



UNIVERSIDAD DE SANCTI SPIRITUS "JOSÉ MARTÍ PÉREZ"
FACULTAD DE CIENCIAS TÉCNICAS
CARRERA DE INGENIERÍA INFORMÁTICA

TRABAJO DE DIPLOMA
PARA OPTAR POR EL TÍTULO DE INGENIERÍA INFORMÁTICA

Título:

Sistema Para el control de pacientes en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico "José Martí Pérez de Sancti Spiritus".

Autor:

Carlos Arturo Barberá Flores

Tutores:

Ing. Yonelky Iznaga Gonzalez

Ing. Reynel Rodríguez Morgado

Sancti Spiritus, 2015

"Año 57 de la revolución"



Pensamiento



“La Informática se convertirá en una poderosísima fuerza científica, económica e incluso política del país...”

Fidel Castro



Dedicatoria

Este Trabajo se lo dedico a las personas más importantes en mi vida:

♥ *A mis padres:*

Quienes me dieron la vida y me enseñaron a caminar en ella.

A mi papá, por su ayuda, comprensión, amor.

A mi mamá, que me ha amado toda la vida, sin importar nada.

♥ *A mi esposa:*

Porque no importa lo que pase siempre está a mi lado brindándome apoyo y mucho amor.

A todos ustedes porque son mi razón de ser

A decorative graphic consisting of a blue wavy ribbon that flows vertically down the right side of the page, then curves horizontally across the bottom. The ribbon has a 3D effect with a lighter blue shadow underneath. The word "Agradecimientos" is written in a blue cursive font across the middle of the ribbon.

Agradecimientos

Agradecimientos

- ✓ *A mis padres por estar ahí siempre que los necesité, por darme su apoyo y sobre todo por darme su confianza.*

- ✓ *A mi abuela, que siempre ha estado ahí cuando la necesite, y por darme todo su cariño.*

- ✓ *A mi prima, que siempre ha sido mi ejemplo a seguir.*

- ✓ *A mi esposa, por estar siempre conmigo dándome su amor y haciéndome una mejor persona.*

- ✓ *A mis tutores por siempre brindarme su ayuda y compartir conmigo tan buenos consejos.*

- ✓ *A todos los trabajadores de Departamento de Registros Médicos y Estadística del Hospital Pediátrico de Sancti Spiritus por permitirme formar parte de ellos en estos últimos meses.*

- ✓ *A mis compañeros de aula por apoyarme en estos cinco años.*

A todos ustedes muchas gracias...

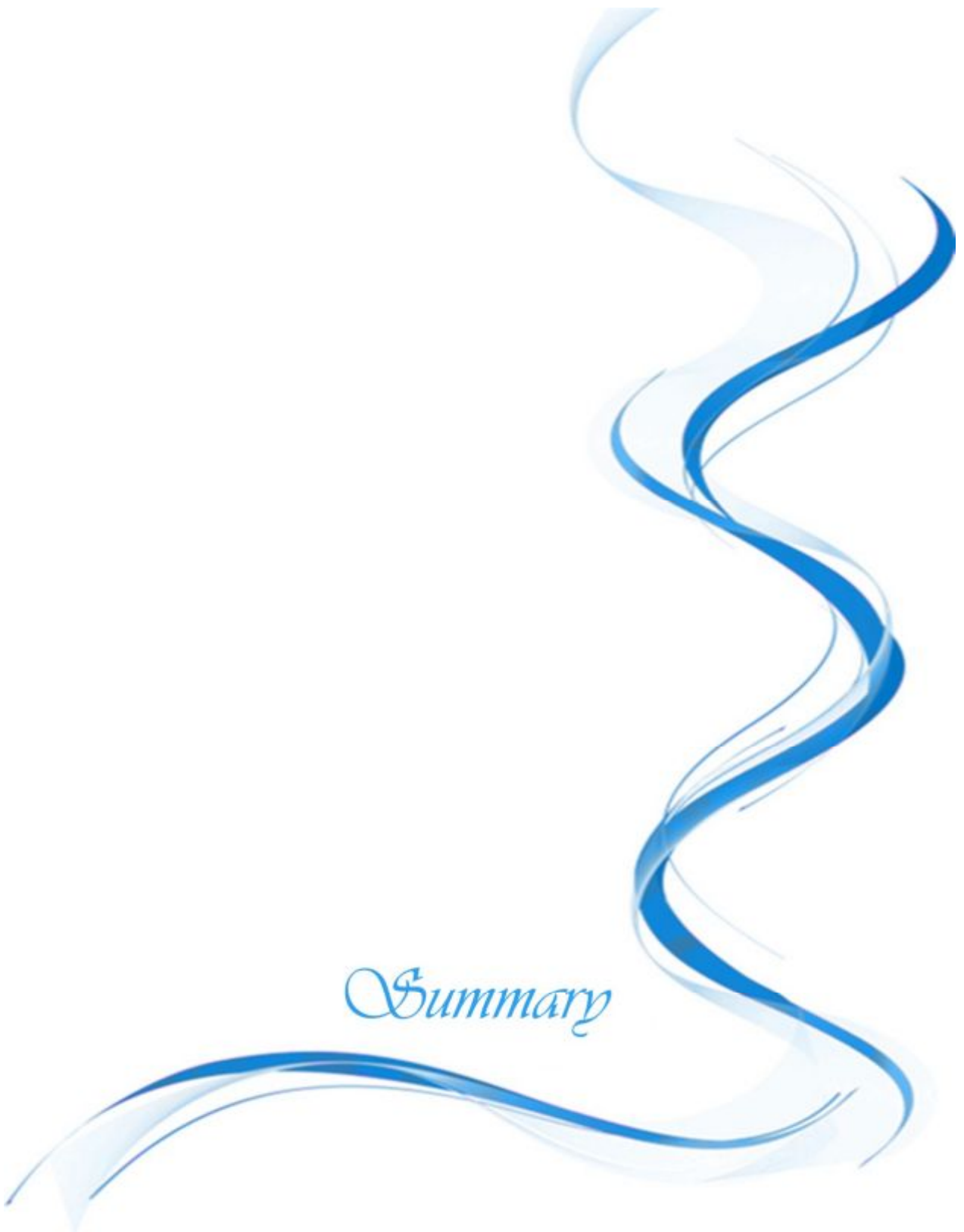


Resumen

Resumen

Este trabajo está dirigido a el Departamento de Registros Médicos y Estadística del Hospital Pediátrico en la provincia de Sancti Spíritus, donde actualmente el proceso de gestión de la información relacionada con los pacientes, la cirugías, las enfermedades se efectúa manualmente y los documentos que se encuentran de manera digital están en diferentes formatos; lo que ha ocasionado problemas para acceder a la información. Es por ello que el objetivo de esta investigación es desarrollar un software que perfeccione la gestión de los pacientes en este departamento.

Las etapas llevadas a cabo en la elaboración del software fueron las planteadas por el Proceso Unificado de Desarrollo (RUP), utilizándose como lenguaje al “Lenguaje Unificado de Modelado” (UML) y como herramienta para el análisis y diseño: el Visual Paradigm para UML 8.0 Enterprise Edition. Para implementar el sistema se utilizó Java como lenguaje de programación lo que proporcionó un código eficiente, para la persistencia de los datos se utilizó PostgreSQL versión 8.4 y como herramienta de programación NETBEANS versión 6.9



Summary

Summary

This work takes place in the Medical Records Department at Children's Hospital in the province of Sancti Spiritus, where currently the management information process related to patients, surgeries, and illnesses is done manually and documents that are so digital are in different formats; which it has caused problems accessing information. That is why the objective of this research is to develop software to refine the management of patients in this department.

The steps taken in the development of the software were posed by the Rational Unified Process (RUP), using as a language the "Unified Modeling Language" (UML) and as a tool for analysis and design: the Visual Paradigm for UML Enterprise Edition 8.0. To implement the system was used as the programming language Java which provided efficient code, for data persistence was used PostgreSQL version 8.4 and as a programming tool NETBEANS version 6.8.

A decorative graphic consisting of a blue wavy ribbon that flows from the top right, down the right side, and then curves back towards the bottom left. The ribbon has a 3D effect with a lighter blue shadow underneath. The word "Índice" is written in a blue, elegant script font in the center of the page, overlapping the ribbon.

Índice

Índice

<i>Resumen</i>	1
<i>Summary</i>	1
<i>Introducción</i>	1
<i>Capítulo 1: Fundamentos teóricos y metodológicos que sustentan el desarrollo de una herramienta informática que perfeccione la gestión de la información sobre los pacientes.</i>	5
<i>Introducción</i>	5
1.1 <i>Gestión de la Información</i>	5
1.2 <i>Registros Médicos y Estadísticos</i>	5
1.3 <i>Antecedentes de los sistemas de gestión de pacientes</i>	6
1.4 <i>Metodologías para el desarrollo de software</i>	8
1.4.1 <i>Metodologías ágiles y tradicionales</i>	8
1.4.2 <i>Metodología Rational Unified Process (RUP)</i>	8
1.4.3 <i>Lenguaje de Modelado Unificado (UML)</i>	9
1.5 <i>Tecnologías para la elaboración de software</i>	10
1.5.1 <i>Aplicaciones de escritorio</i>	10
1.5.2 <i>Aplicaciones Web</i>	11
1.5.3 <i>Arquitectura en n Capas</i>	11
1.6 <i>Lenguajes de Programación</i>	12
1.6.1 <i>C++</i>	12
1.6.2 <i>C#</i>	13
1.6.3 <i>Java</i>	14
1.7 <i>Sistemas Gestores de Base de Datos</i>	15
1.7.1 <i>MySQL</i>	15
1.7.2 <i>PostgreSQL</i>	16
1.8 <i>Herramientas de Desarrollo</i>	18
1.8.1 <i>NetBeans</i>	18

1.8.2 Eclipse	18
Conclusiones.....	18
 <i>Capítulo 2: Descripción de la aplicación propuesta para el control de los pacientes en el departamento de registros médicos y estadística del hospital pediátrico de Sancti Spiritus.</i>	
Introducción.....	19
2.1 Modelo del Negocio.....	19
2.1.1 Identificación de los procesos del negocio	19
2.1.2 Reglas del negocio a considerar	20
2.1.3 Modelo de casos de uso del negocio	20
2.1.4 Actores del Negocio.....	20
2.1.5 Trabajadores del Negocio	21
2.1.6 Diagrama de Casos de Uso del Negocio.....	21
2.1.7 Descripción textual de los casos de uso del negocio	21
2.1.8 Diagramas de Actividades	22
2.1.9 Modelo de objetos del negocio.....	23
2.2 Requerimientos	24
2.2.1 Requerimientos funcionales.....	24
2.2.2 Requerimientos no funcionales	26
2.3 Modelo del Sistema	28
2.3.2 Actores del sistema.....	28
2.3.3 Casos de uso del sistema	29
2.3.4 Diagrama de casos de uso del sistema.....	31
Conclusiones.....	36
 <i>Capítulo III Implementación del software Sistema Gestor de Pacientes del Hospital Pediátrico de Sancti Spíritus:.....</i>	
Introducción:.....	37
3.1 Diagramas de clases del diseño.....	37
3.2 Diseño de la base de datos	37

3.2.1 Diagrama de clases persistentes	37
3.3 Principios de diseño	38
3.3.1 Interfaz.....	38
3.3.2 Tratamientos de errores.....	39
3.3.3 Concepción de la ayuda.....	39
3.3.4 Seguridad	39
3.4 Estándares de codificación.....	39
3.5 Modelo de implementación.....	40
3.5.1 Modelo de despliegue.....	40
3.5.2 Diagrama de componentes.....	41
3.6 Pruebas.....	42
3.6.1 Planificación de pruebas.....	42
3.6.2 Procedimientos de prueba	43
3.6.3 Casos de pruebas.....	45
Conclusiones.....	46
Conclusiones	48
Recomendaciones	49
Bibliografía.....	50
Anexos.....	57

A decorative graphic consisting of a blue wavy ribbon that flows vertically down the right side of the page, then curves horizontally across the bottom. The ribbon has a 3D effect with a lighter blue shadow underneath.

Índice de Tablas

Índice de Tablas

<i>Tabla 1 Actores del Negocio.....</i>	<i>21</i>
<i>Tabla 2 Trabajadores del negocio.....</i>	<i>21</i>
<i>Tabla 3 Descripción textual del caso de uso del negocio: Realizar informes.....</i>	<i>22</i>
<i>Tabla 4 Actores del sistema.....</i>	<i>28</i>
<i>Tabla 5 Descripción de los paquetes</i>	<i>33</i>
<i>Tabla 6 Descripción de los componentes.....</i>	<i>42</i>
<i>Tabla 7 Caso de prueba para el caso de uso Gestionar paciente de cuerpo de guardia.....</i>	<i>46</i>
<i>Tabla 8 Caso de prueba para el caso de uso Gestionar paciente de cirugía.....</i>	<i>46</i>
<i>Tabla 9 Caso de prueba para el caso de uso Gestionar cirugía.....</i>	<i>46</i>
<i>Tabla 10 Caso de prueba para el caso de uso Mostrar los pacientes operados en un día determinado.....</i>	<i>46</i>

A decorative graphic consisting of a blue wavy ribbon that flows vertically down the right side of the page and then curves horizontally across the bottom. The ribbon has a 3D effect with a lighter blue shadow underneath.

Índice de figuras

Índice de Figuras

<i>Figura 1 Diagrama de casos de uso del negocio</i>	<i>21</i>
<i>Figura 2 Diagrama de Actividades del caso de uso Realizar Informe.....</i>	<i>23</i>
<i>Figura 3 Modelo de objetos del negocio.....</i>	<i>23</i>
<i>Figura 4 Diagrama de casos de uso por paquetes.....</i>	<i>33</i>
<i>Figura 5 Diagramas de casos de uso: Paquete Seguridad.....</i>	<i>34</i>
<i>Figura 6 Diagramas de casos de uso: Paquete Administración.....</i>	<i>34</i>
<i>Figura 7 Diagramas casos de uso: Paquete Gestión</i>	<i>35</i>
<i>Figura 8 Diagramas casos de uso: Paquete Reportes</i>	<i>35</i>
<i>Figura 9 Diagramas casos de uso: Paquete Reportes</i>	<i>36</i>
<i>Figura 10 Diagrama de clases persistentes</i>	<i>38</i>
<i>Figura 11 Diagrama de despliegue</i>	<i>41</i>
<i>Figura 12 Diagrama de componentes</i>	<i>41</i>

A decorative graphic consisting of a blue wavy ribbon that flows vertically down the right side of the page, then curves horizontally across the bottom. The ribbon has a 3D effect with a lighter blue shadow underneath. The word "Introducción" is written in a blue, cursive font across the middle of the horizontal part of the ribbon.

Introducción

Introducción

El crecimiento exponencial del desarrollo científico y tecnológico ha provocado un vertiginoso incremento en la producción de conocimientos, cuya transmisión adquiere cada vez mayor importancia en el mundo actual.

La ciencia y la tecnología ayudan a aumentar y mejorar el rendimiento de las personas y de las organizaciones. La informatización de los procesos actualmente es de gran importancia, ya que ayudan a mejorar todo lo relacionado con lo social y económico, dentro del entorno tecnológico.

A medida que las empresas crecen y se desarrollan son más ineludibles las necesidades que se presentan en el día a día ya que se requiere mayor control de personal, por otro lado se debe tener un control preciso más minucioso de todos los procesos que desarrolla la empresa en su quehacer laboral.

Debido a que todas estas actividades son de gran importancia se desarrollan diferentes técnicas para controlarlas, y para muchos de estos casos se hace uso de la tecnología y la informática ya que se ha comprobado que ayudan a agilizar los procesos, también a que sean más eficientes, igualmente ayuda a mejorar el orden de la empresa.

Esta es la Era de la Información y de las Tecnologías Informáticas. Poseer información, generarla, transformarla, almacenarla y transmitirla es sinónimo de poder y progreso. Las nuevas características que el espacio digital ha incorporado a estas tecnologías han hecho que sean conocidas como Tecnologías de la Información y las Comunicaciones (TIC), que sin dudas, han llegado para quedarse.

De esta forma, las TIC se han convertido en uno de los recursos más importantes de la sociedad, trayendo como consecuencia un avance superior en la transmisión e intercambio de datos, información y conocimientos, a los cuales se puede acceder sin tener en cuenta barreras geográficas o limitaciones del tiempo, transformando elementos fundamentales que condicionan la comunicación tales como el espacio y el tiempo, a lo que se adiciona la capacidad de interacción en el proceso de comunicación. Es por ello que en los últimos años casi todos los países del mundo han establecido e implementado proyectos, políticas y estrategias para promover el uso de las TIC y aprovechar los beneficios y los aportes que estas ofrecen para las organizaciones y la sociedad en general.

Las compañías para desarrollarse necesitan manejar gran cantidad de datos a la vez y gracias a los sistemas de gestión estos es posible, es decir son una herramienta para mantener el control de los datos de manera eficaz, veraz, y oportuna.

Para enfrentar desafíos cotidianos, es necesario cubrir las necesidades de la información, aunque la estadística, la computación y la ingeniería están muy desarrolladas, casi nunca se posee toda la información necesaria para tomar decisiones. Esta dinámica, ligada a la necesidad de obtener utilidades, obliga a las empresas a requerir información más eficiente, y a diseñar sistemas que permitan conseguir y gestionar de la mejor manera los sistemas administrativos.

El sector de la salud es uno de los muchos que implementa el uso de las TIC debido a que es un sector con un alto grado de regulación, de gestión fundamentalmente pública, altamente fragmentado y está muy influido por la información. La calidad y la viabilidad económica de una asistencia tal como demanda la sociedad actual depende de la incorporación efectiva de las nuevas tecnologías. La incorporación de las TIC al mundo sanitario está suponiendo la mejora de la calidad de vida de los ciudadanos, favoreciendo el desarrollo de herramientas en áreas como la investigación, gestión, planificación, información, prevención o en el diagnóstico.

Una de las organizaciones existentes en Cuba interesadas en el uso de las TIC es el Ministerio de Salud Pública (MINSAP), donde el uso de las TIC constituye una herramienta para mejorar el impacto de las intervenciones en la comunidad, permitiendo un acceso más equitativo y eficiente a los servicios, mejorando la oportunidad de la atención y la costo-efectividad de los tratamientos.

La recolección, procesamiento y diseminación de la información en salud es responsabilidad de la Dirección Nacional de Registros Médicos y Estadística de Salud del Ministerio de Salud Pública de Cuba (DNE-MINSAP). La información generada abarca áreas tales como demografía, mortalidad, morbilidad, recursos, servicios y formación de personal calificado. El Sistema Estadístico de Salud de Cuba utiliza el método de registro continuo y la aplicación de encuestas por muestreo, posee cobertura nacional, con departamentos de estadísticas y registros médicos en todas las unidades e instituciones de salud, estructuras municipales, provinciales y nación, donde se ubica la Dirección Nacional de Registros Médicos y Estadísticas de Salud del MINSAP (Alonso & Zacca, 2011).

Específicamente en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico “José Martí y Pérez” de Sancti Spiritus, el proceso de gestión de los datos relacionados con los pacientes y las enfermedades se torna muy complejo, debido a la no existencia de una herramienta informática que agilice dicho proceso y los especialistas deben dedicar más tiempo y esfuerzo al realizar todo el proceso de forma manual.

Lo anterior mencionado trae consigo la ineficiencia y la lentitud en el proceso. En los diferentes informes que se realizan existe repetición innecesaria de los datos, pérdida de información y falta de seguridad. Además existe la dificultad en el acceso a los datos y el manejo de los mismos, ya que al hacerse en formato duro existe un gran volumen de información casi imposible de consultarse a mano por los especialistas.

*De la situación descrita anteriormente se identifica como **Problema de Investigación:***

¿Cómo contribuir a la gestión de la información de los pacientes en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico “José Martí Pérez” de Sancti Spíritus?

*Para ello esta investigación establece como **Objetivo General:***

Desarrollar una aplicación de escritorio que contribuya a la gestión de la información de los pacientes en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico “José Martí Pérez” de Sancti Spíritus.

*Para dar cumplimiento al Objetivo General surgen las siguientes **Preguntas de Investigación:***

- 1. ¿Qué fundamentos teóricos y metodológicos sustentan la elaboración de una herramienta informática que contribuya a la gestión de la información de los pacientes?*
- 2. ¿Cómo diseñar una aplicación de escritorio que contribuya a la gestión de la información de los pacientes en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico “José Martí Pérez” de Sancti Spíritus?*
- 3. ¿Cómo implementar una aplicación de escritorio que contribuya a la gestión de la información de los pacientes en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico “José Martí Pérez” de Sancti Spíritus?*

*Para darle respuesta a las preguntas de investigación se plantean las siguientes **Tareas de Investigación:***

- 1. Determinar los fundamentos teóricos y metodológicos que sustentan la elaboración de una herramienta informática que contribuya a la gestión de la información de los pacientes.*
- 2. Diseñar una aplicación de escritorio que contribuya a la gestión de la información de los pacientes en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico “José Martí Pérez” de Sancti Spíritus.*
- 3. Implementar la aplicación de escritorio que contribuye a la gestión de la información de los pacientes en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico “José Martí Pérez” de Sancti Spíritus.*

La investigación queda estructurada en tres capítulos, un primer capítulo que contiene los fundamentos teóricos y metodológicos que sustentan la investigación, resumido en el hilo conductor que lo organiza; el segundo capítulo está dirigido a diseñar una aplicación de escritorio que contribuya a la solución del problema, para luego en el tercer capítulo mostrar los resultados del diseño de la aplicación de escritorio, con las conclusiones generales, recomendaciones, la bibliografía consultada y finalmente los anexos con información complementaria de los resultados obtenidos.

A decorative graphic consisting of a blue wavy ribbon that flows vertically down the right side of the page and then curves horizontally across the bottom. The ribbon has a 3D effect with a lighter blue shadow underneath. The text "Capítulo 1" is centered in the lower-left area of the page.

Capítulo 1

Capítulo 1: Fundamentos teóricos y metodológicos que sustentan el desarrollo de una herramienta informática que perfeccione la gestión de la información sobre los pacientes.

Introducción

Este capítulo abarca los fundamentos teóricos y metodológicos asociados al tema de la investigación. Describe los principales conceptos asociados a la gestión de la información sobre los pacientes del hospital pediátrico en la provincia de Sancti Spíritus, asimismo se explica la necesidad de informatizar el proceso. Se realiza un estudio detallado de las distintas metodologías, tecnologías y herramientas actuales de desarrollo utilizadas para el análisis, diseño e implementación del sistema sobre las cuales se apoya la propuesta de trabajo.

1.1 Gestión de la Información

La Gestión de la información se trataría de la explotación de la información para la consecución de los objetivos de la entidad. Su creación, adquisición, procesamiento y difusión (Arévalo, 2008).

Josep Curto se centra en el significado etimológico del concepto y lo define como un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma. Se establece, por lo tanto, como una disciplina transversal que aparece entrelazada en las diferentes capas o tejidos de una organización, en los conceptos de management (recursos humanos, marketing, finanzas, estrategias, operaciones) y les proporciona soporte (Curto, 2006).

1.2 Registros Médicos y Estadísticos

Se denominan registros médicos a los formularios en que se inscriben de modo ordenado y se hace constar las observaciones o eventos relacionados con la salud humana, los que permiten organizar y controlar a los anteriores y los que sirven para compilar y transmitir estadísticas de salud. A principios de la década del ochenta del siglo pasado, a los Departamentos de Estadísticas de las unidades del Sistema Nacional de Salud, se les cambió el nombre por el de Registros Médicos. Este último es más usado en los países de América Latina para denominar la unidad organizativa que tiene bajo su responsabilidad el control de los pacientes inscriptos, la entrega de turnos para servicio ambulatorio, el control de pacientes hospitalizados, el archivo de

historias clínicas y procesar la información para calcular los indicadores que permiten evaluar las actividades que realiza la institución (Massabot, Viera, & Pérez, 2005).

El Departamento de Estadística del Ministerio de Salud Pública modificó su estructura organizativa para otra capaz de dar respuesta al procesamiento, no solo de estadística de mortalidad y enfermedades de declaración obligatoria, sino que además permitiera elaborar y publicar las estadísticas de recursos, servicios prestados, los indicadores que permitan evaluar los diferentes programas y desarrollar investigaciones a partir de los registros continuos. Se implantó, un sistema de supervisión de la calidad de los registros y procesamiento de los datos estadísticos para las provincias y sus unidades (Massabot et al., 2005).

Al finalizar el siglo XX los Registros Médicos contaban con cobertura total del país, integridad muy próxima al 100 % e información de calidad. El nivel alcanzado pone de manifiesto cómo el Ministerio de Salud Pública, en particular, y el estado cubano, en general, han intervenido en la evolución de los mismos por estar íntimamente vinculados a la gestión de salud y sus resultados (Massabot et al., 2005).

Su misión es:

Garantizar la producción de estadísticas de calidad a través del Sistema Estadístico Nacional ejerciendo una adecuada dirección, ejecución y control de la captación de las cifras, así como su adecuada difusión de acuerdo con las necesidades del país en cuanto a información estadística.

Su visión es:

Construiremos un sistema estadístico profesional capaz de responder con calidad y oportunidad a las necesidades de información estadística del país para cumplir las metas del mismo y su adecuado reflejo internacional, conscientes de ser útiles y necesarios a la sociedad.

1.3 Antecedentes de los sistemas de gestión de pacientes

Los sistemas de gestión de pacientes son aquellos en que las computadoras se limitan a realizar las operaciones de procesamiento físico de la información. Las personas que integran el sistema, asumen todas las labores de generación de la información primaria y de análisis de información de resultados.

En nuestro país existen varios sistemas para el control de pacientes, de los cuales ninguno cumple con las características demandadas por el cliente.

CONSAN: *El Sistema “CONSAN” constituye un esfuerzo por dotar de un sistema de información estadístico a los Centros de Atención Integral a personas que viven con VIH/SIDA (CAIP VIH/SIDA) del país relacionado con el Objeto Social aprobado para las instituciones y cuyo destino es que las informaciones sean estudiadas, analizadas y utilizadas para la toma de decisiones y posteriormente almacenadas en los soportes informáticos de departamentos y subdirecciones. El sistema es capaz de garantizar la comprobación de los registros primarios el levantamiento adecuado de cada información, su correcto almacenamiento, facilita el trabajo al disponer de la información necesaria relacionada con los servicios que se brindan en el Centro y facilita su análisis por parte de los directivos en particular, así como de los órganos colegiados de dirección (Vázquez, Álvarez, & Padrón, 2011).*

En el mundo existen varios sistemas para la gestión de pacientes que pueden ser descargados desde internet, tanto desde sus sitios oficiales, como de sitios de terceros, y estos últimos en alguna u otra medida tienen licencia para poder comercializarlos. Hay que destacar que todas estas descargas del producto son demos, por lo que tienen un período de prueba, para poder hacer uso del sistema hay que pagar. A continuación se exponen algunos de esos sistemas con sus características fundamentales.

Software Pediatra: *Entre sus principales características se encuentra dar alta de nuevos pacientes, seguimiento de las citas programadas y consulta del historial, administración e impresión de recetas en base a las vacunas y/o diagnóstico del paciente, catálogos de vacunas y medicamentos, seguimiento de pendientes y control estadístico de la entidad, se puede operar en el mismo sistema la administración de perfiles de acceso para diferente médicos pediatras con su propio seguimiento de pacientes y citas.*

Clini.One: *Permite gestionar (programación, recursos, colaboradores, pacientes,...). Algunas de sus características esenciales, se muestran a continuación:*

Interacción con pacientes: facilita información de tratamientos, consultas, presupuestos, agendas, citas, recordatorios, etc. Permite publicar el directorio de especialistas y servicios.
Interacción con médicos y equipos: consulta de agenda, planificación de intervenciones, casos clínicos de pacientes, estudios y trabajos clínicos.

Kaplan: *El objetivo primordial de la aplicación es el control de pacientes y la generación del libro de quirófano que los centros médicos.*

1.4 Metodologías para el desarrollo de software

1.4.1 Metodologías ágiles y tradicionales

Al momento de realizar un software hay varias metodologías que podemos usar para desarrollarlo y debemos hacer la pregunta ¿Cuál vamos a usar?, ¿Cuál nos conviene más? ...Y estas preguntas deben ser respondidas en base al estudio de cada una de esas metodologías.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software, tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar software de calidad, incluyendo heurísticas de construcción y criterios de comparación de modelos de sistemas.

Las metodologías tradicionales nos ofrecen una buena solución para proyectos donde el entorno es estable y donde los requisitos se conocen con exactitud y están bien definidos, porque no están pensadas para trabajar con incertidumbre. La mayoría de las decisiones se toman al principio.

Las metodologías ágiles tienen sus inicios a principios de la década de 1990, con el surgimiento de un enfoque que fue bastante revolucionario para su momento ya que iba en contra de toda creencia de que mediante procesos altamente definidos se iba a lograr obtener software en tiempo, costo y con la requerida calidad. En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace formalmente el término “ágil” aplicado al desarrollo de software. Están pensadas para proyectos con pocos principios, requisitos cambiantes, donde no exista un contrato formal, o el mismo sea flexible (Calderón, 2009).

Según los conceptos antes analizados y las características del negocio en cuestión se determina que la metodología tradicional es la apropiada ya que este proyecto tiene numerosos requisitos que están bien definidos y el cliente no es parte del equipo de desarrollo.

1.4.2 Metodología Rational Unified Process (RUP)

El Proceso Unificado de Rational se adapta a las características del negocio que se lleva a cabo en la UPEC además porque brinda facilidades para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Además es una metodología con la capacidad

de adaptación a cualquier proyecto, cuyo objetivo es producir software de alta calidad, cumplimentando los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos (Jacobson, Booch, & James Rumbaugh, 2000).

RUP junto con el Lenguaje Unificado de Modelado UML, constituye una de las metodologías estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Principales características:

- ✓ Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- ✓ Pretende implementar las mejores prácticas en Ingeniería de Software
- ✓ Desarrollo iterativo
- ✓ Administración de requisitos
- ✓ Uso de arquitectura basada en componentes
- ✓ Control de cambios • Modelado visual del software
- ✓ Verificación de la calidad del software

No obstante, los verdaderos aspectos definitorios de RUP se resumen en tres frases clave: está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software (Jacobson et al., 2000).

1.4.3 Lenguaje de Modelado Unificado (UML)

UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos). (Krall, s. f.). Es un lenguaje de modelado visual de propósito general orientado a objetos. Independiente de cualquier fabricante comercial.

El término “lenguaje” ha generado bastante confusión respecto a lo que es UML. En realidad el término lenguaje quizás no es el más apropiado, ya que no es un lenguaje propiamente dicho, sino una serie de normas y estándares gráficos respecto a cómo se deben representar los esquemas relativos al software. Mucha gente piensa por confusión que UML es un lenguaje de programación y esta idea es errónea.

Ventajas de UML(López & Ruiz, 2011):

- ✓ *Es estándar, facilita la comunicación*
- ✓ *Está basado en meta modelo con una semántica bien definida*
- ✓ *Se basa en una notación gráfica concisa y fácil de aprender y utilizar*
- ✓ *Se puede utilizar para modelar sistemas software en diversos dominios:*
- ✓ *Sistemas de información empresarial, Sistemas WEB, sistemas críticos y de tiempo real, etc.*
- ✓ *Incluso en sistemas que no son software*
- ✓ *Es fácilmente extensible*
- ✓

Inconvenientes de UML(López & Ruiz, 2011):

- ✓ *No es una metodología. Además de UML, hace falta una metodología OO*
- ✓ *No cubre todas las necesidades de especificación de un proyecto software*
- ✓ *No define los documentos textuales o el diseño de interfaces de usuario*
- ✓ *Faltan ejemplos elaborados en la documentación*
- ✓ *Puede resultar complejo alcanzar un conocimiento completo del lenguaje*

Según (Orallo, 2002) UML tiene como objetivos:

- *Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.*
- *Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.*
- *Construir: A partir de los modelos especifica-dos se pueden construir los sistemas diseñados.*
- *Documentar: Los propios elementos gráficos sirven como documentación del sistema des-arrollado que pueden servir para su futura re-visión.*

1.5 Tecnologías para la elaboración de software

1.5.1 Aplicaciones de escritorio

“Por aplicaciones de escritorio se entiende toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea Windows o Linux. Las aplicaciones de escritorio se desarrollan para cubrir necesidades específicas de la empresa, como la contabilidad, gestión de personal, etc.” (Landay, 2009)

Las aplicaciones de escritorio son responsables, rápidas, multiplataforma y soportan diseño con múltiples capas (Landay, 2009).

Por las características anteriores se seleccionó realizar una aplicación de escritorio en el presente proyecto, ya que una aplicación web aumentaría el tráfico de la red generando dificultad en la navegación pues la interconexión entre las computadoras en la sede de la UPEC en Sancti Spíritus no es óptima. También el software será manipulado solo por dos personas que trabajan en el mismo departamento; por lo que se determinó que con una aplicación de escritorio se lograría mayor rapidez y efectividad.

1.5.2 Aplicaciones Web

De forma breve, una aplicación Web se puede definir como una aplicación en la cual el usuario por medio de un navegador realiza peticiones a una aplicación remota accesible a través de Internet (o a través de un Intranet) y que recibe una respuesta que se muestra en el propio navegador (Mora, 2002).

Las aplicaciones web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o el desarrollo del comercio electrónico. Además, una aplicación web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser gestión de clientes, contabilidad o inventario, a través de una página web (Mora, 2002).

1.5.3 Arquitectura en n Capas

En el diseño de sistemas informáticos actuales se suele usar las arquitecturas multinivel o programación por capas, teniendo en cuenta las ventajas que esta ofrece. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables es decir que pueden ampliarse con facilidad en caso de que las necesidades aumenten y entre las que más se destacan se encuentra el modelo de tres capas.

*La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido. La primera capa se denomina **capa de presentación** y normalmente consiste en una interfaz gráfica de usuario de algún tipo. La capa intermedia, o **capa de negocio**, consiste en la aplicación o lógica de empresa, y la tercera capa, la **capa de datos**, contiene los datos necesarios para la aplicación (Chappell & Kirk, 2006).*

La capa intermedia (lógica de aplicación) es básicamente el código al que recurre la capa de presentación para recuperar los datos deseados. La capa de presentación recibe entonces los datos y los formatea para su presentación. Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Pueden construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación siempre que ésta presente una interfaz claramente definida a la capa de presentación. La tercera capa es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio (Peláez, 2009).

Ventajas de la arquitectura en tres capas:

- *Mejoras en las posibilidades de mantenimiento. Debido a que cada capa es independiente de la otra los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo.*
- *Escalabilidad. Como las capas están basadas en diferentes maquinas, el escalamiento de la aplicación hacia afuera es razonablemente sencillo.*
- *Flexibilidad. Como cada capa puede ser manejada y escalada de forma independiente, la flexibilidad se incrementa.*
- *Disponibilidad. Las aplicaciones pueden aprovechar la arquitectura modular de los sistemas habilitados usando componentes que escalan fácilmente lo que incrementa la disponibilidad.*

1.6 Lenguajes de Programación

1.6.1 C++

El lenguaje C++ se comenzó a desarrollar en 1980. Al comienzo era una extensión del lenguaje de C que fue denominada C with clases. En la actualidad C++ es un lenguaje versátil, potente y general. El C++ mantiene las ventajas de C en cuanto a riquezas de operadores expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades de y limitaciones del C original (Bustamante, Aguinaga, Aibar, Alaizola, & Lazcano, 2004).

Algunas de las características más importantes de C++ son (Echeverría, Sánchez, & Álvaro Prieto Ramos, 2011):

- ✓ Soporte de diferentes estilos de programación.
- ✓ Espacios de nombres.
- ✓ Encapsulación de información mediante el concepto de clases, apoyado también en los modificadores de acceso.
- ✓ Soporte de herencia simple y múltiple.
- ✓ Funciones virtuales, que posibilitan el polimorfismo.
- ✓ Sobrecarga de funciones y operadores.
- ✓ Soporte de excepciones.
- ✓ Biblioteca estándar, base para el desarrollo de un gran número de otras bibliotecas.
- ✓ Programación genérica mediante el soporte de plantillas.

1.6.2 C#

C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java y las combina en uno solo. La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos (Seco, 2004).

Algunas características de C# (Seco, 2004):

- ✓ Sencillez: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.
- ✓ Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos.
- ✓ Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.
- ✓ Gestión automática de memoria: C# también proporciona un mecanismo de liberación de recursos determinista a través de la instrucción `using`.
- ✓ Extensibilidad de tipos básicos: C# permite definir, a través de estructuras, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos.

1.6.3 Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de la década de 1990. Este lenguaje toma la estructura de sintaxis (escritura de código fuente) de C y C++ incorporando un modelo de objetos más simples y eliminando las herramientas de bajo nivel (que suele inducir muchos errores como la manipulación de punteros en C y C++)(Capriolo, 2011).

Java, siendo robusto, seguro, fácil de usar, fácil de entender, y descargable desde la red, es un lenguaje base excelente para aplicaciones de base de datos.

Java es un lenguaje útil para casi todo tipo de problemas. Podemos citar como funcionalidades de Java varias(Rodríguez, 2006):

1. **Aplicaciones “cliente”:** *son las que se ejecutan en un solo ordenador (por ejemplo el portátil de tu casa) sin necesidad de conectarse a otra máquina. Pueden servirte por ejemplo para realizar cálculos o gestionar datos.*
2. **Aplicaciones “cliente/servidor”:** *son programas que necesitan conectarse a otra máquina (por ejemplo un servidor de datos) para pedirle algún servicio de forma más o menos continua, como podría ser el uso de una base de datos. Pueden servir por ejemplo para el teletrabajo: trabajar desde casa pero conectados a un ordenador de una empresa.*
3. **Aplicaciones web:** *que son programas Java que se ejecutan en un servidor de páginas web. Estas aplicaciones reciben “solicitudes” desde un ordenador y envían al navegador (Internet Explorer, Firefox, Safari, etc.) que actúa como su cliente páginas de respuesta en HTML.*

Características del Lenguaje Java según (Capriolo, 2011):

- ✓ *Orientado a Objetos*
- ✓ *Simple*
- ✓ *Distribuido*
- ✓ *Multiplataforma*
- ✓ *Es robusto: Recolección de basura*
- ✓ *Seguro*
- ✓ *Portable*
- ✓ *Interpretado*
- ✓ *Multihilo (multithreaded)*

- ✓ *Dinámico*

Por todos estos motivos y beneficios que nos brinda el lenguaje Java, se seleccionó como el lenguaje de programación indicado para el desarrollo de este proyecto.

1.7 Sistemas Gestores de Base de Datos

Un sistema de base de datos es básicamente un sistema computarizado para guardar registros; es decir, es un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones. La información en cuestión puede ser cualquier cosa que sea de importancia para el individuo u organización; en otras palabras, todo lo que sea necesario para auxiliarle en el proceso general de su administración (Date, 2001).

Características de un sistema gestor de bases de datos:

- 1. Independencia: Los datos se organizan independientemente de las aplicaciones que los vayan a usar (independencia lógica) y de los ficheros en los que vayan a almacenarse (independencia física).*
- 2. Los usuarios y las aplicaciones pueden acceder a los datos mediante el uso de lenguajes de consulta (p.ej. SQL, Query-by-example...).*
- 3. Centralización: Los datos se gestionan de forma centralizada e independiente de las aplicaciones.*
- 4. Consistencia e integridad de los datos.*
- 5. Fiabilidad (protección frente a fallos) y seguridad (control de acceso a los datos).*

Es posible imaginar a las bases de datos como un armario. Los usuarios del sistema pueden realizar una serie de operaciones, por ejemplo:

- ✓ *Agregar nuevos archivos vacíos a la base de datos.*
- ✓ *Insertar datos dentro de los archivos existentes.*
- ✓ *Recuperar datos de los archivos existentes.*
- ✓ *Modificar datos en archivos existentes.*
- ✓ *Eliminar datos de los archivos existentes.*
- ✓ *Eliminar archivos existentes de la base de datos.*

1.7.1 MySQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción

atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha (Santillán, Ginestà, & Mora, 2007).

Características importantes del Software de Base de Datos MySQL:

- ✓ *Trabaja en múltiples plataformas.*
- ✓ *Posee numerosos tipos de datos: enteros con y sin signo, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, y ENUM.*
- ✓ *Records de longitud fija y variable.*
- ✓ *Soporte de operadores y funciones completo en las cláusulas SELECT y WHERE de las consultas.*
- ✓ *Soporte para las cláusulas SQL: GROUP BY y ORDER BY. Además de funciones de agrupación (COUNT (), AVG (), STD (), SUM (), MAX (), MIN (), and GROUP_CONCAT ()).*

1.7.2 PostgreSQL

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley. Con cerca de una década de desarrollo tras él, PostgreSQL es un gestor de bases de datos de código abierto, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, tcl y Python)(Equipo de desarrollo de PostgreSQL, 2001).

Prestaciones de PostgreSQL según (Ginestà & Mora, 2007):

- ✓ *Está desarrollado en C, con herramientas como Yacc y Lex.*
- ✓ *La API de acceso al SGBD se encuentra disponible en C, C++, Java, Perl, PHP, Python y TCL, entre otros.*
- ✓ *Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.*
- ✓ *Su administración se basa en usuarios y privilegios.*
- ✓ *Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC.*
- ✓ *Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.*
- ✓ *Es altamente confiable en cuanto a estabilidad se refiere.*

- ✓ *Puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética (soundex), etc.*
- ✓ *Control de concurrencia multi-versión, lo que mejora sensiblemente las operaciones de bloqueo y transacciones en sistemas multi-usuario.*
- ✓ *Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.*
- ✓ *Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.*

Ventajas y Desventajas de PostgreSQL (Silva, Kinderman, & Velasquez, 2011):

Ventajas

- *Ampliamente popular - Ideal para tecnologías Web.*
- *Fácil de Administrar.*
- *Su sintaxis SQL es estándar y fácil de aprender.*
- *Footprint bajo de memoria, bastante poderoso con una configuración adecuada.*
- *Multiplataforma.*
- *Capacidades de replicación de datos.*
- *Soporte empresarial disponible.*

Desventajas:

- *Sin experticia, configurar llega a ser un caos.*
- *Es fácil de vulnerar sin protección adecuada.*
- *El motor MyISAM es instalado por defecto y carece de capacidades de integridad relacional.*
- *InnoDB genera mucho footprint en memoria al indizar.*
- *El toolset empresarial tiene un costo adicional por suscripción anual.*
- *Realizar revisiones llegar a ser una labor manual y tediosa para el DBA.*
- *Reducida cantidad de tipos de datos.*

Por las características mostradas anteriormente se seleccionó PostgreSQL como SGBD en este proyecto.

1.8 Herramientas de Desarrollo

1.8.1 NetBeans

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso(Álvarez, 2011).

NetBeans es uno de los entornos Java más utilizados, esta herramienta es posible expandirla mediante plug-ins, añadiendo así funcionalidades que el paquete básico no trae por defecto mientras se vayan necesitando.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software(Álvarez, 2011).

1.8.2 Eclipse

Eclipse es una plataforma de desarrollo open source basada en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios. En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in). Hay plug-ins para el desarrollo de Java (JDT Java Development Tools) así como para el desarrollo en C/C++, COBOL, etc. La versión instalada en el laboratorio incluye el plug-in JDT(Valencia, 2004).

Conclusiones

En este capítulo se realizó una revisión bibliografía para conocer los conceptos fundamentales de la investigación, se estudiaron las principales herramientas para elaborar el software y atendiendo a sus características y las particularidades del negocio se seleccionaron las siguientes:

- ✓ *RUP (Proceso Unificado de Desarrollo) como la metodología más apropiada para el desarrollo del proyecto.*

- ✓ *UML como el lenguaje de modelación utilizado.*
- ✓ *Para la implementación el lenguaje Java, utilizando como herramienta de programación NetBeans IDE 6.8.*
- ✓ *PostgreSQL 8.4 como sistema gestor de bases de datos*
- ✓ *Visual Paradigm for UML como herramienta de modelado UML.*

La selección de estas tecnologías se basó principalmente en la tecnología código libre y la característica de integrarse para funcionar en cualquier plataforma.

A decorative graphic consisting of a blue wavy ribbon that flows vertically down the right side of the page and then curves horizontally across the bottom. The ribbon has a 3D effect with a lighter blue shadow on its underside and a darker blue highlight on its top edge.

Capítulo 2

Capítulo 2: Descripción de la aplicación propuesta para el control de los pacientes en el departamento de registros médicos y estadística del hospital pediátrico de Sancti Spiritus.

Introducción

En este capítulo se describen los requisitos funcionales y no funcionales del software, se exponen las reglas del negocio a respetar para el diseño de la aplicación, se definen los actores del negocio, los trabajadores y los casos de uso que permitirán conformar el diagrama de casos de uso del negocio. Se abordan los flujos de trabajo de Análisis y Diseño, describiendo los artefactos según la Metodología RUP.

2.1 Modelo del Negocio

El modelado de negocio es una técnica para comprender los procesos de negocio de la organización. Además de identificar los casos de uso y las entidades del negocio relevantes que el software debe soportar, de forma que se puede modelar solo lo necesario para que se comprenda el contexto. El modelado del negocio está soportado por 2 tipos de modelos de UML: modelos de casos de uso y modelo de objetos (Jacobson et al., 2000).

A continuación se explicará el proceso del negocio que se lleva a cabo en el departamento de Registros Médicos y Estadística del Hospital Pediátrico de Sancti Spiritus.

2.1.1 Identificación de los procesos del negocio

Un proceso de negocio se puede definir según (Davenport, & Young 1990 citado en Pressman, 2005) como un conjunto de tareas relacionadas lógicamente que se llevan a cabo para obtener un determinado resultado de negocio. “Dentro del proceso de negocio, se combinan las personas, los equipos, los recursos materiales y los procedimientos de negocio con objeto de producir un resultado concreto” (Pressman, 2010).

Apyados en el concepto anterior se identificó el siguiente proceso del negocio: Gestión de la información sobre los pacientes en el Departamento de Registros Médicos y Estadística del Hospital Pediátrico de Sancti Spiritus.

El Departamento de Registros médicos y Estadística del Hospital Pediátrico de Sancti Spiritus se compone de diferentes áreas, entre las que se encuentran Cuerpo de Guardia y Cirugía, los cuales tienen un gran paso de pacientes diarios, de los cuales se debe tener un registro por cada uno, con sus datos personales, enfermedades, operaciones, entre otros datos. De igual forma la Dirección Nacional de Registros Médicos y Estadísticas solicita frecuentemente información sobre estos pacientes, por ejemplo cantidad de pacientes con una enfermedad, datos personales de pacientes pertenecientes a un grupo de edad determinado, los pacientes operados en una fecha determinada, entre otros datos. Una vez que se solicita determinada información al Director, este le pide al técnico que investigue y este lo hace en el archivo, y después de un tiempo considerable, prepara un informe y se lo entrega al Director.

2.1.2 Reglas del negocio a considerar

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto de negocio (Jacobson et al., 2000). Basado en la definición anterior se identificaron las siguientes reglas del negocio:

- ✓ Cuando un paciente llega al Cuerpo de Guardia se le confecciona un registro.
- ✓ Cada vez que un paciente es operado se le crea un informe operatorio.
- ✓ Cuando al paciente se le crea un informe operatorio, el paciente no se puede eliminar

2.1.3 Modelo de casos de uso del negocio

El modelo de Casos de Uso del Negocio es el encargado de describir los procesos de una empresa utilizando los casos de uso y los actores, en correspondencia, a su vez, con los procesos del negocio y los clientes (Jacobson et al., 2000). Se define a través de: el diagrama de casos de uso, la descripción de estos y el diagrama de actividades de los casos de uso.

2.1.4 Actores del Negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados (Jacobson et al., 2000). Basado en la definición anterior se identifica como actor del negocio el siguiente:

Actor	Descripción
Director	Interesado en realizar un informe del comportamiento los pacientes y las

	enfermedades.
--	---------------

Tabla 1 Actores del Negocio

2.1.5 Trabajadores del Negocio

Un trabajador es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores y manipulando entidades (Jacobson et al., 2000). Apoyándonos en este concepto se define como trabajador del negocio:

Trabajador	Descripción
Técnico	Encargado de recopilar los datos relacionados con los pacientes, además la actualización de los mismos y de realizar un informe para el directivo.

Tabla 2 Trabajadores del negocio

2.1.6 Diagrama de Casos de Uso del Negocio

Un diagrama de casos de uso representa gráficamente a los procesos del negocio como casos de usos y su interacción con los actores. Los diagramas de casos de uso se utilizan para especificar las funcionalidades y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas. O sea es un diagrama que muestra la relación entre los actores y los casos de uso en un sistema (Jacobson et al., 2000).



Figura 1 Diagrama de casos de uso del negocio

2.1.7 Descripción textual de los casos de uso del negocio

Caso de uso del negocio	Realizar informes
-------------------------	-------------------

Actor	Director
Propósito	Evaluar el comportamiento y controlar los pacientes y las enfermedades del Hospital Pediátrico de Sancti Spiritus.
Resumen: El caso de uso inicia cuando el directivo solicita algún tipo de información a al técnico con este propósito. El técnico recopila la información, realiza un informe y se lo proporciona al director. El caso de uso concluye cuando el director obtiene la información deseada.	
Caso de uso asociado	
Curso normal de los eventos	
Acción del actor	Respuesta del negocio
1-Solicita información	
	2-Recopila información
	3-Elabora el informe
	4-Entrega el informe al directivo
5-Obtiene el informe	
Prioridad	Alta
Mejoras	Al estar digitalizada la información los informes se podrán realizar de forma inmediata y se eliminará la posibilidad de cometer errores.

Tabla 3 Descripción textual del caso de uso del negocio: Realizar informes

2.1.8 Diagramas de Actividades

Un diagrama de actividades es el que muestra el flujo de actividad a actividad; los diagramas de actividad tratan la vista dinámica de un sistema. Un caso especial de diagrama de estados (aquellos diagramas que tratan la vista dinámica de un sistema) en el cual todos o casi todos los estados son estados de acción y en el cual todas o casi todas las transiciones son disparadas por la terminación de las acciones en los estados origen (Jacobson et al., 2000).

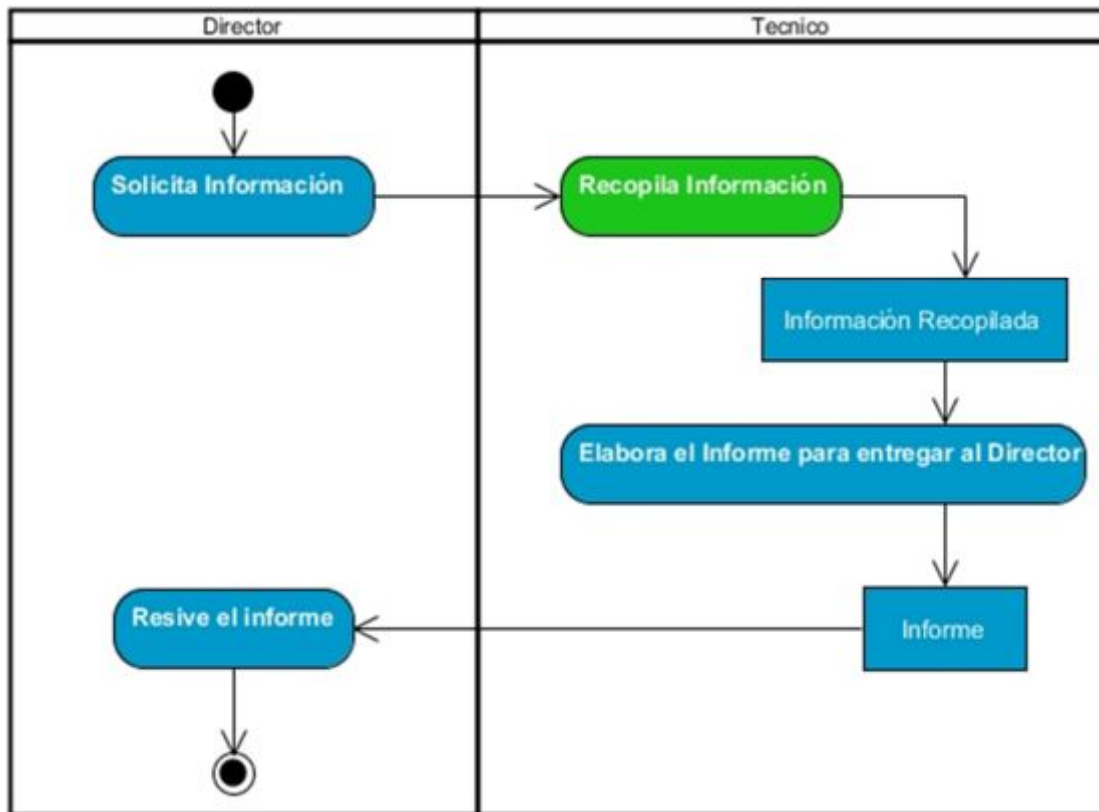


Figura 2 Diagrama de Actividades del caso de uso Realizar Informe

2.1.9 Modelo de objetos del negocio

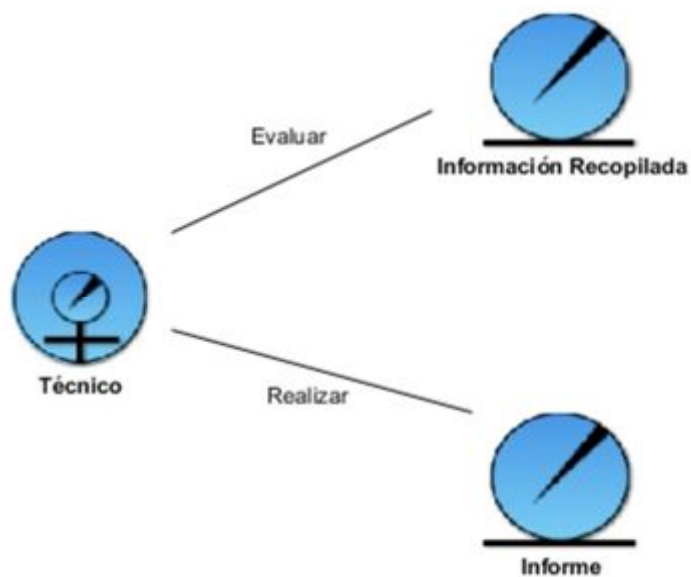


Figura 3 Modelo de objetos del negocio

2.2 Requerimientos

La IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos.

2.2.1 Requerimientos funcionales

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo (Jacobson et al., 2000). Todo esto basándose en las necesidades del usuario.

Los requerimientos funcionales del sistema propuesto son:

- 1. Autenticar Usuario.*
- 2. Cambiar contraseña.*
- 3. Adicionar usuario.*
- 4. Modificar usuario.*
- 5. Eliminar usuario.*
- 6. Adicionar paciente cirugía.*
- 7. Modificar paciente cirugía.*
- 8. Eliminar paciente cirugía.*
- 9. Adicionar cirugía.*
- 10. Modificar cirugía.*
- 11. Eliminar cirugía.*
- 12. Adicionar clasificación de cirugía.*
- 13. Modificar clasificación de cirugía.*
- 14. Eliminar clasificación de cirugía.*
- 15. Adicionar Salón.*
- 16. Modificar Salón.*
- 17. Eliminar Salón.*
- 18. Adicionar tipo de anestesia.*
- 19. Modificar tipo de anestesia.*
- 20. Eliminar tipo de anestesia.*
- 21. Adicionar tipo de cirugía.*

22. *Modificar tipo de cirugía.*
23. *Eliminar tipo de cirugía.*
24. *Adicionar clase de operación.*
25. *Modificar clase de operación.*
26. *Eliminar clase de operación.*
27. *Adicionar ayudantes.*
28. *Modificar ayudantes.*
29. *Eliminar ayudantes.*
30. *Adicionar cirujanos.*
31. *Modificar cirujanos.*
32. *Eliminar cirujanos.*
33. *Adicionar anestesiólogo.*
34. *Modificar anestesiólogo.*
35. *Eliminar anestesiólogo.*
36. *Adicionar enfermeras de salón.*
37. *Modificar enfermeras de salón.*
38. *Eliminar enfermeras de salón.*
39. *Adicionar paciente de cuerpo de guardia.*
40. *Modificar paciente de cuerpo de guardia.*
41. *Eliminar paciente de cuerpo de guardia.*
42. *Adicionar municipio.*
43. *Modificar municipio.*
44. *Eliminar municipio.*
45. *Adicionar enfermedades.*
46. *Modificar enfermedades.*
47. *Eliminar enfermedades.*
48. *Adicionar grupo de enfermedades.*
49. *Modificar grupo de enfermedades.*
50. *Eliminar grupo de enfermedades.*
51. *Mostrar modelo CAAB.*
52. *Mostrar modelo EDA.*
53. *Mostrar modelo IRA.*
54. *Mostrar modelo SFA.*
55. *Mostrar modelo EDO.*

56. *Mostrar modelo Accidente.*
57. *Mostrar modelo Morbilidad.*
58. *Mostrar casos de un Grupo dado en un período de fecha.*
59. *Mostrar casos de una Enfermedad por Grupo de Edades y Municipios.*
60. *Mostrar remitidos de un municipio.*
61. *Mostrar remitidos por municipios.*
62. *Mostrar el porciento anual de pacientes dado un grupo.*
63. *Mostrar casos dada una enfermedad*
64. *Mostrar casos de un grupo de enfermedades en un intervalo de fecha, de un municipio y un grupo de edad determinado.*
65. *Mostrar cantidad de operaciones por clase de operación.*
66. *Mostrar el modelo de cirugía.*
67. *Mostrar el porciento anual de operaciones de una Clase.*
68. *Mostrar todos los pacientes operados en un día determinado.*

2.2.2 Requerimientos no funcionales

Los requisitos no funcionales especifican cualidades y propiedades del sistema, como restricciones del entorno o de la implementación, rendimientos, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad (Jacobson et al., 2000).

Los requerimientos no funcionales del sistema propuesto son los siguientes:

Requerimientos de Apariencia o Interfaz Externa

El software brindará una interfaz sencilla que facilite la interacción del usuario con el mismo, haciéndole sentir que tiene el control absoluto de la aplicación. Esta interfaz está diseñada para que el usuario pueda ir de un lugar a otro del sistema con gran facilidad garantizará su rápido aprendizaje.

Requerimientos de Usabilidad

El producto está orientado para ser usado por diferentes tipos de usuario dependiendo de su funcionalidad. El administrador del es el encargado de definir el acceso que tendrán los usuarios.

Requerimientos de Rendimiento

El sistema propuesto debe ser rápido en el procesamiento de la información así como a la hora de dar respuesta a la solicitud de los usuarios, además debe permitir el acceso simultáneo a los

datos por diferentes usuarios. El sistema deberá recuperarse en un corto período de tiempo ante cualquier falla.

Requerimientos de Soporte

Se requiere un servidor de bases de datos con soporte para grandes volúmenes de información, velocidad de procesamiento y tiempo de respuesta rápido en accesos concurrentes. Se documentará la aplicación para garantizar su soporte.

Requerimientos de Portabilidad

La plataforma seleccionada para desarrollar la aplicación fue Windows, pero puede ser ejecutada desde otras plataformas como Linux, que soporten el lenguaje Java y se pueda instalar el servidor de PostgreSQL.

Requerimientos de Seguridad

El sistema controlará los diferentes niveles de acceso y funcionalidad de los usuarios, de forma que garantice la protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. Debe garantizar la conectividad e integridad de los datos almacenados a través de la red. Debe garantizar la confidencialidad para proteger la información de acceso no autorizado, para lo cual el sistema impondrá un estricto control de acceso que permitirá a cada usuario tener disponible solamente las opciones relacionadas con su actividad.

Confiabilidad

Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario. El sistema en casos de fallos debe garantizar que las pérdidas de información sean mínimas.

Requerimientos de Software

Se debe disponer de un sistema operativo compatible. El sistema debe poderse ejecutar en entornos Windows y Linux. Se necesita tener instalada la máquina virtual de Java (JVM) en las computadoras donde se desee utilizar el software. Debe existir una computadora disponible para el servidor de PostgreSQL.

Hardware:

Las computadoras donde los usuarios utilizarán el software requieren como mínimo:

- Procesador Intel Pentium IV
- 256 Mb de Memoria RAM
- Deben estar conectadas en red con el servidor a través de una tarjeta de red de 100 Mbps

Se requiere de una máquina que funcione como servidor, la cual debe tener como mínimo:

- Procesador Intel Pentium IV
- 512 Mb de Memoria RAM
- 40 GB de Disco Duro.

Requerimientos de Ayuda y Documentación en Línea

El sistema poseerá una ayuda que expondrá de manera clara y precisa al usuario todas las funcionalidades del sistema. La ayuda estará contenida por un menú general permitiéndole al usuario navegar por la misma sin perderse.

2.3 Modelo del Sistema

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario y proporciona la entrada fundamental para el análisis, el diseño y las pruebas (Jacobson, Booch, & James Rumbaugh, 2006).

2.3.2 Actores del sistema

Un actor es una idealización de una persona externa, de un proceso, o de una cosa que interactúa con un sistema. Cada actor participa en uno o más casos de uso. Un actor puede ser un ser humano, otro sistema informático, o un cierto proceso ejecutable (Jacobson et al., 2000).

Los actores de nuestro sistema se presentan a continuación:

Actor	Descripción
Técnico de Cuerpo de Guardia	Encargada de registrar los datos relacionados con los pacientes del Cuerpo de Guardia.
Técnico de Cirugía	Encargada de registrar los datos relacionados con los pacientes de Cirugía.
Administrador	Interesado en administrar y controlar el acceso al software.

Tabla 4 Actores del sistema

2.3.3 Casos de uso del sistema

Un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia (Jacobson et al., 2000).

Para este sistema se definieron los siguientes casos de uso:

- 1. Autenticar Usuario.*
- 2. Cambiar contraseña.*
- 3. Adicionar usuario.*
- 4. Modificar usuario.*
- 5. Eliminar usuario.*
- 6. Adicionar paciente cirugía.*
- 7. Modificar paciente cirugía.*
- 8. Eliminar paciente cirugía.*
- 9. Adicionar cirugía.*
- 10. Modificar cirugía.*
- 11. Eliminar cirugía.*
- 12. Adicionar clasificación de cirugía.*
- 13. Modificar clasificación de cirugía.*
- 14. Eliminar clasificación de cirugía.*
- 15. Adicionar Salón.*
- 16. Modificar Salón.*
- 17. Eliminar Salón.*
- 18. Adicionar tipo de anestesia.*
- 19. Modificar tipo de anestesia.*
- 20. Eliminar tipo de anestesia.*
- 21. Adicionar tipo de cirugía.*
- 22. Modificar tipo de cirugía.*
- 23. Eliminar tipo de cirugía.*
- 24. Adicionar clase de operación.*
- 25. Modificar clase de operación.*
- 26. Eliminar clase de operación.*
- 27. Adicionar ayudantes.*
- 28. Modificar ayudantes.*

29. *Eliminar ayudantes.*
30. *Adicionar cirujanos.*
31. *Modificar cirujanos.*
32. *Eliminar cirujanos.*
33. *Adicionar anestesiólogo.*
34. *Modificar anestesiólogo.*
35. *Eliminar anestesiólogo.*
36. *Adicionar enfermeras de salón.*
37. *Modificar enfermeras de salón.*
38. *Eliminar enfermeras de salón.*
39. *Adicionar paciente de cuerpo de guardia.*
40. *Modificar paciente de cuerpo de guardia.*
41. *Eliminar paciente de cuerpo de guardia.*
42. *Adicionar municipio.*
43. *Modificar municipio.*
44. *Eliminar municipio.*
45. *Adicionar enfermedades.*
46. *Modificar enfermedades.*
47. *Eliminar enfermedades.*
48. *Adicionar grupo de enfermedades.*
49. *Modificar grupo de enfermedades.*
50. *Eliminar grupo de enfermedades.*
51. *Mostrar modelo CAAB.*
52. *Mostrar modelo EDA.*
53. *Mostrar modelo IRA.*
54. *Mostrar modelo SFA.*
55. *Mostrar modelo EDO.*
56. *Mostrar modelo Accidente.*
57. *Mostrar modelo Morbilidad.*
58. *Mostrar casos vistos de un grupo dado en un periodo de tiempo dado.*
59. *Mostrar casos vistos por grupo de enfermedades, municipio, grupo de edades en un período de tiempo.*
60. *Mostrar remitidos de un municipio.*
61. *Mostrar remitidos por municipios.*

- 62. Mostrar el % anual pacientes dado un grupo.4
- 63. Mostrar cantidad de operaciones por clase de operación.
- 64. Mostrar el modelo de cirugía.
- 65. Mostrar el % anual de operaciones de una Clase.
- 66. Mostrar todos los pacientes operados en un día determinado.

2.3.4 Diagrama de casos de uso del sistema

Para facilitar el trabajo con los casos de uso y la organización de los elementos se agrupan los casos de uso en paquetes, los cuales se muestran a continuación.

Nombre del Paquete	Funcionalidades	Criterio de Agrupamiento	Actores
Seguridad	<ul style="list-style-type: none"> ✓ Autenticar usuario ✓ Cambiar contraseña 	Funcionalidades (Requerimientos funcionales)	Usuario
Administración	<ul style="list-style-type: none"> ✓ Gestionar Usuario 	Funcionalidades (Requerimientos funcionales)	Administrador
Gestión	<ul style="list-style-type: none"> ✓ Gestionar paciente cirugía ✓ Gestionar cirugía ✓ Gestionar clasificación de cirugía ✓ Gestionar salón ✓ Gestionar tipo de anestesia ✓ Gestionar tipo de cirugía ✓ Gestionar clase de operación ✓ Gestionar ayudantes ✓ Gestionar cirujanos ✓ Gestionar anestesiólogo ✓ Gestionar enfermeras de 	Funcionalidades (Requerimientos funcionales)	Administrador, Técnico de Cuerpo de Guardia y Técnico de Cirugía.

	<p>salón</p> <ul style="list-style-type: none"> ✓ Gestionar paciente de cuerpo de guardia ✓ Gestionar municipio ✓ Gestionar enfermedades ✓ Gestionar grupo de enfermedades 		
Reportes	<ul style="list-style-type: none"> ✓ Mostrar modelo CAAB ✓ Mostrar modelo EDA ✓ Mostrar modelo IRA ✓ Mostrar modelo SFA ✓ Mostrar modelo EDO ✓ Mostrar modelo accidente ✓ Mostrar modelo morbilidad ✓ Mostrar Casos vistos de un Grupo dado en un periodo de tiempo dado ✓ Mostrar casos vistos de un grupo determinado, de determinado municipio y un grupo de edad en un periodo de tiempo dado. ✓ Mostrar remitidos de un municipio. ✓ Mostrar remitidos por municipios. ✓ Mostrar el % anual pacientes dado un grupo. ✓ Mostrar cantidad de operaciones por clase 	Funcionalidades (Requerimientos funcionales)	Administrador, Técnico de Cuerpo de Guardia y Técnico de Cirugía.

	<p>de operación.</p> <ul style="list-style-type: none">✓ <i>Mostrar el modelo de cirugía.</i>✓ <i>Mostrar el % anual de operaciones de una Clase.</i>✓ <i>Mostrar todos los pacientes operados en un día determinado.</i>		
--	---	--	--

Tabla 5 Descripción de los paquetes

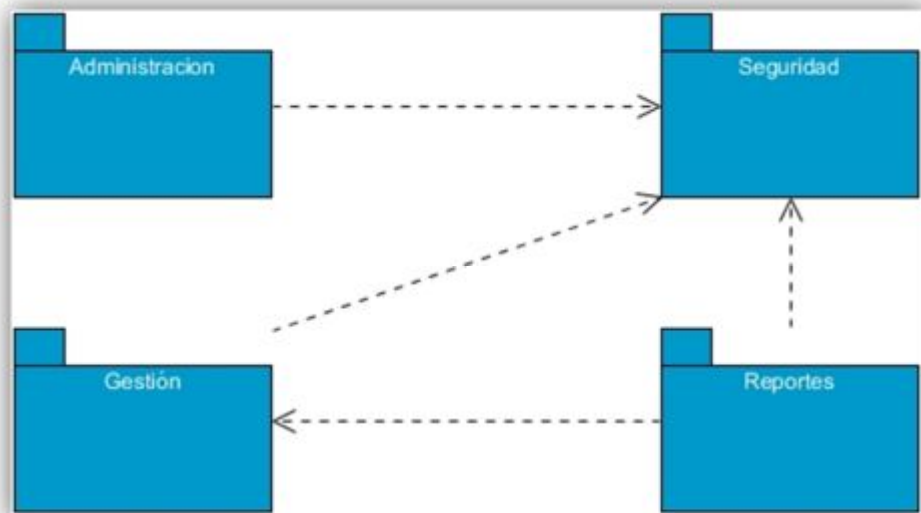


Figura 4 Diagrama de casos de uso por paquetes

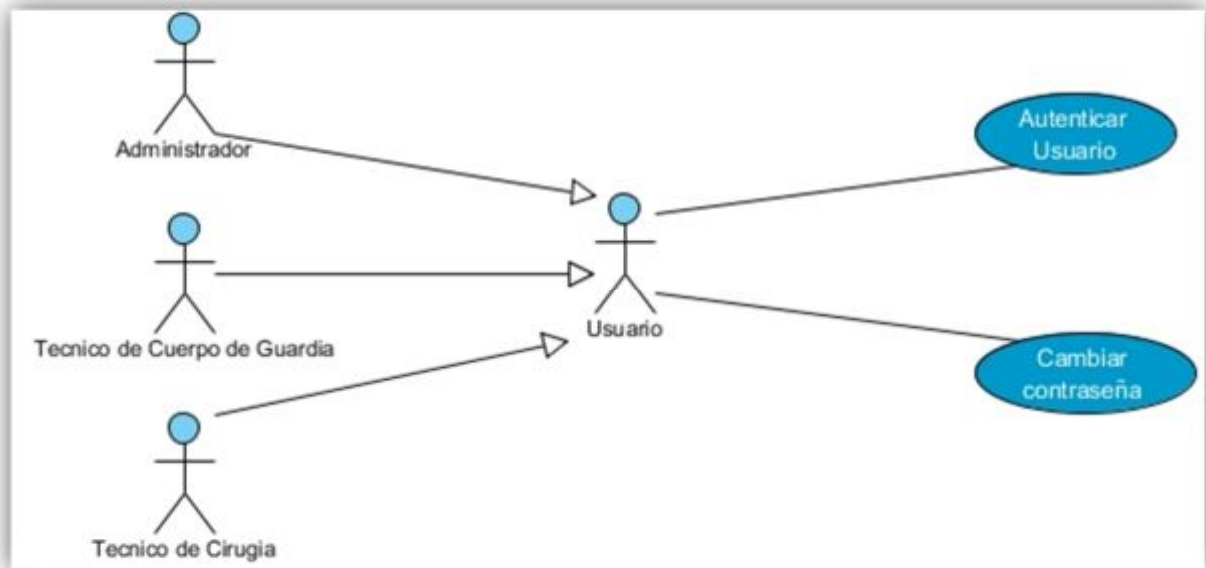


Figura 5 Diagramas de casos de uso: Paquete Seguridad

Descripción textual de los casos de uso Paquete Seguridad: [Anexo 1](#)



Figura 6 Diagramas de casos de uso: Paquete Administración

Descripción textual de los casos de uso Paquete Administración: [Anexo 2](#)

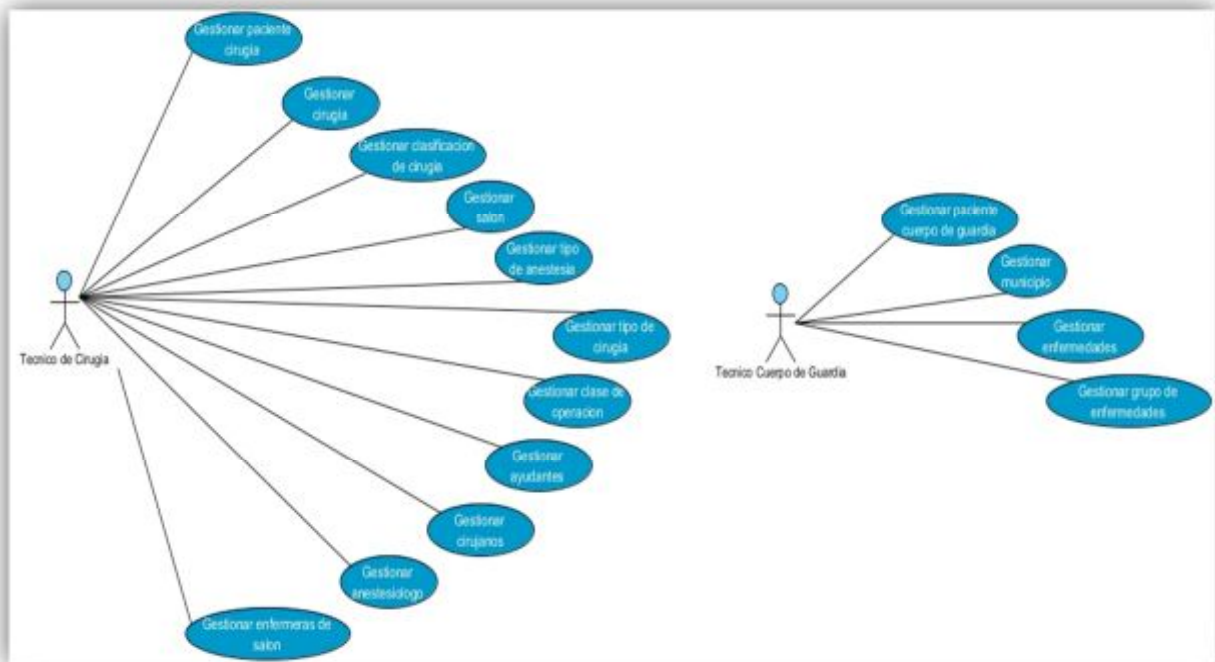


Figura 7 Diagramas casos de uso: Paquete Gestión

Descripción textual de los casos de uso Paquete Gestión: [Anexo 3](#)

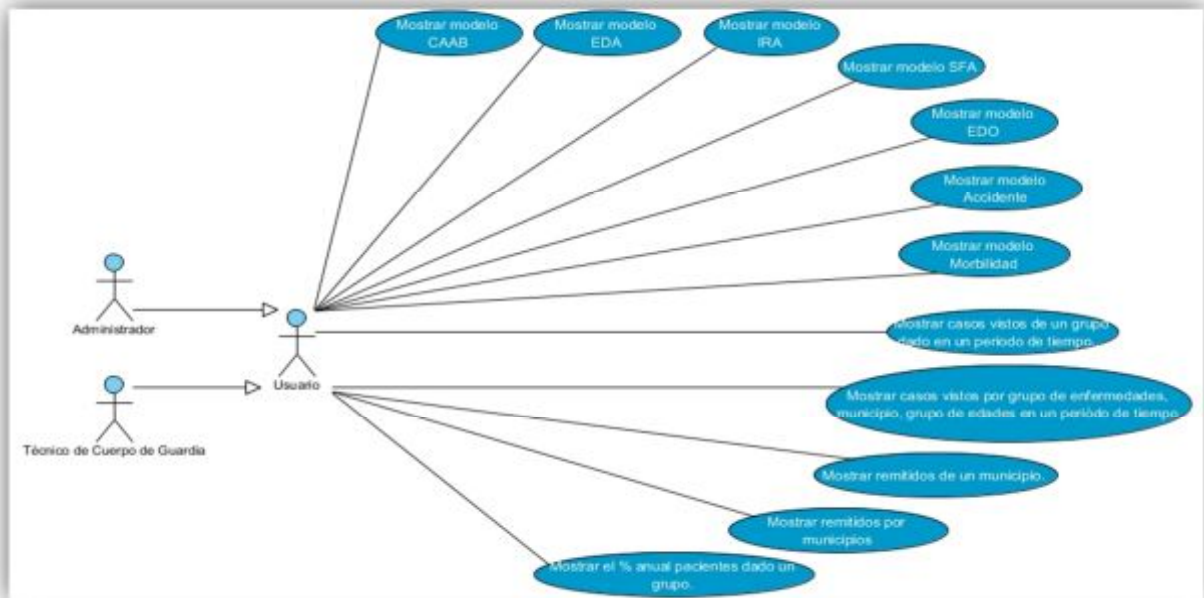


Figura 8 Diagramas casos de uso: Paquete Reportes

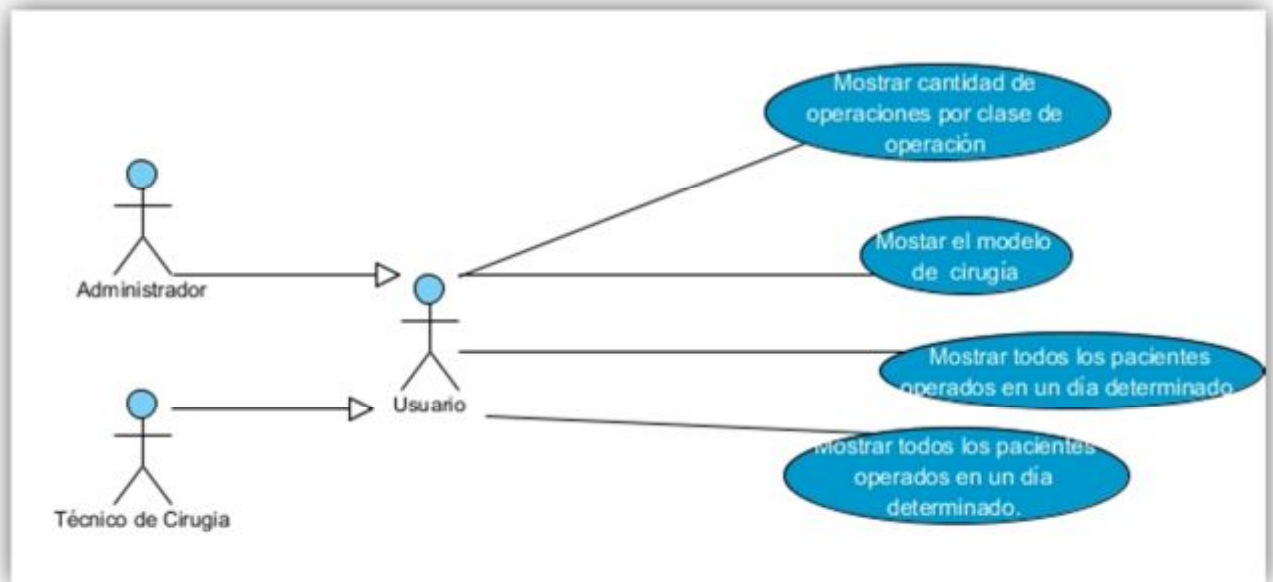


Figura 9 Diagramas casos de uso: Paquete Reportes

Descripción textual de los casos de uso Paquete Gestión: [Anexo 4](#)

Conclusiones

A través del modelo del sistema se identificaron actores del sistema (Técnico de Cuerpo de Guardia, Técnico de Cirugía, Administrador), además se obtuvieron 34 casos de usos del sistema que fueron descritos y documentados a través del correspondiente diagrama. Se especificaron también los requerimientos funcionales y no funcionales. Todo contribuye a una mejor distribución de cómo funcionará el software a implementarse, así como la documentación de cada respuesta a las necesidades del cliente.

A decorative graphic consisting of a blue wavy ribbon that flows from the top right, curves down, and then flows back up towards the bottom left. The ribbon has a slight 3D effect with a lighter blue shadow underneath.

Capítulo 3

Capítulo III Implementación del software Sistema Gestor de Pacientes del Hospital Pediátrico de Sancti Spíritus:

Introducción:

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería, es donde se fomenta la calidad en el desarrollo del software.

Es la única manera de traducir con precisión los requerimientos funcionales del cliente en un sistema o producto de software.

En el presente capítulo se plasman los resultados de la etapa del diseño. Se presentan diagramas como por ejemplo: diagrama de clases persistentes y modelo de datos, diagrama de componente y de despliegue. Además se describen los principios de diseño aplicados.

3.1 Diagramas de clases del diseño

Una clase de diseño es una abstracción de una clase o construcción similar en la implementación del sistema (Jacobson et al., 2000).

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Contiene: clases, asociaciones y atributos; interfaces, con sus operaciones y constantes; métodos; información sobre los tipos de atributos; navegabilidad y dependencias (Jacobson et al., 2000).

Los Diagramas de clases del diseño fueron elaborados a partir de los diferentes casos de uso del sistema:

Diagramas de clases del diseño: Paquete Seguridad ver [anexos 5](#).

Diagramas de clases del diseño: Paquete Administración ver [anexos 6](#).

Diagramas de clases del diseño: Paquete Reportes ver [anexos 7](#).

Diagramas de clases del diseño: Paquete Gestión ver [anexos 8](#)

3.2 Diseño de la base de datos

3.2.1 Diagrama de clases persistentes

En el diagrama de clases persistentes aparecen las clases que persisten, las cuales poseen la capacidad de mantener su valor en el espacio y en el tiempo (Jacobson et al., 2006). Está

compuesto por clases, asociaciones y atributos; interfaces, con sus operaciones y constantes; métodos; información sobre los tipos de atributos, entre otros.

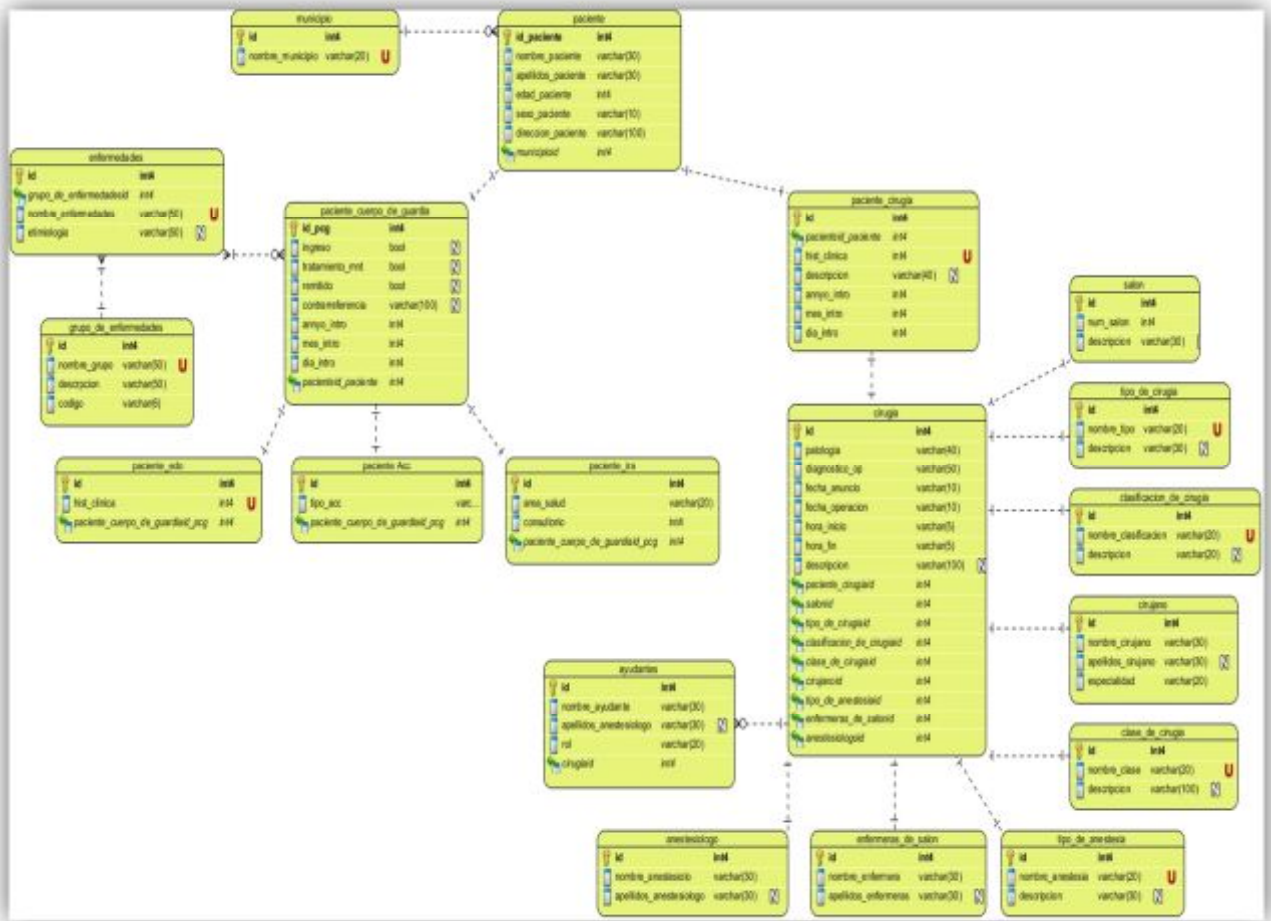


Figura 10 Diagrama de clases persistentes

3.3 Principios de diseño

3.3.1 Interfaz

Uno de los aspectos más notables en el uso de un sistema que lo puede llevar a un éxito o a un fracaso es la consistencia de la interfaz de usuario, es por ello que se necesita lograr un diseño sencillo y amigable; para lograr estos aspectos se tiene en cuenta la tipografía, navegación y colores. A continuación se muestran algunas características de la interfaz diseñada:

Se utilizaron colores agradables, suaves debido a que su uso requiere concentración, además de que brindan a la aplicación una visión de formalidad. El vocabulario a utilizar será el más adecuado para el usuario sin emplear palabras técnicas de informática. El software consta de un menú que aparece de manera permanente en el cual se encuentran todas las funcionalidades del sistema, el usuario podrá acceder a ellas en dependencia de los permisos que tenga. Además consta con una ayuda en la cual se encuentran descritas todas las funcionalidades del sistema.

3.3.2 Tratamientos de errores

Para garantizar la integridad del sistema, el mismo valida constantemente la información, esto se realiza con la intención de minimizar la entrada de errores por parte del usuario. En caso de errores se le comunica al usuario mediante mensaje de alerta en un lenguaje fácil de comprender.

3.3.3 Concepción de la ayuda

El sistema cuenta con una ayuda capaz de brindarle al usuario una detallada explicación de cómo sería la manipulación y funcionamiento del mismo. La ayuda está formada por un menú interno que contiene todas las funcionalidades del sistema. El usuario podrá navegar por cada uno de ellos y consultar su funcionalidad, descrita de un modo fácil de entender.

3.3.4 Seguridad

El sistema mantiene un fuerte mecanismo de seguridad, basado en un nombre de usuario y contraseña para el acceso al mismo. Las personas encargadas de actualizar cualquier información presente en la base de datos del sistema, tendrán nombres de usuario y contraseñas únicas, evitando que esta acción pueda ser realizada por cualquier otro usuario. Además los usuarios del sistema tendrán solamente permiso de acceso a las funcionalidades correspondientes a su rol. Para garantizar una mayor seguridad al sistema se utilizó el método de encriptación de cadenas MD5, el cual permite la encriptación de la información pero no descriptarla.

3.4 Estándares de codificación

Actualmente se hallan estándares de codificación para la mayoría de los lenguajes existentes. El uso de ellos partiendo de las convenciones definidas permite una mejor

comunicación entre los programadores creando las condiciones para la reusabilidad y el mantenimiento de los sistemas. Por lo que se decide que las variables, nombres de funciones, de consultas y objetos del documento son cortos, claros y describen su propósito. Los nombres de las clases se escriben con mayúscula, las variables con minúsculas y las funciones que están compuestas por más de una palabra se escribe primero con minúscula y la primera letra de las demás palabras con mayúscula.

Los objetos o tipos de control se nombran según el valor de su contenido. Los inicios y cierre de ámbito se encuentran alineados debajo de la declaración a la que pertenecen y se evitan si hay sólo una instrucción. Los signos lógicos y de operación se separan por un espacio antes y después de los mismos. El código se encuentra comentado de modo que no sea necesario revisar todo el código para entender lo que está programado. Se determinaron estos estándares de codificación basados en el estándar Java Code Convection.

3.5 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo del diseño, se implementan en términos de componentes, describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguajes de programación utilizados y cómo dependen unos componentes de otros (Jacobson et al., 2000).

3.5.1 Modelo de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuyen las funcionalidades entre los nodos de cómputo (Jacobson et al., 2000). El modelo representa dos nodos: en el lado del servidor se encuentra PostgreSQL como servidor de base de datos y se comunica con el cliente a través del protocolo TCP/IP. Se visualiza la aplicación en el lado del cliente.

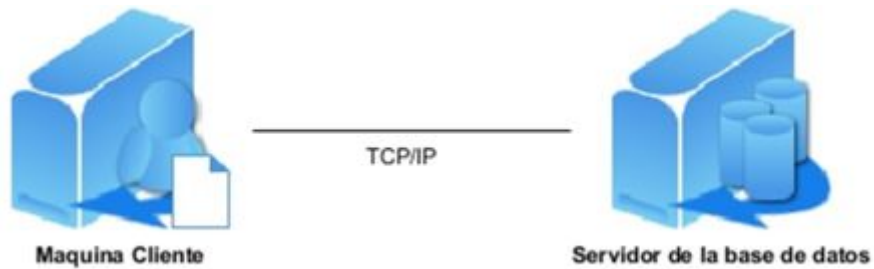


Figura 11 Diagrama de despliegue

3.5.2 Diagrama de componentes

Un diagrama de componentes muestra un conjunto de elementos del modelo, se emplea para modelar la vista estática de un sistema, muestra la organización y dependencias lógicas entre los componentes del software ya sean bibliotecas, ejecutables o componentes binarios.

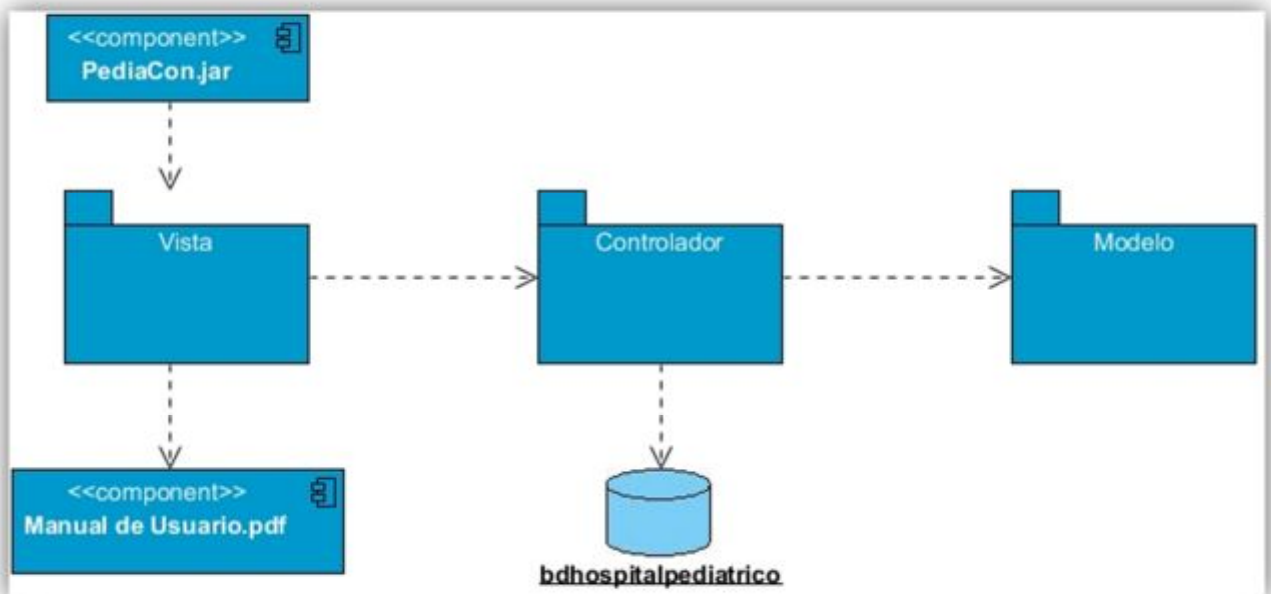


Figura 12 Diagrama de componentes

Componente	Descripción
<i>PediaCon.jar</i>	<i>Este componente representa el ejecutable de la aplicación</i>
<i>Vista</i>	<i>Dentro de este subsistema se encuentran</i>

	<i>todas las clases que muestran una interfaz de usuario.</i>
<i>Modelo</i>	<i>Este subsistema representa la lógica del negocio, dentro de él se encuentran las clases entidades con las entidades del negocio y sus atributos.</i>
<i>Controlador</i>	<i>Este componente representa la clase que facilita el acceso a datos a través de la cual se van a comunicar las clases entidades del modelo y la base de datos. Este componente es usado desde el subsistema Modelo.</i>
<i>bdhospitalpediatrico</i>	<i>Representa la base de datos de la aplicación.</i>

Tabla 6 Descripción de los componentes

3.6 Pruebas

Las pruebas son un flujo de trabajo donde se verifica el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros (Jacobson et al., 2000). En los siguientes subepígrafes se exponen las planificaciones de prueba, los procedimientos de prueba y los casos de prueba para algunos de los casos de uso más importantes del sistema según el juicio del autor.

3.6.1 Planificación de pruebas

El objetivo de la planificación de las pruebas es la representación de una estrategia de prueba, planificar su esfuerzo y estimar los requisitos de la misma. Se organiza el plan de pruebas utilizando un rango de valores de entrada. El modelo de casos de uso y los requisitos que debe tener el sistema para ejecutar correctamente el proceso, los mismos ayudan a definir un tipo adecuado de pruebas y a estimar el esfuerzo necesario para llevar a cabo las pruebas. Se desarrolla una estrategia de prueba para la iteración donde se decide qué tipo de pruebas ejecutar, cómo y cuándo ejecutarlas y cómo determinar si el esfuerzo de prueba tiene éxito (Jacobson et al., 2000).

A continuación se exponen la planificación de pruebas para cinco de los casos de uso del sistema.

-Caso de uso: Gestionar paciente de cuerpo de guardia.

Casos de prueba:

Insertar un paciente de cuerpo de guardia.

-Caso de Uso: Gestionar paciente de cirugía.

Casos de prueba:

Insertar un paciente de cirugía.

-Caso de uso: Gestionar Cirugía.

Casos de prueba:

Insertar una cirugía.

-Caso de uso: Mostar modelo de Enfermedades de difusión obligatoria.

Casos de prueba:

Mostrar todos los pacientes que pertenezcan al grupo de Enfermedades de difusión obligatoria

-Caso de uso: Mostrar los pacientes operados en un día determinado.


Casos de prueba:

Seleccionar el día para mostrar los pacientes operados.

3.6.2 Procedimientos de prueba


Los procedimientos de pruebas especifican como realizar uno o varios casos de pruebas. Estos incluyen información adicional, como los valores de entrada del caso de uso a utilizar, la forma en la que estos valores han de ser introducidos en el interfaz de usuario y lo que hay que verificar. A continuación se exponen los procedimientos de pruebas para los casos de uso del sistema seleccionados durante la planificación de pruebas.

Procedimiento de prueba del caso de uso: Gestionar paciente de cuerpo de guardia.


- 1) Hacer clic en el ítem Gestionar paciente de cuerpo de guardia que se encuentra dentro de la barra de menú, en el menú Gestión de datos y dentro del submenú Cuerpo de Guardia, o simplemente dar clic en el botón  que se encuentra en la barra de herramientas.*

- 2) Se realiza una consulta en la base de datos tomando todos los pacientes pertenecientes al área de Cuerpo de guardia.
- 3) El reporte se llena con los datos consultados.
- 4) El reporte es mostrado al usuario.

Procedimiento de prueba del caso de uso: Gestionar paciente de cirugía.

- 1) Hacer clic en el ítem *Gestionar paciente de cirugía* que se encuentra dentro de la barra de menú, en el menú *Gestión de datos* y dentro del submenú *Cirugía*, o simplemente dar clic en el botón  que se encuentra en la barra de herramientas.
- 2) Se realiza una consulta en la base de datos tomando todos los pacientes pertenecientes al área de cirugía.
- 3) El reporte se llena con los datos consultados.
- 4) El reporte es mostrado al usuario.

Procedimiento de prueba del caso de uso: Gestionar cirugía.

- 1) Hacer clic en el ítem *Gestionar cirugía* que se encuentra dentro de la barra de menú, en el menú *Gestión de datos* y dentro del submenú *Cirugía*, o simplemente dar clic en el botón  que se encuentra en la barra de herramientas.
- 2) Se realiza una consulta en la base de datos tomando todas las cirugías.
- 3) El reporte se llena con los datos consultados.
- 4) El reporte es mostrado al usuario.

Procedimiento de prueba del caso de uso: Mostar modelo de Enfermedades de difusión obligatoria.

- 1) Hacer clic en el ítem *Mostar modelo de Enfermedades de difusión obligatoria* que se encuentra dentro de la barra de menú, en el menú *Reportes* y dentro del submenú *modelos*.
- 2) Se realiza una consulta en la base de datos tomando todos los pacientes que pertenezcan a ese grupo.
- 3) El reporte se llena con los datos consultados.

- 4) El reporte es mostrado al usuario.

Procedimiento de prueba del caso de uso: Mostrar los pacientes operados en un día determinado.

- 1) *Hacer clic en el ítem Mostrar los pacientes operados en un día determinado que se encuentra dentro de la barra de menú, en el menú reportes y dentro del submenú información útil.*
- 2) *El sistema muestra una ventana llamada Mostar pacientes operados en.*
- 3) *Se muestra la ventana llamada Mostar pacientes operados en. La ventana muestra un campo de fecha, donde se seleccionara la fecha que se realizará la consulta para mostrar los pacientes operados en la misma.*
- 4) *Seleccionar la fecha para mostrar los pacientes operados.*
- 5) *Dar clic en el botón que lleva por nombre Generar*
- 6) *El sistema pasa como parámetro los datos de la caja de texto descrita en el paso cuatro al procedimiento almacenado en el sistema.*
- 7) *El proceso almacenado se ejecuta y llena el reporte con los datos de los pacientes.*
- 8) *El usuario visualiza los pacientes operados en el día seleccionado.*

3.6.3 Casos de pruebas.

Los casos de pruebas especifican la forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. Es un conjunto de entradas y resultados esperados que ejercitan a un componente con el propósito de causar fallas y detectar defectos. A continuación se exponen los casos de pruebas realizados a los diferentes casos de uso del sistema seleccionados para la planificación y los procedimientos de prueba.

Caso de uso: Gestionar paciente de cuerpo de guardia
Caso de prueba: <i>Insertar un paciente de cuerpo de guardia.</i>
Entrada: <i>Un usuario intenta pulsar el botón Insertar sin antes haber introducido correctamente los datos pertenecientes al paciente de cuerpo de guardia.</i>
Resultado: <i>Se muestra al usuario un mensaje diciéndole “Debe Introducir <nombre del campo que falta>”.</i>
Condiciones: <i>Estar autenticados con el rol técnico de cuerpo de guardia o administrador.</i>

Tabla 7 Caso de prueba para el caso de uso Gestionar paciente de cuerpo de guardia

Caso de uso: Gestionar paciente de cuerpo de cirugía
Caso de prueba: <i>Insertar un paciente de cirugía.</i>
Entrada: <i>Un usuario intenta pulsar el botón Insertar sin antes haber introducido correctamente los datos pertenecientes al paciente de cirugía.</i>
Resultado: <i>Se muestra al usuario un mensaje diciéndole “Debe Introducir <nombre del campo que falta>”.</i>
Condiciones: <i>Estar autenticados con el rol técnico de cirugía o administrador.</i>

Tabla 8 Caso de prueba para el caso de uso Gestionar paciente de cirugía

Caso de uso: Gestionar cirugía
Caso de prueba: <i>Insertar una cirugía.</i>
Entrada: <i>Un usuario intenta pulsar el botón Insertar sin antes haber introducido correctamente los datos pertenecientes a la cirugía.</i>
Resultado: <i>Se muestra al usuario un mensaje diciéndole “Debe Introducir <nombre del campo que falta>”.</i>
Condiciones: <i>Estar autenticados con el rol técnico de cirugía o administrador.</i>

Tabla 9 Caso de prueba para el caso de uso Gestionar cirugía

Caso de uso: Mostrar los pacientes operados en un día determinado.
Caso de prueba: <i>Seleccionar el día para mostrar los pacientes operados.</i>
Entrada: <i>Un usuario intenta pulsar el botón generar sin antes haber seleccionado la fecha que quiere que se muestren los pacientes.</i>
Resultado: <i>Se muestra al usuario un mensaje diciéndole “Debe Introducir fecha”.</i>
Condiciones: <i>Existir pacientes operados en esa fecha en la base de datos del sistema.</i>

Tabla 10 Caso de prueba para el caso de uso Mostrar los pacientes operados en un día determinado

Conclusiones

Durante el desarrollo de este capítulo se elaboró el diagrama de componente en el cual se graficó de manera clara las relaciones entre los subsistemas, los componentes y la base de

Capítulo 3

datos; el diagrama de despliegue y lo referente a la base de datos. Además se describieron los principios de diseño seguidos, específicamente el diseño de la interfaz de usuario, los estándares de codificación, la concepción de la ayuda, el tratamiento de excepciones y la seguridad del sistema.

A decorative graphic consisting of a blue wavy ribbon that flows from the top right towards the bottom left, curving around the text. The ribbon has a gradient from light blue to a darker blue and a slight 3D effect with shadows.

Conclusiones

Conclusiones

Con la realización del presente proyecto se arribó a las siguientes conclusiones:

- ✓ *El estudio de los fundamentos teóricos y metodológicos para el desarrollo de este software permitió seleccionar las tecnologías y herramientas más adecuadas y se determinó que la metodología RUP era la apropiada para el análisis, diseño e implementación del sistema.*
- ✓ *Se diseñó un software partiendo de la descripción de los procesos del negocio y de la modelación de los casos de uso correspondientes al sistema, así como los requisitos funcionales y no funcionales definidos por el usuario. Se empleó la metodología RUP, como lenguaje de modelación el Lenguaje Unificado de Modelado y como herramienta de modelado se utilizó Visual Paradigm.*
- ✓ *Se implementó un software capaz de perfeccionar la gestión de la información sobre los pacientes del Hospital Pediátrico en Sancti Spíritus, utilizando el lenguaje de programación Java, la herramienta de desarrollo NetBeans 6.8 y el Sistema Gestor de Base de Datos PostgreSQL 8.4 por ser tecnologías de código abierto y tener la característica de integrarse para funcionar en cualquier plataforma.*

A decorative graphic consisting of a blue wavy ribbon that flows vertically down the right side of the page, then curves horizontally across the bottom. The ribbon has a 3D effect with a lighter blue shadow on its underside.

Recomendaciones

Recomendaciones

- ✓ *Agregar como nueva funcionalidad al software el manejo de las historias clínicas de los pacientes de cirugía.*
- ✓ *Gestionar nuevas áreas (es decir áreas como miscelánea, psiquiatría, etc.).*

A decorative graphic consisting of a blue wavy ribbon that flows vertically down the right side of the page, then curves horizontally across the bottom. The ribbon has a 3D effect with lighter blue highlights and shadows.

Bibliografía

Bibliografía

- Alonso, E. D., & Zacca, E. (2011). *Sistema de salud de Cuba*. *Salud Pública de México*, Vol. 53(No. 2). Recuperado a partir de <http://www.scielosp.org/pdf/spm/v53s2/12.pdf>
- Álvarez, E. (2011, febrero 11). *Aprende Java Netbeans: ¿Qué es Netbeans?* Recuperado 25 de marzo de 2015, a partir de <http://java-netbeans1.blogspot.com/2011/02/que-es-netbeans.html>
- Angeles, J. A. de los. (2002, Febrero 16). *Programación Orientada a Objetos. Introducción a la programación orientada a objetos*. Universidad Autónoma de Puebla (México). Recuperado a partir de http://pdi.topografia.upm.es/m.manso/docencia/Informatica_plan92/Curso-2002-2003/poo.pdf
- Arévalo, J. A. (2008, febrero 15). *Gestión de la Información, gestión de contenidos y conocimiento*. Universidad de Salamanca. Facultad de Traducción y Documentación.
- Bustamante, P., Aguinaga, I., Aibar, M., Alaizola, L., & Lazcano, I. (2004, febrero). *Aprenda C++ Básico como si estuviera en primero*. Recuperado a partir de <http://www4.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/cpp/basico/cppbasico.pdf>
- Calderón. (2009). *Metodologías Ágiles*.
- Camejo, I. G. (2000). *Algunas reflexiones sobre el concepto de información y sus implicaciones para el desarrollo de la información*. *ACIMED*, Vol. 8(No. 30). Recuperado a partir de http://bvs.sld.cu/revistas/aci/vol8_3_00/aci05300.pdf
- Capriolo, R. P. (2011). *Java y NetBeans*. Recuperado a partir de

http://javaagricola.wikispaces.com/file/view/0_Java+y+NetBeans.pdf

Cecilia, B., & Ivonne, C. (2009, septiembre). Monografía sobre la Metodología de Desarrollo de Software, Rational Unified Process (RUP). Universidad Don Bosco. Recuperado a partir de http://rd.udb.edu.sv:8080/jspui/bitstream/123456789/257/1/47400_tesis.pdf

Chappell, D., & Kirk, S. (2006). De N-capas a .NET. Desarrollo de aplicaciones. Recuperado a partir de <http://www.microsoft.com/spanish/msdn/articulos/archivo/081102/voices/dncapas.asp>

Curto, J. (2006). Reflexiones sobre la Tecnología de la Información. Information Management.

Daniela, M. R., Esteban, M. P. P., & Belem, R. R. D. (2012, abril 17). Segundo trabajo: Herramienta Case Visual Paradigm | Herramientas Automatizadas. Recuperado a partir de <http://dianbeel.blogspot.com/2012/06/segundo-trabajo-herramienta-case-visual.html>

Date, C. J. (2001). Introducción a los sistemas de bases de datos (7ma ed.). Pearson Educación.

Echeverría, R. R., Sánchez, E. S., & Álvaro Prieto Ramos. (2011). Programación Orientada a Objetos. Recuperado a partir de <http://www.libreriaalvaro.com/libropoo.pdf>

Equipo de desarrollo de PostgreSQL. (2001). Tutorial de PostgreSQL.

Fowler, M., & Scott, K. (1999). UML gota a gota. Pearson Educación. Recuperado a partir de <https://books.google.com.cu/books?id=AL0YkFeaHwIC>

Ginestà, M. G., & Mora, O. P. (2007). Bases de datos en PostgreSQL.

Recuperado a partir de http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf

Introducción a las bases de datos Fundamentos de diseño de bases de datos. (S. f.). Recuperado a partir de <http://elvex.ugr.es/idbis/db/docs/intro/B%20Bases%20de%20Datos.pdf>

Jacobson, I., Booch, G., & James Rumbaugh. (2000). *El proceso de unificado de desarrollo de software*. Addison Wesley.

Jacobson, I., Booch, G., & James Rumbaugh. (2006). *El proceso de unificado de desarrollo de software*. Félix Varela.

Krall, C. (s. f.). *¿Qué es y para qué sirve UML? Versiones de UML (Lenguaje Unificado de Modelado). Tipos de diagramas UML*. Recuperado 10 de marzo de 2015, a partir de http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=611

Landay, J. (2009). Recuperado a partir de <http://www.gnuconsultores.com/es/ingenieria/desarrollo/escritorio>

Laura. (2011, Agosto 1). *PROGRAMACIÓN JAVA: Entorno de desarrollo integrado (IDE)*. Recuperado a partir de <http://programacion-laura.blogspot.com/2011/08/entorno-de-desarrollo-integrado-ide.html>

López, P., & Ruiz, F. (2011, marzo). *Ingeniería del Software 1*. Universidad Cantabria – Facultad de Ciencias. Recuperado a partir de <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-i/materiales-de-clase-1/is1-t02-trans.pdf>

Martínez, R. (2010). *Sobre PostgreSQL* | www.postgresql.org.es. Recuperado 15 de abril de 2014, a partir de

http://www.postgresql.org.es/sobre_postgresql

Massabot, N. E. R., Viera, R. M. F., & Pérez, E. R. J. (2005). Los Registros Médicos en Cuba. *Revista Cubana Salud Pública*, 4.

Mirian Milagros Díaz Flores. (2004). RUP vs Xp. Recuperado a partir de <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>

Mora, S. L. (2002). Programación de aplicaciones web: Historia, Principios básicos y Clientes Web. Club Universitario. Recuperado a partir de <http://gpsl.dlsi.ua.es/~slujan/programacion-aplicaciones-web-historia-principios-basicos-clientes-web>

Oficina Nacional de Estadística. (2007, Enero). CARACTERÍSTICAS Y EVOLUCIÓN DEL SISTEMA ESTADÍSTICO NACIONAL DE CUBA.

Orallo, E. H. (2002, septiembre 2). El Lenguaje Unificado de Modelado (UML). Recuperado a partir de [url:http://www.disca.upv.es/enheror/pdf/ActaUML.PDF](http://www.disca.upv.es/enheror/pdf/ActaUML.PDF)

Ortí, C. B. (2010, septiembre 20). LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN (T.I.C.). Universidad de Valencia. Recuperado a partir de <http://www.uv.es/~bellochc/pdf/pwtic1.pdf>

Peláez, J. (2009, mayo 29). Arquitectura basada en capas. Recuperado a partir de <http://geeks.ms/blogs/jkpelaez/archive/2009/05/29/arquitectura-basada-en-capas.aspx>

Pérez, M. T. G. (2010, mayo). Sistemas Gestores de Base de Datos. *Revista Digital «Innovación y Experiencias Educativas»*, (No. 30). Recuperado a partir de http://www.csi-csif.es/andalucia/modules/mod_ense/revista/pdf/Numero_30/TERES

[A GARZON 1.pdf](#)

Pressman, R. S. (2010). *Ingeniería del Software. Un Enfoque Práctico* (7ma ed.). McGraw-Hill. Recuperado a partir de <http://eva.sepyc.gob.mx:8383/greenstone3/sites/localsite/collect/ciencia1/index/assoc/HASH015f/ceb375c1.dir/33040073.pdf>

Quiroga, L. A. (2002). *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. Acimed, Vol. 10(No. 05). Recuperado a partir de http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm

Ramírez, I. (2013, abril 10). *¿Qué es Java? | A fondo | Softonic*. Recuperado 16 de febrero de 2015, a partir de <http://articulos.softonic.com/que-es-java>

Riola, J. C. C. (2008). *METODOLOGÍAS ÁGILES HERRAMIENTAS Y MODELO DE DESARROLLO PARA APLICACIONES JAVA EE COMO METODOLOGÍA EMPRESARIAL*. UPC - Barcelona. Recuperado a partir de <url:https://upcommons.upc.edu/pfc/bitstream/2099.1/5608/1/50015.pdf>

Rodríguez, A. (2006). *¿QUÉ ES JAVA? CONCEPTO DE PROGRAMACIÓN ORIENTADA A OBJETOS FRENTE A PROGRAMACIÓN ESTRUCTURADA*. Recuperado a partir de http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=318

Santillán, L. A. C., Ginestà, M. G., & Mora, Ó. P. (2007). *Bases de datos en MySQL*. Recuperado a partir de http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf

- Santos, M. Á. S. (2007, diciembre). *Conceptos Básicos de Lenguajes de Programación*. Recuperado a partir de http://ggyma.geo.ucm.es/docencia/documentos/informatica/Informatica_5.pdf
- Sanz, C. H. i. (2012, noviembre 5). *¿Qué es el PostgreSQL? | NaN.tic*. Recuperado 15 de abril de 2014, a partir de <http://www.nan-tic.com/es/2012/que-es-el-postgresql/>
- Seco, J. A. G. (2004). *El lenguaje de programación C#*. Recuperado a partir de <http://dis.um.es/~bmoros/privado/bibliografia/LibroCsharp.pdf>
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *Fundamentos De Bases De Datos (4ta. Edición)*. McGraw- Hill. Recuperado a partir de <https://unefazuliasistemas.files.wordpress.com/2011/04/fundamentos-de-bases-de-datos-silberschatz-korth-sudarshan.pdf>
- Silva, C., Kinderman, H., & Velásquez, V. (2011, noviembre 26). *PostgreSQL: Ventajas y Desventajas*. Recuperado 16 de febrero de 2015, a partir de <http://postgresql-ads.blogspot.com/2011/11/ampliamente-popular-ideal-para.html>
- Sistema de información estadística para los centros de atención a personas con VIH/SIDA del país - Ilustrados! (s. f.)*. Recuperado a partir de <http://www.ilustrados.com/tema/12055/Sistema-informacion-estadistica-para-centros-atencion.html>
- Uñoja, R. H. (2015, marzo 10). *Ingeniería de Software: METODOLOGIAS DE DESARROLLO DE SOFTWARE TRADICIONALES VS AGILES*. Recuperado 10 de marzo de 2015, a partir de <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-de-software.html>
- Valencia, U. de. (2004). *Eclipse (2.1) y Java*. Recuperado a partir de

http://www.uv.es/~jgutierrez/MySQL_Java/TutorialEclipse.pdf

Vázquez, M. E., Álvarez, R. de la C. L., & Padrón, H. U. (2011). Sistema de información estadística para los centros de atención a personas con VIH/SIDA del país - Ilustrados! Recuperado 2 de marzo de 2015, a partir de <http://www.ilustrados.com/tema/12055/Sistema-informacion-estadistica-para-centros-atencion.html>

A decorative graphic consisting of a blue wavy ribbon that flows from the top right towards the bottom left, curving and overlapping itself. The ribbon has a gradient from a darker blue to a lighter, almost white blue, giving it a three-dimensional, ethereal appearance.

Anexos

Anexos

Anexo 1 Descripción textual de los casos de uso Paquete Seguridad:

CU # 1 Autenticar usuario	
Actor	Usuario (inicia)
Propósito	Controlar el acceso al software.
Resumen: comienza cuando un usuario solicita entrar al software introduciendo sus datos de usuario. Si el usuario está registrado y sus datos son correctos accederá al software con determinados permisos; en caso contrario se le informará con un mensaje. El caso de uso finaliza cuando se ejecuta una de las operaciones descritas anteriormente.	
Referencias	RF-1
Prototipo	Ver anexo 8

Anexo 1.1 Descripción del caso de uso Autenticar Usuario

CU # 2 Cambiar contraseña	
Actor	Usuario (inicia)
Propósito	Mantener la seguridad del sistema y la confiabilidad de manera que la contraseña sólo sea conocimiento del usuario.
Resumen: El caso de uso comienza cuando el usuario solicita el cambio de su contraseña, introduce su nombre de usuario, contraseña anterior y la nueva contraseña dos veces, de ser correctos los datos se guarda la nueva contraseña, de lo contrario se emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones anteriores.	
Referencias	RF-2
Prototipo	Ver anexo 9

Anexo 1.2 Descripción del caso de uso Cambiar Contraseña

Anexo 2 Descripción textual de los casos de uso Paquete Administración:

CU # 3 Gestionar usuarios	
Actor	Administrador(inicia)
Propósito	Controlar el acceso al software mediante el establecimiento de los

	<i>privilegios de los usuarios.</i>
Resumen:	<i>Comienza cuando el usuario accede dentro del menú usuarios al submenú gestionar usuarios para adicionar un nuevo usuario, modificar o eliminar uno existente; luego introduce los datos pertinentes. De ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones anteriores.</i>
Referencias	<i>RF-3,4,5</i>
Prototipo	<u>Ver anexo 10</u>

Anexo 2.1 Descripción del caso de uso Gestionar usuario

Anexo 3 Descripción textual de los casos de uso Paquete Gestión:

CU # 4 Gestionar paciente cirugía	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar los pacientes del área de cirugía.</i>
Resumen:	<i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar paciente de cirugía para introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>
Referencias	<i>RF-6,7,8</i>
Prototipo	<u>Ver Anexo 11</u>

Anexo 3.1 Descripción del caso de uso Gestionar paciente de cirugía

CU # 5 Gestionar cirugía	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar las cirugías realizadas.</i>
Resumen:	<i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar cirugía para introducir una nueva o para modificar o eliminar una existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las</i>

<i>operaciones antes descritas.</i>	
Referencias	<i>RF-9,10,11</i>
Prototipo	Ver anexo 12

Anexo 3.2 Descripción del caso de uso Gestionar cirugía

CU # 6 Gestionar clasificación de cirugía	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar las clasificaciones de cirugía.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar clasificación de cirugía para introducir una nueva o para modificar o eliminar una existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>	
Referencias	<i>RF-12,13,14</i>
Prototipo	Ver anexo 13

Anexo 3.3 Descripción del caso de uso Gestionar clasificación de cirugía

CU # 7 Gestionar salón	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar los salones existentes.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar salón para introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>	
Referencias	<i>RF-15,16,17</i>
Prototipo	Ver Anexo 14

Anexo 3.4 Descripción del caso de uso Gestionar salón

CU # 8 Gestionar tipo de anestesia	
---	--

Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar los tipos de anestesia existentes.</i>
Resumen:	<i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar tipo de anestesia para introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>
Referencias	<i>RF-18,19,20</i>
Prototipo	Ver Anexo 15

Anexo 3.5 Descripción del caso de uso Gestionar tipo de anestesia

CU # 9 Gestionar tipo de cirugía	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar los tipos de cirugías existentes.</i>
Resumen:	<i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar tipo de cirugía para introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>
Referencias	<i>RF-21,22,23</i>
Prototipo	Ver Anexo 16

Anexo 3.6 Descripción del caso de uso Gestionar tipo de cirugía

CU # 10 Gestionar clase de cirugía	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar las clases de cirugía existentes.</i>
Resumen:	<i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar clase de cirugía para introducir una nueva o para modificar o eliminar una existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>
Referencias	<i>RF-24,25,26</i>
Prototipo	Ver Anexo 17

Anexo 3.7 Descripción del caso de uso Gestionar clase de cirugía

CU # 11 Gestionar ayudantes	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar los ayudantes.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar ayudantes para introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>	
Referencias	<i>RF-27,28,29</i>
Prototipo	Ver Anexo 18

Anexo 3.8 Descripción del caso de uso Gestionar ayudantes

CU # 12 Gestionar cirujanos	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar los cirujanos existentes.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar cirujanos para introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>	
Referencias	<i>RF-30,31,32</i>
Prototipo	Ver Anexo 19

Anexo 3.9 Descripción del caso de uso Gestionar cirujanos

CU # 13 Gestionar anestesiólogos	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar los anestesiólogos existentes.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar anestesiólogo para</i>	

<i>introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>	
Referencias	RF-33,34,35
Prototipo	Ver Anexo 20

Anexo 3.10 Descripción del caso de uso Gestionar anestesiólogos

CU # 14 Gestionar enfermeras de salón	
Actor	<i>Técnico de Cirugía (inicia)</i>
Propósito	<i>Controlar las enfermeras de salón existentes.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cirugía y dentro de este accedemos al menú gestionar enfermeras de salón para introducir una nueva o para modificar o eliminar una existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>	
Referencias	RF-36,37,38
Prototipo	Ver Anexo 21

Anexo 3.11 Descripción del caso de uso Gestionar enfermeras de salón

CU # 15 Gestionar pacientes de cuerpo de guardia	
Actor	<i>Técnico de Cuerpo de Guardia(inicia)</i>
Propósito	<i>Controlar los pacientes del área de cuerpo de guardia.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú pacientes de cuerpo de guardia para introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>	
Referencias	RF-39,40,41
Prototipo	Ver Anexo 22

Anexo 3.12 Descripción del caso de uso Gestionar pacientes de cuerpo de guardia

CU # 16		Gestionar municipio
Actor	<i>Técnico de Cuerpo de Guardia(inicia)</i>	
Propósito	<i>Controlar los municipios existentes.</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cuerpo de guardia y dentro de este accedemos al menú gestionar municipio para introducir uno nuevo o para modificar o eliminar uno existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>		
Referencias	<i>RF-42,43,44</i>	
Prototipo	Ver Anexo 23	

Anexo 3.13 Descripción del caso de uso Gestionar municipio

CU # 17		Gestionar enfermedades
Actor	<i>Técnico de Cuerpo de Guardia(inicia)</i>	
Propósito	<i>Controlar las enfermedades existentes.</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cuerpo de guardia y dentro de este accedemos al menú gestionar enfermedades para introducir una nueva o para modificar o eliminar una existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.</i>		
Referencias	<i>RF-45,46,47</i>	
Prototipo	Ver Anexo 24	

Anexo 3.14 Descripción del caso de uso Gestionar enfermedades

CU # 18		Gestionar grupo enfermedades
Actor	<i>Técnico de Cuerpo de Guardia(inicia)</i>	
Propósito	<i>Controlar los grupos de enfermedades existentes.</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú gestión de datos al submenú cuerpo de guardia y dentro de este accedemos al menú gestionar grupo de</i>		

CU # 21		Mostrar modelo IRA
Actor	usuario(inicia)	
Propósito	Mostrar los pacientes que pertenecen al grupo Infección Respiratoria Aguda (IRA).	
<p>Resumen: Comienza cuando el usuario accede dentro del menú reportes al submenú modelos, luego debe seleccionar “Mostrar el modelo de Infección Respiratoria Aguda”, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</p>		
Referencias	RF-53	
Prototipo	Ver Anexo 28	

Anexo 4.3 Descripción del caso de uso mostrar modelo IRA

CU # 22		Mostrar modelo SFA
Actor	usuario(inicia)	
Propósito	Mostrar los pacientes que pertenecen al grupo Síndrome Febril Agudo (SFA).	
<p>Resumen: Comienza cuando el usuario accede dentro del menú reportes al submenú modelos, luego debe seleccionar “Mostrar el modelo de Síndrome Febril Agudo”, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</p>		
Referencias	RF-54	
Prototipo	Ver Anexo 29	

Anexo 4.4 Descripción del caso de uso mostrar modelo SFA

CU # 23		Mostrar modelo EDO
Actor	usuario(inicia)	
Propósito	Mostrar los pacientes que pertenecen al grupo Enfermedades de Difusión Obligatoria (EDO).	
<p>Resumen: Comienza cuando el usuario accede dentro del menú reportes al submenú modelos, luego debe seleccionar “Mostrar modelo de Enfermedades de Difusión Obligatoria”, en caso de existir información el sistema la muestra, en caso contrario</p>		

<i>muestra un mensaje comunicando que no hay información.</i>	
Referencias	<i>RF-55</i>
Prototipo	Ver Anexo 30

Anexo 4.5 Descripción del caso de uso mostrar modelo EDO

CU # 24		Mostrar modelo accidentes
Actor	<i>usuario(inicia)</i>	
Propósito	<i>Mostrar los pacientes que pertenecen al grupo accidentes</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú modelos, luego debe seleccionar “Mostrar modelo de accidentes”, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>		
Referencias	<i>RF-56</i>	
Prototipo	Ver Anexo 31	

Anexo 4.6 Descripción del casos de uso mostrar modelo accidentes

CU # 25		Mostrar modelo morbilidad
Actor	<i>usuario(inicia)</i>	
Propósito	<i>Mostrar los pacientes que pertenecen al grupo morbilidad</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú modelos, luego debe seleccionar “Mostar modelo de morbilidad”, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>		
Referencias	<i>RF-57</i>	
Prototipo	Ver Anexo 32	

Anexo 4.7 Descripción del caso de uso mostrar modelo accidentes

Mostrar casos de un Grupo dado en un período de fecha

CU # 26		Mostrar casos de un Grupo dado en un período de fecha.
Actor	<i>usuario(inicia)</i>	
Propósito	<i>Mostrar los casos existentes pertenecientes a un grupo de enfermedades en un periodo de tiempo definido por el usuario.</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú</i>		

<i>Información útil, luego debe seleccionar “Mostrar casos de un Grupo dado en un período de fecha”, luego debe elegir el grupo y el periodo de tiempo en que desee buscar los pacientes, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>	
Referencias	RF-58
Prototipo	Ver Anexo 33

Anexo 4.8 Descripción del caso de uso mostrar casos de un Grupo dado en un período de fecha.

Mostrar casos de una Enfermedad por Grupo de Edades y Municipios

CU # 27 Mostrar casos de una Enfermedad por Grupo de Edades y municipios.	
Actor	<i>usuario(inicia)</i>
Propósito	<i>Mostrar los casos de una Enfermedad por Grupo de Edades y Municipios.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú Información útil, luego debe seleccionar “Mostrar los casos de una Enfermedad por Grupo de Edades y Municipios.”, luego debe elegir el grupo y el municipio, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>	
Referencias	RF-59
Prototipo	Ver Anexo 34

Anexo 4.9 Mostrar casos de una Enfermedad por Grupo de Edades y municipios.

CU # 28 Mostrar remitidos de un municipio	
Actor	<i>usuario(inicia)</i>
Propósito	<i>Mostrar los remitidos de un municipio en específico.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú Información útil, luego debe seleccionar “Mostrar remitidos de un municipio”, luego debe elegir el municipio en que desee buscar los pacientes, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje</i>	

<i>comunicando que no hay información.</i>	
Referencias	<i>RF-60</i>
Prototipo	Ver Anexo 35

Anexo 4.10 Descripción del caso de uso Mostrar remitidos de un municipio.

CU # 29 Mostrar remitidos por municipio	
Actor	<i>usuario(inicia)</i>
Propósito	<i>Mostrar la cantidad de remitidos por municipio.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú Información útil, luego debe seleccionar “Mostrar remitidos por municipios”, luego debe elegir el municipio en que desee buscar los pacientes, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>	
Referencias	<i>RF-61</i>
Prototipo	Ver Anexo 36

Anexo 4.11 Descripción del caso de uso Mostrar remitidos por municipio.

CU # 30 Mostrar el por ciento anual pacientes dado un grupo	
Actor	<i>usuario(inicia)</i>
Propósito	<i>Mostrar el por ciento anual pacientes dado un grupo.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú porcientos (%), luego debe seleccionar “Mostrar el por ciento anual pacientes dado un grupo”, luego debe elegir el año y el grupo de enfermedades al cual quiera calcular el por ciento, en caso de existir pacientes en ese y pertenecientes a ese grupo el sistema calcula el por ciento y lo muestra, en caso contrario muestra un mensaje comunicando que no hay pacientes en ese grupo.</i>	
Referencias	<i>RF-62</i>
Prototipo	Ver Anexo 37

Anexo 4.12 Descripción del caso de uso Mostrar el por ciento anual pacientes dado un grupo.

CU # 31 Mostrar casos dada una enfermedad	
Actor	<i>usuario(inicia)</i>
Propósito	<i>Mostrar los pacientes dada una enfermedad</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú Información útil, luego debe seleccionar “Mostrar casos dada una enfermedad”, luego debe elegir la enfermedad, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>	
Referencias	<i>RF-63</i>
Prototipo	Ver Anexo 38

Anexo 4.13 Descripción del caso de uso Mostrar casos dada una enfermedad.

CU # 32 Mostar casos de un grupo de enfermedades en un intervalo de fecha, de un municipio y un grupo de edad determinado.	
Actor	<i>usuario(inicia)</i>
Propósito	<i>Mostar casos con sus datos pertenecientes a un grupo de enfermedades en un intervalo de fecha, de un municipio y un grupo de edad determinado.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú Información útil, luego debe seleccionar “Mostar casos de un grupo de enfermedades en un intervalo de fecha, de un municipio y un grupo de edad determinado”, luego debe elegir el grupo de enfermedades, el periodo, el municipio, el grupo de edad y el año en que desee ver los pacientes, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>	
Referencias	<i>RF-64</i>
Prototipo	Ver Anexo 39

Anexo 4.14 Mostar casos de un grupo de enfermedades en un intervalo de Fecha, de un municipio y un grupo de edad determinado.

CU # 31		Mostrar cantidad de operaciones por clase de operación
Actor	<i>usuario(inicia)</i>	
Propósito	<i>Mostrar la cantidad de operaciones por clase de operación.</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú Información útil, luego debe seleccionar “Mostrar cantidad de operaciones por clase de operación”, luego debe elegir el año en que desee ver la cantidad de pacientes por clase, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>		
Referencias	<i>RF-65</i>	
Prototipo	Ver Anexo 40	

Anexo 4.15 Descripción del caso de uso Mostrar cantidad de operaciones por clase de operación.

CU # 33		Mostrar modelo de cirugía
Actor	<i>usuario(inicia)</i>	
Propósito	<i>Mostrar los pacientes operados contenidos en un modelo predefinido.</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú modelos, luego debe seleccionar “Mostrar modelo de cirugía”, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>		
Referencias	<i>RF-66</i>	
Prototipo	Ver Anexo 41	

Anexo 4.16 Descripción del caso de uso Mostrar modelo de cirugía.

CU # 34		Mostrar el porcentaje anual de operaciones de una Clase
Actor	<i>usuario(inicia)</i>	
Propósito	<i>Mostrar el porcentaje anual de operaciones de una Clase.</i>	
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú</i>		

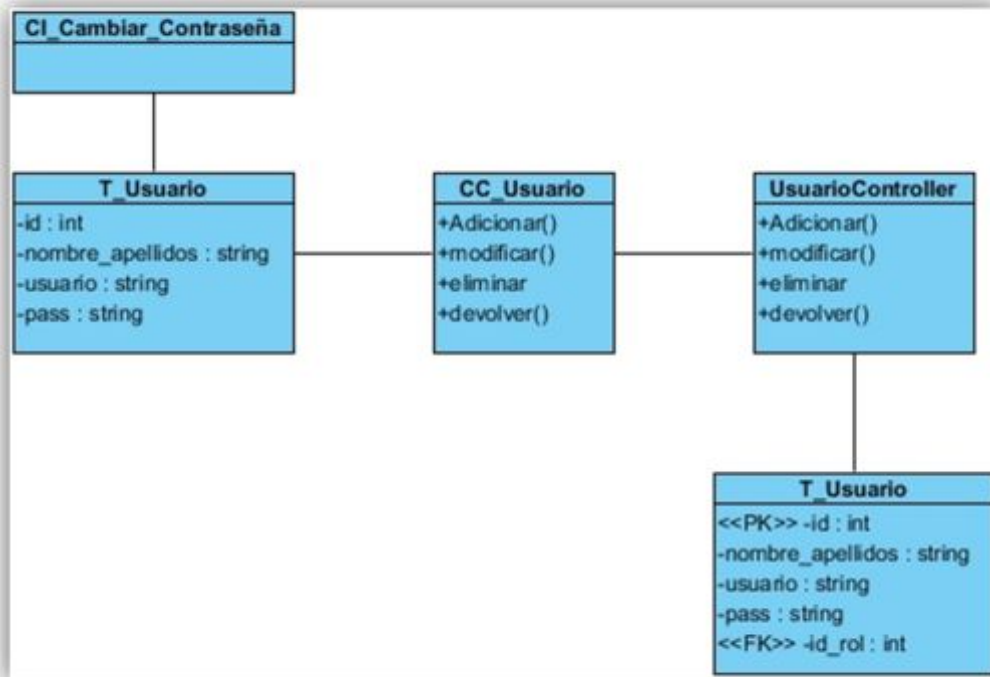
<i>porcientos (%), luego debe seleccionar “Mostrar el porciento anual de operaciones de una Clase”, luego debe elegir el año y la clase a la cual quiere calcular el porciento, en caso de existir pacientes en ese año y pertenecientes a esa clase el sistema calcula el porciento y lo muestra, en caso contrario muestra un mensaje comunicando que no hay pacientes con esa clase de operación.</i>	
Referencias	<i>RF-67</i>
Prototipo	Ver Anexo 42

Anexo 4.17 Descripción del caso de uso Mostrar el porciento anual de operaciones de una Clase.

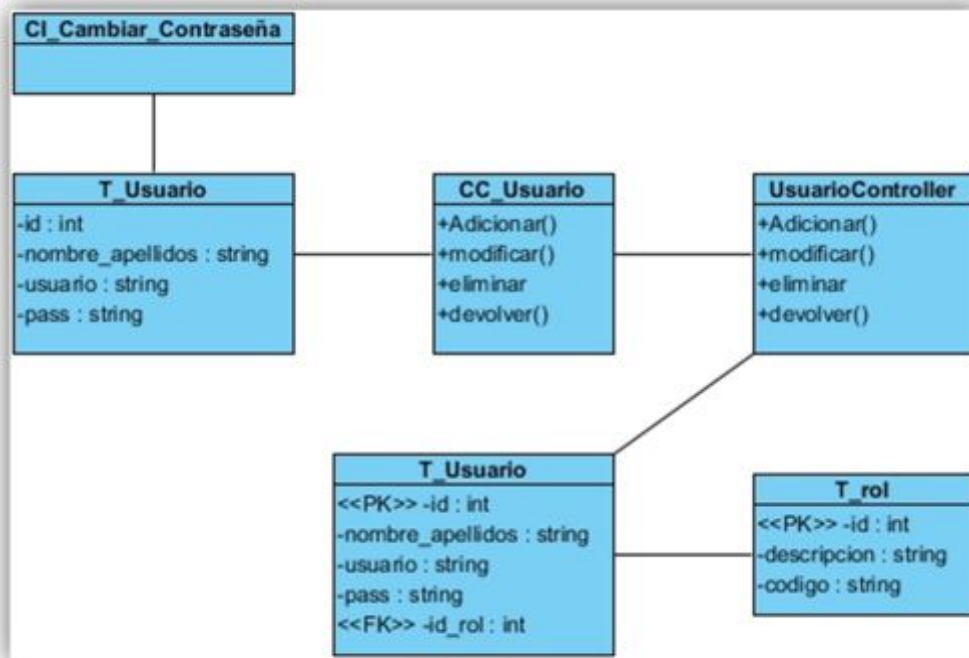
CU # 35	Mostrar todos los pacientes operados en un día determinado
Actor	<i>usuario(inicia)</i>
Propósito	<i>Mostrar todos los pacientes operados en un día determinado.</i>
Resumen: <i>Comienza cuando el usuario accede dentro del menú reportes al submenú Información útil, luego debe seleccionar “Mostrar todos los pacientes operados en un día determinado”, luego debe elegir el día en que desee ver los pacientes operados, en caso de existir información el sistema la muestra, en caso contrario muestra un mensaje comunicando que no hay información.</i>	
Referencias	<i>RF-68</i>
Prototipo	Ver Anexo 43

Anexo 4.18 Descripción del caso de uso Mostrar el porciento anual de operaciones de una Clase.

Anexo 5 Diagramas de clases del diseño: Paquete Seguridad

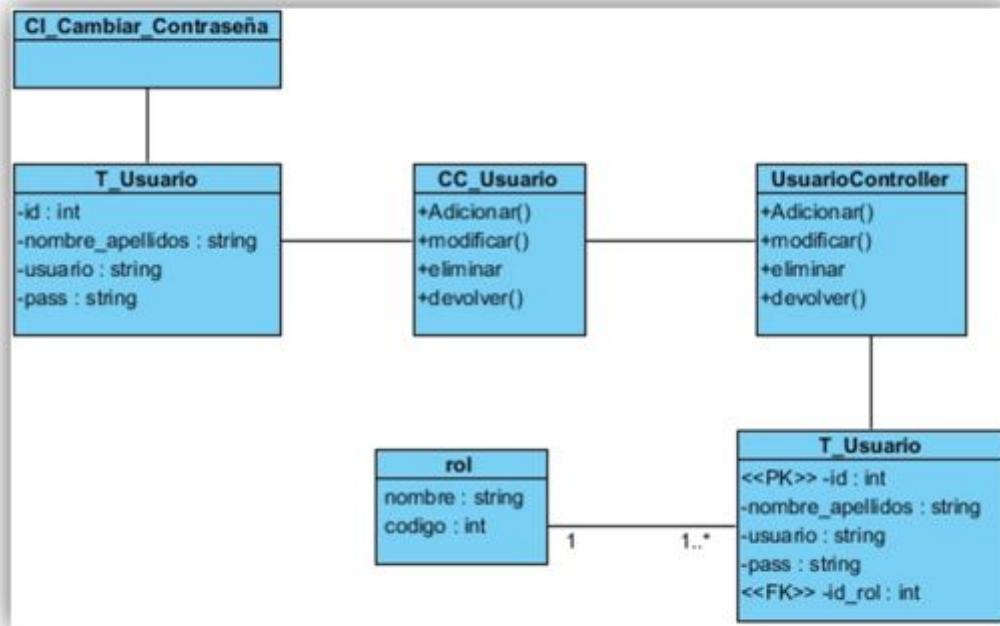


Anexo 5.1 Diagrama de clases del diseño. Caso de uso: Autenticar Usuario



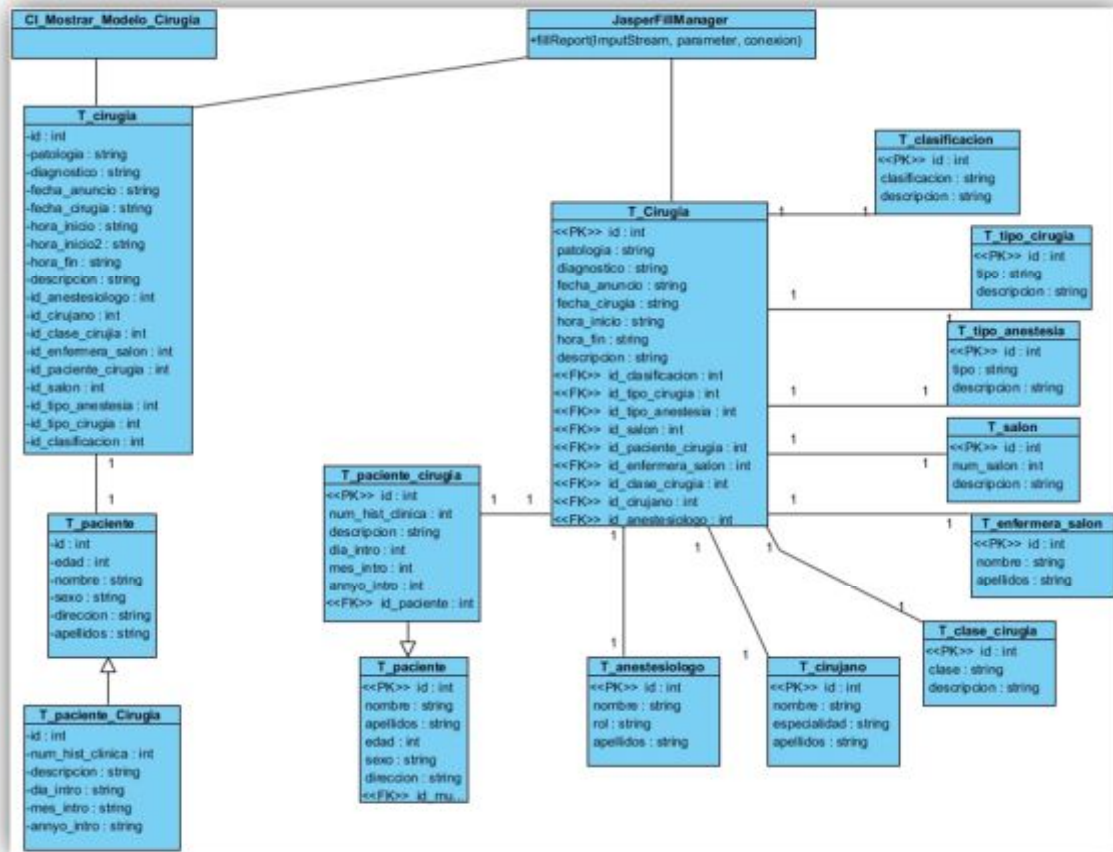
Anexo 5.2 Diagrama de clases del diseño. Caso de uso: Cambiar contraseña

Anexos 6 Diagramas de clases del diseño: Paquete Administración

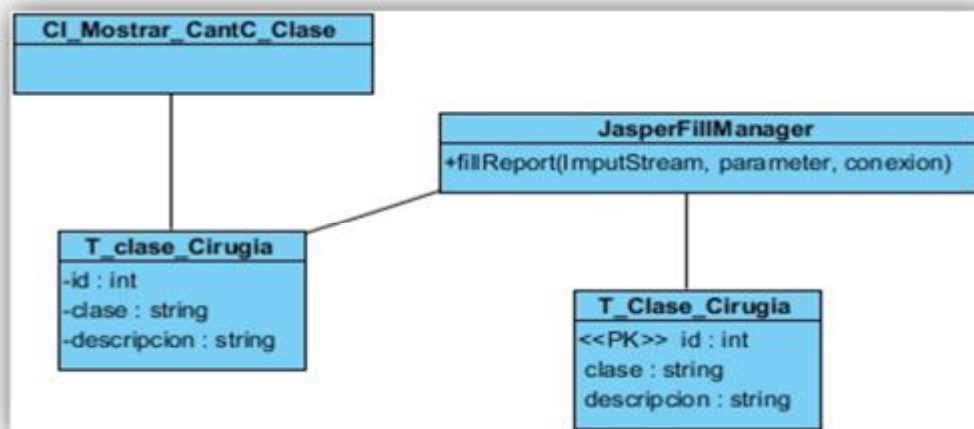


Anexo 6.1 Diagrama de clases del diseño. Caso de uso: Gestionar usuario

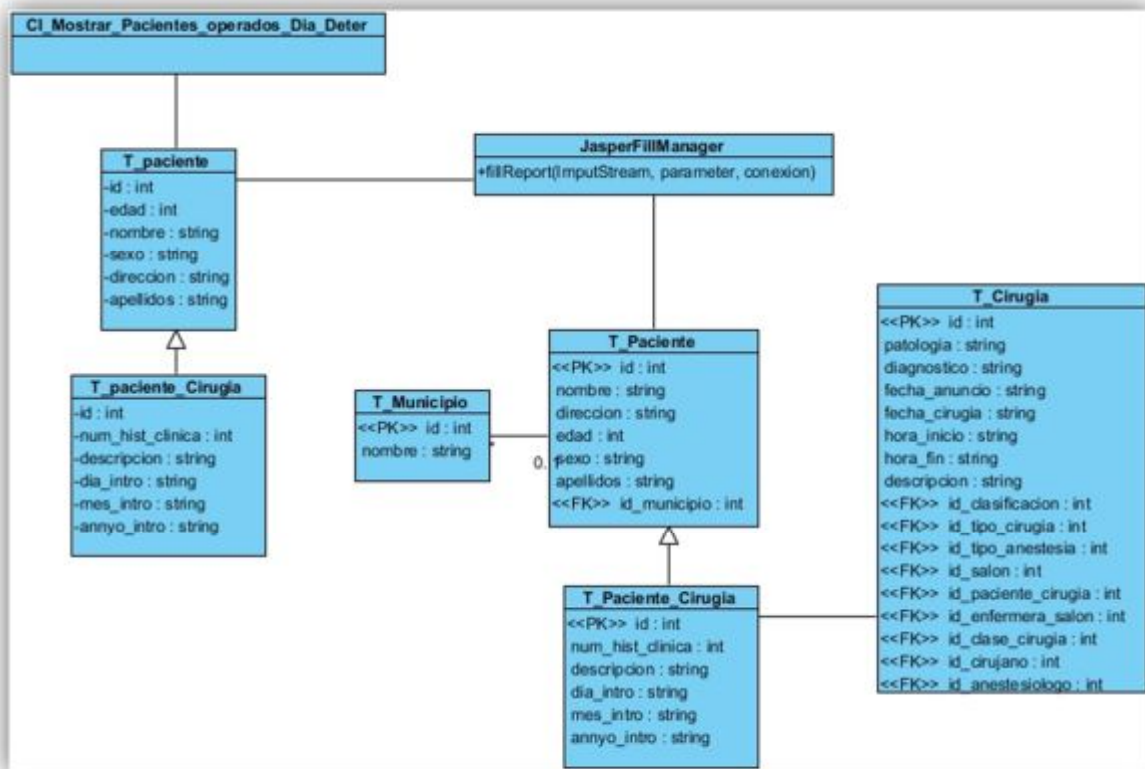
Anexos 7 Diagramas de clases del diseño: Paquete Reportes



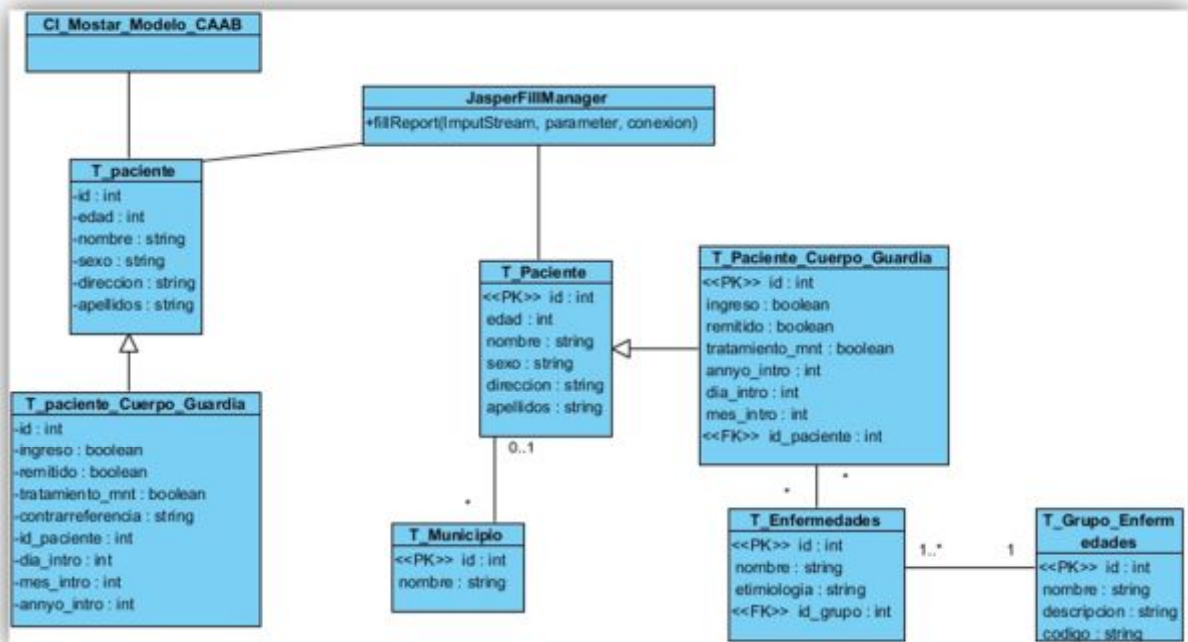
Anexo 7.1 Diagrama de clases del diseño. Caso de uso: Mostrar modelo de cirugía



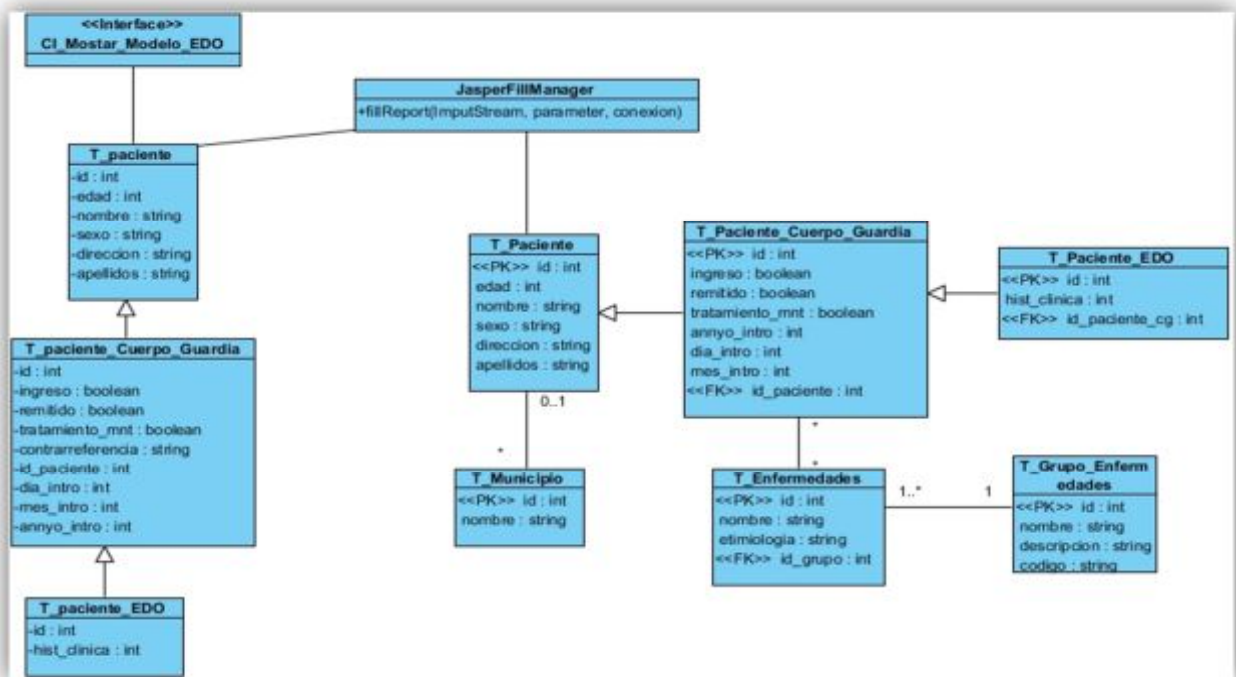
Anexo 7.2 Diagrama de clases del diseño. Caso de uso: Mostrar cantidad de operaciones por clase de operación



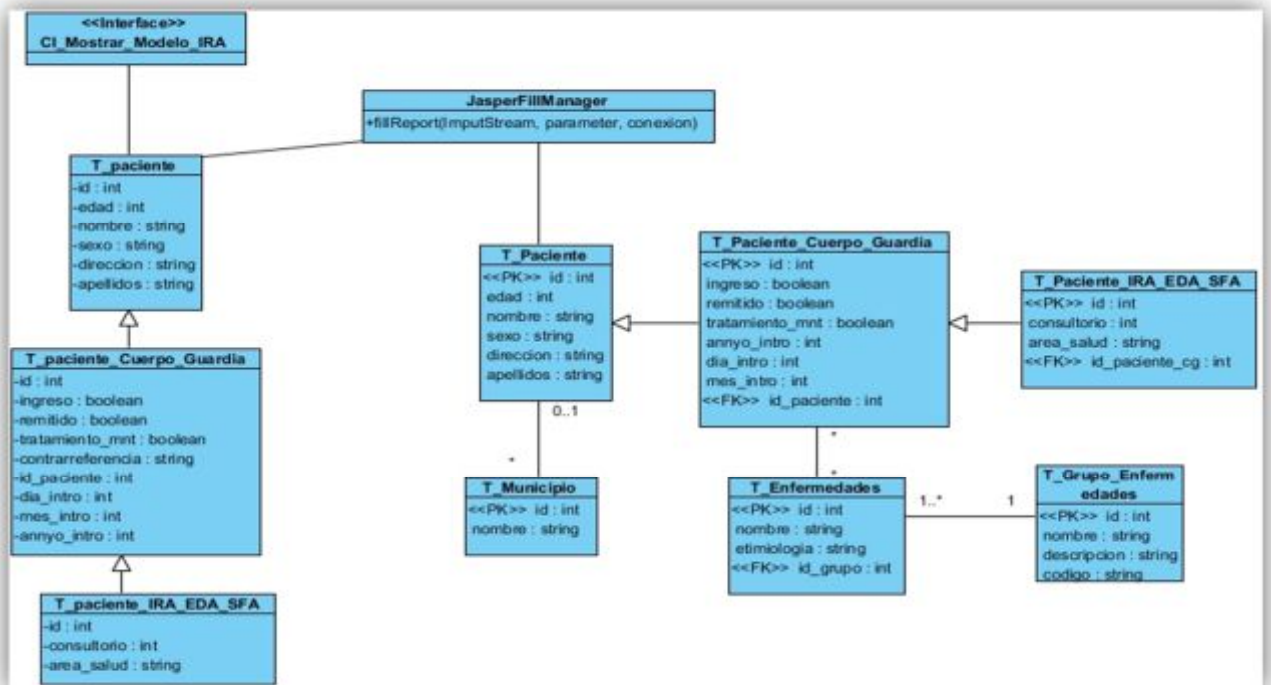
Anexo 7.3 Diagrama de clases del diseño. Caso de uso: Mostrar todos los pacientes operados en un día determinado



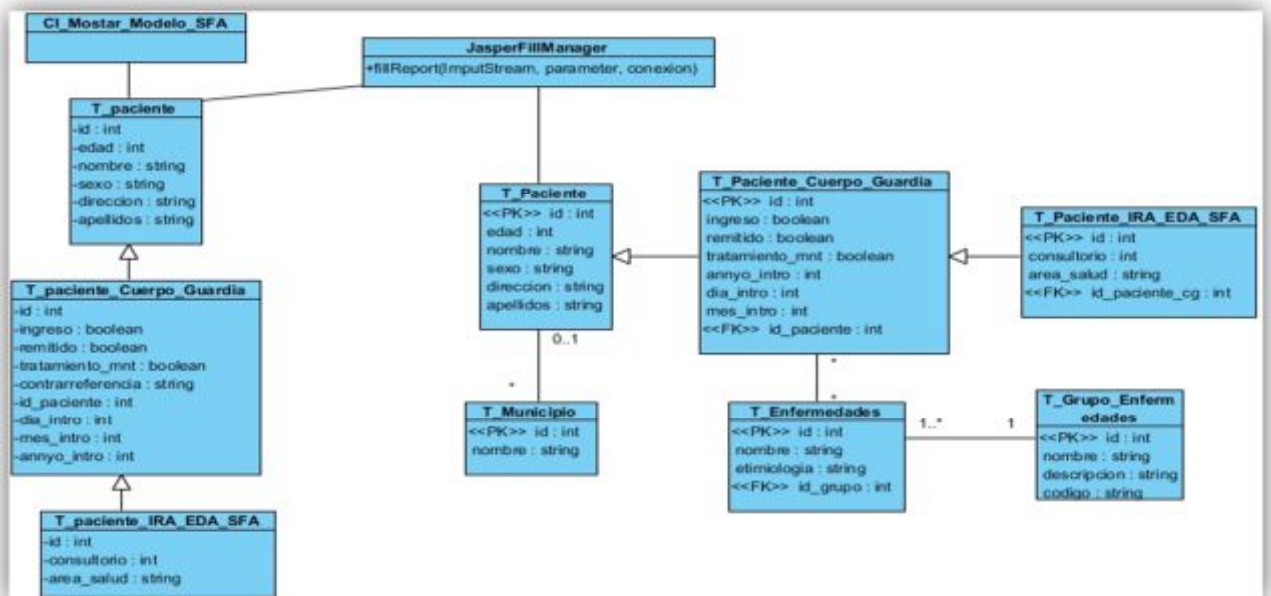
Anexo 7.4 Diagrama de clases del diseño. Caso de uso: Mostrar modelo de Crisis Aguda de Asma Bronquial.



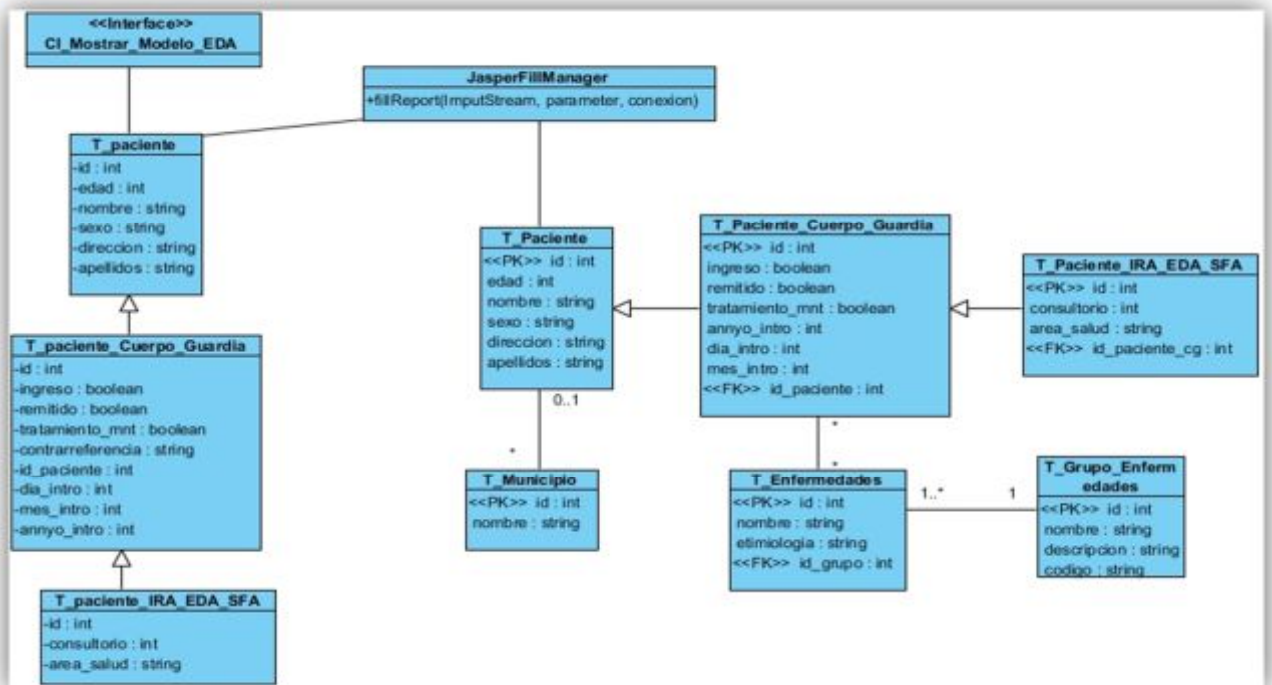
Anexo 7.5 Diagrama de clases del diseño. Caso de uso: Mostrar modelo de Enfermedades de Difusión Obligatoria.



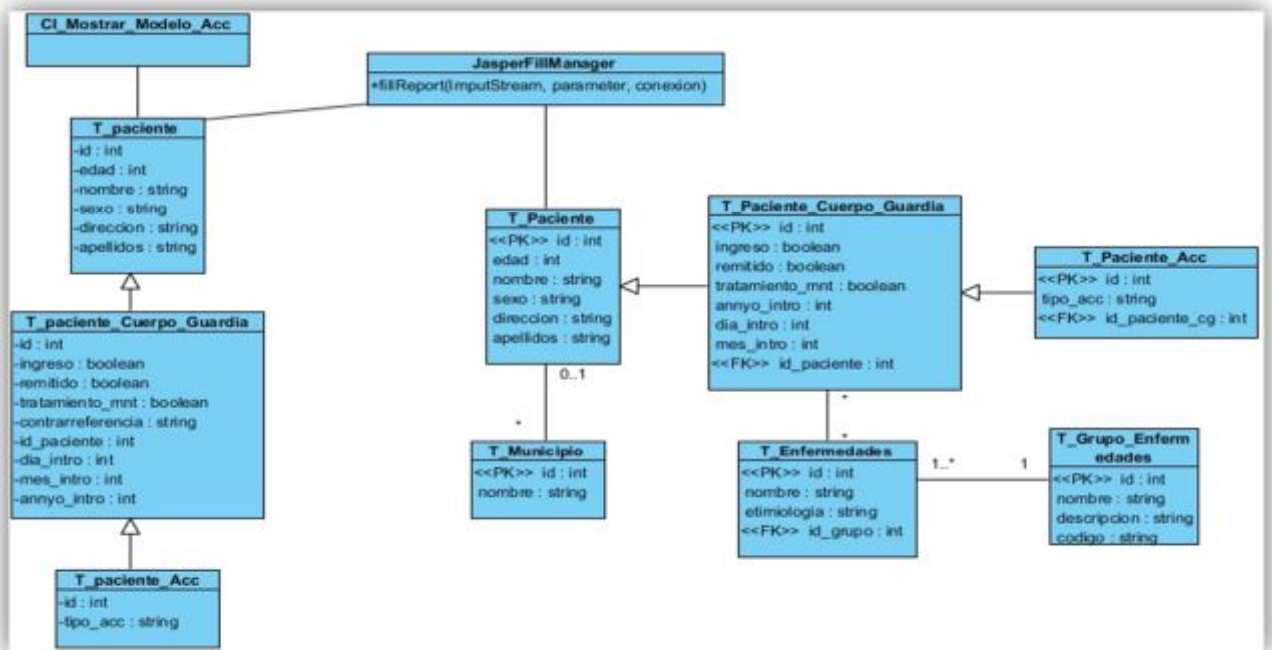
Anexo 7.6 Diagrama de clases del diseño. Caso de uso: Mostrar modelo de Infección Respiratoria Aguda.



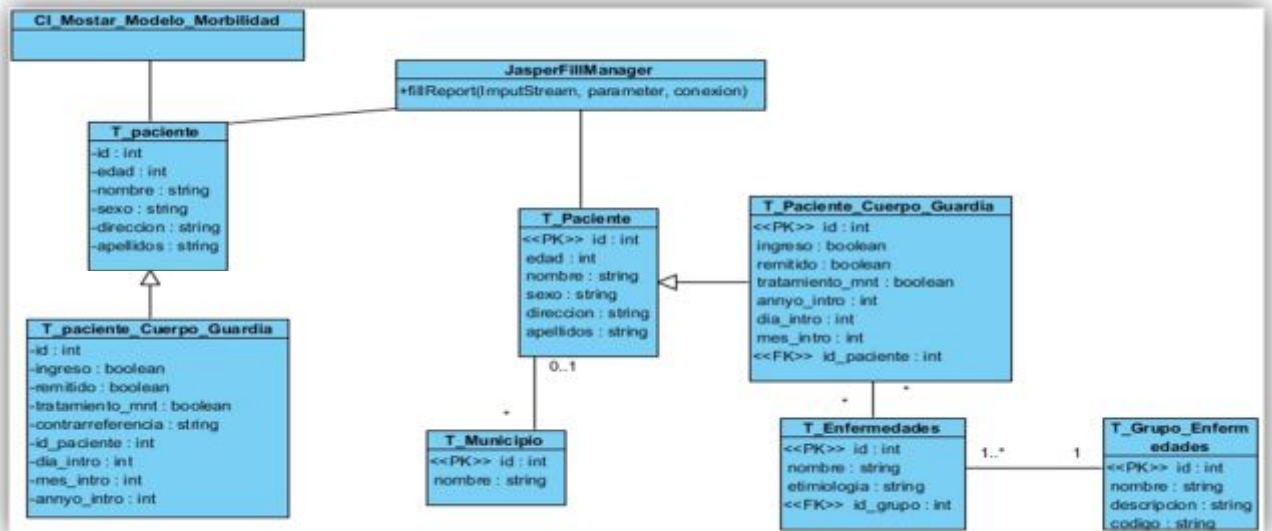
Anexo 7.7 Diagrama de clases del diseño. Caso de uso: Mostrar modelo de Síndrome Febril Agudo.



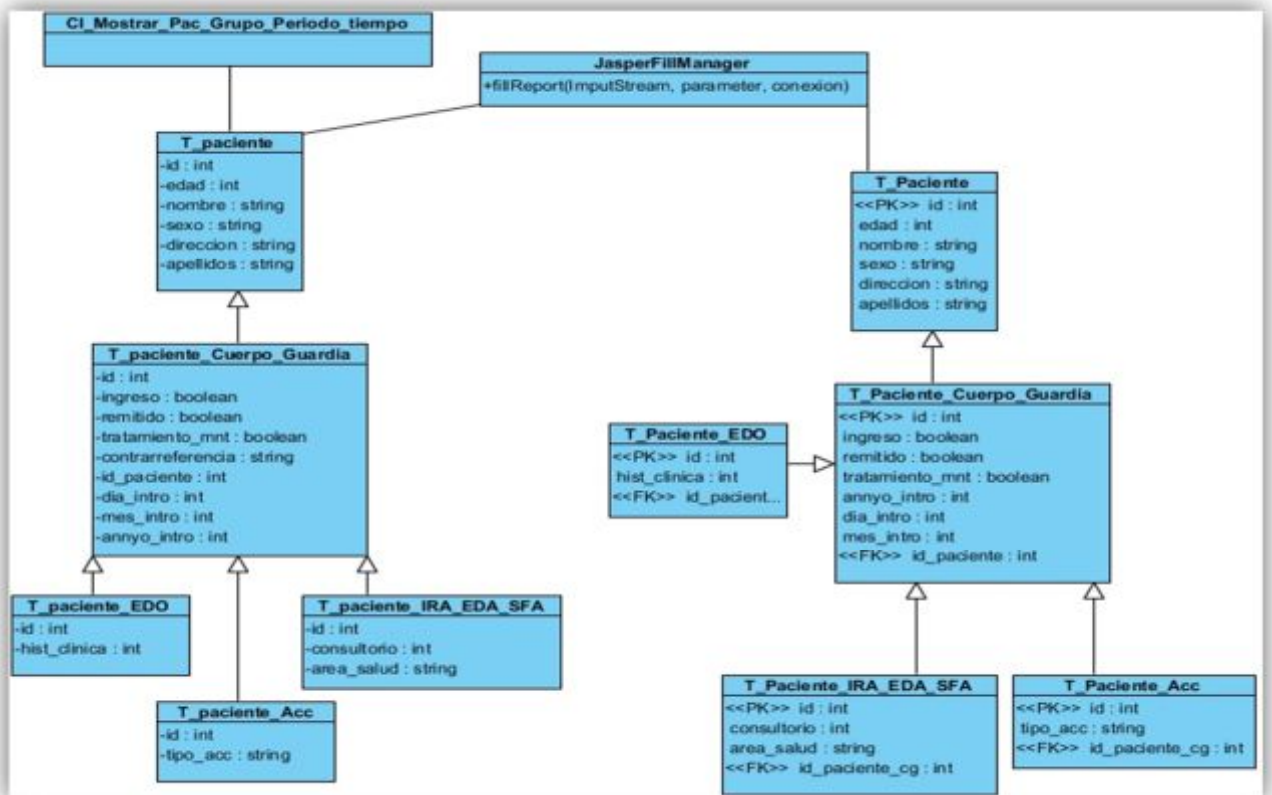
Anexo 7.8 Diagrama de clases del diseño. Caso de uso: Mostrar modelo de Enfermedad Diarreica Aguda.



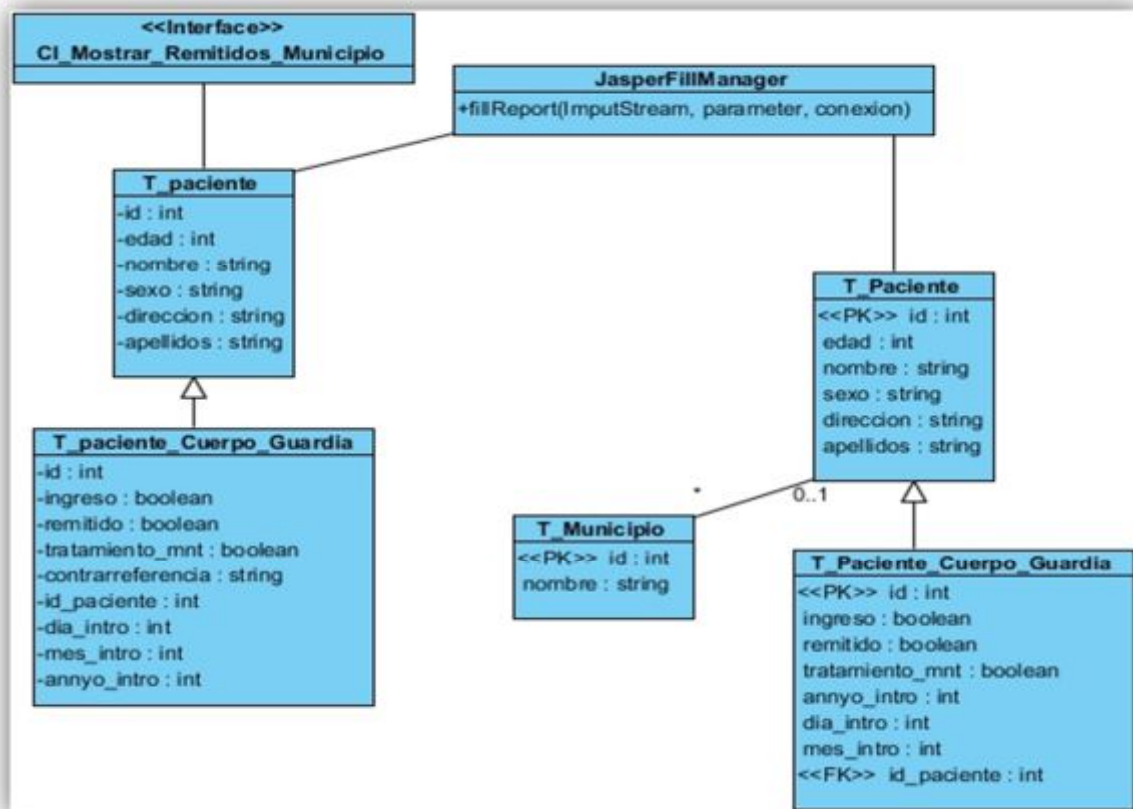
Anexo 7.9 Diagrama de clases del diseño. Caso de uso: Mostrar modelo de Accidentes



Anexo 7.10 Diagrama de clases del diseño. Caso de uso: Mostrar modelo de Morbilidad

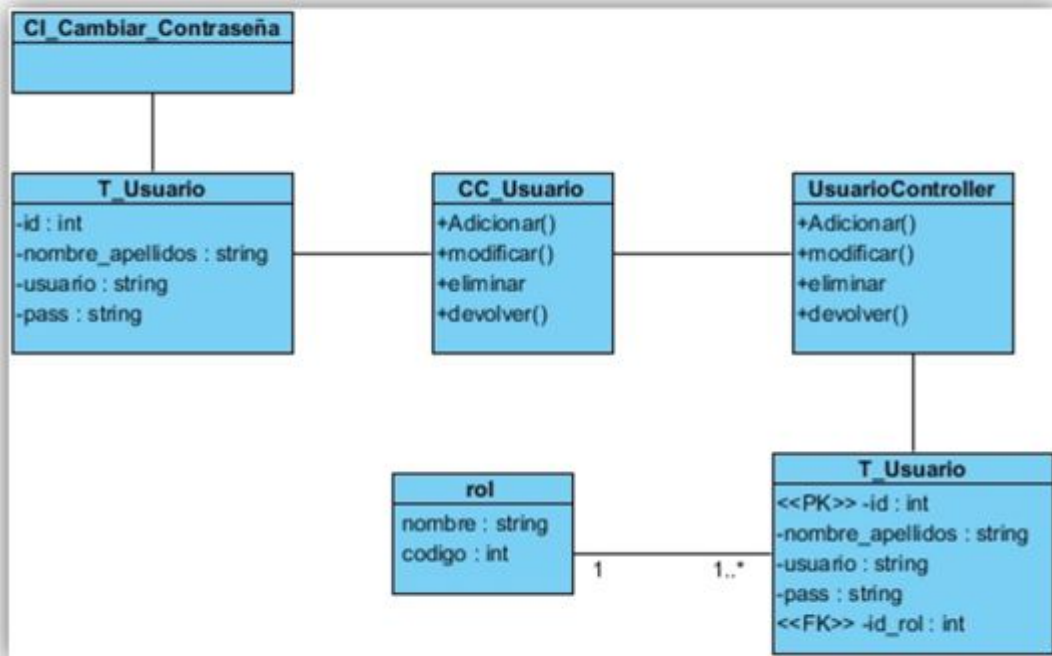


Anexo 7.11 Diagrama de clases del diseño. Caso de uso: Mostrar casos vistos de un grupo dado en un periodo de tiempo dado.

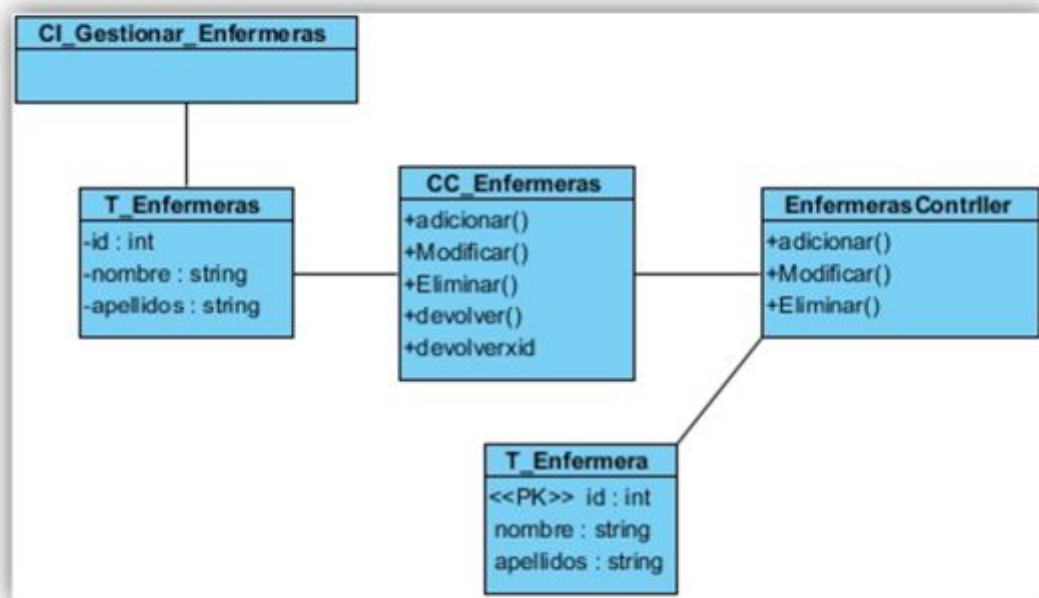


Anexo 7.12 Diagrama de clases del diseño. Caso de uso: Mostrar remitidos de un municipio.

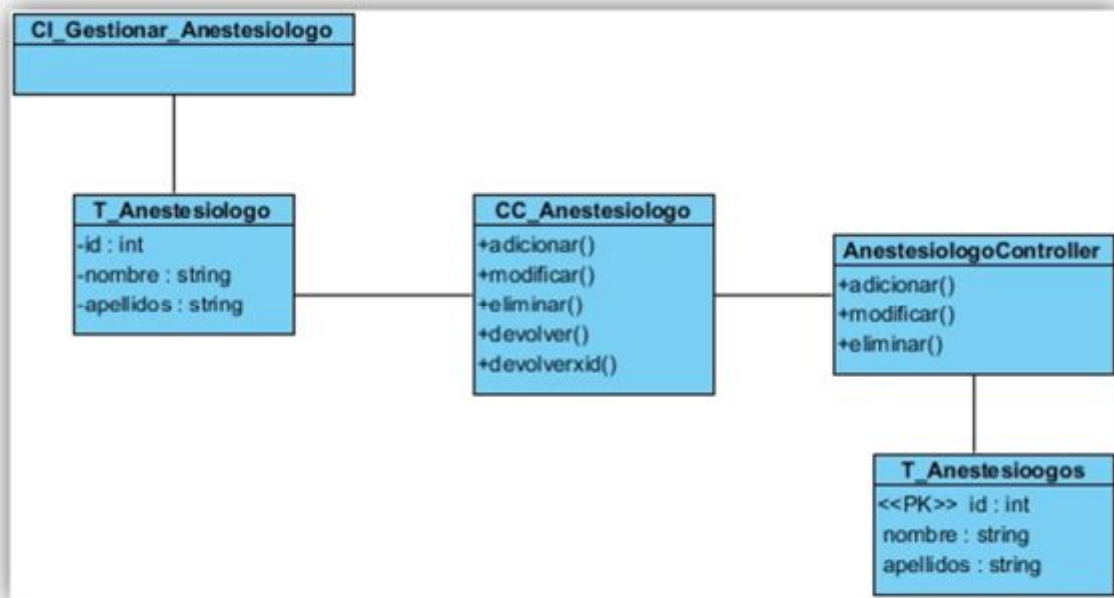
Anexos 8 Diagramas de clases del diseño: Paquete Gestión



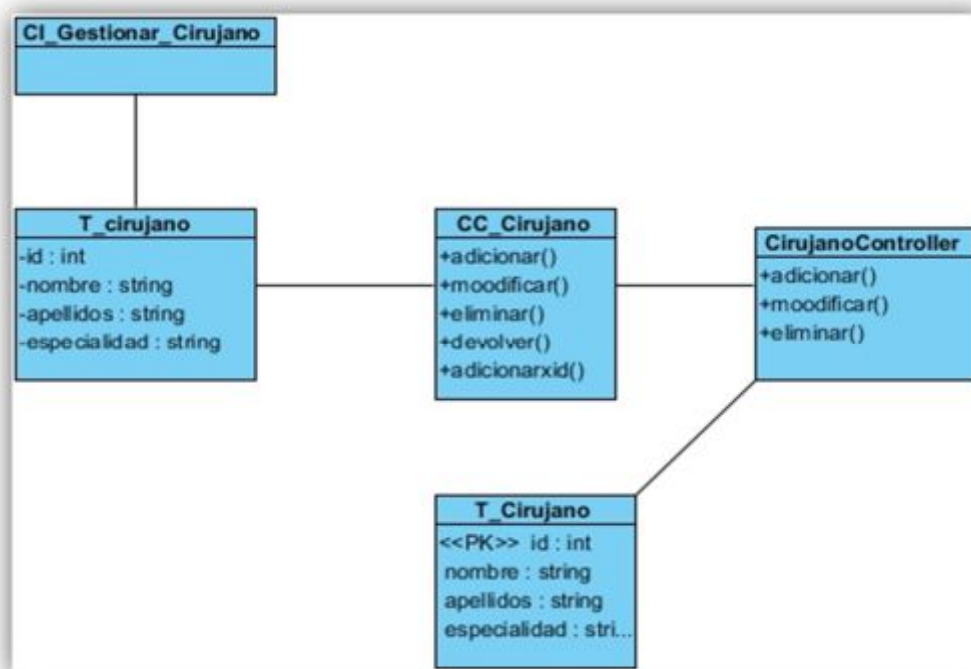
Anexo 8.1 Diagrama de clases del diseño. Caso de uso: Gestionar Usuario



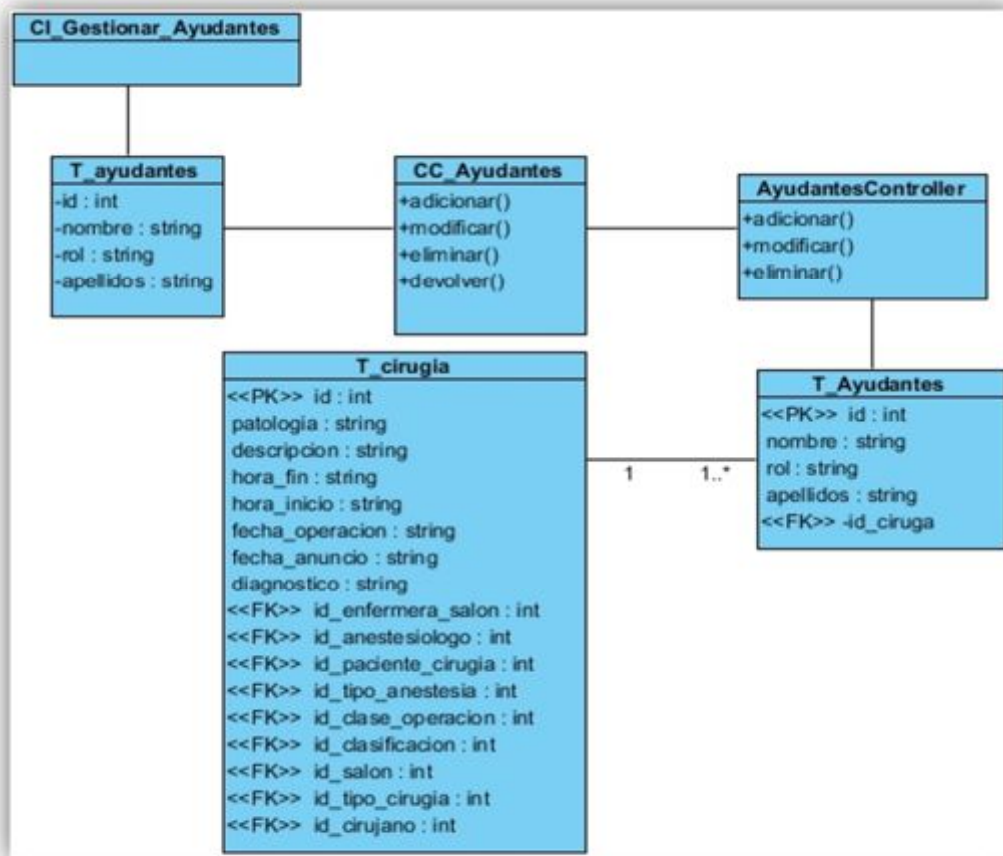
Anexo 8.2 Diagrama de clases del diseño. Caso de uso: Gestionar enfermeras de salón



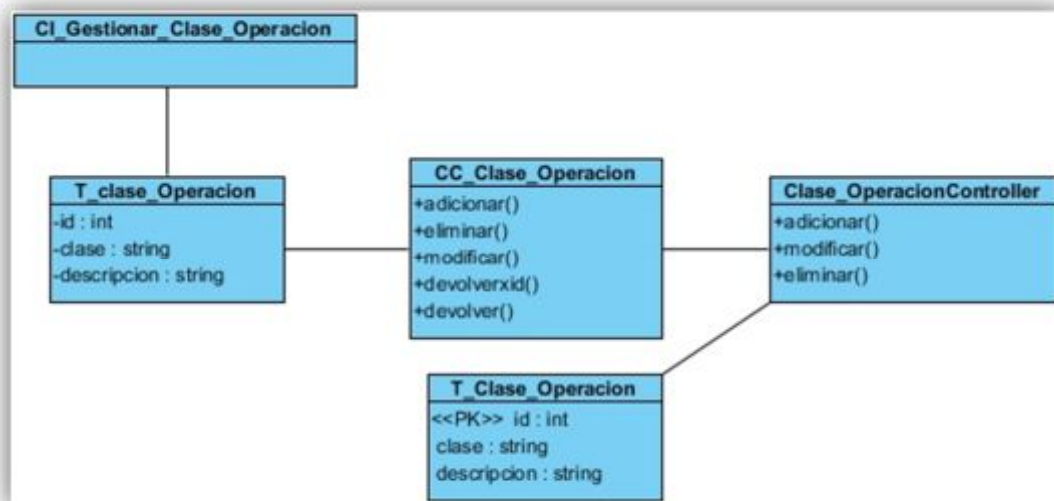
Anexo 8.3 Diagrama de clases del diseño. Caso de uso: Gestionar anestesiólogo



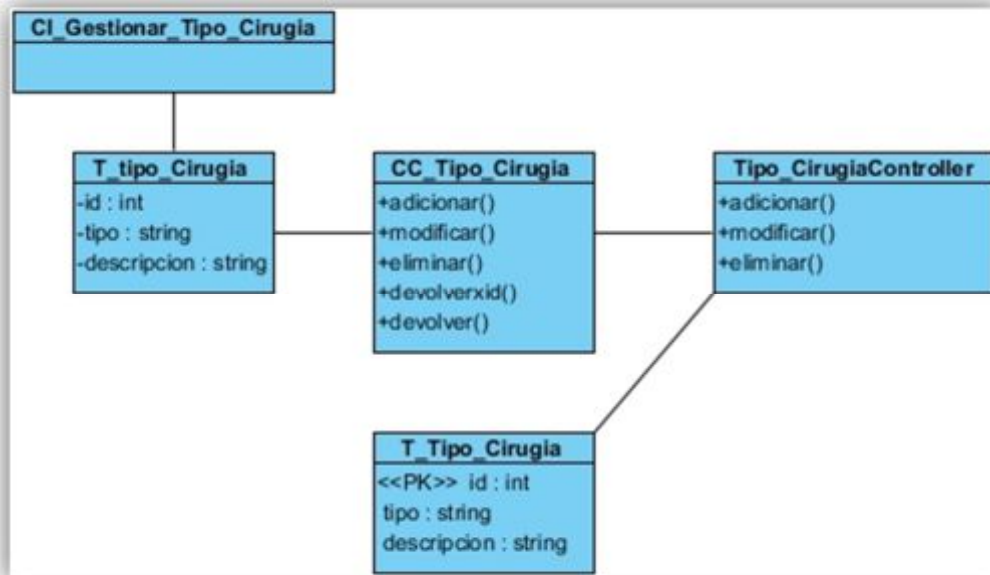
Anexo 8.4 Diagrama de clases del diseño. Caso de uso: Gestionar cirujano



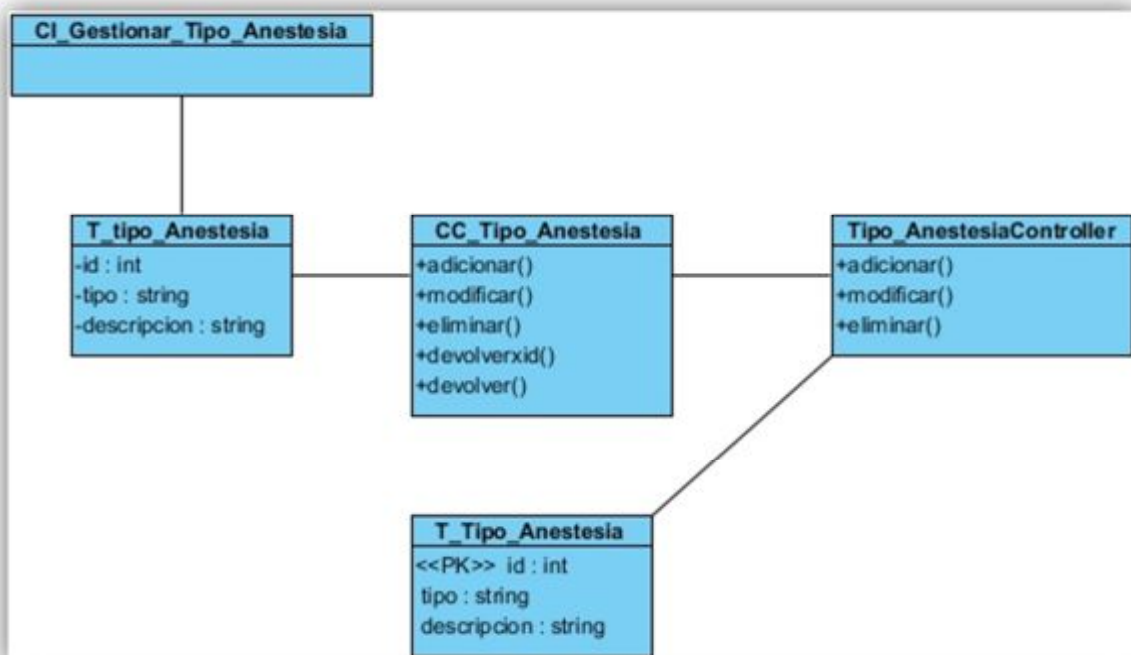
Anexo 8.5 Diagrama de clases del diseño. Caso de uso: Gestionar ayudantes



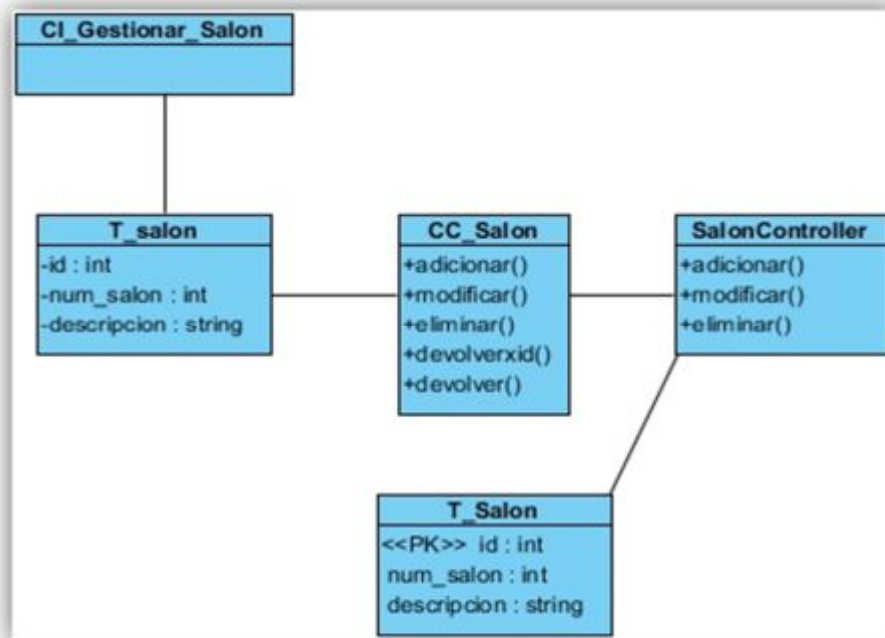
Anexo 8.6 Diagrama de clases del diseño. Caso de uso: Gestionar clase de cirugía



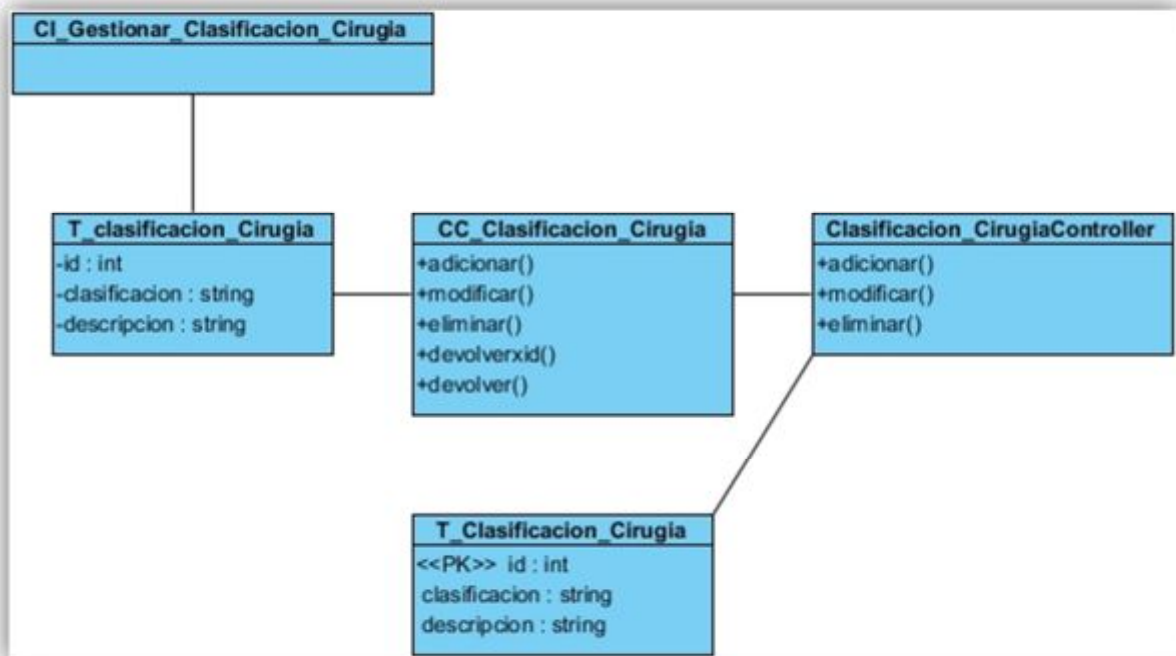
Anexo 8.7 Diagrama de clases del diseño. Caso de uso: Gestionar tipo de cirugía



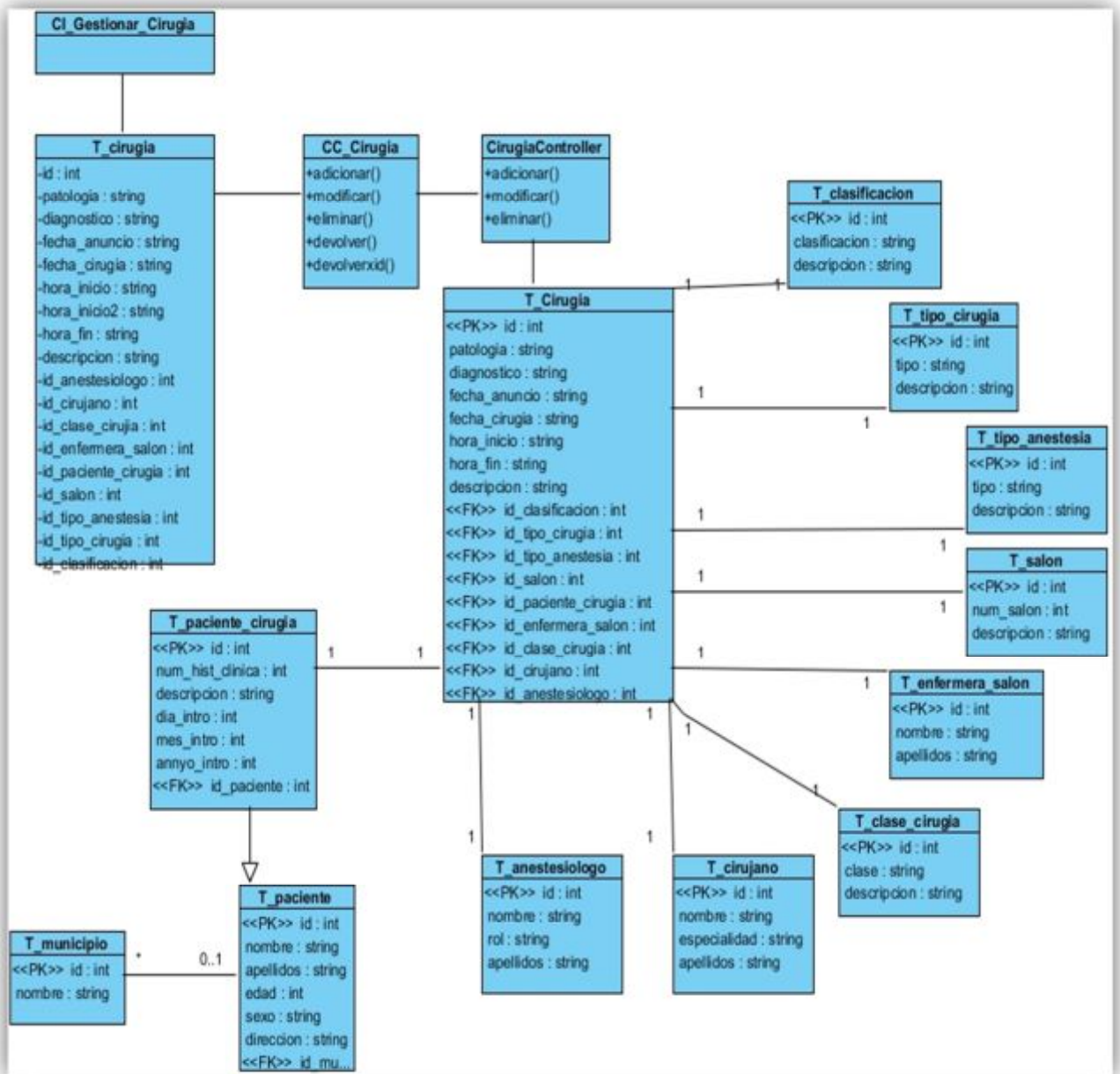
Anexo 8.8 Diagrama de clases del diseño. Caso de uso: Gestionar tipo de anestesia



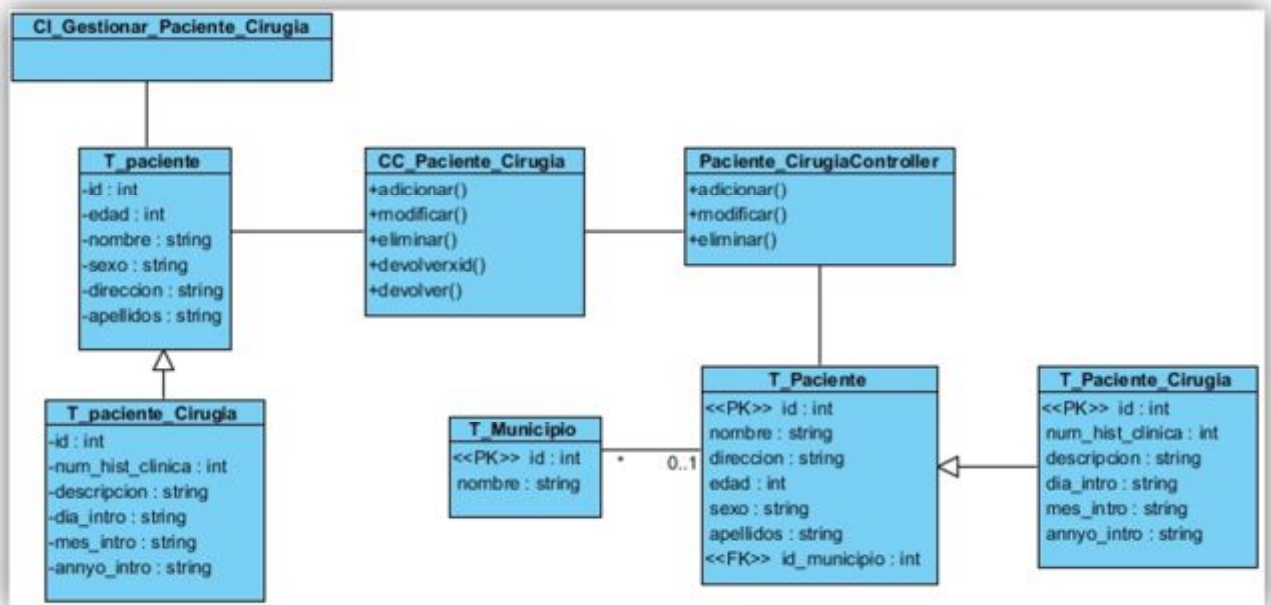
Anexo 8.9 Diagrama de clases del diseño. Caso de uso: Gestionar salón



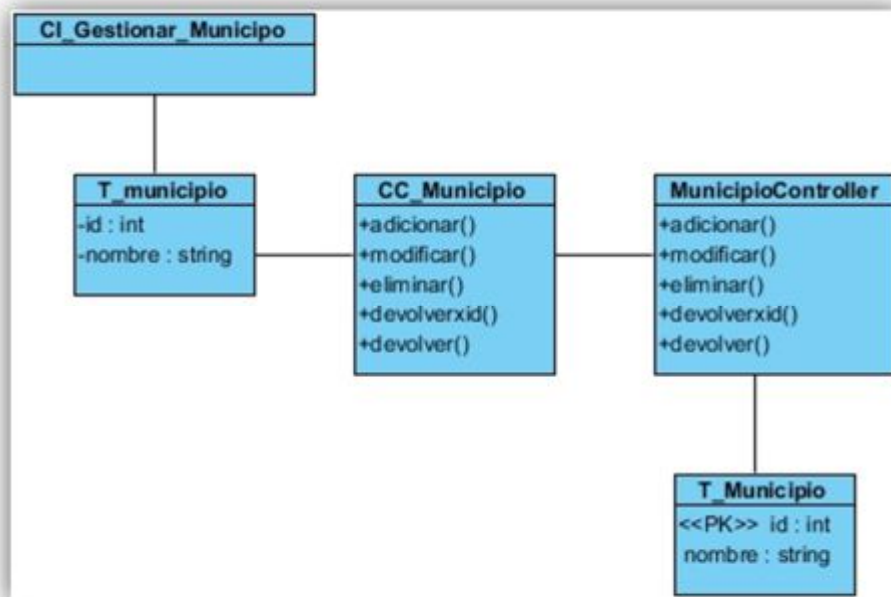
Anexo 8.10 Diagrama de clases del diseño. Caso de uso: Gestionar clasificación de cirugía



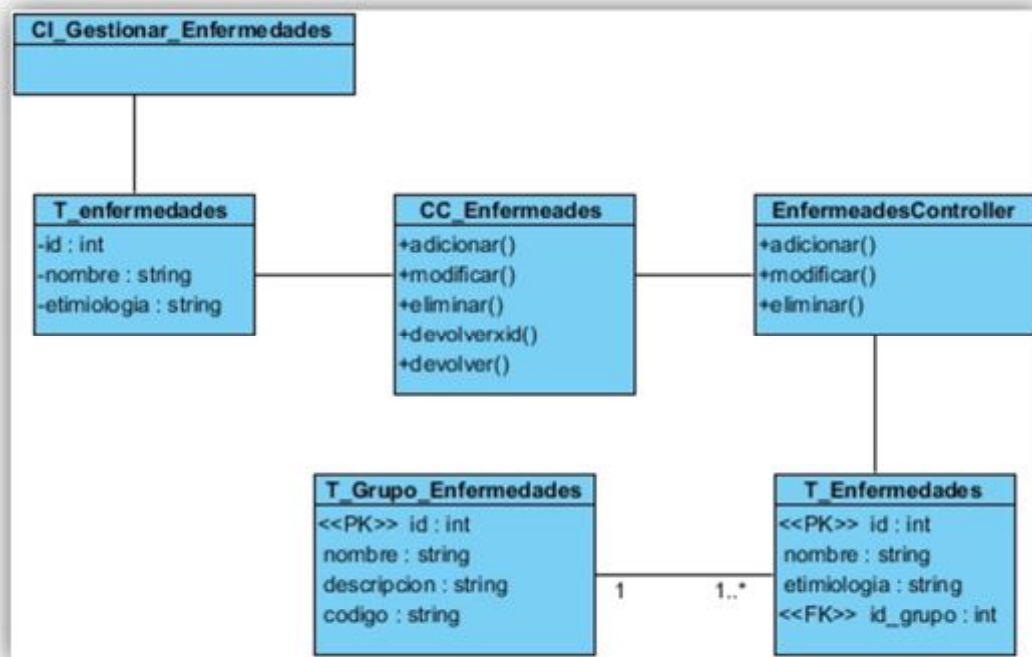
Anexo 8.11 Diagrama de clases del diseño. Caso de uso: Gestionar cirugía



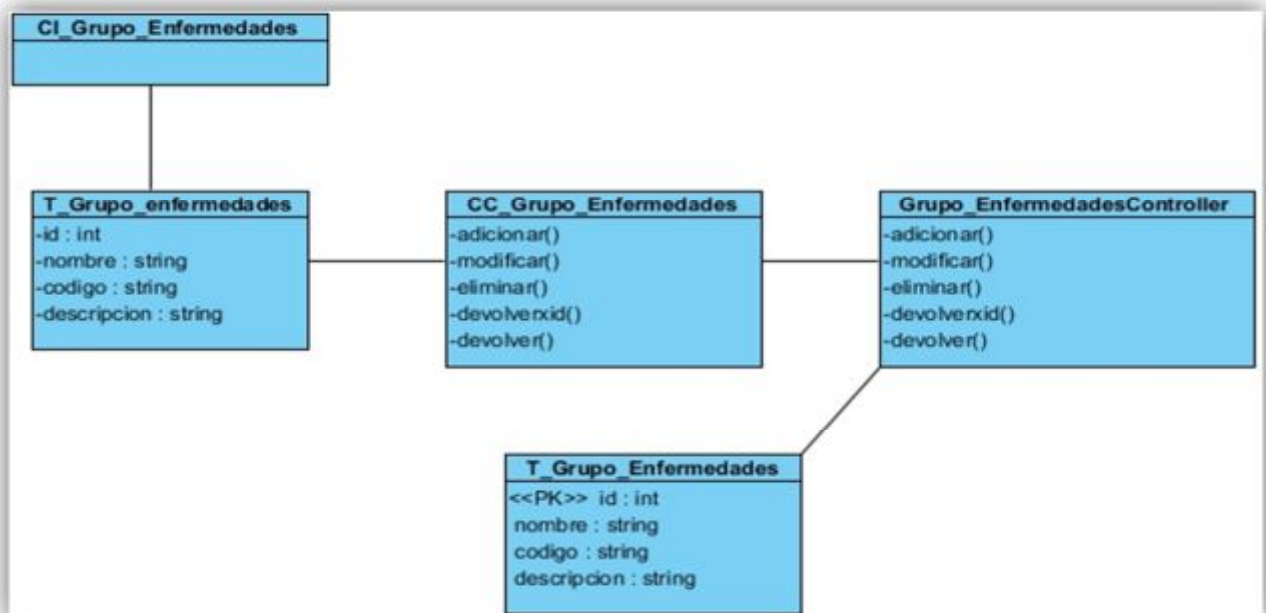
Anexo 8.12 Diagrama de clases del diseño. Caso de uso: Gestionar paciente de cirugía



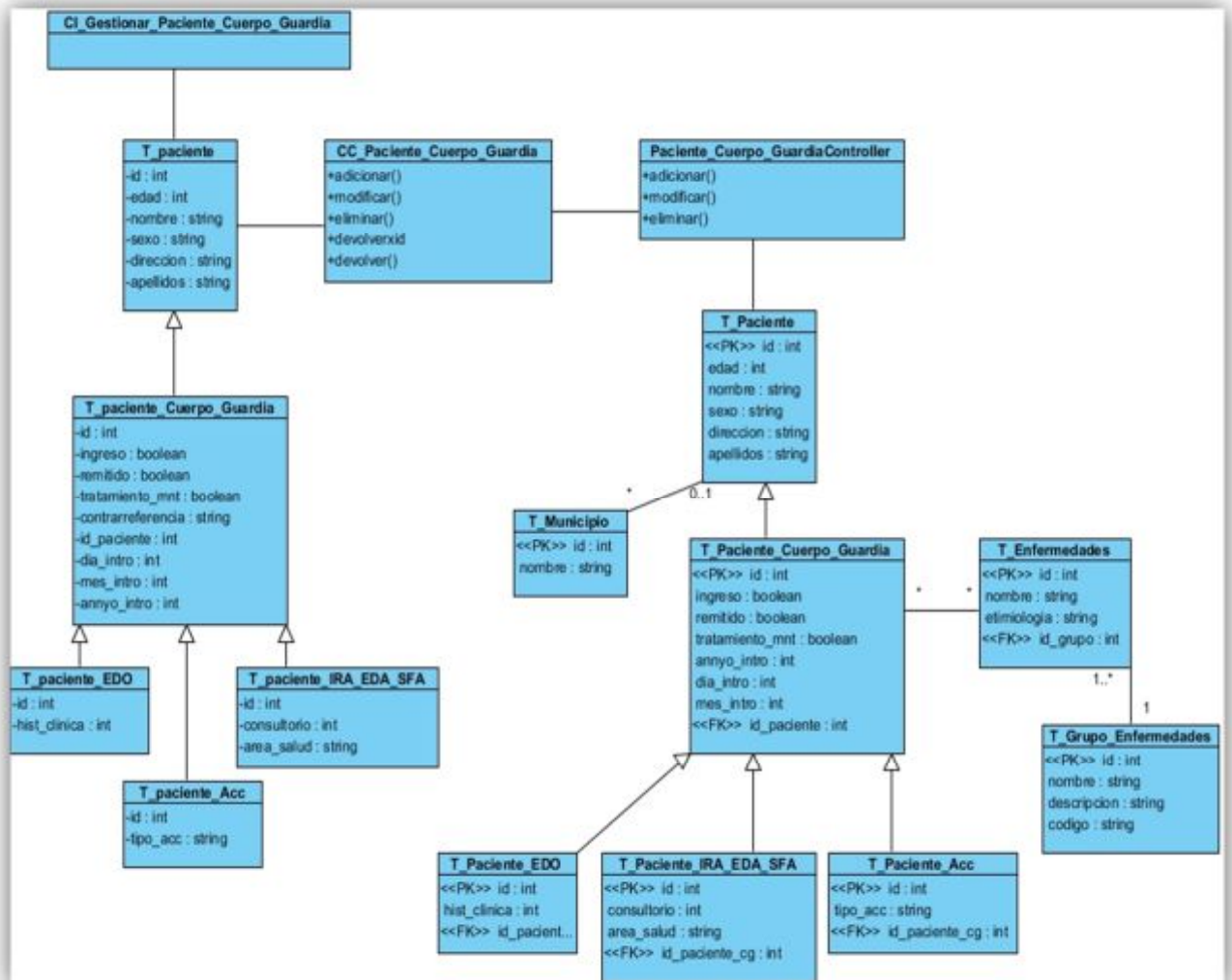
Anexo 8.13 Diagrama de clases del diseño. Caso de uso: Gestionar municipio



Anexo 8.14 Diagrama de clases del diseño. Caso de uso: Gestionar enfermedades



Anexo 8.15 Diagrama de clases del diseño. Caso de uso: Gestionar grupo de enfermedades



Anexo 8.16 Diagrama de clases del diseño. Caso de uso: Gestionar paciente de cuerpo de guardia

The screenshot shows a dialog box titled "Autenticación" with a close button in the top right corner. The main heading is "Autenticación" in green, accompanied by a yellow padlock icon. Below the heading, there are two input fields: "Usuario:" with the text "carlos" and "Contraseña:" with seven black dots. At the bottom, there are two buttons: "Aceptar" (highlighted in blue) and "Cancelar".

Anexo 9 Prototipo del caso de uso: Autenticar Usuario

The screenshot shows a dialog box titled "Cambiar Contraseña" with a close button in the top right corner. The main heading is "Cambiar Contraseña" in green, accompanied by a yellow padlock icon. Below the heading, there are four input fields: "Usuario:" with the text "carlos", "Contraseña Anterior:" with seven black dots, "Nueva Contraseña:" with seven black dots, and "Verificar la nueva Contraseña:" with seven black dots. At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Anexo 10 Prototipo del caso de uso: Cambiar contraseña

Gestionar Usuario

Usuario

Nombre y Apellidos:

Nombre de Usuario:

Contraseña:

Rol:

Nombre	Usuario	Rol
Julia Rosa Rodriguez	jular	Técnico de Cirugía
Anisley Annalie Rodri...	anisleya	Técnico de Cuerpo de...
Carlos Arturo Barberá...	carlos	Administrador

Anexo 11 Prototipo del caso de uso: Gestionar usuario

Paciente Cirugía Hoy: 2015/06/08

Nombre:

Historia Clínica:

Apellidos:

Descripción:

Edad:

Sexo:

Dirección:

Municipio:

Nro. HC	Nombre	Apellidos	Edad	Sexo	Dirección	Municipio
823	Rosa	Flores	12	Femenino	as	Yaguajay
25	Cristianoooo	Ronal	30	Masculino	Portugal	Yaguajay
26	Sergio manuel	perez	14	Masculino	calle del me...	Sancti Spiritus
122	Manuela	Suarez ctfj	1	Femenino	Ave del Par...	Yaguajay

Anexo 12 Prototipo del caso de uso: Gestionar Paciente de cirugía

Gestionar Cirugía

Cirugía

Paciente: Cirujano:

Patología:

Diagnóstico:

Operatorio:

Fecha Anuncio: Fecha Operación:

Tipo Cirugía:

Salón:

Clasificación:

Tipo Operación:

Tipo Anestesia:

Anestesiologo:

Enfermera de Salón:

Hora inicio: Hora final:

Nombre	Patología	Cirujano	Fecha Operación	Salón	hora inicio	hora final
Cristianoooo	afasdjf	Olivia	2016/05/05	1	12:33	14:55
Sergio manuel	sdf	Carlos Manuel	2016/04/08	2	23:43	12:33
Manuela	higado	Carlos Manuel	2015/04/30	2	01:54	04:55
Manuela	dndn	Carlos Manuel	2015/04/03	1	12:33	14:22
Manuela	xfhgf	Olivia	2016/05/19	2	14:33	15:14
Manuela	Bilis	Carlos Manuel	2015/04/24	2	04:11	02:58

Anexo 13 Prototipo del caso de uso: Gestionar cirugía

Gestionar Clasificación de la Cirugía

Clasificación de la Cirugía

Nombre de la Clasificación:

Descripción:

Nombre de la Clasificación	Descripción
Limpia	
Limpia Contaminada	
Contaminada	
Sucia	

Anexo 14 Prototipo del caso de uso: Gestionar clasificación de la cirugía

Gestionar Salón

Salón

Salón (Número): 7

Descripción: Es nuevo

Numero del Salón	Descripción
2	Es grande
1	Es mas Pequeño

Icons: +, edit, undo, delete, left arrow, right arrow

Anexo 15 Prototipo del caso de uso: Gestionar salón

Gestionar Tipo de Anestesia

Tipo de Anestesia

Nombre del Tipo de Anestesia:

Descripción:

Nombre del Tipo de Anestesia	Descripción
Local	solo unos pinchazos
general	es total

Icons: +, edit, undo, delete, left arrow, right arrow

Anexo 16 Prototipo del caso de uso: Gestionar tipo de anestesia

The screenshot shows a window titled "Gestionar Tipo de Cirugía". It contains a form with the following fields:

- Nombre del Tipo de Cirugía:** A text input field.
- Descripción:** A larger text area for entering details.

Below the form is a table with the following data:

Nombre del Tipo de Cirugía	Descripción
Reintervención	Hay que Operar nuevamente
Mínimo Acceso	

At the bottom of the window are several icons: a green plus sign, a pencil, a yellow circular arrow, a red X, and two blue arrows pointing left and right.

Anexo 16 Prototipo del caso de uso: Gestionar tipo de cirugía

The screenshot shows a window titled "Gestionar Clase de Cirugía". It contains a form with the following fields:

- Nombre de la Clase:** A text input field.
- Descripción:** A larger text area for entering details.

Below the form is a table with the following data:

Nombre del Tipo de Operac...	Descripción
Mayor Urgente	Una cirugía Muy Grandey es ...
Mayor Electiva	Una cirugía grande pero ele...
Menor Electiva	Cirugía pequeña
Menor Urgente	

At the bottom of the window are several icons: a green plus sign, a pencil, a yellow circular arrow, a red X, and two blue arrows pointing left and right.

Anexo 18 Prototipo del caso de uso: Gestionar clase de cirugía



Gestionar Ayudantes

Ayudantes

Nombre: Armando

Apellidos: Casas Prieto

Rol: Estudiante

Operación: Salón: 2 Fecha: 2016/04/08 HI: 23:43 HF: 12:33

Nombre	Apellidos	Rol	Operación
Leidi	batista Bur	Tecnico	Salón: 2 Fecha: 20...
Lenar	Meneses	asistente	Salón: 2 Fecha: 20...

Anexo 19 Prototipo del caso de uso: Gestionar ayudantes



Gestionar Cirujano

Cirujano

Nombre: Carlos

Apellidos: Casas

Especialidad: Neuro

Nombre	Apellidos	Especialidad
Carlos Manuel	Ferrer	Cirujano Plastico
Olivia	Martínez González	Neurocirujana
Ignacio	Morgado Ferrer	Cardiocirujano

Anexo 20 Prototipo del caso de uso: Gestionar cirujano

Gestionar Anestesiólogo

Anestesiologo

Nombre:

Apellidos:

Nombre	Apellidos
Carlos	Lorenzo Valdivia
Ines	Flores Cabrera

Toolbar: +, ✎, ↶, ✖, ⏪, ⏩

Anexo 21 Prototipo del caso de uso: Gestionar Anestesiólogo

Gestionar Enfermeras de Salón

Enfermera de Salón

Nombre:

Apellidos:

Nombre	Apellidos
Ilana	Mayedo Sanchez
Annalie	Rodriguez Perez

Toolbar: +, ✎, ↶, ✖, ⏪, ⏩

Anexo 22 Prototipo del caso de uso: Gestionar enfermeras de salón

Gestionar Paciente de Cuerpo de Guardia

Paciente de Cuerpo de Guardia

Hoy: 2015/06/08

Nombre: ¿Ingresó? ¿Es Remitido?

Apellidos: ¿Tiene Tratamiento de Medicina Natural?

Edad: Contrarreferencia:

Sexo:

Dirección: Enfermedad que padece:

Municipio:

Nombre	Apellidos	Edad	Sexo	Dirección	Municipio	Ingreso
Enrique	Pastor	1	Masculino	Ave de	Yaguajay	No
Javier	Perdomo Villa	1	Masculino	Gangarina 125	Yaguajay	No
Pepe	Ruiz Fernandez	13	Masculino	Mariano Her ...	Sancti Spiritus	Si
asdf	asdf	2	Masculino	asd	Sancti Spiritus	Si
frank	Pastor	18	Masculino	Ave del	Sancti Spiritus	Si
af	asdf	2	Masculino	adfasdf	La Sierpe	No

+ ✎ ↶ ✖ ⏪ ⏩

Anexo 23 Prototipo del caso de uso: Gestionar paciente de cuerpo de guardia

Gestionar Municipio

Municipio

Nombre:

Nombre del Municipio

- Yaguajay
- Sancti Spiritus
- Trinidad
- La Sierpe
- Taguasco

+ ✎ ↶ ✖ ⏪ ⏩

Anexo 24 Prototipo del caso de uso: Gestionar municipio

Gestionar Enfermedades

Enfermedades

Nombre de la Enfermedad: Pulmonia

Etiología: ...

Grupo a que Pertenece: Infeccion Respiratoria Aguda

Nombre	Etiología	Grupo
traumatismo severo	dsdsdsdsd	Accidentes
Neumonia	dfg	Morbilidad
Infección Respiratori...	g	Infeccion Respiratori...

+ ✎ ↺ ✖ ⏪ ⏩

Anexo 25 Prototipo del caso de uso: Gestionar enfermedades

Gestionar Grupo de Enfermedades

Grupo Enfermedades

Nombre del Grupo: Enfermedades de Difusion obligatoria

Código: EDO

Descripción: ...

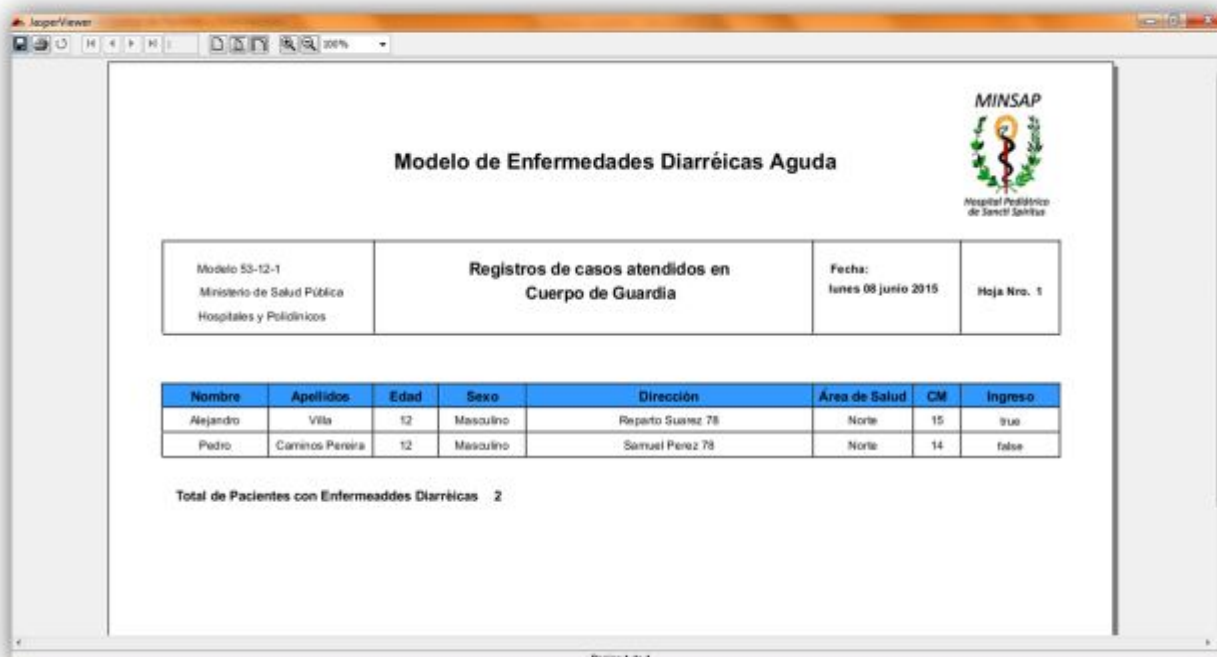
Nombre del Grupo	Descripción	Código
Infeccion Respiratori...	Afecta los	IRA
Accidentes	sdsd	ACC
Morbilidad	zF	MORB

+ ✎ ↺ ✖ ⏪ ⏩

Anexo 26 Prototipo del caso de uso: Gestionar grupo de enfermedades



Anexo 27 Prototipo del caso de uso: Mostrar modelo de Crisis Aguda de Asma Bronquial



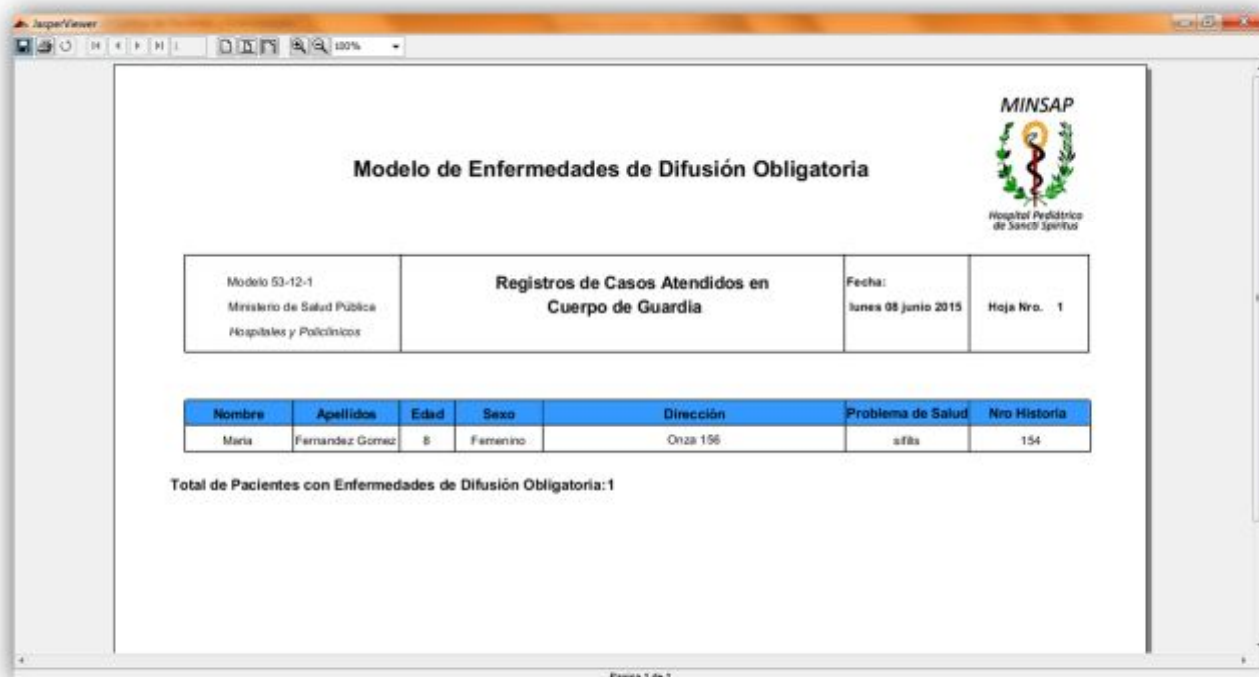
Anexo 28 Prototipo del caso de uso: Mostrar modelo de Enfermedades de Diarreicas Agudas



Anexo 29 Prototipo del caso de uso: Mostrar modelo de Infección respiratoria Aguda



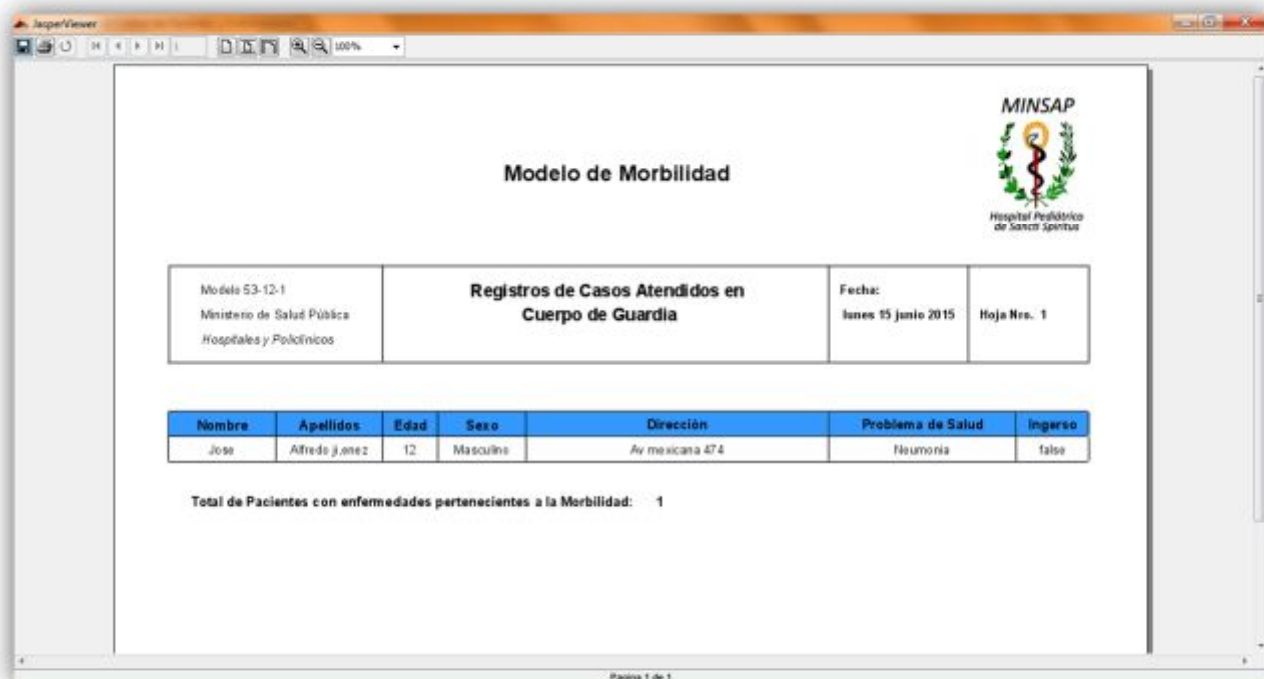
Anexo 30 Prototipo del caso de uso: Mostrar modelo de Síndrome Febril Agudo



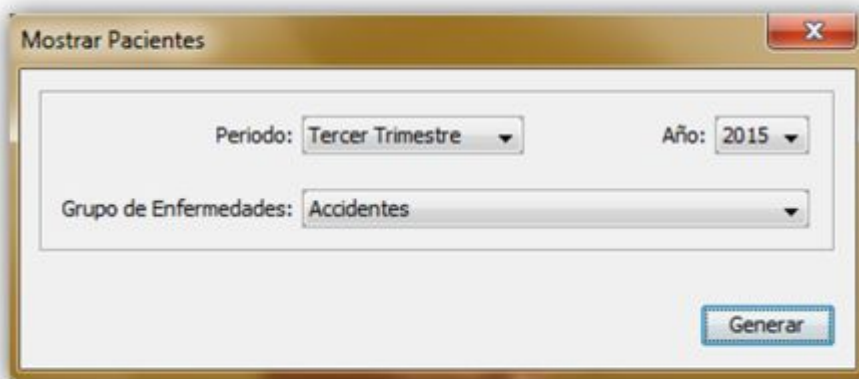
Anexo 31 Prototipo del caso de uso: Mostrar modelo de Enfermedades de Difusión Obligatoria



Anexo 32 Prototipo del caso de uso: Mostrar modelo de Accidentes



Anexo 33 Prototipo del caso de uso: Mostrar modelo de Morbilidad



Anexo 34 Prototipo del caso de uso: Mostrar casos de un Grupo dado en un período de fecha

A screenshot of a software window titled "Mostrar Pacientes". It contains three dropdown menus: "Grupo de edad" set to "5 a 9", "Municipio" set to "Sancti Spiritus", and "Enfermedad" set to "Neumonía". A "Generar" button is located at the bottom right.

Anexo 35 Prototipo del caso de uso: Mostrar casos de una Enfermedad por Grupo de Edades y Municipios

A screenshot of a software window titled "Mostrar Remitidos por Municipio". It features a single dropdown menu for "Municipio" set to "La Sierpe" and a "Generar" button at the bottom right.

Anexo 36 Prototipo del caso de uso: Mostrar pacientes remitidos de un municipio

A screenshot of a software window titled "Cantidad de remitidos por Municipio". It includes filters for "Mes" (6) and "Año" (2015). Below the filters is a table with two columns: "Municipio" and "Cantidad de Remitidos".

Municipio	Cantidad de Remitidos
Taguasco	2
La Sierpe	1

A "Generar" button is located at the bottom right of the window.

Anexo 37 Prototipo del caso de uso: Mostrar cantidad de remitidos por municipio

Porciento Anual de Pacientes dado un grupo

Porciento Anual de Pacientes dado un grupo

Año: 2015 Grupo: Morbilidad

14.29 %

Generar

Anexo 38 Prototipo del caso de uso: Mostrar porciento anual de pacientes dado un grupo

Mostrar Pacientes

Enfermedad: Infección Respiratoria Aguda

Generar

Anexo 39 Prototipo del caso de uso: Mostrar casos dada una enfermedad

Mostrar Pacientes

Grupo: Accidentes

Grupo de Edades de: 0 a 4

Período: Primer Trimestre Año: 2016

Municipio: Sancti Spiritus

Generar

Anexo 40 Prototipo del caso de uso: Mostar casos de un grupo de enfermedades en un intervalo de fecha, de un municipio y un grupo de edad determinado

Cantidad de operaciones por clase de cirugía

Año: 2015

Clase de Operación	Cantidad Operaciones
Menor Electiva	1
Menor Urgente	1
Mayor Electiva	2

Generar

Anexo 41 Prototipo del caso de uso: Mostrar Cantidad de operaciones por clase de cirugía

Informe Operatorio de Pacientes

Ministerio de Salud Pública
Hospital Pediátrico "Jose Martí
Perez" de Sancti Spiritus
Departamento de Registros
Médicos y Estadística

Informe Operatorio

Historia Clínica No. 26

Nombre		Apellidos			
Sergio manuel		perez			
Edad	Sexo	Fecha Anuncio	Fecha Operación	Hora Inicio	Hora Fin
14	Masculino	2016/04/08	2016/04/08	23:43	12:33
Salón	Clasificación de la Operación				
2	Mayor Urgente		Limpia Contaminada		Reintervención
Cirujano			Especialidad		
Carlos Manuel Ferrer			Cirujano Plastico		
Anestesiologo			Enfermera de Salón		
Carlos Lorenzo Valdivia			Iliana Mayedo Sanchez		
Patología Operatoria			Diagnóstico Operatorio		
sdf			asdf		

Página 1 de 7

Anexo 42 Prototipo del caso de uso: Mostrar modelo de cirugía

Porciento Anual de Operaciones dada la clase

Año 2016 Clase de Operación Mayor Electiva

66.67 %

Generar

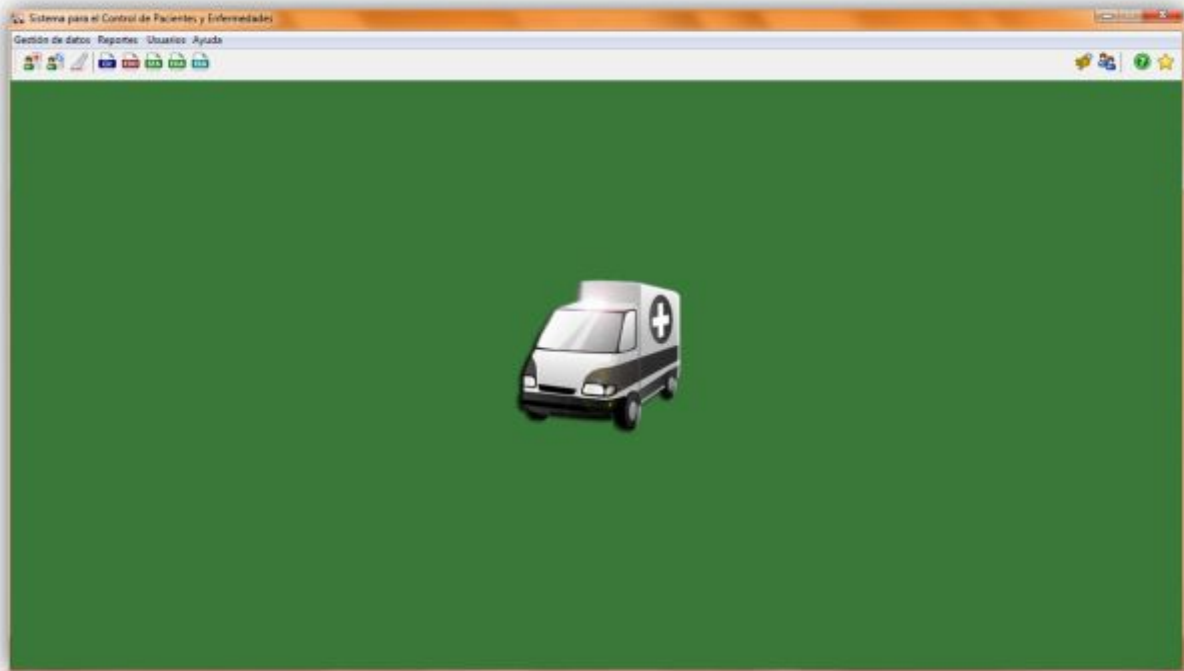
Anexo 43 Prototipo del caso de uso: **Mostrar el porciento anual de operaciones de una Clase**

Seleccione el Día

Día: 2015/06/03

Generar

Anexo 44 Prototipo del caso de uso: **Mostrar todos los pacientes operados en un día determinado**



Anexo 45 Ventana Principal



Anexo 46 Ayuda del Software