza.

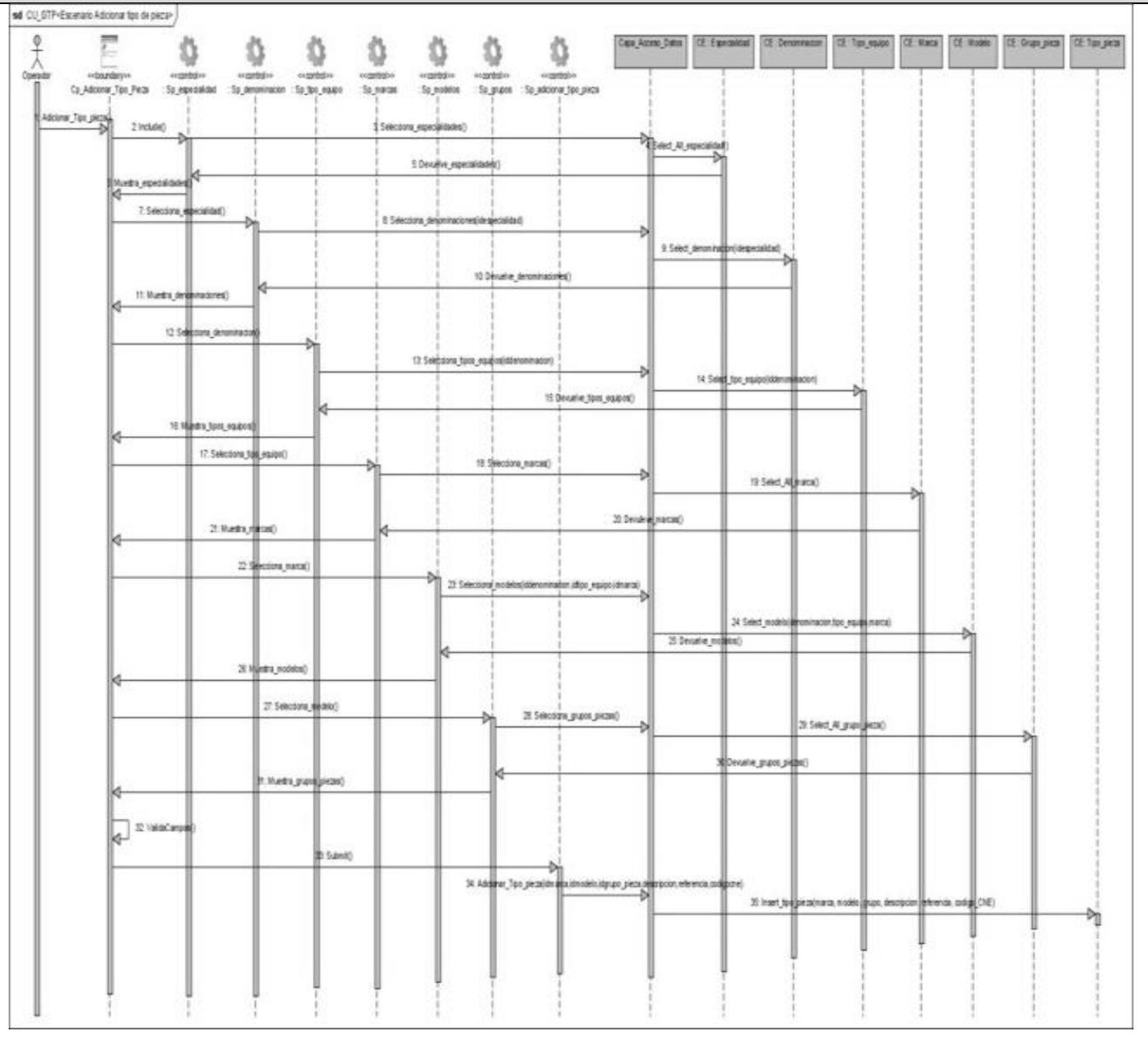


Figura 14 Diagrama de secuencia del caso de uso gestionar tipo de piezas, escenario: adicionar tipo de pieza.

Nota: El resto de los diagramas de secuencia se pueden encontrar en el expediente de proyecto.

3.3 Descripción de las clases.

Tabla 16 Descripción de la clase Cl_adicionar_tipo_pieza

Nombre: Cl_adicionar_tipo_pieza	
Tipo de clase: interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	MostrarDenominacion()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra al usuario las denominaciones existentes dada la especialidad.
Nombre:	MostrarTipoEquipo()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra los tipos de equipos dada la denominación.
Nombre:	MostrarMarca()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra las marcas de equipos que existen.
Nombre:	MostraModelo()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra los modelos de equipos dada la denominación, tipo de equipo y marca.
Nombre:	MostrarGrupoPieza()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra los grupos de piezas que existen.
Nombre:	validar_adicionar_tipo_pieza()
Descripción:	Valida la entrada de datos.

Tabla 17 Descripción de la clase CI_adicionar_pieza

Nombre: CI_adicionar_pieza	
Tipo de clase: interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	MostrarDenominacion()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra al usuario las denominaciones existentes dada la especialidad.
Nombre:	MostrarTipoEquipo()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra los tipos de equipos dada la denominación.
Nombre:	MostrarMarca()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra las marcas de equipos que existen.
Nombre:	MostraModelo()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra los modelos de equipos dada la denominación, tipo de equipo y marca.
Nombre:	MostrarGrupoPieza()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra los grupos de piezas que existen.
Nombre:	MostrarTipoPieza()
Descripción:	Función que accede a una clase controladora la cual se conecta a la BD y muestra los tipos de piezas dada la marca el modelo y el grupo de pieza.
Nombre:	validar_adicionar_pieza()
Descripción:	Valida la entrada de datos.

Tabla 18 Descripción de la clase CCadicionar_tipo_pieza

Nombre: CC_adicionar_tipo_pieza	
Tipo de clase: controladora	
Atributo	Tipo
\$obj	Acceso_Datos
\$idmarca1	Int
\$idmodelo1	Int
\$idgrupo1	Int
\$descripcion1	String
\$referencia1	String
\$codigocone1	String
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla 19 Descripción de la clase CCadicionar_pieza

Nombre: CC_adicionar_pieza	
Tipo de clase: controladora	
Atributo	Tipo
\$obj	Acceso_Datos
\$idimportador	Int
\$costomn	Int
\$nocontrato	Int
\$costodivisas	Int
\$coimportador	String
\$fechaentrada	Date
\$codigocone1	String
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla 20 Descripción de la clase area_salud

Nombre: area_salud		
Descripción: Registro de áreas de salud		
Atributo	Tipo	Descripción
idarea_salud	int(10) unsigned	Identificador del área de salud
idprovincia	int(10) unsigned	Identificador de la provincia
idmunicipio	int(10) unsigned	Identificador del municipio
area_salud	varchar(200)	Nombre del área de salud

Tabla 21 Descripción de la clase baja

Nombre: baja		
Descripción: Registro de bajas.		
Atributo	Tipo	Descripción
idequipo	int(10) unsigned	Identificador del equipo
idunidad_salud	int(10) unsigned	Identificador de la unidad de salud
fecha	date	Fecha en que se produjo
observacion	text	Observaciones que se planteen

Tabla 22 Descripción de la clase denominación

Nombre: denominación		
Descripción: Registro de denominaciones.		
Atributo	Tipo	Descripción
iddenominacion	int(10) unsigned	Identificador de la denominación
idespecialidad	int(10) unsigned	Identificador de la especialidad
denominacion	varchar(45)	Nombre de la denominación

Tabla 23 Descripción de la clase despacho

Nombre: despacho		
Descripción: Registro de despachos.		
Atributo	Tipo	Descripción
iddespacho	int(10) unsigned	Identificador del despacho
idunidad_electromedicina	int(10) unsigned	Identificador de la unidad de Electromedicina.
idequipo	int(10) unsigned	Identificador del equipo
idunidad_salud	int(10) unsigned	Identificador de la unidad de salud
idreporte	int(10) unsigned	Identificador del reporte
idsolicitud	int(10) unsigned	Identificador de la solicitud
idpieza	int(10) unsigned	Identificador de la pieza
idunidad_medida	int(10) unsigned	Identificador de la unidad de medida.
cantidad	float	Cantidad a despachada
fecha_realizado	varchar(20)	Fecha en la que se realizo

Tabla 24 Descripción de la clase equipo

Nombre: equipo		
Descripción:		
Atributo	Tipo	Descripción
idequipo	int(10) unsigned	Identificador del equipo
idunidad_salud	int(10) unsigned	Identificador de la unidad de salud

idsoftware	int(10) unsigned	Identificador de software
idprioridad	int(10) unsigned	Identificador de la prioridad
idmarca	int(10) unsigned	Identificador de la marca
idmodelo	int(10) unsigned	Identificador del modelo
idubicacion	int(10) unsigned	Identificador de la ubicación
idprograma	int(10) unsigned	Identificador del programa
idestado	int(10) unsigned	Identificador del estado
anno_fabricacion	year(4)	Año de fabricación
fecha_instalacion	date	Fecha en que se instaló
noserie	varchar(20)	Número de serie
noinventario	int(11)	Número de inventario
detalleubicacion	text	Detalles de la ubicación
observaciones	text	Observaciones que se planteen
codigo	varchar(50)	Código del equipo

Tabla 25 Descripción de la clase especialidad

Nombre: especialidad		
Descripción: Registro de especialidades		
Atributo	Tipo	Descripción
idespecialidad	int(10) unsigned	Identificador de la especialidad
especialidad	varchar(45)	Nombre de la especialidad

Tabla 26 Descripción de la clase especialista

Nombre: especialista		
Descripción: Registro de los especialistas		
Atributo	Tipo	Descripción
idpersona	int(10) unsigned	Identificador de persona

idunidad_electromedicina	int(10) unsigned	Identificador unidad_electromedicina
idgraduado_de	int(10) unsigned	Identificador graduado de
idprovincia	int(10) unsigned	Identificador de provincia
idmunicipio	int(10) unsigned	Identificador de municipio
idplaza	int(10) unsigned	Identificador de plaza
idestado_civil	int(10) unsigned	Identificador de estado_civil
idcargo	int(10) unsigned	Identificador de cargo
idescala	int(11) unsigned	Identificador de escala
idcategoria	int(10) unsigned	Identificador de categoría
foto	varchar(255)	Imagen del especialista
padre	varchar(50)	Nombre del padre
madre	varchar(50)	Nombre de la madre
situación	varchar(255)	Situación que presenta el especialista
enfermedad	varchar(255)	Enfermedades que padece el especialista
peso	varchar(20)	Peso del especialista
tallacamisa	varchar(20)	Talla de camisa del especialista
nacimiento	text	Fecha de nacimiento del especialista
direccion	varchar(255)	Dirección particular del especialista
sexo	varchar(20)	Nombre sexo
registro	text	Fecha de registro del especialista
tallapantalon	varchar(20)	Talla de pantalón del especialista
calzado	int(10)	Número de calzado del especialista

Tabla 27 Descripción de la clase importador

Nombre: importador		
Descripción: Registro de importadores.		
Atributo	Tipo	Descripción
idimportador	int(10) unsigned	Identificador del importador
importador	varchar(50)	Nombre del importador.

Tabla 28 Descripción de la clase movimiento

Nombre: movimiento		
Descripción: Registros de movimientos		
Atributo	Tipo	Descripción
idmovimiento	int(10) unsigned	Identificador del movimiento
idpieza	int(10) unsigned	Identificador de la pieza
origen_idunidad_electromedicina	int(10) unsigned	Identificador de la unidad de Electromedicina de origen
idunidad_medida	int(10) unsigned	Identificador de la unidad de medida.
cantidad	float	Cantidad movida.
destino_idunidad_electromedinina	int(10)	Identificador de la unidad de Electromedicina destino.

Tabla 29 Descripción de la clase pedido

Nombre: pedido		
Descripción: Registro de los pedidos		
Atributo	Tipo	Descripción
idtipo_pieza	int(10)	Identificador del tipo de pieza

	unsigned	
idunidad_electromedicina	int(10) unsigned	Identificador de la unidad de Electromedicina
anno	int(10) unsigned	Año en que se produce
idunidad_medida	int(10) unsigned	Identificador de la unidad de medida
cantidad_1ro	float	Cantidad del primer trimestre
cantidad_2do	float	Cantidad del segundo trimestre
cantidad_3ro	float	Cantidad del tercer trimestre
cantidad_4to	float	Cantidad del cuarto trimestre

Tabla 30 Descripción de la clase pieza

Nombre: pieza		
Descripción:		
Atributo	Tipo	Descripción
idpieza	int(10) unsigned	Identificador de la pieza
idtipo_pieza	int(10) unsigned	Identificador del tipo de pieza
idimportador	int(10) unsigned	Identificador del importador
codigoimportador	varchar(50)	Código del importador
costo_mn	float	Costo en moneda nacional
costo_divisa	float	Costo en divisas
nocontrato	int(11)	Número de contrato
fecha_entrada	varchar(10)	Fecha de entrada.

Nota: El resto de las tablas están descritas en el expediente de proyecto.

3.4 Patrones de diseño empleados.

Los patrones GRASP acrónimo de General Responsibility Assignment Software Patterns (Patrones de Software para la asignación General de Responsabilidad).” *Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones*”. (8). En el desarrollo de la aplicación se utilizaron dichos patrones evidenciándose de la siguiente manera:

Patrón Experto: Asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Es un patrón que se usa más que cualquier otro al asignar responsabilidades; este principio suele utilizarse en el diseño orientado a objetos. En la aplicación se utiliza este patrón cuando se utiliza la clase Acceso_Datos que tiene responsabilidades específicas es la interacciona con los datos persistentes del sistema.

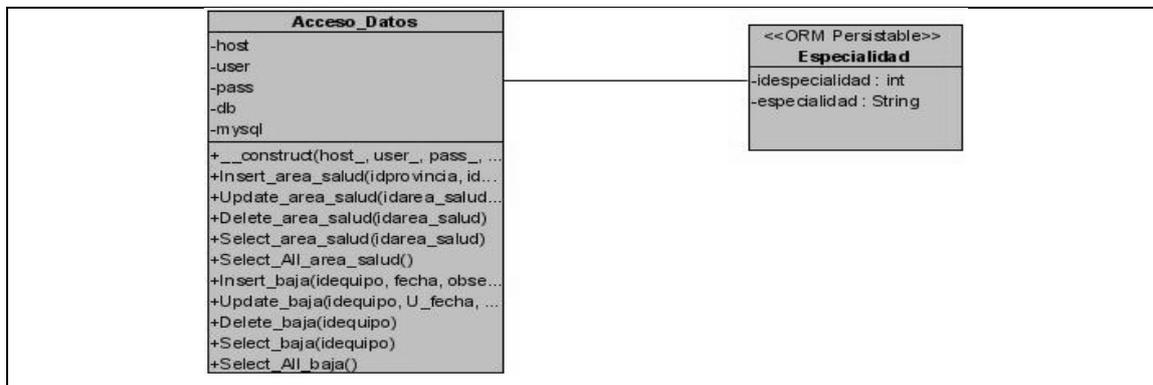


Figura 15 Ejemplo del Patrón Experto

Patrón Creador: Se asignaron responsabilidades a una clase de crear un objeto de otra clase, donde la creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

Ejemplo: en la clase CC_adicionar_pieza se crea un objeto de la clase Acceso_Datos para acceder a los métodos de esta Insert_pieza().

Alta Cohesión: Asignar una responsabilidad de modo que la cohesión siga siendo altas. Es un principio que debemos tener presente en todas las decisiones de diseño:

es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. El uso de este patrón se manifiesta fundamentalmente en las clases de acceso a datos, teniendo estas el mínimo de atributos y funciones a utilizar, para facilitar la reutilización y el mantenimiento de dichas clases a la hora de realizar cambios que puedan afectar la integridad del código. Cada elemento del diseño realiza una labor única dentro del sistema, no desempeñada por el resto de los elementos, es decir, se asignan responsabilidades a las clases de manera que todos sus métodos tengan un comportamiento bien definido. El uso de este patrón se evidencia en la clase de acceso a datos donde se encuentran el mínimo de atributos y funciones a realizar, permitiendo que sean reutilizadas y cualquier cambio de ella no influye en el resto de la aplicación.

Bajo Acoplamiento: Cada clase está acoplada a las clases estrictamente necesarias. Lo que está relacionado con patrones como Experto o Alta Cohesión, influyen en la decisión de asignar responsabilidades. El acoplamiento tal vez no sea tan importante, si no se busca la reutilización.

Ejemplo la clase CI_ adicionar_ pieza está acoplada a las clases necesarias para su funcionamiento como a la clase CC_ especialidad pero esta es a su vez utilizada por la clase CI_ adicionar_ tipo_ pieza.

3.5 Arquitectura empleada.

Para el desarrollo de la aplicación se asume la arquitectura definida por el Sistema Nacional de Salud la cual propone como patrón arquitectónico 3 capas: presentación, lógica o negocio y datos. Cada capa tiene relación con la siguiente ejemplo la capa de presentación solo tendría relación con la de negocio y si necesita acceder a la de datos lo haría a través de esta. Esto permite dividir mejor el trabajo y los recursos para la aplicación porque se puede desplegar cada capa por separado .

3.7 Diagrama de despliegue.

La aplicación estará dispuesta para tres centros regionales distribuidos por todo el país, los cuales se encuentran situados en Ciudad Habana (Región Occidental), Villa Clara (Región Central) y Holguín (Región Oriental) garantizando así una completa cobertura nacional. En cada centro regional el sistema estará desplegado como se muestra en el diagrama de despliegue. Estos centros regionales servirán al servidor central que se encuentra en el CNE. El principal objetivo de la distribución de la aplicación por los centros regionales es garantizar la continuidad del servicio en caso de existir problemas con la conexión. Luego de restablecida la conexión el sistema se actualizará mediante un sistema de sincronización implementado.

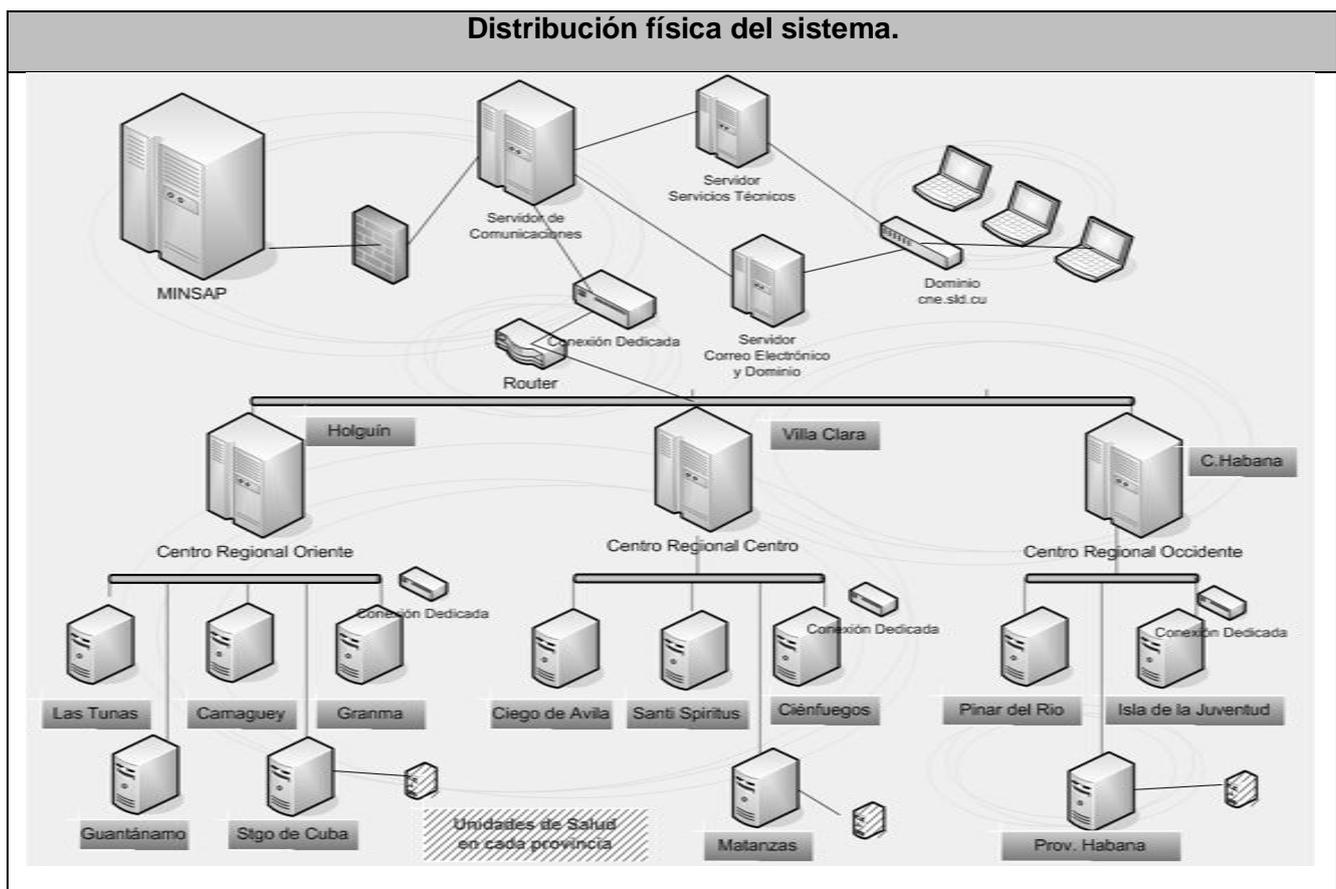


Figura 17 Distribución física del sistema.

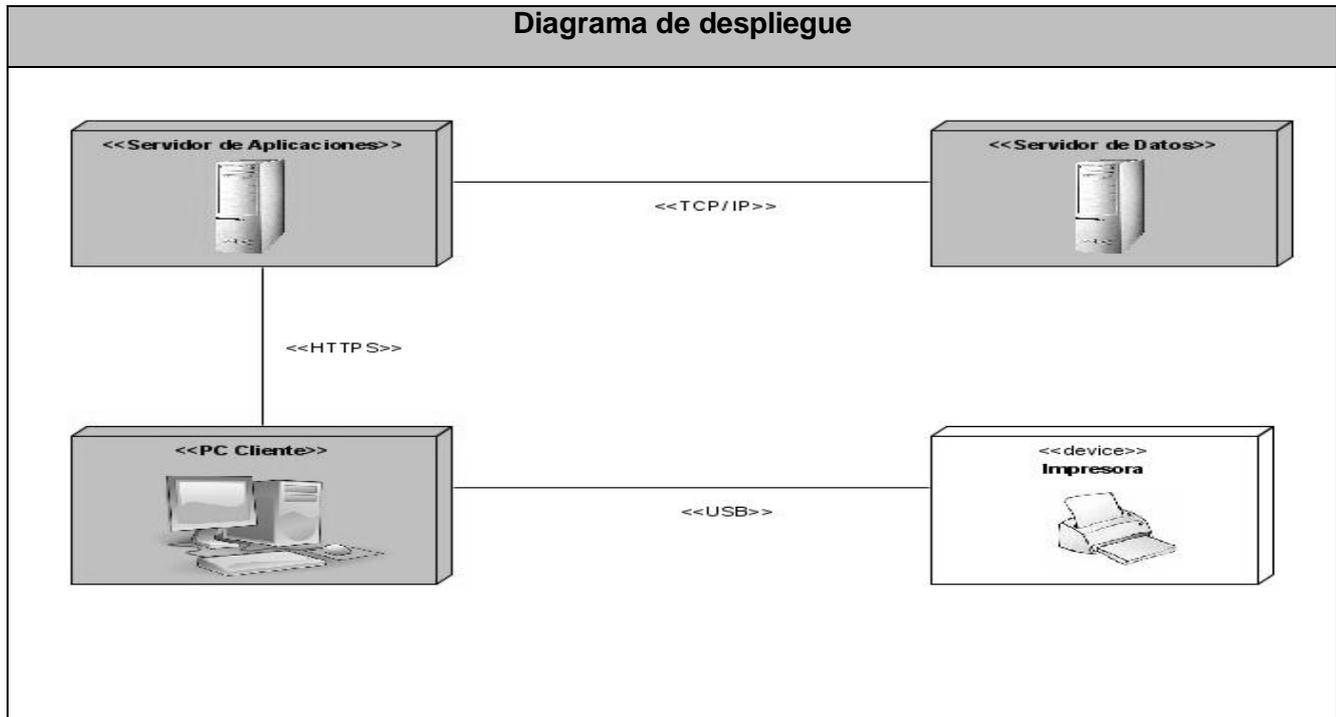


Figura 18 Diagrama de despliegue

3.8 Definiciones de diseño que se apliquen.

Para el desarrollo de una aplicación se llevan de la mano una serie de responsabilidades que dan un funcionamiento correcto y acorde a los requerimientos y gustos de los clientes. Para dar cumplimiento a dichos requerimientos se trabajó con formularios Web organizados, utilizando un mismo color, predominando el azul en todas las páginas y un mismo tipo y tamaño de letra con textos claros. Se utilizó Ajax para recibir y enviar información a la base de datos, evitando un continuo refrescamiento de las páginas a la hora de seleccionar algunos campos que necesitan información de la base de datos. Para garantizar estas responsabilidades se definió la aplicación de algunos de los patrones GRASP.

3.9 Tratamiento de errores.

La aplicación constará con un chequeo para los errores para cada uno de los campos existentes en cada una de las interfaces, esto se realiza mediante funciones de validación de datos desarrolladas en JavaScript. Con estas validaciones se garantiza que los datos que sean enviados no estén en blanco y tengan el formato correcto. Con el uso de las expresiones regulares en las funciones JavaScript se garantiza el formato correcto para campos que sean números, letras, correo electrónico y demás.

Cuando se solicita realizar alguna operación de eliminación se le pide confirmar la acción ya que estas son irreversibles. La aplicación garantizará que el usuario teclee la menor cantidad de datos posibles, para esto se contará con cuadros de opción, menú de selección lo cual facilitará la entrada de datos.

3.10 Seguridad.

La seguridad de la aplicación será responsabilidad del sistema en general (SIGICEM) el cual implementará una jerarquía de accesos para los diferentes usuarios del sistema ya que todos no tendrán los mismos privilegios sobre las posibles acciones a realizar.

Conclusiones.

En el presente capítulo se realizaron los diferentes diagramas de diseño y diagramas de secuencia por caso de uso, se describieron las clases utilizadas en el diseño. Se mostró el modelo de datos y las descripciones de las tablas en el mismo. También se hizo una definición del diseño aplicado, la forma de tratar los errores que pudieran ocurrir en la aplicación, la seguridad del sistema y la interfaz. Se mostró la posible distribución del sistema mediante el diagrama de despliegue.

CAPÍTULO 4: IMPLEMENTACIÓN.

Introducción.

Este capítulo abarca detalles de la fase de elaboración para la implementación del sistema. Se realiza el modelo de implementación con sus respectivos diagramas de componentes pertenecientes a los casos de uso definidos en las iteraciones anteriores.

4.1. Modelos de componentes.

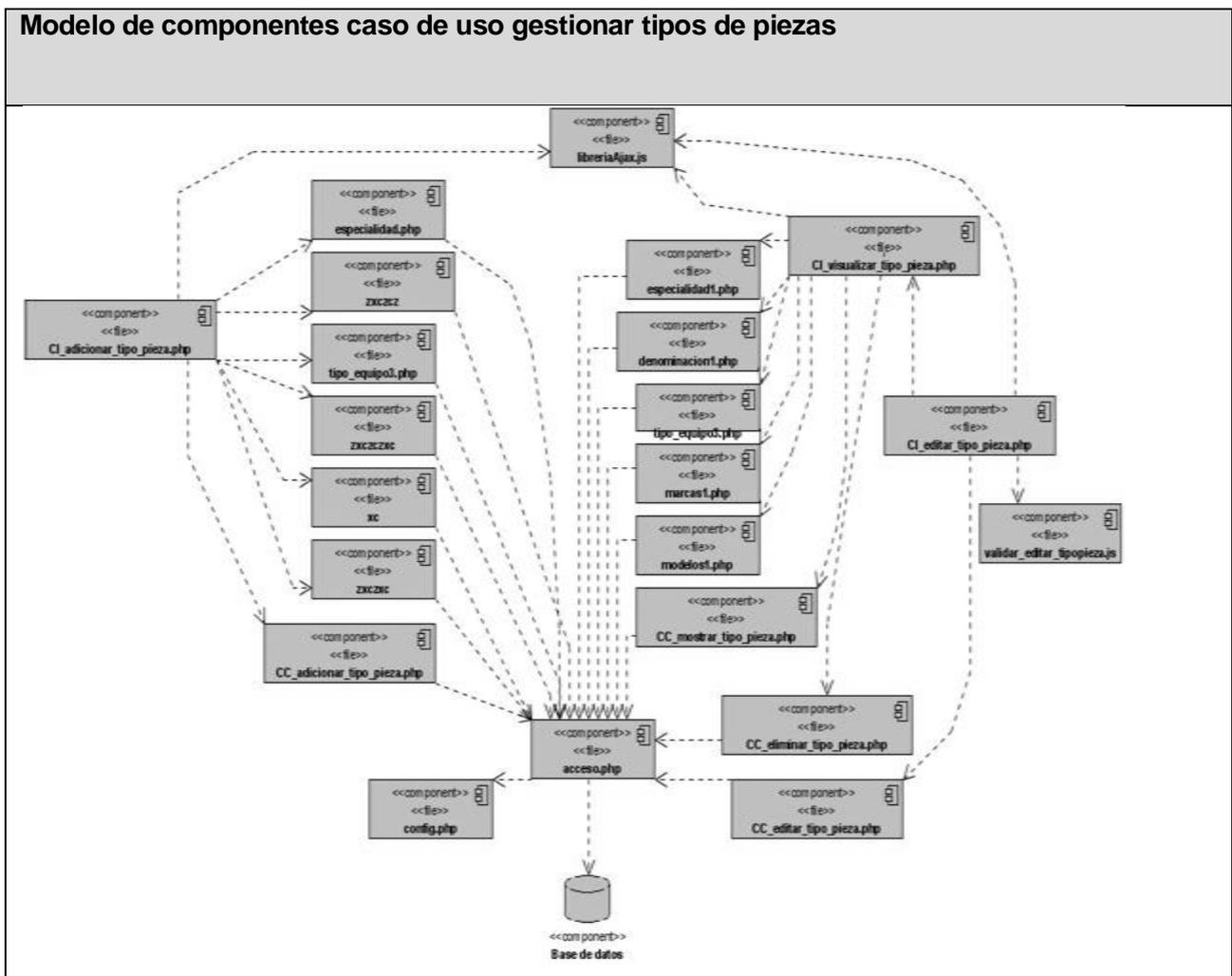


Figura 19 Modelo de componentes caso de uso gestionar tipos de piezas

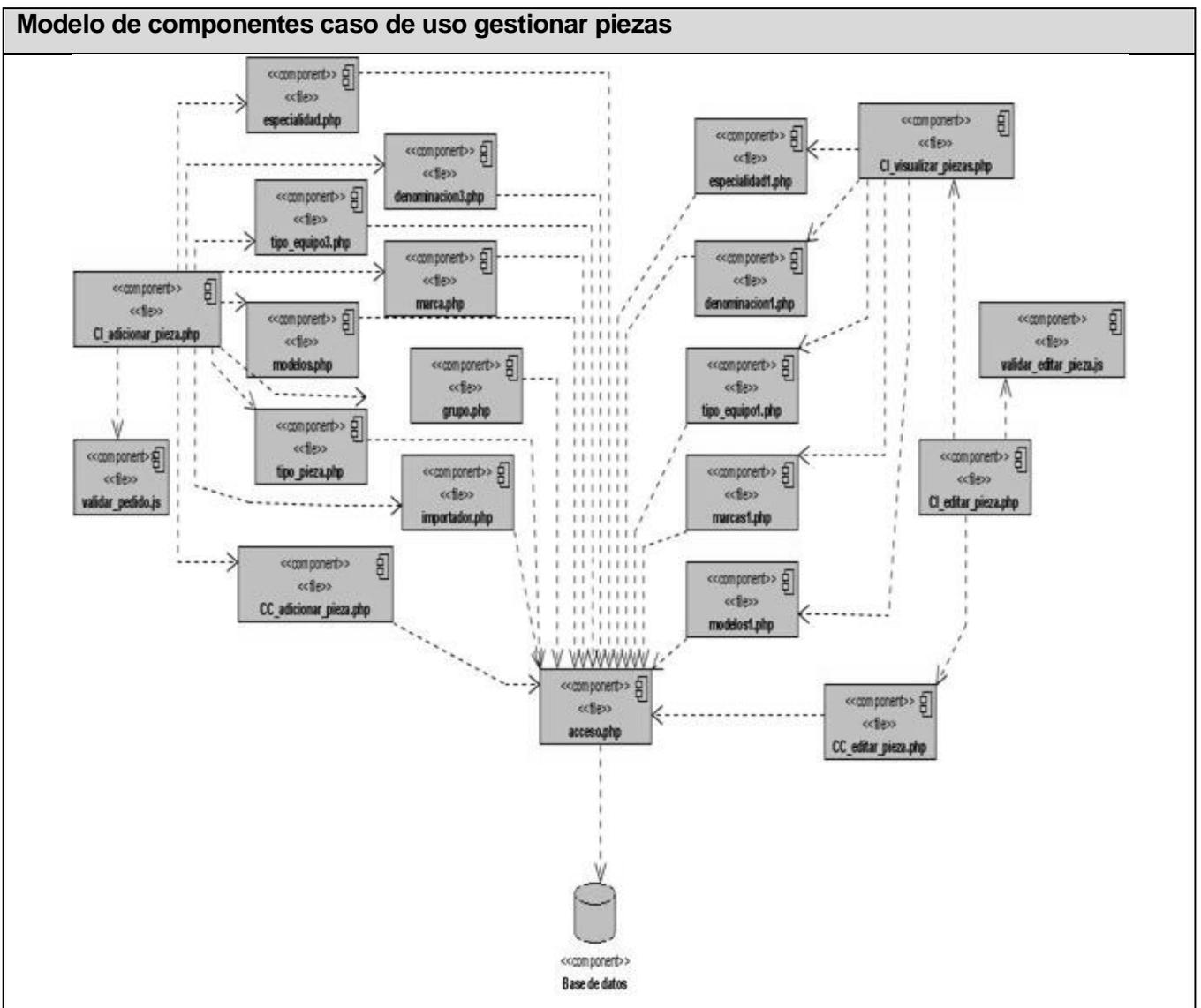


Figura 20 Modelo de componentes caso de uso gestionar piezas

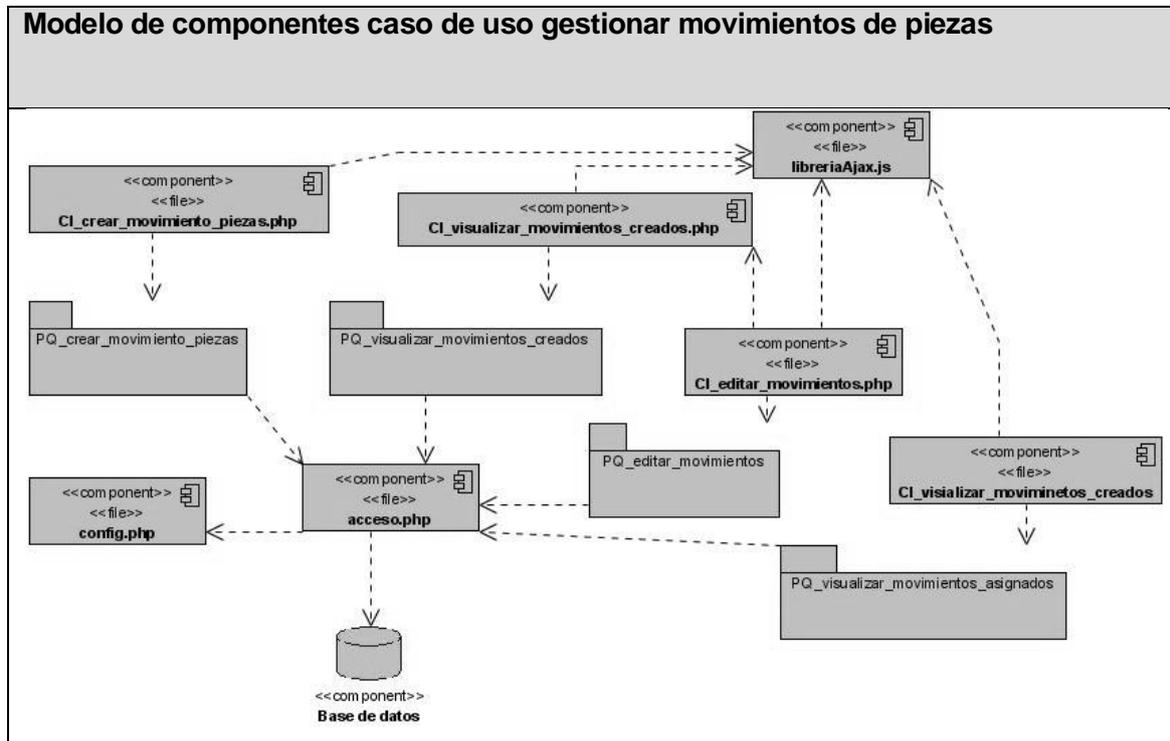


Figura 21 Modelo de componentes caso de uso gestionar movimientos de piezas

Nota: Las clases interfaz se relacionan con todos los componentes contenidos en los paquetes representados.

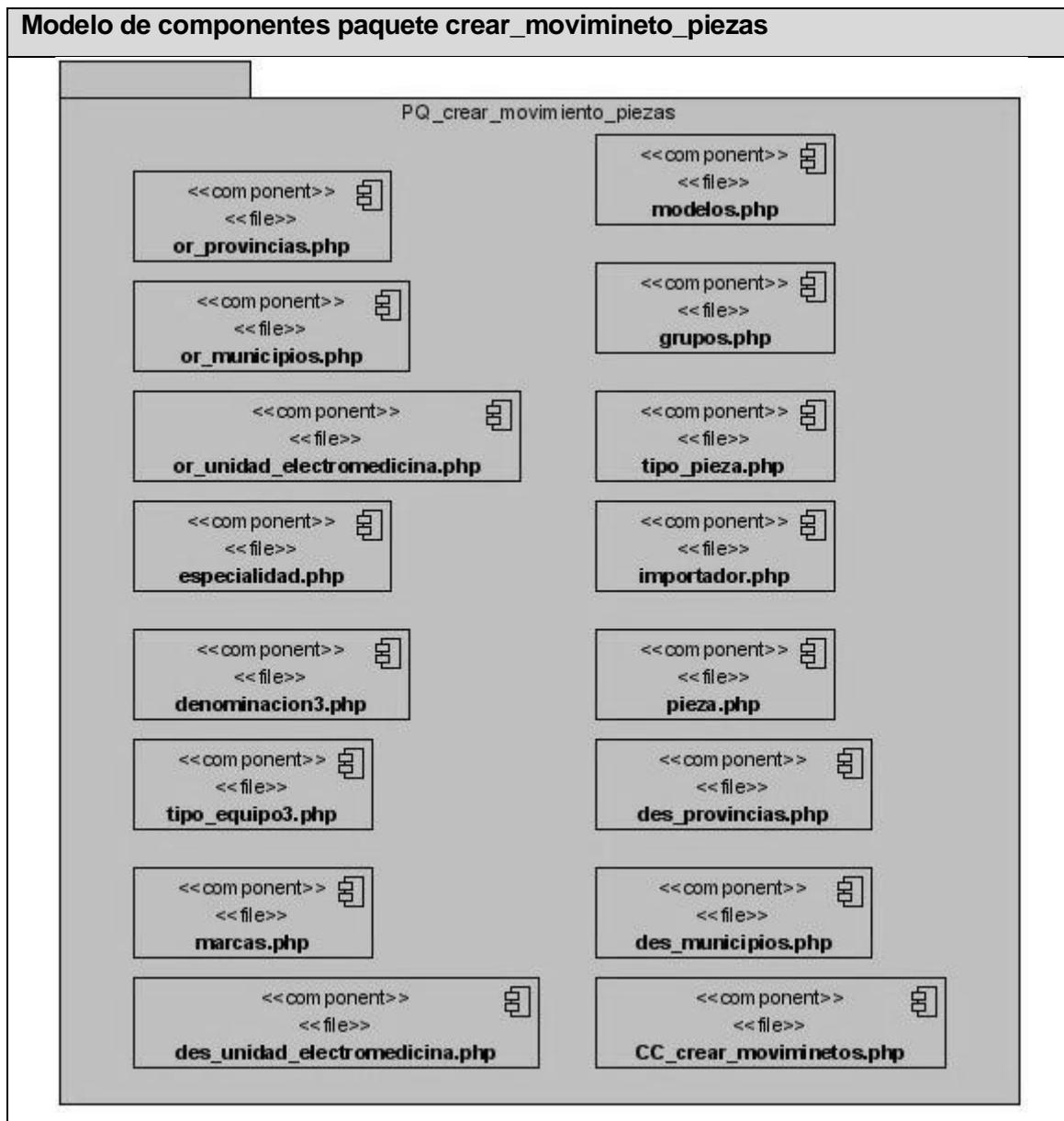


Figura 22 Modelo de componentes paquete crear_movimineto_piezas

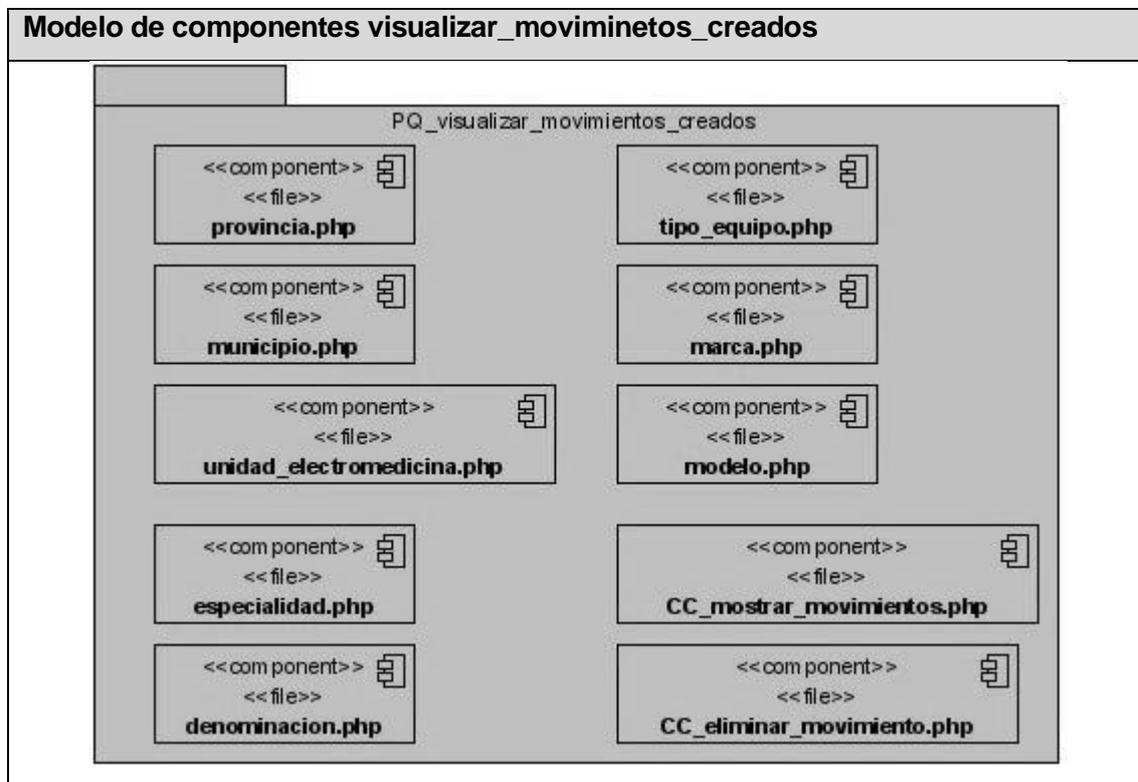


Figura 23 Modelo de componentes visualizar_moviminetos_creados

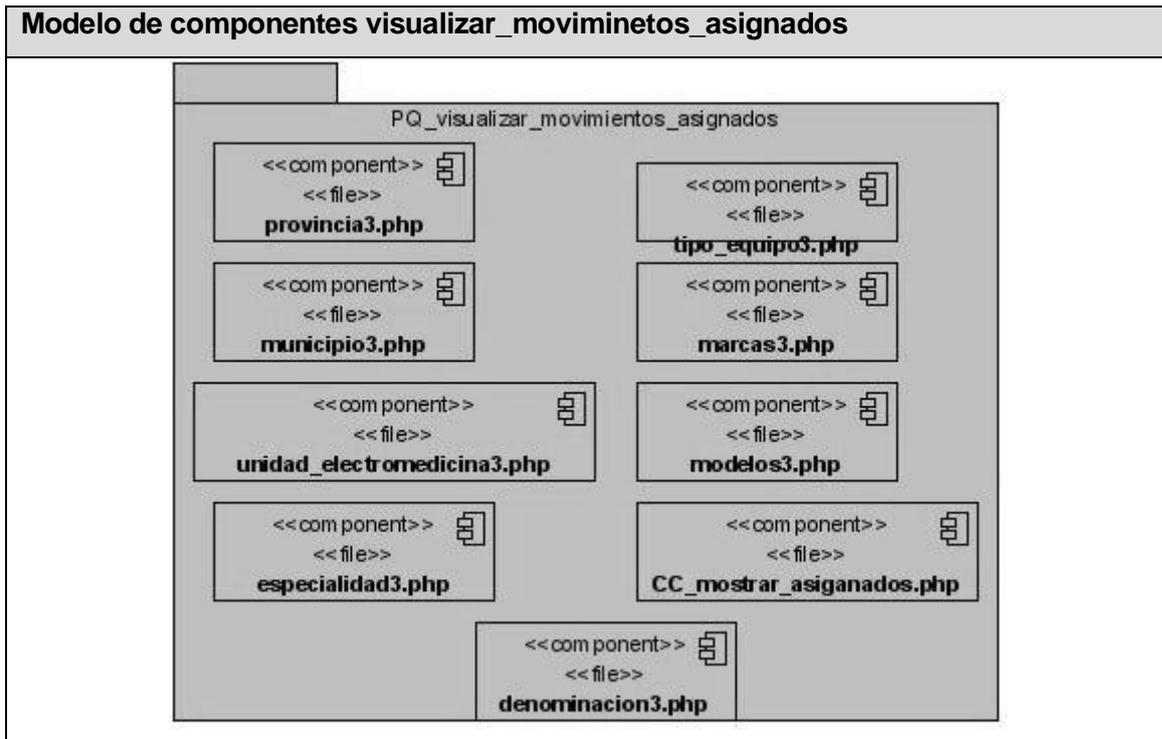


Figura 24 Modelo de componentes visualizar_moviminetos_asignados

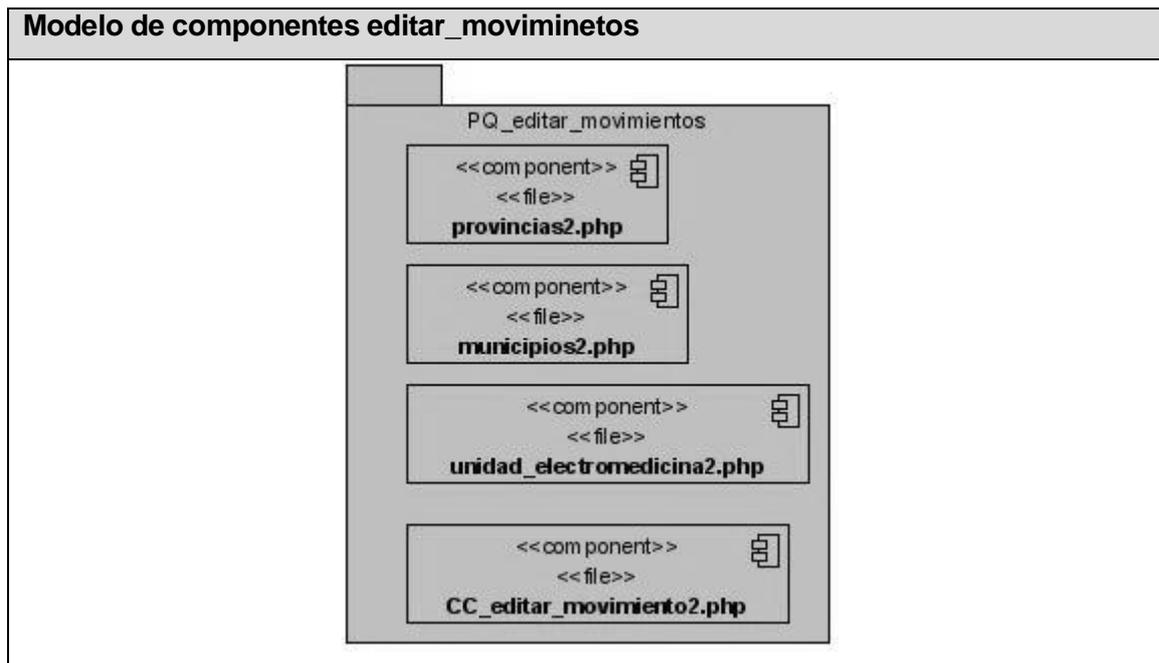


Figura 25 Modelo de componentes editar_moviminetos

Nota: El resto de los diagramas de componentes se pueden encontrar en el expediente de proyecto.

Conclusiones.

Este capítulo arrojó la forma en la que están distribuidos los diferentes componentes de la aplicación a través del modelo de componentes. También se mostraron los paquetes de componentes relacionados al caso de uso gestionar movimientos.

CONCLUSIONES

La culminación del análisis, diseño e implementación del módulo para la gestión de piezas de repuesto da solución a los problemas relacionados con el procesamiento y búsqueda de información, simplificando el trabajo de las personas.

Se considera además que el presente trabajo ha cumplido con los objetivos propuestos:

- Se definieron las funcionalidades del módulo para la gestión de la información de las piezas de repuesto.
- Se diseñó un módulo para la gestión de la información de las piezas de repuesto.
- Se diseñó una base de datos para almacenar información de las piezas de repuesto.
- Se implementó un módulo para la gestión de la información de las piezas de repuesto.

RECOMENDACIONES

Luego de haber concluido el presente trabajo se recomienda:

- Incluir en la ayuda del sistema general (SIGICEM) las funcionalidades de la aplicación para que sirva de apoyo a la navegación.
- Obtener una sólida integración del módulo al sistema (SIGICEM).
- Integración futura al RIS.
- Migrar a otro sistema gestor de BD.
- La utilización del material como bibliografía para las investigaciones futuras en esta área de desarrollo.
- Estudio de la posible migración a un framework de trabajo

REFERENCIA BIBLIOGRÁFICA

1. *Creacion de las condiciones para el uso masivo de las TIC.* [Online] MINREX, 2005. [Cited: 12 4, 2008.]
http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Masivo_TIC.htm.
2. *www.web.jet.es.* [Online] Noviembre 27, 06 . [Cited: 1 12, 2008.]
http://web.jet.es/amozarrain/sistema_gestion.htm.
3. Ramírez, MSc. Livia M. Reyes. *www.eprints.rclis.org.* [Online] Acimed, 2007. [Cited: enero 10, 2008.]
http://eprints.rclis.org/archive/00008960/01/Sistemas_de_informaci%C3%B3n_para_la_prensa.pdf.
4. Dante, Gloria Ponjuan. *Principios de la gestion de la informacion.*
<http://articulo.mercadolibre.com.ve>. [Online] marzo 12, 2006. [Cited: 12 05, 2007.]
<http://articulo.mercadolibre.com.ve>.
5. Lency, Nora. *www.winred.com.* [Online] septiembre 25, 2003. [Cited: enero 5, 2008.]
<http://winred.com/management/erp-exito-o-fracaso/gmx-niv116-con1953.htm>.
6. Chiesa, Florencia. *www.itba.edu.ar.* [Online] febrero 15, 2003. [Cited: 12 10, 2007.]
<http://www.itba.edu.ar/capis/webcapis/planma.html>.
7. DATAHOUSE COMPANY . *www.datahousecompany.com.ar.* [Online] enero 11, 2007. [Cited: 12 1, 27.] <http://www.datahousecompany.com.ar/comercial1.html>.
8. SapArgentina. *SAP Argentina.* [Online] [Cited: febrero 10, 2008.]
<http://www.sap.com/argentina/solutions/duet/index.epx>.
9. Booch, G., Rumbaugh, J., Jacobson. *El lenguaje Unificado de Modelado.* Addison-Wesley : s.n., 1999.
10. *Manual practico de HTML.* [Online] [Cited: 12 12, 2007.] <http://www.etsit.upm.es>.
11. [Online] [Cited: 12 14, 2007.] <http://www.webestilo.com/javascript>.
12. *TECNICOS LINUX .* [Online] Febrero 27t, 2006. [Cited: 12 7, 2007.]
<http://tecnicoslinux.com.ar/web/node/36>.
13. Parada, Indira Plá. *Portal digital para el Centro de control estatal de equipos medicos.* 2007.
14. —. *Portal digital para el Centro de control estatal de equipos medicos.* 2007.
15. Castellanos, Y. *Portal de las Misiones Sociales de la República Bolivariana de Venezuela.* [Online] Trabaja de diploma para optar por el título de Ingeniero

Informático, Instituto Superior Politécnico "Jose Antonio Echeverría", Ciudad de la Habana, junio 2005. [Cited: diciembre 3, 2007.]

16. Reyes, Ricardo Armando Machorro. www.revistaupiicsa.20m.com. [Online] Agosto 2004. [Cited: enero 12, 2008.]

<http://www.revistaupiicsa.20m.com/Emilia/RevMayAgo04/Machorro1.pdf>.

17. Pacheco., Alexander Quesada. *Sistema automatizado para el contro y gestión del transporte en el grupo de la electronica en el MIC*. Ciudad Habana : s.n., 2007.

BIBLIOGRAFÍA

[Fecha de consulta 24/3/2007].

[Online] [Cited: 12 14, 2007.] <http://www.webestilo.com/javascript>.

Booch, G., Rumbaugh, J., Jacobson. *El lenguaje Unificado de Modelado*. Addison-Wesley : s.n., 1999.

Castellanos, Y. *Portal de las Misiones Sociales de la República Bolivariana de Venezuela*. [Online] Trabajo de diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico "Jose Antonio Echeverría", Ciudad de la Habana, junio 2005. [Cited: diciembre 3, 2007.]

Chiesa, Florencia. www.itba.edu.ar. [Online] febrero 15, 2003. [Cited: 12 10, 2007.] <http://www.itba.edu.ar/capis/webcapis/planma.html>.
Clases de Ingeniería del Software I, curso 2005-2006, UCI.

Creacion de las condiciones para el uso masivo de las TIC. [Online] MINREX, 2005. [Cited: 12 4, 2008.]
http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Masivo_TIC.htm.

Dante, Gloria Ponjuan. *Principios de la gestion de la informacion*.
<http://articulo.mercadolibre.com.ve>. [Online] marzo 12, 2006. [Cited: 12 05, 2007.]
<http://articulo.mercadolibre.com.ve>.

DATAHOUSE COMPANY. www.datahousecompany.com.ar. [Online] enero 11, 2007. [Cited: 12 1, 27.] <http://www.datahousecompany.com.ar/comercial1.html>.

Definición de Software Libre. Disponible en <<http://www.gnu.org/philosophy/free-sw.es.html>> [Fecha de consulta 4/2/2007].

Introducción a php. Disponible en: < www.ciberteca.net/webmaster/php>

Primera Edición por Prentice Hall, *Hispanoamericana S.A.* 1999.

Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda Edición por Prentice Hall.

Lency, Nora. www.winred.com. [Online] septiembre 25, 2003. [Cited: enero 5, 2008.] <http://winred.com/management/erp-exito-o-fracaso/gmx-niv116-con1953.htm>.
Lenguajes de Programación: "Programación Web". Disponible en < <http://lenguajes-de-programacion.com/programacion-Web.shtml>> [Fecha de consulta 14/4/2007].

Manual practico de HTML. [Online] [Cited: 12 12, 2007.] <http://www.etsit.upm.es>. Pacheco., Alexander Quesada. *Sistema automatizado para el control y gestión del transporte en el grupo de la electrónica en el MIC*. Ciudad Habana : s.n., 2007.

Ponjuán, *Gestión de Información en las organizaciones*.

Ramírez, MSc. Livia M. Reyes. www.eprints.rclis.org. [Online] Acimed, 2007. [Cited: enero 10, 2008.] http://eprints.rclis.org/archive/00008960/01/Sistemas_de_informaci%C3%B3n_para_la_prensa.pdf.

Reyes, Ricardo Armando Machorro. www.revistaupiicsa.20m.com. [Online] Agosto 2004. [Cited: enero 12, 2008.] <http://www.revistaupiicsa.20m.com/Emilia/RevMayAgo04/Machorro1.pdf>.

SapArgentina. SAP Argentina. [Online] [Cited: febrero 10, 2008.] <http://www.sap.com/argentina/solutions/duet/index.epx>.

TECNICOS LINUX . [Online] Febrero 27t, 2006. [Cited: 12 7, 2007.] <http://tecnicoslinux.com.ar/web/node/36>.

www.web.jet.es. [Online] Noviembre 27, 06 . [Cited: 1 12, 2008.] http://web.jet.es/amozarrain/sistema_gestion.htm.

GLOSARIO DE TÉRMINOS

AJAX: (Asynchronous Javascript And XML): Es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

CASE: (Computer Aided Software Engineering), (Ingeniería de Software Asistida por Ordenador): Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo.

CNE: (Centro Nacional de Electromedicina): Tiene sus inicios después del triunfo de la Revolución en el año 1961, fundado por Nuestro Comandante en jefe Fidel Castro Ruz. El país contaba con pocos profesionales en el sector los cuales trabajaban en pequeñas fábricas privadas que producían equipos médicos y el estado los fue reuniendo para formar los primeros talleres de reparación provinciales que ya para el año 64 prestaban algunos servicios.

CRP: (Capacity Resource Planning): Software que se utilizan para planificar los insumos medidos en unidades de tiempo (horas, minutos).

CU: (Casos de uso): Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

ERP: (Enterprise Resource Planning): La Planificación de Recursos Empresariales, o simplemente ERP, es un conjunto de sistemas de información gerencial que permite la integración de ciertas operaciones de una empresa, especialmente las que tienen que ver con la producción, la logística, recursos humanos, el inventario, los envíos y la contabilidad

GADE: (Grupo de Automatización y Desarrollo de Electromedicina): Fundado el 28 de junio del 2006 con la misión de: Apoyar la gestión informativa, investigativa y administrativa del Centro Nacional de Electromedicina, utilizando las herramientas que ofrecen las tecnologías de la informática y las telecomunicaciones.

GRASP: General Responsibility Assignment Software Patterns (Patrones de Software para la asignación General de Responsabilidad).

HTML: (HyperText Markup Language), (Lenguaje de marcas hipertextuales): Lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

MRP: (Material Requirement Planning): Software que se basan en soluciones de gestión de abastecimientos y se limitan a la planificación de las necesidades de consumos (materias primas, materiales, etc.) dejando el planeamiento de equipos a las filosofías CRP.

OBE: (Organizaciones Básicas de Electromedicina).

PHP: (Hypertext PreProcessor): Lenguaje de programación para el desarrollo de web dinámicas, con sintaxis parecida a la C. Originalmente se conocía como *Personal Hombre Page tools*, herramientas para páginas personales (en Internet).

RIS: (Registro Informatizado de Salud).

RRHH: (Recursos Humanos): Departamento de Recursos Humanos.

SIGEM: (Sistema de Gestión para Electromedicina): Sistema diseñado para el uso exclusivo del Sistema Nacional de Electromedicina por las prestaciones que tiene que van desde el Aseguramiento Técnico, el Inventario, la Gestión de la Estadística Técnica hasta la Gestión de los Recursos Humanos.

SIGICEM: (Sistema de Gestión para la Ingeniería Clínica y Electromedicina).

SNS: (Sistema Nacional de Salud).

Tecnología Médica: No es más que los medicamentos, los equipamientos, los procedimientos médicos y quirúrgicos utilizados en la atención médica y los sistemas organizativos con los que se presta la atención sanitaria. Por lo tanto, la tecnología médica no son sólo las máquinas o medicamentos, sino también la propia práctica clínica y el modo en que esta se organiza. (16)

UML: (Unified Modeling Language): Es un lenguaje de modelado para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir.

