

# Universidad de las Ciencias Informáticas

## Facultad 7



### **Titulo: Diseño y Servicios Web para el Registro de Población de la Atención Primaria del Sistema de Información para la Salud.**

**Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas**

**Autores:** Yusniel Brocard Delfino  
Yoiler Joaquín Frometa Moreno  
Maikel Suárez Corrales

**Tutores:** Ing. Alberto Acuña Sánchez  
Ing. Lourdes Escalona Peral

Ciudad de La Habana, Julio de 2007

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Yusniel Brocard Delfino  
Firma del Autor

\_\_\_\_\_  
Yoiler Joaquín Frometa Moreno  
Firma del Autor

\_\_\_\_\_  
Maikel Suárez Corrales  
Firma del Autor

\_\_\_\_\_  
Alberto Acuña Sánchez  
Firma del Tutor

\_\_\_\_\_  
Lourdes Escalona Peral  
Firma de la Tutora

## DATOS DE CONTACTOS

**Ing. Alberto Acuña Sánchez:** Graduado en la especialidad de Ingeniería Informática en el año 1996. Posee la Categoría Docente de Profesor Auxiliar y cursa la maestría de Gestión de Proyectos Informáticos. Imparte la asignatura de Ingeniería de Software en la Facultad 7 desde el curso 2004-2005. Ha presentado ponencias y trabajos en eventos científicos obteniendo diferentes reconocimientos y premios. Es Jefe de Proyectos dentro del equipo APS para el desarrollo de aplicaciones para la salud en la Empresa Softel.

Correo electrónico: chino@softel.cu

**Lourdes Escalona Peral:** Profesor graduado de Ingeniero Informático en el año 2004 en la Universidad de Holguín. Ha impartido las asignaturas ISW 1 y 2 y Seminario de Tesis. Posee la categoría docente de Instructor y cursa la maestría de Ciencias de la Computación en la UCI. Jefe de Dpto. de la Especialidad de la Facultad 7.

Correo electrónico: lescalonap@uci.cu

**Mirna Cabrera Hernández:** Graduado de Ing. Sistema Automatizado de Dirección Técnico Económico (SAD) en el año 1986 en el ISPJAE. Posee categoría docente de Profesor Auxiliar y cursa la maestría de Gestión de Proyectos Informáticos. Ha impartido la asignatura Gestión de Software en la Facultad 7 desde el curso 2005-2006. Ha presentado ponencias en eventos científicos nacionales e internacionales. Se desempeña como Líder del Proyecto APS en la Empresa Softel.

Correo electrónico: mirna@softel.cu

## **AGRADECIMIENTOS**

Ante todo, agradecerle a nuestro Comandante en Jefe Fidel Castro Ruz, por tener siempre esa visión futura y confiar en las nuevas generaciones. A la Revolución, a la Universidad de las Ciencias Informáticas y a todos los profesores que de una forma u otra han colaborado con nuestro desarrollo científico y social. A todos los especialistas de la Empresa Softel, en especial a Mirna por ser profesora, jefa y madre a la vez.

A nuestros padres y familiares, ya que sin su apoyo, confianza y amor nada de esto fuera posible.

A Johander por la valiosa ayuda que nos dio, así como a “la tenca” por atendernos, siempre que lo necesitábamos, al “kiki”, a Luis, el “F1 con patas” y en fin a todos nuestros compañeros de estudio y trabajo con los que diariamente convivimos.

## DEDICATORIA

*En especial a mis padres Fernando Brocard y Magdeline Delfino, por darme ese cariño inmenso y guiarme siempre en cada paso de mi vida, a mi hermano para que tome el ejemplo, a mi abuelito Brongel por hacerme conciencia por ese camino del saber, al igual que al resto de mis queridos abuelos y de mi familia porque todos de una manera u otra han aportado un granito en el desarrollo de mis cinco años, a Mirna Cabrera por enseñarnos a ser responsables y organizados, a mis compañeros de siempre, al grupo “Lost”, Andro, Luis, Kiki, Yoiler y Henry, al piquete del laboratorio 6 de la IP por compartir y aprender de ellos, a mi compañeros de tesis, a mi cosi por ayudarme y quererme mucho en estos últimos años, a Yaque, Liuris y Yaimi por ser buenas amigas también. En fin a todos porque todos forman parte de mi vida...*

*Yusniel Brocard*

*Este trabajo va dedicado a todos aquellos, que de una forma u otra han contribuido a su realización. A todos mis familiares y en especial a mis padres María Teresa y Joaquín Guillermo, espero haberles realizado una parte de sus sueños. Me gustaría que le sirva de guía a mi hermanita del alma Yoleidis y a Pirri que ellos son los que siguen. No puede faltar el piquete del laboratorio 6 (“La crème de la crème”) y por último a Danay por no perder la cordura en estos días de máximo estrés.*

*Yoiler Frometa*

*Dedico este trabajo a todas aquellas personas que han tenido algo que ver de cierta forma con el mismo. A mi mamá, mi papá y familia en general, que siempre estuvieron pendiente de mis actos, a mi hermanita Mariuski. A Mirna por ser un gran ejemplo de responsabilidad para mí, a mis amigos todos que nos ayudaron en el trabajo, a Yunaisi por haberme brindado su cariño y apoyo, a todos en general. A todos, Muchas Gracias.*

*Maikel Suárez*

*"La base de un desarrollo impetuoso en los años futuros debe basarse en una ciencia cada vez más desarrollada"*

*Ché Guevara.*

# RESUMEN

Actualmente, no existe una solución informática que permita la captación y procesamiento de la información relacionada con los pacientes y las familias atendidas por el Equipo Básico de Salud (EBS) en cada área de salud.

El presente trabajo tiene como objetivo diseñar e implementar una aplicación Web que facilite al Equipo Básico de Salud la gestión de la información asociada a la población que atiende y así apoyar en la organización de las funciones y actividades que debe desarrollar el mismo en su comunidad, centradas en torno al proceso de dispensarización.

Para su desarrollo se modelaron los flujos de trabajo de Diseño e Implementación, teniendo en cuenta el proceso de desarrollo unificado (RUP) y el lenguaje de modelado visual (UML). Además se han utilizado las tecnologías y herramientas relacionadas con servidor Web Apache, MySql, PHP, XML, XSL, todas siguiendo las políticas trazadas por el MINSAP de software libre y código abierto. Se aplicó la arquitectura orientada a servicio y basada en componentes.

El desarrollo de la aplicación permitió reducir los tiempos de desarrollo, gracias a la reutilización del código y al alto nivel de integración que tiene con otros componentes, logrando así mantener toda la información centralizada y disponible para el Equipo Básico de Salud. Permite realizar la evaluación de salud familiar, así como la intervención planificada e integral de la salud de los individuos y familias, esto facilitará el proceso de toma de decisiones en los diferentes niveles de dirección.

**Palabras Claves:** Sistema Nacional de Salud, Sistema de Información para la Salud, Registro de Población, Historia de Salud Familiar, Informatización.



# TABLA DE CONTENIDO

RESUMEN.....	I
INTRODUCCIÓN.....	1
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA .....</b>	<b>6</b>
1.1 Introducción.....	6
1.2 Sistema Nacional de Salud. ....	6
1.2.1 Informatización del Sistema Nacional de Salud. ....	7
1.2.2 Informatización de la Atención Primaria de Salud. ....	10
1.2.2.1 Proceso de Dispensarización. ....	11
1.2.3 Registro Informatizado de Salud (RIS).....	14
1.2.4 Solución integral propuesta para la Informatización del SNS.....	14
1.3 Sistemas automatizados existentes vinculados al campo de acción.....	17
1.4 Políticas para el Desarrollo de la Informatización del SNS.....	19
1.5 Tendencias y Tecnologías Actuales a Considerar y Utilizar. ....	20
1.5.1 Internet. Funcionamiento. ....	20
1.5.2 Aplicaciones Web. ....	20
1.5.3 XML/Web Services. Service Oriented Architecture. ....	21
1.5.4 Entorno Distribuidos. Modelo Cliente Servidor.....	21
1.5.5 Modelo Cliente Servidor de Tres Capas.....	22
1.5.6 Servidor Web Apache.....	23
1.5.7 Arquitectura Basada en Componentes (CBA). ....	24
1.6 Patrones de Diseño. ....	25
1.6.1 Modelo Vista Controlador (MVC). ....	25
1.6.2 Single Sign On (SSO). ....	26
1.7 Lenguajes de Programación Web.....	26
1.8 Sistemas de Gestión de Bases de Datos (SGBD).....	30
1.9 Desarrollo basado en RUP bajo la herramienta Rational Rose. ....	33
1.10 UML (Unified Modeling Language) .....	33

1.11 Rational Rose.....	33
1.12 Plataforma de Servicio (PLASER).....	35
1.13 Herramientas a utilizar. ....	36
1.14 Conclusiones. ....	36
<b>CAPÍTULO 2 DISEÑO DEL SISTEMA.....</b>	<b>37</b>
2.1 Introducción.....	37
2.2 Modelo del Diseño. ....	37
2.2.1 Justificación del Uso de Patrones.....	37
2.2.1.1 Modelo – Vista – Controlador.....	38
2.2.1.2 Arquitectura en 3 Capas.....	39
2.2.1.3 Arquitectura Basada en Componentes (CBA). ....	39
2.2.1.4 Arquitectura Orientada a Servicios (SOA). ....	40
2.2.1.5 Fachada. ....	41
2.2.1.6 Proxy.....	42
2.2.2 Estructura del Diseño.....	43
2.2.3 Diagramas de Clases del Diseño.....	43
2.3 Conclusiones. ....	55
<b>CAPÍTULO 3 IMPLEMENTACIÓN DEL SISTEMA.....</b>	<b>56</b>
3.1 Introducción.....	56
3.2 Justificación de Integración con otros Módulos. ....	56
3.2.1 Componente de Seguridad (SAAA). ....	57
3.2.2 Registro de Áreas de Salud (RAS).....	57
3.2.3 Registro de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud (RCIE).....	57
3.2.4 Registro de Problemas de Salud de la Atención Primaria (RPSAP).....	58
3.2.5 Registro de Ciudadano (RC). ....	58
3.3 Modelo de Implementación.....	58
3.3.1 Diagrama de Componentes.....	58
3.3.2 Diagrama de Despliegue. ....	59

3.4 Descripción de los Métodos más complejos.....	60
3.5 Estándares de diseño, codificación y tratamiento de errores.....	62
3.5.1 Estándares de Diseño. ....	62
3.5.2 Tratamientos de Errores.....	65
3.5.2 Estándares de codificación.....	67
3.6 Conclusiones. ....	69
CONCLUSIONES .....	70
RECOMENDACIONES.....	71
REFERENCIAS BIBLIOGRÁFICAS .....	72
BIBLIOGRAFÍA.....	75
ANEXOS.....	78
GLOSARIO DE TÉRMINOS.....	81

# INTRODUCCIÓN

***“La sociedad se conmueve ante la noticia del secuestro o asesinato de una criatura, pero permanece criminalmente indiferente ante el asesinato en masa que se comete con miles y miles de niños que mueren todos los años por falta de recursos...”***

***Fidel Castro***

Antes del triunfo de la Revolución, la atención médica y hospitalaria se caracterizaba por el predominio de servicios de carácter privado y mutualista. Esta modalidad en los servicios de salud impedía el acceso a las personas de más bajos ingresos, las que contaban como única opción con las Casas de Socorro que atendían principalmente los casos de urgencia. Desde el propio triunfo revolucionario se comenzó a trabajar por la creación del Sistema Nacional de Salud (SNS) que llevó la acción del trabajador de la salud a los lugares más apartados. [1]

Las profundas transformaciones sociales, políticas y económicas que siguieron al triunfo de la Revolución en 1959 permitieron enfrentar los problemas sanitarios existentes y elevar sustancialmente los niveles de vida y salud de la población. Los servicios de salud se distribuyeron a lo largo y ancho del país y se pusieron al alcance de toda la población a la que se le facilitó disponer de atención médica preventivo-curativa, oportuna, eficaz y sin limitación económica dado que los mismos le resultaban gratuitos. [2]

Se produce la primera reforma del sector salud en los años 60, cuyas principales medidas fueron: Fundación del Sistema Nacional de Salud, Implementación del Servicio Médico Rural y Creación de las Áreas de Salud y los Policlínicos.

Con la creación del Sistema Nacional de Salud y su principio de que la salud es derecho del pueblo y responsabilidad del Estado, se garantizó la accesibilidad legal y económica, de igual forma con la alfabetización, el aumento del nivel de escolaridad de la población y la educación para la salud del pueblo, se propició la accesibilidad cultural al sistema. La atención médica de carácter gratuito cubrió a toda la población, lo que trajo consigo una mejora notable de la situación del estado de salud de la población y un fortalecimiento del sistema de salud, su nivel primario de atención y de la medicina familiar.

Como respuesta a la necesidad de continuar elevando el nivel de salud de la población se realiza la más importante reforma del sector salud en los años 80: la creación del Modelo del Médico y Enfermera de la Familia como forma de organización de la Atención Primaria, responsabilizado con la atención del individuo, la familia y la comunidad, mostrando un gran impacto debido a la disminución en los servicios hospitalarios de los ingresos, las consultas de urgencias, las intervenciones quirúrgicas y los medios auxiliares. [3]

La Revolución, enmarcada en el proceso actual de la “Batalla de Ideas” se ha dado a la tarea de informatizar la sociedad cubana, donde conjuntamente con el Ministerio de Salud Pública (MINSAP), ha trazado estrategias para la reorientación del SNS hacia la Atención Primaria, fundamentalmente hacia los Médicos y Enfermeras de la Familia, dúo que da vida al Equipo Básico de Salud (EBS), piedra angular de la Atención Primaria de Salud (APS).

Para llevar a cabo el Proceso de Informatización del SNS abarcando a la APS y al policlínico como eje fundamental, fue encomendada esta tarea por la dirección del MINSAP y el Ministerio de Informática y Comunicaciones (MIC), a la Empresa SOFTEL, empresa cubana dedicada a la ejecución de soluciones informáticas para la salud, que en conjunto con la Universidad de las Ciencias Informáticas (UCI) y Médicos Especialistas en Medicina General Integral en calidad de expertos funcionales, tienen la misión en el marco del Proyecto APS de elaborar un producto de software que facilite la gestión de la información y la toma de decisiones en este nivel de atención.

Las Áreas de Salud en Cuba están constituidas por el área geográfica a la que presta sus servicios una unidad de salud que contemple el Programa de Trabajo del Médico y la Enfermera de la Familia. En estas unidades (pueden ser policlínicos u hospitales rurales) se atiende a la mayoría de la población en el nivel de atención primaria. Las Áreas de Salud están relacionadas con las Localidades que atienden, y en dependencia de su ubicación geográfica con los Hospitales Base que sirven de apoyo a la hora de remitir un paciente a un nivel de atención de salud secundario, todos están relacionados con las Poblaciones que se atienden en los EBS y que forman parte de los diferentes Grupos Básicos de Trabajo (GBT) del Área de Salud, así como con la composición o plantilla definida para los GBT y sus EBS, según la estructura organizativa propuesta por la Atención Primara de Salud.

El EBS gestiona periódicamente los datos del paciente y sus problemas de salud mediante la Historia de Salud Familiar, documento oficial que recoge como un todo los datos referentes a la familia, por ejemplo las condiciones materiales de vida familiar, índice de hacinamiento, entre otros, lo que le permite al médico realizar la planificación de las acciones que se llevarán a cabo para mantener un control del estado de salud de su población.

Teniendo en cuenta lo anteriormente expuesto y la necesidad de buscar una respuesta a las dificultades que frenan el desarrollo eficiente de las actividades del EBS, los esfuerzos estarán encaminados a resolver la siguiente **Situación Problemática**:

Durante el transcurso de los años se ha observado una notable mejora de los servicios médico-sanitarios del país, lo que ha repercutido positivamente en la calidad de vida de los habitantes; pero a la vez se han detectado problemas que dificultan sin duda alguna el control de la información que se genera en el proceso de atención a pacientes y familias.

Después de revisado e investigado el proceso actual de atención a la población se han detectado algunos problemas presentes en las actividades que desarrollan los EBS, que a continuación se tratan de exponer:

1. Dificultad para actualizar los constantes cambios de la Historia Salud Familiar (HSF), ya que el actual diseño del modelo estadístico vigente no lo contempla.
2. El diseño del modelo actual de la Historia de Salud Familiar no contempla la captación de información referente al movimiento poblacional (altas, bajas y sus causas), al Área de Salud determinada y al Grupo Básico de Trabajo.
3. En el modelo actual de la Historia de Salud Familiar no se concibe el desglose detallado de las Enfermedades, Factores de riesgos y Discapacidades para cada paciente, lo que dificulta conocer la Incidencia y Prevalencia de los Problemas de Salud.
4. Es imposible acceder a dicha información estando fuera del área del EBS a la cual pertenece la familia.
5. Toda la labor de gestión de la información de la HSF se realiza de forma manual, lo que trae como consecuencia omisiones, errores y en muchos casos la no actualización de la información.

6. La Historia de Salud Familiar (HSF) y el modelo de planificación de acciones de salud son documentos que se deterioran con el paso del tiempo provocando que se pierda la información que contienen.
7. Demoras en la actualización de la Historia Salud Familiar (HSF).

Según el criterio de los Médicos y Enfermeras de la Familia se puede plantear, que establecer el control de la información relacionada con los pacientes, así como el estado de cumplimiento de los diferentes programas, de forma manual, como se ha venido realizando hasta el momento, representa el empleo de gran parte del tiempo de la jornada laboral en estas tareas, que sin dudas pudiera ser destinado a labores asistenciales y de prevención.

Por otra parte, la labor del Médico de la Familia abarca disímiles actividades, sobre las que frecuentemente es necesario emitir informes y ofrecer datos estadísticos, para uso local o a nivel de instancias superiores. De modo similar el Análisis de la Situación de Salud (ASS), proceso básico en la atención primaria, que resume la situación de salud del área y refleja el impacto de las intervenciones del Médico de la Familia en la comunidad, requiere de los datos de la población que se atiende, todo lo cual conlleva a un procesamiento de la información que en ocasiones se torna engorroso y consume tiempo en su realización.

Por tanto el **Problema a resolver** será: ¿Cómo automatizar la gestión de la información de las Historias de Salud Familiar para solucionar los problemas existentes en los Equipos Básicos de Salud?

El **Objetivo general** es: Diseñar e Implementar una aplicación Web que facilite al Equipo Básico de Salud la gestión de la información asociada a la población que atiende.

El **Objeto de Estudio** lo constituye el proceso de gestión de la información en la Atención Primaria de Salud.

Como **Campo de acción** se define el proceso de gestión de la información de la Historia de Salud Familiar correspondiente a la población que atiende el Equipo Básico de Salud.

Como **Idea a Defender** se ha identificado, que al realizar el Diseño e Implementación de la gestión de la información de la Historia de Salud Familiar (HSF) se propondrá un Registro de Población que

garantizaría al Equipo Básico de Salud mayor dominio y control de la información asociada a sus pacientes, la vigilancia de la salud familiar y el procesamiento estadístico para la toma de decisiones en los diferentes niveles.

Las **Tareas de la Investigación** que se proponen:

1. Asimilar la Arquitectura definida por el MINSAP (Orientada a Servicios y Basada en Componentes (OBA\_CBA)) para el desarrollo de sus aplicaciones y el del Registro Informatizado de Salud (RIS) utilizando la Plataforma de Servicios PLASER.
2. Describir las herramientas y tecnologías informáticas que se utilizan en la propuesta.
3. Realizar un estudio del estado del arte y la fundamentación teórica de la investigación.
4. Modelar y Documentar el flujo de trabajo “Diseño” a partir del resultado del flujo de trabajo “Análisis”.
5. Modelar y Documentar el flujo de trabajo “Implementación”.
6. Implementar los componentes (Servicios Web) del Módulo RPOB.

La información contenida en este documento referente a la investigación, diseño y desarrollo de la aplicación ha sido estructurada como se describe a continuación:

**Capítulo 1:** Fundamentación Teórica. En este capítulo se da una breve explicación de la fundamentación del tema, las tendencias de las tecnologías actuales a tener en cuenta. Además se expone la tecnología y herramientas utilizadas en el desarrollo del sistema.

**Capítulo 2:** Diseño del Sistema. Se describen los aspectos relacionados con el diseño de la solución propuesta, se modelan los diagramas de clases del diseño y las clases persistentes, se muestra el modelo de datos, y se especifican los patrones para el diseño gráfico de la aplicación.

**Capítulo 3:** Implementación del Sistema. Refleja los aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de componentes y despliegue, se aborda la descripción de los estándares de diseño y codificación, además del tratamiento de errores en la implementación del sistema.

# CAPÍTULO 1

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción.

El objetivo fundamental de este capítulo es abordar los aspectos que se utilizan como soporte teórico del sistema a diseñar e implementar. Se exponen a través de una descripción de los conceptos asociados al problema, de la estructura organizativa del Sistema Nacional de Salud (SNS), así como de los conceptos principales para comprender la gestión de la información de los pacientes y la familia que son atendidos por el Equipo Básico de Salud (EBS). Además se presentan las características de las herramientas, metodologías y tecnologías escogidas para la solución del problema, así como se argumenta la elección de las mismas y la utilidad de las tecnologías de la informática y las comunicaciones.

### 1.2 Sistema Nacional de Salud.

La garantía de atención médica gratuita a toda la población cubana se convirtió desde los primeros momentos del triunfo de la Revolución en uno de los paradigmas sociales fundamentales, en correspondencia con la esencia humanista y de justicia social que caracteriza a nuestro proceso revolucionario. [4]

Desde el propio triunfo revolucionario se adoptaron medidas para transformar la salud pública en Cuba, una de las principales y más novedosa fue la creación del Sistema Nacional de Salud (SNS), designándose al Ministerio de Salud Pública como su organismo rector. [5]

Esta estructura organizativa comenzó a realizar importantes reformas a partir de los años 60, como parte fundamental de las transformaciones del período revolucionario y en respuesta al respeto más absoluto de uno de los derechos humanos fundamentales de todo ciudadano. Surge el servicio de hospitales rurales llevando la atención médica a zonas apartadas de la geografía nacional, se dan los primeros pasos para el fortalecimiento de la atención primaria; surgen los policlínicos integrales como una unidad asistencial creada para brindar servicios y resolver los principales problemas existentes en los primeros años de la revolución.

En la década del 70, por los cambios en el cuadro de morbilidad - mortalidad, los servicios prestados en los policlínicos integrales cobran nuevas funciones, cambiando la estructura de los mismos, pasando a una atención médica general, surgiendo así el policlínico comunitario donde prestaban atención los médicos generales. [6]

En la década del 80 surge el Programa del Médico y la Enfermera de la Familia, sentando precedentes en la salud pública internacional por su carácter novedoso y futurista, especialmente con la implantación y desarrollo del modelo de atención de Medicina Familiar a partir de 1984. [7]

En 1996, el SNS adoptó desde el punto de vista organizativo, estrategias fundamentales y priorizó cuatro programas básicos para continuar perfeccionándose: el Programa de Atención Materno Infantil (PAMI), de Control de Enfermedades Transmisibles y Crónicas no Transmisibles, y el de Atención al Adulto Mayor, todos los que han sido monitorizados, controlados y evaluados de acuerdo a la metodología establecida.

El Programa del Médico y la Enfermera de la Familia, se ratifica como el eje del actual desarrollo estratégico, orientándose el resto de las estrategias en función del mismo. Este modelo de atención es la mayor fortaleza y potencialidad que tiene el SNS. Por su existencia, filosofía, bases teóricas y lo que ha podido proporcionar al sistema de salud se ha logrado mantener los indicadores de salud y satisfacer las necesidades de la población, constituyendo un pilar básico de la Salud Pública Cubana. [8] **(Ver Anexo 1)**

Con más de 20 años de experiencia en este programa se comienzan a experimentar cambios para la atención primaria, de esta forma, servicios que antes eran exclusivos de hospitales son abiertos en instituciones de la atención primaria; surgiendo así hace aproximadamente 2 años el novedoso modelo de policlínico con nuevas funciones, acercando los servicios a la población, para hacer realidad las palabras de nuestro Comandante en Jefe: “... **una profunda revolución en los servicios de salud tendrá lugar en nuestra Patria...**” [9]

### 1.2.1 Informatización del Sistema Nacional de Salud.

La informatización del SNS tiene como objetivo acercar eficientemente y con calidad la prestación de los servicios de salud a la población, por lo que se pretende implementar un Programa General de

Informatización del SNS, que apoye las Estrategias y Políticas trazadas por la dirección del país y del MINSAP; de manera que se logre la incorporación progresiva y sistemática de las Tecnologías de la Información y las Comunicaciones (TIC) en función de la adquisición y gestión del conocimiento y los servicios de salud.

Se quiere que las instituciones del país alcancen un elevado nivel de informatización de las actividades que brindan, partiendo del Sistema de Atención Primaria y tomando como eje al policlínico, de manera que se logre un incremento de la calidad, efectividad y eficiencia de los servicios que se presten a la población, contribuyendo al logro de la satisfacción de los usuarios del Sistema Nacional de Salud.

Como solución integral significa la articulación de un nuevo paradigma en la prestación de servicios de salud, regido por el principio básico de lograr acercar cada vez más los servicios de salud a la población.

Entre los principales impactos esperados con la Informatización del SNS se pueden mencionar: [10]

Para la población:

- Equidad distribuida de acceso a servicios, tecnologías e información de salud independientemente de áreas geográficas, ni niveles de atención.
- Disfrutará la sensación de ser atendida por un personal médico mejor preparado y actualizado, elevando su confianza hacia el sistema de atención.
- Reducción del número de desplazamientos innecesarios entre instituciones de salud con el consecuente impacto en su vida social.
- Reducción de tiempos de esperas para el acceso a servicios especializados con la posibilidad de recibirlos en su propio escenario social.

Para el SNS:

- Gestión oportuna de una información confiable y actualizada que propiciará una optimización considerable de recursos.
- Elevación de la capacidad y calidad de la toma de decisiones asistenciales y gerenciales por la disposición oportuna de información actualizada para todos los niveles del SNS, que permitirá una rápida transferencia de la información sanitaria de un paciente.

- Disponer de un soporte y herramientas poderosas para la formación y actualización constante de sus miembros desde sus propios escenarios de desempeño, potenciando la investigación científica metacéntrica, nacional e internacional.
- Elevará el papel del Médico y la Enfermera de la Familia, incrementando su nivel científico y profesional.

En las líneas generales del Desarrollo Informático en la Salud se encuentran: la Atención Primaria, Secundaria y Terciaria, el Sistema Integrado de Urgencia Médica, Vigilancia de Salud, Telemedicina, Medicamentos y Fármacos, Epidemiología, Biblioteca y Universidad Virtual, Docencia Médica, entre otros.

SOFTEL, Empresa del Ministerio de la Informática y las Comunicaciones (MIC), tiene la misión de generar las soluciones informáticas e implementar un sistema de excelencia para el desarrollo y mantenimiento de productos de software especializados en salud y además organizar un esquema para la prestación de los servicios informáticos a dicho sector.

En la actualidad, se utiliza una estrategia nunca antes concebida en el país en un proceso de desarrollo de software, con una organización del proceso productivo a través de una eficiente gestión de requerimientos, donde participan desde un inicio, médicos y trabajadores de la salud, vinculados directamente a la Atención Primaria en calidad de expertos funcionales y en estrecho vínculo con los especialistas de Informática.

A través del proceso de desarrollo unificado (RUP), y haciendo uso del Lenguaje Unificado de Modelado (UML) se describen los procesos que se proponen para automatizar.

SOFTEL ejecuta estos objetivos en colaboración con la Universidad de las Ciencias Informáticas (UCI), para lograr la vinculación a la producción desde los primeros años de estudio de los estudiantes y los profesores líderes de proyectos y la formación en un segundo perfil en temas relacionados con la salud.

La experiencia de este trabajo en SOFTEL debe constituir el inicio de buenas prácticas en la producción de software con alta calidad y un ejemplo de normativa para los proyectos que deben irse abriendo de ahora en adelante en la Informatización del Sector de la Salud. [11]

El Sistema de Salud Cubano, posee en el nivel de Atención Primaria una plataforma ideal para articular los avances de las tecnologías de la información en función de hacer más eficiente todo el aparato estratégico y administrativo que rodea al propio sistema.

### **1.2.2 Informatización de la Atención Primaria de Salud.**

El Proceso de Informatización de la Atención Primaria de Salud (APS), es un proyecto priorizado para el SNS, cuyo objetivo fundamental consiste en la creación del Sistema Informatizado de Atención Primaria que permita la gestión médica, interacción con los consultorios del Médico de la Familia, obtención de estadísticas y apoyo en la logística de los nuevos servicios.

En el marco del Programa de Informatización de la Sociedad Cubana, el Proyecto APS en su concepción general se propone abordar el análisis, diseño y desarrollo de un producto de software, siguiendo las buenas prácticas internacionales y las normativas del MINSAP, logrando que facilite la gestión de la información en la Atención Primaria, acorde a los cambios y necesidades de este sector, permitiendo el flujo de información hacia los diferentes niveles de toma de decisiones. [12]

En esta nueva etapa de fortalecimiento del Sistema Nacional de Salud, la APS es el eje fundamental de estas transformaciones, teniendo como objetivo esencial convertir a los Policlínicos en centros de atención primaria de salud de la más alta calidad, cada vez más accesibles a la población, consolidando el Sistema Municipal de Salud, para dar cumplimiento al principio de la descentralización de las soluciones según los problemas de salud de la comunidad.

La automatización de la gestión de la APS debe comenzar por utilizar las tecnologías que permitan modelar la gestión de la información en este nivel, para almacenar, procesar, recuperar y comunicar información clínica y administrativa, relativa a todas las actividades de los policlínicos y unidades de la atención primaria.

Además debe tener la capacidad de comunicación y de integración de toda la información, independientemente de donde se haya generado y que sirva para el aprendizaje basado en experiencias

compartidas entre los profesionales en el país y fuera de nuestras fronteras, así como para lograr la integración con los procesos de los otros niveles de atención.

### 1.2.2.1 Proceso de Dispensarización.

En la APS se llevan a cabo varios procesos indispensables para el control efectivo de la salud de los pacientes. **La dispensarización** es un elemento fundamental en el trabajo del médico de familia, siendo un proceso organizado, continuo y dinámico de evaluación e intervención planificada, con un enfoque clínico, epidemiológico y social, del estado de salud de las personas y familias, coordinado y liderado por el EBS.

La dispensarización permite al médico de familia conocer y accionar sobre los principales problemas de salud, tales como riesgos, enfermedades y secuelas o discapacidades que afectan su calidad de vida y no permiten alcanzar indicadores de salud positiva. [13]

La **Atención Dispensarizada** consiste en el registro, agrupamiento y control de las personas y familias. Esta acción facilita la observación y atención médica permanente y dinámica de individuos y grupos con el objetivo de controlar riesgos y daños a la salud individual y colectiva y promover estilos de vida saludables.

Entre los **Propósitos de la Dispensarización** de un paciente se incluyen:

- ❖ Desarrollar un proceso de mejora continua del estado de salud de los individuos y familias.
- ❖ Elevar la satisfacción de la población con los servicios de salud que brinda el sistema.
- ❖ Determinar el estado de salud de individuos y familias.
- ❖ Promover estilos de vida saludables en los individuos y familias.
- ❖ Identificar e intervenir riesgos, enfermedades y otros daños a la salud individual y familiar.
- ❖ Facilitar la intervención multidisciplinaria en los problemas de salud individual y familiar.
- ❖ Aportar la información necesaria sobre la salud individual y familiar para el desarrollo del Análisis de la Situación de Salud.
- ❖ Mejorar la eficiencia en el trabajo del sistema de medicina familiar.

El proceso de dispensarización de un paciente se distingue por varias **Etapas de Dispensarización**, las cuales son:

- ❖ Registro y notificación de los pacientes que sean sanos o no, en la Historia de Salud Familiar por medio de la presentación espontánea de los individuos a la consulta o por visitas programadas a las viviendas.
- ❖ Control médico para determinar la presencia de factores de riesgo, enfermedades, discapacidades o secuelas que afecten la salud.
- ❖ Confección y actualización de la historia clínica individual.
- ❖ Clasificación en el Grupo Dispensarial.
- ❖ Orientación de medidas terapéuticas, de promoción y prevención de salud para lograr un mayor estado de bienestar y equilibrio biopsicosocial del individuo.
- ❖ Seguimiento y evaluación de los problemas según el Grupo Dispensarial.

Los individuos se clasifican en diferentes **Grupos de Dispensarización** a los que se les brinda una atención activa y controlada periódicamente, cuya frecuencia depende de la edad y el grupo en el que se clasificó y se definen estrategias y atención especializada para los grupos de acuerdo con sus características (clasificación). Los Grupos son:

**Grupo I:** Personas supuestamente sanas: en este grupo se incluyen todas aquellas personas que no padezcan ninguna enfermedad ni estén en riesgo de padecerlas.

**Grupo II:** Personas en riesgo o con factores de riesgo: en este grupo se incluyen todas aquellas personas que no padezcan ninguna enfermedad, pero tiene factores de riesgo para padecer enfermedades o accidentes que pueden repercutir en su salud.

El factor de riesgo ha sido definido como toda característica o circunstancia determinada de una persona o un grupo de personas que según los conocimientos que se poseen, asocia a los interesados a un riesgo anormal de sufrir un proceso patológico o de verse afectado desfavorablemente por tal proceso. Algunas enfermedades son consideradas factores de riesgos para la aparición de otras más graves, en estos casos la persona es considerada del grupo III.

Para poder prevenir, es necesario tener un conocimiento sobre el riesgo y sus implicaciones sobre la salud, para lograr entonces modificar actitudes, conductas, normas y condiciones familiares que faciliten esta labor; pero el conocimiento por sí mismo, no regula directamente el comportamiento del sujeto. Es conocido que el hombre bebe, fuma o come en exceso, conociendo en muchos casos lo perjudicial de estos hábitos para su salud y la repercusión de los mismos en la aparición o descompensación de múltiples enfermedades como las cardiovasculares, diabetes y respiratorias, entre otras.

Modificando los tres componentes de las actitudes: afectivo, cognitivo y conducta; a través de la intervención activa del sujeto, es que se puede lograr evitar las conductas de riesgo para prevenir enfermedades y discapacidades.

**Grupo III:** Personas enfermas: en este grupo se incluyen todas aquellas personas que padezcan alguna enfermedad, pero tiene discapacidad.

**Grupo IV:** Personas con discapacidades: en este grupo se incluyen todas aquellas personas que tienen discapacidad por enfermedades o accidentes.

Para la clasificación de los pacientes en Grupo Dispensariales se realiza el siguiente algoritmo: [14]

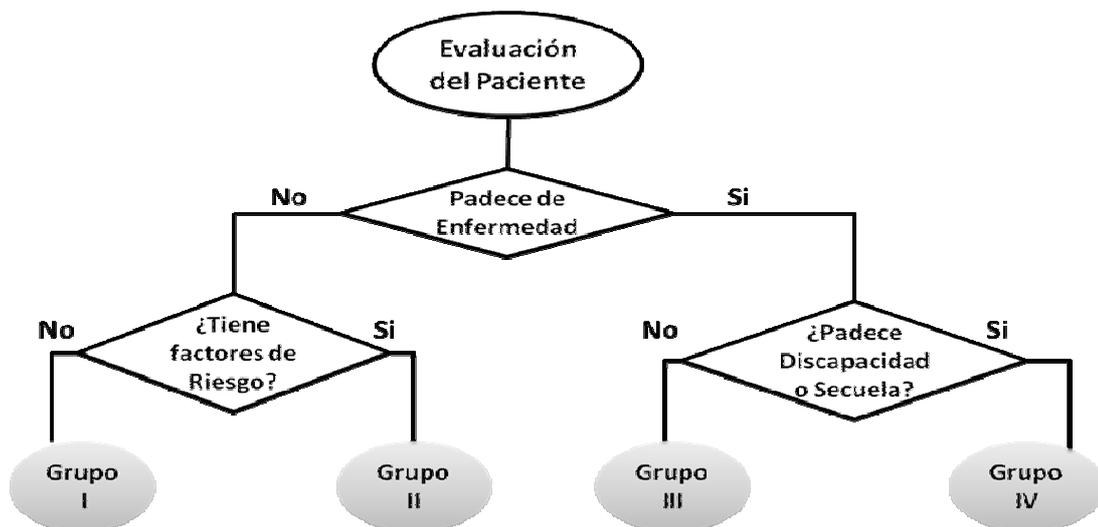


Figura 1.1 Algoritmo de Clasificación para los Grupos Dispensariales.

### 1.2.3 Registro Informatizado de Salud (RIS).

La Informatización de la Salud Cubana no ofrece un mecanismo único de integración de los sistemas de información desarrollados, estos en la actualidad se presentan como componentes aislados, lo cual trae consigo la duplicación de información y la consiguiente falta de integridad de la misma.

El Registro Informatizado de Salud (RIS), sentó las bases para la existencia de un sistema formado por componentes, desarrollados con un nivel de cohesión y acoplamiento que le permiten ser capaces de interactuar entre ellos y de esta forma reutilizar la información gestionada por cada componente.

El Proyecto APS vinculado a la informatización de la salud en Cuba encamina su tarea a analizar, diseñar y desarrollar un producto de software, único en su tipo, que hereda las características del RIS, pero que se caracteriza por ser un sistema distribuido de componentes distantes geográficamente, en constante interacción a través de la Red Telemática de Salud de Cuba, INFOMED, para dar respuesta a los procedimientos establecidos por el SNS para este nivel de atención.

Por tanto, es necesario desarrollar una arquitectura que garantice la máxima disponibilidad de cada uno de sus componentes, que permita la recuperación del sistema ante posibles fallos de conectividad o resolver problemas tales como la recuperación de la información, independientemente de su ubicación.

El RIS se basa en una arquitectura orientada a servicios, desarrollado con tecnología XML Web Services e implementado con PHP y MySQL. Desde el año 2003 forman parte del RIS los siguientes componentes: Registro de Unidades de Salud, Registro de Profesionales de la Salud, Registro de Ubicación, Registro del Ciudadano y Registro de Equipos de Salud.

### 1.2.4 Solución integral propuesta para la Informatización del SNS.

Para lograr la Informatización en este sector se pretende que todos los módulos estén incluidos en un conjunto de aplicaciones que formarán parte del **Sistema de Información para la Salud (SISalud)**, compuesto a su vez por el **Registro Informatizado de Salud (RIS)**, el **Sistema Informatizado de**

**Atención Primaria (SIAP), el Sistema Informatizado de Gestión Hospitalaria (SIGH) y el Sistema Informatizado de la Atención Especializada (SIAE).** [15]

**Registro Informatizado de Salud (RIS):** Está formado por los registros que son administrados o gestionados a nivel nacional o central y por los registros que pueden ser accedidos desde cualquier nivel de atención o institución de salud para lograr la continuidad en el seguimiento del paciente.

En esta nueva etapa de análisis, diseño e implementación se incorporarán al RIS, el Registro de Áreas de Salud, Registro de Servicios Médicos, Registro de Estudiantes, Registro de la Clasificación Internacional de Enfermedades y Problemas relacionados con la Salud y los codificadores propios de APS que se gestionan también a nivel central y definen diferentes aspectos que son utilizados localmente: Registro de Conductas e Indicadores y Registro de Problemas de Salud de la Atención Primaria.

**Sistema Informatizado de Atención Primaria (SIAP):** Se incluirán en la etapa actual los módulos propios de este nivel de atención: Registro de Actividades Diarias EBS y Registro de Población. Estos módulos constituirán una nueva herramienta para la transformación de los servicios que se brinda en este nivel, ya que integrarán diversos subsistemas como las actividades diarias del EBS, la dispensarización y la planificación de las acciones de salud, tanto individual como familiar.

En una segunda etapa continuarán incorporándose al SIAP los próximos módulos que se definan, según las prioridades del usuario.

**Sistema Informatizado de Gestión Hospitalaria (SIGH):** Se agruparán aquí los módulos que pertenecen al nivel de atención secundario u hospitalario y que serán definidos para próximos desarrollos. En esta etapa comenzará a formar parte del mismo el Registro de Autopsias, diseñado en la etapa actual bajo el Proyecto APS por la integración que tiene con el resto de los módulos que se desarrollan.

**Sistema Informatizado de la Atención Especializada (SIAE):** Se agruparán aquí los módulos que pertenecen al nivel de atención terciario o especializado: La Red Hospitales Especializados como Nefrología, Cardiología, Neurología y otros, además de la Red Bancos de Sangre.

El Sistema Nacional de Salud se organiza en 3 niveles de atención:

**Atención Primaria:** Se brinda a nivel de los policlínicos y/o hospitales rurales a través del Programa de Medicina Familiar y abarca a todos los Equipos Básicos de Salud (EBS).

Constituye el primer contacto del paciente sano o enfermo con el sistema de salud, que puede brindarse en locales adaptados para consultas o en el domicilio de los pacientes, a cualquier instancia del sistema de salud, aunque generalmente se realiza en el Consultorio Médico.

La función principal de la atención primaria es la promoción-prevención de salud en las diferentes comunidades, además se realizan procedimientos diagnósticos y terapéuticos que no requieren técnicas complejas, que aplicadas con calidad pueden resolver la mayor parte de los padecimientos que afectan a las poblaciones. Se diagnostican enfermedades graves que pueden ser derivadas a niveles de atención superiores, realizan seguimiento a personas con padecimientos crónicos y pueden otorgar bienestar a pacientes con patologías incurables. En general tiene carácter ambulatorio y comprende tanto a personas aparentemente sanas como a enfermas y/o discapacitadas. [16]

La atención primaria de salud es un nivel cualitativamente superior de atención médica, cuya esencia radica en la participación activa de la comunidad; donde las poblaciones de objetos pasivos, en espera de que se le ofrezcan soluciones, pasan a ser sujetos protagónicos activos ante sus propios problemas de salud. Decir participación comunitaria, es decir liderazgo, comunicación, cambio de hábitos y de estilos de vida, auto responsabilidad y acción creadora.

**Atención Secundaria:** Se brinda a nivel de las instituciones hospitalarias, por lo general son de carácter provincial, o sea atienden a toda la población de una provincia determinada. Se proporciona en un segundo escalón, al cual el paciente tiene acceso a través de una remisión del personal médico de la atención primaria o sin ella, acudiendo directamente la persona necesitada de atención médica.

Puede tener carácter ambulatorio (policlínicos especializados, servicios externos hospitalarios) o de hospitalización. En el mismo se brindan procedimientos diagnósticos y terapéuticos de elevada complejidad, que dan respuesta a los problemas moderados y graves de salud.

**Atención Terciaria:** Es aquella que por su condición muy especializada, sólo se brinda en determinados centros, ejemplo: Instituto de Neurocirugía, Instituto de Cirugía Cardiovascular, Instituto de Nefrología, Instituto de Gastroenterología, entre otros o en centros hospitalarios y/o de investigación categorizados como centros de referencia nacional y en algunos casos de referencia internacional.

Entre los Principios Rectores del MINSAP está el carácter estatal y social de la medicina, accesibilidad y gratuidad de los servicios, orientación profiláctica, aplicación adecuada de los adelantos de la ciencia y la técnica, participación de la comunidad e intersectorialidad, colaboración internacional y la centralización normativa y descentralización ejecutiva. [17]

Tiene como Funciones Rectoras ejercer el control y la vigilancia epidemiológica de las enfermedades y sus factores de riesgo, la vigilancia sanitaria de todos los productos que pueden tener influencia sobre la salud humana, regulación y control de las investigaciones biomédicas, normar las condiciones higiénicas y el saneamiento del medio ambiente, regular el ejercicio de la medicina y de las actividades que le son afines y ejercer la evaluación, el registro, la regulación y el control de los medicamentos de producción nacional y de importación, equipos médicos y material gastable y otros de uso médico. [18]

En el actual proceso de perfeccionamiento, el MINSAP se ha trazado como estrategias de desarrollo el perfeccionamiento de la atención primaria, la revitalización hospitalaria, el desarrollo del programa nacional de medicamentos y medicina natural y tradicional, el desarrollo de la tecnología de punta e investigación, así como contar con sistemas para urgencia, óptica, estomatología, asistencia social, control económico, atención al hombre y los cuadros.

### **1.3 Sistemas automatizados existentes vinculados al campo de acción.**

De acuerdo con los estudios realizados no se ha llevado a cabo ningún sistema encaminado a solucionar los problemas anteriormente mencionados, teniendo en cuenta su integración con el resto de las aplicaciones informáticas del Sistema de Información para la Salud.

Actualmente en el nivel de atención primaria, el EBS no tiene a su disposición de modo generalizado un sistema informático que en correspondencia con el nivel de automatización del sector, posibilite la

eliminación de estas dificultades, teniendo en consideración que las soluciones informáticas constituyen una respuesta actual a las necesidades de registro, procesamiento y recuperación de la información generada en cualquier esfera de la actividad humana.

Existen intentos de trabajo como el SYSAPS, que está orientado a la atención primaria de salud (APS), siendo una aplicación Web, que emplea las enormes facilidades de comunicación que brinda la red de redes (Internet) y que permita gestionar datos sobre los pacientes sean: edad, sexo, antecedentes patológicos personales, y otros datos de la historia clínica individual así como datos de la historia de salud familiar respecto a las viviendas, como las condiciones estructurales, cultura sanitaria, etc.; así mismo se permite consultar y reportar información referente a las reacciones adversas a los medicamentos.

Existen otros trabajos como APUS que es el Sistema Informático para la Gestión Médica en la APS. Este sistema brinda la información necesaria para la formulación y actualización periódica del Diagnóstico de la Situación de Salud, evaluar la eficiencia de los principales servicios y actividades médicas realizadas y sus resultados en términos de salud, en función del objetivo principal del Equipo de APS que consiste en la elevación del nivel de salud de la población bajo su responsabilidad. [19]

En el ámbito internacional se encuentra el Sistema Castor el cual tiene una serie de funcionalidades entre las que se destacan la captura inicial del paciente, además de los datos generales (registro de edad, inclusión de fotografía). Captura de consulta externa (signos vitales, entrevista exploración física y estudio de laboratorio, selección de diagnósticos y codificación automática según la CIE 10, etc.). Captura y seguimiento a controles de Diabetes mellitus, hipertensión arterial, embarazo, nutrición y registros de atención por lesiones e inmunizaciones de cada paciente.

Búsqueda e identificación de población en riesgo por género de edad, fechas y combinación de enfermedades. Presentación de historial de atención de cada paciente y detalle de cada una de las consultas (hora, fecha, lugar, medico, etc.)

#### 1.4 Políticas para el Desarrollo de la Informatización del SNS.

El objetivo principal de la informatización del SNS es acercar eficientemente y con calidad la prestación de los servicios de salud a la población. Algunas de las políticas para el desarrollo de este proceso son las siguientes: [20]

1. El proceso de informatización responde a las Políticas y Principios Socialistas.
2. Esta actividad es una estrategia vital y prioritaria del Sistema Nacional de Salud.
3. El desarrollo de producciones, inversiones y donaciones de sistemas informáticos deben responder a estrategias, planes de desarrollo, políticas de estandarización y proyectos que serán aprobados centralmente.
4. Todos los productos y servicios se integrarán a la ciber-infraestructura del sector y se realizarán en lo fundamental sobre sistemas abiertos, arquitectura orientada a los servicios y basadas en componentes, utilizando software libre y de calidad.

Para el desarrollo de las aplicaciones se acordó utilizar la Tecnología XML Web Services, de reciente aparición, la cual se caracteriza por:

- ❖ La más moderna y acertada tecnología para implementar cómputo distribuido.
- ❖ Arquitectura basada en componentes y orientada a servicios, no a objetos.
- ❖ Se ajusta a la nueva forma de distribución del software, que es orientada a servicios y no a productos.

Se definió el desarrollo de un producto portable que no depende del motor de base de datos escogido para su desarrollo y funciona tanto con el Sistema Operativo "Windows" como "Linux".

Como lenguaje de programación se utilizó el PHP y el gestor de bases de datos MySql, en los servidores se montaron los sistemas Apache para el caso de la plataforma Linux y Windows 2000 NT Server en el caso de la plataforma Windows.

## **1.5 Tendencias y Tecnologías Actuales a Considerar y Utilizar.**

### **1.5.1 Internet. Funcionamiento.**

Internet es una red de redes a nivel mundial que interconecta a millones de computadoras para formar un nuevo mundo. Estas son capaces de compartir información a través de un conjunto de protocolos TCP/IP. Lógicamente para lograr esta interconexión es necesario un canal de comunicación. Básicamente hoy en día uno de los protocolos de mayor uso es HTTP (protocolo de transferencia de hipertexto). Es usado para el envío y recepción de páginas web, además que es posible la transmisión de ficheros, fotos y respuestas a programas.

### **1.5.2 Aplicaciones Web.**

Una aplicación web es un sitio web, con la diferencia de que tiene que responder a reglas de negocio. Permite a los usuarios del sistema interactuar con estas, logrando así una interacción dinámica. Los sitios web son un conjunto de ficheros alojados en un servidor web, generalmente no se necesita de arquitecturas complejas para lograr su buen funcionamiento. Ambos necesitan en común un servidor web por ejemplo: IIS y hoy en día ha aumentando el uso de Apache (Software Libre) como preferencia, un lenguaje de programación como por ejemplo: PHP, JSP, Perl, Python, ASP/ASP.Net; aunque ciertamente ASP no es un lenguaje de programación, sino una arquitectura de desarrollo web en la que se pueden usar por debajo distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET, o VBScript/JScript para ASP), un Sistema Gestor de Base de Datos (DBMS) por sus siglas en inglés como por ejemplo las más usadas son: Microsoft SQL Server, Oracle, IBM Informix, IBM DB2 todas estas versiones comerciales; también se tienen otro conjunto importante pero que son versiones libres estos son: PostgreSQL, MySQL, SQLite, y MySQL que ha aumentado su uso a nivel mundial.

Para que estas aplicaciones sean usadas por los usuarios es necesario tener solamente un navegador web en el lado del cliente que realice peticiones http al servidor e interprete código HTML para dar formato a la información.

### 1.5.3 XML/Web Services. Service Oriented Architecture.

Un servicio web es cualquier servicio web disponible en Internet que usa como lenguaje de comunicación estándar XML y no se encuentra atado a ningún lenguaje de programación específico. Existen varios protocolos para la implementación de los servicios web: XML-RPC y SOAP. Estos protocolos son usados como medio de intercambio de mensajes estándar. Otros protocolos pueden servir como medio de transporte cuando se envía un documento XML por la red. Para el consumo de un servicio web es necesario tener una interfaz que sea legible para los humanos. Para ellos se usa WSDL (Web Service Description Language) que está basado en XML y contiene todos los parámetros necesarios para establecer una conexión con el servicio web, como puede ser el nombre del método, parámetros de entrada, parámetros de salida, etc.

También es necesario contar con un directorio UDDI (Universal Description Discovery and Integration). Es un modelo de directorios para Servicios web. Es una especificación para mantener directorios estandarizados de información acerca de los servicios web, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente. UDDI utiliza WSDL para describir las interfaces de los servicios web. Es un lugar en el cual se pueden buscar cuales son los servicios web disponibles, es una especie de directorio en el cual se encuentran los servicios web publicados, así como publicar los servicios web que se desarrollen. [21]

### 1.5.4 Entorno Distribuidos. Modelo Cliente Servidor.

La arquitectura Cliente-Servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

### **Ventajas del modelo Cliente Servidor:**

- ✓ El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- ✓ Se reduce el tráfico de red considerablemente. El cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.
- ✓ El sistema es fácil de escalar.

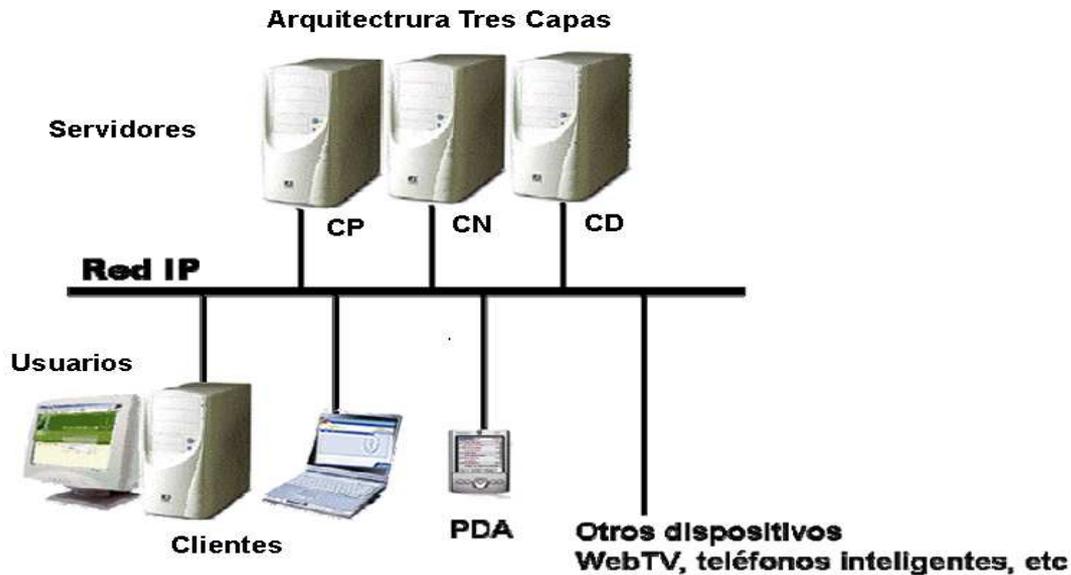
### **1.5.5 Modelo Cliente Servidor de Tres Capas.**

Es un modelo en el que las tareas se distribuyen por diferentes aplicaciones, estas aplicaciones pueden incluso estar en diferentes máquinas. Está compuesto por aplicaciones clientes que se encargan de las tareas más simples, como son la validación de datos, hacer peticiones al servidor de negocio, mientras que las aplicaciones servidoras son las encargadas de realizar las tareas de más trabajo y consumo de recursos. Como se dijo anteriormente en este modelo las aplicaciones cliente suelen estar en máquinas comunes sin muchos requerimientos de hardware, mientras que para las aplicaciones servidoras es necesario máquinas con mucha más potencia.

Este modelo trae muchos beneficios como son:

Los desarrollos son mucho más cortos. Mayor duración de las aplicaciones. Es posible reconfigurar el servidor o los clientes sin la necesidad de alterar los programas que constituyen la aplicación. Aumento de rendimiento y reducción de gastos, debido a que la aplicación se puede redistribuir por distintas máquinas.

El modelo Cliente/Servidor de 3 Capas está compuesto por 3 capas fundamentales: la Capa de Presentación (CP) que se encarga de hacer las validaciones de datos, hace peticiones a la Capa de Negocio (CN), que se encarga de procesar la información y construir las respuestas a las peticiones hechas. Toda la información que usa la CN para generar la respuesta, proviene de la Capa de Datos (CD) que normalmente suele ser una base de datos donde puede radicar un sistema gestor de base de datos genérico, pero debe existir un lenguaje común de comunicación ODBC. [22]



### 1.5.6 Servidor Web Apache

El servidor Web Apache es el más usado en el mundo. Más del 60% de los servidores web del mundo lo usan, existen muchas características que hacen a Apache ser el mejor servidor web, estas son:

- ✓ Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Cualquiera que posea una experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada. Esto significa que hay una gran cantidad de módulos Apache disponibles para su utilización.
- ✓ Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.
- ✓ Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod-perl.
- ✓ Apache funciona en Linux y en otros sistemas de Unix, además de que también funciona en Windows.

- ✓ Soporte para CGI (Common Gateway Interface): Apache soporta CGI utilizando los módulos `mod_cgi` y `mod_cgid`. Es compatible con CGI y aporta características extendidas como personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores Web.
- ✓ Soporte de host virtuales: Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales.
- ✓ Soporte de autenticación HTTP: Apache soporta autenticación básica basada en la Web. Está también preparado para autenticación basada en la digestión de mensajes, que es algo que los navegadores Web populares ya han implementado. Apache puede implementar autenticación básica utilizando tanto archivos estándar de contraseña como los DBM, llamadas a SQL o llamadas a programas externos de autenticación.
- ✓ Soporte de scripts PHP: Este lenguaje de script ha comenzado a ser muy utilizado y Apache ofrece un amplio soporte de PHP utilizando el módulo `mod_php`.
- ✓ Soporte de servlets de Java: Los servlets de Java y las Java Server Pages (JSP) se están convirtiendo en algo muy común en los sitios Web dinámicos. Puede ejecutar servlets de Java utilizando el premiado entorno Tomcat con Apache. [23]

### 1.5.7 Arquitectura Basada en Componentes (CBA).

La Arquitectura Basada en Componentes tiene como objetivo construir aplicaciones complejas mediante ensamblado de módulos (**componentes**), que han sido previamente diseñados por otras personas a fin de ser reusados en múltiples aplicaciones. Cada componente debe describir de forma completa las interfaces que ofrece, así como las interfaces que requiere para su operación. Y debe operar correctamente con independencia de los mecanismos internos que utilice para soportar la funcionalidad de la interfaz. Es actualmente una de las más prometedoras técnicas para incrementar la calidad del software, abreviar los tiempos de acceso al mercado y gestionar el continuo incremento de su complejidad.

### **Ventajas de desarrollar software basado en componentes:**

- ✓ **Reutilización del software.** Permite alcanzar un mayor nivel de reutilización de software.
- ✓ **Simplifica las pruebas.** Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
- ✓ **Simplifica el mantenimiento del sistema.** Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
- ✓ **Mayor calidad.** Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

### **1.6 Patrones de Diseño.**

Un patrón de diseño es una solución aceptada como correcta a un problema de diseño correcto. Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema y permite que algunos aspectos de la estructura del sistema puedan cambiar independientemente de otros aspectos. Facilitan la reusabilidad, extensibilidad y mantenimiento.

#### **1.6.1 Modelo Vista Controlador (MVC).**

Para el diseño de aplicaciones con sofisticadas interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador.

#### **Elementos del patrón:**

- ✓ **Modelo:** contendrá una lista de usuarios autorizados junto con sus contraseñas, la base de datos sobre la cual efectuar las consultas y los módulos (clases o librerías de funciones) que se encarguen de realizar las operaciones (comprobar la información del usuario, construir la consulta, efectuar la consulta).
- ✓ **Vista:** muestra la información del modelo al usuario y al necesitar poco o nada de código, puede ser desarrollada por un equipo de diseñadores independiente de los programadores.

- ✓ Controlador: gestiona las entradas del usuario y serviría de “pegamento”, recogería las órdenes y datos que el usuario enviase desde la vista, las traduciría en operaciones del modelo de negocio y, en base a los resultados, mostrará los resultados en uno u otro formulario de la vista.

### 1.6.2 Single Sign On (SSO).

Se entiende por Single Sign On (SSO) toda tecnología que facilite las tareas de inicio de sesión y de acceso a recursos de red desde distintas plataformas. Se trata de un proceso que no tiene por qué constar necesariamente de una sola operación, ni conllevar siempre un inicio de sesión. La utilización de SSO en entornos mixtos es muy compleja, ya que no existe ninguna solución universal que pueda aplicarse a todos los entornos y sistemas operativos.

### ¿Por qué MVC y SSO?

Aplicando este patrón se logra separar el modelo de negocios de la presentación usada, así la información podrá ser vista en múltiples formatos. Esta separación permite diversas vistas para una misma lógica de negocios. Los usuarios se autenticarán una sola vez así se garantizará una sola interfase de autenticación y de esta forma no habrá preocupaciones por parte de los usuarios de escribir contraseñas cortas, y los administradores no se verán obligados a navegar entre múltiples interfases para dar altas o bajas a los usuarios.

### 1.7 Lenguajes de Programación Web.

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial. Para realizar una página con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL (Uniform Resource Locators), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita. El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de internet, como es el http (Hypertext Transfer Protocol). Con el comienzo de Internet y la programación web, se

desfasaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, buscando ficheros pequeños para facilitar la carga de los mismos. La programación web se orientaba a un diseño muy cargado interactuando con el usuario, mientras que al empezar a competir con millones de webs se ha optado más por el diseño sencillo y de fácil comprensión.

Entre los lenguajes que trabajan del lado del servidor se pueden citar algunos, que se destacan por ser los más sobresalientes como son PERL, ASP, PHP, Java y JSP. Los lenguajes que trabajan del lado del cliente se encuentran el JavaScript, XSLT y el Visual Basic Script, estos dos últimos al combinarse con el HTML forman lo que se conoce como DHTML, es decir, salida estándar dinámica o HTML dinámico.

### **ASP**

Microsoft introdujo esta tecnología llamada Active Server Pages en diciembre de 1996, para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web utilizando diferentes lenguajes scripts (Visual Basic Script o JavaScript de Microsoft) y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente. Actualmente se ha presentado ya la segunda versión: el ASP.NET, con el objetivo de resolver las limitaciones de ASP y posibilitar la creación de software como servicio, este comprende algunas mejoras respecto al ASP en cuanto a las posibilidades del lenguaje y rapidez con la que funciona. La mayor desventaja de este lenguaje es que solo se puede implementar sobre los Servidores Web de Microsoft. [24]

### **JSP**

JSP es un acrónimo de Java Server Pages, es una tecnología que permite mezclar HTML estático con HTML generado dinámicamente. Orientada principalmente a crear páginas Web con programación en Java. Muchas páginas Web que están construidas con programas CGI son casi estáticas, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI, incluyendo los servlets, hacen que genere la página completa mediante este programa, incluso aunque la mayoría de ella sea siempre lo mismo. JSP permite crear dos partes de forma separada. Con JSP se crean aplicaciones Web que se ejecuten en variados Servidores Web, de múltiples plataformas. [25]

### 1.7.1 Lenguajes de programación Web a utilizar.

#### PHP

PHP (acrónimo de Hypertext Preprocessor) es un lenguaje "del lado del servidor" (esto significa que PHP funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas Web dinámicas. Puede ser incluido con facilidad dentro del código HTML, y permite una serie de funcionalidades tan extraordinarias que se ha convertido en el favorito de millones de programadores en todo el mundo. Combinado con la base de datos MySQL, es el lenguaje estándar a la hora de crear sitios de comercio electrónico o páginas Web dinámicas.

❖ Entre sus características fundamentales están:

**Gratuito:** Al tratarse de software libre, puede descargarse y utilizarse en cualquier aplicación, personal o profesional, de manera completamente libre.

**Gran popularidad:** Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código.

**Enorme eficiencia:** Con escaso mantenimiento y un servidor gratuito como por ejemplo el servidor Apache el cual es usado en nuestra aplicación y puede soportar sin problema millones de visitas diarias.

Sencilla integración con múltiples bases de datos. Esencial para una página Web verdaderamente dinámica, además puede conectarse también con PostgreSQL, Oracle, dbm, filePro, interbase o cualquier otra base de datos compatible con ODBC (el Estándar Abierto de Conexión con Bases de Datos). Posee además una extensión DBX de abstracción de base de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión.

Entre los diez competidores principales de PHP se puede citar a Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) y Allaire ColdFusion. En comparación con estos productos, PHP cuenta con muchas **ventajas**, entre las que se encuentran las siguientes:

Alto rendimiento, Interfaces para una gran cantidad de sistemas de base de datos, diferentes Bibliotecas incorporadas para muchas tareas Web habituales, bajo coste facilidad de aprendizaje y uso, portabilidad y acceso a código abierto.

PHP fue concebido en 1994 y es fruto del trabajo de un hombre, Rasmus Lerdorf. Ha sido adoptado por otras personas de talento y ha experimentado varias transformaciones importantes hasta convertirse en el producto actual. En octubre de 2002, era utilizado por más de nueve millones de dominios de todo el mundo y su número crece rápidamente, actualmente es usado por más de 20 millones de dominios. **(Ver Anexo 2)**

### **Java Script**

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript se puede crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. [26]

### **XSLT**

XSLT es el acrónimo de Extensible Stylesheet Language Transformation, es un lenguaje que se usa para convertir documentos XML en otros documentos XML; puede convertir un documento XML que obedezca a un DTD a otro que obedezca otro diferente, un documento XML bien formado a otro que siga un DTD, o, lo más habitual, convertirlo a "formatos finales", tales como WML (usado en los móviles WAP) o XHTML.

Los programas XSLT están escritos en XML, y generalmente, se necesita un procesador de hojas de estilo, o *stylesheet processor* para procesarlas, aplicándolas a un fichero XML. [27]

De acuerdo con las características antes expuestas de los múltiples lenguajes existentes, se llega a la conclusión de que los lenguajes a utilizar serán PHP y XSLT. Por la velocidad de PHP a la hora de procesar los datos, ser uno de los lenguajes más universales de la actualidad, por tener una comunidad tan grande y ser soportado en varias plataformas. XSLT es el lenguaje universal de transformación de documentos, éste permite definir las variadas presentaciones, además de ser muy rápido a la hora de procesar los documentos.

### **1.8 Sistemas de Gestión de Bases de Datos (SGBD).**

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en una colección de datos interrelacionados y una colección de programas para acceder a esos datos. El objetivo principal de un SGBD es proporcionar un entorno que sea tanto conveniente como eficiente para las personas que lo ocupan en el almacenamiento y recuperación de la información.

Para plasmar los tres niveles en el enfoque o modelo de datos seleccionado, es necesaria una aplicación que actúe de interfaz entre el usuario, los modelos y el sistema físico. Esta es la función que desempeñan los SGBD, y que pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

Un SGBD tiene los siguientes objetivos específicos:

- ✓ Independencia de los datos y los programas de aplicación
- ✓ Minimización de la redundancia
- ✓ Integración y sincronización de las bases de datos
- ✓ Integridad de los datos
- ✓ Seguridad y protección de los datos

- ✓ Facilidad de manipulación de la información

### **SQL Server**

SQL Server 2000 proporciona completas capacidades de programación de bases de datos basadas en estándares Web. La perfecta compatibilidad con el lenguaje XML y los estándares de Internet le proporcionan la capacidad para almacenar y recuperar fácilmente datos en formato XML con procedimientos almacenados integrados. También puede utilizar datagramas de actualización de XML para insertar, actualizar y eliminar datos con facilidad. [28]

#### **Ventajas:**

- ✓ Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles.
- ✓ Administración y optimización simplificadas: resulta sencillo administrar bases de datos de forma centralizada junto a todos los recursos empresariales.
- ✓ Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server.
- ✓ Gestión: Con una completa interfaz gráfica que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.

#### **Desventajas:**

- ✓ Licencias con costos altos.
- ✓ Plataformas Windows.

### **PostgreSql**

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS), está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. [29]

**Ventajas:**

- ✓ Soporta transacciones y desde la versión 7.0, llaves foráneas (integridad referencial).
- ✓ Soporta un subconjunto de SQL92 mayor que el que soporta MySQL.

**Desventajas:**

- ✓ Consume bastantes recursos y carga más el sistema.

**1.8.1 Sistema de Gestión de Base Datos a Utilizar.**

**MySql**

Es un *sistema de administración de bases de datos relacional (RDBMS)*. Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2. Es un SGBD basado en Open Source (Código abierto) diseñado para los sistemas Unix formando parte de la tecnología LAMP (Linux, Apache, MySQL y PHP), aunque existen versiones para Windows. Actualmente está en su versión 5.0.6-beta incluyendo procedimientos almacenados (stored procedures), disparadores (triggers), vistas (views) y muchas otras características. [30]

Son muchas las razones para escoger MySql como solución de misión crítica para la administración de datos:

- ✓ Es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico
- ✓ Mucho más rápido que la mayor parte de sus rivales.
- ✓ Se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.

**Desventajas:**

- ✓ Es gratis para aplicaciones de código abierto, de lo contrario hay que pagar licencia comercial.

### **1.9 Desarrollo basado en RUP bajo la herramienta Rational Rose.**

Cada día la producción de software busca adecuarse más a las necesidades del usuario, esto trae como consecuencia que aumente en tamaño y complejidad. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo.

El Proceso Unificado es un proceso de desarrollo de software (conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software). Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

El Proceso Unificado está basado en componentes. Utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial de RUP, sus desarrollos fueron paralelos. No obstante los verdaderos aspectos definitorios del proceso unificado se resumen en tres fases claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. [31]

### **1.10 UML (Unified Modeling Language).**

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas. [32]

### **1.11 Rational Rose.**

Existen herramientas Case de trabajo visuales como el Analise, el Designe, el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos. En la actualidad la mejor y más utilizada en

el mercado mundial es Rational Rose y es la que se utiliza en la modelación de este proyecto. Rational Rose es la herramienta de modelación visual que provee el modelado basado en UML.

La Corporación Rose ofrece un Proceso Unificado (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la de pruebas. Para cada una de estas etapas existe una herramienta de ayuda en la administración de los proyectos.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

Se utilizó el Proceso Unificado de Software (RUP) para desarrollar el proyecto. Este es un proceso que garantiza la elaboración de todas las fases de un producto de software orientado a objetos. RUP utiliza UML, que es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Rational Rose permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP), en concreto:

- ✓ Modelado del negocio
- ✓ Captura de requisitos (parcial)
- ✓ Análisis y diseño (completo)
- ✓ Implementación (como ayuda)
- ✓ Control de cambios y gestión de configuración (parte).

### **Características Principales:**

- ✓ Admite como notaciones: UML, COM, OMT y Booch.
- ✓ Realiza Chequeo semántico de los modelos.
- ✓ Ingeniería “de ida y vuelta”: Rose permite generar código a partir de modelos y viceversa.
- ✓ Desarrollo multiusuario.
- ✓ Integración con modelado de datos.
- ✓ Generación de documentación.

- ✓ Tiene un lenguaje de script para poder ampliar su funcionalidad.

### 1.12 Plataforma de Servicio (PLASER).

PLASER, acrónimo de Plataforma de Servicio, constituye una plataforma sobre la que se pueden desplegar aplicaciones XML Web Services. En su concepción se han utilizado estándares actuales y normas abiertas lo cual facilita la programación y homogeneidad de los componentes. La plataforma de Servicios (PLASER), está conformada fundamentalmente por varias clases en PHP o una librería de clases, que puede o no ser usada para que un componente se integre al Registro Informatizado de Salud (RIS) pero de no ser usada la seguridad corre por parte del programador. En esta versión PLASER sólo soporta como llamada RPC el protocolo SOAP, pero en futuras versiones se piensa en otros protocolos de transportes o incluso el acceso local a código a nivel de File System, de forma tal que para el programador es totalmente transparente si la invocación del proceso es remoto, local, por SOAP, directamente a código, etc.



**Figura 1.3** Distribución de la Plataforma de Servicios (PLASER).

Este sistema está concebido completamente sobre Arquitectura Basada en Componentes y Orientada a Servicios, usando el paradigma de XML Web Services específicamente SOAP. En su concepción se han utilizado estándares actuales y normas abiertas. PLASER desde el punto de vista estructural permite trabajar con cualquier base de datos que cumpla con la norma SQL ANSI 92; pero desde el punto de vista de implementación solo trabaja con las bases de datos soportadas por el componente DBX, ya que encapsula a dicho componente y lo utiliza para el acceso a bases de datos.

PlaSer en su estructura encapsula varios componentes:

XSL: Extensión para transformaciones de ficheros XML.

Pear SOAP: Biblioteca para la comunicación SOAP.

DBX: Extensión para la abstracción del acceso a datos.

### 1.13 Herramientas a utilizar.

Se decidió que se utilizaría el Rational Rose Enterprise Edition 2003, para sustentar la documentación, como modelador visual de la notación UML (Unified Modeling Language) para la confección de los diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades y ayuda a comprender la importancia del proceso de desarrollo orientado a objetos.

También Stylus Studio 5.1 para crear los ficheros XSL a través de escenarios XML y para la edición del código PHP se usará el NuSphere, que es un editor para programadores con soporte para múltiples formatos, similar a otras aplicaciones como el Zend Studio, facilita el trabajo de programación con numerosas características de gran utilidad como un sistema de ayuda, plantillas de documentos y fragmentos de código frecuentes, código de colores para comandos en PHP, Perl, Javascript, SQL y HTML. Además, esta herramienta incluye un cliente de FTP y un servidor Web integrados, totalmente configurables según tus necesidades de trabajo, como servidor Web Apache 2.0 con PHP 4.3.4; MySQL-Front es una sencilla y útil aplicación diseñada especialmente como cliente para desarrolladores que trabajan con MySQL, como servidor de bases de datos se escogerá MySQL 4 en cualquiera de sus versiones, otro cliente usado es el EMS SQL Manager 2005 que es una aplicación de alto desempeño para la administración y desarrollo de servidores de bases de datos MySQL. Se utilizó Dreamweaver 8 para el diseño de las distintas interfaces de la aplicación web y llevarlas al formato XHTML.

### 1.14 Conclusiones.

En este capítulo, se profundizó en el conocimiento de algunos conceptos necesarios para la comprensión de la investigación. Además se realizó un análisis y descripción de las tecnologías y herramientas que serán utilizadas a lo largo del desarrollo del sistema propuesto, y se fundamentaron las elecciones del lenguaje usado (PHP), el sistema gestor de bases de datos (MySQL), y la metodología a utilizar (RUP).

## CAPÍTULO 2 DISEÑO DEL SISTEMA

### 2.1 Introducción.

En este capítulo se describe el diseño de la solución propuesta a partir del análisis llevado a nivel de casos de usos. Se construyen los diagramas de clases del diseño, y de las clases persistentes, el modelo de datos. Además, se definen los principios de diseño y la justificación de los distintos patrones que se utilizan en la arquitectura y diseño del sistema.

### 2.2 Modelo del Diseño.

El modelo de diseño intenta preservar la estructura definida por el modelo de análisis. Es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto a otras restricciones del entorno de implementación tienen su impacto en el sistema que se desarrolla.

#### 2.2.1 Justificación del Uso de Patrones.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Estos identifican las **Clases**, **Instancias**, **Roles**, **Colaboraciones** y la **distribución de responsabilidades**. **Un patrón de diseño es una solución estándar para un problema común de programación. Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios, un proyecto o estructura de implementación que logra una finalidad determinada. Un lenguaje de programación de alto nivel, una manera más práctica de describir ciertos aspectos de la organización de un programa. Conexiones entre componentes de programas y la forma de un diagrama de objeto o de un modelo de objeto.**

### Patrones utilizados:

#### Arquitectura

- ✓ Modelo Vista-Controlador.
- ✓ Tres Capas (Layers).
- ✓ Basada en Componentes (CBA).
- ✓ Orientada a Servicios (SOA).

#### Diseño

- ✓ Fachada.
- ✓ Proxy.

#### 2.2.1.1 Modelo – Vista – Controlador.

Uno de los patrones que ha demostrado ser fundamental, a la hora de diseñar aplicaciones web es el Modelo-Vista-Control (MVC). Este patrón propone la separación en distintos componentes de la interfaz de usuario (vistas), el modelo de negocio y la lógica de control. Una vista es una “fotografía” del modelo (o una parte del mismo) en un determinado momento. Un control recibe un evento disparado por el usuario a través de la interfaz, accede al modelo de manera adecuada a la acción realizada, y presenta en una nueva vista el resultado de dicha acción. Por su parte, el modelo consiste en el conjunto de objetos que modelan los procesos de negocio que se realizan a través del sistema.

A la hora de utilizar el MVC en aplicaciones web es conveniente utilizar un único servlet como controlador para toda la aplicación. Este control gestiona todas las peticiones, incluyendo invocaciones a servicios de seguridad, gestión de excepciones, selección de la siguiente vista, etc. Esto se conoce como el patrón Front Controller (controlador frontal o fachada). El poder centralizar en un solo punto servicios como la gestión de conexiones a base de datos, comprobaciones de seguridad o gestión de errores favorecen que la aplicación sea mucho más robusta y aísla de todos estos aspectos, al resto de los componentes.

### **2.2.1.2 Arquitectura en 3 Capas.**

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

Las capas de presentación, de negocio y de datos pueden residir en un único servidor (no sería lo normal), si bien lo más usual es que haya una multitud de servidores donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo servidor, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más servidores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios servidores, los cuales recibirán las peticiones del servidor en que resida la capa de negocio.

Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más servidores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de servidores sobre los cuales corre la capa de datos.

En una arquitectura de tres niveles, los términos "capas" y "niveles" no significan lo mismo ni son similares. El término "capa" hace referencia a la forma de cómo una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos. En cambio, el término "nivel", corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física.

### **2.2.1.3 Arquitectura Basada en Componentes (CBA).**

El Desarrollo de Software Basado en Componentes (DSBC) se centra en el desarrollo de grandes sistemas de software mediante la integración de componentes de software ya existentes. Su filosofía es: "Reutilice lo que existe, no construya". Realiza y mejora la flexibilidad y la mantención, proporcionando la reducción de costos en el desarrollo de software, incremento en la velocidad de ensamble de nuevos sistemas y reducción de carga de mantenimiento para correcciones y mejoras.

La visión de un componente como proveedor de servicios es la siguiente:

- ✓ Entidad ejecutable e independiente. Cuanto mayor es la independencia más fácil es la comprensión, mejorar, extender, adaptar, corregir y el mantenimiento.
- ✓ Publica interfaz de servicios suministrados y las interacciones son a través de ésta.
- ✓ Generalmente también define interfaz de servicios que debe proveer el sistema que lo utiliza.

En la medida de independencia se busca **Alta Cohesión** y **Bajo Acoplamiento**. La cohesión es la medida de cuán focalizado esta el módulo en una cosa y en el acoplamiento es la medida de cuán conectado está el modulo con el ambiente.

### 2.2.1.4 Arquitectura Orientada a Servicios (SOA).

Es una manera de diseñar los sistemas de software y su ambiente, para permitir que los servicios puedan ser provistos a aplicaciones de usuarios finales, procesos de negocio “ejecutables”, o bien, otros servicios; a través de la publicación y descubrimiento de la interfaz de los servicios. Tiene grandes ventajas tales como:

#### ❖ Reusabilidad de Servicios:

Esto ayuda en la disminución de tiempo y costos de desarrollo de las aplicaciones al utilizar servicios disponibles ya desarrollados, para resolver problemáticas comunes a otras aplicaciones. Disminución del riesgo al utilizar algo ya probado (robusto). Logro de mejoras globales en las aplicaciones al mejorar un servicio que están siendo usado por varias aplicaciones, agilidad frente al cambio. Disminución de los costos de administración y operación.

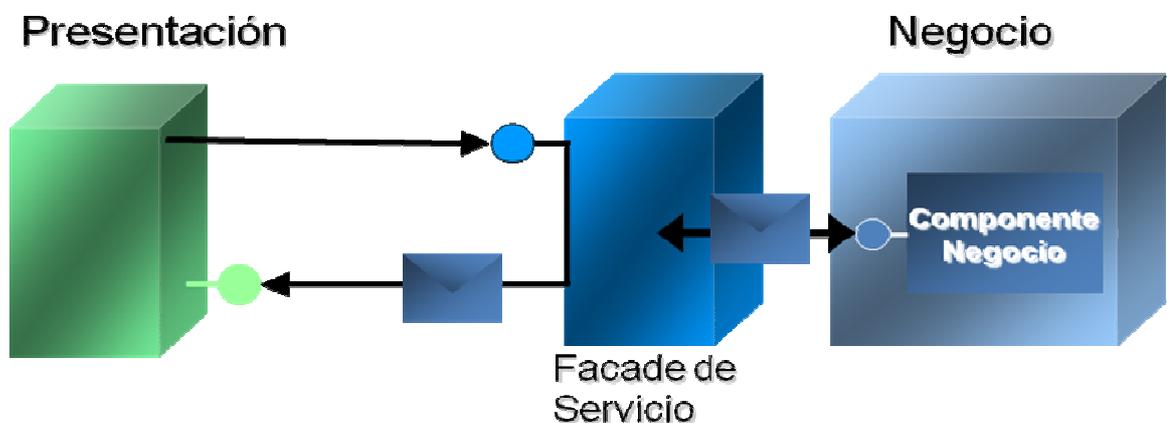
#### ❖ Interoperabilidad de Aplicaciones:

Este trata la disminución de la complejidad de interrelación entre aplicaciones al utilizar un “protocolo común” de entendimiento (INTERFAZ COMÚN). Disminución de la complejidad de la integración, al interactuar con un elemento que se abstrae de la tecnología y ubicación de los servicios.

### 2.2.1.5 Fachada.

El patrón fachada trata de simplificar las interfases entre dos sistemas o componentes de las distintas aplicaciones del proyecto, ocultando un sistema complejo detrás de una clase que hace las veces de pantalla o fachada. La idea principal es la de ocultar todo lo posible la complejidad del sistema distribuido y el conjunto de clases o componentes que lo forman, de forma que solo se ofrezca un (o unos pocos) punto de entrada al sistema tapado por la fachada. Además de usar una clase fachada para comunicar las dos partes o componentes, es la de aislar los posibles cambios que se puedan producir en alguna de las partes. Si cambias, por poner un ejemplo, el medio de comunicación o de almacenamiento de una de las partes, la otra, que por ejemplo hace la presentación, no tiene porque enterarse, y viceversa.

Fachada provee una única y sencilla interfaz para la gran mayoría de los clientes que realizan peticiones simples. Fachada no oculta los objetos del subsistema, por lo que aquellos clientes que requieren mayor personalización pueden realizar las peticiones por encima de Fachada. Fachada provee una solución para el diseño de software en capas y como tal es realizado en la arquitectura de tres capas del proyecto APS.



**Figura 2.1** Patrón Fachada.

### 2.2.1.6 Proxy

El patrón **Proxy** se utiliza como intermediario para acceder a un objeto, permitiendo controlar el acceso a él. Para este se necesita crear objetos que consumen muchos recursos, por lo que no es necesario instanciarlos a no ser que el cliente lo solicite o se cumplan otras condiciones determinadas.

**ProxPla (Plaser\_Server)**: Es un fichero de configuración cuya función es servir de proxy ante los pedidos que llegan de la capa de presentación y a la vez los enruta, pues es el único que sabe donde están los componentes. También controla y chequea los derechos, trasea, autentifica, etc. Constituye el primer nivel de seguridad. **ProxPla** utiliza un puerto para leer diferente al que usan los componentes, los cuales están locales.

Por defecto la arquitectura está pensada para que exista un servidor con la capa de presentación (directorio 'web/'), este siempre pasa a través de otro servidor que es el 'proxpla' (proxy plaser) y este a su vez es el único que llega hasta el 'core/'. Evidentemente que todo esto se puede hacer en una sola máquina y por eso es el uso de los Virtual Host, además el uso de estos Virtual Host es para tener ficheros logs independientes para cada componente y que sea más fácil el rastreo de errores.

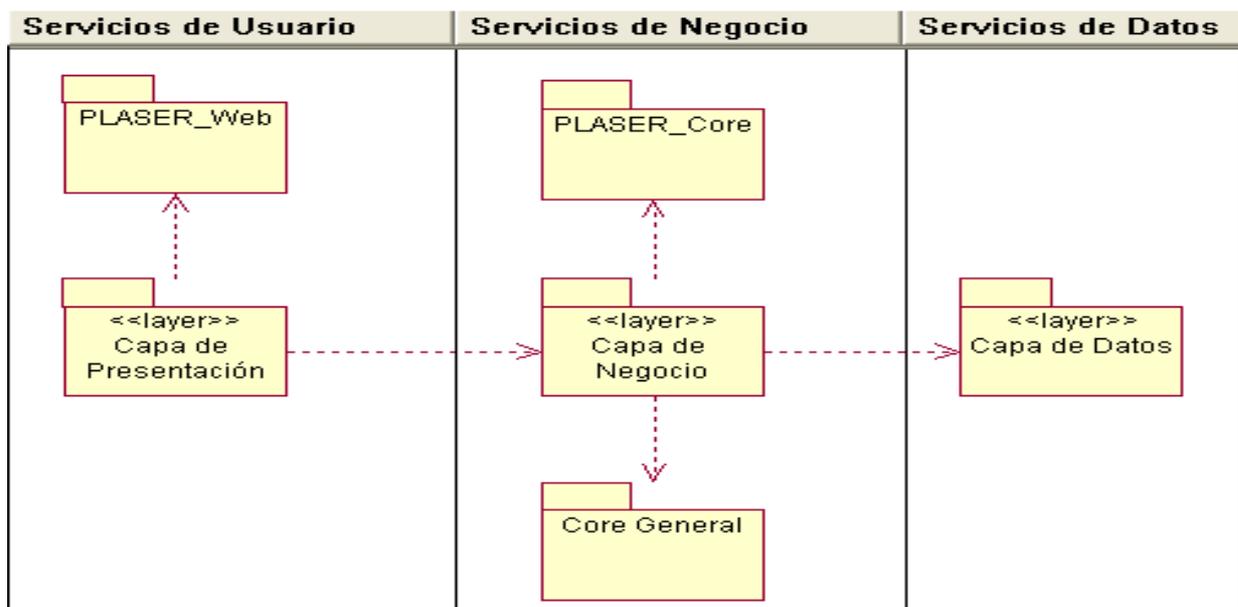
Usted puede cambiar toda la configuración para tener un solo sitio web y separar los componentes por directorios diferentes. **(Ver Anexo 3)**

Las características del patrón proxy nos brindan varias utilidades:

- 1- Capa de seguridad: Se puede crear un anillo de seguridad que envuelva al componente en cuestión, o sea, dejar el componente del core en una zona completamente segura y confiable sin velar por la seguridad.
- 2- Sobrecargar o prependizar: Similar a la técnica de seguridad, pero puede usarse además de la seguridad a prependizar algún business logic que se necesite antes de la invocación del método del core sin tener que modificar el componente del core o negocio.

3- Acceso a componentes externos: Permitir el acceso a componentes externos al sistema, aparentando que son componentes del sistema interno, o sea, se puede crear una ruta hasta el componente externo, pareciendo que pertenece al sistema.

### 2.2.2 Estructura del Diseño.



**Figura 2.2** Estructura del Diseño.

### 2.2.3 Diagramas de Clases del Diseño.

El Diagrama de Clases es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones. RUP brinda una solución para las extensión UML para la Web, donde en este existen los estereotipos, los valores etiquetados y las restricciones.

**Los distintos Estereotipos Web son:**



- La página servidora (Server Page) representa la página Web que tiene código que se ejecuta en el servidor.



- El formulario (Form) es la colección de elementos de entrada que son parte de una página cliente. Sus atributos son los elementos de entrada del formulario.



- La página cliente (Client Page) representa una página Web, con formato HTML. Son interpretadas por el navegador.

En las distintas relaciones entre las clases, el código servidor se encarga de construir o generar el resultado html que conforma el código cliente (*Build*), los formularios envían sus datos al código servidor para ser procesados los pedidos (*Submit*), y además forman parte del código cliente o resultado html, y es por esto que la relación entre la clase empleada para el código cliente y la clase empleada para el formulario es de agregación. Entre páginas clientes pueden existir vínculos (*Link*).

Hasta Desde	Client Page	Form	Server Page
Client Page	<<Link>> , <<redirect>>	Contiene	<<Link>> , <<redirect>>
Form	Agregado por.	---	<<Submit>>
Server Page	<<Build>>, <<redirect>>		<<Redirect>>

**Tabla 2.1** Relaciones entre las clases principales que conforman la extensión de UML para Web.

Diagrama de Clases de Gestionar Problemas de Salud.

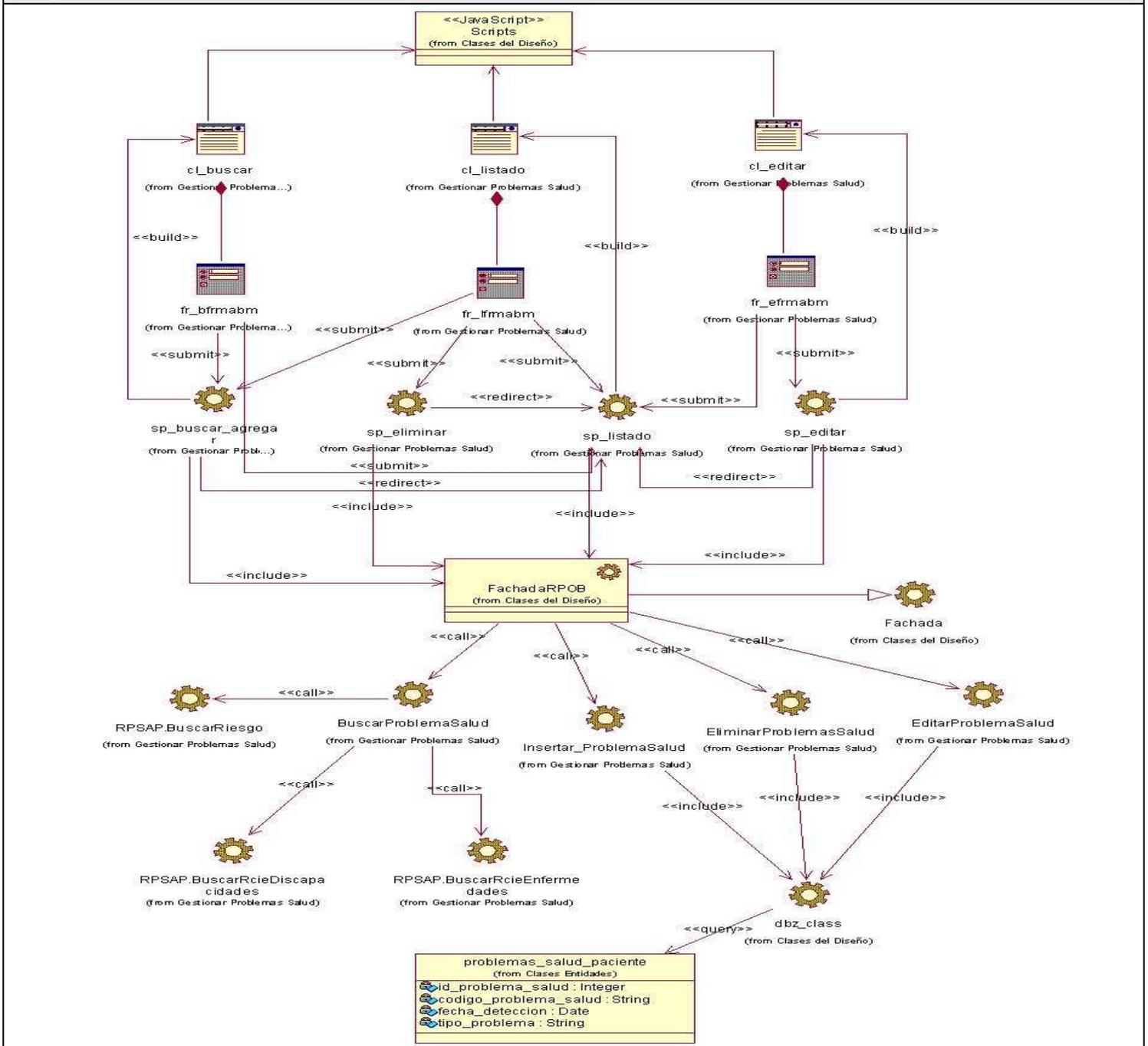


Diagrama de Clases de Gestionar Seguimiento Diario de un Paciente.

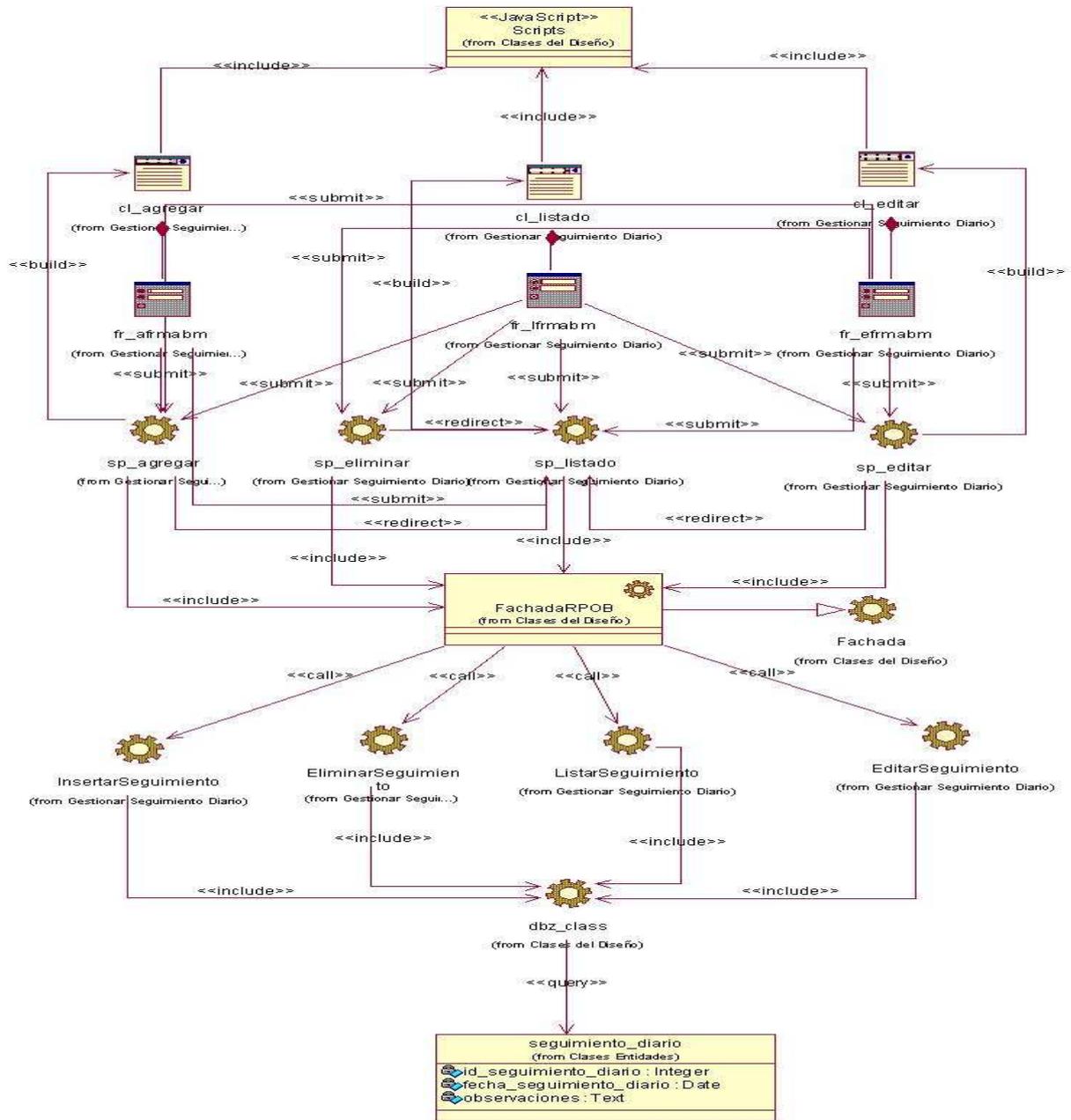


Diagrama de Clases de Gestionar Grupo Dispensarial de un Paciente.

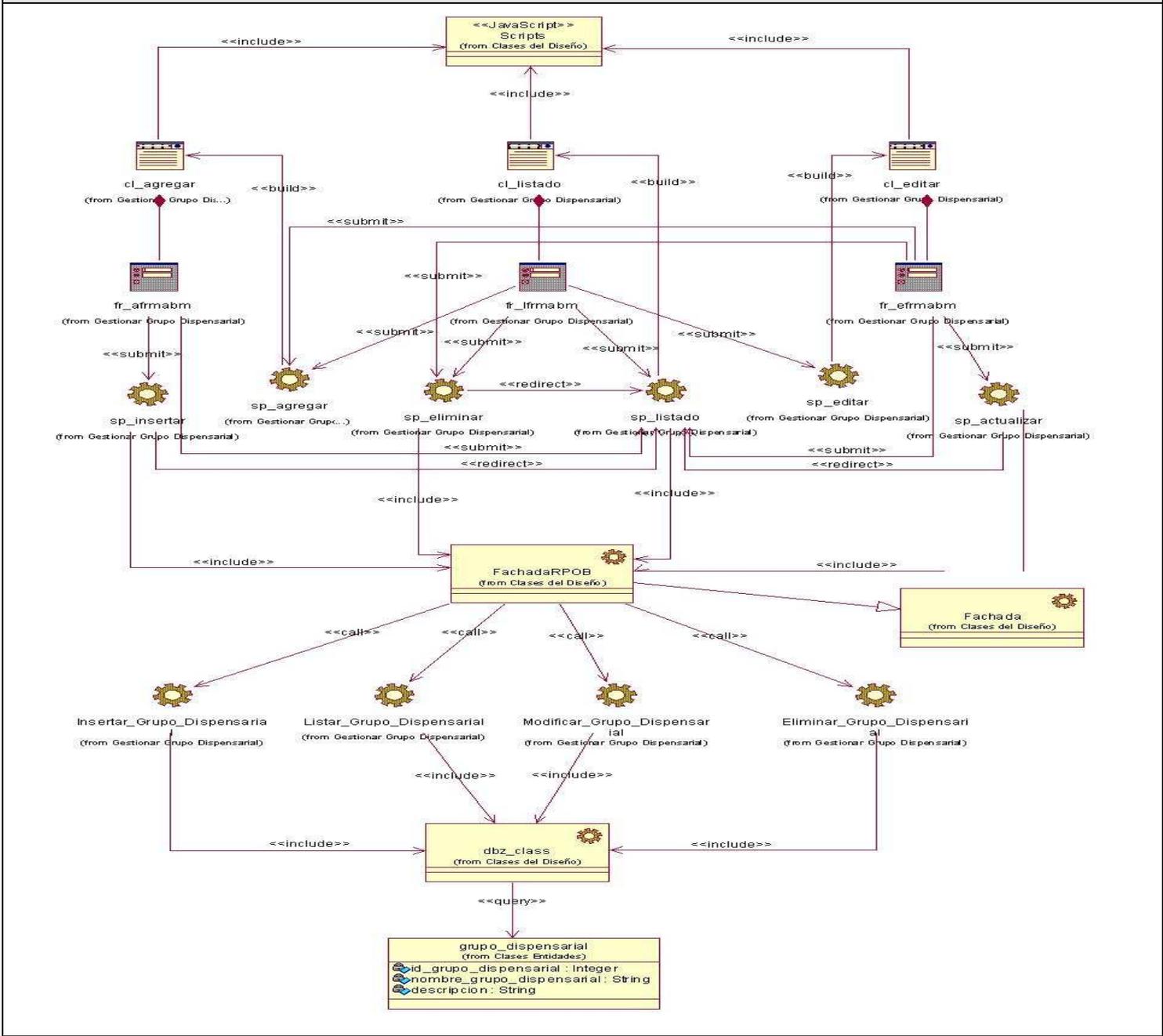
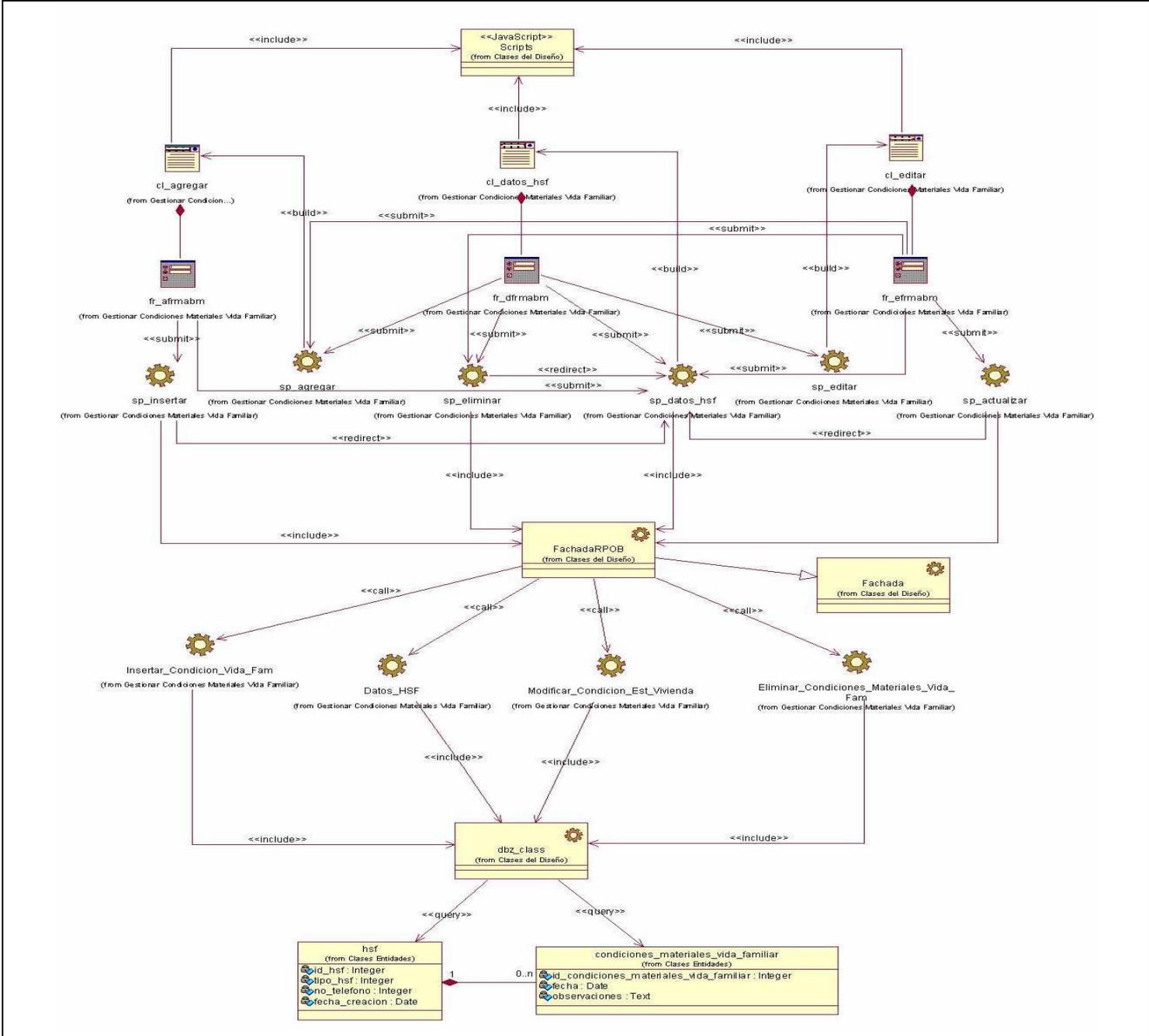


Diagrama de Clases de Gestionar Condiciones Materiales de Vida Familiar.



### 2.2.4 Descripción de las clases y atributos.

<b>Nombre</b>	<b>cl_listado</b>
<b>Tipo de Clase</b>	<b>Client Page</b>
<b>Descripción General</b>	
<p>La clase <i>cl_listado</i> es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite a los visualizadores o Editores Nacionales, Provinciales, Municipales o de Unidad de Salud la realización de listado y búsquedas de información específica permitiendo generar documentos Portable Document Format (.pdf) y Microsoft Office Excel (.xls). A través de esta página se puede visualizar la información y el resultado de las búsquedas es paginada, permitiendo la movilidad por tales resultados e inclusive ir directamente a la última. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Problemas de Salud.</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> </ul>	

**Tabla 2.2** Descripción de la clase cliente *cl\_listado*.

<b>Nombre</b>	<b>cl_editar</b>
<b>Tipo de Clase</b>	<b>Client Page</b>
<b>Descripción General</b>	
<p>La clase <i>cl_editar</i> es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite a los editores a nivel de Unidades de Salud la posibilidad de modificar la información específica. A través de esta página se puede agregar o eliminar la información relacionada. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Problemas de Salud.</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> </ul>	

**Tabla 2.3** Descripción de la clase cliente *cl\_editar*.

<b>Nombre</b>	<b>cl_agregar</b>
<b>Tipo de Clase</b>	<b>Client Page</b>
<b>Descripción General</b>	
<p>La clase <i>cl_agregar</i> es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite capturar los datos que serán insertados en la base de datos de la aplicación, posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Historia de Salud Familiar.</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> <li>❖ Gestionar Evaluación Salud Familiar.</li> <li>❖ Gestionar Intervención Familiar.</li> </ul>	

**Tabla 2.4** Descripción de la clase cliente *cl\_agregar*.

<b>Nombre</b>	<b>sp_listado</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Descripción General</b>	
<p>La clase <i>sp_listado</i> es una clase que se ejecuta del lado del servidor en la capa de presentación. Su función es construir la página cliente. Esta envía los datos necesarios a la capa de negocio, ya sea un listado o búsqueda, luego aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Historia de Salud Familiar.</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> <li>❖ Gestionar Problema de Salud.</li> <li>❖ Gestionar Intervención Familiar.</li> </ul>	

**Tabla 2.5** Descripción de la clase servidora *sp\_listado*.

<b>Nombre</b>	<b>sp_insertar</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Descripción General</b>	
<p>La clase <i>sp_insertar</i> es una clase que se ejecuta del lado del servidor en la capa de presentación. Su función es enviar los datos hacia la Capa de Negocio a través de la clase <i>FachadaRPOB</i>, la cual invoca al método del negocio para insertar los datos involucrados y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase <i>sp_insertar</i>.</p> <p>Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Historia de Salud Familiar.</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> <li>❖ Gestionar Problema de Salud.</li> </ul>	

**Tabla 2.6** Descripción de la clase servidora *sp\_insertar*.

<b>Nombre</b>	<b>sp_agregar</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Descripción General</b>	
<p>La clase <i>sp_agregar</i> es una clase que se ejecuta del lado del servidor en la capa de presentación. Su función es enviar los datos hacia la Capa de Negocio a través de la clase <i>FachadaRPOB</i>, la cual invoca al método del negocio para insertar los datos involucrados y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase <i>sp_agregar</i>.</p> <p>Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Historia de Salud Familiar.</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> <li>❖ Gestionar Problema de Salud.</li> </ul>	

**Tabla 2.7** Descripción de la clase servidora *sp\_agregar*.

<b>Nombre</b>	<b>sp_editar</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Descripción General</b>	
<p>La clase sp_editar es una clase que se ejecuta del lado del servidor en la capa de presentación. Esta envía los datos necesarios a la capa de negocio que se necesitan modificar, ya sea un listado o búsqueda, luego aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML con los datos ya modificados la cual hace una redirección a la clase sp_listado. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Intervención Familiar</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> <li>❖ Gestionar Problema de Salud.</li> </ul>	

**Tabla 2.8** Descripción de la clase servidora *sp\_editar*.

<b>Nombre</b>	<b>sp_eliminar</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Descripción General</b>	
<p>La clase sp_eliminar es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su actividad es recibir los valores de la página cliente cl_listado o cl_editar para posteriormente enviarlos hacia la Capa de Negocio a través de la clase FachadaRPOB, la cual invoca al método del negocio para eliminar los datos involucrados y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase sp_eliminar, la cual hace una redirección a la clase sp_listado. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Intervención Familiar</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> <li>❖ Gestionar Problema de Salud.</li> </ul>	

**Tabla 2.9** Descripción de la clase servidora *sp\_eliminar*.

<b>Nombre</b>	<b>sp_actualizar</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Descripción General</b>	
<p>La clase <code>sp_actualizar</code> es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su trabajo es recibir los valores de la página cliente <code>cl_editar</code> para posteriormente enviarlos hacia la Capa de Negocio a través de la clase <code>FachadaRPOB</code>, la cual invoca al método del negocio para modificar los datos involucrados y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase <code>sp_actualizar</code>, la cual hace una redirección a la clase <code>sp_listado</code>. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>❖ Gestionar Historia de Salud Familiar.</li> <li>❖ Gestionar Seguimiento Diario.</li> <li>❖ Gestionar Grupo Dispensarial.</li> <li>❖ Gestionar Condiciones Materiales de Vida Familiar.</li> </ul>	

**Tabla 2.10** Descripción de la clase servidora *sp\_actualizar*.

<b>CLASE</b>	<b>PROPÓSITO</b>
CDBZ	<p>Clase para la conexión con bases de datos MySQL que hace uso del módulo <code>dbx</code> de PHP para sus funcionalidades.</p> <p>Crea un objeto conexión que permite realizar consultas, y recuperar los resultados; insertar, eliminar y actualizar datos. Está localizada en la capa de acceso a datos y dentro de la plataforma de servicio <b>PLASER</b>, que es utilizada, por la empresa, para la reutilización de los distintos componentes y lograr una calidad adecuada, que implemente el patrón de diseño Modelo Vista Controlador (MVC).</p>
Fachada	<p>Clase general que se encuentra dentro de la plataforma de servicio <b>PLASER</b>, que permite aplicar un encapsulamiento al mismo, creándose de esta forma un alto nivel de abstracción que le da una mejor funcionalidad a la arquitectura utilizada.</p>
CFachadaRPOB	<p>Clase que hereda de <code>Fachada</code>, que aplica una fachada en la capa de presentación, disminuyendo así la carga de negocio en gran medida. Esta clase implementa el patrón <code>Fachada</code>, de esta forma la aplicación solo le hará las peticiones a esta clase.</p>

**Tabla 2.11** Descripción de las clases generales.

<b>Nombre</b>	<b>Buscar_Paciente</b>
<b>Tipo de Clase</b>	<b>Server page</b>
<b>Caso de Uso</b>	<b>Gestionar Paciente</b>
<b>Parámetros de Entrada</b>	
<b>Nombre</b>	<b>Tipo</b>
id_ciudadano	int
numeroHC	int
IdGrupoDispensarial	int
IdNivelEducativo	int
IdProfesionUOficio	int
estado_paciente	string
es_jefe_nucleo	int
id_grupo_etareo	int
offset	int
cantidad	int
<b>Descripción General</b>	
<p>La clase <i>Buscar_Paciente</i> es una clase que se ejecuta del lado del servidor en la Capa de Negocio. Su actividad es devolverle al usuario un listado de los integrantes de las poblaciones que atiende en las distintas historias de salud familiar, depende con los distintos parámetros de búsqueda.</p>	

**Tabla 2.12** Descripción de método del negocio *Buscar\_Paciente*.

<b>Nombre</b>	<b>Eliminar_Ingreso</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Caso de Uso</b>	<b>Gestionar Ingreso en el Hogar</b>
<b>Parámetros de Entrada</b>	
<b>Nombre</b>	<b>Tipo</b>
id_ingreso_hogar	Int
<b>Descripción General</b>	
<p>La clase <i>Eliminar_Ingreso</i> es una clase que se ejecuta del lado del servidor en la Capa de Negocio. Su función es eliminar de la base datos el ingreso en el hogar correspondiente con el identificador seleccionado.</p>	

**Tabla 2.13** Descripción de método del negocio *Eliminar\_Ingreso*.

<b>Nombre</b>	<b>Insertar_hsf</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Caso de Uso</b>	<b>Gestionar Historia de Salud Familiar</b>
<b>Parámetros de Entrada</b>	
<b>Nombre</b>	<b>Tipo</b>
num_hsf	int
tipo_hsf	string
fecha_creación	date
teléfono	int
id_dirección	int
<b>Descripción General</b>	
La clase <i>Insertar_hsf</i> es una clase que se ejecuta del lado del servidor en la Capa de Negocio. Su función es insertar la historia de salud en la base datos correspondiente con el tipo de historia y el resto de los datos.	

**Tabla 2.14** Descripción de método del negocio *Insetar\_hsf*.

<b>Nombre</b>	<b>Modificar_Grupo_Dispensarial</b>
<b>Tipo de Clase</b>	<b>Server Page</b>
<b>Caso de Uso</b>	<b>Gestionar Grupo Dispensarial</b>
<b>Parámetros de Entrada</b>	
<b>Nombre</b>	<b>Tipo</b>
id_grupo_dispensarial	int
nombre	string
descripción	string
<b>Descripción General</b>	
La clase <i>Modificar_Grupo_Dispensarial</i> es una clase que se ejecuta del lado del servidor en la Capa de Negocio. Su función es modificar el grupo dispensarial del paciente que se está dispensarizando.	

**Tabla 2.15** Descripción de método del negocio *Modificar\_Grupo\_Dispensarial*.

### 2.3 Conclusiones.

En este capítulo, se realizó el diseño de la solución propuesta por el analista que se dará a la futura aplicación, obteniéndose a partir del análisis de los procesos del negocio, un conjunto de funcionalidades que la misma tendrá, las cuales se representaron mediante un Diagrama de Casos de Uso. Finalmente se describieron paso a paso, todas las acciones que llevaron a cabo los actores del sistema. A partir de este punto se puede comenzar a construir la aplicación, teniendo en cuenta todos los requerimientos y funcionalidades que se derivaron de la etapa anterior.

## CAPÍTULO 3 IMPLEMENTACIÓN DEL SISTEMA

### 3.1 Introducción.

El objetivo principal de este capítulo es abordar los aspectos relacionados con la construcción e implementación de la solución propuesta, se brinda una detallada justificación de la integración con otros módulos, se modelan los diagramas de componentes y diagrama de despliegue, se describen los estándares de diseño, codificación y además el tratamiento de errores para la solución del sistema.

### 3.2 Justificación de Integración con otros Módulos.

El módulo en desarrollo, nombrado Registro de Población (RPOB), es una aplicación que según estudios realizados no se ha desarrollado antes con este nivel de integración. Existen antecedentes del sistema relacionado con el tema, que no se integran al Sistema de Información para la Salud (SISalud). Sin embargo la nueva aplicación RPOB cumple con las necesidades de integración con otros sistemas, ya que se propone su desarrollo con una arquitectura basada en componentes y orientada a servicios, utilizando servicios web de otros componentes ya implementados para su funcionamiento. Los mismos son: Registro de Áreas de Salud (RAS), Registro de Problemas de Salud de Atención Primaria (RPSAP) y Registro de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud (RCIE).

Además por ser esta una aplicación basada en servicios web se hace muy fácil su integración, aclarando que deben estar en funcionamiento los componentes antes mencionados. Para el desarrollo de la misma se usa SOAP (Simple Object Access Protocol), aumentando su portabilidad, ya que no se ata a un lenguaje de programación específico. Con el uso de SOAP la interoperabilidad se hace más ligera, al hacer uso de XML para el transporte de datos. Esto provee grandes ventajas a la hora de la integración con otros sistemas externos, ya que todo el intercambio de información es a través de XML y teniendo en cuenta esta tecnología es soportada por la mayoría de las plataformas de desarrollo.

### **3.2.1 Componente de Seguridad (SAAA).**

Está basado en el modelo de Autenticación, Autorización y Auditoría (**AAA**). La autenticación debe ser la primera acción del usuario en el sistema y consiste en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica. Si el usuario autenticado no se encuentra registrado se reporta un error de acceso. En caso contrario, se autoriza su acceso y se crea un certificado digital y se retornan todos los datos y permisos del usuario, desglosado por módulos. Cada Petición de usuario, autorizada o no, es registrada, así como el día, mes, año, hora, minuto, segundo en que se registra y si fue o no autorizada.

### **3.2.2 Registro de Áreas de Salud (RAS).**

Gestiona la información de las Áreas de Salud a nivel nacional, permitiendo un control de las mismas, su composición según la estructura organizativa propuesta por la Atención Primaria de Salud, así como sus integrantes según la plantilla del Grupo Básico de Trabajo y Equipo Básico de Salud. Brinda a la población información actualizada sobre las características de las Unidades de Salud que brindan servicios, dónde se ubican, áreas geográficas que atiende, servicios que brinda, especialidades que ofrecen sus servicios. Teniendo en cuenta que el personal de salud ya está ubicado en un área de salud y pertenece a un EBS, el médico puede seleccionar la dirección en la cual está ubicada la vivienda.

### **3.2.3 Registro de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud (RCIE).**

Gestiona la estructura de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud y permite realizar búsquedas dinámicas de acuerdo a criterios seleccionados por el usuario. Ofrece un lenguaje unificado y estándar para la información de la morbimortalidad del país. Actualmente se encuentra vigente la décima edición (CIE-10). Este brinda las posibles causas de ingresos en el hogar de un paciente al EBS.

### **3.2.4 Registro de Problemas de Salud de la Atención Primaria (RPSAP).**

Gestiona una Clasificación Especial para los problemas de salud que se presentan en el nivel de Atención Primaria de Salud, tratando que resulte una clasificación más breve que la CIE-10, de manejo más simple, dada la naturaleza de tales problemas. Permite realizar búsquedas dinámicas de acuerdo a criterios seleccionados por el usuario. Ofrece un lenguaje unificado y estándar para la información de la morbilidad en la Atención Primaria de Salud. Este componente le brinda al RPOB todo lo necesario para la realización del proceso de dispensarización, así como la planificación de acciones de salud.

### **3.2.5 Registro de Ciudadano (RC).**

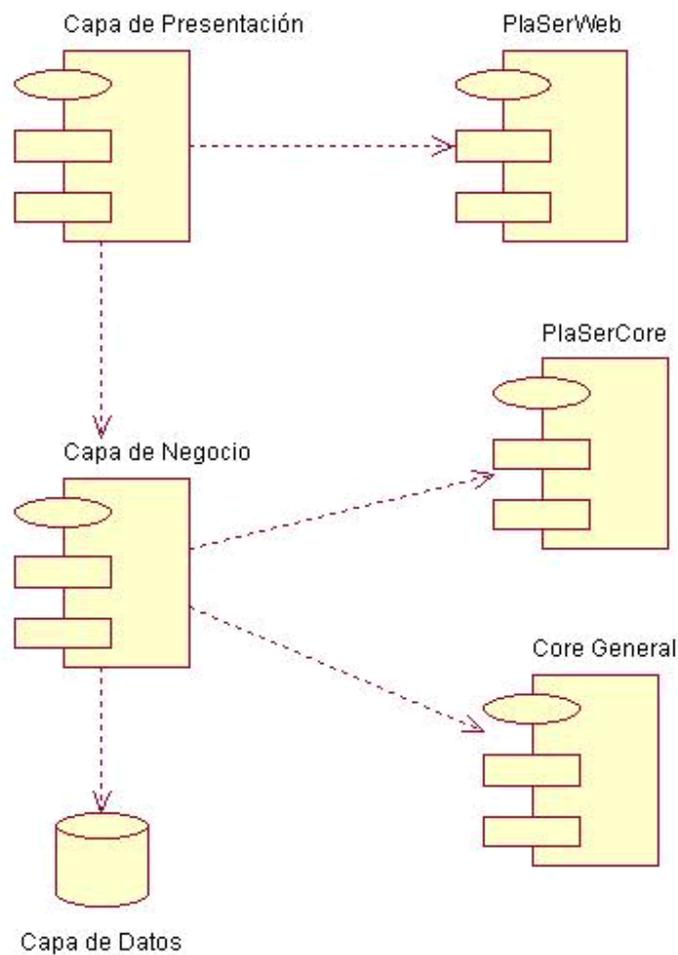
Su función es que exista un registro único con los datos personales de cualquier ciudadano que trabaje en la salud y/o reciba sus servicios. Los datos fundamentales están relacionados con la información que se recoge por el Carné de Identidad y se podrán ir adicionando otros que sean necesarios para los diferentes componentes que necesiten conectarse a este registro para buscar los datos de una persona. Este componente les brinda al RPOB toda la información referente a los ciudadanos para los procesos de gestión de la información de los integrantes de las historia de salud familiar.

## **3.3 Modelo de Implementación.**

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo tipo de ficheros necesarios para la implantación y despliegue del sistema.

### **3.3.1 Diagrama de Componentes.**

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente. [33]



**Figura 3.1** Diagrama de Componentes.

### 3.3.2 Diagrama de Despliegue.

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. [34]

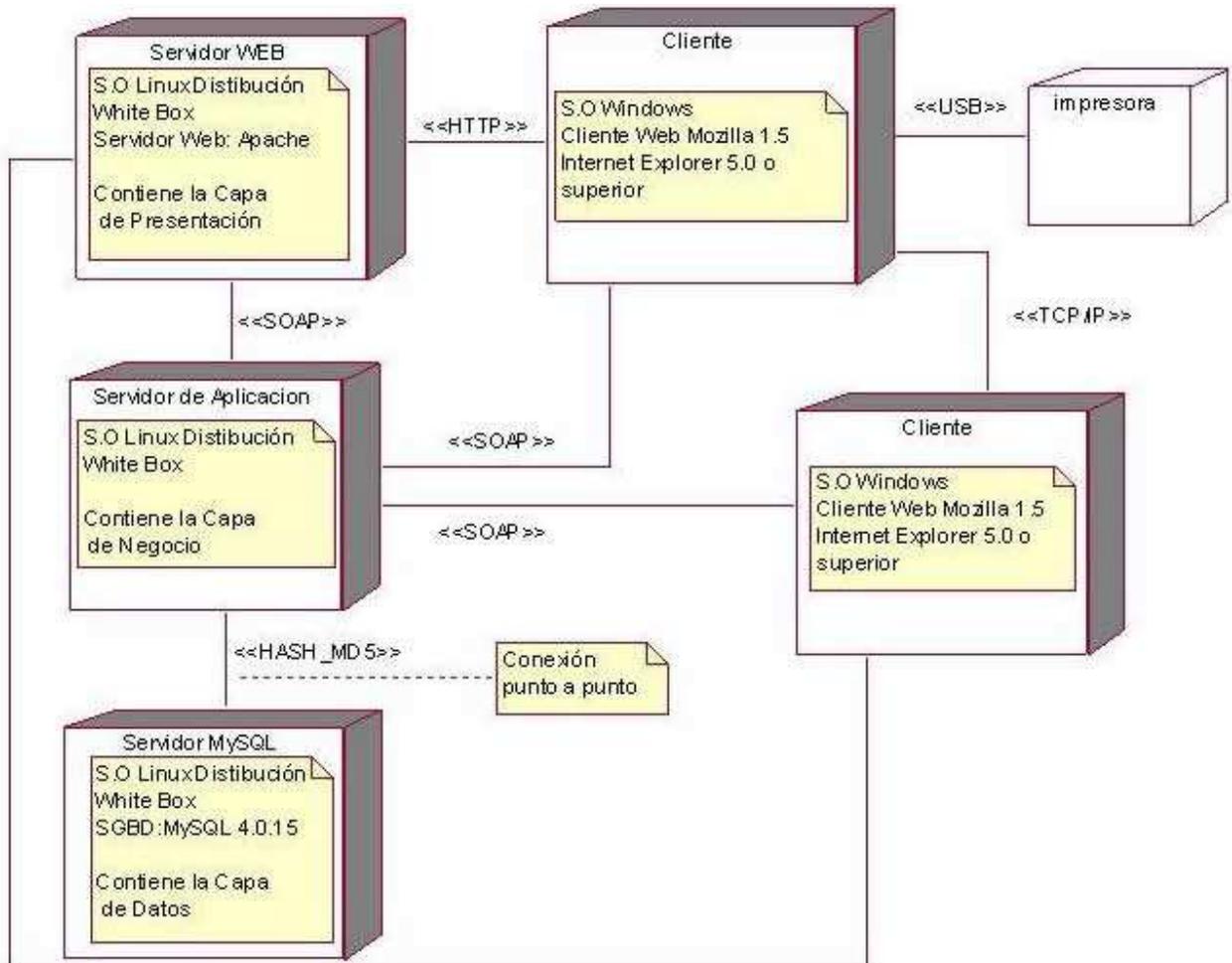


Figura 3.2 Diagrama de Despliegue.

### 3.4 Descripción de los Métodos más complejos.

#### Método Buscar\_Paciente

El método realiza una búsqueda de todos los pacientes que atiende el médico autenticado cumpliendo con los demás parámetros de entrada. Recibe como parámetros el Carné de Identidad, nombre, primer

apellido, segundo apellido, sexo, número de historia clínica individual, nivel educacional, número de historia de salud familiar, profesión u oficio, labor que realiza, si es o no jefe de núcleo, grupo dispensarial, un grupo de edad, población a la que pertenece el paciente y la unidad de salud a la que pertenece el médico (los dos últimos son obligatorios), si los parámetros son pasados vacíos el método devuelve todos los pacientes a los cuales tiene acceso el médico que se autentica.

Para realizar esta búsqueda, se construye una consulta dinámica, en dependencia de los parámetros de entrada. Se consulta la tabla de los pacientes en el módulo RPOB y se obtienen los identificadores de los ciudadanos que retornó la consulta realizada y con estos se invoca un servicio web del Registro de Ciudadano (RC), *BuscarCiudadano* donde se obtienen los datos personales de los pacientes.

Para conformar la respuesta final se concatenan las respuestas del módulo población con la del RC y se retornan los todos atributos del paciente unido con sus respectivos datos personales.

### **Método Datos\_HSF**

El método obtiene lo que constituye la Historia de Salud Familiar (HSF) con todos sus datos. Ellos son: todos los datos personales de los pacientes que pertenecen a la familia, además de la edad y otros datos importantes como nivel educacional, profesión u oficio, labor que realiza, y grupo dispensarial al que pertenece. Además obtiene todas las evaluaciones de funcionamiento familiar, las condiciones estructurales de la vivienda, las evaluaciones de salud familiar hechas y las intervenciones familiares que se han realizado a la familia. Para ello el método recibe el número de HSF que es único. También se utiliza un identificador de la unidad de salud a la que pertenece el médico autenticado para determinar el nombre de la unidad. Para esto se invoca un servicio web, *Buscar\_Total*, que obtiene todos los atributos de esta unidad, pero solo se toma su nombre.

Para la obtención de la dirección física de la vivienda a la que está asociada la HSF se consulta la tabla de HSF y se obtiene un identificador de dicha dirección y con esto se invoca el servicio web, *AgenteBuscarDireccion*, del Registro de Áreas de Salud (RAS) para obtener el texto legible de la dirección particular. Para la obtención del Grupo Básico de Trabajo y del Equipo Básico de Salud se invoca otro servicio del RAS llamado *ListarDerechos*, que devuelve una estructura con dicha información.

Para la obtención de los datos, tanto personales, como de los atributos del módulo RPOB realiza una consulta donde se obtienen los identificadores de dichos pacientes, que constituyen los identificadores en el RC, es invocado el servicio web *BuscarCiudadano*, y la respuesta de éste es concatenada con la información brindada por RPOB conformándose la respuesta necesaria. Luego para obtener los demás datos de la HSF se consulta la Base de Datos (BD) de RPOB y con todas las respuestas de las consultas realizadas, tanto a la BD local como a todos los componentes externos se construye una respuesta única que es retornada.

### 3.5 Estándares de diseño, codificación y tratamiento de errores.

#### 3.5.1 Estándares de Diseño.

Para lograr una mayor eficiencia en el proceso de trabajo, y sobre todo para lograr una coherencia formal entre todos los módulos del sistema, y que sean identificados así como parte de un todo, se han pautado una serie de elementos comunes que facilitarán su reconocimiento y el uso que se haga de ellos.

Se diseñará una Pantalla Inicial global del **Sistema de Información para la Salud**, desde la cual se accederá a los diferentes módulos del RIS, del SIAP, del SIGH y SIAE. Esta pantalla contará con accesos a los diferentes módulos, informaciones generales, guías de ayuda, sistema de avisos que genera cada registro y enlaces definidos.

Así mismo será diseñada una Pantalla Inicial para cada una de las aplicaciones, que contará con accesos a todas las utilidades, avisos, ayuda y un enlace para regresar a la Pantalla Inicial del **Sistema de Información para la Salud**.

La estructura base de las aplicaciones es la misma para todos los módulos: las pantallas más usadas, los modelos establecidos, las rutas de navegación, las utilidades básicas, la organización de los elementos en pantalla y el diseño de identificadores serán comunes para todos.

Para particularizar el diseño de cada módulo se ha definido entonces una pauta de dos colores básicos en este azul y gris entre otros colores, con sus degradaciones hacia blanco y negro, así como la diferenciación por logotipo e imagen principal del cabezal, que identificará a cada módulo.

Su diseño está determinado fundamentalmente por el principio de la usabilidad, teniendo en cuenta que no se trata de un sitio web, sino de una aplicación de trabajo donde el diseño tiene como principal propósito facilitar su uso, comprensión y navegación, por encima de ornamentos inútiles, aunque manteniendo pautas estéticas, orgánicas y agradables.

Formalmente, usabilidad se define como la medida en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción, en un contexto de uso especificado [ISO 9241-11].

La resolución óptima para la cual están diseñadas las aplicaciones es de 1024 x 768 px. El fondo siempre será blanco y los elementos de pantalla de los colores definidos para cada módulo.

Se ha definido un cabezal pequeño de 65 px de altura, más pequeño que el utilizado en las páginas Web, que recomiendan cabezales de hasta 80 px de altura.

El menú principal siempre estará situado en una barra izquierda vertical. No existirá barra vertical de menú situada a la izquierda de la página (como usualmente se hace) para ampliar el espacio de trabajo, pues estará reservado lo más amplio posible para la inserción de grandes tablas y formularios que constituyen la base fundamental de estas aplicaciones.

El logo siempre estará ubicado en el extremo superior izquierdo de la página, es una imagen que cuenta con un ancho de 270 px y se corresponde con el nombre de cada módulo. Estará constituido por un juego tipográfico en *Frankling Gothic Medium*, y en el caso de las aplicaciones propias del Proyecto APS, estando especificado dentro del logo como una especie de genérico.

La tipografía será siempre Tahoma, por su amplia legibilidad y por las facilidades conocidas que brinda para la lectura digital. El menú principal será a 7 ptos y los submenús a 6 ptos. Los demás puntajes se definirían en dependencia de las necesidades puntuales de cada pantalla.

El espacio de trabajo comienza 33 px por debajo del menú. El espacio intermedio que queda es también con fondo blanco y está reservado para el texto de ubicación dentro del sitio (justificado a la izquierda) y para ubicar los botones propios de la pantalla (justificados a la derecha). Estos se organizarán en una o dos filas, de hasta cuatro botones (13 x 72 px) cada una. Los botones se corresponden también con los colores pautados.

Entre los elementos comunes del menú principal se encuentran *Inicio* para regresar a la página inicial del módulo, *Salir* para desconectarse del sistema, y *Otros Módulos* para facilitar los enlaces a otros módulos necesarios. Es también común a casi todos los botones del menú principal en el lado izquierdo *Configurar* para la configuración de codificadores, *Cierre* para la realización de cierre estadístico y *Reportes* para generar reportes de actividades u operaciones.

Es común para todos los módulos el diseño de una serie de ventanas, en las que solo cambiarían los colores, en dependencia de cada uno. Son estas las ventanas de precaución, error, validación de datos, etc.

En cuanto a los elementos de diseño del interior de las pantallas, es decir, de las tablas, formularios, etc., se definen los edit que se utilicen con una altura de 16 px y la separación entre estos y entre ellos y los bordes de tablas será de 8 px. Será de 8 px la separación entre el texto y el edit. Los textos de estos campos serán justificados siempre a la derecha, es decir, justificados a 8 pto de cada edit.

En el caso de tablas generadas por búsquedas, que ordenan una serie de elementos, y necesiten selección, se harán a través de checkboxes justificados a la izquierda de la tabla. Siempre habrá un checkbox en la fila de título, también a la izquierda, que facilite *seleccionar todos*. Es necesario destacar que estas tablas pueden tener una cantidad grande de líneas generadas por la búsqueda, por lo que debe quedar pautado que hasta 10 resultados la tabla funcione con scroll, pero más de esta cantidad será entonces por paginado, con un *select*, con la cantidad de páginas hasta llegar al total de resultados de la búsqueda.

Existen detalles que serán definidos particularmente en cada uno de los módulos, ya que satisfacen a necesidades específicas de los mismos.

El interés general es mantener el diseño y la estructura del sitio lo más simple posible, la simplicidad es entendimiento del contenido, de la estructura, es facilidad para encontrar lo que se busca, es también velocidad de descarga.

### 3.5.2 Tratamientos de Errores.

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. Son una forma clara para controlar los errores sin confundir el código con muchas instrucciones de control del error. Cuando se verifica un error se pone en marcha una excepción que, si se recibe enseguida, permite gestionar un error.

Durante la ejecución, en las clases pueden provocarse errores de diferentes tipos y diversos grados de gravedad. Cuando se invocan métodos sobre un objeto, se puede encontrar con problemas internos de estado (valores incongruentes), detectar errores con los objetos o datos que manipula (como la dirección a un archivo o red), querer acceder sobre un archivo ya cerrado u otros problemas. Proporcionan una manera de verificar los errores y poder controlarlos si fuera el caso si abortar el código.

Las excepciones son condiciones excepcionales que pueden ocurrir dentro del programa durante su ejecución (por ejemplo una división por cero, se agote la memoria disponible, que se pierda la comunicación, que no se produzca el resultado esperado ante alguna petición, etc.) y que requieren recursos especiales para su control.

La correcta programación de excepciones significa diseñar los algoritmos pensando únicamente en la forma habitual en la que deben ejecutarse, manejando las situaciones extraordinarias a parte. De esta manera se consigue un diseño mucho más estructurado, legible, robusto y fácil de mantener.

Las excepciones serán tratadas en la capa de negocio con un código único y entendible, que será enviado a la capa intermedia (middleware), el código se devolverá en un Soap\_fault cuyos elementos son faultCode, faultString, faultActor, y detail.

El **faultCode** es un código texto usado para indicar la clase del error, codificado de la siguiente forma: Código del proyecto - código del módulo, APS - RPOB: 1.5, este indica error 5 en el método 1 del módulo Registro de Población y perteneciente al proyecto Atención Primaria de Salud.

El **faultString** es una explicación humanamente legible del error. Debe tenerse en cuenta que este texto pudiera ser mostrado al operador final del sistema.

Ejemplo: Formato de entrada no válido para la fecha de historia de salud familiar.

El **faultActor** es un texto que indica quien provocó el error, en este caso siempre será el nombre del método que eleva la excepción.

Ejemplo: Modificar\_Grupo\_Dispensarial

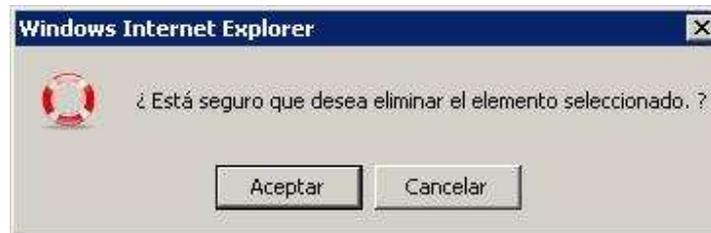
El **detail** se usa para llevar mensajes de error específicos de aplicaciones, en este caso será empleado únicamente en errores cuya resolución depende del Centro de Control, en cualquier otro caso este elemento debe estar vacío. **(Ver Anexo 4)**

Existen errores que pueden evitarse en la capa de presentación, serán tratados mediante funciones del lenguaje Java Script *del lado del cliente*, a través de mensajes de alerta.



**Figura 3.3** Ejemplo de un mensaje de alerta.

También serán tratadas las validaciones a la hora de eliminar algún elemento del lado del cliente, mostrado en el navegador una ventana de si está seguro queriendo eliminar el elemento seleccionado. Si se acciona el botón *Aceptar* se procede a la acción, de lo contrario no se ejecuta.



**Figura 3.4** Ejemplo de un mensaje de alerta al eliminar un elemento.

### 3.5.2 Estándares de codificación.

Actualmente se hallan estándares de codificación para la mayoría de los lenguajes existentes. El uso de ellos partiendo de las convenciones definidas permite una mejor comunicación entre los programadores creando las condiciones para la reusabilidad y el mantenimiento de los sistemas. Para definir el estilo de codificación a seguir en la aplicación se utilizó la notación estándar establecida para aplicaciones desarrolladas en PHP (PHP Coding Standard), que mayormente está basada en el estándar de código para aplicaciones en C++ ( C++ Coding Standard) que se puede representar como [COD03].

Las etiquetas de apertura y cierre del lenguaje serán de la forma `<?php ?>`, ya que siempre están disponible en cualquier configuración.

Se harán uso de los arreglos predefinidos para el manejo de los valores enviados por el usuario `$_GET`, `$_POST`, `$_FILES` evitando el uso de `$_REQUEST`.

Para nombrar las variables se seguirá la regla de escribir los identificadores con letras minúsculas y en español, utilizando como separador para las palabras el carácter “\_” tratando de usar nombres sugerentes a la acción de la variable.

Todos los campos id van a comenzar con el identificador (id) seguido del nombre del campo. Ejemplo id \_ grupo.

Los arreglos empezarán con el identificador array y las palabras no se separaran con el carácter “\_”. (Ejemplo \$args (id\_direccion))

Con el propósito de distribuir los esfuerzos y mejorar los rendimientos de la aplicación, se utilizó la programación orientada a servicios, creando clases genéricas que permiten la definición y en el caso de las clases se pondrá delante la letra C. Ejemplo: CFachada y en el de los métodos no se usarán abreviaturas y las palabras continuas deben comenzar con mayúsculas. Ejemplo: ListarProblemasSalud.

Para comentar el código se utilizará, en el caso de una línea, al final de la misma el carácter “//” y seguido el comentario y en el caso de un bloque se utilizará los caracteres “/\* \*/”.

Se usará una indentación en el código de cuatro espacios para facilitar la lectura de éste. Las llaves se usarán poniendo la llave inicial en una línea para ella sólo, y en su respectiva columna la llave final también en una línea.

El idioma de las clases auxiliares como sesión y error, será el inglés para garantizar la homogeneidad con las programadas en este ámbito en el mundo, en el caso de los Servicios Web y la interfase de administración se usará el español para esclarecer los objetivos de cada método o script a utilizar.

Para lograr que las comparaciones sean seguras, se colocarán siempre los valores constantes a la izquierda de la comparación "if (6 == \$variable)", con esto garantizará la generación de un error cuando por error escriba '=' y no '=='. Se utilizará el operador “?” para sentencias cortas, preferiblemente que ocupen una sola línea. La sentencia **switch** siempre tendrá la opción **default** y se evitará el uso de **continue** y **break**, ya que podrían perder la vista lógica del código fuente.

El almacenamiento de la información será en scripts SQL para construir la base de datos e interactuar con ella desde las aplicaciones.

Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas (Ejemplo: SELECT \* FROM tb\_pacientes)

Los nombres de las tablas deben ir en minúsculas y cada palabra separada por línea abajo "\_". (Ejemplo: tb\_nombre\_tabla)

En el caso de los XSL será con el mismo nombre que el fichero de la capa de presentación.

Los controles seguirán el siguiente tratamiento:

Control	Prefijo	Ejemplo
Botón	Btn	btnAceptar
Etiqueta	Lbl	lblNombre
Lista/Menú	Mn	mnPrincipal
Campo de Texto	Txt	txtFecha
Botón de Opción	Opt	optSexo
Casilla de Verificación	Chx	chxBorrar

**Tabla 3.1** Estándares para los controles.

Las páginas HTML se harán sin incluir código y todas las funciones JavaScript que se usarán se escribirán dentro de ficheros “.js”.

Para la capa de datos tienen que nombrar la base de datos poniendo el identificador del proyecto “APS” seguido del carácter “\_” y del nombre del módulo. (Ejemplo: bd\_rpob).

Los campos de la base de datos se nombrarán igual que las variables.

Cada método que se defina debe seguir la siguiente estructura, acrónimo del módulo. Nombre del método. (Ejemplo: RPOB\_InsertarPaciente).

### 3.6 Conclusiones.

En este capítulo se efectuó la justificación de la integración con los distintos componentes del sistema, de los cuales se invocan los servicios para el desarrollo del módulo. Se realizaron los distintos modelos del flujo de trabajo Implementación, así como las distintas pautas seguidas para el diseño de interfaz y los métodos usados para los tratamientos de errores.

## CONCLUSIONES

La aplicación desarrollada, cuyo diseño e implementación se refleja en el documento, ha tenido como objetivo fundamental, diseñar e implementar una aplicación Web que facilite al Equipo Básico de Salud, gestionar la información de la Historia de la Salud Familiar asociada, a la población que atiende.

- 📄 Se investigó sobre sistemas existentes, a nivel internacional y nacional, que pudieran resolver el problema planteado y se estudiaron las tendencias, tecnologías y herramientas a utilizar para el desarrollo de la solución propuesta.
- 📄 La aplicación se realizó sobre la base del proceso de desarrollo unificado RUP y el lenguaje de modelado UML como apoyo para la realización de los flujos de trabajo del diseño e implementación.
- 📄 Se utilizó para la implementación del módulo Registro de Población la arquitectura definida por el MINSAP (Orientada a Servicios y Basada en Componentes).
- 📄 Se desarrolló la aplicación web utilizando la Plataforma de Servicio **PLASER**.
- 📄 Se implementaron los Servicios Web del módulo Registro de Población lo que agilizará todo el proceso de gestión de la información referente a la familia.

## RECOMENDACIONES

Una vez cumplido con el objetivo general propuesto y en vista a que se incrementen las funcionalidades brindadas por el sistema, así como posibilitar el uso del mismo, se recomienda:

- 🖥️ Desplegar la aplicación en INFOMED para que sea utilizada en los 4 policlínicos del piloto del municipio Cerro.
- 🖥️ Realizar el desarrollo de los distintos reportes de RPOB en la próxima iteración del módulo, que contribuyan a la toma de decisiones en los distintos niveles de dirección.
- 🖥️ Incluir en la concepción del componente de avisos de SISalud los mensajes generados por el Registro de Población.
- 🖥️ Publicar los servicios web de RPOB para que puedan ser consumidos por cualquier aplicación informática que los necesite.
- 🖥️ Definir y poner en práctica el plan de capacitación necesario para que el personal de salud de las diferentes unidades asistenciales se familiaricen con el sistema.
- 🖥️ Crear las condiciones de configuración en los módulos que se relacionan con el Registro de Población.

## REFERENCIAS BIBLIOGRÁFICAS

1. CUBAMINREX, *Derechos humanos en Cuba*. [En Línea ][Citado el : 11 de febrero de 2007] Disponible en: [www.cubaminrex.cu/Enfoques/ddhh\\_salud\\_tc.htm](http://www.cubaminrex.cu/Enfoques/ddhh_salud_tc.htm).
2. Díaz, Novás J. y Fernández, Sacasas J. Del Policlínico Integral al médico de la familia. *Rev. Cubana Medicina General Integral* 1989; 5(4):556-564.
3. Idem a la referencia 2.
4. Idem a la referencia 1.
5. Castro Ruz, Fidel. *Discurso pronunciado por el Presidente de la República de Cuba, en la Tercera Graduación del Contingente del Instituto Superior de Ciencias Médicas de la Habana*. Teatro "Carlos Marx". Ciudad de la Habana. 27 de agosto de 1990.
6. Colectivo de autores cubanos. Carpeta metodológica de Atención Primaria de Salud y Medicina Familiar. Capítulo Dispensarización: página 44-47, Barcelona 2000.
7. Idem a la referencia 2.
8. Ministerio de Salud Pública. MINSAP. *El cuidado de la salud en Cuba*. Escuela Nacional de Salud Pública, 2003.
9. Castro Ruz, Fidel. *Discurso pronunciado en la Clausura del VI Seminario Internacional de Atención Primaria*, Ciudad de la Habana. 28 de noviembre de 1997. Disponible en: <http://www.cuba.cu/gobierno/discursos/1997/esp/f281197e.html>.
10. Delgado Ramos, Ariel y Vidal Ledo, María. <http://bvs.sld.cu>. [En línea] 2006. [Citado el: 02 de febrero de 2007.] [http://bvs.sld.cu/revistas/spu/vol32\\_3\\_06/spu15306.htm#cargo](http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm#cargo).
11. Cabrera Hernández, Mirna et al. Organización del proceso productivo del Proyecto Atención Primaria de Salud (APS).XVI Forum de Ciencia y Técnica. Ciudad Habana. Julio 2006.
12. Idem a la referencia 11.
13. Idem a la referencia 6.
14. Marín Díaz, Miguel E. et al. *Manual de Fundamentos del Sistema Nacional de Salud*. Ciudad Habana. Noviembre 2005.
15. Cabrera Hernandez, Mirna. et al. *Propuesta de Esquema Sistema Integral de Salud (SiSalud)*. La Habana: s.n., 2006.

16. Martínez Calvo, Silvia y Gómez de Haz, Héctor. *Análisis de Situación de Salud*. Escuela Nacional de Salud Pública, 2003
17. Idem a la Referencia 6.
18. Idem a la Referencia 6.
19. CEDISAP. INFORMÁTICA MÉDICA. *APUS: Sistema Informático para la Gestión Médica en la APS*. [En Línea] [Citado el: 24 de junio del 2007]. Disponible en: <http://www.sld.cu/instituciones/cedisap/atepri1.htm>.
20. Idem a Referencia 10.
21. Ethan. Cerami, *Web Services Essentials* (Primera edición ed.). O'Reilly. Pag 20 – 24.
22. Comunidad de Programción, de MYGNET. Introducción a la arquitectura Cliente/Servidor. [En Línea]. [Citado el: 20 de mayo de 2007] Disponible en: <http://www.mygnet.com/articulos/vb/82/>.
23. Mohammed J. Kabir, *La biblia Servidor Apache 2*, pág. 41-43.
24. Alvarez, Rubén. *DESARROLLO WEB. Introducción a la programación en ASP*. [En Línea] [Citado el: 03 de junio de 2007] Disponible en: <http://www.desarrolloweb.com/articulos/244.php>.
25. PROGRAMACIÓN EN CASTELLANO. *Java en Castellanos*. [En Línea]. [Citado el: 13 de febrero de 2007] Disponible en: <http://www.programacion.net/java/>.
26. Alvarez, Miguel. *DESARROLLO WEB. Que es JavaScript?* [En línea] [Citado: el 13 de febrero de 2007]. Disponible en: <http://www.desarrolloweb.com/articulos/25.php>.
27. Merelo, Juan. *XSLT. Tutorial de XSL*. [En línea] [Citado: el 13 de febrero de 2007] Disponible en: <http://geneura.ugr.es/~jmerelo/XSLT/>.
28. Riordan, M. Rebecca. *Aprenda Programación en Microsoft SQLServer 2000 YA*. [Consultado el: 26 de marzo de 2007]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg01713.pdf>.
29. *POSTGRESQL COMUNITARIO MÉXICO. ¿Que es PostgreSql?* Última actualización: Noviembre 2005. [Citado el: 29 de marzo de 2007]. Disponible en: <http://www.postgresql.org.mx/?q=node/6>.
30. Gilfillan, Ian. *La Biblia MySql*. Multimedia Anaya. ISBN: 8441515581. Pág. 33- 40.
31. Jacobson, Ivar y Booch, Grady y Rumbaugh, James. *El proceso unificado de software*. Primera edición. Pearson Educación, S.A. 2000
32. TLDP-ES. *Modelado de Sistemas con UML*. [En Línea]. [Citado el: 29 de marzo de 2007]. Disponible en: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/index.html>.

33. Rey Jimenez, Alexandra. CONSULTORÍA DE SEGURIDAD. Que es UML? : Diagrama de Componentes, [En Línea] [Citado el: 01 de junio de 2007]. Disponible en : <http://www.creangel.com/uml/componente.php>
34. Rey Jiménez, Alexandra. CONSULTORÍA DE SEGURIDAD. Que es UML? : Diagrama de Despliegue. [En Línea] [Citado el: 01 de junio de 2007]. [Disponible en: <http://www.creangel.com/uml/despliegue.php>].

## BIBLIOGRAFÍA

**Álvarez, Miguel.** DESARROLLO WEB. *Que es JavaScript?* [En línea] [Consultado: el 13 de febrero de 2007]. [Disponible en: <http://www.desarrolloweb.com/articulos/25.php>].

**Álvarez, Rubén.** DESARROLLO WEB. *Introducción a la programación en ASP.* [En Línea] [Consultado el: 03 de junio de 2007] [Disponible en: <http://www.desarrolloweb.com/articulos/244.php>]

**Castro Ruz, Fidel.** *Discurso pronunciado por el Presidente de la República de Cuba, en la Tercera Graduación del Contingente del Instituto Superior de Ciencias Médicas de la Habana.* Teatro "Carlos Marx". Ciudad de la Habana. 27 de agosto de 1990.

**Castro Ruz, Fidel.** *Discurso pronunciado en la Clausura del VI Seminario Internacional de Atención Primaria,* Ciudad de la Habana. 28 de noviembre de 1997. [Disponible en: <http://www.cuba.cu/gobierno/discursos/1997/esp/f281197e.html>]

**Castro Ruz, Fidel.** *Discurso pronunciado en el acto de inauguración de obras del extraordinario programa de salud ya en marcha, que se lleva a cabo en Cuba, efectuado en el teatro Astral,* el 7 de abril del 2003.

**Castro Ruz, Fidel.** *Discurso pronunciado por el Presidente de la República de Cuba, en el acto conmemorativo del aniversario 40 del Instituto de Ciencias Básicas y Preclínicas Victoria de Girón.* El 17 de octubre de 2002.

**CEDISAP. INFORMÁTICA MÉDICA.** *APUS: Sistema Informático para la Gestión Médica en la APS.* [En Línea] [Consultado el: 24 de junio del 2007]. [Disponible en: <http://www.sld.cu/instituciones/cedisap/atepri1.htm>].

**Colectivo de autores cubanos.** Carpeta metodológica de Atención Primaria de Salud y Medicina Familiar. Capítulo Dispensarización: página 44-47, Barcelona 2000.

**CUBAMINREX.** *Derechos humanos en Cuba.* [En Línea] [Consultado el: 11 de febrero de 2007] [Disponible en: [www.cubaminrex.cu/Enfoques/ddhh\\_salud\\_tc.htm](http://www.cubaminrex.cu/Enfoques/ddhh_salud_tc.htm)].

**Díaz, Novás J. y Fernández, Sacasas J.** Del Policlínico Integral al médico de la familia. *Rev. Cubana Medicina General Integral* 1989; 5(4):556-564.

**Ethan, Cerami.** *Web Services Essentials* (Primera edición ed.). O'Reilly. Pág 20 – 24.

**Gilfillan, Ian.** *La Biblia MySql.* Multimedia Anaya. ISBN: 8441515581. Pág. 33- 40.

**Jacobson, Ivar y Booch, Grady y Rumbaugh, James.** *El proceso unificado de software.* Primera edición. Pearson Educación, S.A. 2000.

**Martínez Calvo, Silvia y Gómez de Haz, Héctor.** *Análisis de Situación de Salud.* Escuela Nacional de Salud Pública, 2003

**Ministerio de Salud Pública, MINSAP.** *El cuidado de la salud en Cuba.* Escuela Nacional de Salud Pública, 2003.

**Mohammed J. Kabir,** *La biblia Servidor Apache 2,* pág. 41-43.

**Revista Habanera de Ciencias Médica.** *Consideraciones sobre el Proyecto de Informatización de la Atención Primaria de Salud.* Volumen 3, No. 10, año 2004. [Disponible en: [http://www.ucmh.sld.cu/rhab/editorial\\_rev10.htm](http://www.ucmh.sld.cu/rhab/editorial_rev10.htm)].

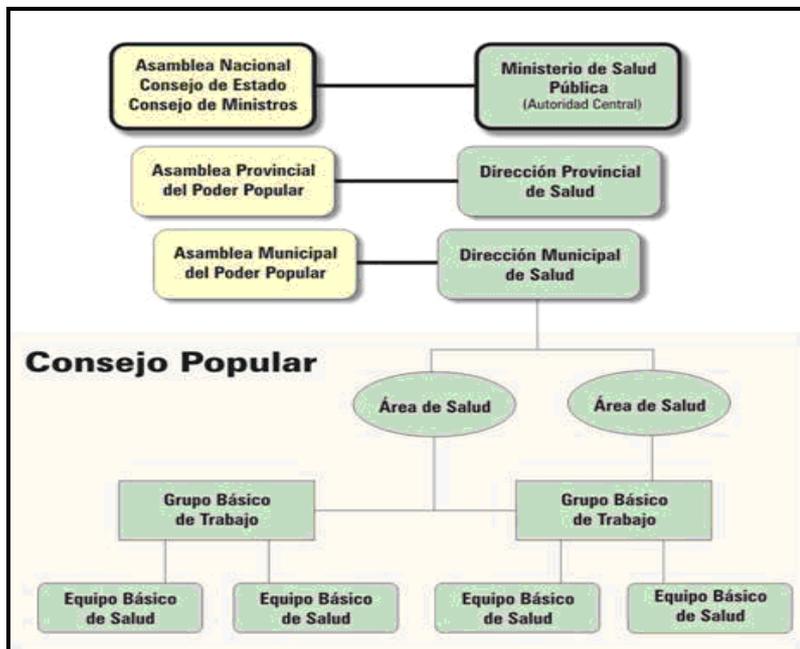
**Ramírez Márquez, Abelardo y Castell-Florit Serrate, Pastor y Mesa, Guillermo.** *El Sistema Nacional de Salud de Cuba.* ENSAP, 2003.

**PROGRAMACIÓN EN CASTELLANO.** *Java en Castellanos.* [En Línea]. [Consultado el: 13 de febrero de 2007] [Disponible en: <http://www.programacion.net/java/>].

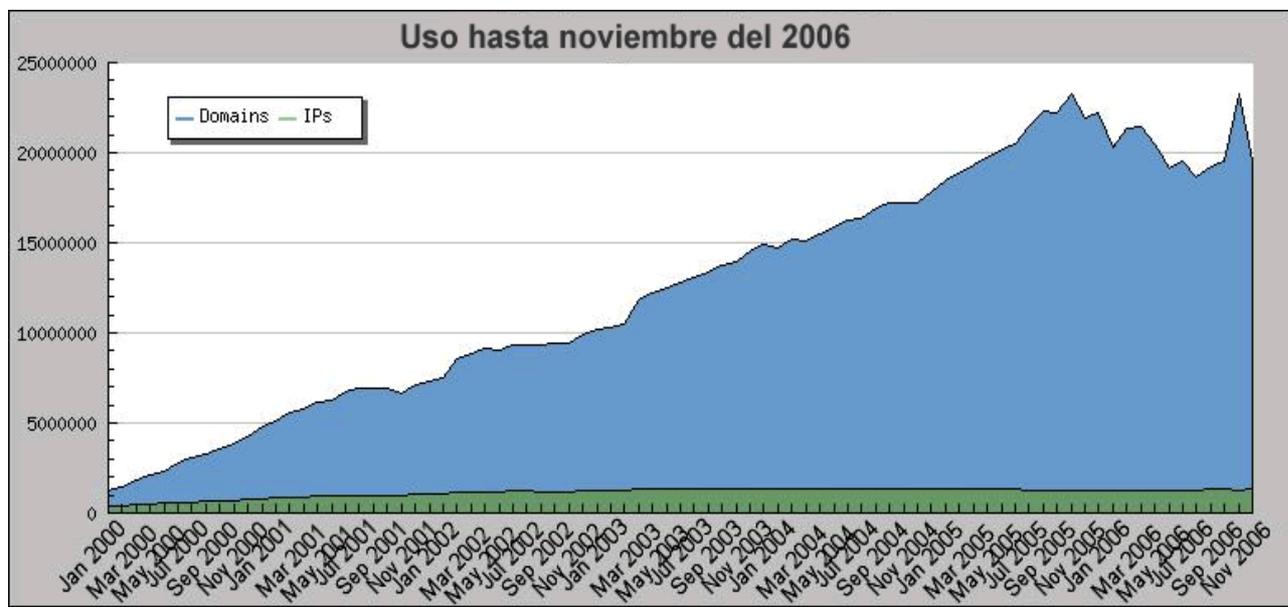
**Rey Jimenez, Alexandra.** CONSULTORÍA DE SEGURIDAD. *Que es UML? : Diagrama de Componentes*, [En Línea]. [Consultado el: 01 de junio de 2007]. [Disponible en: <http://www.creangel.com/uml/componente.php>].

**Rey Jiménez, Alexandra.** CONSULTORÍA DE SEGURIDAD. *Que es UML? : Diagrama de Despliegue*. [En Línea]. [Consultado el: 01 de junio de 2007]. [Disponible en: <http://www.creangel.com/uml/despliegue.php>].

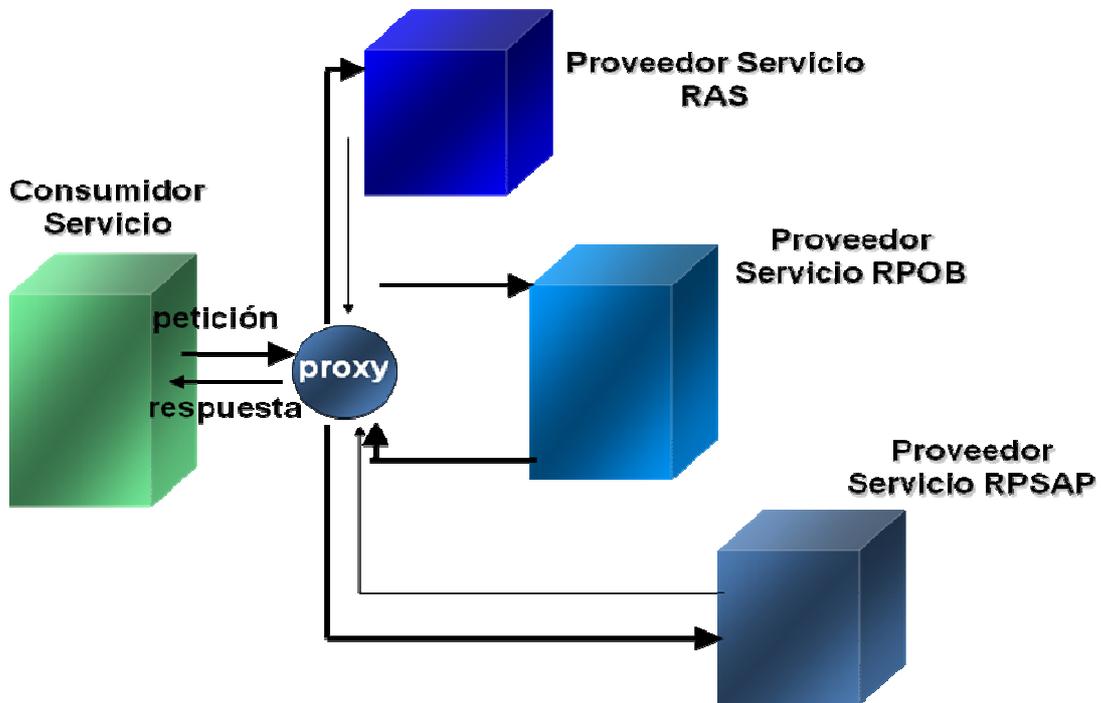
ANEXOS



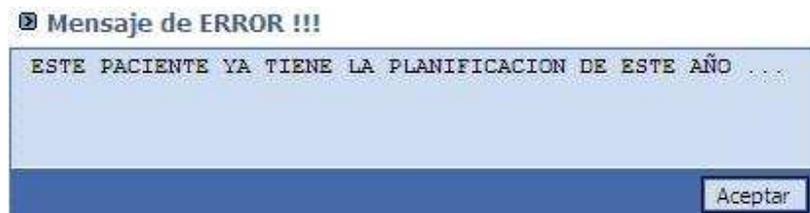
Anexo 1. Esquema de la Estructura Organizativa del MINSAP.



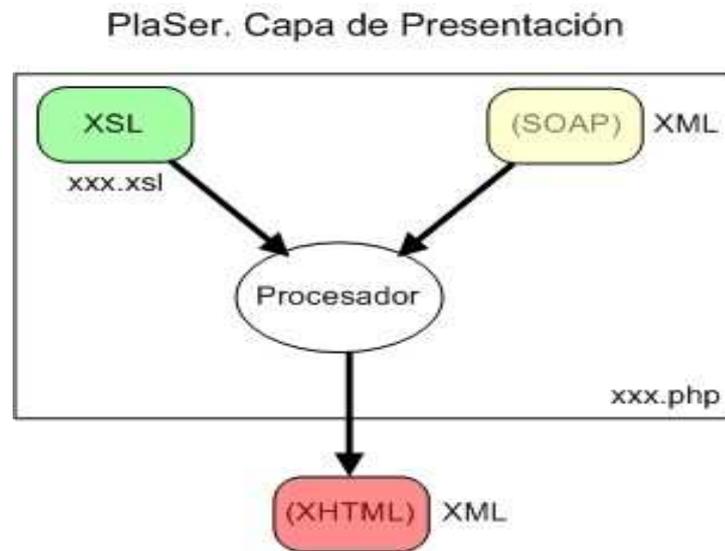
Anexo 2. Crecimiento de la utilización de PHP hasta Noviembre 2006.



Anexo 3. Patrón Proxy.



Anexo 4. Mensaje de error.



**Anexo 5.** Capa de Presentación.

## GLOSARIO DE TÉRMINOS

**Acciones de salud:** Actividades que realiza el EBS: Consultas, Terrenos, vacunación y pruebas citológicas.

**Afección:** Sinónimo de problema de salud.

**Citología Orgánica:** Proceder realizado a las mujeres entre 25 y 60 años de edad, consistente en la toma de una muestra por raspado del cuello uterino y su estudio en el laboratorio. Permite detectar afecciones tales como neoplasias de cuello, infecciones, etc.

**Condiciones materiales de vida:** Se refiere a las condiciones de la vivienda determinadas por: las Condiciones estructurales de la vivienda, el Índice de Hacinamiento, el Equipamiento doméstico básico y la satisfacción de la familia con los ingresos.

**Consulta Médica:** Relación interpersonal médico paciente que puede ocurrir en el Consultorio o en el hogar del paciente, en la que el médico realiza acciones de prevención, diagnóstico, tratamiento y Rehabilitación para la solución de los problemas identificados.

**Discapacidad:** Es toda restricción o ausencia de la capacidad de realizar una actividad en la forma o dentro del margen que se considera normal para un ser humano.

**Dispensarización:** Es el proceso organizado, continuo y dinámico de evaluación e intervención planificada e integral, con un enfoque clínico, epidemiológico y social, del estado de salud de los individuos y familias. Es un proceso coordinado y liderado por el EBS.

**Dispensarizado:** Individuo que:

- Está registrado en el Consultorio del Médico de la Familia.
- Está evaluado integralmente y clasificado según el Grupo dispensarial al que corresponda.
- Está recibiendo la intervención continua que su estado de salud necesita.

- Tiene actualizada la intervención periódica de su estado de salud según la frecuencia establecida para el Grupo dispensarial al cual pertenece.

**Enfermedades crónicas no transmisibles:** Enfermedades que una vez que surgen evolucionan durante toda la vida del paciente y que cursan con períodos de compensación y descompensación. No se transmiten. Ej. Hipertensión arterial, Asma bronquial, Diabetes Mellitus, etc.

**Enfermedades transmisibles:** Enfermedades originadas por agentes infecciosos (virales, bacterianos o parasitarios) que se transmiten entre los seres humanos o a partir de animales. Ej. Hepatitis Viral, Oxiuriasis, Sífilis, etc.

**Entrevista médica:** Método de trabajo directo en el que sus integrantes, entrevistado y entrevistador, tienen elementos de interés común que se aportan mutuamente.

**Equipamiento doméstico básico:** Relacionado con la existencia o no de equipos electrodomésticos de tipo básico, tales como Refrigerador, Televisor o radio y plancha.

**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.

**Evaluación de la salud familiar:** El estado de salud de la familia puede evaluarse como sigue:

- Familias sin Problemas de salud: cuando la Salud de sus integrantes y/o las Condiciones materiales de vida sean favorables y de no serlo, no influyan en la dinámica de las relaciones internas y el cumplimiento de las Funciones básicas de la misma, conservando la familia su equilibrio y capacidad plena para enfrentar y solucionar estas situaciones.

- Familias con Problemas de salud: cuando existan dificultades con las Condiciones materiales de vida y/o la salud de sus integrantes con una repercusión negativa en la dinámica de las relaciones internas y el cumplimiento de las Funciones básicas de la misma, no conservando la familia su equilibrio y capacidad para enfrentar y solucionar estas situaciones.

**Factores de riesgos:** Condiciones que aumentan la probabilidad de padecer una enfermedad o situación determinadas.

**Familia disfuncional:** Cuando el diagnóstico de la aplicación del FF-SIL, dé como resultado que es una Familia disfuncional y la familia no cumpla con las Funciones básicas.

**Familia funcional:** Cuando el diagnóstico de la aplicación del FF-SIL, dé como resultado que es una Familia funcional y la familia cumpla con las Funciones básicas.

**FF-SIL:** Test que se realiza a un miembro de la familia con el objetivo de determinar su percepción del Funcionamiento familiar.

**Funciones básicas:** Las funciones básicas se clasifican en:

- Función económica: relacionada con la satisfacción o no de las necesidades básicas materiales.
- Función educativa: Relacionada con el comportamiento social de los miembros de la familia.
- Función afectiva: Relacionada con la expresión de los miembros de la familia de signos que evidencien o no satisfacción afectiva con su familia.
- Función reproductiva: Relacionada con la planificación o no de las parejas de su descendencia y la existencia o no de ajuste sexual entre ellas.

**Grupo Básico de Trabajo:** Equipo de trabajo multidisciplinario integrado por un grupo de EBS (entre 15 y 20), por especialistas de Medicina Interna, Gineco-obstetricia, Pediatría, licenciado en psicología y de MGI, todos en función de interconsulta y de profesores, por una enfermera supervisora, un técnico de higiene y epidemiología y un técnico en Trabajo Social. El grupo cumple funciones asistenciales, docentes y gerenciales dirigidas a incrementar la calidad de la atención a la salud de la población.

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

**Grupos de edades:** Rango de edades en que se distribuye la población.

**Historia Clínica Individual:** Documento que utiliza el EBS y otros especialistas para recoger información general del individuo, antecedentes patológico personales, antecedentes patológicos familiares, datos del interrogatorio y examen físico, además de anotar todas las acciones realizadas en relación con el problema de salud que afecta al individuo. Este consta de varias partes para su confección que pueden variar en dependencia de la especialidad.

**Historia de Salud Familiar:** 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.

**Incidencia:** Casos nuevos de una afección.

**Índice de Hacinamiento:** Densidad de personas que duermen en la vivienda por número de locales que esta tiene.

**Ingresos en el hogar:** Es la atención médica integral, domiciliaria y diaria que brinda el EBS a aquellos pacientes en los que las condiciones o evolución de su afección no requieren, necesariamente, de internamiento en el hospital, pero sí de Encamamiento, aislamiento o reposo, y que se realiza siempre que se cumplan los requisitos necesarios para ello.

**Intervención educativa:** Cuando según el criterio del EBS la familia sea tributaria de recibir información con el objetivo de producir conocimiento sobre determinados temas de salud y promover la reflexión para estimular la adopción de estilos de vida saludables y patrones de relación funcionales y adaptativos a los cambios, incluyendo los de salud - enfermedad.

**Intervención terapéutica:** Cuando según el criterio del EBS la familia sea tributaria de recibir este tipo de intervención (tratamiento) por ser una Familia disfuncional.

**Migración Poblacional:** Movimiento de personas de un Consultorio Médico de la Familia hacia el Área de Atención de otro Consultorio.

**Nivel educacional:** Describe el nivel de escolaridad que corresponde a cada integrante de la familia, según la siguiente clasificación.

**Perl:** Es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo web, programación en red, desarrollo de GUI y más.

**Prevalencia:** Casos existentes de una afección.

**Problemas de salud:** Generalización de riesgo, discapacidad y enfermedades.

**Rehabilitación:** Rama de la medicina encargada de ayudar al paciente discapacitado a recuperar o mejorar las funciones perdidas para su reincorporación como miembro útil a la sociedad.

**Riesgo de disfuncionalidad:** Incumple con una de las Funciones básicas (Función reproductiva), pero no repercute en la dinámica de las relaciones internas de la familia.

**RPC:** (*Remote Procedure Call*, Llamada a Procedimiento Remoto) es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos. El protocolo es un gran avance sobre los sockets usados hasta el momento. De esta manera el programador no tenía que estar pendiente de las comunicaciones, estando éstas encapsuladas dentro de las RPC.

**Satisfacción de necesidades básicas:** Información relacionada con la satisfacción o no de las necesidades de alimentación, recreación, instrucción y de medios para garantizar la higiene personal y ambiental. Está determinado por las entradas económicas del núcleo familiar y su utilización en la satisfacción de las necesidades antes descritas.

**SOAP:** (*Simple Object Access Protocol*) es un protocolo estándar creado por Microsoft, IBM y otros, está actualmente bajo el auspicio de la W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

**Terrenos:** Atención que realiza el EBS a los pacientes en el domicilio y/o instituciones, con la finalidad de proporcionar asistencia médica, orientaciones preventivas, búsquedas de Factores de riesgos y seguimiento a las enfermedades crónicas.

**Validismo:** Que se vale por sí mismo. Dícese de la capacidad del individuo para realizar las actividades básicas (vestirse, peinarse, deambular) que le permiten no depender de otras personas.

**XHTML:** (*eXtensible Hypertext Markup Language*) Lenguaje extensible de marcado de hipertexto para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.