



# **UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

## **FACULTAD 7**

Trabajo de Diploma para Optar por el Título de  
Ingeniero en Ciencias Informáticas

Estadísticas de Adulto Mayor, Medios de Diagnóstico e  
Higiene-Epidemiología del Sistema Integral para la  
Atención Primaria de Salud

**Autora:**

Neilianys Salas Romero

**Tutores:**

Ing. Abdel Alonso Martínez

Ing. Nelson Francisco Fernández Pérez

**La Habana, junio de 2011**

**“Año 53 de la revolución”**

### Datos de Contacto

**Nombre:** Ing. Abdel Alonso Martínez - (aamartinez@uci.cu): Graduado de Ingeniero en Ciencias Informáticas en el año 2008. Actualmente labora en la Universidad de Ciencias Informáticas (UCI), desempeñándose como profesor de la Facultad # 7 vinculado a la producción. Actualmente es parte del proyecto Sistema Integral para la Atención Primaria de Salud (SIAPS) donde se desarrolla como Arquitecto de la Información y programador en el módulo Medios de Diagnóstico. Ha cursado varios cursos de postgrado como parte de su superación profesional.

**Nombre:** Ing. Nelson Francisco Fernández Pérez - (nfernandez@uci.cu): Graduado de Ingeniero en Ciencias Informáticas en el año 2010. Actualmente labora en la Universidad de Ciencias Informáticas (UCI), pertenece al Centro Informática Médica (Cesim), desempeñándose como Recién Graduado En Adiestramiento vinculado a la producción. Actualmente es parte del proyecto Sistema Integral para la Atención Primaria de Salud (SIAPS) donde se desarrolla como Jefe del Módulo de Configuración. Ha cursado varios cursos de postgrado como parte de su superación profesional.

## Resumen

La Atención Primaria de Salud está formada por varias instituciones encargadas de brindar atención integral de salud a la población. Dentro de estas instituciones, se encuentra el departamento de estadísticas, el cual no brinda directamente atención a la población, pero es de suma importancia puesto que guarda todo lo referente a la información general del centro. Con el objetivo de mejorar el trabajo que ejerce el personal de esta área, se realiza el proceso de gestión de la información estadística. En dicho departamento se crean los reportes estadísticos de las áreas Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología para así facilitar el trabajo en las unidades de la salud, donde se controla el flujo y procesamiento de la información estadística generada.

El sistema propuesto permite al área de salud generar información de los reportes estadísticos, así como realizar un análisis básico de los datos estadísticos, su control y comportamiento durante un período de tiempo determinado. Garantizando que el flujo de la información sea el mismo para todos los niveles y permitiendo obtener reportes específicos de los módulos Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología.

**Palabras Claves:** Atención Primaria de Salud, Estadísticas.

## Tabla de Contenidos

Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	6
1.1 Sistema Nacional de Salud.....	6
1.1.1 Atención Primaria de Salud (APS): .....	6
1.2 Marco Conceptual.....	6
1.3 Análisis de soluciones existentes.....	8
1.3.1 Ámbito Nacional.....	8
1.3.2 Ámbito internacional .....	9
1.4 Tecnologías, herramientas y metodologías. ....	11
1.4.1 Estilos Arquitectónicos. ....	11
1.4.2 Tecnologías en la Capa de Presentación. ....	13
1.4.3 Tecnología en la Capa de Negocio .....	14
1.4.4 Tecnologías en la Capa de Acceso a Datos.....	15
1.4.5 Lenguaje de programación.....	15
1.4.6 Herramientas a utilizar.....	16
1.4.7 Sistema gestor de Bases de Datos.....	17
1.4.8 Modelador de esquemas de Base de Datos.....	17
1.4.9 Metodologías de desarrollo .....	18
1.4.10 Modelo para la mejora y evaluación de procesos. ....	18
Capítulo 2. Características del Sistema .....	21
2.1 Modelo de Negocio .....	21

# TABLA DE CONTENIDOS

---

2.2 Diagrama de proceso del negocio.....	21
2.2.1 Realizar reportes de Medios de Diagnósticos.....	21
2.2.2 Realizar reportes de Adulto Mayor.....	24
2.2.3 Realizar reportes de Higiene y Epidemiología.....	27
2.3 Propuesta del sistema.....	30
2.3.1 Requerimientos.....	30
2.3.2 Especificación de los Requerimientos de Software.....	30
2.3.3 Requerimientos funcionales.....	30
2.3.4 Requerimientos no funcionales.....	31
Capítulo 3. Diseño del Sistema.....	37
3.1 Modelos de diseño.....	37
3.2 Patrones de diseño.....	37
3.3 Definición de Elementos de Diseño.....	39
3.3.1 Diagramas de Clases del Diseño.....	40
3.3.1.1 Reporte Ayudas Técnicas.....	40
3.3.1.2 Reporte Trayectoria Social.....	49
Capítulo 4. Implementación.....	59
4.1 Justificación de Integración con otros sistemas.....	59
4.2 Propuesta de seguridad del módulo.....	59
4.3 Implementación.....	60
4.4 Diagrama de Despliegue.....	60
4.5 Estándares de codificación y tratamiento de excepciones.....	61

# TABLA DE CONTENIDOS

---

4.5.1 Estándares de codificación.....	61
4.5.2 Tratamiento de excepciones. ....	63
Conclusiones.....	64
Recomendaciones.....	65
Referencias Bibliográficas .....	66
Bibliografía.....	68
Glosario de Términos .....	71

## Introducción

El desarrollo científico y tecnológico que viene ocurriendo de forma ascendente desde los años 40 del pasado siglo XX, ha tenido una significación especial para la actividad científica y una influencia extraordinaria en la vida social de todos los países, en todos los ámbitos, dígase económico, político y médico como cultural. En este periodo, más que en ningún otro, la humanidad ha sido testigo de grandes avances que han trascendido, en mayor o menor grado, en todos los países del mundo, al compás de la llamada Revolución Científico Técnica.

Cuba no se ha visto ausente a estos cambios puesto que desde 1959 la mayoría de las empresas comerciales e industriales del país utilizaban equipos basados en tarjetas perforadas, fue aquí que se inició prácticamente la informatización de la sociedad cubana y constituye hoy un gran paso de avance para desarrollar el país en las condiciones del mundo actual, que exige grandes recursos financieros, humanos y materiales. Con la introducción en el país de las computadoras personales (PC) en la década de los años 80, acompañadas de sistemas operativos y software de aplicaciones, se dio paso al proceso de informatización de la sociedad cubana.

El programa rector para la informatización de la sociedad cubana persigue promover el uso masivo de las Tecnologías de la Información y las Comunicaciones (TIC) a escala nacional, teniendo en cuenta los objetivos generales estratégicos que el país se ha propuesto y buscando impulsar de manera coherente todos los sectores, donde la salud y la educación constituyen unos de los más importantes.

El sistema de salud cubano por su carácter único, propicia la adopción de estrategias nacionales para la ejecución de las políticas trazadas en los más altos niveles de dirección y su concreción en las acciones correspondientes. Este comprende tres niveles de atención médica organizados en: Atención Primaria, Secundaria y Terciaria.

- ❖ Nivel de Atención Primaria: Se caracteriza principalmente por ser el primer contacto que realiza la población con la atención médica, ya que va dirigido a cada persona, a cada familia, a la comunidad y al medio ambiente.
- ❖ Nivel de Atención Secundaria: Se brinda en las instituciones hospitalarias que por lo general son de carácter provincial. En este nivel se atiende a toda la población que requiera de dichos servicios, pues aquí se llevan a cabo acciones mucho más complejas que en el nivel primario.

- ❖ Nivel de Atención Terciaria: Este nivel brinda servicios especializados de gran complejidad, dispone de recursos humanos muy capacitados y especializados, además de contar con tecnologías desarrolladas para diagnósticos y tratamientos. A este nivel pertenecen los Hospitales Especializados e Institutos de Investigaciones.

El desarrollo en el campo de la salud no está dado solamente por las transformaciones que a medicina se refieren, sino también, por todo lo que el país sea capaz de realizar para que la atención médica sea cada vez más eficiente y en beneficio total de la población. La informatización es un ejemplo de las tareas que se realizan para lograr los objetivos planteados de proporcionar un mejor servicio a la sociedad cubana.

A pesar de todos los avances tecnológicos que se han alcanzado en la actualidad no se ha solucionado correctamente el proceso de garantizar una correcta gestión de la información. La cantidad de información que se necesita procesar va ascendiendo cada vez más, debido a que los sistemas que se utilizan para controlar los datos generados por algunos procesos de gestión y control se han hecho más complejos.

Una de las áreas de vital importancia, donde se genera la información, que tiene como principal función obtener datos evidentes acerca de los logros o dificultades existentes, es el área de las estadísticas.

La estadística es una rama de las matemáticas que se ocupa de reunir, organizar y analizar datos numéricos, que se viene usando de manera sencilla desde los comienzos de la civilización, donde utilizaban representaciones gráficas y otros símbolos en pieles, rocas, palos de madera y paredes de cuevas para contar el número de personas, animales o ciertas cosas, ya hacia el año 3000 a.C. los babilonios usaban pequeñas tablillas de arcilla para recopilar datos en tablas. En la actualidad, la estadística se ha convertido en un método efectivo para describir con exactitud los valores de datos económicos, políticos, sociales, psicológicos, biológicos y físicos, y sirve como herramienta para relacionar y analizar dichos datos. [1]

El departamento de estadísticas dentro de la entidad médica, es el encargado de la gestión de los resúmenes estadísticos. Los mismos permiten la mejora de los procesos y una pronta toma de decisiones efectivas de acuerdo a los indicadores resultantes de cada departamento, en los resúmenes que pueden ser obtenidos.

Actualmente las instituciones médicas no cuentan con un sistema estadístico capaz de gestionar uniformemente la información generada, integrado en un sistema único, que relacione toda la información.



Los resúmenes emitidos, dependen de que la información sea entregada en tiempo por parte del personal que labora en los departamentos.

Es necesario que el proceso de actualización de la información se realice eficientemente, para garantizar una correcta sincronización de la información que reflejan. El control de las estadísticas, está totalmente influenciado por el personal que labora en el centro, en ocasiones el flujo de datos puede afectar el trabajo del personal pues se incrementan las atenciones durante el trabajo diario y en otras, falta de preparación sobre las nuevas tecnologías.

Los informes estadísticos se realizan de forma manual, lo que trae como consecuencia que ocurran errores en su recolección, puedan deteriorarse o perderse y que la búsqueda se torne engorrosa en un momento determinado. Dificulta la toma de decisiones oportunas y eficaces en los diferentes niveles de atención. Los informes contienen información consolidada de cualquier departamento del centro, en dependencia de las variables a ser representadas en el reporte.

Este elevado número de inconvenientes y situaciones en los centros repercuten en el correcto funcionamiento de la entidad. Lo que ocasiona que no existan informes con datos estadísticos que permitan hacer estudios y análisis de la población atendida, enfermedades más comunes y otros reportes. Además, la carencia de información estadística en tiempo real a los niveles de departamento y consultas, posibilita la demora de la información por los niveles de gerencia administrativa.

Lo anteriormente referido evidencia el incremento en el volumen de páginas de los registros, además de no posibilitar la generación de gráficos dinámicos, que permitan a los directivos y trabajadores, tener un consolidado actualizado y al alcance de todos. Lo que en ocasiones, dificulta la generación de estadísticas confiables y actualizadas.

Por todo lo antes expuesto, se hace necesario dar solución a las dificultades existentes para hacer más eficiente el proceso de generación de reportes en las áreas de Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología mediante la presente investigación. Se plantea como **problema a resolver**: ¿Cómo viabilizar el proceso de gestión de la información estadística en las áreas de Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología de la Atención Primaria de Salud?

Se define como **objeto de estudio**: el proceso de gestión de la información estadística en la Atención Primaria de Salud. Donde el **campo de acción** está enmarcado en: El proceso de gestión de la

información estadística en las áreas de Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología de la Atención Primaria de Salud.

Para la solución del problema se plantea como **objetivo general**: Desarrollar los reportes estadísticos de las áreas Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología en la Atención Primaria de Salud, para garantizar una mayor agilidad y confiabilidad en dicho proceso.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

- ❖ Realizar un análisis crítico y valorativo de los sistemas informáticos relacionados con las estadísticas de las áreas Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología, existentes a nivel nacional e internacional, estableciendo similitudes con la investigación en curso.
- ❖ Analizar la metodología, plataforma, tecnologías, librerías y herramientas en las que se apoya el proyecto SIAPS para garantizar la generación de los reportes estadísticos.
- ❖ Analizar los procesos de negocio asociados a la gestión de la información relacionada con la generación de reportes estadísticos de las áreas de Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología, logrando un modelo único como guía para la implementación.
- ❖ Identificar los reportes estadísticos de cada uno de los departamentos de las áreas Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología en la Atención Primaria de Salud.
- ❖ Aplicar la arquitectura J2EE 5.0, aprobada por el CESIM para el desarrollo de las aplicaciones web en los diferentes Departamentos de Producción.
- ❖ Obtener, mediante la guía de desarrollo propuesta por el proceso de mejora, los artefactos correspondientes a las fases: “Modelación del Negocio”, “Requisitos”, “Diseño e Implementación”.
- ❖ Implementar los reportes estadísticos de los departamentos de Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología en la Atención Primaria de Salud.
- ❖ Realizar los Diseños de Casos de Pruebas de cada uno de los reportes estadísticos implementados en las áreas Adulto Mayor, Medios de Diagnósticos e Higiene y Epidemiología en la Atención Primaria de Salud.

Este documento se encuentra compuesto por cuatro capítulos donde se refleja todo el trabajo investigativo, así como todo lo referente al diseño del sistema y la implementación de la solución propuesta, distribuido de la siguiente manera:

El **Capítulo I** Fundamentación Teórica: Contiene los aspectos esenciales para entender el entorno del problema a resolver. Se describen los conceptos fundamentales asociados al dominio del problema, el estado del arte de los sistemas informáticos en la atención primaria que gestionan la información relacionada con los problemas de salud, los aspectos generales del Sistema Nacional de Salud, además de incluir las tendencias y las tecnologías.

En el **Capítulo II** Características del sistema: Se plasma el marco conceptual asociado a la información que será manipulada por el sistema, llegándose a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

En el **Capítulo III** Diseño del Sistema: Se realiza el diseño del sistema a desarrollar, con el propósito de refinar y estructurar los requisitos obtenidos con anterioridad para facilitar la comprensión, preparación, modificación y mantenimiento de los mismos. Describe los aspectos relacionados al diseño de la solución propuesta, se modelan los diagramas de clases del diseño.

En el **Capítulo IV** Implementación: Se presenta la propuesta de solución para lograr una gestión eficiente de los requerimientos previamente identificados, se modela el diagrama de despliegue, aborda la descripción de los estándares de diseño, codificación, seguridad y además del tratamiento de errores en la solución del sistema.

## Capítulo 1. Fundamentación Teórica

Este capítulo tiene como objetivo fundamental abordar distintos aspectos que se utilizan como soporte teórico para el desarrollo de la aplicación. Se realiza un análisis de las tendencias actuales, lenguajes, y herramientas más utilizadas en el desarrollo de aplicaciones web, y se presentan las tecnologías y herramientas a utilizar para llevar a cabo la solución del problema que se enfrenta.

### 1.1 Sistema Nacional de Salud

En 1960 se realiza la transformación del Ministerio de Salubridad en Ministerio de Salud Pública (MINSAP) y se crea el Sistema Nacional de Salud (SNS). El Ministerio de Salud Pública (MINSAP) es el Organismo rector del Sistema Nacional de Salud (SNS). Encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la Salud Pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica. La Organización Mundial de la Salud (OMS) define al Sistema Nacional de Salud como: Un sistema que engloba todas las organizaciones, instituciones y recursos cuyo principal objetivo es llevar a cabo actividades encaminadas a mejorar la salud. El sistema nacional de salud en la actualidad se encuentra dividido en tres niveles los cuales son: Nivel de Atención Primaria de salud, Nivel de Atención Secundaria de salud y Nivel de atención Terciaria de salud.

**1.1.1 Atención Primaria de Salud (APS):** Debe ser entendida como el primer nivel de contacto de los individuos, familia y comunidad con el sistema nacional de salud llevando la atención primaria lo más cerca posible del lugar donde viven y trabajan las personas.

### 1.2 Marco Conceptual

*Estadísticas:* Se ocupa de los métodos y procedimientos para recoger, clasificar, resumir, hallar regularidades y analizar los datos, siempre y cuando la variabilidad e incertidumbre sea una causa intrínseca de los mismos; así como de realizar inferencias a partir de ellos, con la finalidad de ayudar a la toma de decisiones y en su caso formular predicciones. Podríamos por tanto clasificar la Estadística en descriptiva, cuando los resultados del análisis no pretenden ir más allá del conjunto de datos, e inferencial cuando el objetivo del estudio es derivar las conclusiones obtenidas a un conjunto de datos más amplio.

[2]

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

*Estadística descriptiva:* Describe, analiza y representa un grupo de datos utilizando métodos numéricos y gráficos que resumen y presentan la información contenida en ellos. [3]

*Estadística inferencial:* Apoyándose en el cálculo de probabilidades y a partir de datos muestrales, efectúa estimaciones, decisiones, predicciones u otras generalizaciones sobre un conjunto mayor de datos.

*Estadísticas de Salud:* Es la información numérica, imprescindible y cuantificable para conocer el Estado de Salud de la Población así como para planificar, organizar, evaluar y controlar programas y acciones de salud. La información estadística de salud debe reflejar lo más fiel posible la realidad objetiva que mide. Las estadísticas de salud tienen cada vez mayor alcance y complejidad en la medida que se amplía y desarrolla el que hacer en salud. [4]

Las estadísticas de la salud en el país están estructuradas de la siguiente forma: la Oficina Nacional de Estadística (ONE) es el órgano central del estado encargado de dirigir, ejecutar y controlar la política nacional en relación con las estadísticas oficiales del país. Los diferentes ministerios y niveles nacionales del estado cuentan con sus direcciones de estadísticas que satisfacen las demandas de información numérica de acuerdo a los requerimientos de cada uno y se rigen metodológicamente por la ONE.

*Medios de Diagnósticos:* Es un instrumento, procedimiento o conjunto de procedimientos a través de los cuales se llega a determinar el padecimiento de un paciente, clasificándose en diferentes especialidades diagnósticas. Por ejemplo: Laboratorio Clínico, Endoscopía, Optometría, Electrocardiograma, entre otros. La estadística en esta área juega un papel fundamental, puesto que si algún especialista necesita saber cuáles exámenes se le realizaron a un determinado paciente para saber su padecimiento, estos se encuentran registrados.

*Higiene y epidemiología:* La Epidemiología es la ciencia que estudia la frecuencia de aparición de enfermedades que afecten a comunidades humanas. Enfermedades que en la mayoría de los casos se conocen en forma de brotes, donde se depende de la higiene que se tenga al manipular alimentos, el cuidado con el medio ambiente, con la basura que se vierte diariamente en las calles, las bahías y los gases que las fabricas sin distinción dispersan en la atmosfera. Por esta causa la Higiene y la Epidemiología están relacionadas directamente y en la actualidad han surgido muchas instituciones que responden a esta asociación con el objetivo de engrandecer la higiene de la población, erradicando así la propagación de muchos casos de brotes epidérmicos.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

*Adulto mayor:* Es la atención que se les brinda a las personas de la tercera edad, donde tienen casas de abuelos, hogares de ancianos, entre otros, en los cuales se les dedica la atención médica necesaria. La estadística en este sector es de vital importancia, puesto que el objetivo de la misma es tener recopilados todos los datos necesarios de los pacientes desde que ingresan hasta que egresan.

## 1.3 Análisis de soluciones existentes

Con el desarrollo de las TIC el uso de la informática como herramienta de ayuda a la medicina es una realidad en auge. En la actualidad existen varios sistemas informáticos encaminados a la salud, de ellos un pequeño grupo se centran en la automatización de la gestión de la información relacionada con las Estadísticas. A continuación se especifican algunos de estos sistemas:

### 1.3.1 Ámbito Nacional

#### **Infomed: Estadísticas de Salud**

En este sitio podrá consultar las Estadísticas oficiales del estado de salud de la población cubana y comparaciones de esta con otros países. También encontrará datos de interés sobre el desempeño de la salud pública en Cuba. Se brinda información actualizada referente a aspectos medulares del estado de salud de la población cubana. Puede descargarlo en formato PDF.

**SIEN:** Sistema de Información Estadístico Nacional. Jerarquizado por la ONE independientemente de donde se recolecte, tabule y valide la información. Ejemplo: SIE de Nacimientos. Se recolecta y valida la Información estadística en las unidades de salud y se trasmite, tabula y emite por la ONE y sus dependencias. Los subsistemas del SIEN, emiten información general, de amplio uso por los diferentes sectores de la actividad política y socio económico del país. [5]

**SIEC:** Sistemas de Información Estadísticos Complementarios. Uno para cada Organismo de la Administración Central del Estado Cubano (OACE). Ej. SIE-C Salud, SIE-C Educación. Cada SIE-C produce estadísticas específicas de cada sector. [6]

**SIEL:** Sistemas de Información Estadísticos Locales. Se diseñan para territorios o regiones de acuerdo a necesidades locales. Requieren aprobación de las autoridades estadísticas de niveles superiores. El SIEL un sistema diverso debido a la versatilidad y se expresa por la carga estadística, número de variables, alcance geográfico, áreas del conocimiento e indicadores de salida que provee. Es uno de los sistemas informáticos mayores del país, con alto precio de ejecución. Cuenta con múltiples indicadores, de los

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

cuales algunos permanecen almacenados y otros son poco utilizados por los administradores y los demás usuarios. Cuando la carga de información es muy grande es probable que el sistema se deteriore, y esto unido a las dificultades de recursos, hace necesario el mantenimiento y la racionalización permanente.

El SIEL de Salud posee respaldo legal e incluye 68 subsistemas de información y 23 Reportes o Información adelantada (Partes), el mismo capta, procesa y emite información estadística de diferentes áreas del conocimiento la que se utiliza por los usuarios del sector y de otros sectores e instituciones del ámbito nacional e internacional. Poner en conocimiento de los usuarios el caudal de información estadística disponible, resulta importante.

En ocasiones, los usuarios no conocen o conocen parcialmente, la información que el SIE de Salud provee, las periodicidades y fechas de emisión, la forma y el lugar en que está almacenada, así como de qué manera pueden recuperarla. Todo sistema de información ha de tener un sub-sistema que se encargue de la recolección, flujo y procesamiento de datos que permita generar información numérica oportuna, confiable y necesaria. Este subsistema es llamado Sistema Estadístico. Para que el Sistema Estadístico se convierta en Sistema de Información, debe además contener elementos de análisis, evaluación, comparación y control que permita el mejor desempeño del proceso de toma de decisiones.

[7]

## 1.3.2 Ámbito internacional

### **DevInfo Sistema de Información Estadística del Fondo de las Naciones Unidas para la Infancia (UNICEF)**

DevInfo es un sistema informático para el manejo y difusión de informaciones estadísticas en función del cumplimiento de los Objetivos de Desarrollo del Milenio (ODM) propuestos por la Organización de las Naciones Unidas (ONU). Dicho programa informático es capaz de brindar extraordinarias posibilidades tanto en tablas, cuadros y gráficos con cuanto dato estadístico se deseen introducir, lo mismo a nivel local, nacional, como del resto de los países miembros de la ONU.

Este sistema se desarrolló en tecnología Microsoft. NET y proporciona el acceso a indicadores organizados por sectores, objetivos, temas y otros mecanismos para el manejo de datos. Se pueden importar datos desde una hoja de cálculo (Excel) mediante un formato estandarizado con el que trabaja y para obtener información se obtiene en formato HTML, PDF o XML.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

El sistema por su parte permite analizar los datos necesarios para la toma de decisiones con base en hechos reales; realizar el seguimiento y evaluación de resultados; relacionar los distintos niveles de planificación geográfica (nacional, sub-nacional y regional) y mostrar estadísticas básicas de los datos registrados: conteo, máximo, mínimo, media y desviación estándar. Dichos datos analizados, relacionados especialmente con el bienestar de la infancia se incluyen luego en la publicación insignia de la UNICEF. [8]

## **INEbase Sistema de Información Estadístico del Instituto Nacional de Estadísticas España (INE)**

INEbase es el sistema que utiliza el INE para el almacenamiento de la información estadística en España.

Contiene toda la información que el INE produce en formatos electrónicos, y siempre está actualizada y disponible en resúmenes de tablas y resultados detallados. La unidad básica de INEbase es la operación estadística, definida como el conjunto de actividades que conducen a la obtención de resultados estadísticos sobre un determinado sector o tema a partir de datos recogidos de forma individualizada. Para cada operación estadística en el sistema se tiene diseñada una página que presenta toda la información disponible relativa a la misma. En cuanto a la información numérica, para cada operación se presenta un resumen de resultados que contiene los datos estadísticos y un anexo de tablas con los principales resultados. Se publican normalmente en ficheros en formato Word, Excel o pdf. Incluyen también un apartado con los resultados detallados de la estadística, donde se encuentran los últimos resultados publicados hasta los más recientes. [9]

## **SUBDEIS**

El Subdepartamento de Estadísticas e Información de Salud (SUBDEIS) del Servicio de Salud Metropolitano Norte (SSMN), es una aplicación existente en Chile que tiene como objetivo general, coordinar, administrar e integrar un Sistema Único de Información en Salud de la Dirección del Servicio de Salud Metropolitano Norte, para que las estadísticas sean oportunas, homogéneas, confiables, de calidad y de utilidad para apoyar la planeación de los servicios de salud, evaluar el impacto de los programas médicos y mejorar la atención médica. Esta aplicación se encarga de la recogida de las estadísticas de población, vitales, morbilidad y atenciones de salud. [10]

De acuerdo con el estudio realizado los sistemas anteriormente expuestos no cumplen con los requisitos de integración plasmados en la estrategia de informatización de la salud en Cuba, lo cual imposibilita la



uniformidad en cuanto al uso del mismo sistema, y el acceso a ellos por el personal de estadísticas de la Atención Primaria para desarrollar sus actividades de la misma forma.

## 1.4 Tecnologías, herramientas y metodologías.

Las tecnologías, herramientas y metodologías a usar para dar cumplimiento al objetivo general de la investigación serán las definidas por el CESIM para el desarrollo de sistemas informáticos. A continuación se exponen definiciones, características y ventajas de su utilización.

### 1.4.1 Estilos Arquitectónicos.

#### Arquitectura en 3 capas

Entre los estilos de programación encontramos la programación por capas, este es un estilo de programación que tiene como objetivo fundamental la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

**Capa de presentación:** Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso.

**Capa de negocio:** Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse.

**Capa de datos:** Es donde residen los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos. [11]

#### Modelo Vista Controlador (MVC)

Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. Utilizando este tipo de patrón es posible conseguir más calidad y un mantenimiento más fácil. Al margen de todo esto, una de las cosas más importantes que permite el uso de este patrón consiste en normalizar y estandarizar el

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

desarrollo de Software. El patrón de diseño software MVC establece que la arquitectura se divide en tres partes fundamentales:

## **Modelo**

Todas las aplicaciones de *software* dejan a los usuarios manipular ciertos datos que proceden de una realidad sobre la que se pretende actuar. A estos datos en estado puro, que representan el estado de la realidad se les llama modelo: modelan la parte de la realidad sobre la que se desea actuar. El modelo, pues, es el objeto que representa y trabaja directamente con los datos del programa: gestiona los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los diferentes controladores y/o vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuándo deben reflejar un cambio en el modelo.

## **Vista**

La vista es el objeto que maneja la presentación visual de los datos gestionados por el Modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interacciona con el modelo a través de una referencia al propio modelo.

## **Controlador**

El controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Entra en acción cuando se realiza alguna operación, ya sea un cambio en la información del modelo o una interacción sobre la Vista. Se comunica con el modelo y la vista a través de una referencia al propio modelo. Además, JSF opera como un gestor que reacciona ante los eventos provocados por el usuario, procesa sus acciones y los valores de estos eventos, y ejecuta código para actualizar el modelo o la vista. [12]

## **Desarrollo de Software Basado en Componentes (DSBC)**

El ser humano desde tiempos muy antiguos aprendió a reutilizar el conocimiento existente y en la actualidad esta estrategia está presente en el desarrollo de software, debido a la complejidad de los sistemas computacionales actuales.

El proceso de construcción de una pieza de software con componentes ya existentes, da origen al principio de reutilización del software, mediante el cual se promueve que los componentes sean

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

implementados de una forma que permita su utilización funcional sobre diferentes sistemas en el futuro. [13]

El desarrollo de software basado en componentes permite reutilizar piezas de código pre-elaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión. [14]

Los beneficios más importantes de esta arquitectura son los siguientes:

**Reusabilidad de Servicios:** Reducción considerable de tiempos y costos de desarrollo de aplicaciones al utilizar servicios disponibles ya desarrollados, para resolver problemáticas comunes a otras aplicaciones. Aumentando por esta razón la robustez del nuevo sistema, al utilizarse software ya probado.

**Interoperabilidad de aplicaciones:** Disminución de la complejidad en el proceso de integración, pues se interactúa con elementos que se abstraen de la tecnología y ubicación de los servicios.

Se debe tener en cuenta que este tipo de arquitectura permite la integración de múltiples sistemas, lo que implica una dependencia en la información gestionada por cada uno de ellos. Los sistemas integrados deben convertir al software en una solución cualitativamente superior pero debe ser una relación de dependencia absoluta.

## 1.4.2 Tecnologías en la Capa de Presentación.

### JavaServer Faces (JSF) 1.2

La tecnología JavaServer Faces constituye un marco de trabajo (framework) de interfaces de usuario del lado de servidor para aplicaciones web basadas en tecnología Java.

Toda aplicación JavaServer Faces debe incluir una librería de etiquetas personalizadas que define las etiquetas que representan componentes UI, así como una librería de etiquetas para controlar otras acciones importantes, como validadores y manejadores de eventos. La implementación de JavaServer Faces, de Sun proporciona estas dos librerías. La librería de etiquetas de componentes elimina la necesidad de codificar componentes UI en HTML u otro lenguaje de marcas, lo que se traduce en el empleo de componentes completamente reutilizables. Y la librería principal (core) hace fácil registrar eventos, validadores y otras acciones de los componentes. [15]

### RichFaces 3.3.1

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

RichFaces es una librería de componentes visuales para JSF. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF. Se integra perfectamente en el ciclo de vida de JSF y contiene un set de componentes visuales, los más comunes para el desarrollo de una aplicación web rica (Rich Internet Application), con un amplio número que cubren casi todas las necesidades. [16]

## **Ajax4JSF**

Ajax4jsf es una librería open source que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript.

Ajax4jsf presenta mejoras sobre los propios beneficios del framework JSF incluyendo el ciclo de vida, validaciones, facilidades de conversión y el manejo de recursos estáticos y dinámicos. Permite definir un evento en una página que invoca una petición Ajax y luego las áreas de la página deberían sincronizarse con el árbol de componentes JSF después de que la petición Ajax cambie los datos en el servidor. [17]

## **Facelets**

Facelets es un framework simplificado de presentación, donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF. Permite crear plantillas para construir un árbol de componentes de forma que puedan definirse como composición de otros.

## **1.4.3 Tecnología en la Capa de Negocio.**

### **Servidor de Aplicaciones**

#### **Jboss Seam 2.1.1**

JBoss Seam es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB). Con seam basta agregar anotaciones propias de éste a los objetos Entidad y Session de EJB, logrando escribir menos código Java y XML.

Otra característica importante es que se pueden hacer validaciones en los POJOs (Plain Old Object Java) como además manejar directamente la lógica de la aplicación y de negocios desde los Session Beans.

#### **Enterprise Java Beans (EJB) 3.0**

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Es una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables utilizando el lenguaje de programación java. Permite a los desarrolladores enfocarse en la lógica del negocio sin tener que emplear tiempo en la conformación del código de la infraestructura. Desde el punto de vista del desarrollador es un pedazo de código java que se ejecuta en un entorno especializado denominado contenedor EJB que proporciona un conjunto de servicios. El servicio de persistencia es proporcionado por un framework especializado denominado proveedor de persistencia.

## **Plain Old Java Object (POJO's)**

Son las siglas utilizadas por programadores java para enfatizar el uso de clases simples y que no dependen de un framework en especial.

## **1.4.4 Tecnologías en la Capa de Acceso a Datos.**

### **Hibernate 3.3**

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Genera sentencias SQL a partir de sentencias HQL (Hibernate Query Language). A través de la implementación del estándar JPA que provee Hibernate 3.3, se puede realizar el acceso a datos. [18]

### **Java Persistence API (JPA)**

Proporciona un modelo de persistencia basado en POJO's para mapear bases de datos relacionales en Java. El Java Persistence API fue desarrollado por el grupo de expertos de EJB 3.0, aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones clientes; incluso fuera de la plataforma Java EE, por ejemplo, en aplicaciones Java SE.

En su definición, se han combinado ideas y conceptos de los principales frameworks de persistencia como Hibernate, Toplink y JDO, y de las versiones anteriores de EJB. Todos estos cuentan actualmente con una implementación JPA.

## **1.4.5 Lenguaje de programación.**

Se utiliza Java como lenguaje de programación orientado a objetos, el cual fue desarrollado por Sun Microsystems a principio de los años noventa. Java hace uso de la sintaxis de C y C++, pero tiene un

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir muchos errores, como la manipulación directa de punteros o memoria.

Tiene como ventaja que es un lenguaje multiplataforma, que se ha extendido cobrando cada día más importancia tanto en el ámbito de Internet como en la informática en general.

Java permite programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema.

## 1.4.6 Herramientas a utilizar.

### Eclipse 3.5.2

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma. La definición que da el proyecto Eclipse acerca de su software es: "una especie de herramienta universal un IDE abierto y extensible para todo y nada en particular". Eclipse es, en el fondo, únicamente un almacén (workbench) sobre el que se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los plugins adecuados.

### Visual Paradigm 6.4

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. VP-UML aporta a los desarrolladores de software una plataforma de desarrollo puntera para construir aplicaciones de calidad, mejores y más baratas con rapidez. Aporta una excelente interoperabilidad con otras herramientas CASE y muchos de los entornos IDE líderes del mercado. [19]

### Jasper Reports 3.0.1

Jasper Reports es un API Open Source que permite generar reportes en formatos PDF, XML, HTML, CSV, XLS, RTF, y TXT con solo unas cuantas líneas de código y guardar este reporte en un archivo. Permite integrar en los reportes cosas como gráficas de todos tipos, tablas cruzadas, y sub reportes para poder mostrar mejor los datos. Jasper Reports proporciona además un editor WYSIWYG para crear toda la parte visual de los reportes (formatos, posiciones, tipos de letra, estilos, imágenes, decoraciones, etc.) llamada iReport, la cual es un complemento perfecto para Jasper Reports.

## 1.4.7 Sistema gestor de Bases de Datos.

Como gestor de base de datos utilizaremos PostgreSQL 8.4, este es el gestor más avanzado de código abierto en la actualidad, ofrece servicios de 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). Es extensible, brinda gran estabilidad y confiabilidad y es multiplataforma. Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/PGSQL. Este lenguaje es comparable al lenguaje procedural del sistema de gestión de base de datos relacional Oracle, PL/SQL. Una de sus ventajas es la excelente documentación que existe, además de contar con una comunidad de usuarios y desarrolladores a los que acudir en caso de tener problemas.

**PgAdmin 1.10.5:** es la aplicación gráfica más popular y robusta de licencia Open Source para la administración de bases de datos en PostgreSQL. Se puede usar en Linux y Windows. Permite gestionar versiones a partir de PostgreSQL 7.3, así como versiones comerciales. Está diseñado para responder a las necesidades de los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Su interfaz gráfica soporta todas las características de PostgreSQL y permite que la administración sea mucho más sencilla. La aplicación incluye también un editor SQL con resaltado de sintaxis, un editor de código en el servidor, un agente para lanzar script programados y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets, y puede encriptarse mediante SSL para mayor seguridad.

## 1.4.8 Modelador de esquemas de Base de Datos.

Toad Data Modeler es una estupenda ayuda a la hora de diseñar y crear bases de datos. No solamente nos permite crear con toda libertad toda clase de esquemas, diagramas y diseños, sino que también genera el código SQL para construirlas. Quizá la mayor ventaja de Toad Data Modeler sea la versatilidad, ya que soporta cualquier sistema gestor de bases de datos: Access, Firebird, InterBase, MySQL, Oracle, Paradox, Postgre, Sybase, etc. Es decir, que las bases de datos que creamos con este programa servirán para cualquier sistema que utilicemos. Toad Data Modeler permite abordar el proceso de creación de una base de datos desde los cimientos: construyendo el diagrama entidad-relación, definiendo las reglas de integridad referencial.

## 1.4.9 Metodologías de desarrollo

### Proceso Unificado de Desarrollo de Software (RUP).

El Proceso Unificado de Desarrollo de Software (RUP), es una metodología de desarrollo de software orientada a objetos que proporciona un método disciplinado para asignar las tareas y responsabilidades dentro del equipo de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que resuelva las necesidades del usuario dentro de un cronograma predecible y al menor costo posible. Se caracteriza por ser iterativo e incremental, centrado en la arquitectura, guiado por casos de uso y dividir el proceso en ciclos de desarrollo que se agrupan en fases en las cuales las actividades se distribuyen entre 9 flujos de trabajo. Cada fase finaliza con un hito donde se debe tomar una decisión importante.

Para la descripción de los productos de trabajo resultantes de cada uno de los flujos de trabajo de RUP así como de las fases del ciclo vida definido en el IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos se utiliza UML, con la especificación del uso del estándar BPMN para la descripción de los procesos de negocio.

### Lenguaje Unificado de Modelado (UML) 2.0

El Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

### Notación utilizada para modelar los procesos del negocio.

La Notación para el Modelado de Procesos de Negocio (BPMN) es un nuevo estándar que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. BPMN proporciona a los negocios la capacidad de entender sus procedimientos internos en una notación gráfica, facilitando a las organizaciones la habilidad para comunicar esos procedimientos de una manera estándar.

## 1.4.10 Modelo para la mejora y evaluación de procesos.

Capability Maturity Model Integration (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Las mejores prácticas de CMMI se publican en documentos llamados modelos, los cuales contienen el conjunto de prácticas relacionadas que son ejecutadas de forma conjunta para conseguir determinados objetivos.



# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Así es como el modelo CMMI establece una medida del progreso, conforme al avance en niveles de madurez. Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. El alcanzar estas áreas o estadios se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables.

CMMI consta de 22 áreas de procesos distribuidas dentro de 5 niveles de madurez:

- ❖ Nivel 1: Inicial.
- ❖ Nivel 2: Administrado.
- ❖ Nivel 3: Definido.
- ❖ Nivel 4: Cuantitativamente administrado.
- ❖ Nivel 4: Optimizado.

Las organizaciones son evaluadas y reciben una calificación de nivel 1-5 siguiendo los niveles de madurez. Este enfoque se denomina Representación Escalonada.

La Universidad de las Ciencias Informáticas se encuentra desarrollando un proceso de mejora con el objetivo de obtener el Nivel 2 de CMMI. Las áreas de procesos que lo forman son:

- ❖ Administración de Requisitos (REQM).
- ❖ Planeación del Proyecto (PP).
- ❖ Monitoreo y Control del Proyecto (PMC).
- ❖ Medición y Análisis (MA).
- ❖ Aseguramiento de la Calidad de Procesos y productos (PPQA).
- ❖ Administración de la Configuración (CM).
- ❖ Administración de Acuerdos con Proveedores (SAM).

Dentro de las áreas de procesos mencionadas anteriormente la Universidad se encuentra enfocada en la Administración de Requisitos.

El objetivo de la Administración de Requisitos es gestionar los requisitos de los elementos del proyecto y sus componentes e identificar inconsistencias entre estos requisitos, el plan de proyecto y los elementos de trabajo. En este proceso se deben gestionar todos los requisitos del proyecto, tanto técnicos como no técnicos. Estos requisitos han de ser revisados conjuntamente con la fuente de los mismos así como con las personas que se encargarán del desarrollo posterior. Para llevar a cabo estas actividades es utilizado

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

el documento IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos realizado por la universidad y cuyo objetivo es definir el proceso de administración de requisitos.

Este documento establece el ciclo de vida a seguir asociado a los proyectos involucrados en el proceso de mejora, el cual consta de 9 fases (Estudio Preliminar, Modelación del Negocio, Requisitos, Análisis y Diseño, Pruebas Internas, Pruebas de Liberación, Despliegue y Soporte). En este se establece por cada fase la relación con los subprocesos descritos en el libro de procesos específico para el área de Administración de Requisitos.

La descripción del IPP-3510:2009 Libro de Proceso para la Administración de Requisitos incluye la definición de roles, sus responsabilidades y las habilidades en la ejecución de las actividades de los distintos procesos y los productos típicos de trabajos que se obtienen como resultado de la ejecución de dichas actividades.

En este capítulo se expusieron los requisitos y premisas definidos para la informatización del sector de la salud en el país. Se profundizó en el conocimiento de algunos conceptos necesarios para la comprensión de la presente investigación, así como en los antecedentes históricos de las estadísticas en la APS en los intentos de informatizarla, tanto en el ámbito nacional como internacional. Se realizó un análisis de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto.

## Capítulo 2. Características del Sistema

Este capítulo está dedicado a describir las características generales del sistema. Donde se abordará todo lo referente al modelado del negocio, mostrando claramente los aspectos fundamentales que darán paso a la descripción de la solución que se propone. Además se describen los principales procesos del negocio identificados y se especifican los requisitos funcionales y no funcionales de la solución.

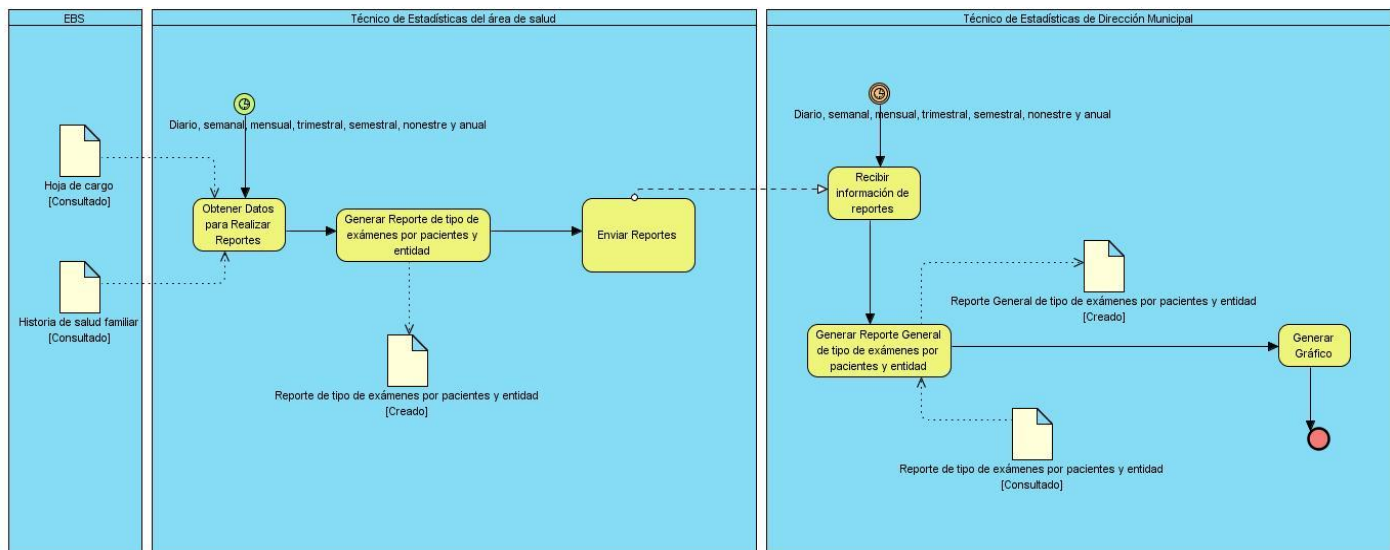
### 2.1 Modelo de Negocio

Para el desarrollo de cualquier sistema un paso fundamental es el análisis y la comprensión de los procesos existentes en el negocio, el cual posibilita y de hecho desencadena, una mirada más amplia y más profunda sobre la organización. El modelado del negocio es la técnica empleada para comprender los procesos del negocio de la organización. Las intenciones que se persiguen al realizarse este modelo son: comprender los problemas actuales e identificar mejoras potenciales, asegurarse de que los clientes, usuarios finales y desarrolladores tengan una idea común de la organización y derivar los requerimientos del sistema a partir del modelo de negocio que se obtenga.

Para la realización del modelado se utiliza la Notación de Gestión de Procesos de Negocios (BPMN). La cual aporta una mayor visibilidad de las actividades que se realizan y un mejor entendimiento del negocio, facilitando así la comunicación entre el diseño y la implementación de los procesos del negocio.

### 2.2 Diagrama de proceso del negocio.

#### 2.2.1 Realizar reportes de Medios de Diagnósticos.



## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

### Descripción del proceso Realizar reportes de Medios de Diagnósticos.

Nombre:	Realizar reportes de Medios de Diagnósticos.
Objetivos:	Llevar control estadístico de los exámenes realizados en un área de salud, en un límite de tiempo determinado.
Evento(s) que lo generan:	La necesidad de los técnicos de estadísticas de ver algún reporte referente a Medios de Diagnósticos.
Precondiciones:	El EBS notifique al técnico de estadísticas las hojas de cargo y la historia de salud familiar.
Poscondiciones:	Creación del reporte de tipo de exámenes por pacientes y entidad.
Reglas de Negocio:	Reglas 1, 3, 4 (Ver referencia 22)
Responsables:	Técnicos en estadísticas.
Clientes internos:	CESIM.
Clientes externos:	Técnico de estadísticas.
Entradas:	Hoja de Cargo, Historia de Salud Familiar.
Salidas:	Reporte de tipo de exámenes por pacientes y entidad.
Actividades:	<ol style="list-style-type: none"> <li>1- Obtener datos para realizar reportes.</li> <li>2- Generar Reporte de tipo de exámenes por pacientes y entidad.</li> <li>3- Enviar Reportes.</li> <li>4- Recibir Información de reportes.</li> <li>5- Generar Reporte General de tipo de exámenes por pacientes y</li> </ol>

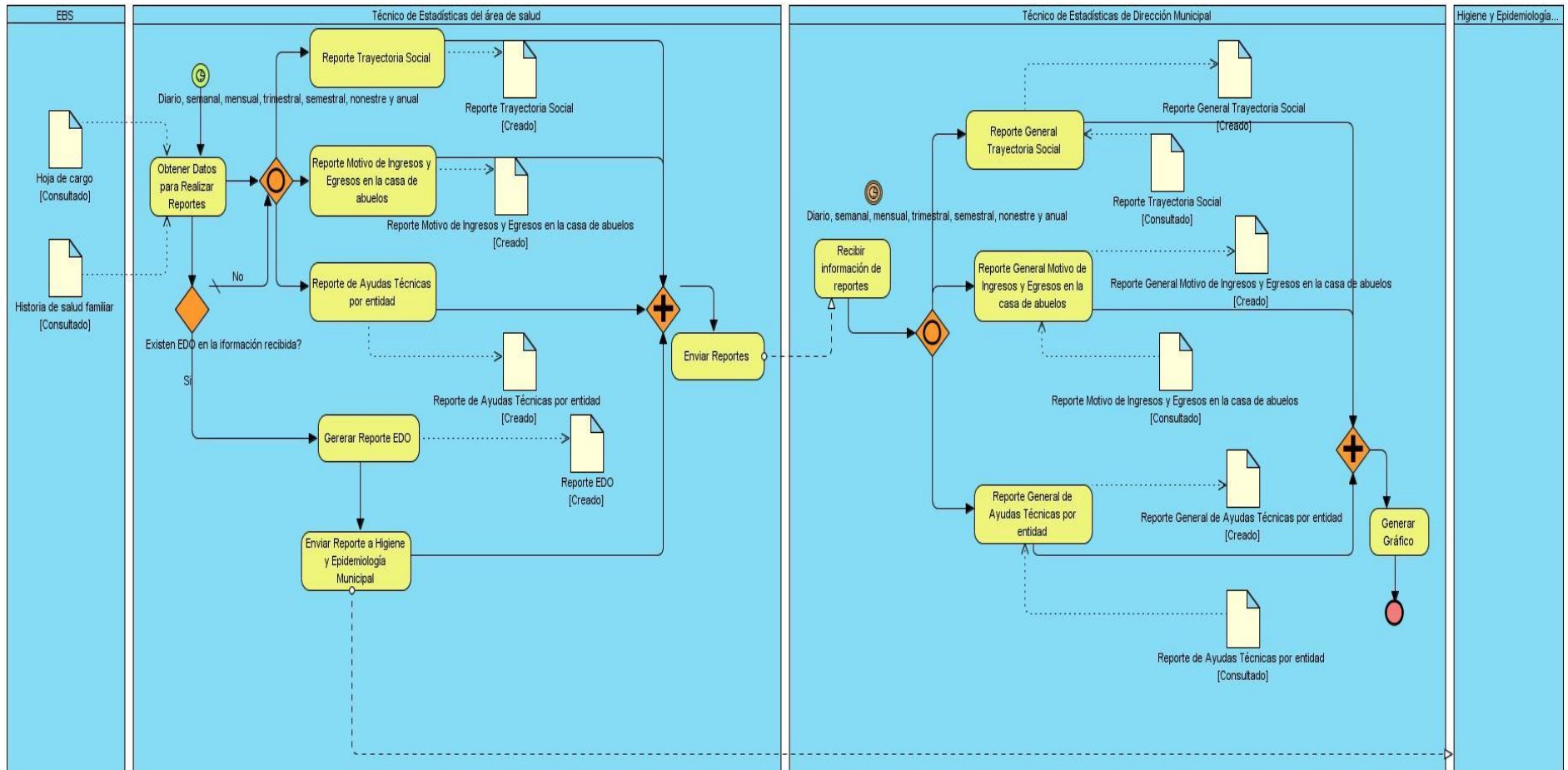
## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

entidad.

6- Generar Gráficos.

## 2.2.2 Realizar reportes de Adulto Mayor.



## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

### Descripción del proceso Realizar reportes de Adulto Mayor.

Nombre:	Realizar reportes de Adulto Mayor.
Objetivos:	Llevar control estadístico de la cantidad de pacientes que ingresan y egresan en las casas de abuelos en un límite de tiempo determinado.
Evento(s) que lo generan:	La necesidad de los técnicos de estadísticas de ver algún reporte referente al área de Adulto Mayor.
Precondiciones:	El EBS notifique al técnico de estadísticas las hojas de cargo y la historia clínica familiar.
Poscondiciones:	Creación del Reporte Trayectoria Social. Creación del Motivo de Ingresos y Egresos en la casa de abuelos. Creación del Reporte de Ayudas Técnicas.
Reglas de Negocio:	Reglas 1, 3, 4 (Ver referencia 22)
Responsables:	Técnicos en estadísticas.
Clientes internos:	CESIM.
Clientes externos:	Técnico de estadísticas.
Entradas:	Hoja de Cargo, Historia de Salud Familiar.
Salidas:	Reporte Trayectoria Social Reporte Motivo de Ingresos y Egresos en la casa de abuelos

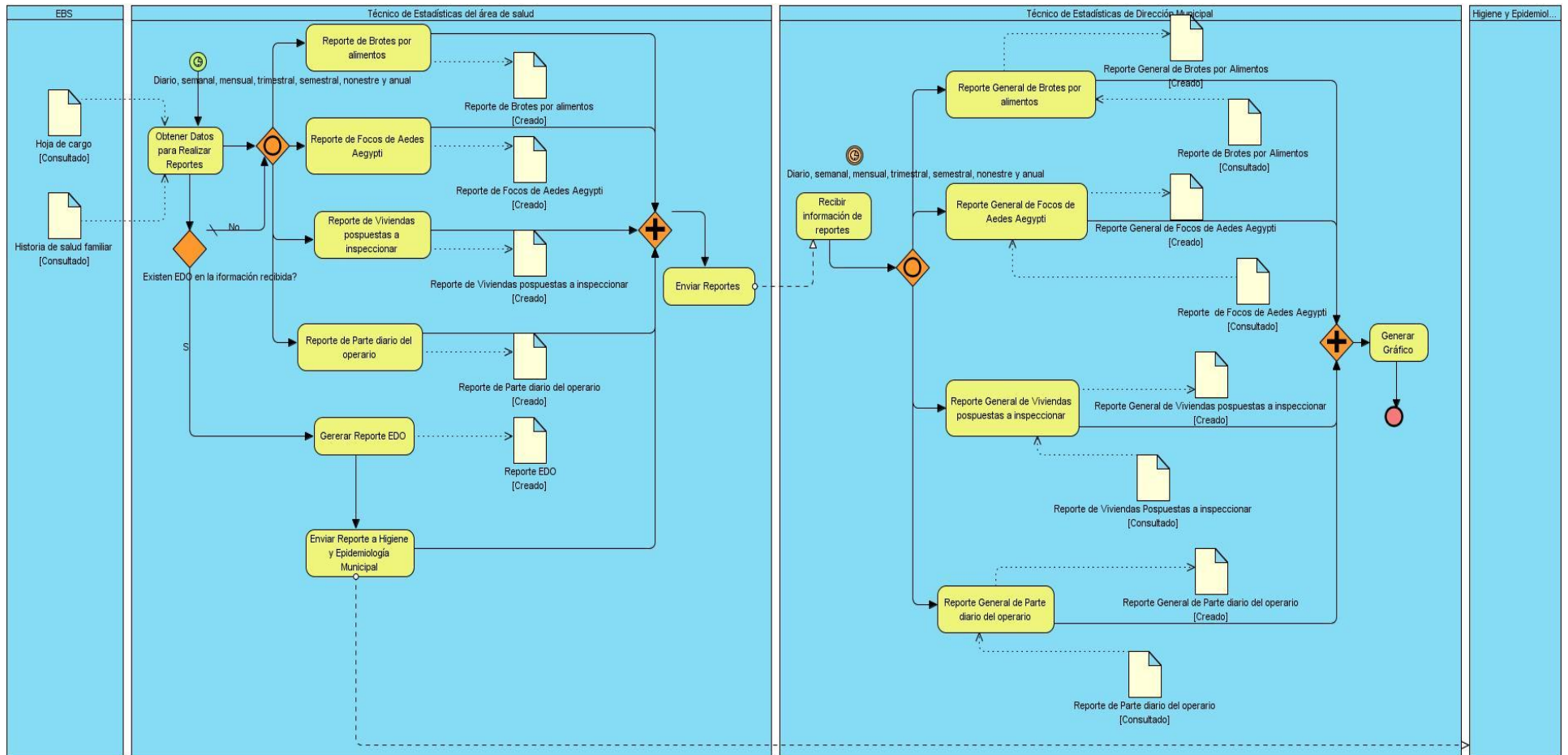
## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

	Reporte Ayudas técnicas
Actividades:	<ol style="list-style-type: none"><li>1- Obtener datos para realizar reportes.</li><li>2- Generar Reporte de Trayectoria Social.</li><li>3- Generar Reporte Motivo de Ingresos y Egresos en la casa de abuelos.</li><li>4- Generar Reporte Ayudas técnicas.</li><li>5- Generar Reporte EDO.</li><li>6- Enviar Reporte a Higiene y Epidemiología Municipal.</li><li>7- Enviar Reportes.</li><li>8- Recibir Información de reportes.</li><li>9- Generar Reporte General de Trayectoria Social.</li><li>10- Generar Reporte General de Motivo de Ingresos y Egresos en la Casa de Abuelos.</li><li>11- Generar Reporte General de Ayudas Técnicas.</li><li>12- Generar Gráficos.</li></ol>



## 2.2.3 Realizar reportes de Higiene y Epidemiología.



## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

### Descripción del proceso Realizar reportes de Higiene y Epidemiología.

Nombre:	Realizar reportes de Higiene y Epidemiología.
Objetivos:	Llevar control estadístico de la cantidad de pacientes infectados por brotes y viviendas inspeccionadas en un límite de tiempo determinado.
Evento(s) que lo generan:	La necesidad de los técnicos de estadísticas de ver algún reporte referente al departamento de Higiene y Epidemiología.
Precondiciones:	El EBS notifique al técnico de estadísticas las hojas de cargo y la historia de salud familiar.
Poscondiciones:	Creación del Reporte de Brotes por Alimentos. Creación del Reporte de Focos de Aedes Aegypti. Creación del Reporte de Viviendas pospuestas a inspeccionar. Creación del Reporte de Parte de Actividades del Operario.
Reglas de Negocio:	Reglas 1, 3, 4 (Ver referencia 22)
Responsables:	Técnicos en estadísticas.
Clientes internos:	CESIM.
Clientes externos:	Técnico de estadísticas.
Entradas:	Hoja de Cargo, Historia de Salud Familiar.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

Salidas:	<p>Reporte de Brotes por Alimentos.</p> <p>Reporte de Focos de Aedes Aegypti.</p> <p>Reporte de Viviendas pospuestas a inspeccionar.</p> <p>Reporte de Parte de Actividades del Operario.</p>
Actividades:	<ol style="list-style-type: none"><li>1- Obtener datos para realizar reportes.</li><li>2- Generar Reporte de Trayectoria Social.</li><li>3- Generar Reporte Motivo de Ingresos en la casa de abuelos.</li><li>4- Generar Reporte Motivo de Egresos en la casa de abuelos.</li><li>5- Generar Reporte Ayudas técnicas.</li><li>6- Generar Reporte EDO.</li><li>7- Enviar Reporte a Higiene y Epidemiología Municipal.</li><li>8- Enviar Reportes.</li><li>9- Recibir Información de reportes.</li><li>10- Generar Reporte General de Trayectoria Social.</li><li>11- Generar Reporte General de Motivo de Ingresos en la Casa de Abuelos.</li><li>12- Generar Reporte General de Motivo de Egresos en la Casa de Abuelos.</li><li>13- Generar Reporte General de Ayudas Técnicas.</li><li>14- Generar Gráficos.</li></ol>

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

### 2.3 Propuesta del sistema.

#### 2.3.1 Requerimientos.

La Metodología RUP propone como uno de sus flujos de trabajo el de Gestión de Requerimientos, el cual tiene su mayor peso en la fase de inicio, teniéndose como artefactos fundamentales de este flujo, la especificación de requisitos, actores y casos de usos del sistema.

#### 2.3.2 Especificación de los Requerimientos de Software.

Los requerimientos de software son condiciones o capacidades que tiene que alcanzar o poseer un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Estos posibilitan determinar la solución tecnológica a algún problema existente, logrando como resultado, un informe detallado de los requerimientos necesarios para la solución a dicho problema, que permita alcanzar los objetivos y solucionar los problemas existentes. [20]

Los requisitos se pueden clasificar en: funcionales y no funcionales, donde la primera clasificación corresponde a las especificaciones del sistema, mientras que los últimos expresan las cualidades que el producto debe tener.

#### 2.3.3 Requerimientos funcionales.

Los requerimientos funcionales especifican capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. [21]

Después de analizados los procesos del negocio, se han definido los siguientes requisitos funcionales:

1- Generar Trayectoria Social por Entidad.	16- Generar Focos de Aedes Aegypti.
2- Mostrar Trayectoria Social por Entidad.	17- Mostrar Focos de Aedes Aegypti.
3- Exportar Trayectoria Social por Entidad.	18- Exportar Focos de Aedes Aegypti.
4- Generar Motivo de Ingresos en la Casa de Abuelos.	19- Generar Viviendas Pospuestas a inspeccionar.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

5- Mostrar Motivo de Ingresos en la Casa de Abuelos.	20- Mostrar Viviendas Pospuestas a inspeccionar.
6- Exportar Motivo de Ingresos en la Casa de Abuelos.	21- Exportar Viviendas Pospuestas a inspeccionar.
7- Generar Ayudas Técnicas por entidad.	22- Generar Parte de Actividades del Operario.
8- Mostrar Ayudas Técnicas por entidad.	23- Mostrar Parte de Actividades del Operario.
9- Exportar Ayudas Técnicas por entidad.	24- Exportar Parte de Actividades del Operario.
10- Generar Brotes por Alimentos.	25- Generar Tipo de exámenes por pacientes.
11- Mostrar Brotes por Alimentos.	26- Mostrar Tipo de exámenes por pacientes.
12- Exportar Brotes por Alimentos.	27- Exportar Tipo de exámenes por pacientes.
13- Generar Motivo de Egresos en la Casa de Abuelos.	
14- Mostrar Motivo de Egresos en la Casa de Abuelos.	
15- Exportar Motivo de Egresos en la Casa de Abuelos.	

### 2.3.4 Requerimientos no funcionales.

Los requerimientos no funcionales tienen que ver con las propiedades o cualidades que de una forma u otra limitan al sistema y que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

### Usabilidad

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

- ❖ Usuarios normales: 20 días
- ❖ Usuarios avanzados: 30 días

### Confiabilidad

- ❖ En los servidores de los policlínicos y en el Centro de Datos Nacional se garantizará una arquitectura de máxima disponibilidad, tanto de servidores de aplicación como de base de datos.
- ❖ Se garantizarán políticas de respaldo a toda la información, evitando pérdidas en caso de desastres ajenos al sistema.
- ❖ Las informaciones médicas relacionadas con los pacientes y que vayan a ser intercambiadas con otros policlínicos por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas.
- ❖ Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan.
- ❖ Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.
- ❖ Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.
- ❖ Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- ❖ Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude.
- ❖ Los mecanismos serán capaces de informar al personal autorizado sobre posibles irregularidades que den indicios sobre la introducción de información falseada.
- ❖ El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

- ❖ Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista.
- ❖ El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.
- ❖ Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo.
- ❖ Se permitirá realizar réplica de la base de datos entre las diferentes entidades y estas a su vez con el Centro de Datos, esta réplica se podrá hacer de forma manual y automatizada a través de la red.
- ❖ Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema, para esto debe existir un registro de trazas que almacene todas las transacciones realizadas en el sistema, indicando para cada caso como mínimo: usuario que realizó la transacción, tipo de operación que se realizó, fecha y hora en que se realizó la operación e información contenida en el registro modificado.

### **Eficiencia**

- ❖ Se deberá usar siempre que sea posible el patrón Singleton, destruir referencias que ya no estén siendo usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.
- ❖ El sistema minimizará el volumen de datos en las peticiones y optimizará el uso de recursos críticos como la memoria.
- ❖ Se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.
- ❖ El sistema adoptará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

### **Soporte**

- ❖ Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación.
- ❖ El subsistema de escritorio brindará la posibilidad de actualizarse y realizar el soporte mediante la tecnología Java Web Start.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

- ❖ Las notificaciones de las deficiencias detectadas en la aplicación desplegada deberán realizarse por escrito.
- ❖ Una vez notificada por la entidad médica, la deficiencia detectada en la aplicación desplegada, el equipo de desarrollo deberá solucionarla en un período de 7 días.
- ❖ La capacitación y entrenamiento del profesional de salud para el uso del sistema, se realizará en un período de 6 meses.

### **Restricciones de diseño**

- ❖ La capa de presentación contendrá todas las vistas y la lógica de la presentación.
- ❖ El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio.
- ❖ La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario.
- ❖ La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas.
- ❖ El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.
- ❖ La nomenclatura de los bases de datos, tablas, campos de las tablas, esquemas, procedimientos almacenados o funciones, vistas, triggers y tipos de datos se elaborarán siguiendo los estándares definidos en el proyecto.
- ❖ La codificación se realizará siguiendo el estándar definido en el departamento.
- ❖ Las interfaces se realizarán siguiendo el estándar definido en el departamento.

### **Requisitos para la documentación de usuarios en línea y ayuda del sistema.**

- ❖ Se posibilitará el uso de ayudas y manuales dinámicos en línea sobre el funcionamiento del sistema.



# CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

## Interfaz

- ❖ Las ventanas del sistema contendrán los datos claros y bien estructurados para facilitar la interpretación correcta de la información.
- ❖ La interfaz contará con teclas de función y menús desplegables que faciliten y aceleren su utilización.
- ❖ La entrada de datos incorrecta será detectada claramente e informada al usuario.
- ❖ Todos los textos y mensajes en pantalla aparecerán en el idioma inherente al país o región donde se despliegue el sistema.
- ❖ Se interactuará con el sistema alas RIS para realizar solicitudes y obtener resultados de estudios radiológicos e imagenológicos a través de una interfaz definida en el proyecto.
- ❖ Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven).
- ❖ El sistema usará el formato estándar WSDL (Web Services Description Language) para la descripción de los servicios web.
- ❖ El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos.
- ❖ El sistema utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.
- ❖ Los equipos autoanalizadores se podrán programar desde el sistema para realizar análisis y obtener resultados automáticamente a través de una interfaz definida en el proyecto.

## Requerimientos de hardware

- ❖ Las estaciones de trabajo debe tener como mínimo 256 Mb de memoria RAM y un microprocesador de 2.0 Hz.
- ❖ La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad, residencia de la información y aplicaciones bajo esquemas seguros y confiables.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

- ❖ Los servidores de base de datos deberán tener: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.
- ❖ Los servidores de aplicaciones deberán tener: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.
- ❖ Los servidores de Intercambio deberán tener: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

### **Requerimientos de software**

- ❖ El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL).
- ❖ El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google Chrome 1 y Firefox 2.

En este capítulo, se realizó la descripción y modelación de los procesos del negocio asociados al campo de acción, obteniendo una perspectiva del sistema que se desea desarrollar partiendo de los requerimientos funcionales y no funcionales, los cuales definen las capacidades y cualidades que la aplicación debe tener. Con la culminación de este capítulo se sentaron las bases para el exitoso desarrollo de las disciplinas de Diseño e Implementación del Módulo de Estadísticas.

## Capitulo 3. Diseño del Sistema

El Diseño de un sistema de software da una mayor visión de los aspectos relacionados con los requerimientos no funcionales y favorece a la definición de una arquitectura estable y sólida, creando un plano del modelo de implementación.

Desempeña un papel muy importante en el desarrollo de un software porque describe todos los aspectos del sistema a construir, dentro del diseño es donde se fomenta la calidad del Proyecto. El Diseño es la única manera de materializar con precisión los requerimientos del cliente.

En este capítulo se especifica el modelo de diseño del sistema mediante diagramas de clases poniendo de manifiesto el flujo de procesos. Seguidamente se describen cada una de las clases y diagramas de clases, correspondientes al proceso involucrado, justificándose además el uso de algunos patrones.

### 3.1 Modelos de diseño.

Los Modelos de Diseño son una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas con la implementación.

Estos pueden contener: diagramas, clases, paquetes, subsistemas, cápsulas, protocolos, interfaces, relaciones, colaboraciones, atributos, entre otros que se puedan considerar en el sistema. Esta estructura es muy significativa para la arquitectura en general, debido a que no solo permite visualizar, especificar y documentar modelos estructurales, sino que también posibilita la construcción de sistemas ejecutables, aplicando ingeniería directa e inversa.

### 3.2 Patrones de diseño.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es un modelo que se puede seguir para la realización de una actividad determinada, es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características.

El modelo de datos se encarga del almacenamiento de los datos, su estructura y las relaciones entre los mismos. Para la administración de los datos se utiliza el framework Hibernate. El cual tiene como ventajas el control de la accesibilidad a la información almacenada, así como la realización de rápidas consultas.

## CAPÍTULO 3. DISEÑO DEL SISTEMA

---

Hibernate implementa una serie de patrones para la administración del modelo de datos, entre ellos están los siguientes:

- ❖ **Active Record:** Permite la asociación de filas únicas de la base de datos con objetos del lenguaje de programación usado.
- ❖ **Identity Map:** Mantiene referencias a los objetos cargados y devuelve las mismas cada vez que se solicita un objeto ya cargado. De esta forma, actúa como un caché dentro de la transacción, lo cual también contribuye a reducir el número de accesos a la base de datos.
- ❖ **Identity Field:** Permite mapear las relaciones entre referencias entre objetos y claves entre tablas.
- ❖ **Foreign Key Mapping:** Permite mapear las relaciones de asociación de clave foránea.
- ❖ **Association Table Mapping:** Permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, lo cual crea una base de datos orientada a objetos virtuales, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).
- ❖ **Lazy Load:** Carga los distintos componentes de una clase a medida que son solicitados.
- ❖ **Query Object:** Permite manipular las consultas en tiempo de ejecución. Dado que las consultas son los objetos, se pueden cambiar para agregar o quitar condiciones. Esto proporciona la flexibilidad necesaria para crearlas o modificarlas dinámicamente.

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Entre estos tipos de patrones se pueden destacar los siguientes:

- ❖ **Experto:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.
- ❖ **Creador:** Este patrón como su nombre lo indica es el que crea, el que guía la asignación de responsabilidades relacionadas con la creación de objetos, asigna la responsabilidad de que una clase B cree un Objeto de la clase A.
- ❖ **Alta cohesión:** La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento del diseño debe realizar una

labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

- ❖ **Bajo acoplamiento:** El acoplamiento de una clase es el conjunto de dependencias que tiene con otras clases. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la menor repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.
- ❖ **Controlador:** Asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

En la realización del presente modelo de diseño y con el objetivo de mejorar la calidad de los diagramas correspondientes a esta disciplina fueron aplicados los patrones mencionados anteriormente.

### 3.3 Definición de Elementos de Diseño.

La representación de las clases del diseño está asociada al uso de UML para el modelado de Aplicaciones WEB, siendo identificadas las siguientes clases: Página Servidor (Server Page), Página Cliente (Client Page) Formulario (Form), para la representación de las clases contenedoras del código de las páginas servidoras, clientes y los formularios respectivamente.

Para modelar los diagramas de paquetes y los de clases del diseño se utilizará la siguiente nomenclatura:

Diagrama de paquetes: Diagrama de paquetes\_<Nombre del módulo>, Diagramas de clases del diseño: DCD\_<Nombre de la funcionalidad>.

La clase contenedora del framework JSF se encarga de generar el código de las páginas clientes (<<construye>>), los formularios componen las páginas clientes (permite que entre las páginas clientes y sus formularios exista la relación de composición), a través de los cuales se muestran e introducen los datos necesarios, los cuales son enviados hacia el contenedor del framework JSF que construyó la pagina cliente asociada (<<envía>>). Las páginas clientes incluyen clases con ficheros JavaScript y librerías (<<incluyen>>).

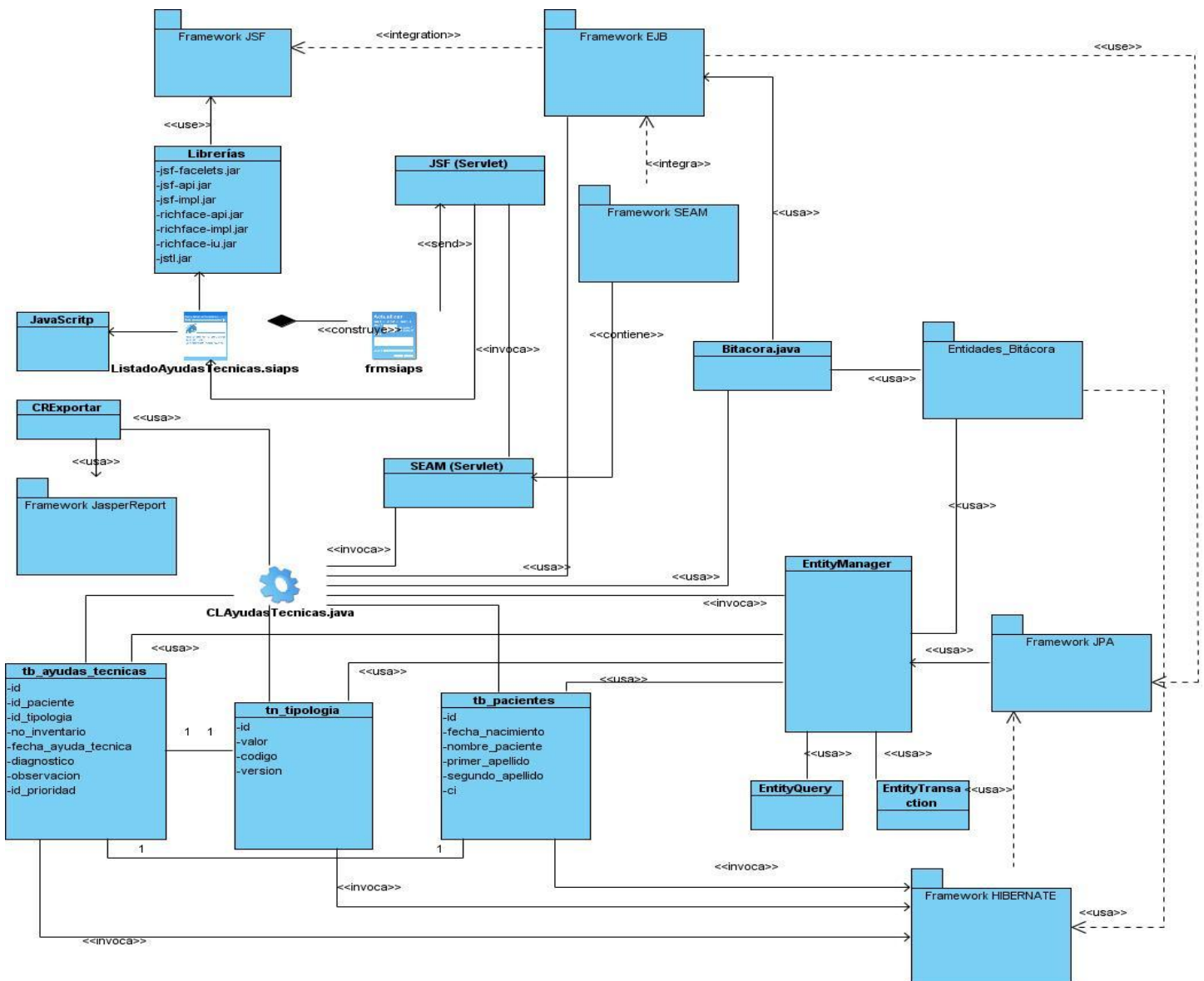
Para representar las clases previamente mencionadas se procedió usando la siguiente nomenclatura, frmsiaps, <Nombre de la vista>.siaps, CC<Nombre de la Opción>.java (Registrar), CL<Nombre de la Opción>.java (Listar), CM<Nombre de la Opción>.java (Modificar), CE<Nombre de la Opción>.java (Eliminar),

CD<Nombre de la Opción>.java (Detalles) y <Tr o Tn o Tb><Nombre de la Opción>.java, para los formularios, páginas clientes, clases servidoras y entidades respectivamente ubicadas en los diferentes niveles de la aplicación.

En los diagramas se encuentran otros elementos como son los frameworks JSF, EJB 3.0, JasperReport, JPA, Hibernate y Seam. Además, existen un conjunto de clases entre las que se identifican librerías, JavaScript, JSF (Servlets), Seam (Servlets), Bitácora, entre otras.

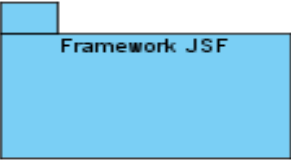
## 3.3.1 Diagramas de Clases del Diseño.

### 3.3.1.1 Reporte Ayudas Técnicas.



<b>Capa de Presentación</b>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid #ADD8E6; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>JavaScript</b></p> </div> <p>Figura 1. Clase JavaScript</p>	<p>Proveer el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.</p>
<b>Descripción:</b>	
<p>Permite realizar las validaciones del lado del cliente y construir páginas más dinámicas integradas a un navegador web</p>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid #ADD8E6; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>Librerías</b></p> <ul style="list-style-type: none"> <li>-jsf-facelets.jar</li> <li>-jsf-api.jar</li> <li>-jsf-impl.jar</li> <li>-richface-api.jar</li> <li>-richface-impl.jar</li> <li>-richface-iu.jar</li> <li>-jstl.jar</li> </ul> </div> <p>Figura 2. Clase Librerías</p>	<p>Permite la creación de interfaces.</p>
<b>Descripción:</b>	
<p>Contiene un conjunto de librerías que permiten construir una interfaz de usuario con componentes reutilizables y extensibles. Reducen significativamente la carga de construir y mantener aplicaciones web con componentes de interfaz del lado del servidor.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid #ADD8E6; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>JSF (Servlet)</b></p> </div> <p>Figura 3. Clase JSF Servlet</p>	<p>Interceptar las peticiones a las páginas JSF.</p>
<b>Descripción:</b>	
<p>Es el controlador de JSF que intercepta las peticiones de las páginas clientes, asociándoles a estas páginas, clases java que recogen la información introducida y que disponen de métodos que responden a las acciones del usuario. Además, prepara el contexto JSF antes de enrutar a las páginas</p>	

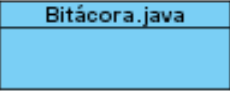
correspondientes e interviene en la construcción de la respuesta para generar la vista, luego de ser invocada una petición.

<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 4. Paquete Framework JSF</p>	<p>Simplificar el desarrollo de las interfaces de usuario en aplicaciones java basadas en el patrón Modelo-Vista-Controlador.</p>

**Descripción:**

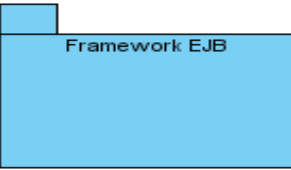
La tecnología Java Server Faces es un framework de los componentes de la interfaz de usuario y es válido para todas aquellas aplicaciones web basadas en la tecnología JAVA, está basado en el patrón MVC. Forma parte de la especificación JEE 5.

### Capa de Negocio

<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 5. Clase Bitácora.java</p>	<p>Proveer las funcionalidades para realizar las auditorias del sistema.</p>

**Descripción:**

Permite realizar las funciones para almacenar datos como la fecha, hora, usuario, contraseña, entre otros, del usuario que inicia y finaliza la sesión. Además, datos como el módulo y funcionalidad accedidos así como las acciones ejecutadas.

<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 6. Paquete Framework EJB</p>	<p>Encapsular la lógica de negocio que cumplimenta el propósito de la aplicación.</p>
<b>Observaciones:</b>	
<p>El Framework EJB está incluido en las capas de Negocio y Persistencia.</p>	


**Descripción:**

Es una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables utilizando

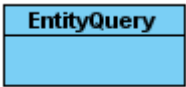


el lenguaje de programación Java. El objetivo de Enterprise JavaBeans (EJB) 3.0 es simplificar el desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java. Forma parte de la especificación JEE 5.


### Capa de Datos

Nombre:	Propósito:
 <p>Figura 7. Clase EntityManager</p>	<p>Gestionar las entidades suministrando servicios de persistencia.</p>

**Descripción:**  
Permite realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) que impliquen entidades.

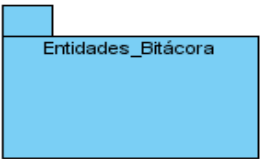
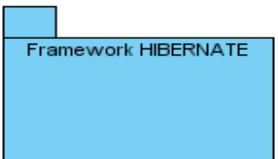
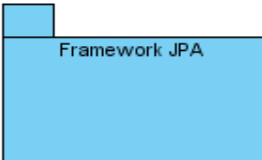
Nombre:	Propósito:
 <p>Figura 8. Clase EntityQuery</p>	<p>Agregar consultas que pueden aplicarse a las entidades del modelo.</p>

**Descripción:**  
Permite encontrar objetos persistentes manejando cierto criterio de búsqueda. Permite realizar peticiones a la base de datos y controla cómo se ejecuta dicha petición. Se utiliza para enlazar los parámetros de la petición, limitar el número de resultados devueltos por la petición y para ejecutar dicha petición.

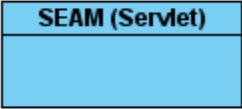
Nombre:	Propósito:
 <p>Figura 9. Clase EntityTransaction</p>	<p>Agrupar las operaciones sobre datos persistentes en una unidad transaccional.</p>

**Descripción:**  
Permite realizar operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.


Nombre:	Propósito:
---------	------------

 <p>Figura 10. Clase Entidades Bitácora</p>	<p>Agrupar las entidades que contienen la información de las auditorías del sistema.</p>
<b>Descripción:</b>	
<p>Contiene el conjunto de entidades que poseen la información de los usuarios y sus trazas en cuanto a sesión utilizada, módulos accedidos, funcionalidades permitidas, acciones realizadas y atributos modificados. Son utilizadas por la clase Bitácora.java para realizar las auditorías del sistema.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 11. Paquete Framework HIBERNATE</p>	<p>Proveer el mapeo objeto/relacional con la base de datos.</p>
<b>Descripción:</b>	
<p>Conjunto de clases agrupadas en componentes que constituyen una herramienta de Mapeo objeto/relacional ó ORM de código abierto (Object Relational Mapping) y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada permite generar Bases de Datos en cualquiera de los entornos soportados: Oracle, PostgreSQL, DB2, MySQL, entre otras.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 12. Paquete Framework JPA</p>	<p>Unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.</p>
<b>Descripción:</b>	
<p>Conjunto de clase agrupadas en componentes que constituyen la API de persistencia desarrollada para</p>	

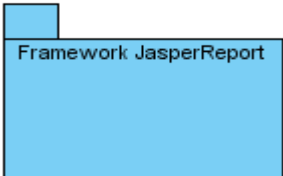
la plataforma Java EE e incluida en el estándar EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes de software EJB. Permite unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.

Nombre:	Propósito:		
<div style="text-align: center; margin-bottom: 10px;">  </div> <p>Figura 13. Clase SEAM Servlet</p>	<p>Proveer la interacción de la capa de presentación con la de negocio.</p> <tr style="background-color: #e6f2ff;"> <th style="padding: 5px;">Observaciones:</th> </tr> <td style="padding: 10px; vertical-align: top;"> <p>No se encuentra en ninguna capa en específico, sino que se encuentra ubicado entre la capa de Presentación y la de Negocio.</p> </td>	Observaciones:	<p>No se encuentra en ninguna capa en específico, sino que se encuentra ubicado entre la capa de Presentación y la de Negocio.</p>
Observaciones:			



**Descripción:**  
 Es el controlador de SEAM que capta las peticiones derivadas de la interacción del usuario después de interactuar con el Servlet de JSF. Enrutar las peticiones hacia los Beans que posibilitarán darle respuesta a la petición solicitada. Interviene en la integración de las capas de presentación y negocio.

Nombre:	Propósito:
<div style="text-align: center; margin-bottom: 10px;">  </div> <p>Figura 14. Clase CRExportar</p>	<p>Proveer las funcionalidades para exportar a diferentes formatos.</p>


**Descripción:**  
 Permite realizar las funciones para exportar a diferentes tipos de archivos mediante el uso de las librerías del Framework JasperReport.

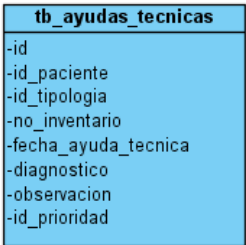
Nombre:	Propósito:
<div style="text-align: center; margin-bottom: 10px;">  </div> <p>Figura 15. Paquete framework</p>	<p>Añadir características de generación de reportes a aplicaciones java.</p>

JasperReport.	
<b>Descripción:</b>	
<p>Es una librería de clases de Java de código abierto desarrollada para facilitar el agregar capacidades de reporte a las aplicaciones Java. Permite realizar reportes de código abierto que tiene como función el llevar documentos ricos en contenido a la pantalla, a la impresora, o a archivos PDF, HTML, XLS, CSV y XML.</p>	

Capa de Presentación	
Nombre:	Propósito:
<div style="text-align: center;">  <p><b>ListadoAyudasTecnicas.siaps</b></p> </div> <p>Figura 16. Clase ListadoAyudasTecnicas.siaps</p>	<p>Proveer la interacción con el usuario.</p>
<b>Descripción:</b>	
<p>La clase ListadoAyudasTecnicas.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá listar todos los datos necesarios para crear el reporte y que todo quede correctamente registrado. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>	
Nombre:	Propósito:
<div style="text-align: center;">  <p><b>frmsiaps</b></p> </div> <p>Figura 17. frmsiaps</p>	<p>Enviar los datos a las páginas servidoras.</p>
<b>Descripción:</b>	

La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente la cual posibilita ver el reporte de las ayudas técnicas. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$\_POST.

Capa de Negocio	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>CLAyudasTecnicas.java</b></p> <p>Figura 18.</p>	<p>Proveer una respuesta a las peticiones realizadas en la vista.</p>
<b>Descripción:</b>	
<p>La clase CLAyudasTecnicas.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de mostrar los datos del reporte de las ayudas técnicas solicitados desde la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.</p>	

Capa de Datos	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 19. Tb_ayudas_tecnicas</p>	<p>Proveer el mapeo con la base de datos.</p>
<b>Descripción:</b>	

La clase `tb_ayudas_tecnicas.java` es una clase que se ejecuta del lado del servidor. En ella se almacenan todos los datos referentes a la ayuda técnica que recibió el paciente. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.

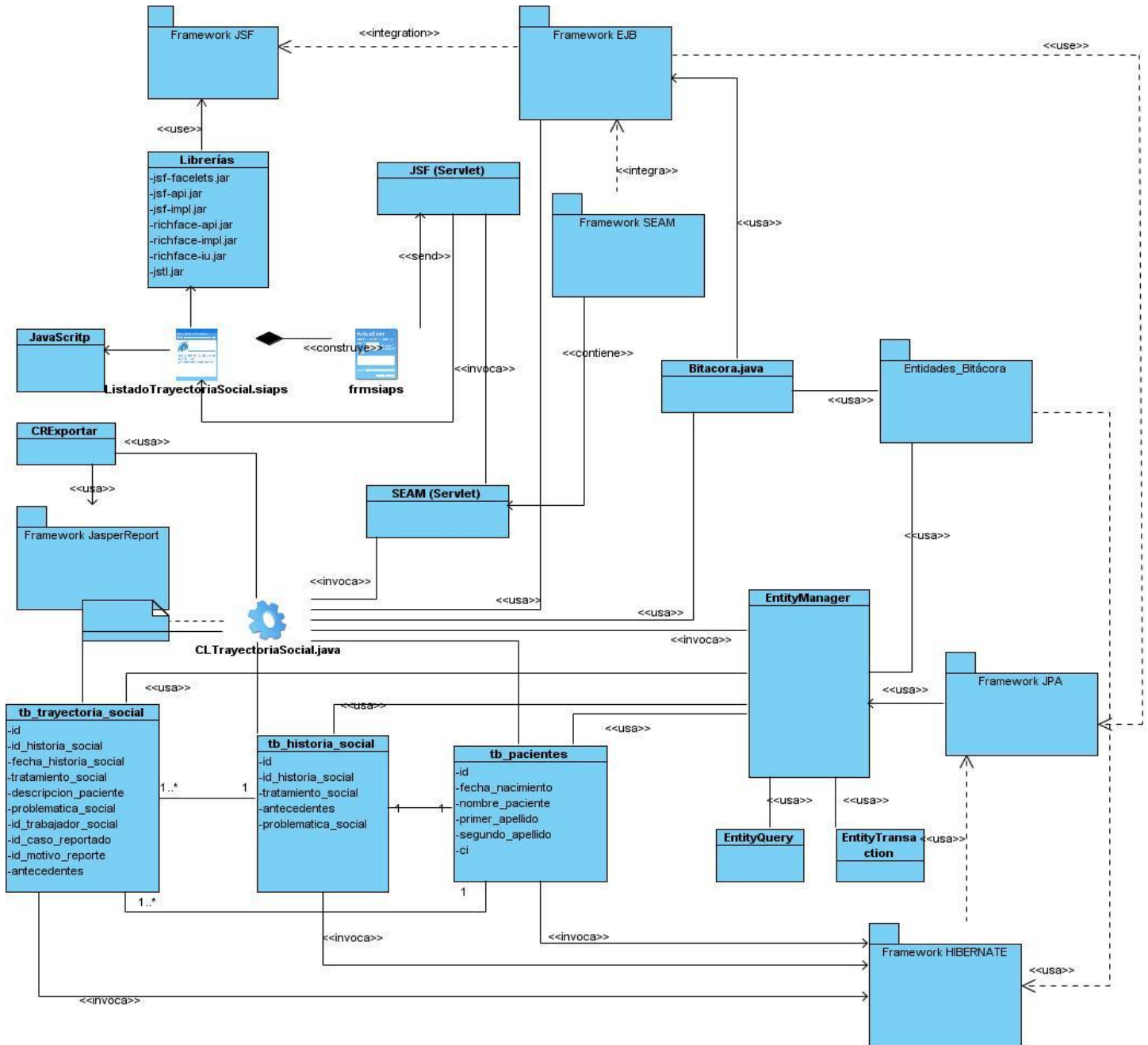
Nombre:	Propósito:
<div style="border: 1px solid #00aaff; padding: 5px; width: fit-content; margin: 0 auto;"> <p style="text-align: center; margin: 0;"><b>tn_tipologia</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-valor</li> <li>-codigo</li> <li>-version</li> </ul> </div> <p style="margin-top: 10px;">Figura 20. Tn_tipologia</p>	<p>Proveer el mapeo con la base de datos.</p>




**Descripción:**  
 La clase `tn_tipologia.java` es una clase que se ejecuta del lado del servidor. En ella se puede consultar el tipo de ayuda técnica que recibió el paciente. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.

Nombre:	Propósito:
<div style="border: 1px solid #00aaff; padding: 5px; width: fit-content; margin: 0 auto;"> <p style="text-align: center; margin: 0;"><b>tb_pacientes</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-fecha_nacimiento</li> <li>-nombre_paciente</li> <li>-primer_apellido</li> <li>-segundo_apellido</li> <li>-ci</li> </ul> </div> <p style="margin-top: 10px;">Figura 21. Tb_pacientes</p>	<p>Proveer el mapeo con la base de datos.</p>

**Descripción:**  
 La clase `tb_pacientes.java` es una clase que se ejecuta del lado del servidor. En ella se almacena los datos de los pacientes. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.

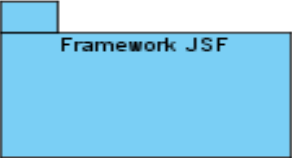
## 3.3.1.2 Reporte Trayectoria Social.



<b>Capa de Presentación</b>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 1. Clase JavaScript</p>	<p>Proveer el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.</p>
<b>Descripción:</b>	
<p>Permite realizar las validaciones del lado del cliente y construir páginas más dinámicas integradas a un navegador web</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 2. Clase Librerías</p>	<p>Permite la creación de interfaces.</p>
<b>Descripción:</b>	
<p>Contiene un conjunto de librerías que permiten construir una interfaz de usuario con componentes reutilizables y extensibles. Reducen significativamente la carga de construir y mantener aplicaciones web con componentes de interfaz del lado del servidor.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 3. Clase JSF Servlet</p>	<p>Interceptar las peticiones a las páginas JSF.</p>
<b>Descripción:</b>	
<p>Es el controlador de JSF que intercepta las peticiones de las páginas clientes, asociándoles a estas páginas, clases java que recogen la información introducida y que disponen de métodos que responden a las acciones del usuario. Además, prepara el contexto JSF antes de enrutar a las páginas correspondientes e interviene en la construcción de la respuesta para generar la vista, luego de ser</p>	



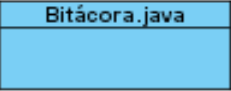
invocada una petición.

Nombre:	Propósito:
 <p>Figura 4. Paquete Framework JSF</p>	<p>Simplificar el desarrollo de las interfaces de usuario en aplicaciones java basadas en el patrón Modelo-Vista-Controlador.</p>

**Descripción:**


La tecnología Java Server Faces es un framework de los componentes de la interfaz de usuario y es válido para todas aquellas aplicaciones web basadas en la tecnología JAVA, está basado en el patrón MVC. Forma parte de la especificación JEE 5.

### Capa de Negocio

Nombre:	Propósito:
 <p>Figura 5. Clase Bitácora.java</p>	<p>Proveer las funcionalidades para realizar las auditorias del sistema.</p>

**Descripción:**

Permite realizar las funciones para almacenar datos como la fecha, hora, usuario, contraseña, entre otros, del usuario que inicia y finaliza la sesión. Además, datos como el módulo y funcionalidad accedidos así como las acciones ejecutadas.


Nombre:	Propósito:
 <p>Figura 6. Paquete Framework EJB</p>	<p>Encapsular la lógica de negocio que cumplimenta el propósito de la aplicación.</p>
Observaciones:	
<p>El Framework EJB está incluido en las capas de Negocio y Persistencia.</p>	

**Descripción:**


Es una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables utilizando el lenguaje de programación Java. El objetivo de Enterprise JavaBeans (EJB) 3.0 es simplificar el

desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java. Forma parte de la especificación JEE 5.


### Capa de Datos

Nombre:	Propósito:
 <p>Figura 7. Clase EntityManager</p>	<p>Gestionar las entidades suministrando servicios de persistencia.</p>

**Descripción:**  
 Permite realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) que impliquen entidades.

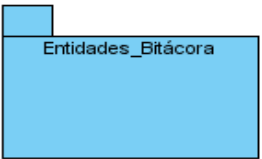
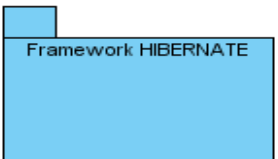
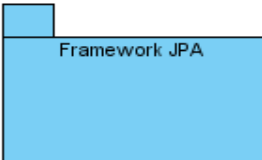
Nombre:	Propósito:
 <p>Figura 8. Clase EntityQuery</p>	<p>Agregar consultas que pueden aplicarse a las entidades del modelo.</p>

**Descripción:**  
 Permite encontrar objetos persistentes manejando cierto criterio de búsqueda. Permite realizar peticiones a la base de datos y controla cómo se ejecuta dicha petición. Se utiliza para enlazar los parámetros de la petición, limitar el número de resultados devueltos por la petición y para ejecutar dicha petición.

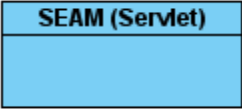
Nombre:	Propósito:
 <p>Figura 9. Clase EntityTransaction</p>	<p>Agrupar las operaciones sobre datos persistentes en una unidad transaccional.</p>

**Descripción:**  
 Permite realizar operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.


Nombre:	Propósito:

 <p>Figura 10. Clase Entidades Bitácora</p>	<p>Agrupar las entidades que contienen la información de las auditorías del sistema.</p>
<b>Descripción:</b>	
<p>Contiene el conjunto de entidades que poseen la información de los usuarios y sus trazas en cuanto a sesión utilizada, módulos accedidos, funcionalidades permitidas, acciones realizadas y atributos modificados. Son utilizadas por la clase Bitácora.java para realizar las auditorías del sistema.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 11. Paquete Framework HIBERNATE</p>	<p>Proveer el mapeo objeto/relacional con la base de datos.</p>
<b>Descripción:</b>	
<p>Conjunto de clases agrupadas en componentes que constituyen una herramienta de Mapeo objeto/relacional ó ORM de código abierto (Object Relational Mapping) y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada permite generar Bases de Datos en cualquiera de los entornos soportados: Oracle, PostgreSQL, DB2, MySQL, entre otras.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 12. Paquete Framework JPA</p>	<p>Unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.</p>
<b>Descripción:</b>	
<p>Conjunto de clase agrupadas en componentes que constituyen la API de persistencia desarrollada para</p>	

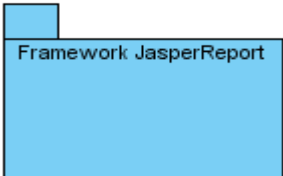
la plataforma Java EE e incluida en el estándar EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes de software EJB. Permite unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.

Nombre:	Propósito:		
<div style="text-align: center; margin-bottom: 10px;">  </div> <p>Figura 13. Clase SEAM Servlet</p>	<p>Proveer la interacción de la capa de presentación con la de negocio.</p> <tr style="background-color: #e6f2ff;"> <th style="padding: 5px;">Observaciones:</th> </tr> <td style="padding: 10px; vertical-align: top;"> <p>No se encuentra en ninguna capa en específico, sino que se encuentra ubicado entre la capa de Presentación y la de Negocio.</p> </td>	Observaciones:	<p>No se encuentra en ninguna capa en específico, sino que se encuentra ubicado entre la capa de Presentación y la de Negocio.</p>
Observaciones:			

**Descripción:**  
 Es el controlador de SEAM que capta las peticiones derivadas de la interacción del usuario después de interactuar con el Servlet de JSF. Enrutar las peticiones hacia los Beans que posibilitarán darle respuesta a la petición solicitada. Interviene en la integración de las capas de presentación y negocio.


Nombre:	Propósito:
<div style="text-align: center; margin-bottom: 10px;">  </div> <p>Figura 14. Clase CRExportar</p>	<p>Proveer las funcionalidades para exportar a diferentes formatos.</p>


**Descripción:**  
 Permite realizar las funciones para exportar a diferentes tipos de archivos mediante el uso de las librerías del Framework JasperReport.

Nombre:	Propósito:
<div style="text-align: center; margin-bottom: 10px;">  </div> <p>Figura 15. Paquete framework</p>	<p>Añadir características de generación de reportes a aplicaciones java.</p>

## CAPÍTULO 3. DISEÑO DEL SISTEMA


JasperReport.	
<b>Descripción:</b>	
Es una librería de clases de Java de código abierto desarrollada para facilitar el agregar capacidades de reporte a las aplicaciones Java. Permite realizar reportes de código abierto que tiene como función el llevar documentos ricos en contenido a la pantalla, a la impresora, o a archivos PDF, HTML, XLS, CSV y XML.	

Capa de Presentación	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>ListadoTrayectoriaSocial.siaps</b></p> <p>Figura 16. Clase ListadoTrayectoriaSocial.siaps</p>	Proveer la interacción con el usuario.
<b>Descripción:</b>	
La clase ListadoTrayectoriaSocial.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá listar todos los datos necesarios para crear el reporte y que todo quede correctamente registrado. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.	
<b>Nombre:</b>	<b>Propósito:</b>

 <p><b>frmsiaps</b></p> <p>Figura 17. frmsiaps</p>	<p>Enviar los datos a las páginas servidoras.</p>
---	---

**Descripción:**

La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente la cual posibilita ver el reporte de las Trayectorias Sociales. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$\_POST.

Capa de Negocio	
Nombre:	Propósito:
 <p><b>CLTrayectoriaSocial.java</b></p> <p>Figura 18.</p>	<p>Proveer una respuesta a las peticiones realizadas en la vista.</p>

**Descripción:**

La clase CLTrayectoriaSocial.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de mostrar los datos del reporte de trayectoria social solicitados desde la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

<b>Capa de Datos</b>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="margin: 0;"><b>tb_trayectoria_social</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-id_historia_social</li> <li>-fecha_historia_social</li> <li>-tratamiento_social</li> <li>-descripcion_paciente</li> <li>-problematika_social</li> <li>-id_trabajador_social</li> <li>-id_caso_reportado</li> <li>-id_motivo_reporte</li> <li>-antecedentes</li> </ul> </div> <p>Figura 19. Tb_trayectoria_social</p>	<p>Proveer el mapeo con la base de datos.</p>
<b>Descripción:</b>	
<p>La clase <code>tb_trayectoria_social.java</code> es una clase que se ejecuta del lado del servidor. En ella se almacenan todos los datos referentes a la trayectoria social de un paciente. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="margin: 0;"><b>tb_historia_social</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-id_historia_social</li> <li>-tratamiento_social</li> <li>-antecedentes</li> <li>-problematika_social</li> </ul> </div> <p>Figura 20. Tb_historia_social</p>	<p>Proveer el mapeo con la base de datos.</p>
<b>Descripción:</b>	
<p>La clase <code>tb_historia_social.java</code> es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos referentes a la historia social de un paciente. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y</p>	

JPA.	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid black; background-color: #e6f2ff; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>tb_pacientes</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-fecha_nacimiento</li> <li>-nombre_paciente</li> <li>-primer_apellido</li> <li>-segundo_apellido</li> <li>-ci</li> </ul> </div> <p>Figura 21. Tb_pacientes</p>	<p>Proveer el mapeo con la base de datos.</p>
<b>Descripción:</b>	
<p>La clase <code>tb_pacientes.java</code> es una clase que se ejecuta del lado del servidor. En ella se almacena los datos de los pacientes. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>	

En el capítulo se presentó la solución: se definieron los diagramas de clases del diseño, la descripción de los patrones utilizados así como la descripción de cada una de las clases del diseño con el objetivo de entender mejor el sistema, empleando para ello el Lenguaje unificado de Modelado (UML) y la herramienta case Visual Paradigm. Fue detallada la estructura del diseño, la cual complementa los requerimientos funcionales y no funcionales. Todo esto constituyó un paso fundamental para el desarrollo del flujo de Trabajo de Implementación.



### Capítulo 4. Implementación

Este capítulo es una secuencia del flujo de análisis y diseño. En el mismo son vistos los principales elementos de la implementación del sistema. También se presenta el Diagrama de Despliegue de la solución propuesta y se definen los elementos clave para la seguridad, el tratamiento de errores y los estilos de código a utilizar en la aplicación.

#### 4.1 Justificación de Integración con otros sistemas

El módulo Estadísticas se relaciona con los restantes módulos del Sistema Integral para la Atención Primaria de Salud alas SIAPS, teniendo una estrecha relación con los módulos Medios de Diagnósticos, Adulto Mayor e Higiene y Epidemiología, logrando una comunicación a nivel de Bases de Datos. El Módulo Configuración le permite a Estadísticas de Medios de Diagnósticos, Adulto Mayor e Higiene y Epidemiología obtener la información relacionada con los nomencladores y departamentos para la realización de los reportes, además de suministrar los datos necesarios de los pacientes y las entidades donde pertenecen.

#### 4.2 Propuesta de seguridad del módulo

Debido al correcto funcionamiento, la importancia y confidencialidad de los procesos que se llevan a cabo en el departamento de Estadísticas, surge la necesidad de que estos sean realizados con la calidad requerida y que el acceso a la información de los pacientes atendidos solo sea permisible para las personas que se encuentren acreditadas, todo esto trae consigo que se tengan en cuenta una serie de requisitos de seguridad de modo que no se pueda afectar el funcionamiento del sistema.

Para que el sistema sea seguro se llevará a cabo un control entre los usuarios y sus contraseñas, permitiendo el acceso según el nivel al que pertenece el usuario, logrando así la visibilidad sólo a las áreas establecidas, de acorde a la función que realizan. Las contraseñas solo podrán ser cambiadas por el usuario o por el administrador del sistema.

Para lograr la fidelidad de los datos, todo el intercambio entre el sistema y otros sistemas que soliciten información desde cualquier entidad, se realizará de forma cifrada eliminando posibilidades de acceso o modificación de la misma.

### 4.3 Implementación

Con los resultados obtenidos del Modelo de Diseño se comienza a desarrollar el flujo de trabajo de Implementación. En este flujo de trabajo se describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue, también se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. Uno de los principales propósitos de este flujo de trabajo consiste en desarrollar la arquitectura y el sistema como un todo.

### 4.4 Diagrama de Despliegue.

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos, además proveen una vista arquitectónica de alto nivel del sistema. También ayudan a visualizar el camino de la implementación y muestran cómo y dónde se desplegará el sistema. Los nodos representan recursos de cómputos entre los cuales existen relaciones que representan el medio de comunicación. Permiten comprender la correspondencia entre la arquitectura de software y la arquitectura de hardware.

De manera general el diagrama de despliegue del Módulo de Estadísticas quedaría de la siguiente forma:

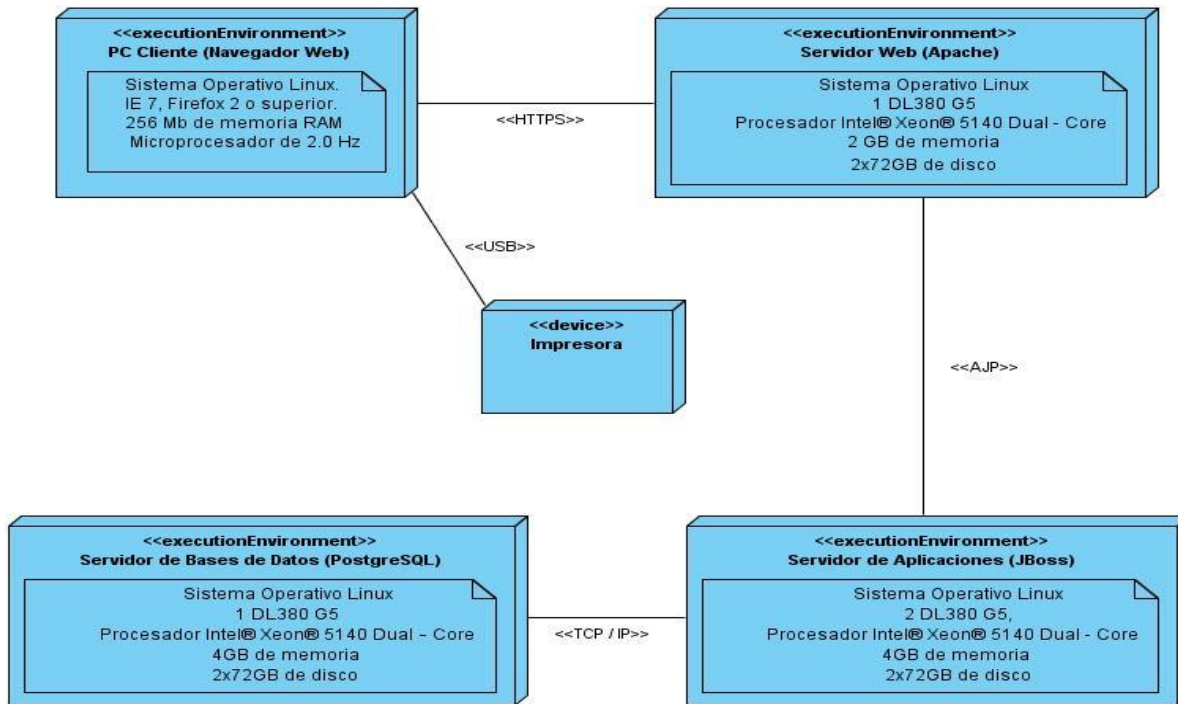


Diagrama de Despliegue.

## 4.5 Estándares de codificación y tratamiento de excepciones.

Con el objetivo de que exista uniformidad entre las aplicaciones integradas al Sistema Integral para la Atención Primaria de Salud alas SIAPS se han definido una serie de estándares para el diseño así como para la codificación, además se determinó como se realizará el tratamiento de errores para que los códigos fuentes de las aplicaciones y los mensajes que se emitan mantengan una igualdad.

### 4.5.1 Estándares de codificación.

Para la implementación del sistema propuesto se utilizaron varios estándares de codificación como son:

**Idioma:** Se debe utilizar como idioma el español, las palabras no se acentuarán.

**Comentarios, líneas y espacios en blanco.**

Las líneas en blanco brindan mejor facilidad de lectura separando secciones de código que están lógicamente relacionadas.

Todos los ficheros fuente comienzan con un comentario en el que se lista el nombre de la clase, información de la versión, fecha, y copyright.

Se deben usar siempre dos líneas en blanco en las siguientes circunstancias:

- ❖ Entre las secciones de un fichero fuente
- ❖ Entre las definiciones de clases e interfaces.

Se debe usar siempre una línea en blanco en las siguientes circunstancias:

- ❖ Entre métodos
- ❖ Entre las variables locales de un método y su primera sentencia
- ❖ Antes de un comentario de bloque o de un comentario de una línea.
- ❖ Entre las distintas secciones lógicas de un método para facilitar la lectura.

Se deben usar espacios en blanco en las siguientes circunstancias:

Una palabra clave del lenguaje seguida por un paréntesis debe separarse por un espacio.

### **Indentación y longitud de la línea.**

Se deben emplear cuatro espacios como unidad de indentación. La construcción exacta de la indentación (espacios en blanco contra tabuladores) no se especifica. Los tabuladores deben ser exactamente cada 8 espacios.

Evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

### **Variables, constantes, clases y métodos.**

Todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres subguión "\_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje.

Los nombres de las variables deben ser cortos pero con significado. La elección del nombre de una variable debe ser un mnemónico, designado para indicar a un observador casual su función. Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.

Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión ("\_"). (Las constantes ANSI se deben evitar, para facilitar su depuración.)

Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Mantener los nombres de las clases, simples y descriptivos. Usar palabras completas, evitar acrónimos y abreviaturas.

Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula.

### **4.5.2 Tratamiento de excepciones.**

Los sucesos que ocurren durante la ejecución de un programa dificultando el flujo normal de las sentencias se les llaman excepciones, estas son situaciones anómalas que requieren un tratamiento especial, no tienen por qué ser errores. El tratamiento de excepciones es una estructura de control de los lenguajes de programación diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla.

En el sistema propuesto se utilizan todas las facilidades que brinda la plataforma para el tratamiento de excepciones. Para cada fragmento de código donde se espere una situación anómala, se definen las excepciones correspondientes para luego ser tratadas evitando la interrupción del sistema. El uso de diferentes tecnologías y la integración que existe entre ellas, permiten capturar y controlar posibles situaciones desde diferentes puntos de la aplicación.

En este capítulo se mostraron los resultados de la etapa de construcción del sistema. Se realizaron los distintos modelos del flujo de trabajo Implementación, describiendo la distribución física del sistema y sus componentes, así como las distintas pautas seguidas para el diseño de interfaz y los métodos usados para el tratamiento de errores.

### Conclusiones

La realización del presente trabajo ha posibilitado cumplir con los objetivos propuestos, por lo que se pueden plantear las siguientes conclusiones:

Los procesos de negocio llevados a cabo en los departamentos de estadísticas en la APS demostraron la necesidad de desarrollar un sistema informático para la gestión de la información. A través de estos procesos se identificaron las funcionalidades previas a tener en cuenta para el desarrollo del sistema propuesto. La aplicación web desarrollada facilita la gestión de la información de los servicios estadísticos en la Atención Primaria de Salud, genera los reportes de las áreas Medios de Diagnósticos, Adulto Mayor e Higiene y Epidemiología y se garantizó un ambiente de desarrollo que permite la creación de un sistema robusto y flexible a partir de la utilización de la arquitectura definida.

### Recomendaciones

Para el buen desempeño y puesta en marcha de la aplicación se hacen las siguientes recomendaciones:

- ❖ Desplegar la aplicación en los policlínicos, consultorios y demás centros de la Atención Primaria de Salud, para así facilitar el trabajo de los técnicos de estadísticas de estas instituciones.
- ❖ Trabajar en la implementación de los reportes estadísticos en el resto de los módulos, e implementar las funcionalidades que permitan la generación gráfica de los mismos.

## Referencias Bibliográficas

1. Historia de la Estadística. [En línea] [Citado el: 20 de octubre de 2010.] <http://www.profesorenlinea.cl/matematica/estadisticaHistoria.htm>
2. Bioestadística, Dpto. Medicina Preventiva y salud Pública. [En línea] [Citado el: 20 de octubre de 2010.] <http://www.bioestadistica.uma.es/libro/node3.htm>
3. Ibidem a la anterior.
4. **Gran Álvarez, Dra. Miriam Alicia y Catañeda Abascal, Dra. Ileana.** *ESTADÍSTICAS SANITARIAS. TEMAS DOCENTES.* Ciudad de la Habana 2000.
5. **Gómez Fernández, Vanesa y Rivero Alemán, Rubén.** *Estadísticas Descriptivas del Registro de Fallecidos.* La Habana 2009.
6. Ibidem a la anterior.
7. Ídem a la anterior.
8. UNICEF. [En línea] [Citado el: 6 de diciembre de 2010.] [http://www.unicef.org/spanish/statistics/index\\_24300.html](http://www.unicef.org/spanish/statistics/index_24300.html)
9. INEbase. Instituto Nacional de Estadísticas. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.ine.es/inebmenu/queesinebase.htm>
10. Servicio de salud Metropolitano Norte. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.ssmn.cl/Estadistica/menuestadistica.htm>
11. Arquitectura Basada en Componentes. [En línea] [Citado el: 18 de diciembre de 2010.] <http://www.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes>.
12. Beneficios de la Tecnología JavaServer Faces. [En línea] [Citado el: 18 de diciembre de 2010.] <http://www.jcp.org/en/jsr/detail?id=152>.
13. **Sánchez Suárez, José Manuel.** *Introducción a RichFaces.*



## REFERENCIAS BIBLIOGRÁFICAS

---

14. **López Hernández, Yurien y Crombet Pérez, Yilianne Arias.** *Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS.* La Habana : s.n., 2010.
15. **Alonso Ramos, Juan.** Introducción a Ajax4jsf. [En línea] [Citado el: 20 de diciembre de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
16. **José Alberto.** [En línea] [Citado el: 20 de diciembre de 2010.] <http://www.versionzero.com/noticia/210/visual-paradigm-for-uml>.
17. Programas.com. [En línea] [Citado el: 28 de enero de 2011.] <http://toad-data-modeler.programas-gratis.net/>.
18. Programas.com. [En línea] [Citado el: 28 de enero de 2011.] <http://toad-data-modeler.programas-gratis.net/>.
19. **Crombet Pérez, Yilianne Arias y López Hernández, Yurien.** *Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS.* La Habana : s.n., 2010.
20. **Sánchez Fornaris, Maite y Alcantara Rabí, Elvia Dayanis y Hernández Luque, Eylin.** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. [En línea] 15 de Febrero de 2011. [http://vinculando.org/articulos/sociedad\\_america\\_latina/propuesta\\_guia\\_de\\_medidas\\_para\\_evaluacion\\_sistemas\\_informacion.html](http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html).
21. Ibídem a la anterior.
22. **Morató Simón, Julia.** *0116\_RNeg\_Reglas del Negocio. Sistema Integral para la Atención Primaria de Salud (SIAPS). Módulo Estadísticas.*

## Bibliografía

1. **Alonso Ramos, Juan.** Introducción a Ajax4jsf. [En línea] [Citado el: 20 de diciembre de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
2. Arquitectura Basada en Componentes. [En línea] [Citado el: 18 de diciembre de 2010.] <http://www.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes>.
3. **Autores, Colectivo.** cpicm HABANA. *Informática en la Salud Pública Cubana*. [En línea] 2007. [Citado el: 12 de octubre de 2009.] [http://www.cpicmha.sld.cu/hab/vol13\\_2\\_07/hab17207.htm](http://www.cpicmha.sld.cu/hab/vol13_2_07/hab17207.htm).
4. Beneficios de la Tecnología JavaServer Faces. [En línea] [Citado el: 18 de diciembre de 2010.] <http://www.icp.org/en/jsr/detail?id=152>.
5. Bioestadística, Dpto. Medicina Preventiva y salud Publica. [En línea] [Citado el: 20 de octubre de 2010.] <http://www.bioestadistica.uma.es/libro/node3.htm>
6. **Crombet Pérez, Yilianne Arias y López Hernández, Yurien.** *Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS. La Habana : s.n., 2010.*
7. Desarrollo de Software basado en Componente. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.lambdasi.com.ar/textocomp.asp>
8. **Gómez Fernández, Vanesa y Rivero Alemán, Rubén.** *Estadísticas Descriptivas del Registro de Fallecidos. La Habana 2009.*
9. **Gran Álvarez, Dra. Miriam Alicia y Catañeda Abascal, Dra. Ileana.** *ESTADÍSTICAS SANITARIAS. TEMAS DOCENTES. Ciudad de la Habana 2000.*
10. Historia de la Estadística. [En línea] [Citado el: 20 de octubre de 2010.] <http://www.profesorenlinea.cl/matematica/estadisticaHistoria.htm>
11. INEbase. Instituto Nacional de Estadísticas. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.ine.es/inebmenu/queesinebase.htm>

12. INFOMED. Aspectos Generales del Sistema Nacional de Salud. Infomed. Portal de Salud de Cuba. [En línea] [Citado el: 6 de Diciembre del 2010] Disponible en: [http://www.sld.cu/sistema\\_de\\_salud/aspectos.html](http://www.sld.cu/sistema_de_salud/aspectos.html).
13. **José Alberto.** [En línea] [Citado el: 20 de diciembre de 2010.] <http://www.versionero.com/noticia/210/visual-paradigm-for-uml>.
14. **López Hernández, Yurien y Crombet Pérez, Yilianne Arias.** *Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS.* La Habana : s.n., 2010.
15. Oficina Nacional de Estadísticas. [En línea] [Citado el: 6 de Diciembre del 2010] Disponible en: <http://www.one.cu/>
16. Programas.com. [En línea] [Citado el: 28 de enero de 2011.] <http://toad-data-modeler.programas-gratis.net/>).
17. Programas.com. [En línea] [Citado el: 28 de enero de 2011.] <http://toad-data-modeler.programas-gratis.net/>).
18. **Ramos Delgado, Ariel y Ledo Vidal, María.** Scielo Cuba. *Estrategias de Informatización del Sistema Nacional de Salud.* [En línea] 2006. [Citado el: 18 de octubre de 2009.] [http://scielo.sld.cu/scielo.php?pid=S0864-34662006000300015&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S0864-34662006000300015&script=sci_arttext).
19. **Sánchez Fornaris, Maite y Alcantara Rabí, Elvia Dayanis y Hernández Luque, Eyllin.** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. [En línea] 15 de Febrero de 2011. [http://vinculando.org/articulos/sociedad\\_america\\_latina/propuesta\\_guia\\_de\\_medidas\\_para\\_evaluacion\\_sistemas\\_informacion.html](http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html).
20. **Sánchez Suárez, José Manuel.** *Introducción a RichFaces.*
21. Servicio de salud Metropolitano Norte. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.ssmn.cl/Estadistica/menuestadistica.htm>
22. Todo sobre estadística. [En línea] [Citado el: 20 de Octubre del 2010] Disponible en: <http://todoestadistica.blogspot.com/>

23. UNICEF. [En línea] [Citado el: 6 de diciembre de 2010.]  
[http://www.unicef.org/spanish/statistics/index\\_24300.html](http://www.unicef.org/spanish/statistics/index_24300.html)

## Glosario de Términos

**Adulto Mayor:** Es la atención que se les brinda a las personas de la tercera edad, donde tienen casas de abuelos, hogares de ancianos, entre otros, en los cuales se les dedica la atención médica necesaria.

Atención Primaria de Salud (**APS**): Nivel asistencial que constituye la puerta de entrada del paciente al Sistema Nacional de Salud, donde debe darse solución alrededor del 80% de los problemas que afectan a la población. En este nivel se realizan acciones educativas, curativas y de rehabilitación.

**Electrocardiograma:** es el registro grafico de las variaciones de potencial eléctrico de la actividad del corazón (fibras miocárdicas), en un tiempo determinado.

**Endoscopia:** es un procedimiento médico que utiliza un sistema óptico para poder ver en el interior del tubo digestivo.

**Entidad:** Centro médico de atención, dígase policlínico, médico de familia.

Equipo Básico de Salud (**EBS**): Binomio conformado por el médico y enfermera de la familia, que atiende una población geográficamente determinada, que puede estar ubicado en la comunidad, centros laborales o educacionales.

**Exámenes:** Pruebas que se realizan para encontrar una enfermedad antes de que comiencen los síntomas.

Grupo dispensarización: Grupo a que corresponde el individuo de acuerdo a la clasificación de Dispensarización en aparentemente sano, con riesgos, enfermo, deficiente o discapacitado.

**Higiene y epidemiología:** La Epidemiología es la ciencia que estudia la frecuencia de aparición de enfermedades que afecten a comunidades humanas. Enfermedades que en la mayoría de los casos se conocen en forma de brotes, donde se depende de la higiene que se tenga al manipular alimentos, el cuidado con el medio ambiente, con la basura que se vierte diariamente en las calles, las bahías y los gases que las fabricas sin distinción dispersan en la atmósfera.

Historia de Salud Familiar (**HSF**): Documento utilizado en los Consultorios del médico y la enfermera de la familia donde se plasma información relacionada con las características biológicas, socioeconómicas e higiénicas de la familia y sus integrantes.

# GLOSARIO DE TÉRMINOS

---

**Hoja de Cargo (HC):** Documento que almacena la relación de pacientes enfermos que se hayan presentado a la consulta ese día.

**Informática:** Disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

**Informatizar:** Proceso de aplicar sistemas o equipos informáticos al tratamiento de la información.

**Laboratorio Clínico:** Es un lugar donde la gente va a hacerse estudios de rutina para saber su estado de salud.

**Medios de Diagnósticos:** Es un instrumento, procedimiento o conjunto de procedimientos a través de los cuales se llega a determinar el padecimiento de un paciente, clasificándose en diferentes especialidades diagnósticas. Por ejemplo Laboratorio Clínico, Endoscopia, Optometría, Electrocardiograma, entre otros.

**Optometría:** es la ciencia que estudia el sistema visual, sus alteraciones no patológicas y los tratamientos ópticos y optométricos así como las normas de salud e higiene visual.

Requisito:

- (1) Una condición o capacidad necesitada por el usuario para resolver un problema o lograr un objetivo.
- (2) Una condición o capacidad que debe poseer un producto o componente de producto para satisfacer un contrato, estándar, especificación, u otros documentos obligatorios formales.
- (3) Una representación documentada de una condición o capacidad como en (1) o (2).

**Sistema Nacional de Salud (SNS):** La Organización Mundial de la Salud (OMS) define al Sistema Nacional de Salud como: "Un sistema que engloba todas las organizaciones, instituciones y recursos cuyo principal objetivo es llevar a cabo actividades encaminadas a mejorar la salud". El sistema nacional de salud en la actualidad se encuentra dividido en tres niveles los cuales son: Nivel de Atención Primaria de salud, Nivel de Atención Secundaria de salud y Nivel de atención Terciaria de salud.