

613.0115  
DUR  
C  
TD 0073-05-01

TD.0073-05-01

**INSTITUTO SUPERIOR POLITÉCNICO "JOSÉ ANTONIO ECHEVERRÍA"  
FACULTAD DE INGENIERÍA INDUSTRIAL  
INGENIERÍA INFORMÁTICA**



# Centro de Control para el Sistema Integral de Salud

**Trabajo para optar por el título de Ingeniero Informático**

## **Autores**

Catherine Durand Llamos

Luis Mariano Reyna Soler

## **Tutor**

Ing. Alfredo Sánchez Rodríguez

**Ciudad de La Habana  
Julio de 2005**

### RESUMEN

El Sistema Nacional de Salud de Cuba, por su misión, cobertura y características, así como por su enfoque estratégico y programático, requiere de un constante flujo de información como vía para disponer en todo momento y a todo nivel de los conocimientos necesarios para dirigir cada una de las actividades que realiza.

Durante la etapa de análisis de los sistemas de información para el Sistema Integral de Salud vinculados al proyecto de informatización de la Atención Primaria de Salud y teniendo en cuenta su realización sobre la base de sistemas distribuidos, se hizo necesario el estudio y la elaboración de un componente denominado Centro de Control, encargado de configurar la distribución de los componentes y la gestión de los usuarios para facilitar la toma de decisiones referente a cómo enrutar las peticiones entre componentes del negocio para mantener la disponibilidad e integridad de la información distribuida que se maneja y proteger el acceso a estos en función de los derechos de cada usuario.

Esta investigación pretende, con la adecuación de patrones de arquitectura y organizando el despliegue de los componentes del sistema, lograr el objetivo anteriormente expuesto.

Para su modelación, se tomaron como indicadores los documentos e informaciones obtenidos en los procesos de análisis y diseño de los diversos componentes del sistema. Se realizó un trabajo integrado con especialistas del MINSAP y SOFTEL, lo cual permitió la formalización clara de su contenido y la obtención de un prototipo.

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA.....</b>	<b>7</b>
1.1 Introducción .....	7
1.2 Sistema Nacional de Salud.....	7
1.2.1 Informatización del Sistema Nacional de Salud .....	8
1.2.2 Informatización de la Atención Primaria de Salud .....	10
1.2.3 Registro Informatizado de Salud (RIS).....	11
1.2.4 Solución integral propuesta para la Informatización del SNS .....	11
1.3 Conceptos fundamentales .....	13
1.4 Objeto de estudio .....	13
1.4.1 Descripción General.....	13
1.4.2 Descripción del proceso actual. ....	16
1.4.3 Situación Problemática .....	17
1.5 Sistemas automatizados existentes vinculados al campo de acción. ....	18
1.5.1 Registro Informatizado de la Salud (RIS).....	18
1.6 Análisis comparativo de otras soluciones existentes con la propuesta .....	19
1.7 Objetivos generales y específicos .....	19
1.8 Conclusiones .....	20
<b>CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES.....</b>	<b>21</b>
2.1 Internet. Funcionamiento .....	21
2.1.1 El estandarte de Internet: World Wide Web .....	21
2.1.2 Aplicaciones Web .....	22
2.2 Tecnología XML/Webservices (SOA).....	23
2.3 DNS (Domain Name Service).....	24
2.4 Plataforma de Servicio (PlaSer).....	24
2.5 Entornos Distribuidos. Modelo Cliente Servidor.....	25
2.5.1 Modelo Cliente Servidor de dos Capas (Two Tier) .....	26
2.5.2 Modelo Cliente Servidor de tres Capas (Three Tier).....	26
2.6 Arquitectura Basada en Componentes (CBA) .....	26
2.7 Servidor Web Apache .....	27
2.8 Middleware .....	28

2.9 Patrón de diseño. Patrón Modelo Vista Controlador (MVC).....	28
2.9.1 ¿Por qué patrón MVC? .....	29
2.10 Patrón de seguridad <i>Single Sign On</i> (SSO).....	29
2.11 Lenguajes que implementan Servicios Web .....	30
2.11.1 Java Script.....	30
2.11.2 PERL.....	31
2.11.3 ASP .....	31
2.11.4 PHP .....	31
2.11.5 XSLT.....	33
2.12 Sistemas de Gestión de Bases de Datos (SGBD).....	34
2.12.1 SQL Server.....	34
2.12.2 MySql.....	35
2.12.3 PostgreSql .....	36
2.13 Desarrollo basado en RUP bajo la herramienta Rational Rose.....	37
2.13.1 UML (Unified Modeling Lenguaje) .....	38
2.13.2 Rational Rose .....	38
2.14 Herramientas a utilizar .....	40
2.15 Conclusiones .....	40
<b>CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN .....</b>	<b>41</b>
3.1 Introducción .....	41
3.2 Modelo del Dominio .....	41
3.3 Requerimientos Funcionales.....	42
3.4 Requerimientos No Funcionales .....	44
3.4.1 Apariencia o Interfaz Externa .....	45
3.4.2 Rendimiento.....	45
3.4.3 Usabilidad .....	45
3.4.4 Portabilidad .....	45
3.4.5 Autonomía.....	45
3.4.6 Seguridad .....	45
3.4.7 Escalabilidad .....	46
3.4.8 Software .....	46
3.4.9 Hardware.....	46
3.5 Descripción del Sistema propuesto .....	47

3.5.1 Concepción general del sistema.....	47
3.6 Modelo de casos de uso del sistema.....	48
3.6.1 Diagrama de Casos de Uso del Sistema.....	49
3.7 Casos de Uso.....	49
3.8 Conclusiones.....	55
<b>CAPITULO 4 CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN .....</b>	<b>56</b>
4.1 Introducción.....	56
4.2 Diagrama de Clases.....	56
4.3 Diseño de la base de Datos .....	65
4.3.1 Diagrama de clases persistentes.....	65
4.3.2 Modelo de Datos .....	66
4.4 Principios de Diseño .....	67
4.4.1 Estándares en la interfaz de la aplicación .....	69
4.4.2 Formatos de reportes.....	72
4.4.3 Concepción general de la ayuda.....	73
4.4.4 Tratamiento de excepciones.....	74
4.5 Estándares de codificación.....	75
4.6 Modelo de despliegue .....	76
4.7 Conclusiones.....	77
<b>CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD .....</b>	<b>78</b>
5.1 Introducción .....	78
5.2 Planificación.....	78
5.2.1 Características del proyecto .....	78
5.3 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo .....	81
5.4 Beneficios tangibles e intangibles.....	84
5.5 Análisis de costos y beneficios .....	84
5.6 Conclusiones.....	85
<b>CONCLUSIONES .....</b>	<b>86</b>
<b>RECOMENDACIONES .....</b>	<b>87</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>88</b>
<b>BIBLIOGRAFÍA .....</b>	<b>90</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>91</b>
<b>ANEXOS .....</b>	<b>97</b>

## ÍNDICE DE TABLAS

Tabla 3.1 Actores del Sistema.....	48
Tabla 4.1 Descripciones de las clases generales del Diseño.....	57
Tabla 5.1 Entradas Externas.....	79
Tabla 5.2 Salidas Externas.....	79
Tabla 5.3 Consultas Externas .....	80
Tabla 5.4 Archivos Lógicos Internos.....	80
Tabla 5.5 Interfaces Externas.....	80
Tabla 5.6 Puntos de Función sin ajustar.....	80
Tabla 5.7 Instrucciones fuentes.....	81
Tabla 5.8 Definición de los Multiplicadores de Esfuerzo (MEj).....	82
Tabla 5.9 Definición de los valores de los Factores de Escala (SFi).....	82
Tabla 5.10 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.....	83

**ÍNDICE DE FIGURAS**

Figura 3.1 Diagrama del modelo del dominio.....	42
Figura 3. 2 Diagrama de Casos de Uso del Sistema.....	49
Figura 4.1 Diagrama de clases: Caso de Uso Autenticarse.....	58
Figura 4.2 Diagrama de clases: Caso de Uso Cambiar Contraseña.....	58
Figura 4.3 Diagrama de clases: Caso de Uso Configurar Nodo.....	59
Figura 4.4 Diagrama de clases: Caso de Uso Configurar Tipo de Nodo.....	60
Figura 4.5 Diagrama de clases: Caso de Uso Configurar Componente.....	61
Figura 4.6 Diagrama de clases: Caso de Uso Configurar Método.....	62
Figura 4.7 Diagrama de clases: Caso de Uso Configurar Nivel.....	62
Figura 4.8 Diagrama de clases: Caso de Uso Gestionar Usuario.....	63
Figura 4.9 Diagrama de clases: Caso de Uso Emitir Reportes.....	64
Figura 4.10 Diagrama de Clases Persistentes.....	65
Figura 4.11 Modelo de Datos.....	66
Figura 4.12 Diagrama de Despliegue.....	77

## INTRODUCCIÓN

El Sistema Nacional de Salud (SNS) cubano viene realizando importantes reformas como parte fundamental de las transformaciones del período revolucionario y no como una imposición de los tiempos actuales de unipolaridad, globalización, de crisis económicas, de demandas de equidad social, sino como una necesidad del propio espíritu de la Revolución, como reflejo del respeto más absoluto de uno de los derechos humanos más importantes de todo ciudadano.

El Ministerio de Salud Pública, rector del Sistema Nacional de Salud ha trazado una estrategia para dar respuesta a los problemas, emergentes y reemergentes. Se persigue incrementar la eficiencia y calidad en los servicios, garantizar la sostenibilidad del sistema, especialmente en términos financieros y, aunque se ha alcanzado un alto nivel de igualdad, seguir estudiando y trabajando para eliminar pequeñas desigualdades reducibles en la situación de salud y la utilización de los servicios entre regiones y grupos de población.

Valerse de las técnicas de informática como agilizadoras de la transmisión y análisis de datos es de gran importancia en el campo de la Salud. Sin ello, hoy día no es posible llevar a cabo una práctica eficaz, no solamente en la gestión hospitalaria o extrahospitalaria sino también en la toma de decisiones clínicas.

El uso de la Informática en la Medicina es una de las aplicaciones más comunes e importantes desde hace varias décadas, y ha permitido al sector de la salud, no sólo contar con métodos novedosos, sencillos y eficaces de gestión administrativa en consultas, hospitales y centros de investigación biomédica, sino también disponer de complejos software que reducen la posibilidad de error en el diagnóstico de las enfermedades, y que aceleran su formulación. A su vez, ofrece una gran ayuda en el campo de la investigación.

Pero la aplicación de la computación en el campo de la Sanidad no se limita a eso; se cuenta para nuestros policlínicos con una propuesta de fortalecimiento de la Informática, cuyos propósitos son fortalecer la conexión entre las instituciones territoriales, lo que favorecería la comunicación y la adopción de decisiones a nivel local y niveles superiores, y fortalecer el Subsistema del Médico y Enfermera de la Familia, entre otros. Debemos recordar que la



tendencia en el mundo de la información apunta a que esta y el conocimiento determinan la base de la mayoría de los procesos.

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

A raíz de solucionar diversos problemas que mejoraran la eficiencia en el funcionamiento del sector de la Salud en Cuba, se presenta el proyecto para la Informatización de la Atención Primaria de Salud, concebido en el marco del proceso de Informatización de la Sociedad Cubana, ubicando su misión, en la definición conceptual de la estrategia a alcanzar, objetivos a cumplimentar, visión a mediano plazo, estructura organizativa, infraestructura de la información, así como en el presupuesto necesario para la ejecución del proyecto.

Cuba enfrenta el reto de informatizar la sociedad con vistas a integrarse plenamente a la infraestructura global de la información y hacer uso óptimo de las nuevas tecnologías, lo que permitirá lograr incrementos sustanciales en la productividad; mejoramiento de la calidad y la eficiencia en toda actividad, tanto industrial como de los servicios.

El rápido desarrollo que han experimentado las nuevas tecnologías de la información y las comunicaciones (NTIC) y la convergencia tecnológica de la microelectrónica, la informática y las comunicaciones, han generado nuevas plataformas e infraestructuras, cuya máxima expresión es Internet. Ellas modifican los criterios de espacio y tiempo y con ello, globalizan y agilizan los procesos en diferentes esferas de la sociedad, en particular, los relacionados con el intercambio de la información y el conocimiento.

La apropiación de estas tecnologías es parte de una nueva era: la del conocimiento; donde la mayoría de los trabajadores se consideran "trabajadores del conocimiento" y en la que el flujo de la información y el conocimiento presenta un impacto mayor en los resultados de las organizaciones que el movimiento de las mercancías o los servicios que prestan.

El trabajo de SOFTEL, Empresa del Ministerio de la Informática y las Comunicaciones (MIC) utiliza una estrategia nunca antes concebida en un proceso de desarrollo de software en el país, 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 de Salud (APS) en calidad de expertos funcionales en estrecho vínculo con los especialistas de Informática, a través de una metodología con la que se logró describir en Lenguaje Unificado de Modelado (UML) los procesos que se propusieron para informatizar. Esto representa en el mundo actual del desarrollo de software el 75% del éxito de un proyecto. 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 pasos a partir de ahora en la Informatización del Sector de Salud. [1]

Está demostrado que existe una necesidad apreciable de utilizar metodologías adecuadas para diseñar soluciones informáticas en el sector de la Salud, con mayor productividad y mejor calidad, aprovechando los avances y métodos actuales en la Ingeniería de Software.

El desarrollo actual de los sistemas de gestión de la información y el conocimiento para la dirección en salud, requiere, tanto de los sistemas de información y de las NTIC como de un profesional, que integre el equipo del primer nivel de dirección en cada entidad capaz de administrarlas. El conocimiento no se gerencia, sino las condiciones en que éste se comparte y debe estar accesible y disponible en forma rápida y segura; sólo así un directivo obtendrá la eficiencia y eficacia necesaria para su gestión.

Cada módulo que pertenece al Sistema Integral de Salud y que están ubicados de forma distribuida en los diferentes sistemas que se proponen, presenta una serie de componentes que les permiten interactuar con otros módulos y de los cuales depende su funcionamiento. En este sentido surge el siguiente **problema**:

¿Cómo mantener la disponibilidad de la información distribuida que se maneja en los componentes del sistema y proteger el acceso a estos en función de los derechos de cada usuario?

**Objeto de estudio:** Proceso de gestión de la información de los componentes del Sistema Integral de Salud y el estudio y aplicación de herramientas y procesos de desarrollo para el diseño de sistemas informáticos.

**Campo de acción:** Proceso que garantice la información necesaria para lograr la seguridad, integridad y disponibilidad de la información del Sistema Integral de Salud y el empleo del proceso de desarrollo RUP (Proceso Unificado de Rational), y las herramientas informáticas Corel Draw para la elaboración de prototipos y Rational Rose para el análisis y diseño del proyecto.

Para resolver el problema se propone el siguiente **objetivo:** Modelar un software con calidad aplicando patrones de arquitectura que permita la configuración de los componentes y la gestión de los usuarios para lograr la disponibilidad e integridad de la información distribuida que se maneja en los componentes del sistema, y proteger el acceso a éstos en función de los derechos de cada usuario.

### **Objetivos específicos:**

1. Realizar un estudio del comportamiento de los diferentes módulos que forman parte del Sistema Integral de Salud para resolver la configuración de la distribución de los componentes dentro de un sistema distribuido que facilitará el enrutamiento de las peticiones hechas entre los mismos.
2. Realizar un estudio de los posibles usuarios que utilizarán el sistema y los permisos y niveles que pueden tener cada uno para acceder a los diferentes módulos.
3. Estudio del patrón **Single Sign On** para garantizar la seguridad basado en su funcionamiento y proteger el acceso a los componentes en función de los derechos de cada usuario.
4. Realizar un estudio de las herramientas informáticas a utilizar en la futura implementación de la solución propuesta.
5. Analizar y diseñar el componente Centro de Control.

Para guiar la investigación se proponen las siguientes **preguntas científicas:**

1. ¿Cómo tener el control de los mensajes que se envían entre los componentes del sistema pudiendo tomar decisiones para un mayor rendimiento del sistema?

2. ¿Cómo aplicar patrones de arquitectura para el sistema distribuido de Atención Primaria de Salud?
3. ¿Cómo desarrollar el sistema para su ejecución en el entorno Web?
4. ¿Cómo obtener la modelación de la aplicación?
5. ¿Cómo tener una fuente de información del análisis y diseño del sistema?

Para dar cumplimiento a las preguntas anteriormente planteadas se definen las siguientes **tareas**:

1. Elaboración de la fundamentación teórica del componente para el proceso de Informatización del Sistema de Atención Primaria de Salud (APS).
2. Estudio del funcionamiento del DNS (Domain Name System).
3. Estudio de los patrones de arquitectura.
4. Estudio de los lenguajes de programación WEB.
5. Estudio de los estándares de Internet.
6. Desarrollar el modelado de la aplicación utilizando el Proceso Unificado de Desarrollo RUP (Rational Unified Process) y el Lenguaje Unificado de Modelación (UML).
7. Documentar todo el proceso de desarrollo.

Para llevar a cabo estas tareas se emplearán métodos empíricos y teóricos de la investigación científica. Dentro de los métodos empíricos tenemos la entrevista y la encuesta. La entrevista posibilitará obtener la información referente a cómo se espera que funcione la aplicación Web, y la encuesta, facilitará información sobre la relación existente entre los diferentes componentes, así como el conocimiento de los diferentes métodos que surgirán en el curso de la investigación. La entrevista y la encuesta serán realizadas a los jefes de proyectos de cada módulo. En los métodos teóricos se usará el análisis y la síntesis cuando a partir de condiciones específicas se llegará a ideas generales. La inducción y la deducción se utilizarán en diferentes etapas de la investigación, especialmente cuando de las ideas generales se deduzcan las particulares.

Para modelar el sistema, se siguió el Proceso Unificado de Software, que utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software, garantizando la elaboración de las fases de análisis y diseño del producto, satisfaciendo así las necesidades y requerimientos del cliente.

Al concluir esta investigación es importante determinar los aportes que brindará. En tal sentido es importante señalar el análisis y diseño de un sistema informático (Centro de Control) que permitirá configurar los componentes y gestionar los usuarios para facilitar la toma de decisiones referente a cómo enrutar las peticiones hechas a los componentes de negocio para mantener la disponibilidad de la información distribuida que se manejará en los componentes del sistema y cómo proteger el acceso a estos en función de los derechos de cada usuario. Ayudará a comprender la Arquitectura Basada en Servicios y el Desarrollo Basado en Componentes.

El Proyecto de la Atención Primaria de Salud (APS) hace uso de un **framework** que incluye Acceso a Base de Datos; comunicación SOAP entre componentes y un Sistema de Seguridad basado en el modelo de Autenticación.

Se logrará un adiestramiento en el lenguaje de programación PHP, para el desarrollo o implementación de la capa de negocio y el XSLT como lenguaje para la transformación de los documentos XML para la capa de presentación. El uso de XML Web Services como tecnología, garantizará la reusabilidad del código en el desarrollo de los sistemas y su empleo será muy útil en la integración de los módulos desarrollados.

El trabajo se estructura en cinco capítulos: Los dos primeros son introductorios comenzando por "Fundamentación del Tema", donde se muestran los principales conceptos manipulados en el transcurso de la investigación; luego "Tendencias y Tecnologías Actuales", donde se hace una breve referencia al estado del arte de las herramientas utilizadas en el mundo para dar solución a problemas similares. Los capítulos "Descripción de la Propuesta de Solución" y "Construcción de la Solución Propuesta" se orientan al análisis y diseño del software respectivamente; y finalmente aparece el capítulo "Estudio de Factibilidad", donde se analizan los balances de costo beneficio del software que se propone, además de Conclusiones, Recomendaciones, Referencias bibliográficas, Bibliografía y Anexos.

# **CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA**

## **1.1 Introducción**

El desarrollo de la Informática tiene un papel fundamental para el desarrollo de los diversos sectores en la sociedad. La informatización del Sistema Nacional de la Salud no está excepto de ello, por lo que las tecnologías han permitido una mayor eficiencia en el funcionamiento de este sector, uno de los logros fundamentales de la Revolución Cubana.

El objetivo fundamental de este capítulo es abordar distintos aspectos que se utilizan como soporte teórico del sistema diseñado. Se exponen a través de una descripción los conceptos asociados al problema y la estructura organizativa del Sistema Nacional de Salud (SNS). Se define además el objeto de estudio, la situación problemática y los objetivos generales y específicos de la presente investigación.

## **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 justa que caracteriza a nuestro proceso 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. [2]

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. [3]

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. [4]

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.

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), Programa de Control de Enfermedades Transmisibles, Programa de Enfermedades Crónicas no Transmisibles, y el Programa de Atención al Adulto Mayor, todos los que han sido monitorizados, controlados y evaluados de acuerdo a la metodología establecida. [5]

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 proporcionarle 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.

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. [6]

### **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

systemática de las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC) 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. [7]

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 podemos mencionar:

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 multicéntrica, nacional e internacional.



- Elevará el papel del Médico y 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.

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.

El Sistema de Salud Cubano, posee en el nivel de Atención Primaria una plataforma ideal para articular los avances de las nuevas 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. [8]

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.

En esta nueva etapa de fortalecimiento del Sistema Nacional de Salud, la Atención Primaria de Salud (APS) es el eje fundamental de estas transformaciones, teniendo como objetivo fundamental 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 informatización 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. 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.

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

El Registro Informatizado de Salud (RIS) está orientado a la gestión de información de interés nacional. El sitio en que se encuentra tiene alcance nacional, abarcando desde el nivel primario que es la Unidad Básica de la Salud hasta la Dirección Nacional del MINSAP. Desde cualquier computadora y desde cualquier sitio del país, se puede tener acceso a la información y se registra la trayectoria de los médicos, equipos médicos y no médicos, personal docente y demás información que el sistema pueda brindar y ser útil al interés del cliente.

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 Integral de Salud (SISalud)**, compuesto a su vez por el **Registro Informatizado de Salud (RIS)**, el **Sistema Informatizado**

**de Atención Primaria (SIAP) y el Sistema Informatizado de Gestión Hospitalaria (SIGH).**  
(Ver Anexo I).

**Registro Informatizado de Salud (RIS):** Está formado por los registros que son administrados o gestionados a nivel nacional o central y que integran el **Registro No Médico Informatizado de Salud (RNMIS)** 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, agrupándose éstos en el **Registro Médico Informatizado de Salud (RMIS).**

**Registro No Médico Informatizado de Salud (RNMIS):** En esta nueva etapa de análisis, diseño y desarrollo se incorporarán al RNMIS: el Registro de Áreas de Salud, Registro de Medios de Diagnóstico, Registro del Clasificador Internacional de Enfermedades y Problemas de Salud (CIE) 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 Dispensarización. De igual forma se ubicarán en el RNMIS todos los registros que en la actualidad pertenecen al RIS, mencionados en el epígrafe anterior.

**Registro Médico Informatizado de Salud (RMIS):** Estará formado por todos los módulos o componentes que no son del dominio de Atención Primaria propiamente, pero procesan y generan información que se obtiene de este nivel comunitario y además lo retroalimenta. En esta primera etapa se desarrollan: Registro de Enfermedades de Declaración Obligatoria (EDO), Registro de Fallecidos y Registro de Partos y Nacimientos.

**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.

### **1.3 Conceptos fundamentales**

**Componentes:** Es un subsistema que es el núcleo de la aplicación donde se encuentran los métodos propios de cada sistema.

**Nodo:** Es una ubicación física en la que se encuentran determinados componentes.

**Tipo de nodo:** Define los componentes que van a estar ubicados en nodos que tienen una distribución similar.

**Usuarios:** Personal que tiene acceso a los diferentes módulos que forman parte Sistema Integral de Salud

**Administradores:** Personal que accede al Centro de Control para configurar los distintos nodos, componentes, métodos y nivel y que además gestiona la información de todos los usuarios.

### **1.4 Objeto de estudio**

#### **1.4.1 Descripción General**

El Ministerio de Salud Pública (MINSAP) es el Organismo rector del Sistema Nacional de Salud, encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la Salud Pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica.

El SNS se estructura en tres niveles que se corresponden con la estructura político-administrativa del país. El nivel nacional está representado por el Ministerio de Salud Pública que es el órgano rector con funciones metodológicas, normativas y de coordinación y control, al cual se le subordinan directamente los centros universitarios, institutos de investigaciones, centros hospitalarios de asistencia médica altamente especializados, centros de distribución y comercializadoras de suministros y tecnologías médicas, así como otros centros y entidades nacionales destinados a actividades técnicas y de apoyo. (Ver Anexo II).

## CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA

Los otros dos niveles están representados por las direcciones provinciales y municipales de salud que agrupan a las instituciones de salud en su respectivo nivel y que, al igual que en el nivel central, se subordinan desde el punto de vista administrativo a las estructuras de Gobierno en los distintos niveles organizativos, representando sus intereses ante ellos y dando respuesta a las demandas y necesidades de la población.

Las áreas de salud y las unidades que prestan servicios en la atención primaria se subordinan a los Consejos Populares, quienes se encargan de resolver los problemas más específicos de la población de su radio de acción. Además la labor de los EBS se subordina a los Delegados de Circunscripción pertenecientes a los Consejos Populares.

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.

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, autoresponsabilidad y acción creadora.

## CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA

**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.

Podemos destacar entre los Principios Rectores del MINSAP 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.

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.[9]

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.

### **1.4.2 Descripción del proceso actual.**

El proceso informatizado de la salud tiene como objetivo el desarrollo de sistemas informáticos que generen un proceso fiable para la toma de decisiones, partiendo de su célula fundamental: Atención Primaria de Salud (APS).

APS se presenta en la actualidad como una agrupación de sistemas que se acoplan entre ellos para dar respuesta al negocio de Atención Primaria de Salud propuesto por el Sistema Nacional de Salud Cubano.

Los sistemas o módulos que se están modelando de APS heredan cada uno de ellos la arquitectura generada por el RIS, pero a diferencia de este APS no responde a una distribución centralizada de sus componentes, sino que estos se encuentran distribuidos e interactuando entre sí aprovechando la infraestructura de comunicaciones de la Red Telemática de Salud Cubana: INFOMED.

Dichos módulos se encuentran en la actualidad en fase de desarrollo o implementación, para lo cual ha tenido que hacerse reingeniería a los procesos que ya habían sido implementados en diferentes sistemas de información desarrollados bajo tecnologías mas viejas y sin visión integradora.

La Arquitectura contempla un componente local denominado Server al cual llegan todas las peticiones SOAP hechas a cualquier componente del sistema basado en el Patrón Modelo-Vista-Controlador (MVC). Es necesario contar con un módulo de configuración de componentes que permita facilite el enrutamiento de estas peticiones, en función de diseñar la estrategia de enrutamiento de mensajes teniendo en cuenta las características de cada módulo y para garantizar la disponibilidad mencionada anteriormente.

Después del análisis del despliegue de los componentes se llegó a la conclusión que existen tres formas en las que se comporta dicho enrutamiento:

- Enrutar el mensaje al componente remoto y en caso de fallo al componente local.
- Enrutar el mensaje al componente local y en caso de fallo al componente remoto.

- Enrutar el mensaje al componente local solamente.

Las peticiones antes de enrutarse deben haber sido chequeadas según la configuración de los derechos que también se establecen en un módulo de Configuración del Centro de Control, el cual utiliza el componente de seguridad SAAA que implementa el mecanismo de seguridad **Single Sig On** y que a su vez estará clonado en cada nodo de distribución de componentes en función de chequear localmente los derechos.

### **1.4.3 Situación Problemática**

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 Proyecto de Informatización de la Salud en Cuba encamina sus tareas a la elaboración de módulos que se encuentran distribuidos en los Centros de Salud, ubicando su desarrollo en la Atención Primaria de Salud (APS), piedra angular del Sistema Nacional de Salud de Cuba.

La misión del proyecto es implementar un Sistema Integral de Salud (ubicando su labor en la Atención Primaria de Salud), que responda a las estrategias trazadas por el MINSAP con el objetivo de elevar la calidad y eficiencia de los servicios que se brindan a nuestra población.

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

El Proyecto de Atención Primaria de Salud (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 de Salud, para dar respuesta a los procedimientos establecidos en el Sistema de Salud Cubano para este nivel de atención.



Cada módulo implementado consta de diferentes componentes. La actualización de los módulos depende del flujo de la información que es almacenada, por lo que es imprescindible la relación entre ellos.

En la fase de conceptualización de los diferentes módulos, se vio la necesidad de elaborar un componente denominado Centro de Control, encargado de configurar los componentes y gestionar los usuarios para facilitar la toma de decisiones referente a cómo enrutar las peticiones hechas a los componentes de negocio para mantener la disponibilidad de la información distribuida que se maneja; balancear la carga de procesamiento al organizar el despliegue de los componentes del sistema y proteger el acceso a estos en función de los derechos de cada usuario. Esto contribuye a la integridad, rapidez y confiabilidad del sistema a implementar.

Es por ello, que se requieren copias locales de los componentes del sistema que puedan dar respuesta a los posibles fallos en las invocaciones de los métodos de los componentes remotos, lo cual exige además la puesta en marcha de un mecanismo de replicación de la información que garantice la integridad de la misma.

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

#### **1.5.1 Registro Informatizado de la Salud (RIS).**

El Registro Informatizado de la Salud (RIS), es un sistema disponible en esta Red Telemática de Salud de Cuba: INFOMED.

Para compartir la información con otros sistemas, RIS fue desarrollado siguiendo una arquitectura multicapas, orientada a servicios y basada en componentes. La capa de negocio está disponible a otras aplicaciones utilizando las tecnologías de *xml Web Services*, específicamente SOAP; utiliza PlaSer (Plataforma de Servicios) para la programación y homogeneidad de los módulos.

En RIS se recogen algunos servicios tales como Registro de Ubicación, Registro de Unidades de Salud, Registro del Ciudadano, Registro de Personal de Salud y Registro de Equipos de Salud.

Se dispone además de una capa de presentación para el web que agrupa funcionalidades del negocio, dicha agrupación responde al nombre de módulo. Cada módulo posee su correspondiente componente en la capa de negocio aunque también interactúa con otros.

Para el desarrollo de RIS se obtuvo un framework denominado PlaSer (Plataforma de Servicio) que funciona bajo la arquitectura descrita anteriormente y pone a disposición de los desarrolladores clases implementadas para cada una de las capas presentes en la arquitectura.

### **1.6 Análisis comparativo de otras soluciones existentes con la propuesta**

Actualmente no existe ningún componente de arquitectura capaz de direccionar las peticiones según los patrones de direccionamiento configurados en función de lograr una alta disponibilidad y el balanceo de carga.

El proyecto de Informatización de la Atención Primaria de Salud, incorpora este componente denominado Centro de Control el cual contiene un módulo de configuración donde se gestionará la información relacionada con la distribución de cada uno de los componentes del sistema agrupados en nodos, en cada nodo puede encontrarse cualquier componente manejando información propia y sincronizado a un componente central que en muchos casos es un clon o copia del mismo componente local.

En el Centro de Control también residirá una nueva versión del Sistema de Seguridad SAAA para tratar los nuevos niveles de acceso y facilitar la administración de usuarios. La investigación está enmarcada en la modelación del sistema para su futuro desarrollo e implantación.

### **1.7 Objetivos generales y específicos**

El desarrollo de esta investigación tiene como principal objetivo aplicar patrones de arquitectura que permitan implementar un componente reusable por otros módulos. Entre los objetivos específicos se encuentran:

1. Analizar y diseñar el componente de arquitectura: Centro de Control.
2. Configurar la seguridad basado en el patrón *Single Sign On*.
3. Configurar el despliegue de los componentes.

4. Emitir reportes de configuración de los nodos y componentes del sistema distribuido, así como de los usuarios del sistema.

### **1.8 Conclusiones**

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio; y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que rodean al problema y sobre esta base se obtuvieron los objetivos generales y específicos para este trabajo. Este tiene gran importancia porque constituye la base para su futuro desarrollo.

### **CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES**

La aplicación directa de recursos informáticos está cambiando radicalmente muchos de los mecanismos y procedimientos tradicionales en las entidades que han sido totalmente afectadas por las nuevas tecnologías. Se desarrollan más los sistemas, dada la diversidad y potencialidad del equipamiento, en cuanto a capacidad y velocidad de procesamiento, capacidad de almacenamiento masivo y posibilidad de interconexión.

En este capítulo se tratan conceptos fundamentales asociados al sistema con el uso de la metodología de modelación y desarrollo a aplicar en las diferentes fases del desarrollo. Así como la fundamentación del lenguaje que será utilizado comparándolo con otros existentes para demostrar el por qué de su elección.

#### **2.1 Internet. Funcionamiento**

Internet, denominada la Red de Redes, fue creada a partir de un proyecto del departamento de defensa de los Estados Unidos llamado ARPANET (Advanced Research Project Network según su sigla en inglés) fue iniciado en 1969 y cuyo principal propósito era la investigación y desarrollo de protocolos de comunicación para redes de área amplia, para ligar redes de transmisión de información de diferentes tipos; capaces de resistir las condiciones de operación más difíciles y continuar funcionando aún con la pérdida de una parte de la red. [10]

Para que todos estas computadoras puedan coexistir y comunicarse efectivamente entre sí, debe existir un camino físico que las una (líneas telefónicas, conmutadas, redes digitales, enlaces satelitales, microondas, fibra óptica, cable coaxial, etc.), además deben ponerse de acuerdo en cuanto al lenguaje que utilizarán para comunicarse, es decir usar el mismo protocolo de comunicación (TCP/IP).

##### **2.1.1 El estandarte de Internet: World Wide Web**

La aplicación que más ha favorecido el conocimiento y despegue de Internet en todos los ámbitos ha sido el World Wide Web (WWW), o telaraña mundial, de tal forma que incluso en ocasiones se llega a confundir la propia Internet con el Web, como también se conoce. El World Wide Web fue un desarrollo de Tim Berners-Lee, del CERN de Ginebra, que buscaba

desarrollar un medio de publicación e intercambio de información entre los físicos de partículas.  
[11]

El principio general que guía el ámbito Web es el hipertexto y el hipermedia. En breve, un autor crea un documento Web que puede contener texto, gráficos estáticos y/o dinámicos, sonido, entre otros, y lo pone a disposición de todos los posibles usuarios. Estos documentos son accesibles a través del protocolo HTTP (HyperText Transfer Protocol). El lector/usuario del documento tiene, a partir de ese momento, la libertad de elegir que camino, que nuevo documento quiere acceder y consultar.

### **2.1.2 Aplicaciones Web**

Las empresas están aprovechando las indudables ventajas que ofrecen las aplicaciones Web (término genérico para el software que está relacionado con Internet).

Una aplicación Web es un conjunto de páginas dinámicas generadas al vuelo, según una secuencia de parámetros introducidos por el usuario. Es una aplicación informática completa, donde el marco en que se muestran los contenidos es su propio navegador de Internet.

Entre las ventajas que se pueden mencionar están:

- No requieren instalación, pues usan tecnología Web, lo cual nos permite el aprovechamiento de todas las características de Internet.
- Son fáciles de usar (no requieren conocimientos avanzados de computación).
- Alta disponibilidad, ya que puede realizar consultas en cualquier parte del mundo donde tenga acceso a Internet y a cualquier hora.
- Facilitar la comunicación entre gobierno, instituciones educativas, empresas, asociaciones y personas físicas, con el propósito de establecer una relación aún más estrecha entre ellos.
- Particularmente en el aspecto comercial, fomentar una mayor comunicación entre clientes y empresas estableciendo un modelo de operación del negocio más orientado al cliente.

### **2.2 Tecnología XML/WebServices (SOA)**

XML es el acrónimo de Extensible Markup Lenguaje, se ha convertido en un formato estándar en Internet y está diseñado para representar datos estructurados, no es un lenguaje de marcado como su nombre lo especifica; es un metalenguaje para definir otros lenguajes de marcados adecuados a un uso en específico, éste es la base de los servicios Web. XML, al que algunos consideran el Esperanto de los sistemas de información, se emplea principalmente para separar el contenido de la presentación de forma total, o sea, permite representar datos de forma homogénea en entornos heterogéneos, lo que facilita la interoperabilidad entre distintos sistemas.

Los Servicios Web usan SOAP (Simple Object Access Protocol) como protocolo de transporte estándar por su simplicidad, se puede identificar un mensaje SOAP como un documento XML conformado por una envoltura “envelope” obligatoria, un encabezamiento “header” opcional y un cuerpo “body”, también obligatorio [SOAP-W3C], y la forma de acceder a ellos es a través del WSDL (Web Services Description Languages). [12]

Estos servicios deben publicar una interfaz que funja como un contrato de servicio y donde se describan cada una de las funciones que provee además de las funciones que estos ofrecen, como realizar el intercambio de mensajes, especificar el contenido de una petición y el aspecto de la respuesta en una notación inequívoca. Además de describir el contenido de los mensajes, WSDL define dónde está disponible el servicio y qué protocolo de comunicaciones utilizar para hablar con el servicio. Esto significa que el archivo WSDL define todo lo necesario para escribir un programa que interactúe con un Servicio Web.

Cada vez más las empresas exigen aplicaciones más complejas, con menos tiempo y presupuesto que antes. Crear estas aplicaciones, requiere en muchos casos de funcionalidades ya antes implementadas como parte de otros sistema. SOA (Service Oriented Architecture) nace como una estrategia de integración, expone servicios con funcionalidad bien definida a la aplicación que la requiera. De esta manera, una aplicación final simplemente orquesta la ejecución de un conjunto de estos servicios, añade su lógica particular y le presenta una interfaz al usuario final.

Exponer procesos de negocio como servicios es la clave a la flexibilidad de la arquitectura. Esto permite que otras piezas de funcionalidad (incluso también implementadas como servicios) hagan uso de otros servicios de manera natural, sin importar su ubicación física. Así un sistema evoluciona con la adición de nuevos servicios y su mejoramiento. Donde cada servicio evoluciona de una manera independiente. La Arquitectura Orientada a Servicios (SOA) resultante, define los servicios de los cuales estará compuesto el sistema, sus interacciones, y con qué tecnologías serán implementados. La interfaz que utiliza cada servicio para exponer su funcionalidad son gobernadas por contratos, que definen claramente el conjunto de mensajes soportados, su contenido y las políticas aplicables. [13]

### **2.3 DNS (Domain Name Service)**

El módulo DNS (*Domain Name System*) es el responsable, fundamentalmente, de realizar la traducción entre un nombre jerárquico y una serie de "identificadores" asociados. Dichos identificadores pueden constituir cualquier información de interés práctico. [14]

Tradicionalmente estos sistemas se han utilizado para la obtención de la dirección IP de una máquina dado su nombre jerárquico, así como para identificar la máquina o máquinas responsables de un dominio de correo. No sería exagerado decir que el DNS constituye el verdadero núcleo aglutinador de una red de las características de "*Internet*", ya que es lo que proporciona al usuario una visión homogénea y organizada de las máquinas y servicios disponibles.

Todo servidor de nombres debe ser configurado con la lista de los servidores raíz bien conocidos. Estos servidores dicen qué dominios de primer nivel existen y cuales son sus servidores de nombres. Recursivamente, los servidores de esos dominios dicen que subdominios existen y cuáles son sus servidores.

### **2.4 Plataforma de Servicio (PlaSer).**

PlaSer es un sistema desarrollado utilizando el paradigma de XML – Web Services, en una arquitectura basada en componentes y orientada a servicios. En su concepción se han utilizado estándares actuales y normas abiertas. Se desarrolla por el grupo de trabajo dirigido a la Informatización de Sistemas de Salud.

Todo el código ha sido programado en PHP. Esta constituye una plataforma sobre la que se pueden desplegar aplicaciones XML – Web Services, con la ventaja de que el programador no tiene que preocuparse por implementar la seguridad del Sistema, ya que esta es una de las tareas que asume PlaSer. En esta versión PlaSer sólo soporta como llamada RCP el protocolo SOAP, pero en futuras versiones se incorporarán otros protocolos de transporte o incluso el acceso local a código a nivel de File System, de forma tal que para el programador sea totalmente transparente si la invocación del proceso es remoto, local, por SOAP, directamente a código, etc. (Ver Anexo III)

### **2.5 Entornos Distribuidos. Modelo Cliente Servidor**

La arquitectura Cliente-Servidor, es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema, en una arquitectura monolítica no hay distribución; los tres niveles tienen lugar en el mismo equipo, en el modelo cliente-servidor, en cambio, el trabajo se reparte entre dos ordenadores. [15]

Se puede decir que todas las aplicaciones tienen la misma arquitectura básica y se pueden subdividir en tres partes:

1. Interfaz de Usuario: La presentación al usuario, con las entradas de datos y las consultas.
2. Reglas de Negocio: El procesamiento de la información.
3. Accesos a Datos: El control del almacén de datos.

Ventajas del modelo Cliente Servidor:

- El servidor no necesita una elevada capacidad de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce considerablemente el tráfico de red. 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 fácilmente escalable.



### **2.5.1 Modelo Cliente Servidor de dos Capas (Two Tier)**

En este modelo el sistema se separa en dos partes fijas: cliente y servidor estableciendo un middleware que controla las comunicaciones entre ambas, la lógica de las aplicaciones debe estar en el cliente o en el servidor; la comunicación es transparente para el usuario.

Como limitante:

- El sistema no es escalable: No se adapta fácilmente al número de usuarios.
- El sistema no es manejable.
- Bajo rendimiento.

### **2.5.2 Modelo Cliente Servidor de tres Capas (Three Tier)**

Cada uno de los componentes de la aplicación en una arquitectura de tres capas se separa en una sola entidad. Esto te permite implementar componentes de una manera más flexible, es decir, la aplicación tiene que estar preparada para los posibles cambios que el cliente pueda pedir sin tener que reescribir totalmente la aplicación. Este tipo de arquitectura es la más compleja. (Ver Anexo IV)

En esta Arquitectura todas las peticiones de los clientes se controlan en la capa correspondiente a la lógica del negocio. Cuando el cliente necesita hacer una petición se la hace a la capa en la que se encuentra la lógica del negocio. Esto es bastante importante pues eso quiere decir que:

- El cliente no tiene que tener drivers ODBC ni la problemática consiguiente de instalación de los drivers, por tanto se reduce el costo de mantener las aplicaciones cliente.
- El Cliente y el Gestor de Reglas de negocio tienen que hablar el mismo lenguaje (COM, CORBA, SOAP).
- El Gestor de Reglas de Negocio y el Servidor de Datos tienen que hablar el mismo lenguaje (ODBC).

### **2.6 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 rehusados en múltiples aplicaciones. Cada componente debe describir de forma completa la interfaz 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. [16]

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.

### **2.7 Servidor Web Apache**

Es el servidor Web más utilizado en el mundo. Su coste gratuito, gran fiabilidad y extensibilidad le convierten en una herramienta potente y muy configurable.

Dentro de sus puntos fuertes se encuentran:

- Tiene interfaz con todos los sistemas de autenticación.
- Facilita la integración como "plug-ins" de los lenguajes de programación de páginas Web dinámicas más comunes.
- Tiene integración en estándar del protocolo de seguridad SSL.
- Provee interfaz a todas las bases de datos.
- Posee Virtual Host.

Apache fue hecho para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Está ligado a la plataforma (Linux, Windows, UNIX) sobre la cual los individuos o instituciones pueden construir sistemas confiables con fines experimentales o para resolver un problema específico de la organización. [17]

Apache es un software libre, porque sus desarrolladores defienden la teoría de que las transmisiones usando la red deben estar en las manos de todos, y que las compañías de software deben hacer el dinero ofertando servicios con valor añadido tales como módulos especializados, soportes, entre otros, y no siendo dueñas de un protocolo. Así, el proyecto de crear una implementación robusta con referencia absolutamente libre para quien lo quiera usar es un buen paso para evitar la propiedad sobre los protocolos.

### **¿Por qué tres capas y Apache?**

La aplicación necesitará ser flexible, portable y fiable, es decir, estará en servidores Windows o en la familia de los Unix; esto permitirá resolver complejos problemas inmersos en cambios constantes. Las arquitecturas basadas en tres capas permiten a los componentes de negocio correr en una LAN, WAN o Internet. Esto significa que cualquiera con un ordenador y conexión a la Red posee toda la funcionalidad que tendría si se encontrase delante de su sistema de escritorio.

### **2.8 Middleware**

Es una capa de software intermedio entre el cliente y el servidor. Es la capa de software que nos permite gestionar los mecanismos de comunicaciones.

Una aplicación común para el middleware es permitir que programas que se han desarrollado para acceder a una base de datos, puedan acceder otras bases de datos. La mensajería es un servicio común provisto por programas de middleware de tal manera que diferentes aplicaciones se puedan comunicar. [18]

Es un conjunto de interfaz y protocolos estándares de comunicación. Con interfaces estándares de programación, es fácil de implementar una misma aplicación en una variedad de tipos de servidores y de puestos de trabajo. Esta tiene un beneficio para los clientes, puesto que estos compran aplicaciones no servidores, los clientes sólo elegirán entre aquellos servidores donde se ejecuten las aplicaciones que ellos deseen.

### **2.9 Patrón de diseño. Patrón Modelo Vista Controlador (MVC)**

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. Cada patrón permite que algunos aspectos de la estructura del sistema puedan cambiar independientemente de otros aspectos. Facilitan la reusabilidad, extensibilidad y mantenimiento. [19]

Para el diseño de aplicaciones con sofisticados interfaz se utiliza el patrón de diseño Modelo-Vista-Controlador.

Elementos del patrón:

- Modelo: datos y reglas de negocio
- Vista: muestra la información del modelo al usuario
- Controlador: gestiona las entradas del usuario.

### **2.9.1 ¿Por qué patrón MVC?**

Aplicando el patrón Modelo-Vista-Controlador (MVC) se logra separar el modelo de negocios de la presentación usada para ser visualizados, de la lógica utilizada para procesar acciones externas. Esta separación permite múltiples vistas sobre el mismo modelo de datos, lo que simplifica la implementación (permitiendo separar roles de desarrollo) y mantener el sistema.

Este patrón es adecuado para situaciones donde:

- Es necesario presentar los mismos datos de diferentes formas o utilizando diferentes interfaz de usuario.
- Si cambia un dato, los datos se deben de actualizar en todas las representaciones.
- El interfaz de usuario debe de ser fácil de modificar.

### **2.10 Patrón de seguridad *Single Sign On* (SSO)**

*Single Sign On* (SSO) es la solución a algunos de los problemas con que se encuentran aquellas empresas que utilizan importantes aplicaciones empresariales en más de una plataforma. Los usuarios, para identificarse ante varios sistemas, se ven obligados a recordar numerosas contraseñas y a enfrentarse a distintas interfaz. Ésa es la razón de que elijan contraseñas sencillas o de que las escriban en notas adhesivas, poniendo potencialmente en peligro la seguridad del sistema. Los administradores, por su parte, se ven obligados a navegar entre interfaz de administración para dar alta o baja a los usuarios, o para modificar contraseñas en sistemas tan dispares como NT, UNIX o NetWare, de Novell. Se trata de un proceso engorroso que puede afectar a la seguridad de la red. Pues bien, SSO resuelve todos estos problemas. [20]

Las principales ventajas de la aproximación SSO son:

- Encapsulamiento de la infraestructura de seguridad subyacente en el servidor SSO. La implementación, el despliegue y el mantenimiento son más fáciles ya que ninguna de las

partes comunicantes en el sistema distribuido necesita implementar individualmente todas las características y mecanismos de seguridad.

- La interfaz SOAP del servidor SSO hace que la arquitectura SSO sea universal. SSO es él mismo un servicio web. Si el servidor SSO expone el WSDL de su interfaz, el API SSO se puede generar instantáneamente y puesto a disposición.
- El servidor SSO mejora la seguridad de todo el sistema ya que no se necesita pasar por todos los lados las credenciales de seguridad. El único punto que acepta credenciales de seguridad es el propio servidor SSO. La autenticación se puede realizar en un ámbito enorme (fuera del dominio de seguridad particular) mientras que la credenciales de seguridad permanecen dentro del dominio de seguridad particular.

### **2.11 Lenguajes que implementan Servicios Web**

Uno de los pilares de la cual difiere Internet de los restantes medio de comunicación es la personalización de la información del usuario mediante los diversos lenguajes de programación, estos lenguajes se clasifican en lenguajes del lado del cliente y lenguajes del lado del servidor.

Entre los lenguajes que trabajan del lado del servidor podemos citar algunos, que se destacan por ser los más sobresalientes como son PERL, ASP, PHP, Java, JSP, los módulos CGIs e ISAPIs etc., etc. Estos lenguajes desarrollan la lógica de negocio dentro del servidor, además se encargan de los accesos a los distintos Sistemas de Gestión de Bases de Datos. Dentro de los lenguajes que trabajan del lado del cliente se encuentran el JavaScript 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. [21]

#### **2.11.1 Java Script**

Es un lenguaje de programación interpretado, con capacidades elementales orientadas a objeto. El código Javascript es embebido directamente en el código HTML, haciendo fácil la creación de páginas Web con contenido dinámico. Está diseñado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador Web y es soportado por la gran mayoría de los navegadores.

### **2.11.2 PERL**

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el Web. PERL es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. PERL es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP.

### **2.11.3 ASP**

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (JavaScript de Microsoft). [22]

La mayor desventaja de este lenguaje es que solo se puede implementar sobre los Servidores Web de su desarrollador: Microsoft. Actualmente se ha presentado ya la segunda versión: el ASP.NET, que comprende algunas mejoras en cuanto a las posibilidades del lenguaje y rapidez con la que funciona. ASP.NET tiene algunas diferencias en cuanto a la sintaxis con el ASP, de modo que tienen formas distintas de utilizarse. Para implementarlo es necesario montar en el Servidor la Plataforma.NET.

### **2.11.4 PHP**

PHP (Personal Home Page) es el acrónimo de Hypertext Preprocessor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML. [23]

PHP, en el caso de estar montado sobre un servidor Linux o Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros

desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend-2, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales.

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, MSSQL, mSQL, Oracle, Informix, y ODBC, por ejemplo. Incluye funciones para el envío de correo electrónico, upload de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

PHP es la gran tendencia en el mundo de Internet. Últimamente se puede observar un ascenso imparable, ya que cada día son muchísimas más las páginas Web que lo utilizan para su funcionamiento.

Resumiendo, el PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos y contiene unas 40 extensiones estables sin contar las que se están experimentando, además de que:

- Simplicidad. Su sintaxis está inspirada en C, ligeramente modificada para adaptarla al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, le resultará muy fácil aprender PHP.
- Si bien es cierto que hay ciertas características avanzadas que presentan las plataformas J2EE o .NET y que PHP no las tiene, no todas las aplicaciones Internet ameritan tal grado de complejidad. PHP fácilmente puede cubrir más del 75% de las necesidades del mercado.
- Es multiplataforma, es decir, puede ser utilizado en cualquiera de los principales sistemas operativos del mercado actual y es soportado por la mayoría de los servidores Web.
- Es software libre, lo que implica menos costos y servidores más baratos, por lo que podemos usarlo en proyectos comerciales si queremos, sin tener que pagar por su licencia. El tiempo, es uno de los costos más altos que hay que tener en cuenta antes de empezar un proyecto. Para empezar, el tiempo de aprendizaje de PHP es muy corto

gracias a su simplicidad. Luego, el tiempo de desarrollo es también corto. Podríamos hacer más de un proyecto Web con PHP en el mismo tiempo que tomaría hacer un solo proyecto con Java o .NET. Otro aspecto que hay que tener en cuenta es el del hardware. Para desarrollar en PHP no se requiere tener grandes capacidades de hardware, como sí lo requieren los pesados IDEs para programar en Java o .Net. Luego, en el caso de los servidores, una aplicación en PHP no requiere tanta memoria de máquina como podría requerir una aplicación en Java con sus servidores de aplicaciones que podrían requerir hasta varios procesadores y varios Gigas de memoria RAM.

- Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, le resultará muy fácil aprender PHP.
- Su librería estándar es realmente amplia, lo que permite reducir los llamados "costos ocultos", uno de los principales defectos de ASP.

PHP tiene una de las comunidades más grandes en Internet, por lo que es fácil ((con lo que no es complicado)) encontrar ayuda, documentación, artículos, noticias, y demás recursos.

### Desventaja:

La legibilidad de código puede verse afectada al mezclar sentencias HTML y PHP.

### **2.11.5 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. [24]

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. 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.

### ¿Por qué PHP y XSLT?

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 de transformación de documentos, que servirá para definir la presentación del cliente.

### 2.12 Sistemas de Gestión de Bases de Datos (SGBD)

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. [25]

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.

#### 2.12.1 SQL Server

Microsoft SQL Server 2000 es uno de los mejores SGBD base de datos para Windows, es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto

significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.  
[26]

### Ventajas:

- Soporta la configuración automática y la auto-optimización.
- Administración multiservidor para un gran número de servidores.
- Gran variedad de opciones de duplicación de cualquier base de datos.
- Acceso universal a los datos (Universal Data Access).
- Fácil de usar.
- Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles.
- Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server.
- Posee los mejores registros de los benchmarks independientes (TCP), tanto en transacciones totales como en coste por transacción.
- 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.

### **2.12.2 MySql**

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.  
[27]

### Ventajas:

- Diseñado con el objetivo de aumentar la velocidad.
- Consume muy pocos recursos de CPU y memoria. Muy buen rendimiento.

- Tamaño del registro sin límite.
- Buena integración con PHP.
- Utilidades de administración (phpMyAdmin).
- Buen control de acceso usuarios-tablas-permisos.

### Inconvenientes:

- No soporta subconsultas.
- Es gratis para aplicaciones de código abierto, de lo contrario hay que pagar licencia comercial.

### **2.12.3 PostgreSql**

PostgreSQL posee una amplia licencia BSD (esta licencia básicamente consiste en que ves el código, puedes redistribuirlo y puedes modificarlo. Posee una estabilidad y confiabilidad legendaria nunca ha presentado caídas en varios años de operación de alta actividad. [28]

Fue diseñado para ambientes de alto volumen intentando estar a la altura de Óracle, Sybase o Interbase. Escala muy bien al aumentar el número de CPUs y la cantidad de RAM. Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial. Tiene mejor soporte para subselects, triggers, vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos. Sin embargo consume muchos recursos y no escala bien en la plataforma Windows.

### Ventajas:

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

### Inconvenientes:

- Consume bastantes recursos y carga más el sistema.

### ¿Por qué MySQL?

MySQL (Open Source): Según definición de sus autores: es un servidor de base de datos muy rápido, robusto, multitarea y multiusuario. Tiene enfoque relacional y soporta ANSI SQL. Es gratis y se puede bajar de [www.mysql.com](http://www.mysql.com). Su principal objetivo de diseño fue la velocidad, además de consumir muy pocos recursos, tanto de CPU como de memoria. Tiene excelentes utilidades de administración como backup y recuperación de errores. Tiene una excelente integración con PHP. No hay límites en el tamaño de los registros, además de tener un control de acceso de los usuarios bastante amplio.

Trabaja en diferentes plataformas además de soportar múltiples idiomas. Completo y optimizado uso del SQL, además de contener un MyODBC. En cuanto a seguridad confía en la propia del sistema a efectos de robo de las bases de datos, caída del sistema. Usa Listas de Control de Acceso para todas las conexiones, consultas y otras operaciones. [29]

En fin, es un servidor de Base de datos relacional, tiene interfases para desarrollo de clientes en diversos lenguajes, esta disponible en diferentes plataformas, es gratis y se puede disponer del código fuente.

### 2.13 Desarrollo basado en RUP bajo la herramienta Rational Rose

Cada día la producción de software busca adecuarse más a las necesidades del usuario, y como consecuencia aumenta el tamaño y complejidad de las aplicaciones. Por lo que para lograr la rentabilidad de estas producciones se necesita un proceso que integre las múltiples facetas de 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.

### **2.13.1 UML (Unified Modeling Lenguaje)**

UML (Unified Modeling Lenguaje) o Lenguaje de Modelación Unificado es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software). Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque Orientado a Objetos, utilizándose también en el diseño Web. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. [30]

UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas
- Tecnología orientada a objetos
- El cliente participa en todas las etapas del proyecto
- Corrección de errores viables en todas las etapas
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor

UML es desde finales de 1997, un lenguaje de modelado orientado a objetos estándar, de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle, Rational.

### **2.13.2 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 Rational 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 la herramienta del Rational para la etapa de análisis y diseño de sistemas.

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 interfaz en forma individual y luego unirlos con otros componentes del proyecto.

La metodología usada para desarrollar el proyecto fue RUP. Este es un proceso que garantiza la elaboración de todas las fases de un producto de software orientado a objeto. 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

### **2.14 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.

Para el diseño de las páginas Web será utilizado en una primera etapa el Dreamweaver MX de la familia Macromedia. También el Stylus Studio 5.1 que se utilizará por los desarrolladores para crear los ficheros XSL a través de escenarios.

Para la edición del código PHP se utilizará el NuSphere y PHPEd 3.3.3, este último es un editor para programadores con soporte para múltiples formatos, similar a otros editores como el Zend Studio, incluye un cliente de FTP y un servidor Web integrado, como servidor Web Apache 2.0 con PHP 4.3.4; el servidor de bases de datos se escogerá MySQL 4.x.x y PHPmyadmin como su front.

### **2.15 Conclusiones**

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

### **CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

#### **3.1 Introducción**

En este capítulo se describe la propuesta de la solución que fue desarrollada mediante el proceso unificado de software (RUP), que propone un desarrollo iterativo e incremental a través de sus flujos de trabajo y se describen los procesos que tienen que ver con el objeto de estudio.

Después del estudio de la arquitectura del sistema, la distribución de todos los módulos que componen el Sistema Integral de Salud, la gestión que necesita realizar el Centro de Control y por la poca estructuración del proceso estudiado, se utiliza un Modelo de Dominio que permite entender el contexto del problema y realizar la captura de los requisitos necesarios para desarrollar el sistema, agrupando los conceptos fundamentales que se definen en el problema. Este sistema será modelado y desarrollado bajo el marco del Sistema Informatizado de Atención Primaria, ubicado en el Proyecto de Informatización de Atención Primaria de Salud (APS). A partir de este estudio se definieron los requisitos funcionales y no funcionales, obteniéndose la relación de las acciones que realizan los actores en el sistema y su descripción.

#### **3.2 Modelo del Dominio**

Para una mejor comprensión del sistema por parte de los usuarios, clientes y desarrolladores, y por la poca estructuración de los procesos del negocio que tienen que ver con el objeto de estudio, se agrupa en un modelo los principales conceptos que permiten definir el sistema y tener una visión más concreta del objeto de estudio; esto contribuye a un eficiente levantamiento de los requisitos y tener una clara idea de las clases que pueden ser utilizadas en el desarrollo del sistema. Este modelo es conocido como Modelo de Dominio.



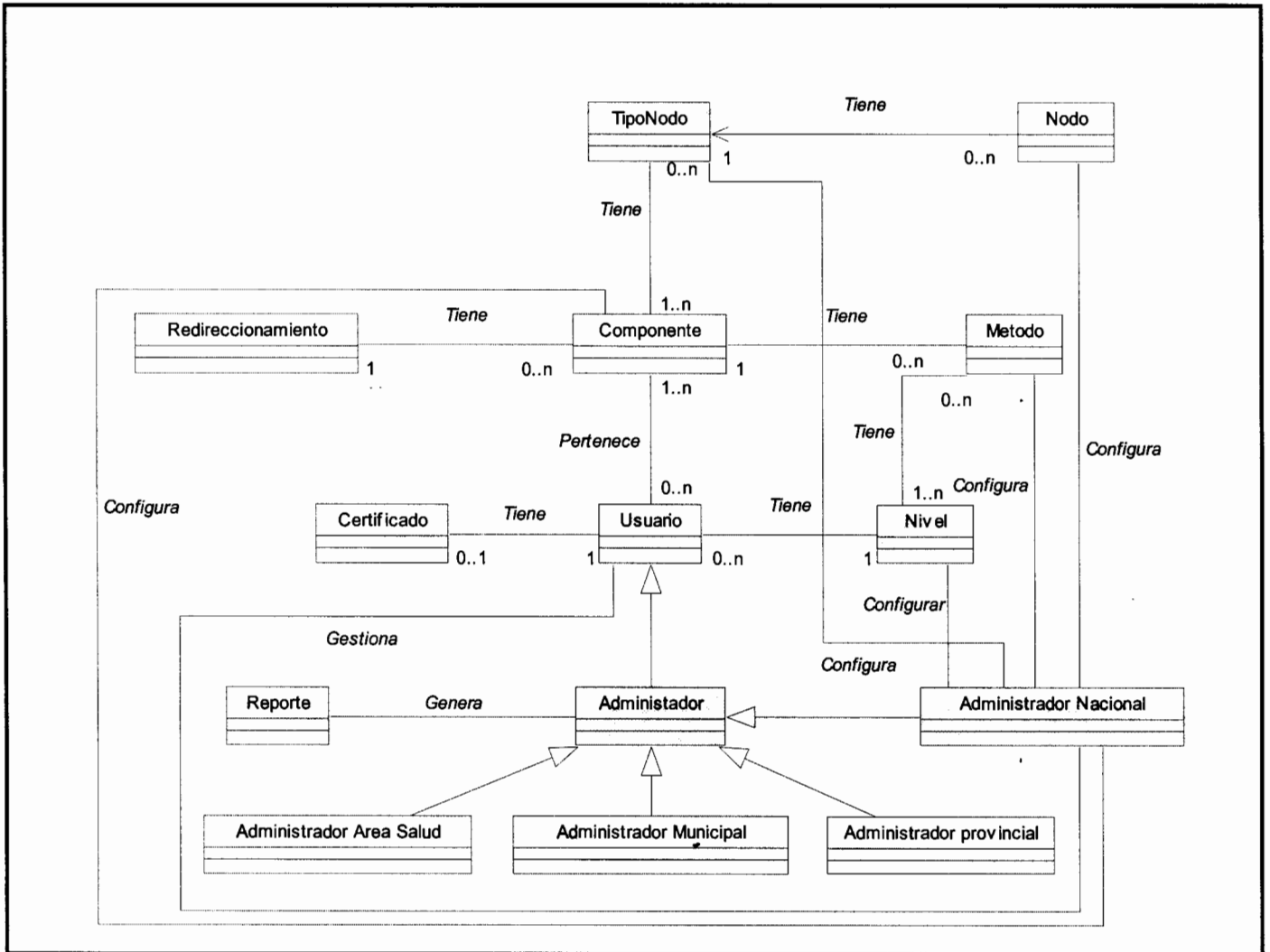


Figura 3.1 Diagrama del modelo del dominio.

### 3.3 Requerimientos Funcionales

Los requerimientos funcionales son las acciones que el sistema permite realizar para que se cumplan los objetivos planteados en el problema. Estas acciones son ejecutadas por los usuarios que utilizan el sistema o el sistema internamente lo resuelve. Una vez analizados los conceptos descritos en el diagrama del Modelo de Dominio se levantaron los siguientes requisitos funcionales:

#### R1 Autenticar Usuario.

Pedir nombre de usuario y contraseña para entrar al Centro de Control.

## **CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

El usuario registrado accederá a editar o visualizar solamente a la información que tiene permitida según su nivel.

Permitir que el usuario registrado que cierre su sesión de trabajo desde cualquier lugar del sistema.

### **R2 Cambiar Contraseña.**

Permitir que el usuario autenticado tenga la posibilidad de cambiar su contraseña, introduciendo siempre la contraseña anterior.

### **R3 Gestionar Usuario.**

Insertar Usuario.

Modificar Usuario.

Eliminar Usuario.

### **R4 Asignar Permisos.**

Asignar al usuario los módulos a los que tiene permiso y el tipo de permiso que tiene en ese módulo.

### **R5 Ubicar a los administradores.**

Ubicar al administrador en una provincia, municipio o área de salud según el nivel del que lo creó.

### **R6 Gestionar Nodo.**

Insertar Nodo.

Modificar Nodo.

Eliminar Nodo.

Buscar Nodo.

Listar Nodo.

### **R7 Gestionar Tipo de Nodo.**

Insertar Tipo de Nodo.

Modificar Tipo de Nodo

Eliminar Tipo de Nodo.

Listar Tipo de Nodo.

### **R8 Gestionar Componentes.**

Insertar Componente.

Modificar Componente.

Eliminar Componente.

Listar Componente.

### **R9 Gestionar Métodos.**

Insertar Métodos.

Modificar Métodos.

Eliminar Métodos.

Listar Método.

Buscar Método.

### **R10 Gestionar Nivel.**

Insertar Nivel.

Modificar Nivel.

Eliminar Nivel.

Listar Nivel.

### **R11 Emitir Reportes.**

Emitir Reporte de los usuarios.

Emitir Reporte de la configuración de los nodos.

## **3.4 Requerimientos No Funcionales**

Los requisitos no funcionales se refieren a las cualidades del Sistema.

### **3.4.1 Apariencia o Interfaz Externa**

La interfaz debe ser sencilla, intuitiva, amigable, debido a que los usuarios no son expertos en el uso de Aplicaciones Web.

### **3.4.2 Rendimiento**

Debe mantener un diseño similar en todas las páginas, las mismas estarán poco cargadas de imágenes, para garantizar una respuesta más rápida del Sistema.

### **3.4.3 Usabilidad**

La aplicación garantizará un acceso fácil y rápido a los usuarios. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general, por lo que el personal que trabaja con el Registro debe contar con el nivel técnico requerido mediante adiestramiento de servicio.

### **3.4.4 Portabilidad**

Permitir que el sistema se ejecute sobre el Sistema Operativo Linux, Windows 98 o superior.

### **3.4.5 Autonomía**

Cada capa se debe construir como componente independiente, facilitando el mantenimiento del software.

### **3.4.6 Seguridad**

Disponer de un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoria (**AAA**).

**Confiabledad:** La información manejada por el sistema está protegida de acceso no autorizado. El sistema debe prevenir posibles fallos y/o errores y presentar facilidades para una rápida recuperación en dichos casos.

**Integridad:** La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes. Deberán existir mecanismos de chequeo de integridad.

## **CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

Se permitirá la creación de copias de respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.

**Disponibilidad:** Los usuarios autorizados tendrán acceso a la información en todo momento. Deberá existir una estrategia de replicación que permita, de manera transparente para el usuario final, balancear la carga de acceso entre múltiples servidores aumentando los tiempos de respuesta y facilitar la recuperación inmediata del sistema si falla uno de ellos.

### **3.4.7 Escalabilidad**

La arquitectura del sistema debe desarrollarse siguiendo un modelo cliente servidor de varias capas, Interfaz de usuario (presentación), Lógica de negocio, Acceso a Datos y Base de datos, con una arquitectura orientada a servicios y basada en componentes. La capa de negocio estará disponible a otras aplicaciones utilizando las tecnologías de *xml web services*.

Todos los componentes del sistema deben desarrollarse siguiendo el principio de máxima cohesión y mínimo acoplamiento.

### **3.4.8 Software**

Los clientes tendrán acceso al Registro de Partos y Nacimientos a través de cualquier navegador Web. Recomendados: Mozilla 1.5, Internet Explorer 4.0 o superior.

### **3.4.9 Hardware**

Requerimientos mínimos del sistema:

Procesador 486DX / 66 MHz o superior.

16 MB de memoria; a más memoria mayor rendimiento.

Monitor VGA o superior.

Ratón Microsoft o compatible.

No existen restricciones específicas en cuanto al Servidor.

RNF16 Impresora local o de red para imprimir los reportes solicitados.

### **3.5 Descripción del Sistema propuesto**

#### **3.5.1 Concepción general del sistema**

Teniendo en cuenta los requisitos funcionales y no funcionales descritos en el epígrafe anterior, se presenta la propuesta de un sistema llamado Centro de Control, que es un componente estrechamente ligado a la arquitectura del sistema y cuyos objetivos principales son garantizar la seguridad, mantener la disponibilidad de la información y la conectividad entre los diferentes componentes que forman parte del Sistema Integral de Salud, atendiendo a que estos se encuentran totalmente distribuidos y no tienen un comportamiento similar.

Por estas razones, se necesita tener centralizada toda la información sobre los usuarios que trabajan con los componentes y los permisos que poseen los mismos sobre los diferentes módulos; además que la distribución y el comportamiento que siguen los componentes tenga su propia configuración regional y sigan un comportamiento determinado, para lograr gestionar la integración y mantener una disponibilidad máxima de la información que se maneja, que debe estar disponible en todo momento.

.Para el cumplimiento de esta tarea se guarda en una base de datos toda la información necesaria para realizar esta gestión. Se necesita tener guardada en una base de datos la dirección **URL** en la que se encuentran los diferentes componentes, y teniendo en cuenta que se tienen ubicaciones con los mismos componentes, existe una entidad que tiene un tipo que identifica a nodos que tienen similares distribución de componentes y que guarda además la información sobre todos los componentes que pertenecen a esa ubicación. De esta manera para cada ubicación similar solo será necesario asignarle el tipo a la cual pertenece y no tener que asignarle a cada nodo similar componente por componente. (Ver Anexo V).

Por problema de seguridad y también de gestión, se necesita tener la relación de los métodos que pertenecen a cada componente y sus datos, por ultimo, y no menos importante, la relación de los usuarios y los permisos que tienen para trabajar con los diferentes módulos, así como la relación de los usuarios administradores encargados de mantener actualizada toda esta información según el nivel al que pertenecen.

## CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

Con este objetivo existe un administrador nacional, que es el único que tiene permiso para realizar la configuración de la distribución de los componentes, esto incluye configurar nodo, tipos de nodo y asignarle los componentes que va a tener ese nodo, así como los métodos que tiene cada componente, y puede definir nuevos niveles atendiendo a la posibilidad de que en determinado momento se necesite escalar los niveles existentes a niveles más específicos.

La gestión de los usuarios es realizada por los diferentes administradores de su mismo nivel y ubicación, desde el nivel nacional hasta el más bajo que exista, y si el usuario es administrador su información será gestionada por administradores de un nivel superior.

### **3.6 Modelo de casos de uso del sistema**

Se definen los actores que van a interactuar con el sistema para hacer una descripción detallada de las secuencias de acciones que realizan, para llevar a cabo los requisitos funcionales que han sido definidos en el sistema.

<b>ACTORES</b>	<b>JUSTIFICACIÓN</b>
Administrador Nacional	Es el usuario que tiene permiso total sobre el sistema. Puede crear usuarios y modificar sus datos y es el único encargado de la configuración de la distribución de los Componentes que incluye gestionar los nodos, tipos de nodos, los componentes, los métodos y los niveles.
Administrador	Es el usuario que en dependencia de su nivel que puede ser nacional, provincial, municipal, área de salud u otro nivel que se defina tiene los permisos para gestionar la información sobre los usuarios que están en su mismo nivel y ubicación, así como los administradores que pertenecen al nivel inmediato inferior. Esta gestión incluye crear usuarios o modificar su información.

Tabla 3.1 Actores del Sistema.

**3.6.1 Diagrama de Casos de Uso del Sistema**

El diagrama de los casos de uso del sistema muestra como se relacionan los actores y los casos de usos que han sido definidos en el sistema. (Ver Figura 3.2).

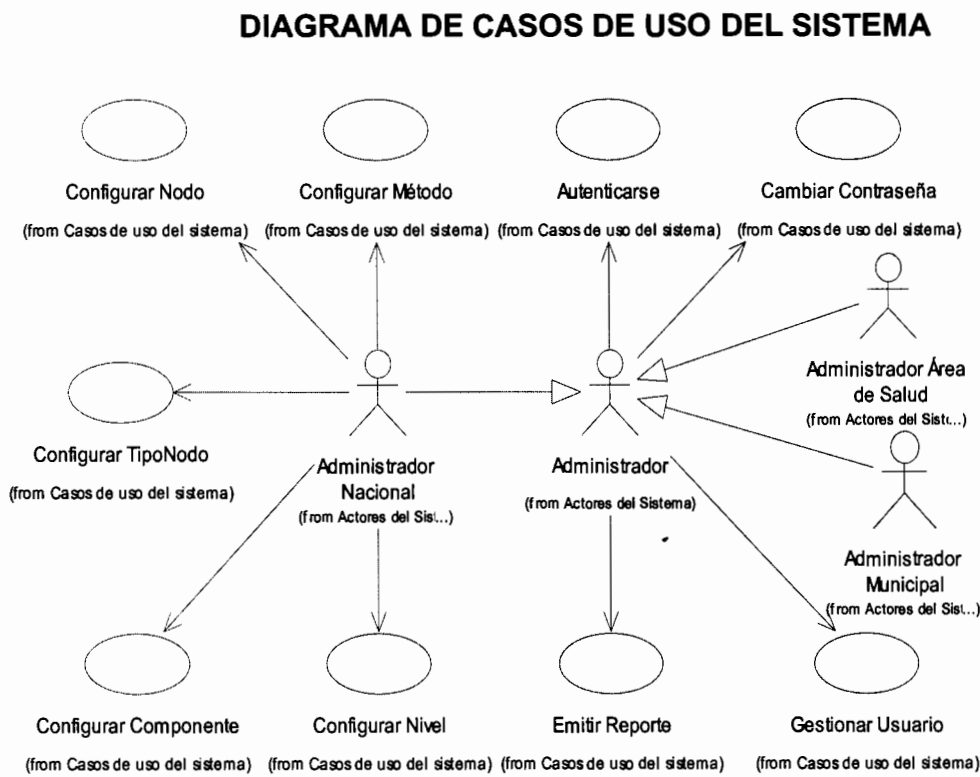


Figura 3. 2 Diagrama de Casos de Uso del Sistema.

**3.7 Casos de Uso**

Los casos de uso son la representación visual de la interacción de un actor con el sistema. A continuación se brinda la descripción de cada uno de los casos de uso que se definieron para el sistema en análisis.

<b>Caso de uso</b>	Autenticarse
<b>Actor</b>	Administrador (inicia)



### CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Propósito</b>	Iniciar sección en el Centro de Control.
<b>Resumen</b>	El caso de uso se inicia cuando un administrador quiere acceder al Centro de Control. Para esto introduce su usuario y contraseña, el sistema verifica los datos, si son incorrectos le informa al administrador mediante un mensaje. En caso contrario el administrador accede al sistema y este crea un certificado para el usuario con la hora y fecha en que se registro y los permisos que tiene. El caso de uso termina cuando el administrador accede al centro de control.
<b>Referencia</b>	R1
<b>Precondiciones</b>	-
<b>Poscondiciones</b>	El administrador accede al centro de control, se crea un certificado según su nivel y el permiso que tiene.

Interfaz Caso de Uso Autenticarse.

(Ver Anexo VI)

<b>Caso de uso</b>	Cambiar Contraseña
<b>Actor</b>	Administrador (inicia)
<b>Propósito</b>	Cambiar contraseña.
<b>Resumen</b>	El caso de uso se inicia cuando un administrador desea cambiar su contraseña. Para esto el debe proporcionar su usuario y contraseña anterior e introducir su nueva contraseña y confirmarla. El sistema verifica que los datos sean correctos, si no lo son el sistema le muestra un mensaje de error al administrador. En caso contrario la contraseña se cambia y termina el caso de uso. El caso de uso termina cuando el administrador cambia la contraseña.
<b>Referencia</b>	R2
<b>Precondiciones</b>	El administrador debe estar autenticado.
<b>Poscondiciones</b>	Se cambia la contraseña del administrador

Interfaz Caso de Uso Cambiar Contraseña.

(Anexo VII)

<b>Caso de uso</b>	Gestionar Usuario
<b>Actor</b>	Administrador (inicia)
<b>Propósito</b>	Administrar usuario
<b>Descripción</b>	El caso de uso se inicia cuando el administrador accede al Centro de Control para gestionar la información de los usuarios. Si desea insertar un nuevo usuario en el menú principal escoge la opción usuario->nuevo. El sistema le muestra una interfaz para que entre los datos del usuario. Estos datos son nombre y apellidos, usuario, contraseña y un campo para mostrar si el usuario es administrador. En caso que el usuario a crear sea administrador el administrador debe ubicarlo una provincia, municipio o unidad de salud en dependencia de su nivel. En caso contrario el administrador debe asignar al usuario los componentes a los que tiene permisos y el tipo de permiso que tiene en cada uno. Cuando el administrador introduce los datos y acepta el sistema verifica que el

## CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

	<p>usuario no haya sido insertado, lo inserta y muestra un mensaje. En caso contrario muestra un mensaje de error y pide los datos nuevamente.</p> <p>En el caso del administrador querer eliminar un usuario o modificarlo, escoge la opción usuario -&gt; buscar, el sistema muestra una pantalla de búsqueda con criterios para que el administrador restrinja la búsqueda y encuentre más fácilmente el usuario que quiere eliminar o modificar, estos criterios son: componente al que pertenece, tipo de usuario y según el nivel la provincia, municipio y área de salud a la que pertenece. El administrador selecciona el usuario y si escoge la opción de eliminar, el sistema le muestra una ventana de verificación. En caso de aceptar lo elimina. Si escoge la opción de modificar, el sistema carga en una pantalla los datos registrados del usuario seleccionado, el administrador modifica los campos que desee y acepta. El caso de uso termina cuando se inserta, elimina o se actualiza la información de un usuario.</p>
<b>Referencia</b>	R3,R4,R5
<b>Precondiciones</b>	El administrador debe estar autenticado y debe estar actualizada la información sobre los componentes, los niveles y los permisos
<b>Poscondiciones</b>	Se crea un usuario, se elimina un usuario o se modifica su información.

Interfaz Caso de Uso Gestionar Usuario.

(Ver Anexo VIII)

<b>Caso de uso</b>	Configurar Nodo
<b>Actor</b>	Administrador Nacional (inicia).
<b>Propósito</b>	Configurar un nodo.
<b>Descripción</b>	<p>El caso de uso se inicia cuando el administrador accede al Centro de Control para realizar la configuración de los nodos. Si desea insertar un nuevo nodo en el menú principal escoge la opción configurar-&gt;nodo-&gt;nuevo. El sistema le muestra una interfaz para que entre los datos del nodo. Estos datos son nombre del nodo, dirección local y el tipo de nodo que tiene asignado. Cuando el administrador introduce los datos y acepta el sistema verifica que el nodo no haya sido insertado, lo inserta y muestra un mensaje. En caso contrario muestra un mensaje de error y pide los datos nuevamente.</p> <p>En el caso del administrador querer eliminar un nodo o modificarlo escoge la opción Configurar-&gt;nodo-&gt;buscar. El sistema muestra una pantalla de búsqueda con criterios para que el administrador inserte criterios y encuentre más fácilmente el nodo que quiere eliminar o modificar. El administrador selecciona el nodo y si escoge la opción de eliminar, el sistema le muestra una ventana de verificación. En caso de aceptar, el sistema verifica que el nodo no tenga asociados un tipo de nodo y lo elimina. En caso contrario alerta al administrador que no se pudo eliminar. Si escoge la opción de modificar el sistema carga en una pantalla los datos registrados del nodo seleccionado. El administrador modifica los campos que desee y acepta. El caso de uso termina cuando el sistema guarda los cambios realizados.</p>

### CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Referencia</b>	R6
<b>Precondiciones</b>	El administrador debe estar autenticado. La información de los tipos de nodos debe estar actualizada.
<b>Poscondiciones</b>	Se eliminan, insertan o modifican los datos de los nodos y se le asigna el tipo de nodo que le corresponde.

Interfaz Caso de Uso Configurar Nodo.

(Ver Anexo IX)

<b>Caso de uso</b>	Configurar TipoNodo
<b>Actor</b>	Administrador Nacional(inicia)
<b>Propósito</b>	Configurar un tipo de nodo
<b>Descripción</b>	<p>El caso de uso se inicia cuando el administrador accede al Centro de Control para configurar los tipos de nodos. Si desea insertar un nuevo tipo de nodo en el menú principal escoge la opción configurar-&gt;Tipodenodo-&gt;nuevo. El sistema le muestra una interfaz para que entre los datos, estos datos son tipo del nodo, y los componentes que están asociados a él. Cuando el administrador introduce los datos y acepta la inserción, el sistema verifica que el tipo de nodo no haya sido insertado, lo inserta y muestra un mensaje al administrador. En caso contrario muestra un mensaje de error y pide los datos nuevamente.</p> <p>En el caso del administrador querer eliminar un tipo de nodo o modificarlo escoge la opción componente -&gt; listar. El sistema muestra una pantalla con la lista de todos los tipos de nodos. El administrador selecciona el tipo de nodo y si escoge la opción de eliminar el sistema le muestra una ventana de verificación. En caso de aceptar el sistema verifica que el tipo de nodo no forma parte de ningún nodo y lo elimina. En caso contrario muestra un mensaje al administrador alertando que no puede ser eliminado. Si el administrador escoge la opción de modificar, el sistema carga en una pantalla los datos registrados del tipo de nodo seleccionado, el administrador modifica los campos que desee y acepta. El sistema guarda los cambios realizados.</p>
<b>Referencia</b>	R7
<b>Precondiciones</b>	El administrador debe estar autenticado y la información de los componentes actualizada.
<b>Poscondiciones</b>	Se eliminan, insertan o modifican los datos de los tipos de nodos.

Interfaz Caso de Uso Configurar TipoNodo.

(Ver Anexo X)

<b>Caso de uso</b>	Configurar Componente (inicia)
<b>Actor</b>	Administrador Nacional
<b>Propósito</b>	Configurar un componente
<b>Descripción</b>	El caso de uso se inicia cuando el administrador accede al Centro de

## CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

	<p>Control para configura un componente. Si desea insertar un nuevo componente en el menú principal escoge la opción componente -&gt;nuevo. El sistema le muestra una interfaz para que entre los datos del componente. Estos datos son nombre del componente, el tipo de redireccionamiento, la dirección nacional y el acrónimo del componente. Cuando el administrador introduce los datos y acepta, el sistema verifica que el componente no haya sido insertado, lo inserta y muestra un mensaje al administrador. En caso contrario muestra un mensaje de error y pide los datos nuevamente.</p> <p>En el caso del administrador querer eliminar un componente o modificarlo, escoge la opción componente -&gt; listar. El sistema muestra una pantalla con la lista de todos los componentes. El administrador selecciona el componente y si escoge la opción de eliminar, el sistema le muestra una ventana de verificación. En caso de aceptar, el sistema verifica que el componente no forma parte de ningún tipo de nodo y que los el o los usuarios asociados a el pertenecen a otro componente y lo elimina. En caso contrario muestra un mensaje al administrador alertando que no puede ser eliminado. Si el administrador escoge la opción de modificar, el sistema carga en una pantalla los datos registrados del componente seleccionado. El administrador modifica los campos que desee y acepta, el sistema guarda los cambios realizados.</p>
<b>Referencia</b>	R8
<b>Precondiciones</b>	El administrador debe estar autenticado y la información de los usuarios que pertenecen al componente y los tipos de nodos debe estar disponible y actualizada.
<b>Poscondiciones</b>	Se insertan, eliminan y actualizan los datos de los componentes.

Interfaz Caso de Uso Configurar Componente.

(Ver Anexo XI)

<b>Caso de uso</b>	Configurar Método
<b>Actor</b>	Administrador Nacional(inicia)
<b>Propósito</b>	Configurar un método
<b>Descripción</b>	<p>El caso de uso se inicia cuando el administrador accede al Centro de Control para gestionar la información de los métodos. Si desea insertar un nuevo método en el menú principal escoge la opción método -&gt;nuevo. El sistema le muestra una interfaz para que entre los datos del método; estos datos son nombre del método, el nivel que puede seleccionar un nivel o varios, el tipo de método, que puede ser de edición o visualización. Marca si el método es confiable o no y el componente al que pertenece. Cuando el administrador introduce los datos y acepta, el sistema verifica que el método no haya sido insertado, lo inserta y muestra un mensaje. En caso contrario muestra un mensaje de error y pide los datos nuevamente.</p> <p>En el caso del administrador querer eliminar un método o modificarlo, escoge la opción método -&gt; buscar. El sistema muestra una pantalla de</p>

### CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

	búsqueda con criterios para que el administrador inserte criterios y encuentre más fácilmente el método que quiere eliminar o modificar. El administrador selecciona el método y si escoge la opción de elimina, el sistema le muestra una ventana de verificación. En caso de aceptar el sistema lo elimina. Si escoge la opción de modificar, el sistema carga en una pantalla los datos registrados del método seleccionado. El administrador modifica los campos que desee y acepta. El sistema guarda los cambios realizados.
<b>Precondiciones</b>	El administrador debe estar autenticado y la información de los componentes debe estar actualizada.
<b>Poscondiciones</b>	Se actualiza la información de los métodos, se elimina o se inserta un método a un componente.
<b>Referencia</b>	R4

Interfaz Caso de Uso Configurar Componente.

(Ver Anexo XII)

<b>Caso de uso</b>	Configurar Nivel
<b>Actor</b>	Administrador Nacional(inicia)
<b>Propósito</b>	Configurar un nivel
<b>Descripción</b>	<p>El caso de uso se inicia cuando el administrador accede al Centro de Control para configurar la información de un nivel. Si desea insertar un nuevo nivel en el menú principal escoge la opción nivel -&gt;nuevo. El sistema le muestra una interfaz para que entre el nombre del nivel, el sistema verifica que el nivel no haya sido insertado, lo inserta y muestra un mensaje. En caso contrario muestra un mensaje de error y pide el dato nuevamente.</p> <p>En el caso del administrador querer eliminar un nivel o modificarlo, escoge la opción nivel-&gt; buscar. El sistema muestra una pantalla de búsqueda con criterios para que el administrador inserte criterios y encuentre más fácilmente el método que quiere eliminar o modificar. El administrador selecciona el método y si escoge la opción de eliminar, el sistema le muestra una ventana de verificación. En caso de aceptar el sistema lo elimina. Si escoge la opción de modificar, el sistema carga en una pantalla los datos registrados del método seleccionado. El administrador modifica los campos que desee y acepta. El sistema guarda los cambios realizados.</p>
<b>Precondiciones</b>	El administrador debe estar autenticado y la información de los componentes, nivel y tipo debe estar actualizada.
<b>Poscondiciones</b>	Se actualiza la información de los métodos, se elimina o se inserta un método a un componente.
<b>Referencia</b>	R10

Interfaz Caso de Uso Configurar Nivel

(Ver Anexo XIII)

## CAPÍTULO 3 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Caso de uso</b>	Emitir Reporte
<b>Actor</b>	Administrador
<b>Propósito</b>	Tener un reporte de los usuarios y administradores del sistema y de la configuración de los nodos.
<b>Descripción</b>	Un administrador necesita visualizar o imprimir información sobre los usuarios y la configuración de los nodos. Para el reporte de los usuarios realiza el reporte por varios criterios. Estos criterios son si el usuario es administrador, el nivel que tiene y de acuerdo al nivel la provincia, el municipio o el área de salud a la que pertenece. Según el nivel del administrador que acceda al centro de control, será la información a la que accederá. Un administrador provincial solo le corresponde visualizar la información de su provincia, el administrador municipal del municipio al que pertenece y el administrador del área de salud a la información de los usuarios de su área de salud
<b>Referencia</b>	R11
<b>Precondiciones</b>	El administrador debe estar autenticado, la información de los usuarios, nodos, tipos de nodos, componentes, niveles y métodos actualizada.
<b>Poscondiciones</b>	Se muestran los reportes y si el administrador desea lo imprime

Interfaz Caso de Uso Emitir Reporte.

(Ver Anexo XIV)

### 3.8 Conclusiones

Este capítulo es fundamental para la comprensión de los clientes, usuarios y desarrolladores del sistema que se está modelando y es la base de la construcción del software.

Aquí se obtuvieron los diferentes requerimientos, tanto funcionales como no funcionales y se describieron de forma detallada las diferentes acciones que realizan los actores en el sistema. Se generaron diagramas fundamentales con conceptos claros para discutir con el cliente y se hizo la propuesta de solución basada en los diferentes diagramas, requisitos y casos de uso.

### **CAPITULO 4 CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN**

#### **4.1 Introducción**

Este capítulo es una secuencia de la descripción de la solución propuesta que toma como base la descripción de los casos de usos, los requisitos funcionales y no funcionales y el dominio del problema para modelar los objetos que interactúan para darle solución, presentándolos a través de diagramas de clases, preparándolo para la siguiente fase de implementación. Modelamos también el modelo de datos que se nos convertirá en base de datos del sistema, los diagramas de despliegue que nos muestran como se relacionan los diferentes paquetes del sistema y especificaremos los principales principios de diseño de la aplicación.

#### **4.2 Diagrama de Clases**

Una de las características más relevantes de la notación UML es su capacidad para absorber nueva semántica sin romper su lógica interna. La necesidad de implementar Servicios Web a través de complejas arquitecturas con múltiples capas de componentes y una gran dispersión geográfica de nodos, ha supuesto todo un reto al abordar su modelado y especificación. Por lo que Jim Conallen ha desarrollado desde 1998 una extensión de la notación UML denominada WAE "Web Application Extensión" que permite rentabilizar toda la gramática interna de UML para modelar aplicaciones con elementos específicos de la arquitectura de un entorno Web.

Alguien dijo alguna vez:

"UML es una caja de herramientas, como un cajón de sastre, no tienes por qué usarlas todas y ni siquiera tienes por qué usarlas para lo que está especificado que se usen".

El proceso de modelado que se ha ejecutado de los módulos que formarán parte del Sistema Integral de Salud en su primera prioridad, se realizó hasta la etapa de diseño, utilizando los estereotipos que ofrece el lenguaje UML, resultando suficientes para dicho modelado.

A pesar de estar creadas todas las condiciones para usar la notación UML denominada WAE por las características propias del proyecto y la arquitectura de los módulos que están en desarrollo, no se utilizaron estos estereotipos Web por la forma en que se ha organizado y ejecutado el Proyecto APS.

## **CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA**

---

La capacitación y preparación acelerada en el uso de las herramientas de modelado con Rational Rose y del lenguaje UML, por parte de todos los integrantes del equipo de trabajo, para modelar todos los procesos de los módulos de esta primera etapa, no dejó espacio de tiempo para continuar la modelación incorporando también el estudio de los temas relacionados con los estereotipos web y a la vez aplicarlos.

Existen además plazos ambiciosos de entrega de los módulos actuales y compromisos con la máxima dirección del MINSAP, de comenzar la implantación piloto de los módulos que se comiencen a liberar en la fase de implementación, en 9 de los 445 policlínicos a partir de septiembre del presente año.

Se considera que para el proceso de modelado de las próximas etapas del proyecto se deben usar estos estereotipos Web, aplicando toda la experiencia acumulada en la concepción de la etapa actual.

<b>.Clase</b>	<b>Propósito</b>
Dbz_class	Clase para la conexión con bases de datos MySql, usa el modulo dbx de PHP para su funcionalidad. Además crea un objeto conexión que permite hacer consultas, y recuperar los resultados; insertar, eliminar y actualizar datos. Esta clase estará en la capa de negocio. Esta clase está dentro de un framework llamado PLASER utilizado por la empresa para la reutilización de los distintos componentes y lograr una calidad adecuada, que implementa el patrón de diseño Modelo Vista Controlador (MVC).
Fachada	Clase general que se encuentra dentro del framework PLASER, que permite aplicar un encapsulamiento al mismo, creándose de esta forma un nivel de abstracción, logrando una mejor arquitectura del sistema.
FachadaCentroControl	Clase que hereda de Fachada, que aplica una fachada en la capa de presentación, disminuyendo así la carga de negocio en gran medida. Esta clase implementa el patrón Fachada, de esta forma la aplicación solo le hará las peticiones a esta clase.

Tabla 4.1 Descripciones de las clases generales del Diseño.



# CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

## Diagrama de clases: Caso de Uso Autenticarse

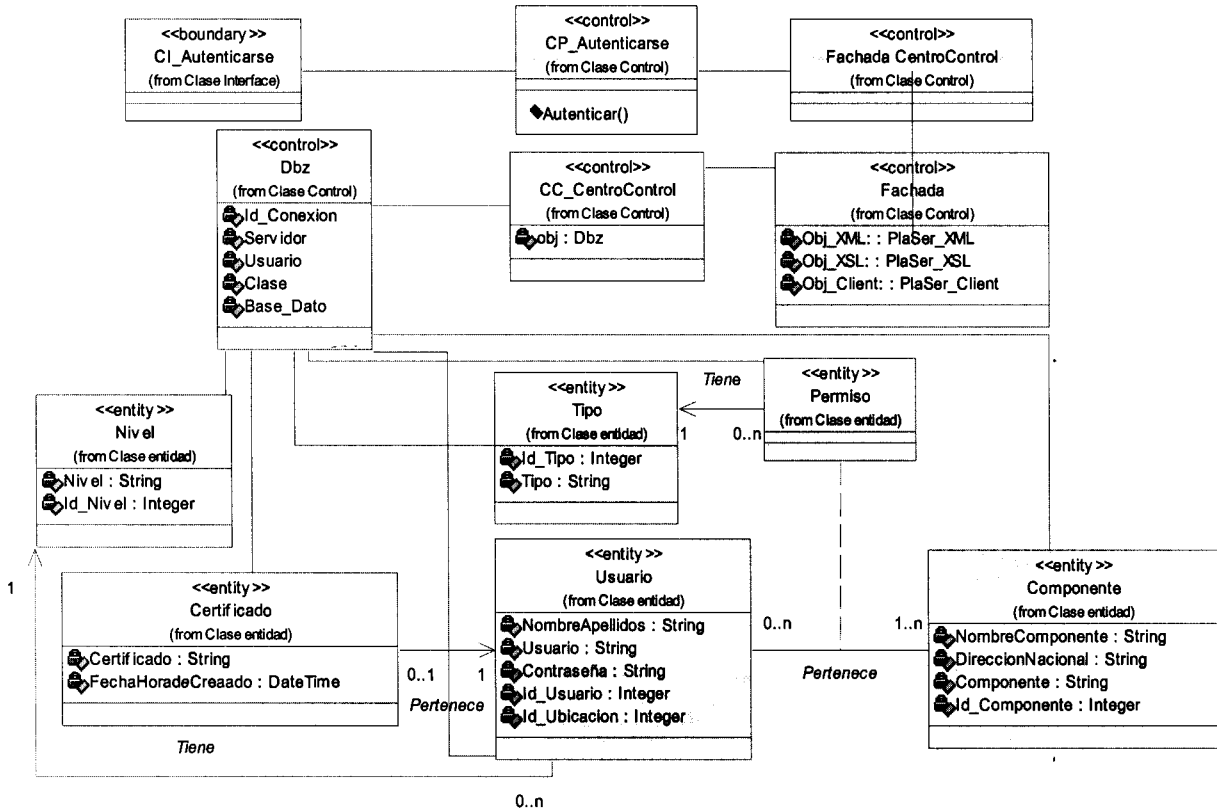


Figura 4.1 Diagrama de clases: Caso de Uso Autenticarse.

## Diagrama de clases: Caso de Uso Cambiar Contraseña

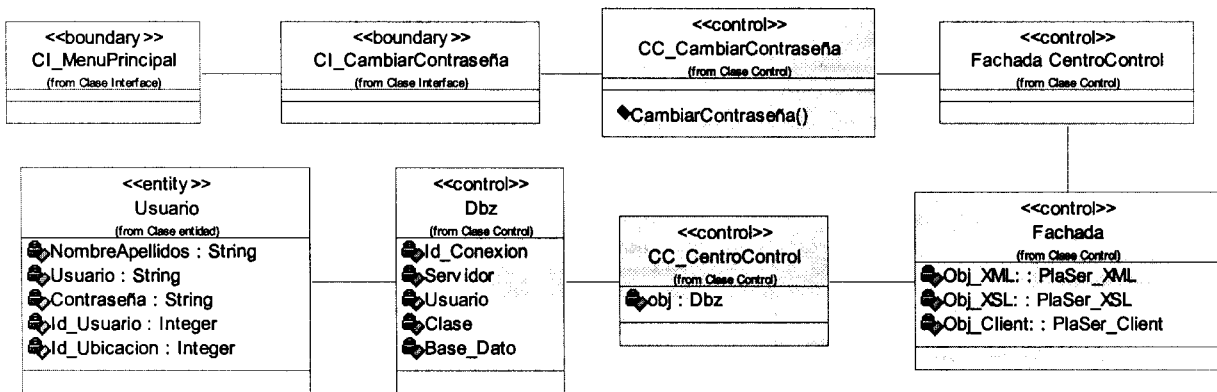


Figura 4.2 Diagrama de clases: Caso de Uso Cambiar Contraseña.

## Diagrama de clases: Caso de Uso Configura Nodo

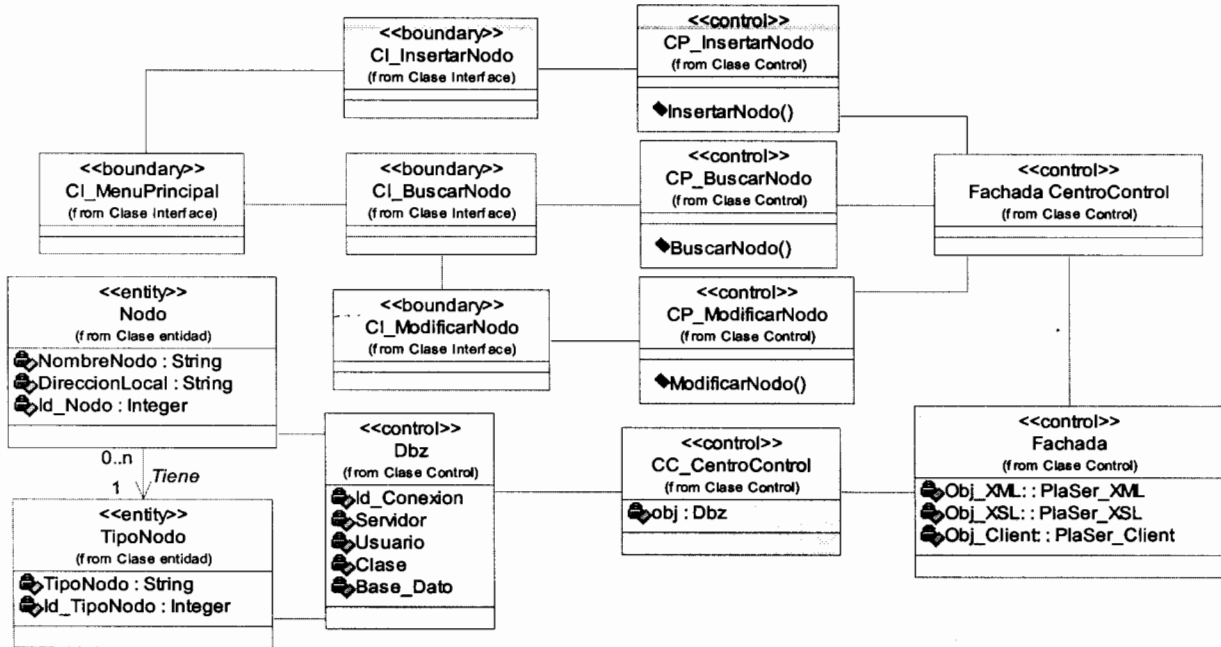


Figura 4.3 Diagrama de clases: Caso de Uso Configurar Nodo.

## CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### Diagrama de clases: Caso de Uso Configurar Tipo de Nodo

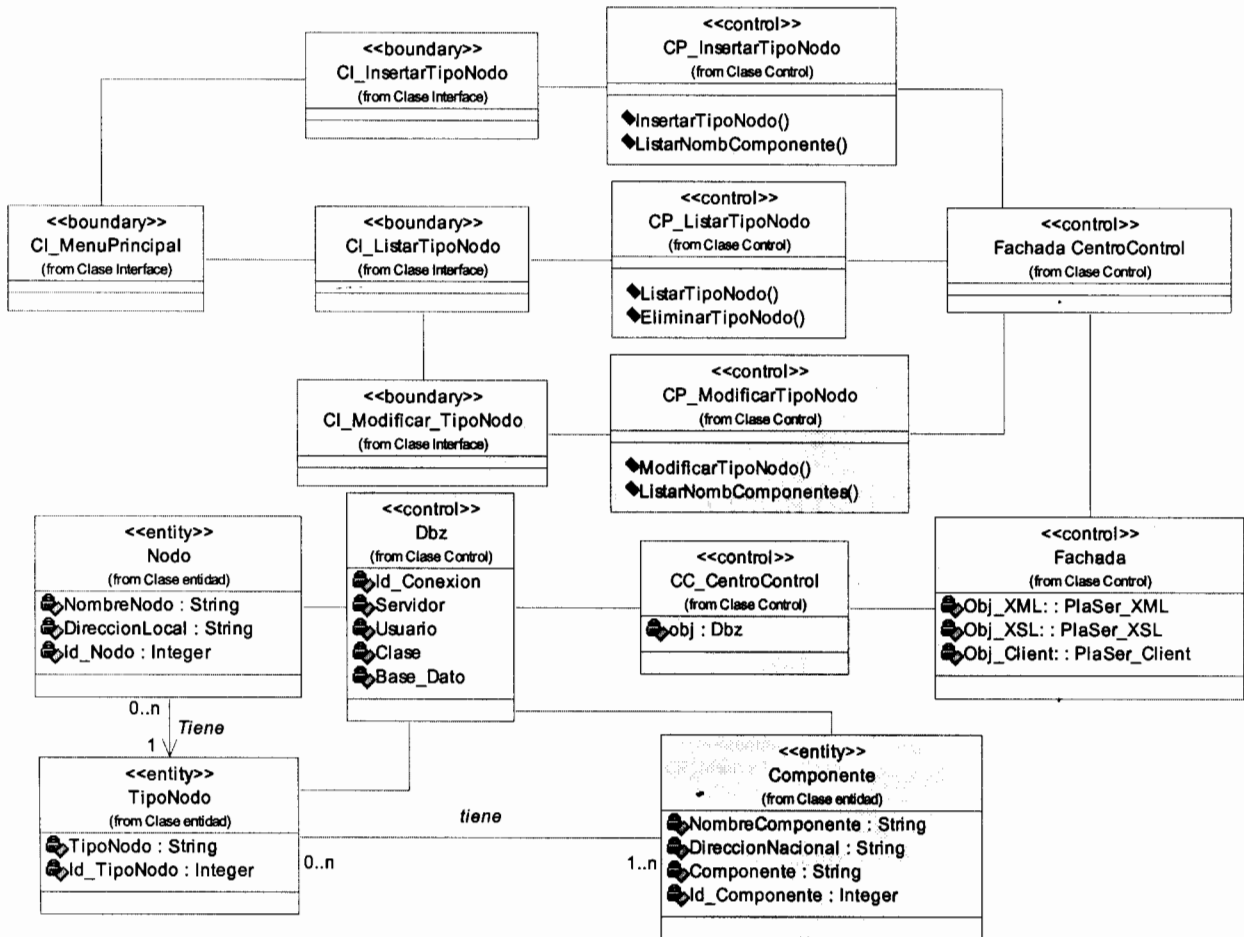


Figura 4.4 Diagrama de clases: Caso de Uso Configurar Tipo de Nodo.

## Diagrama de clases : Caso de Uso Configurar Componente

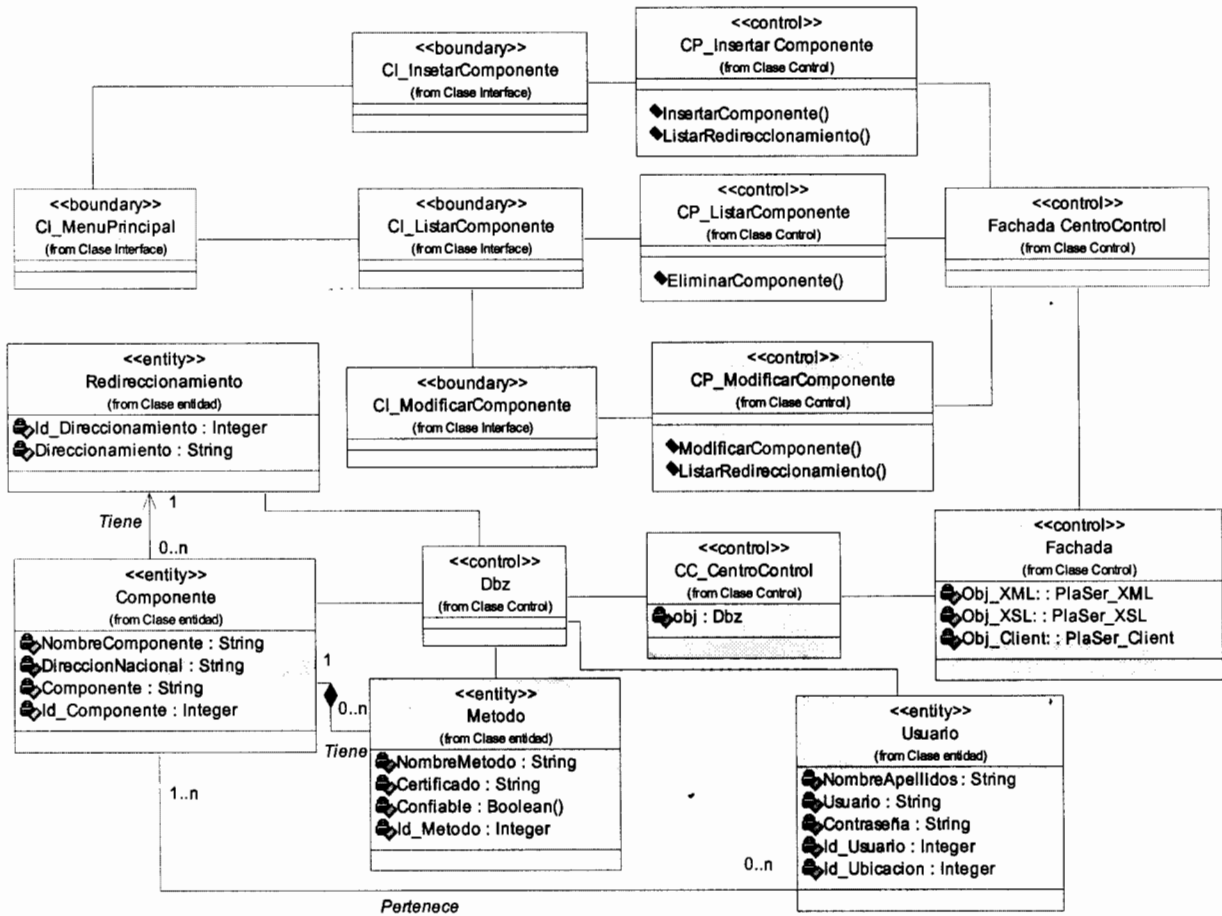


Figura 4.5 Diagrama de clases: Caso de Uso Configurar Componente.

## CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### Diagrama de clases: Caso de Uso Configurar Método

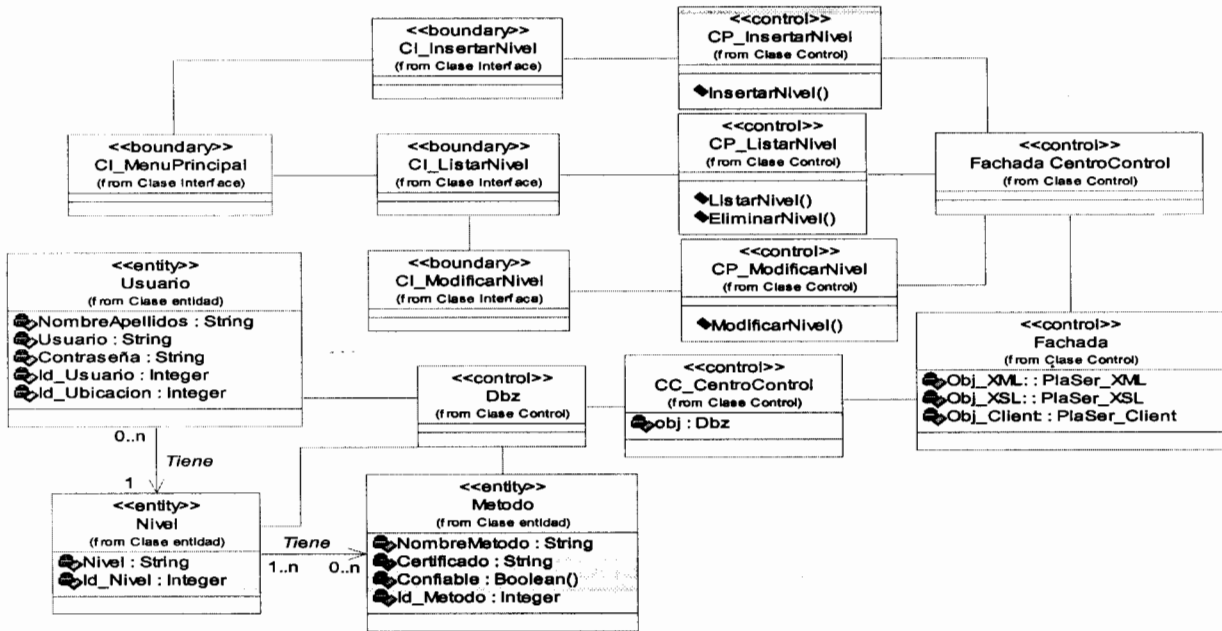


Figura 4.6 Diagrama de clases: Caso de Uso Configurar Método.

### Diagrama de clases : Caso de Uso Configurar Nivel

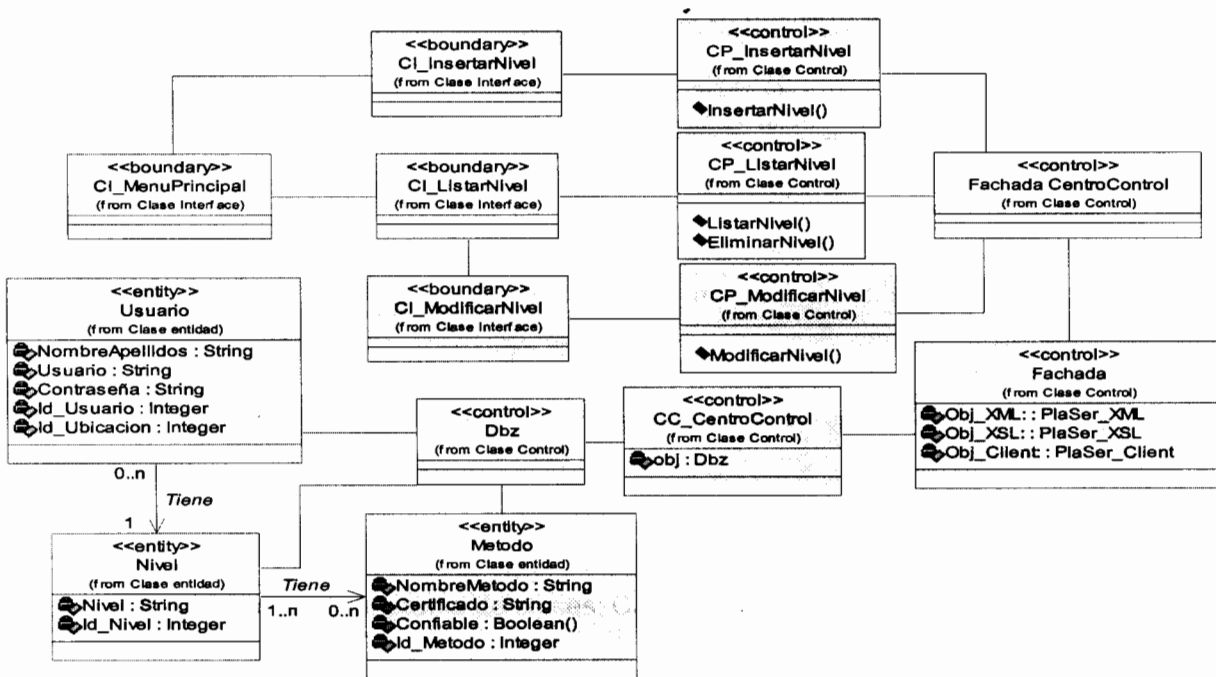


Figura 4.7 Diagrama de clases: Caso de Uso Configurar Nivel

## Diagrama de clases: Caso de Uso Gestionar Usuario.

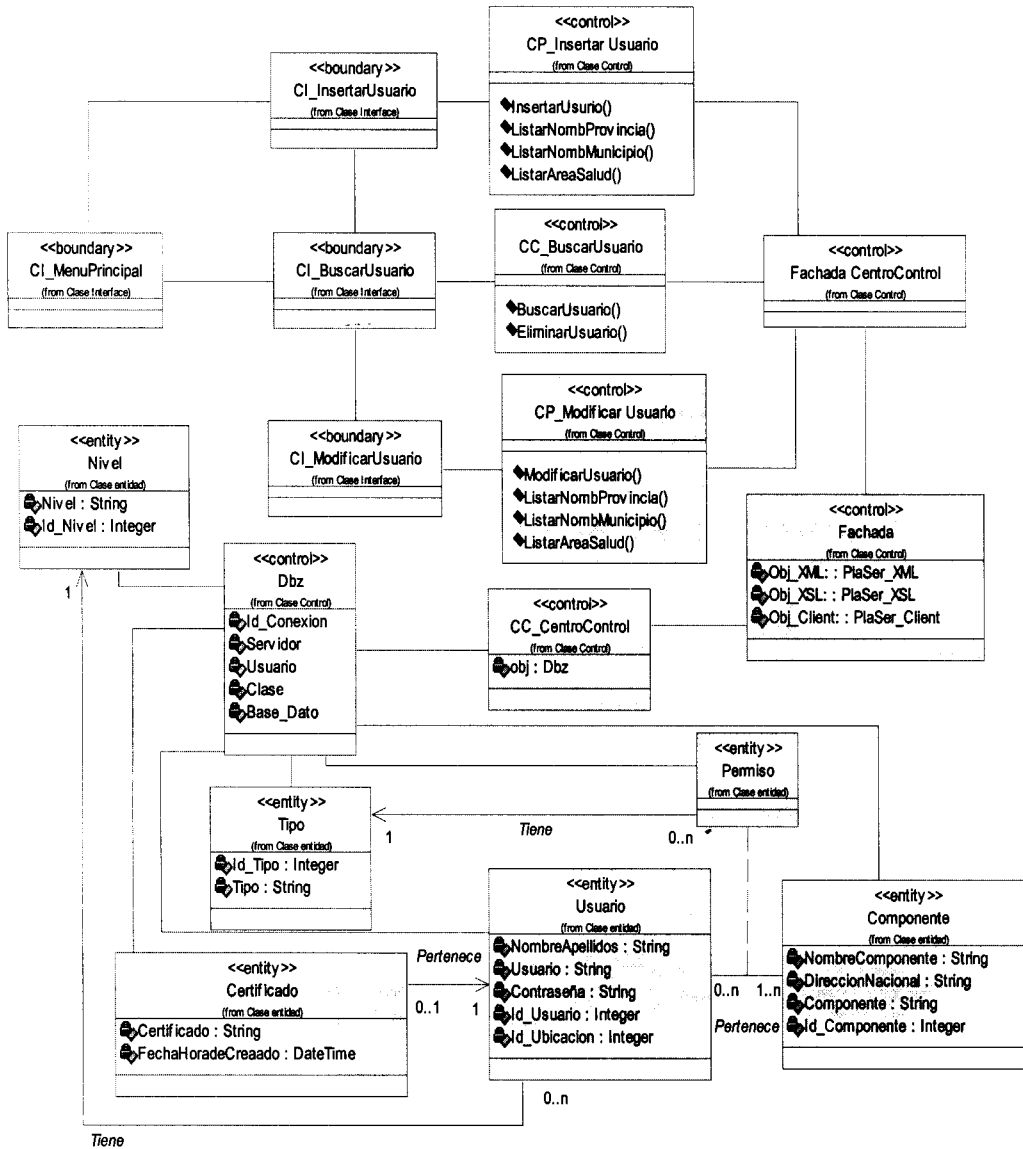


Figura 4.8 Diagrama de clases: Caso de Uso Gestionar Usuario.

## CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### Diagrama de clases: Caso de Uso Emitir Reportes

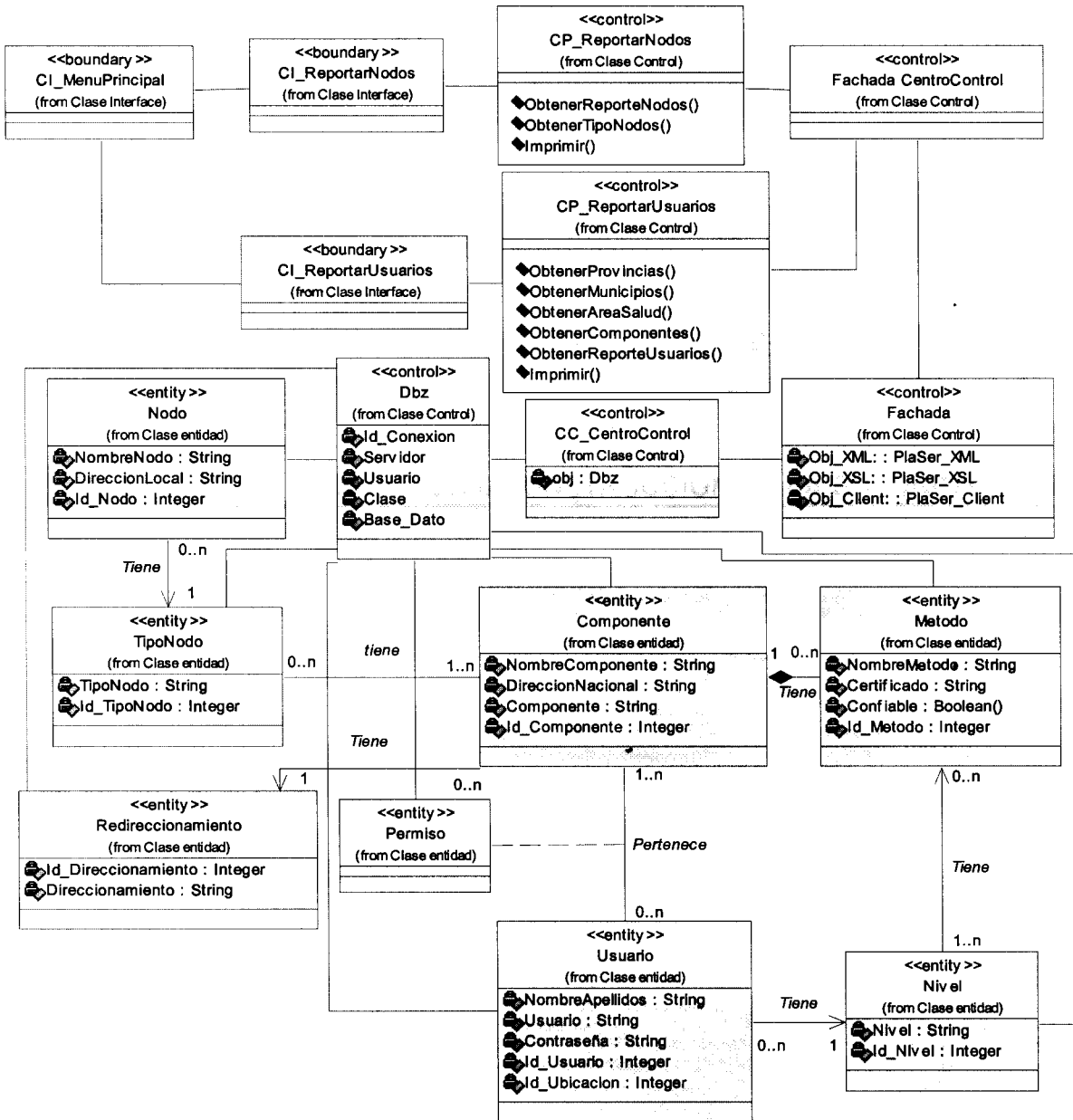


Figura 4.9 Diagrama de clases: Caso de Uso Emitir Reportes.

**4.3 Diseño de la base de Datos**

La base de datos del sistema se modeló partiendo del diagrama de clases persistentes. El diagrama de clases persistentes muestra los datos que no cambian. El diseño de la base de datos se basa en el diagrama de clases persistentes y el modelo de datos, los cuales se basan en los diagramas de clases expuestos en el epígrafe anterior. Los datos reflejados en las clases entidad son los que se capturan y almacenan durante el proceso que describe la aplicación, y estas a su vez son las que componen el diagrama de clases persistentes del cual se deriva el modelo de datos.

**4.3.1 Diagrama de clases persistentes**

**Diagrama de clases Persistentes**

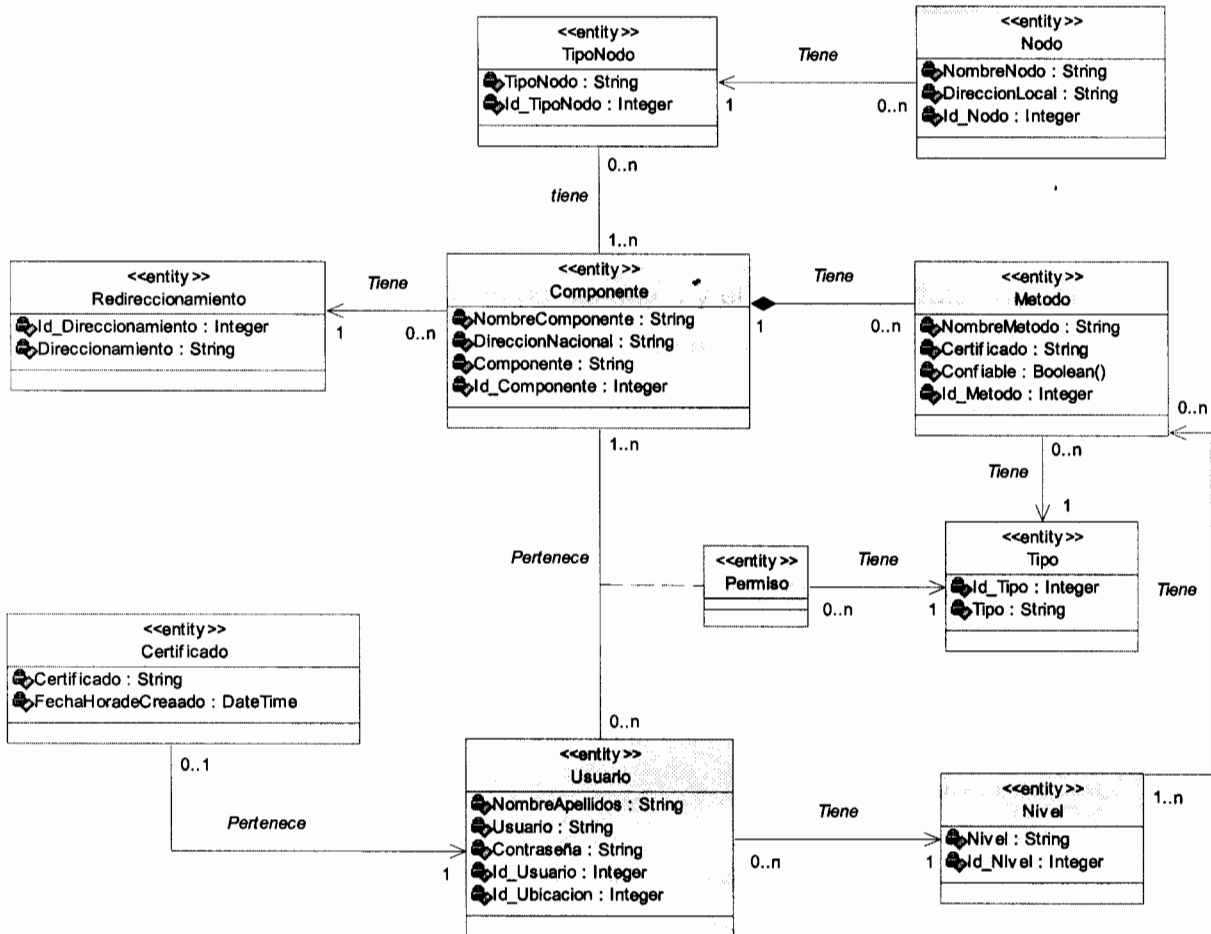


Figura 4.10 Diagrama de Clases Persistentes.



4.3.2 Modelo de Datos

Modelo de datos

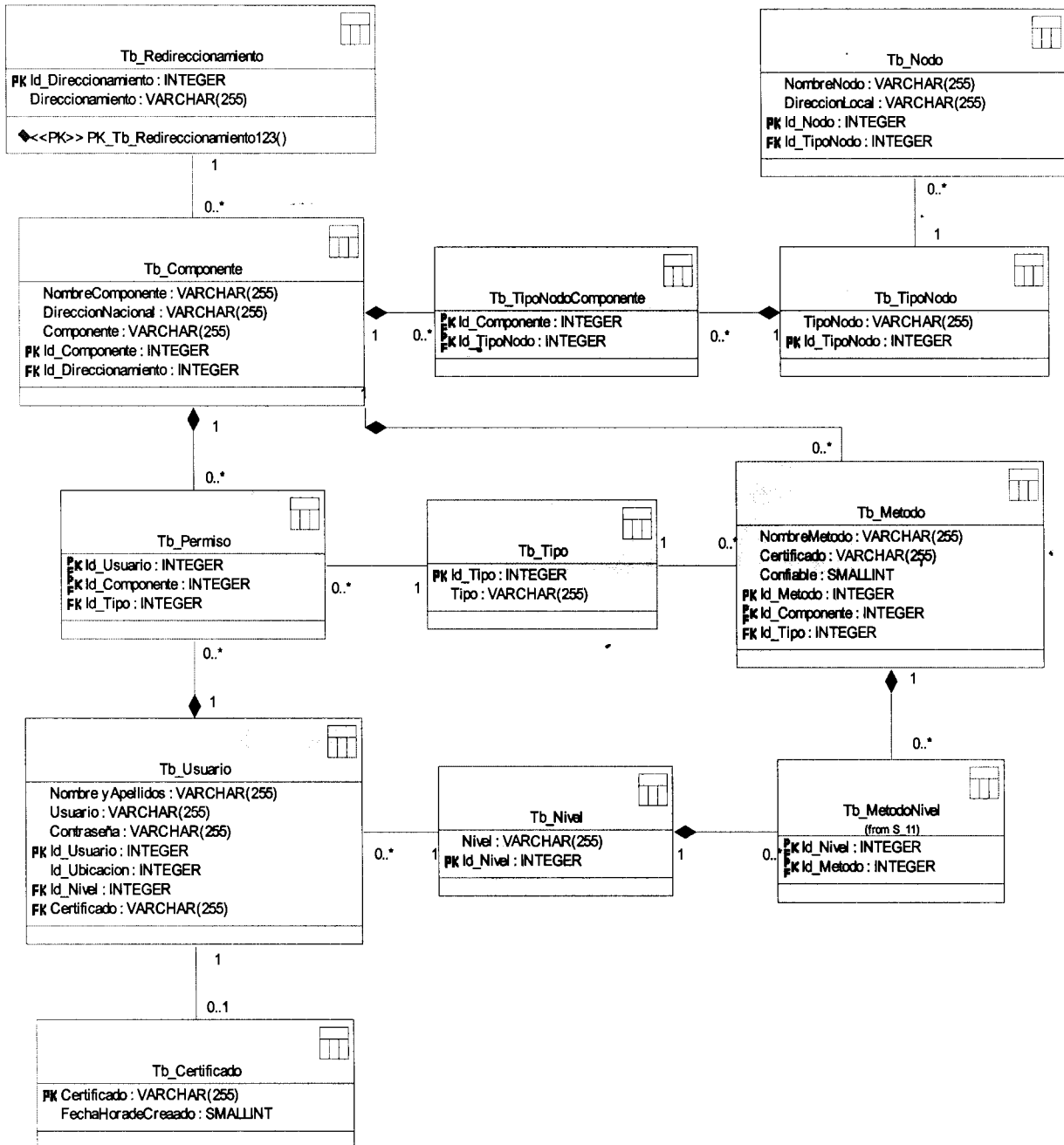


Figura 4.11 Modelo de Datos.

### **4.4 Principios de Diseño**

La aplicación está dirigida principalmente a profesionales de la salud que tienen un conocimiento mínimo, en muchos casos, de lo que es el trabajo en una PC, teniendo esto en cuenta y el innegable hecho de que los usuarios generalmente centran su atención en el ¿cómo usar la aplicación? y no en su estructura interna, los esfuerzos de diseño están orientados a lograr una interfaz clara y fácil de usar para los usuarios, ya que a través de ésta es que interactúan con los recursos que se ponen a su disposición en el ordenador. Para lograr lo que se ha planteado anteriormente, el diseño de la aplicación se basa en los 7 Principios del Diseño Universal o Diseño para Todos:

#### **Principio uso equiparable:**

El diseño es útil y vendible a personas con diversas capacidades.

- Que proporcione las mismas maneras de uso para todos los usuarios: idénticas cuando es posible, equivalentes cuando no lo es.
- Que evite segregar o estigmatizar a cualquier usuario.
- Las características de privacidad, garantía y seguridad deben estar igualmente disponibles para todos los usuarios.
- Que el diseño sea atractivo para todos los usuarios.

#### **Principio uso flexible:**

El diseño se acomoda a un amplio rango de preferencias y habilidades individuales.

- Que ofrezca posibilidades de elección en los métodos de uso.
- Que pueda accederse y usarse tanto con la mano derecha como con la izquierda.
- Que facilite al usuario la exactitud y precisión.
- Que se adapte al paso o ritmo del usuario.

#### **Principio simple e intuitivo:**

El uso del diseño es fácil de entender, atendiendo a la experiencia, conocimientos, habilidades lingüísticas o grado de concentración actual del usuario.

- Que elimine la complejidad innecesaria.
- Que sea consistente con las expectativas e intuición del usuario.
- Que se acomode a un amplio rango de alfabetización y habilidades lingüísticas.
- Que dispense la información de manera consistente con su importancia.

## **CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA**

---

- Que proporcione avisos eficaces y métodos de respuesta durante y tras la finalización de la tarea.

### **Principio Información perceptible:**

El diseño comunica de manera eficaz la información necesaria para el usuario, atendiendo a las condiciones ambientales o a las capacidades sensoriales del usuario.

- Que use diferentes modos para presentar de manera redundante la información esencial (gráfica, verbal o táctilmente).
- Que proporcione contraste suficiente entre la información esencial y sus alrededores.
- Que amplíe la legibilidad de la información esencial.
- Que diferencie los elementos en formas que puedan ser descritas (por ejemplo, que haga fácil dar instrucciones o direcciones).
- Que proporcione compatibilidad con varias técnicas o dispositivos usados por personas con limitaciones sensoriales.

### **Principio con tolerancia al error:**

El diseño minimiza los riesgos y las consecuencias adversas de acciones involuntarias o accidentales.

- Que disponga los elementos para minimizar los riesgos y errores: elementos más usados, más accesibles; y los elementos peligrosos eliminados, aislados o tapados.
- Que proporcione advertencias sobre peligros y errores.
- Que proporcione características seguras de interrupción.
- Que desaliente acciones inconscientes en tareas que requieren vigilancia.

### **Principio que exija poco esfuerzo físico:**

El diseño puede ser usado eficaz y confortablemente y con un mínimo de fatiga.

- Que permita que el usuario mantenga una posición corporal neutra.
- Que utilice de manera razonable las fuerzas necesarias para operar.
- Que minimice las acciones repetitivas.
- Que minimice el esfuerzo físico continuado.

### **Principio tamaño y espacio para el acceso y uso:**

Que proporcione un tamaño y espacio apropiados para el acceso, alcance, manipulación y uso, atendiendo al tamaño del cuerpo, la postura o la movilidad del usuario.

- Que proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie.
- Que el alcance de cualquier componente sea confortable para cualquier usuario sentado o de pie.
- Que se acomode a variaciones de tamaño de la mano o del agarre.
- Que proporcione el espacio necesario para el uso de ayudas técnicas o de asistencia personal.

#### **4.4.1 Estándares en la interfaz de la aplicación**

Los módulos que se desarrollan como parte del Proyecto de APS corresponden a diferentes sistemas que mantienen una profunda interrelación, permitiendo un acoplamiento entre ellos para ofrecer las respuestas que demanda el negocio de la Atención Primaria propuesto por el Sistema Nacional de Salud. Este hecho tiene una repercusión determinante en la definición de la interfaz gráfica que se propone.

Todos los módulos están incluidos en un conjunto de aplicaciones que forman parte del **Sistema Integral de Salud (SISalud)**, compuesto a su vez por el Registro Informatizado de Salud (RIS), el Sistema Informatizado de Atención Primaria (SIAP) y el Sistema Informatizado de Gestión Hospitalaria (SIGH).

En el presente documento nos proponemos definir las pautas generales de diseño a tener en cuenta para este grupo de aplicaciones, ya que todas serán desarrolladas bajo el marco del Proyecto APS.

#### **Diseño de Interfaz Gráfica del Proyecto APS**

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.

## **CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA**

---

Se diseñará una Pantalla Inicial global del **Sistema Integral de Salud**, desde la cual se accederá a los diferentes módulos del RIS, del SIAP y del SIGH. 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 Integral de 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 para cada uno, 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 800 x 600 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 superior horizontal de solo 15 px de altura. No existirá barra vertical de menú situada la izquierda de la página (como usualmente se hace) para ampliar el espacio de trabajo, pues estará reservado lo más amplio posible para la

## CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

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.

Bajo el logo existirá una barra de ubicación dentro del sitio, funcionando como hipervínculo, que servirá como referencia para saber donde se encuentra el usuario o para acceder rápidamente a cualquiera de los niveles superiores de navegación dentro de los que se encuentra. Además se encontrará destacado dentro del menú principal (con un destaque en el color secundario) en cual de los elementos del menú se encuentra el usuario en ese momento.

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. Son también comunes a casi todos los botones del menú principal *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.

## **CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA**

---

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 25 resultados la tabla funcione con scroll, pero más de esta cantidad será entonces por paginado, al estilo de *Google*, con 25 resultados por página.

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. (Ver Anexo XV)

### **4.4.2 Formatos de reportes**

Los reportes se obtendrán en tablas que en algunos casos pueden tener una gran cantidad de elementos en dependencia de la información a visualizar, por lo que debe quedar pautado que hasta un total de 25 resultados la tabla funcione con scroll, y para más de esta cantidad será entonces por paginado, organizado por números consecutivos, con enlaces a los resultados restantes, exceptuando el activo en ese preciso momento.

En algunos casos se hará uso de pequeñas imágenes que indicarán funcionalidades surgidas a partir de la visualización de estos reportes.

Los reportes serán concebidos sobre ventanas, utilizando un formato de letra clara, legible y con colores claros para no recargar y hacer engorrosa su impresión.

### **4.4.3 Concepción general de la ayuda**

Teniendo en cuenta que las ayudas en las aplicaciones Web en general no suelen ser explicaciones detallistas del sistema informático al cual representan, sino que son generalmente simples aclaraciones, informaciones generales de la aplicación o datos de la empresa que le da soporte o realizó el producto se está proponiendo que la ayuda estará accesible como parte del menú en todas las páginas de la aplicación, con el objetivo de que el usuario vea la información que necesita en ese momento. Cada página mostrará como realizar aquellas operaciones que estén relacionadas con la posición donde se encuentre el usuario en dicho momento, además se aportan los conceptos que se manejan en la aplicación, para que el usuario se familiarice con algunas entradas, el entorno de la elaboración de los informes o reportes y otras funcionalidades que se le brindan en el sistema.

La ayuda para todas las aplicaciones del Sistema Integral de Salud estará concebida bajo los principios del soporte técnico en línea, que es una práctica muy utilizada en las aplicaciones Web dinámicas, como la que se está desarrollando y generalmente se implementará con una explicación general de las opciones y con vínculos a sistemas de correo o a otros sitios Web.

Se podrá contar también con un soporte técnico en línea fuera de la aplicación principal para que los usuarios puedan informar acerca de errores que suceden en la aplicación, emitir sugerencias de su funcionalidad o recibir soluciones a las preguntas que de forma “directa” pueden realizar a los administradores y creadores del producto.

Esta forma de ayuda resulta de gran ventaja, ya que contribuye a la resolución de problemas en el software, la gestión de cambios y configuraciones y la actualización y el mantenimiento del producto.

Además se tendrá en cuenta la confección de manuales de usuario y será entregado a los usuarios de cada módulo un manual de usuario en formato digital o en papel, que explicará de forma detallada las principales funcionalidades y opciones que brinda el software.

También está concebida una capacitación técnica directa durante la etapa de implantación del producto, ya que por parte de la empresa productora y bajo acuerdo con los clientes se coordinará la puesta en marcha de cursos de capacitación o entrenamiento en el uso de la



aplicación, dirigida a todos los usuarios potenciales antes y durante la implantación oficial de este producto en el Sistema Nacional de Salud.

### **4.4.4 Tratamiento de excepciones**

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.

Las excepciones son condiciones excepcionales que pueden ocurrir dentro del programa durante su ejecución (por ejemplo una división por cero, que 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.

Los errores en la capa de negocio serán tratados devolviendo un SOAP\_FAULT, cuyos elementos FaultCode, FaultString, FaultActor describiremos a continuación:

#### **FaultCode:**

Código de texto utilizado para indicar la clase de error, codificado de la siguiente manera.

Código del proyecto-código del módulo (:) número del método (.) número del error. Ejemplo: APS-RF: 1.5 que indica error 5 en el método 1 del módulo Registro de Partos y Nacimientos perteneciente al Proyecto APS.

#### **FaultString:**

Una explicación del error asequible al humano (legible). Debe tenerse en cuenta que este texto puede ser mostrado al operador final del sistema. Ejemplo: Formato de entrada no válido para la fecha de cierre estadístico.

### **FaultActor:**

Un texto que indica quien provocó el error, siempre será el nombre del método que eleva la excepción. Ejemplo: realizarcierre.

### **Detail:**

Este elemento se usa para llevar mensajes de error específicos de aplicaciones, se empleará únicamente en errores cuya resolución depende del Centro de Control, en cualquier otro caso este elemento debe estar vacío.

Estos errores una vez en la capa de presentación, serán depurados mediante funciones del lenguaje *Client Side* Java Script, a través de mensajes de alerta.

### **4.5 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) [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 _ enfermedad`.

Los arreglos empezarán con el identificador array y las palabras no se separaran con el carácter “\_”. Ejemplo `Arrayidtipoenfermedad`.

Las estructuras se identificarán poniendo al final del nombre struct. Ejemplo paginadostruct.

### 4.6 Modelo de despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación que muestra las relaciones físicas entre los componentes *hardware* que forman la topología sobre la que se ejecuta el sistema y la distribución de las partes de este en ellos. (Ver Figura 4.3)

La división entre cliente y servidor en un sistema es complicada ya que implica tomar algunas decisiones sobre dónde colocar físicamente sus componentes software, qué cantidad de software debe residir en el cliente, etc. Por lo que para modelarlo hay que identificar los nodos que representan los procesadores cliente y servidor del sistema de acuerdo a las capas que se van a implementar, destacar los dispositivos relacionados con el comportamiento del sistema, proporcionar señales visuales para esos procesadores y dispositivos a través de estereotipos y modelar la tipología de esos nodos mediante un diagrama de despliegue. [14]

Aunque UML no es un lenguaje de especificación de hardware de propósito general, se ha diseñado para modelar muchos de los aspectos hardware de un sistema a un nivel suficiente para que un ingeniero software pueda especificar la plataforma sobre la que se ejecuta el software del sistema y para que un ingeniero de sistemas pueda manejar la frontera entre el hardware y el software.

## CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

Para lograr una mejor escalabilidad, se disponen para el proyecto de tres servidores, uno para la capa de presentación, otro para la capa de reglas del negocio y otro para la base de datos.

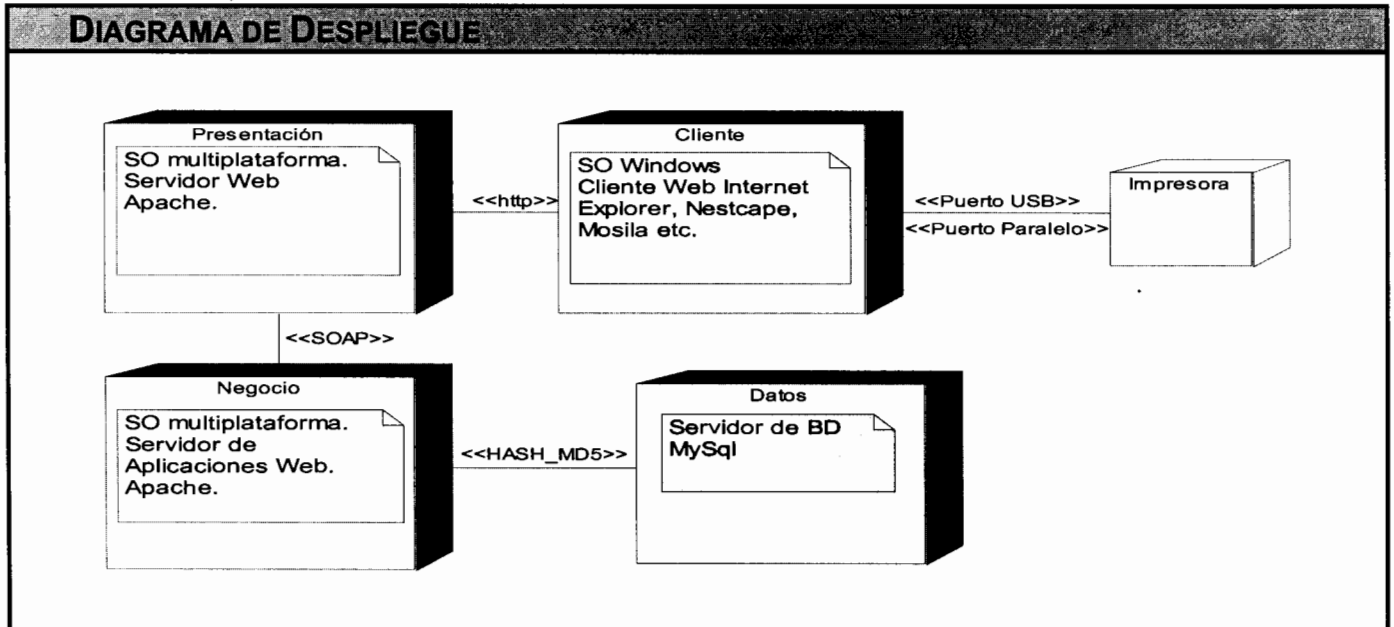


Figura 4.12 Diagrama de Despliegue.

### 4.7 Conclusiones

Se ha logrado modelar todo el sistema que ha sido objeto de estudio durante toda la investigación después de la realización de varias iteraciones a través de los flujos de trabajo que propone RUP. Los diagramas y especificaciones de diseño propuestos son fácilmente entendibles por cualquier programador que posea mínimos conocimientos de ingeniería de software, por lo que son de gran importancia para la futura implementación del sistema.

## **CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD**

### **5.1 Introducción**

El estudio de factibilidad es un paso importante que no se debe obviar en la realización de un proyecto, pues brinda al equipo de trabajo inicial información a priori relacionada con el costo del producto, tiempo estimado de desarrollo, cantidad de personas que intervienen, entre otros. Es decir, sirve para recopilar datos relevantes sobre el desarrollo de un proyecto y en base a ellos tomar la mejor decisión, si procede su estudio, desarrollo o implementación.

En el cálculo de estimación del esfuerzo se ha realizado basado en casos de uso, y usando COCOMO II directamente sobre los Puntos de Función sin ajustar. Este método es el preferido en la actualidad para la estimación del esfuerzo cuando no se tiene información histórica a la cual recurrir, y es el caso que se nos plantea.

#### **Ventajas:**

- Se utilizan unidades físicas tales como las líneas de código para medir la magnitud del producto.
- Los resultados obtenidos son proporcionales a las tareas técnicas.
- Fácil de realizar ya que las unidades de medida son tangibles.
- Es flexible y permite su uso a cualquier tipo de proyecto.

### **5.2 Planificación**

La estimación del proyecto se realizó mediante los puntos de función desajustados, los cuales se utilizan para el cálculo de las instrucciones fuentes. De esta forma se estima la magnitud del sistema y se obtienen además indicadores como la cantidad de hombres, el esfuerzo, el tiempo de duración y el costo del mismo.

#### **5.2.1 Características del proyecto**

Para la realización de este proyecto se tenía sólo 2 meses de experiencia con el lenguaje PHP con noción del gestor de base de datos MySQL y de la filosofía de trabajo de los Servicios Web. Producto de esto se requirió tiempo de investigación para poder usar estas tecnologías. En las tablas que se muestran a continuación aparecen listadas las posibles pantallas de entradas y

## CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

salidas externas, así como las Peticiones, Ficheros internos e Interfaz externa. Todas ellas se clasifican por su nivel de complejidad en: Simple, Media, Compleja.

Entradas Externas	Ficheros	Elementos de datos	Clasificación
Autenticarse	1	2	Simple
Inserta Nodo	2	3	Simple
Modificar Nodo	2	3	Simple
Insertar Componente	1	4	Simple
Modificar Componente	1	4	Simple
Insertar Método	1	5	Simple
Modificar Método	1	5	Simple
Insertar Usuario	2	6	Simple
Modificar Usuario	2	6	Simple
Insertar Tiponodo	2	2	Simple
Modificar TipoNodo	2	2	Simple
Insertar Nivel	1	1	Simple
Modificar Nivel	1	1	Simple
Modificar Contraseña	1	1	Simple
Eliminar Nodo	1	4	Simple
Eliminar Componente	2	6	Simple
Eliminar Método	1	6	Simple
Eliminar Usuario	2	7	Simple
Eliminar TipoNodo	1	2	Simple
Eliminar Nivel	1	2	Simple
<b>TOTAL</b>	<b>Simple: 20 Media: 0 Compleja: 0</b>		

Tabla 5.1 Entradas Externas.

Salidas Externas	Ficheros	Elementos de datos	Clasificación
Listar Componentes	2	4	Simple
Listar Nivel	1	1	Simple
Listar TipoNodo	1	1	Simple
<b>TOTAL</b>	<b>Simple: 3 Media: 0 Compleja: 0</b>		

Tabla 5.21 Salidas Externas.

## CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

Peticiones	Ficheros	Elementos de Datos	Clasificación
Buscar Usuario	4	13	Complejo
Buscar Nodo	2	4	Simple
Buscar Método	3	8	Medio
<b>TOTAL</b>	<b>Simple: 1 Media: 1 Compleja: 1</b>		

Tabla 5.3 Consultas Externas.

Fichero interno	Records	Elementos de datos	Clasificación
Nodo	1	3	Simple
TipoNodo	1	2	Simple
Componente	1	5	Simple
Método	3	6	Simple
Nivel	1	2	Simple
Usuario	4	7	Simple
<b>TOTAL</b>	<b>Simple: 6 Media: 0 Compleja: 0</b>		

Tabla 5.4 Archivos Lógicos Internos.

Interfase externa	Records	Elementos de datos	Clasificación
Mostrar nombre de provincia	1	1	Simple
Mostrar nombre de municipio	2	2	Simple
Mostrar área de salud	1	3	Simple
<b>TOTAL</b>	<b>Simple: 3 Media: 0 Compleja: 0</b>		

Tabla 5.5 Interfaces Externas.

Elementos	Simples		Medios		Complejos		Subtotal de puntos de función
	No.	X Peso	No.	X Peso	No.	X Peso	
Ficheros lógicos internos	6	(*7)	0	(*10)	0	(*15)	42
Ficheros de interfaz externa	3	(*5)	0	(*7)	0	(*10)	15
Entradas externas	20	(*3)	0	(*4)	0	(*6)	60
Salidas externas	3	(*4)	0	(*5)	0	(7)	12
Peticiones	1	(*3)	1	(*4)	1	(6)	13
<b>Total</b>							<b>142</b>

Tabla 5.6 Puntos de Función sin ajustar.

**Instrucciones fuentes:**

COCOMO II, plantea que su cálculo se basa en la cantidad de instrucciones fuentes por punto de función desajustado que genera el lenguaje de programación empleado. Estos datos se conocen a partir de estudios estadísticos realizados a cada lenguaje.

<b>Características</b>	<b>Valor</b>		
Puntos de función desajustados	142		
Lenguaje	PHP	JavaScript	SQL
% de utilización en la aplicación	85% (≈121)	10% (≈14)	5% (≈71)
Instrucciones fuentes por puntos de función	60	56	39
Instrucciones fuentes	7260	784	2769
Total Instrucciones fuentes	10813		

Tabla 5.7 Instrucciones fuentes.

**5.3 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo**

**Multiplicadores de esfuerzo:**

<b>Multiplicador</b>	<b>Descripción</b>	<b>Valor</b>
Complejidad y Confianza del Producto (RCPX)	La complejidad del producto es media.	1
Reutilización Requerida (RUSE)	Se implementa código reutilizable para su aprovechamiento en el proyecto.	1
Inconvenientes de la Plataforma (PDIF)	La plataforma es estable. Requerimientos bajos de almacenamiento y tiempo de ejecución.	0.87
Capacidad Personal (PERS)	La capacidad de los especialistas (analistas-programadores) es alta. La continuidad del personal es alta.	0.83
Experiencia Personal (PREX)	El equipo tiene poco dominio y conocimiento del lenguaje de programación, plataforma y herramientas de desarrollo utilizados. No ha	1.33



## CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

	desarrollado aplicaciones similares, casi ninguna experiencia.	
Facilidades (FCIL)	Se utilizan herramientas e instrumentos de programación modernos.	1
Planificación (SCED)	Los requerimientos de calendario de desarrollo son bajos.	1.14
$EM = \prod_{i=1}^7 E_{mi}$		<b>1.09</b>

Tabla 5.8 Definición de los Multiplicadores de Esfuerzo (MEj).

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED = 1.09$$

**Factores de Escala:**

Factor	Descripción	Valor
PREC	Bastante parecido	2.48
FLEX	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa.	3.04
TEAM	Interacciones principalmente cooperativas. Mediana experiencia previa operando	2.19
RESL	La arquitectura es sólida y los riesgos generalmente se mitigan. Poca incertidumbre, riesgos no son críticos.	1.41
PMAT	Relación con el proceso de madurez del software. Nivel 3.	3.12
$\sum_{i=1}^5 SF_i$		12.24

Tabla 5.9 Definición de los valores de los Factores de Escala (SF<sub>i</sub>).

$$SF = \sum_{i=1}^5 SF_i = PREC + FLEX + RESL + TEAM + PMAT = 12.24$$

**Valores calibrados:**

A=2.94; B=0.91; C=3.67; D=0.24

$$E = B + 0.01 * \Sigma SFi = 0.91 + 0.01 * 12.24 = 1.03$$

$$F = D + 0.2 * (E - B) = 0.24 + 0.2 * (1.03 - 0.91) = 0.26$$

**Cálculo del esfuerzo (PM):**

$$PM = A * (MSLOC)^E * \Pi Emi = 2.94 * (10,813)^{1.03} * 1.09 = 37 \text{ Hombres/Mes.}$$

➤ **Cálculo del tiempo de desarrollo:**

$$TDEV = C * PM^F = 3.67 * (37)^{0.26} = 9,38 \approx 9 \text{ meses (Estimado)}$$

➤ **Cálculo de la cantidad de hombres:**

$$CH = PM / TDEV = 37 / 9 = 4 \text{ hombres}$$

Como el equipo de trabajo está formado realmente por 2 personas, se recalculó el tiempo de desarrollo para la cantidad real de hombres.

$$CH^* = 2 \text{ hombres.}$$

$$TEDV = PM / CH^* = 37 / 2 = 18 \text{ meses.}$$

➤ **Cálculo del costo:**

Asumiendo como salario promedio mensual (SP) \$225.00

$$CHM = CH * SP = 2 * \$225.00 = \$ 450.00$$

$$\text{Costo} = CHM * PM = \$450 * 37 = \$ 16650.00$$

**Cálculos:**

<b>Cálculo de:</b>	<b>Valor</b>
Esfuerzo (PM: Hombres - mes)	37 Hombres/Mes
Tiempo de Desarrollo(meses)	18 Meses
Cantidad de Hombres	2
Costo	\$ 16650.00
Salario medio	\$ 225.00

Tabla 5.10 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

### **5.4 Beneficios tangibles e intangibles**

El desarrollo del módulo Centro de Control traerá consigo numerosos beneficios tangibles e intangibles:

#### **Beneficios tangibles**

En paralelo a la etapa de modelado y de implementación de los módulos de la etapa actual del Proyecto APS, entre los que se encuentra el componente Centro de Control para el Sistema Integral de Salud, se establecen las fórmulas comerciales necesarias para lograr la introducción estable e incremental de estos productos y servicios asociados en el mercado internacional, con especial énfasis en el contexto latinoamericano, de forma tal que los beneficios tangibles esperados se materialicen y así lograr que los servicios médicos se conviertan en la principal fuente de ingreso del país.

#### **Beneficios intangibles**

Garantiza la máxima disponibilidad de cada uno de sus componentes que se encuentran distribuidos en todo el Sistema Nacional de Salud. Este sistema tiene la posibilidad de recuperarse ante posibles fallos de conectividad o resolver problemas tales como recuperar la información independientemente de su ubicación.

Además este es el encargado de tomar decisiones referente a cómo enrutar las peticiones hechas a los componentes de negocio para mantener la disponibilidad de la información distribuida que se maneja y proteger el acceso a estos en función de los derechos de cada usuario, lo que permite lograr una mayor fiabilidad e integridad de los distintos componentes.

### **5.5 Análisis de costos y beneficios**

Al desarrollo de todo producto informático va asociado un costo, el justificarlo depende de los beneficios tangibles e intangibles que produce.

La utilización de un nuevo componente de arquitectura para enrutar las peticiones de los diferentes componentes del sistema, parte de la idea de que los componentes en la actualidad se presentan aislados en diferentes nodos del país, y no existía un mecanismo único de

integración de los sistemas de información desarrollados, lo cual trae consigo la duplicación de información y la consiguiente falta de integridad de la misma.

Con la información brindada por el sistema se puede definir a tiempo las actividades retrazadas en el cumplimiento de lo planificado y de esta manera tomar las medidas pertinentes con mayor rapidez y seguridad.

Pero si es importante el costo del proyecto, más importante son los beneficios que de él se derivan. La informatización del Sistema Integral de Salud es hoy una tarea de máxima prioridad. El desarrollo de esta investigación tiene gran importancia, ya que permite la fiabilidad, actualización y la rapidez de toda la información que se maneja, por tal motivo es imprescindible la implementación del mismo.

### **5.6 Conclusiones**

Una vez terminado el estudio de factibilidad del sistema, se estima un tiempo de 18 meses para su construcción por 2 hombres y su costo asciende a \$16 650.00.

La herramienta propuesta trae consigo una serie de beneficios tangibles e intangibles, ya que la misma va a contribuir a la integridad de los módulos distribuidos en el Sistema de Salud en Cuba y puede ser instalado en cualquier sistema de Salud, lo que indica que es factible implementar la herramienta propuesta.

## **CONCLUSIONES**

Con la presente investigación se concluye que el Componente Centro de Control para el Sistema Integral de Salud cumplirá un papel fundamental en el Proceso General de Informatización del Sistema Nacional de Salud porque su implementación e implantación permitirá la gestión de los usuarios y la configuración de los demás componentes, para garantizar la seguridad e integridad de la información y proteger el acceso a éstos en función de los derechos de cada usuario. Su funcionalidad no se enmarca solamente al nivel de la Atención Primaria, sino al Sistema Integral de Salud en su totalidad.

De esta forma se cumplieron los objetivos generales y específicos planteados al comenzar el estudio:

- Se logró concebir, modelar y especificar un componente denominado Centro de Control que da cumplimiento a los requisitos funcionales planteados inicialmente.
- Se aplicaron en la modelación del sistema patrones de arquitectura sólidos como el **Modelo Vista Controlador** y el patrón de seguridad **Single Sign On** en función de resolver la configuración de la distribución de los nodos para facilitar el enrutamiento de las peticiones hechas a los componentes de negocio de un sistema distribuido y controlar dichas peticiones en correspondencia con los derechos de los usuarios que realizan dichas las peticiones.
- Se realizó un estudio de las herramientas informáticas a utilizar para la futura implementación de la solución propuesta.

Con la culminación de las etapa de análisis y diseño de desarrollo de software se demuestra y el desarrollo de *software* se están convirtiendo en un aspecto fundamental de las estrategias cubanas para el avance socioeconómico se está convirtiendo en un producto muy importante en una sociedad con una población altamente entrenada, educada y capacitada.

### RECOMENDACIONES

Una vez vencidos los objetivos de este proyecto, y teniendo en cuenta las experiencias obtenidas a lo largo de la investigación se recomienda a la Empresa SOFTEL:

- Pasar a la fase de construcción, pues se cuenta con las especificaciones necesarias para comenzar a materializar la implementación del componente.
- Generalizar el campo de aplicación del componente Centro de Control a otros proyectos que sigan una arquitectura Orientada a Servicios y Basada en Componentes.
- Crear un registro en el sistema que permita conocer todas las acciones que realiza el usuario dentro de la aplicación.
- Permitir que este componente se integre con un componente de avisos, actualmente en desarrollo, para brindar la posibilidad a los administradores de enviar información a los grupos de usuarios que utilizan el sistema.
- Validar que el administrador principal sea un personal de la salud, ya que actualmente no es un requisito para administrar el sistema, por lo que debe verificarse en el Registro de Personal de Salud (RP).
- Modelar en una nueva iteración la funcionalidad de enrutamiento de peticiones ya que están creadas las condiciones con la configuración actual concebida en este trabajo.

## REFERENCIAS BIBLIOGRÁFICAS

- [1]. *Consideraciones sobre el Proyecto de Informatización de la Atención Primaria de Salud.* Revista Habanera de Ciencias Médicas. Volumen 3, No. 10, año 2004.
- [2]. Granma, *Discurso pronunciado por el Presidente de la República de Cuba, Fidel Castro Ruz, en el acto conmemorativo del aniversario 40 del Instituto de Ciencias Básicas y Preclínicas Victoria de Girón, 17 de octubre de 2002*
- [3]. De la Osa ,José A. *La semilla del desarrollo de la salud pública en Cuba.*
- [4]. Escuela Nacional de Salud Pública, *El cuidado de la salud en Cuba. Ministerio de Salud Pública, 2003.*
- [5]. Granma, *Discurso pronunciado por el Presidente de la República de Cuba, Fidel Castro Ruz, 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]. Granma, *Discurso pronunciado en la Clausura del VI Seminario Internacional de Atención Primaria, Ciudad de la Habana, 28 de noviembre de 1997.*
- [7]. Lemus, Elia Rosa,*Atención Primaria de Salud y Medicina Familiar.I,* Eugenio Radamés Borroto Cruz; Ramón Aneiros-Riba. Atención Primaria de Salud, Medicina Familiar y Educación Médica, Biblioteca de Medicina Volumen XXXIV, La Paz, 1998.
- [8]. Ramírez Márquez, Abelardo *El Sistema Nacional de Salud de Cuba. /* Pastor Castell-Florit Serrate; Guillermo Mesa. ENSAP, 2003.
- [9]. Ramírez Márquez, Abelardo; Castell-Florit Serrate, Pastor; Mesa, Guillermo. *El Sistema Nacional de Salud de Cuba..* ENSAP, 2003.
- [10]. Ramirez, Enrique J; Bajo, Maximiliano. *La Red de Redes.*  
<http://www.monografias.com/trabajos11/infintern/infintern.shtml>. (13/5/2005).
- [11]. Adell, Jordi; Bellver, Carles. *La Internet como Telaraña: World Wide Web.*  
<http://www.uv.es/~biblios/mei3/Web022.html> (18/5/2005)
- [12]. Barreira Rodríguez, Noelia *Introducción a Servicios Web.* Septiembre, 2002
- [13]. *Computación distribuida Servicios web .Traducción.* Abril, 2004.  
<http://web-services.bankhacker.com>. (20/5/2005).
- [14]. Piker, M, José. *DNS.* <http://www.dcc.uchile.cl/~jpiquer/Internet/DNS/node1.html>.  
(20/5/2005)
- [15]. Alea, Salazar, Caridad; Rodríguez, Cortina, Antonio. *Diseñando Aplicaciones Distribuidas.*  
<http://www.monografias.com/trabajos14/aplicacion-distrib/> . (1/6/2005).

- [16]. Parra, José, David. *Hacia una arquitectura empresarial basada en componentes*. <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art143.asp> (2/6/2005).
- [17]. El servidor Web Apache. <http://es.tldp.org/Tutoriales/doc-servir-web-escuela/doc-servir-web-escuela-html/apache.html> (3/6/2005).
- [18]. Middleware. <http://www.gestiopolis.com/delta/term/TER184.html>. (3/6/2005).
- [19]. Modelo Vista Controlador. [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador). (4/6/2005).
- [20] Suplemento.W2000MAG. *Single Sign On*. [http://www.windowstimag.com/atrasados/2000/40\\_mar00/articulos/suplemento/singlesign2.htm](http://www.windowstimag.com/atrasados/2000/40_mar00/articulos/suplemento/singlesign2.htm). (6/6/2005)
- [21]. Álvarez, Miguel Ángel. *Introducción a los lenguajes del Web*. <http://www.desarrolloweb.com>. (7/6/2005).
- [22]. Moreno, Luciano. *Curso práctico de ASP. Conceptos básicos*. [http://www.htmlweb.net/asp/curso\\_webmaster/cap\\_1/cap\\_1\\_2.html](http://www.htmlweb.net/asp/curso_webmaster/cap_1/cap_1_2.html) . (7/6/2005). ).
- [23]. *Coding Standard for PHP*. 2003. [http://alltasks.net/code/php\\_coding\\_standard.html#important](http://alltasks.net/code/php_coding_standard.html#important) (7/6/2005).
- [24]. XSLT. <http://es.wikipedia.org/wiki/XSLT>. (7/6/2005)
- [25]. Trejo Martínez, Janhil. *Bases de datos*. <http://www.monografias.com> (7/6/2004).
- [26]. Aguilar, Vicente y Suau, Pablo. *MySQL vs. PostgreSQL (18 de Agosto de 2000)*. <http://www.mmlabx.ua.es/mysql-postgres.html> (9/06/2004).
- [27]. Trejo Martínez, Janhil. *Bases de datos*. <http://www.monografias.com> (7/6/2004).
- [28]. Manuales de SQL Server. [http://www.cgrsoftware.com/manuales\\_dir.htm?sqlserver\\_1](http://www.cgrsoftware.com/manuales_dir.htm?sqlserver_1) (07-03-2004).
- [29]. Trejo Martínez, Janhil. *Bases de datos*. <http://www.monografias.com> (7/6/2004).
- [30]. Larman Craig. *“UML y patrones”* 1999, Prentice Hall Iberoamericana.



**BIBLIOGRAFÍA**

COCOMO II. 23 septiembre 2002.

[http://sunset.usc.edu/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/research/COCOMOII/cocomo_main.html) (5/6/2005)

De la Fuente Moya, Antonio. *Cocomo v2, "Modelo de estimación de costes para proyectos software"*. Universidad de Castilla-La Mancha, 1999.

Jacobson, I.; Booch, G. y Rumbaugh, J.. *"El Proceso Unificado de Desarrollo de software"*. 2000. Addison-Wesley.

Lago, Ramiro. *Patrón Modelo-Vista-Controlador* .

<http://www.proactiva-calidad.com/java/patrones/mvc.html>. (5/6/2005).

Metodología MÉTRICA Versión 3. "Técnicas y Practicas utilizadas en los procesos principales y en el proceso de Gestión de *Proyectos*".

Pressman, R. *"Software Engineering. A Practitioner's Approach"*. Fourth Edition. McGraw – Hill. USA, 1999.

Rodríguez, Gutiérrez. *"Arquitectura de Aplicaciones de tres capas"*.

<http://dotnetjunkies.com/WebLog/desarrollonet/archive/2004/06/17/16855.aspx> (7/6/2005)

RUMBAUGH J, Jacobson I., Booch G. El Lenguaje Unificado de Modelado. Manual de Referencia. 2000 Addison Wesley.

Tutorial de PHP y MySQL. [http://es.tldp.org/Manuales-LuCAS/manual\\_PHP/manual\\_PHP/](http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP/)

Martínez, Rafael. *"Manual de PHP"*. <http://www.php.net/manual/es/print/index.php> (9/6/2005)

## GLOSARIO DE TÉRMINOS

**AAA:** Abreviatura de Autenticación, Autorización y Auditoría. Cada uno de los tres pasos fundamentales en la seguridad de las redes de datos

**Acoplamiento:** Es una medida de la interdependencia relativa entre los componentes. Minimizando el acoplamiento se evita el efecto “onda” en la propagación de errores.

**Actor:** Personas o sistemas externos que interactúan con la aplicación

**Apache:** Es el servidor Web, disponible sin costo alguno, distribuido con licencia Open Source. Es compatible con sistemas operativos basados en Unix y con Windows.

**Benchmark:** Es un programa que mide las prestaciones de un ordenador, o de una parte del mismo. Son pruebas para medir el rendimiento y poder verificar que el hardware funciona de forma óptima o para comparar distintas configuraciones.

**Booch:** Método para el análisis y diseño orientado a objetos.

**COCOMO:** Abreviaturas correspondientes a *Constructive Cost Model*. Modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo de software.

**Cohesión:** Es una medida de la fuerza relativa funcional de un componente.

Un componente con cohesión realiza una sola tarea dentro de un procedimiento de software, requiriendo poca interacción con los otros componentes.

**Contraseña:** Palabra o cadena de caracteres, normalmente secreta, para acceder a través de una barrera. Se usa como herramienta de seguridad para identificar usuarios de una aplicación, archivo, o red. Puede tener la forma de una palabra o frase de carácter alfanumérico, y se usa para prevenir accesos no autorizados a información confidencial.

**COM:** Abreviaturas de *Component Object Model*. Un modelo de código binario desarrollado por Microsoft que permite a los programadores desarrollar objetos al los que se puede acceder desde cualquier aplicación de componentes COM.

**CORBA:** Abreviaturas correspondientes a *Common Object Request Broker Architecture*. Es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

**Despliegue:** Representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

**Dreamweaver MX:** Software fácil de usar que permite crear páginas web profesionales.

**Drivers (controladores):** Son los encargados de actuar como interfaz entre el sistema operativo y los dispositivos que componen un ordenador. De esta forma todos los componentes del PC se entienden y trabajan conjuntamente.

**DTD:** Es una declaración en un documento SGML ó XML que especifica restricciones en la estructura del mismo. El DTD puede ser incluido dentro del archivo del documento, pero normalmente se almacena en un fichero ASCII de texto separado. La sintaxis de los DTDs para SGML y XML es similar pero no idéntica.

**Encapsulación:** En programación orientada a objetos, se denomina encapsulación al ocultamiento del estado, es decir, de los datos miembro, de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto. De esta forma el usuario de la clase puede obviar la implementación de los métodos y propiedades para concentrarse sólo en cómo usarlos. Por otro lado se evita que el usuario pueda cambiar su estado de maneras imprevistas e incontroladas.

**Hipertexto:** Es una tecnología que organiza una base de información en bloques distintos de contenidos, conectados a través de una serie de enlaces cuya activación o selección provoca la recuperación de información [Díaz et al, 1996].

**Hipermedia:** Revela que la estructura subyacente está formada por hipervínculos. Esta inclusión, o expansión de las funciones de gestión de información a cualquier forma que pueda presentar la misma, manteniendo como criterio la asociación o relación de conceptos, es lo que caracteriza a un sistema hipermedia.

**HTML:** Abreviatura correspondiente a *Hypertext Markup Language*. Lenguaje utilizado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986.

**InterBase:** Base de Datos que presenta capacidades que permiten que estén varias decenas de usuarios conectados concurrentemente haciendo uso de una misma base de datos Internase. Presenta capacidades que compiten directamente con otras bases de datos de gran capacidad.

**Interfaz:** Término aplicado a la representación gráfica.

**LAN:** Es un acrónimo inglés de *Local Area Network* (Red de Área Local), y que se refiere a las redes locales de ordenadores.

**Linux:** Versión Shareware del conocido sistema operativo Unix. Es un sistema multitarea multiusuario de 32 bits para PC. Se ha desarrollado de forma gratuita por miles de

programadores a lo largo del mundo y está en continua evolución. Recibe su nombre del de el creador del núcleo del sistema; Linus Tolvards.

**MIC:** Abreviatura correspondientes a Empresa del Ministerio de la Informática y las Comunicaciones que utiliza una estrategia nunca antes concebida en un proceso de desarrollo de software en el país, con una organización del proceso productivo a través de una eficiente gestión de requerimientos.

**Microsoft:** Acrónimo de Microcomputer Software Es la industria líder entre todas las compañías de tecnologías, al frente su presidente Bill Gates. Es el desarrollador más poderoso en los software y aplicaciones para computadoras personales y entre sus principales productos está los sistemas operativos Windows.

**MINSAP:** Abreviatura correspondiente a Ministerio de Salud Pública.

**Multiplataforma:** Es un término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas. Por ejemplo la posibilidad de utilizar un programa o software determinado en sistemas Windows y Linux.

**MVC:** Abreviatura correspondiente a Modelo Vista Controlador. Patrón de diseño de aplicaciones con sofisticados interfaz.

**MyODBC:** Conector/ODBC que permite conectar un servidor de base de datos MySQL usando la API de Microsoft Windows y la mayoría de las plataformas Unix, incluyendo aplicaciones y entorno de programación como Microsoft Access, Microsoft Excel. y Borland Delphi.

**Multiusuario:** de *multi*: varios; y *usuarios*, "apto para ser utilizado por muchos usuarios". Dicho sobre un sistema operativo, significa que puede estar ocupado por varios usuarios al mismo tiempo, lo cual permite reducir los tiempos ociosos en el procesador, e indirectamente la reducción de los costos de transmisión, energía y equipamiento para resolver las necesidades de cómputo de los usuarios.

**NTIC:** Abreviatura correspondientes a las Nuevas Tecnologías de la Comunicación y las Telecomunicaciones.

**ODBC:** Es el acrónimo de Open Database Connectivity. Es un programa de interfaz de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacional, utilizando para ello SQL (lenguaje de consulta estructurado).

**OMT:** Abreviaturas correspondientes a Object Modeling Technique. Es una de las metodologías de análisis y diseño orientadas a objetos

**Oracle:** Es una herramienta cliente / servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales.

**phpMyAdmin:** Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar llaves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 47 idiomas.

**Protocolo:** Especificación que describe las reglas y procedimientos que deben cumplir los productos para la realización de actividades en una red, tales como la transmisión de datos. Si se utilizan los mismos protocolos, los productos de diferentes vendedores pueden comunicarse por la misma red.

**Red:** Grupo de ordenadores y otros dispositivos periféricos conectados unos a otros para comunicarse y transmitir datos entre ellos.

**Script:** Archivo de texto que contiene una secuencia de órdenes interpretadas por el sistema, o una aplicación, que es capaz de automatizar una tarea cuando se ejecuta. También se le conoce como guión.

**Servicio Web:** Es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet.

**Sistemas distribuidos:** Un sistema en el que los componentes hardware y/o software ubicados en computadores en red, se comunican y coordinan sus acciones intercambiando mensajes. Colección de ordenadores *autónomos* enlazados por una red y soportados por aplicaciones que hacen que la colección actúe como un servicio integrado.

**SSL:** Abreviaturas correspondientes a *Secure Sockets Layer*. Es un protocolo diseñado por la empresa Netscape Communications, que permite cifrar la conexión, incluso garantiza la autenticación. Se basa en la criptografía asimétrica y en el concepto de los certificados.

**Subconsultas:** Es una sentencia SELECT que aparece dentro de otra sentencia SELECT llamada consulta principal. Se puede encontrar en la lista de selección, en la cláusula WHERE o en la cláusula HAVING de la consulta principal.

**TIC:** Abreviaturas correspondientes a Tecnologías de la Información y de las Comunicaciones.

**SyBase:** Base de Datos que ofrece a las empresas la infraestructura completa necesaria para lograr integrar sus distintas tecnologías. Gracias a su innovador portal empresarial (Enterprise Portal, EP), sus soluciones móviles y sin cables, sus sistemas de gestión de bases de datos de alto rendimiento y sus productos de integración pioneros. Sybase es uno de los proveedores de software independientes más importantes del mundo.

**SNS:** Abreviaturas correspondiente a Sistema Nacional de Salud.

**SOAP:** Son las siglas de Simple Object Access Protocol. Este protocolo deriva de un protocolo creado por David Winer, XML-RPC en 1998. En el núcleo de los servicios Web se encuentra el protocolo simple de acceso a datos SOAP, que proporciona un mecanismo estándar de empaquetar mensajes.

**Software:** Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de maquina.

**SSO:** Abreviatura correspondiente a Single Sign On. Patrón de seguridad para la autenticación de contraseñas.

**TCP/IP:** Se refiere a una suite de protocolos de datos. El nombre viene de 2 de los protocolos que lo conforman: Transmission Control Protocol (TCP) y Internet Protocol (IP).

**UNIX:** Un sistema operativo multiusuario desarrollado por los Laboratorios Bell, escrito en el lenguaje de programación C, muy portable. Es el principal sistema operativo usado por los proveedores de Internet.

**UML:** Abreviatura correspondiente a *Unified Modeling Language*. Es una notación standard para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos.

**URL:** Abreviatura de *Uniform Resource Locator*. Es el sistema de direcciones usado en la Web y otros recursos de Internet. El URL contiene información sobre el método de acceso, el servidor al que se accede y la dirección o el fichero.

**Usuario:** Cualquiera que requiere los servicios de un sistema computacional.

**Usuario autenticado:** Es aquel usuario que ha proporcionado información mediante la cual el mecanismo de seguridad garantiza su identificación al intentar acceder a los componentes del sistema, los mecanismos de autenticación pueden ser tan simples como una contraseña o tan complejos como un dispositivo analizador de patrones retinales.

**Virtual Host:** Se refiere a la práctica de mantener más de un servidor en una sola máquina, así como diferenciarlos por el nombre de servidor que presentan.

**WAN:** Es un acrónimo de *Wide Area Network* que en inglés significa (red de área amplia). Un ejemplo de este tipo de redes sería rediris, la misma Internet o cualquier red en que no esté en un mismo edificio todos sus miembros (sobre la distancia hay discusión posible). Opera en la capa física y de enlace del modelo de referencia OSI.

**WSDL:** Abreviatura correspondiente a *Web Services Description Languages*. Define dónde está disponible el servicio y qué protocolo de comunicaciones utilizar para hablar con el servicio. El archivo WSDL define todo lo necesario para escribir un programa que interactúe con un Servicio Web.

DNS (*Domain Name Service*).

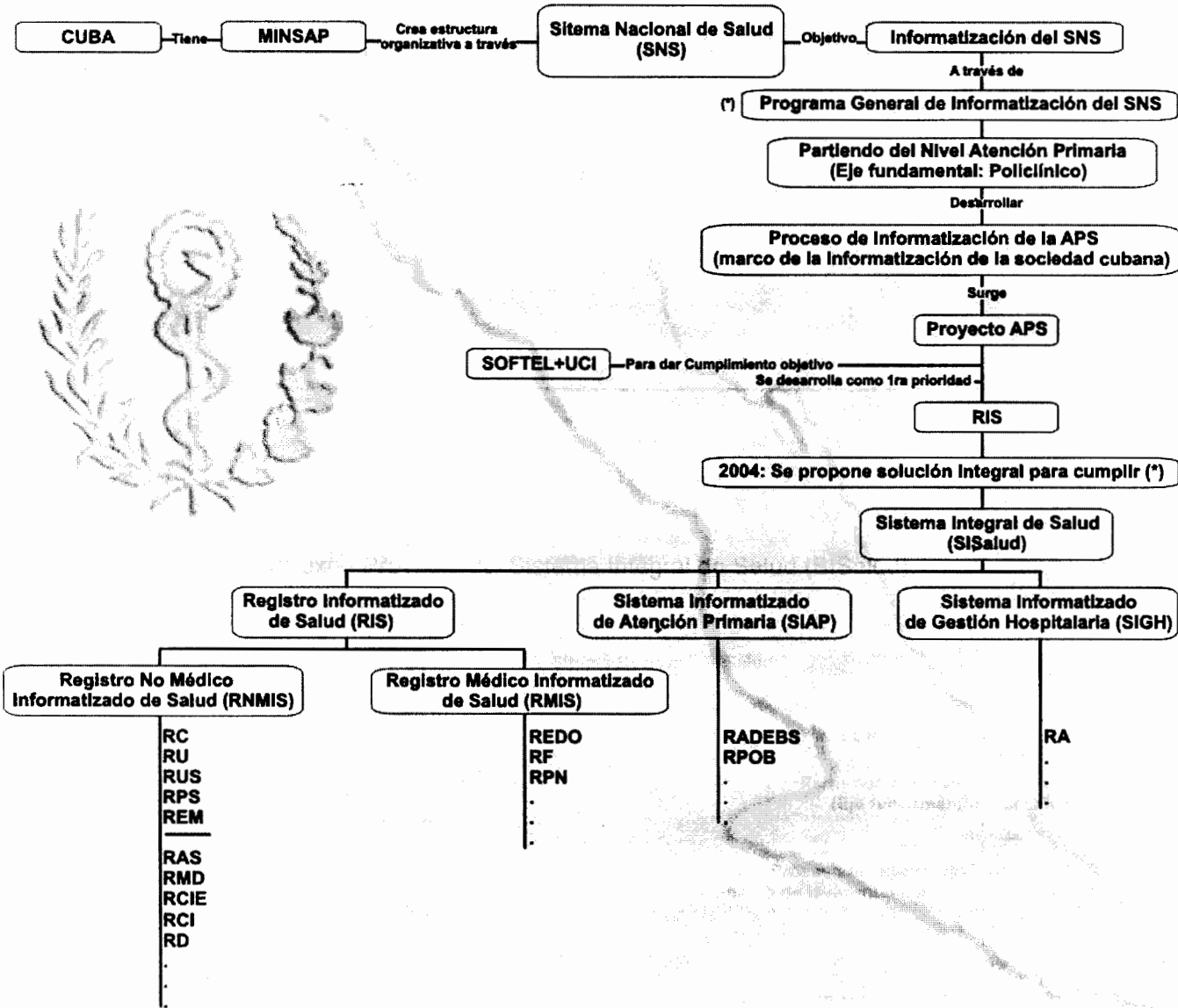
**XHTML:** Acrónimo inglés de *eXtensible Hyper Text Markup Language* (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas.

**XML:** Es el acrónimo del inglés *eXtensible Markup Language* (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide. Su objetivo principal es conseguir una página web más semántica. Aunque una de las principales funciones con las que nace sería suceder al HTML, separando la estructura del contenido y permitiendo el desarrollo de vocabularios modulares, compatibles con cierta unidad y simplicidad del lenguaje (objetivo que se viene desarrollando a través de la especificación XHTML), tiene otras aplicaciones entre las que destaca su uso como estándar para el intercambio de datos entre diversas aplicaciones o software con lenguajes privados como en el caso del SOAP.

**XSLT:** Es el acrónimo de *XML Stylesheets Language for Transformation*, o lenguaje de transformación basado en hojas de estilo. Es un lenguaje de programación que se usa para convertir documentos XML en otros documentos XML.

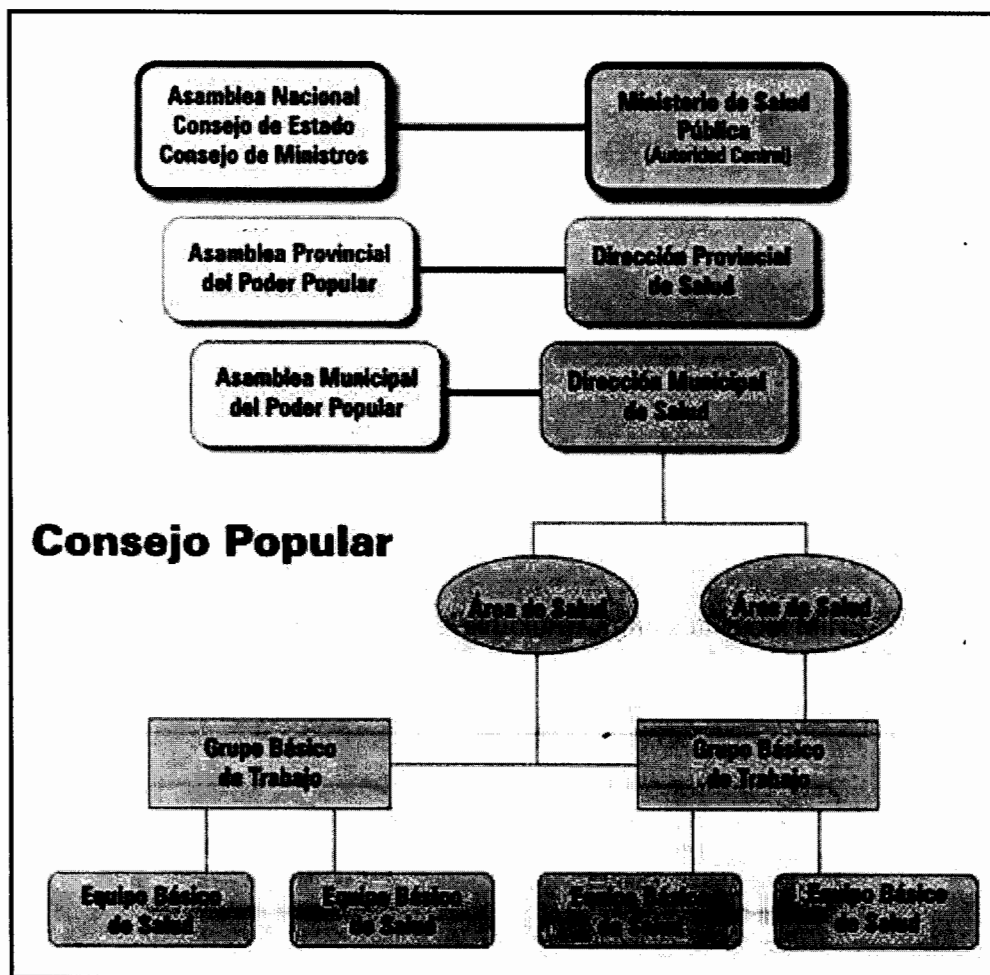
ANEXOS

Anexo I Proyecto del Sistema Integral de Salud (SISalud).

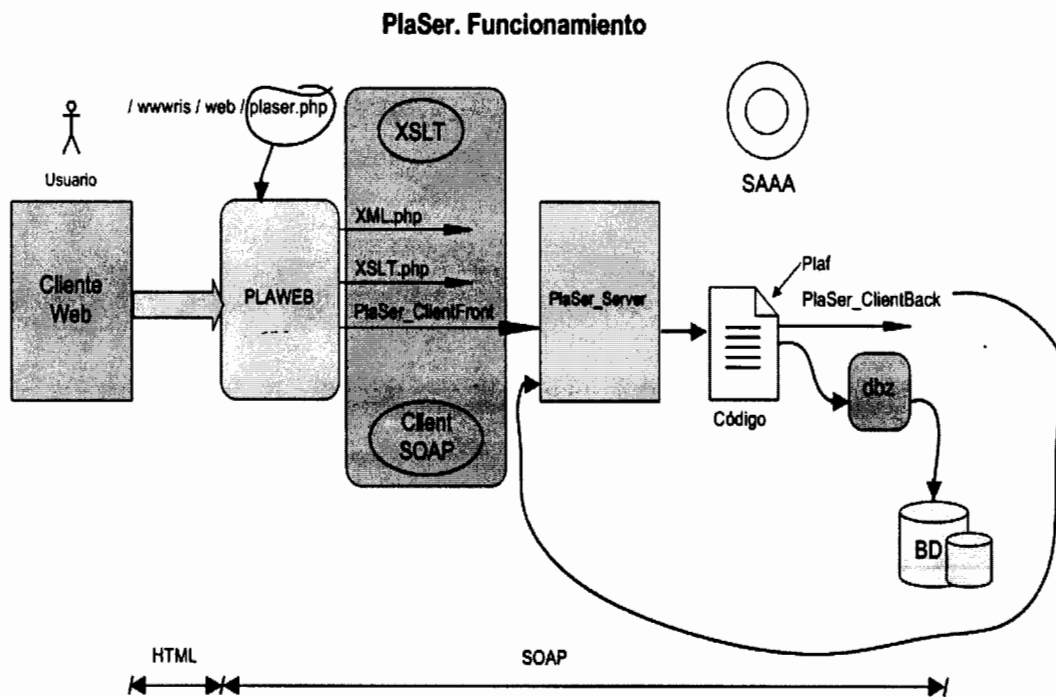




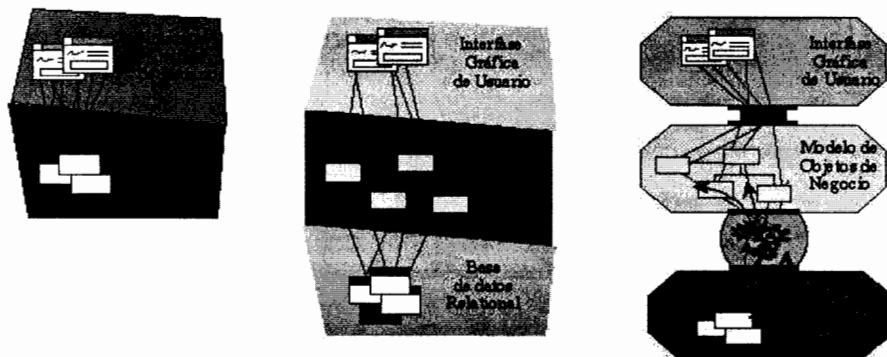
Anexo II Esquema de la Estructura Organizativa del MINSAP.



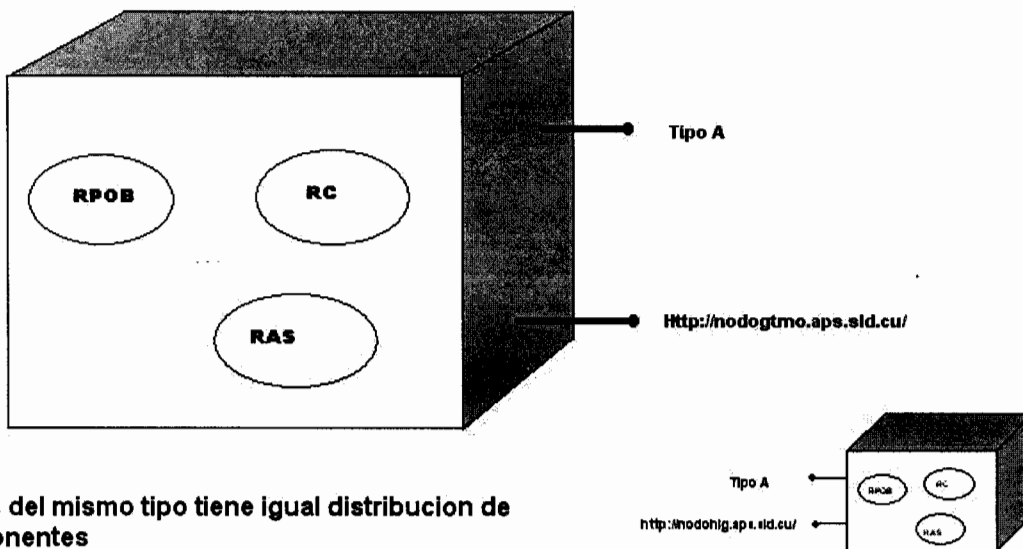
Anexo III Funcionamiento de PlaSer.



Anexo IV Arquitectura lógica de tres capas de una aplicación cliente/servidor

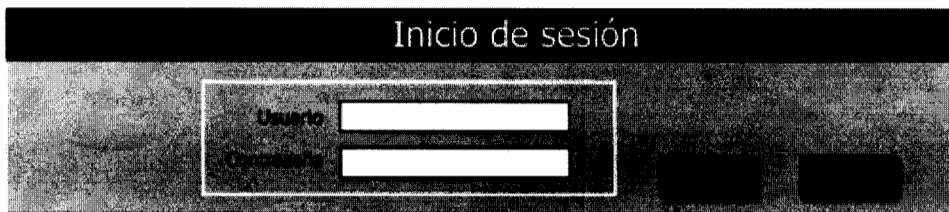


Anexo V Tipo de Nodo.

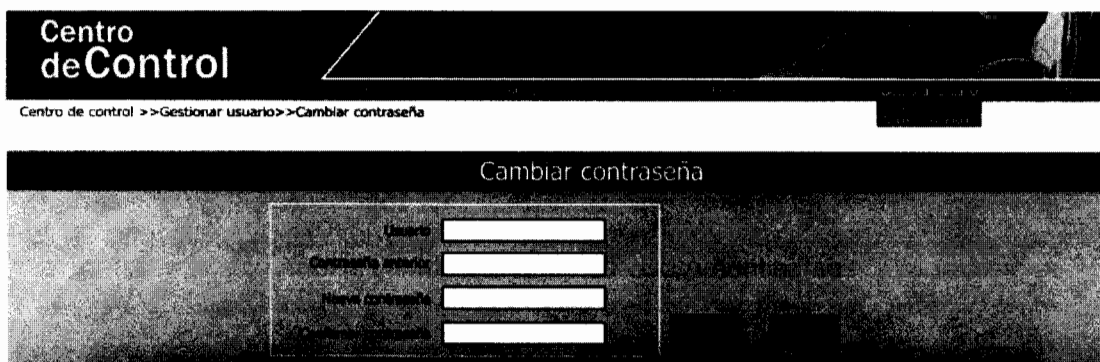


Nodos del mismo tipo tiene igual distribucion de componentes

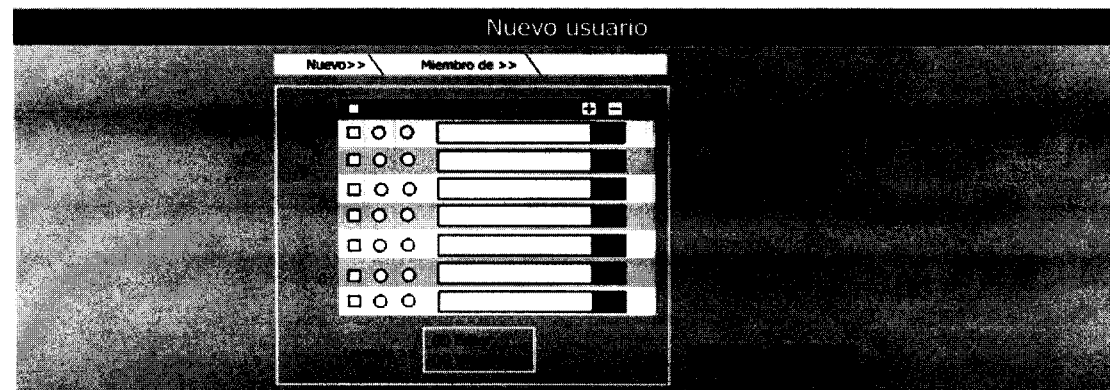
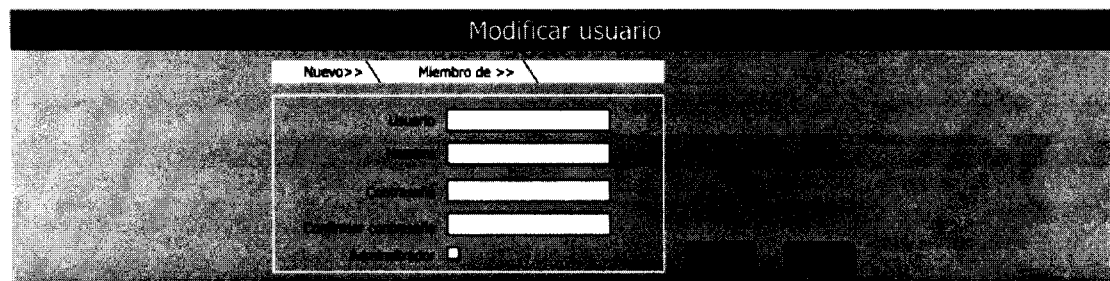
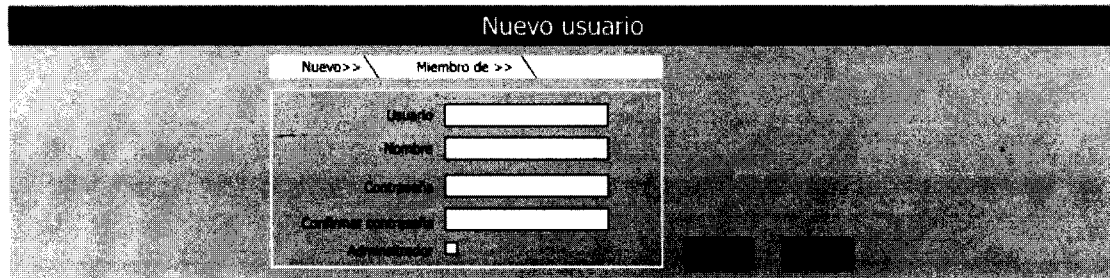
Anexo VI Interfaz Caso de Uso Autenticarse



Anexo VII Interfaz Caso de Uso Cambiar Contraseña



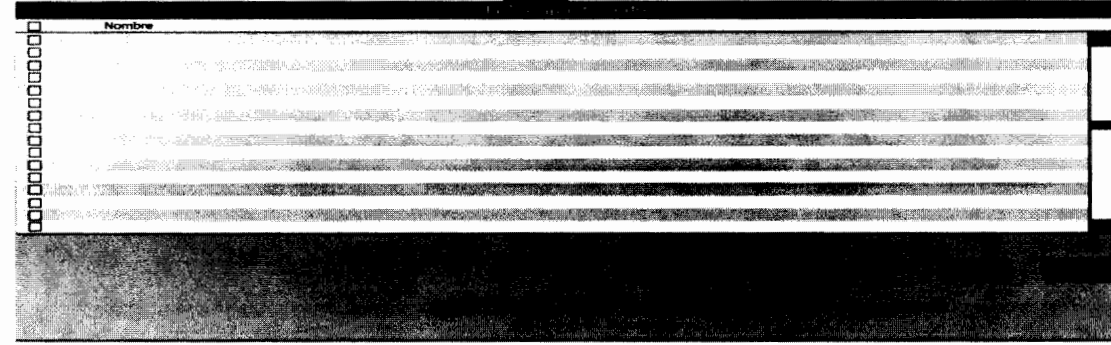
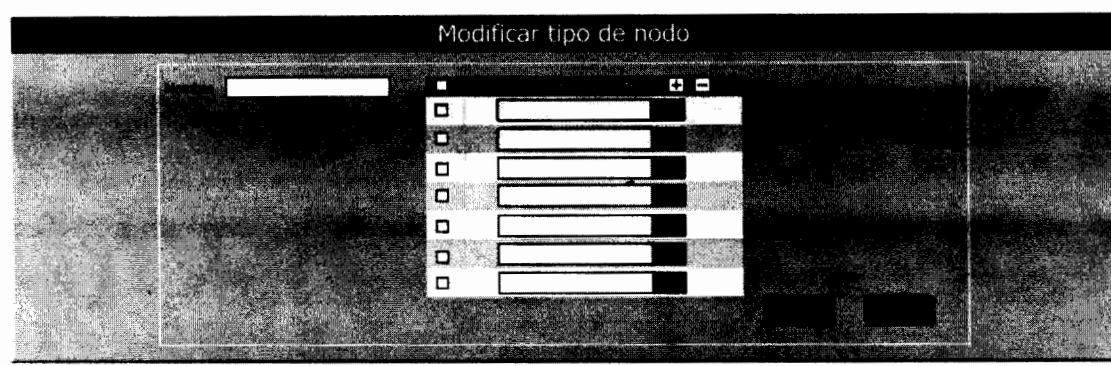
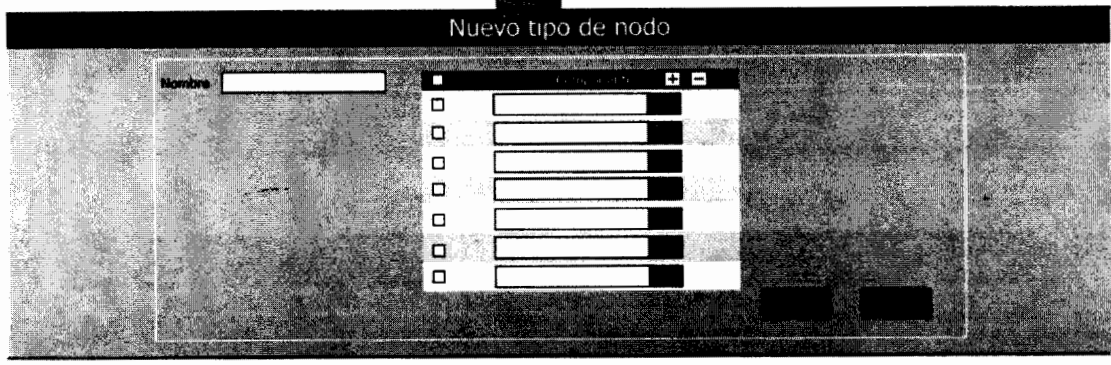
Anexo VIII Interfaces Caso de Uso Gestionar Usuario







Anexo X Interfaces Caso de Uso Configurar TipoNodo



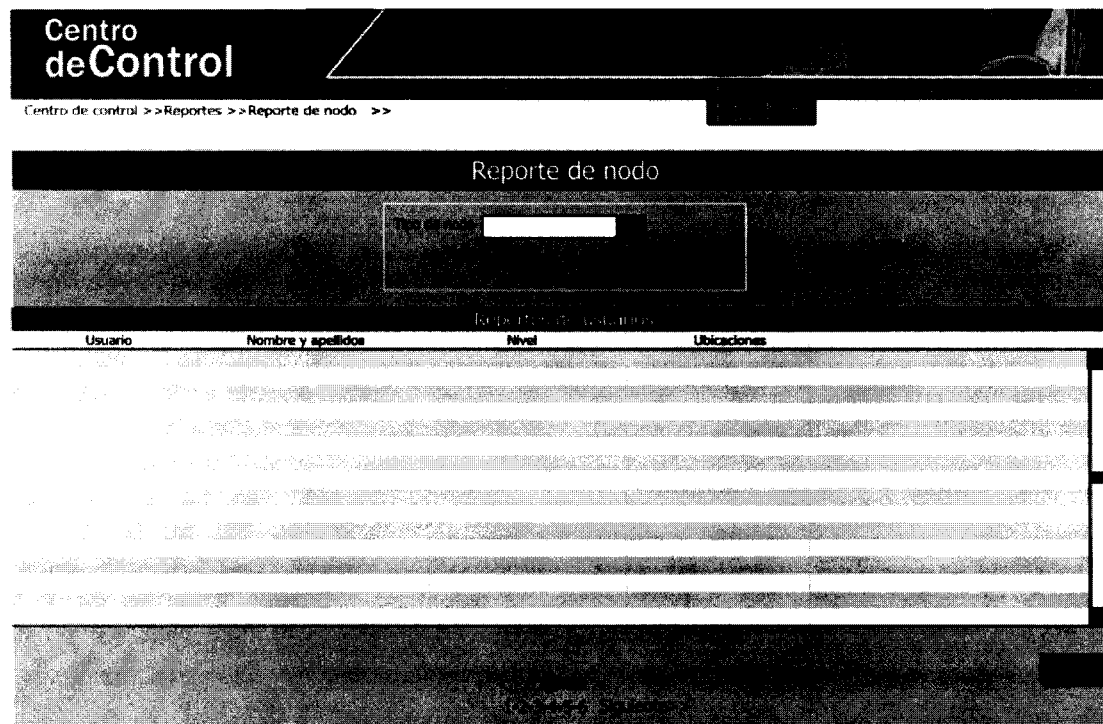
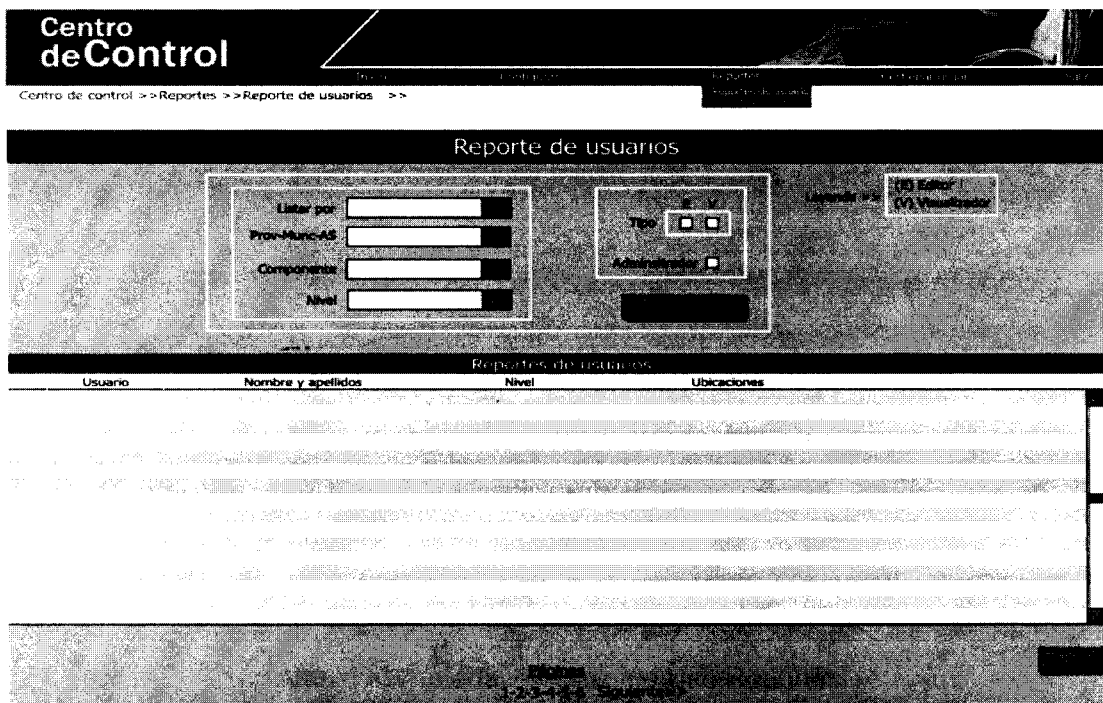








Anexo XIV Interfaces Caso de Uso Reportes.



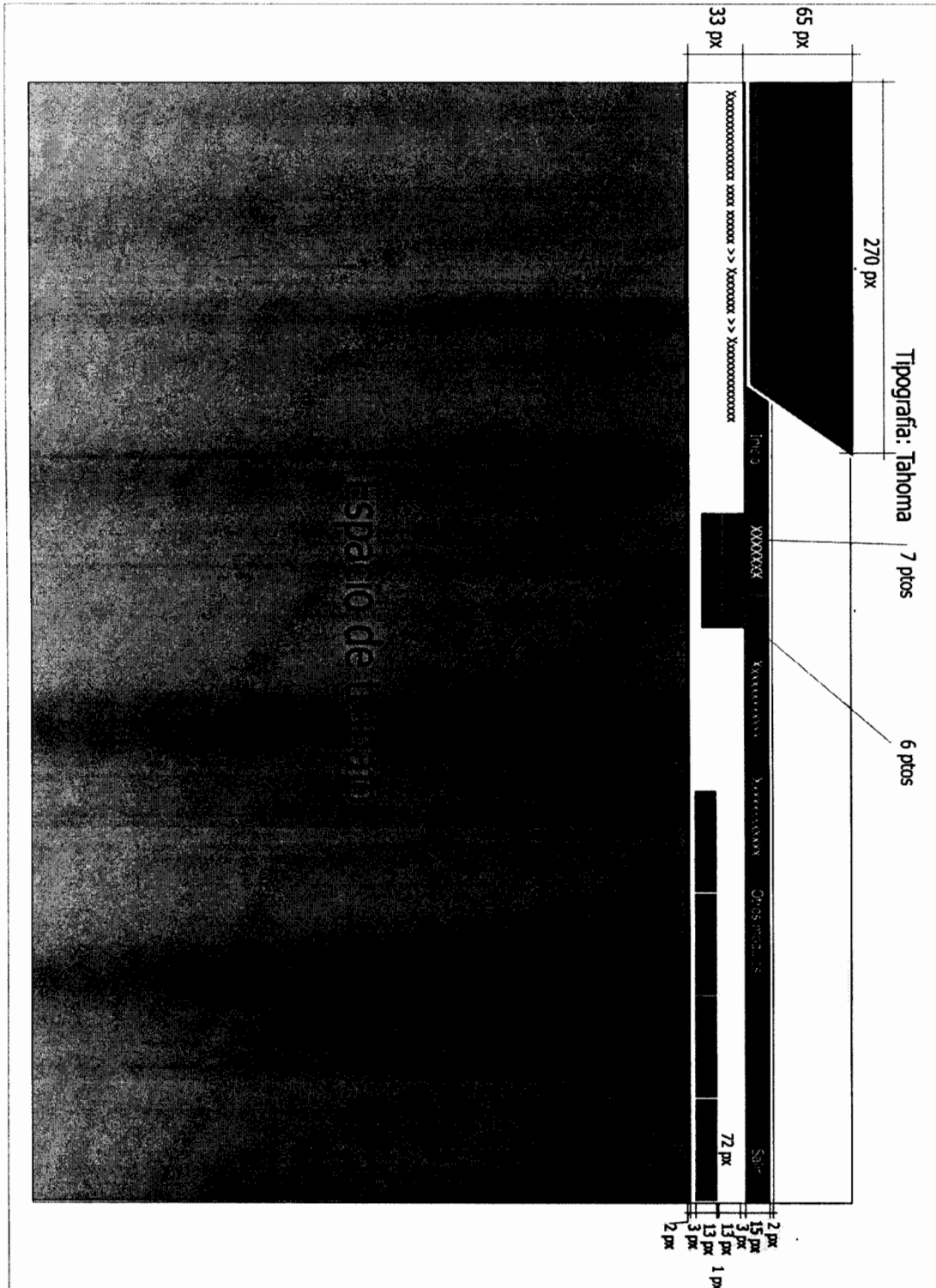
# Centro de Control

Centro de control >>Reportes >>Reporte de componente >>

Reportes de componente			
Nombre	Componente	Dirección nacional	Redireccionamiento
[The table content is extremely faint and illegible in this scan]			

[The footer content is extremely faint and illegible in this scan]

Anexo XV Diseño de Interfaz Gráfica del Proyecto APS



**FE DE ERRATAS**