

**Universidad de las Ciencias Informáticas**  
**Facultad 7**



**Título: Implementación de Estadísticas  
Descriptivas del Registro de Áreas de Salud.**

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

**Autores:** Abdel Alonso Martínez  
Yonhanier Machado Alba

**Tutor:** Ing. Leonardo González

**Asesores:** Dr. Miguel Marín Díaz  
Lic. Caridad Guzmán Vitón

**La Habana, Julio del 2008**

## DECLARACIÓN DE AUTORÍA

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

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

Abdel Alonso Martínez

\_\_\_\_\_

Yonhanier Machado Alba

\_\_\_\_\_

Ing. Leonardo González González

\_\_\_\_\_

### DATOS DE CONTACTO

#### Tutor:

**Ing. Leonardo González González:** Ingeniero en Ciencias Informáticas recién graduado. Posee 2 años de experiencia en el desarrollo de software, donde a desempeñado los roles de programador y diseñador. Actualmente se desempeña como profesor de Práctica Profesional en la Facultad 7. Forma parte del Proyecto Atención Primaria de Salud.

e-mail: [lgonzalezg@uci.cu](mailto:lgonzalezg@uci.cu)

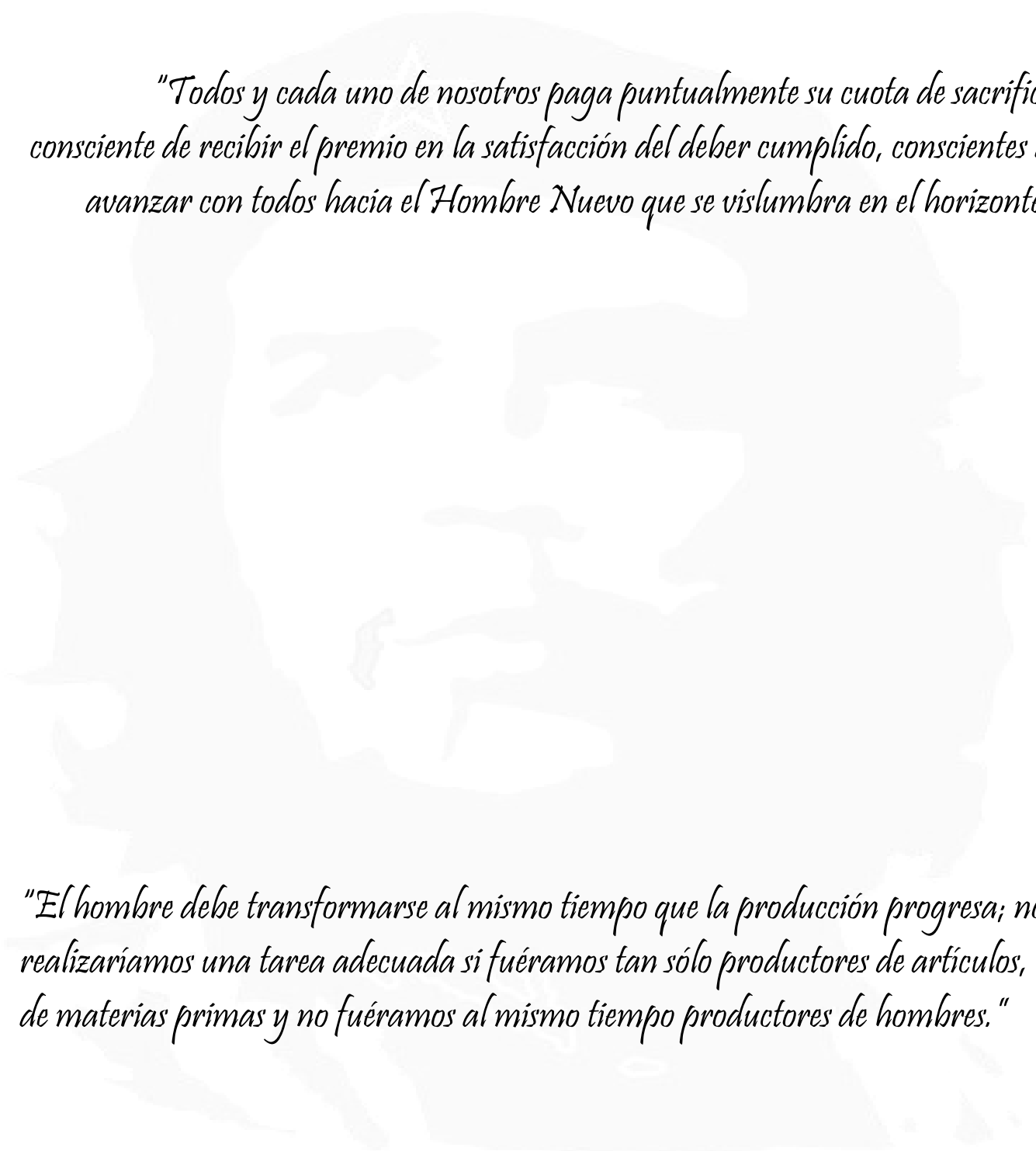
#### Asesores:

**Lic. Gregoria Caridad Guzmán Vitón:** Especialista A en Sistemas Organizativos e Informativos de la empresa Softel. Graduada en Cibernética Matemática en el año 1986 en la Universidad de La Habana. Posee 20 años de experiencia en el desarrollo de software desempeñando diferentes roles. Ha trabajado en las empresas Textilera Rubén Martínez Villena (1987 – 1993) y Softel (1993 hasta la actualidad). Trabajó en la empresa mixta BC BIOCON Internacional S.A. España (1996-2001). Ha pasado varios cursos de superación y postgrado. Ha sido tutora de tesis. Ha participado en eventos científicos.

e-mail: [cary@softel.cu](mailto:cary@softel.cu)

**Dr. Miguel Eusebio Marín Díaz:** Graduado de Médico en el año 1989 en el ISCMH y Especialista de 1er Grado en Medicina General Integral en 1994 en el ISCMH. Posee categoría docente de Profesor Asistente de la Facultad de Ciencias Médicas Comandante Manuel Fajardo donde forma parte de la Cátedra de Medicina General Integral y de Informática Médica. Es Profesor Asistente Adjunto de la UCI. Ha impartido la asignatura de Segundo Perfil de Salud en la Facultad 7 desde el curso 2005-2006. Ha presentado ponencias en eventos científicos nacionales e internacionales. Se desempeña como Experto Funcional del MINSAP del Proyecto APS en la Empresa Softel.

e-mail: [marin@infomed.sld.cu](mailto:marin@infomed.sld.cu).



*"Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte."*

*"El hombre debe transformarse al mismo tiempo que la producción progresa; no realizaríamos una tarea adecuada si fuéramos tan sólo productores de artículos, de materias primas y no fuéramos al mismo tiempo productores de hombres."*

*Che*

**AGRADECIMIENTOS**

*La culminación de este trabajo, a pesar de nuestros esfuerzos, no hubiese sido posible sin la colaboración de un grupo de compañeros que emplearon parte de su tiempo en nuestro empeño porque esta labor tuviera exitosos resultados. Es por ello que quisiéramos aprovechar esta sección para demostrarles, a todos ellos, nuestro agradecimiento infinito por la ayuda prestada.*

*Leque este profundo agradecimiento:*

*A nuestros padres fundamentalmente, por habernos guiado por el buen camino y haber hecho posible que la escalada a esta cima no hubiese sido en vano.*

*A nuestro tutor que nos apoyó en todo momento.*

*A nuestros hermanos y amigos por sus sanos consejos.*

*A aquellos que transitaron estos últimos años junto a nosotros en esta honrosa misión en la cual nos encomendamos hace cinco años para alcanzar nuestros sueños.*

*A nuestras novias por ayudarnos y apoyarnos siempre.*

*A todos nuestros profesores que cumplieron con su objetivo y formaron dos profesionales para el bien de la sociedad.*

*A la Revolución Cubana, en especial al Comandante en Jefe por ser guía, y el principal creador e impulsor de esta Universidad.*

*En fin, para no olvidar a nadie...*

*A todos, muchas gracias!!!*

**DEDICATORIA**

*Dedico este trabajo:*

*A mis padres que siempre me apoyaron y que son en parte responsables de este resultado. A mi hermano que quiero mucho. A Maylencita. A todos mis amigos y compañeros, especialmente a Yosleiby, Yonhanier, Reinaldo y todos los demás, que de una forma u otra siempre estuvieron ahí. A todos los que confiaron en mí.*

*Abdel*

*El esfuerzo y la dedicación que he puesto en esta tesis, va con mucho cariño:*

*A toda mi familia, principalmente a mis padres, quienes me han brindado amor incondicional y siempre me han impulsado a superarme profesionalmente, ofreciéndome aliento constante para lograrlo. A mis hermanos y abuelos a quienes quiero mucho. A mis amigos de Baracoa que han estado conmigo en las buenas y en las malas. A mis colegas de la UCI que me apoyaron en todo momento, principalmente al Newpa, el Yasse, Rey, el Fide, el Joaco, el Yohan, Osmay, Marcos, Hayler, el Yuma, el Daiche, el Yuri. Nunca los olvidaré!*

*Yonhanier*

## RESUMEN

El presente trabajo tiene como objetivo, el modelado y documentación del análisis de las funcionalidades encargadas de brindar las Estadísticas Descriptivas del Registro de Áreas de Salud (RAS). Además el Diseño e Implementación de los Servicios Web para el negocio y la capa de presentación de dichas funcionalidades que permita gestionar la información de las Estadísticas Descriptivas.

Para especificar, construir y documentar los artefactos del sistema propuesto se utilizó el Proceso de Desarrollo Unificado (RUP), el cual hace uso del Lenguaje Unificado de Modelado y el editor visual Macromedia Dreamweaver 8.0, para la creación del prototipo de interfaz de usuario. Además se hizo uso de la tecnología XML Web Service y la arquitectura orientada a servicios y basada en componentes (SOA-CBA) para aplicaciones distribuidas.

Las funcionalidades propuestas permitirán al país contar con un sistema automatizado para la gestión de las estadísticas descriptivas del RAS, acelerando los procesos de transmisión de la información, debido a que no se dispone en el RAS de las funcionalidades que permitan brindar la información de los Grupos Básicos de Trabajo (GBT) y Equipos Básicos de Salud (EBS) que la conforman; poblaciones que atienden, formadas por viviendas o centros laborales y los locales de consultas o de viviendas.

**Palabras claves:** Registro de Áreas de Salud, Estadísticas Descriptivas, Grupos Básicos de Trabajo, Equipos Básicos de Salud.

**TABLA DE CONTENIDOS**

**INTRODUCCIÓN..... 1**

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 7**

    Introducción ..... 7

    1.1 El Sistema Nacional de Salud ..... 7

    1.2 Conceptos asociados al dominio del problema ..... 11

    1.3 Análisis de las soluciones existentes..... 13

    1.4 Tecnologías, Herramientas y Metodologías utilizadas..... 14

        1.4.1 Tecnologías ..... 14

        1.4.2 Metodologías de Desarrollo..... 23

        1.4.3 Lenguajes de Programación Web ..... 24

        1.4.4 Sistemas de Gestión de Bases de Datos (SGBD) ..... 28

        1.4.5 Herramientas a utilizar ..... 32

    1.5 Conclusiones ..... 34

**CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA..... 35**

    2.1. Introducción. .... 35

    2.2. Situación Problémica..... 35

    2.3. Objetos de automatización. .... 36

    2.4. Modelo de Dominio ..... 37

        2.4.1. Conceptos Fundamentales..... 37

        2.4.2 Diagrama del Modelo de Dominio ..... 39

    2.5. Propuesta del Sistema ..... 40

        2.5.1. Especificación de Requerimientos de Software..... 40

            2.5.1.1 Requerimientos Funcionales ..... 40

            2.5.1.2 Requerimientos No Funcionales..... 40

        2.5.2. Modelo de Casos de Uso del Sistema..... 47

            2.5.2.1 Definición de Actor del Sistema..... 47

            2.5.2.2 Diagrama de Casos de Uso del Sistema ..... 48

            2.5.2.3 Descripción de los Casos de Uso del Sistema ..... 50

    2.6 Conclusiones ..... 52



<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>	<b>53</b>
3.1. Introducción .....	53
3.2. Análisis .....	53
3.2.1. Modelo Análisis .....	53
3.2.2. Diagrama de Clases del Análisis .....	54
3.2.3 Diagramas de Interacción .....	58
3.3 Modelo de Diseño .....	64
3.3.1 Justificación del Uso de Patrones.....	64
3.3.2 Estructura del Diseño .....	65
3.3.3 Definición de Elementos del Diseño .....	67
3.3.4 Diagramas de Clases del Diseño .....	69
3.3.5 Descripción de las Clases del Diseño .....	74
3.3.6 Otras Clases Utilizadas .....	81
3.4 Conclusiones .....	82
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>83</b>
4.1 Introducción .....	83
4.2 Dependencias y Relaciones con otros Sistemas .....	83
4.2.1 Componente de Seguridad (SAAA).....	83
4.2.2 Registro de Unidades de Salud (RUS) .....	84
4.2.3 Registro de Localidades (RL) .....	84
4.2.4 Registro de Ubicación (RU).....	84
4.2.5 Registro de Personal de la Salud (RPS).....	85
4.3 Implementación.....	85
4.3.1 Modelo de Implementación .....	85
4.3.1.1 Diagrama de Componentes.....	87
4.3.2 Diagrama de Despliegue .....	88
4.3.3 Descripción de los Métodos del Negocio.....	91
4.3.4 Estándares de diseño, codificación y tratamiento de errores. ....	91
4.3.4.1 Estándares de Diseño .....	91
4.3.4.2 Estándares de Codificación.....	93
4.3.4.3 Tratamiento de Errores .....	95

4.4 Prueba .....	96
4.5 Conclusiones .....	98
<b>CONCLUSIONES .....</b>	<b>99</b>
<b>RECOMENDACIONES .....</b>	<b>100</b>
<b>BLIOGRAFÍA .....</b>	<b>101</b>
<b>ANEXOS .....</b>	<b>103</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>106</b>

## INTRODUCCIÓN

Una política científica nacional se compone de una doctrina, es decir de un conjunto de principios y objetivos. Al analizar cómo podía manifestarse en Cuba esa política, es necesario tener en cuenta lo planteado por Marx en el prólogo a *Contribución a la Crítica de la Economía Política*: "...la humanidad se propone siempre únicamente los objetivos que puede alcanzar, pues, bien miradas las cosas, vemos siempre que estos objetivos sólo brotan cuando ya se dan o por lo menos se están gestando, las condiciones materiales para su realización". En tal sentido, los objetivos de la política científica en Cuba, se orientan a apoyar los objetivos básicos del desarrollo económico y social del país.<sup>1</sup>

Cuando quien estudia el desarrollo de la Ciencia y la Tecnología en Cuba se asoma al proceso de despliegue desde 1959, una de las cuestiones que enseguida nota, y que le asombra, es la clara percepción que, desde los primeros momentos, tuvo la alta dirección de la revolución del papel de la ciencia y la tecnología en el desarrollo económico, el temprano reconocimiento de adoptar rápidas medidas de promoción de lo que hoy se conoce como progreso científico-técnico.

En la actualidad, debido a los grandes avances en el desarrollo de las Tecnologías de la Informática y las Comunicaciones (TICs), es cada vez mayor el interés por mejorar los procesos administrativos, las finanzas y la producción de las instituciones empresariales. Lo que ha impuesto la necesidad de una rápida adopción de sistemas automáticos capaces de facilitar tareas mecánicas y rutinarias, evitar errores y mejorar la satisfacción de los clientes, provocando una gran revolución en todos los procesos de tomas de decisiones.

El Sistema Nacional de Salud (SNS) de Cuba está organizado por tres niveles de atención. El nivel de Atención Primaria de Salud (APS) es el que debe dar solución a la mayor cantidad de problemas de salud de la población, sus actividades se realizan en cualquier unidad del Sistema Nacional de Salud y están relacionados fundamentalmente con las que se realizan en el Policlínico y en los Consultorios del Médico de la Familia, los Hospitales Rurales, los Dispensarios y las Postas Médicas. La Atención Secundaria que es la brindada por los Hospitales principalmente en cada una de las regiones, y la Atención Terciaria que es la que brindan los Hospitales Especializados.<sup>2</sup>

---

<sup>1</sup> Monografías. [cited; Available from: <http://www.monografias.com/trabajos14/informatica-social/informatica-social.shtml#inf>

<sup>2</sup> Rodríguez Osoria, D., Paneque Meschenkov, A. L., Sarmiento Borrero, Y. R., *Diseño y Servicios Web para el Registro de Áreas de Salud de la Atención Primaria del Sistema Nacional de Salud*. 2007, Universidad de las Ciencias Informáticas: Ciudad de La Habana.

La Atención Primaria de Salud cuenta con aplicaciones que la convierten en una parte informatizada dentro del SNS, pero la información a nivel Nacional, Provincial, Municipal y de Unidades de Salud no muestra resultados estadísticos que permitan la toma de decisiones globales en los diferentes niveles de dirección.

Una de las mencionadas aplicaciones es el Registro de Áreas de Salud el mismo gestiona la información de las áreas de salud a nivel nacional, permitiendo un control de las mismas, como están formadas según la estructura organizativa propuesta para la APS, así como los integrantes según la plantilla del GBT y EBS. Este control permitirá:

- Tener a la población mejor informada de las características de las unidades de salud que le brindan atención, como por ejemplo: dónde están ubicadas, qué áreas geográficas atiende, qué servicios brinda, cuantas especialidades ofrecen sus servicios, etc.
- Se organizarán mejor los servicios que se brinden a la población con la ubicación de estos según las características específicas de cada región.<sup>3</sup>

El Ministerio Nacional de Salud Pública (MINSAP) se ha trazado como propósito elevar la eficiencia, seguridad, calidad y estética en todas las direcciones y departamentos de estadísticas, mejorando para ello procesos y sistemas estadísticos que utiliza. Para cumplir este objetivo se le asigna a SOFTEL, empresa especializada en brindar soluciones informáticas para la salud, la misión de desarrollar productos y servicios informáticos que eleven la eficiencia del sistema de salud cubano y se ganen un espacio en el mercado internacional, ofreciendo soluciones informáticas con personal y tecnologías de avanzada.

Para ello se 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 APS, en calidad de expertos funcionales en estrecho vínculo con los especialistas de informática. Participa además la Universidad de las Ciencias Informáticas (UCI), quien a través de una organización de trabajo en proyectos productivos, contribuye con la incorporación del personal que está formando parte de este proceso de desarrollo y a su vez permite adentrar a estudiantes y profesores de la universidad en la vida productiva. 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

---

<sup>3</sup> Sistema de Información para la Salud (SISalud), 2008.

normativa para los proyectos que deben irse abriendo de ahora en adelante en la informatización del sector de la salud.<sup>4</sup>

La necesidad de un enfoque estadístico está actualmente bien reconocido en la investigación y en la práctica de las disciplinas que constituyen la salud pública. Ya que estas estudian comunidades o poblaciones en las que claramente se aplican las leyes de los grandes números y de las fluctuaciones aleatorias. La estadística permite analizar situaciones en las que los componentes aleatorios contribuyen de forma importante en la variabilidad de los datos obtenidos. En salud pública los componentes aleatorios se deben, entre otros aspectos, al conocimiento o a la imposibilidad de medir algunos determinantes de los estados de salud y enfermedad, así como a la variabilidad en las respuestas por los pacientes, similares entre si, que son sometidos al mismo tratamiento. La extensión de los conocimientos y aptitudes de carácter estadístico que necesitan adquirir los profesionales de la salud pública son importantes, porque el conocimiento de los principios y métodos estadísticos y la competencia en su aplicación se necesitan para el ejercicio eficaz de la salud pública, y adicionalmente para la comprensión e interpretación de los datos sanitarios; a fin de discriminar entre opiniones arbitrarias o discrecionales, con respecto a las verdaderamente evaluadas en un contexto científico.<sup>5</sup>

La Estadística Descriptiva es un paso inicial y obligatorio en cualquier tipo de análisis estadístico, analiza metódicamente los datos, simplificándolos y presentándolos en forma clara; eliminando la confusión característica de los datos preliminares. Permite la elaboración de cuadros, gráficos e índices bien calculados; suficientemente claros, como para disipar las dudas y la obscuridad de los datos masivos. Se limita a describir los datos que se analizan. La Estadística Descriptiva, por ejemplo, proporciona información general sobre estas observaciones, como los valores medio y mediano, los errores observados, y la distribución de los valores.

Aunque hay tendencia a generalizar a toda la población las primeras conclusiones obtenidas tras un análisis descriptivo, su poder inferencial es mínimo y debería evitarse tal proceder. Otras ramas de la estadística se centran en el contraste de hipótesis y su generalización a la población.

Básicamente, se lleva a cabo un estudio calculando una serie de medidas de tendencia central, para ver en qué medida los datos se agrupan o dispersan en torno a un valor central.

---

<sup>4</sup> Marín Díaz, M. E. (2004). Consideraciones sobre el Proyecto de Informatización de la Atención Primaria de Salud. Revista Habanera de Ciencias Médicas. 3.

<sup>5</sup> Cantú Martínez, P. C., [cited; Available from: <http://www.respyn.uanl.mx/iv/1/ensayos/bioestadistica.html>

Ejemplos de este tipo de análisis descriptivo pueden encontrarse en la prensa diaria, en la parte de información económico-social: series de tiempo, gráfica de barras, índices de precios, resultados de una encuesta y más elaborado, para más de una variable, en pirámide de edades, comparativas, etc.

Otros ejemplos: Índice de precios de consumo, Resultados deportivos, Accidentes laborales y, en general, hechos cuantificados en valores absolutos (tal cual), en porcentajes (%) o en índices (con un período base inicial = 100).

Existen sistemas informáticos desarrollados a nivel internacional relacionados con las Áreas de Salud, pero sus funcionalidades están dirigidas a la gestión hospitalaria y al trabajo que realizan los médicos para prevenir y dar solución a los problemas de salud que afectan a sus poblaciones. En Cuba se han hecho algunos intentos de mostrar información de las Áreas de Salud, pero no pasan de ser Sitios Web estáticos o trabajos investigativos que muestran mediante imágenes la estructura organizativa que puede tener un Área de Salud, resaltando siempre los Consultorios del Médico de la Familia y la ubicación de las poblaciones que atiende un Equipo Básico de Salud, así como algunos datos estadísticos que se relacionan con las investigaciones y estudios que realizan los médicos con sus poblaciones.<sup>6</sup>

La **Situación Problemática** viene dada a que actualmente, el RAS del Sistema de Información para la Salud (SISalud) gestiona la información de todas las áreas del país pero no cuenta con reportes que permitan consultar la misma en los diferentes niveles de dirección de salud. Se envía información a estos niveles en documentos, que en la medida en que suben por los mismos van disminuyendo su precisión cualitativa, solamente algunos de estos datos (cuantitativos) se recogen en modelos estadísticos que forman parte del Sistema de Información Estadístico Complementaria (SIE-C) del Ministerio de Salud Pública (MINSAP).

Tomando como premisa la infraestructura existente en el país y lo anteriormente planteado, el **Problema a Resolver** es: ¿Cómo consultar de forma eficiente la información de las Áreas de Salud en los diferentes niveles del Sistema Nacional de Salud?

Para lograr este propósito se identificó como **Objeto de Estudio** el proceso de gestión de la información estadística del SISalud.

---

<sup>6</sup> Ídem Referencia 1

El **Campo de Acción** de esta investigación está centrado en el proceso de gestión de las Estadísticas Descriptivas del RAS.

Como **Objetivo General** se ha propuesto, implementar las Estadísticas Descriptivas del RAS, para la obtención de la información estadística en los diferentes niveles del SNS, garantizando la calidad y fiabilidad de la información.

Como **idea a defender** se planteó la siguiente: Con el desarrollo de las funcionalidades que abarca el contenido de los reportes del RAS se permitirá un mejor control de la información y toma de decisiones en los diferentes niveles del SNS.

Para dar cumplimiento a todo lo anteriormente planteado se definieron las siguientes **Tareas de la Investigación**:

- Asimilar la Arquitectura definida por el MINSAP (Orientada a Servicios y Basada en Componentes (SOA\_CBA)) para el desarrollo de sus aplicaciones: PLASER y Registro Informatizado de Salud (RIS).
- Especificar los requerimientos funcionales y no funcionales del sistema.
- Describir las herramientas y tecnologías informáticas para la implementación de los Reportes del Registro de Áreas de Salud.
- Analizar la integración del RAS con otros componentes o sistemas del SNS o fuera de éste (Registro Personal de Salud, Registro de Ubicación, Registro de Localidades, Registro de Servicios Médicos, Registro de Unidades de Salud, Registro de Estudiantes).
- Modelar los Flujos de Trabajo “Modelamiento del Negocio”, “Gestión de Requerimientos”, “Análisis”, “Diseño”, “Implementación” de los Reportes del Registro de Áreas de Salud, siguiendo el Proceso Unificado de Rational (RUP).
- Implementar los Reportes del Registro de Áreas de Salud, utilizando estándares de codificación, tratamiento de errores o excepciones.

La investigación realizada se estructurará en cuatro capítulos como sigue:

En el **Capítulo 1:** Fundamentación Teórica. Se hace referencia a los trabajos realizados que existen en la actualidad y que se vinculan con el nuestro, los principales conceptos asociados al dominio del problema y el estado del arte de las tecnologías y herramientas a utilizar en el desarrollo de la propuesta.

En el **Capítulo 2:** Características del Sistema. Se refleja la investigación realizada con los procesos que tienen lugar en el negocio como objeto de estudio, la situación problemática existente además del objetivo general a cumplir durante el desarrollo del producto. Se describe la propuesta del sistema, se aborda lo referente al funcionamiento del negocio, abordando sus reglas, descripción y las mejoras que propone el mismo. Se describe además la solución propuesta, utilizando los requerimientos funcionales y no funcionales, los casos de uso, el diagrama de casos de uso del sistema y un prototipo de interfaz de usuario.

En el **Capítulo 3:** Análisis y Diseño del Sistema. Se realiza el análisis del sistema a desarrollar, con el propósito de refinar y estructurar los requisitos obtenidos con anterioridad para facilitar la comprensión, preparación, modificación y mantenimiento de los mismos. Describe los aspectos relacionados al diseño de la solución propuesta, se modelan los diagramas de clases del diseño y las clases persistentes, se plantea el modelo de datos, y se especifican los principios para el diseño gráfico.

En el **Capítulo 4:** Implementación y Prueba. Se tratan los aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de componentes y despliegues, aborda la descripción de los estándares de diseño, codificación y además del tratamiento de errores en la solución del sistema. También se aborda el tema relacionado con las pruebas que se le realizan a la solución propuesta.



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### Introducción

Este capítulo tiene como objetivo fundamental abordar distintos aspectos que se utilizan como soporte teórico para el desarrollo de las funcionalidades que agrupan las estadísticas que brinda el Registro de Áreas de Salud. Se argumenta sobre todo el proceso de informatización del Sistema Nacional de Salud. Además se realiza el análisis de las técnicas, tecnologías, metodologías y herramientas de software sobre las cuales se llevara a cabo el proceso de desarrollo de las funcionalidades que permiten agrupar y brindar información estadística de los datos generados por las Áreas de Salud en Cuba.

### 1.1 El Sistema Nacional de Salud

A partir del triunfo de la Revolución el gobierno revolucionario se encargó de enfrentar y erradicar los problemas sanitarios existentes en el país y con esto elevar sustancialmente los niveles de vida y salud de la población. Para lograrlo dictó las primeras medidas en el campo de la salud, las cuales fueron:

- ◆ Fundación del Sistema Nacional de Salud (SNS).
- ◆ Implementación del Servicio Médico Rural.
- ◆ Creación de las Áreas de Salud y los Policlínicos.

El Sistema Nacional de Salud se puede clasificar mediante la distribución administrativa (**Ver Anexo I**) en:

- **Nivel Nacional:** Es el Ministerio de Salud Pública como órgano rector de la salud cubana en sí. Incluye en él todas sus funciones inherentes (metodológicas, normativas, de coordinación y control).
- **Nivel Provincial:** En cada provincia del país existe una dirección de salud (Dirección Provincial de Salud), la cual se subordina a la Asamblea Provincial del Poder Popular ya sea en lo relativo a administración, como al financiamiento.
- **Nivel Municipal:** Al igual que en la provincia, en los municipios existe una dirección de salud que se subordina a la instancia municipal del Poder Popular que le corresponde. En los municipios también los llamados Consejos Populares.

Y según los niveles de atención se puede dividir en:

- **Atención Primaria de Salud (APS):** Este nivel, cuya esencia es la participación activa de la comunidad en la solución de sus problemas de salud, es el nivel más básico de atención al paciente, el cual no sólo se restringe al consultorio médico o al policlínico, sino que puede darse en cualquier institución con carácter médico. Su función está dada por brindar servicios médicos de carácter ambulatorio y de consultoría generalmente, como pueden ser emitir un diagnóstico o un tratamiento con poca complejidad. Si se sospecha una enfermedad grave en una persona se remite al nivel de atención superior que corresponda. Este nivel comprende tanto a personas sanas como enfermos dentro de su jurisdicción. En cuanto a los enfermos se les da seguimiento a sus padecimientos, ya sean crónicos o no, y se les proporciona bienestar de vida a pacientes con patologías incurables.
- **Atención Secundaria de Salud:** A este tipo de atención se llega generalmente mediante remisión por parte del médico en la atención primaria. Ya aquí se ofrecen servicios técnicos y terapéuticos de gran complejidad frente a carencias de salud de moderadas a graves. Estos servicios de salud se obtienen de manera ambulatoria en policlínicos u hospitales o mediante la hospitalización en sí.
- **Atención Terciaria de Salud:** Este nivel de atención se caracteriza por la especialización en el servicio y sólo se brinda en centros específicos con funciones bien enfocadas como centros o institutos de investigación. Ejemplo de esta atención también pueden ser Servicios de Neurocirugía, Nefrología, Cirugía Cardiovascular o Trasplante Renal.

### **Informatización del Sistema Nacional de Salud**

Desde los primeros años de la Revolución Cubana, el estudio y el proceso de los servicios de salud para la población es una estrategia política de interés para el Gobierno Revolucionario y el Ministerio de Salud Pública (MINSAP). Con este proceso se pretende crear una infraestructura informática para el sector, al que se integrarán todos los productos o servicios, respondiendo a una Arquitectura Orientada a Servicios y Basada en Componentes (SOA-CBA). En el año 2003 el MINSAP en conjunto con el Ministerio de la Informática y las Comunicaciones (MIC), acuerdan un grupo de acciones y políticas a tener en cuenta para informatizar el SNS, dentro de las cuales se destacan:

- Alinearse con tecnología de punta y estándares de calidad existentes en el mundo adecuándolos a nuestras condiciones particulares.
- Desarrollar una Dirección Integrada de Proyectos para garantizar la viabilidad, sostenibilidad y

mantenimiento.

- Todas las inversiones y proyectos que se desarrollen para el SNS deben considerar el elemento informático desde su concepción inicial.
- La superación y especialización de la informática en la salud será una actividad básica para la formación de los recursos humanos.
- Los productos se desarrollarán con tecnología LAMP (Linux, Apache, MySQL y PHP) para garantizar su sostenibilidad en el tiempo donde se emplearán estándares internacionales para productos relacionados con la salud pública.

Informatizar este sector es de gran importancia pues permite dar una respuesta más eficiente en el campo de la información científica. Además de posibilitar la integración con otros sectores de la sociedad y facilita la cultura general de sus trabajadores, así como su superación profesional.

Actualmente se ha asumido y puesto en marcha de manera gradual de un sistema que gestione la información, el cual se conoce como Sistema de Información para la Salud (SISalud), y está formado por el Registro de Informatizado de Salud (RIS), el Sistema Informatizado de Atención Primaria (SIAP), el Sistema Informatizado de Gestión Hospitalaria (SIGH) y el Sistema Informatizado de la Atención Especializada (SIAE).<sup>7</sup>

El RIS es una plataforma de aplicaciones, abierta, con una interfaz de programación que permite incorporar nuevos módulos compatibles. Este sistema permite disponer información única, confiable y en tiempo real, para la toma de decisiones. Se encuentra centralizada, es decir disponible desde cualquier nivel del SNS. Garantiza la integridad de la información, permitiendo interconectar entre si las diferencias existentes y eliminando de esta forma la duplicación de la misma. Permite consolidar la información en el eslabón fundamental de la atención de salud: el policlínico de nuevo tipo y su estrecha interrelación con los niveles de atención secundario y terciario, lo que garantiza el éxito del proceso de integración.<sup>8</sup>

---

<sup>7</sup> Ortega Valdés, A., Ramírez Rodríguez, Y., *Análisis del Registro de Áreas de Salud de la Atención Primaria del Sistema Nacional de Salud*. 2007, Universidad de las Ciencias Informáticas: Ciudad de La Habana.

<sup>8</sup> López González, T., *Análisis del Registro de Enfermedades de Declaración Obligatoria*. 2007, Universidad de las Ciencias Informáticas: Ciudad de La Habana.

Entre los módulos que pertenecen al RIS se encuentra el Registro de Áreas de Salud (RAS), teniendo en cuenta que en cualquier nivel de atención de salud, se gestiona la información de las distintas áreas de salud existentes en el país, pero que dicha información no se puede consultar, nos hemos dado a la tarea de implementar las funcionalidades en el RAS que permita resolver dicha dificultad.

El proyecto de automatización de las Estadísticas Descriptivas del RAS, esta basado en Servicios Web, esto permitirá la captación de datos, así como la elaboración y visualización de reportes desde las distintas instituciones de salud, una vez que se garantice la conectividad de las mismas.

Con la implementación de estas funcionalidades se dará solución a los problemas que afectan la obtención y consulta de la información de las áreas de salud desde cualquier nivel del SNS. Además que garantizará que los reportes de información estadística sean realizados en tiempo real.

### **El Sistema de Información para la Salud (SISalud)**

El Sistema de Información para la Salud (SISalud) es la integración de un conjunto de aplicaciones cuyo propósito fundamental es la informatización del Sistema Nacional de Salud (SNS). Cada uno de los componentes que integran este sistema desarrolla sus procesos y los pone a disposición del resto, permitiendo la interoperabilidad, el intercambio de información entre los mismos y el acceso a la información desde los diferentes niveles de dirección, ya sea nacional, provincial, municipal o de unidad de salud.

SI-Salud está compuesto por el Registro Informatizado de Salud (RIS), el Sistema Informatizado de Atención Primaria (SIAP) y el Sistema de Gestión Hospitalaria (SIGH). El RIS está formado por los componentes que integran el Registro no Médico de Información de Salud y el Registro Médico Informatizado de Salud. El SIAP contempla los módulos que son propios de APS, tales como el Registro de Actividades Diarias del Equipo Básico de Salud (EBS) y el Registro de Población que permitirán la transformación de los servicios que se brindan en este nivel de atención. El SIGH agrupa los módulos que pertenecen al nivel de Atención Secundaria tales como el registro de Autopsias.<sup>9</sup>

---

<sup>9</sup> Ídem Referencia 6

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

Es la solución informática integral para la Salud Pública, acorde con los objetivos de la informatización de la sociedad cubana. Constituido por un conjunto de aplicaciones independientes (módulos del sistema) que se interconectan según las necesidades del flujo de información. Es además la herramienta que permite a los usuarios autorizados combinar la información de los diferentes módulos que lo componen, para obtener una información integral en tiempo real para la toma de decisiones en los diferentes niveles de dirección, la docencia, investigación y la gestión en salud.

El RIS, es un sistema formado por componentes, desarrollados con un nivel de cohesión y acoplamiento que le permiten ser capaces de interactuar entre ellos y de esta forma reutilizar la información gestionada por cada componente. El Registro de Áreas de Salud es uno de los módulos del RIS.<sup>10</sup>

### **1.2 Conceptos asociados al dominio del problema**

#### **Áreas de Salud**

Las Áreas de Salud en Cuba están constituidas por el área geográfica a la que presta sus servicios una unidad de salud que contemple el Programa de Trabajo del Médico y la Enfermera de la Familia. En estas unidades (pueden ser policlínicos u hospitales rurales) se atiende a la mayoría de la población en el nivel de atención primaria.

#### **Dirección Nacional de Estadísticas**

La Dirección Nacional de Estadística, es la unidad organizativa del Ministerio de Salud Pública de Cuba, que metodológicamente establece la captación, procesamiento, publicación y diseminación de la información estadística en materia de salud. Consta para ello de una estructura con representación en todas y cada una de las unidades del Sistema Nacional de Salud. Su visión es tomar decisiones correctas basadas en información confiable y su misión es proveer información estadística oportuna para guiar las acciones en los propósitos de mejoría de la salud de la población cubana.

#### **Estadísticas de la Salud**

En salud pública se emplean, consciente o no, muchos conceptos estadísticos al adoptar decisiones relativas a diagnósticos clínicos, o bien al predecir probables resultados de un programa de

---

<sup>10</sup> Ídem Referencia 6

intervención en la población. Y considerando que la estadística es una excelente base para comprender muchos fenómenos reales y para orientar la resolución de problemas relativos a estos, es importante poder definirla mediante el conocimiento de diferentes autores:

Ligia Moya (1989)

"...la rama del saber que trata del desarrollo y aplicación de métodos eficientes de recolección, elaboración, presentación, análisis e interpretación de datos numéricos".

Lwanga y Tye (1987)

"...disciplina que comprende los métodos estadísticos y el estudio de métodos científicos de acopio, tratamiento, reducción, presentación, análisis e interpretación de datos y de hacer deducciones y sacar conclusiones de datos numéricos".

Spiegel (1992)

"...estudia los métodos científicos para recoger, organizar, resumir y analizar datos, así como para sacar conclusiones válidas y tomar decisiones razonables basadas en tal análisis".

Daniel (2002)

"...es un campo del estudio relacionado con la recopilación, organización y resumen de datos y la obtención de inferencias acerca de un conjunto de datos cuando sólo se observa una parte de ellos".

En la práctica, al margen de su definición, la estadística suele variar su nombre, particularmente debiéndose al campo o área de aplicación. De tal manera que aplicada al campo de la salud, se denominaría *estadística de salud*, en el área de estudio y caracterización de la población humana, se le llama *estadística demográfica*, en su parte social como *estadística social* y en las ciencias biológicas, *bioestadística*. Y en el ámbito de estudio de la salud pública hace acopio de todos los métodos estadísticos empleados por las diferentes disciplinas que la constituyen como las antes citadas.

Entre los objetivos más importantes relacionados con la estadística y que contribuyen al campo de la salud pública y sectores relacionados tenemos los siguientes:

- Permite comprender los fundamentos racionales en que se basan las decisiones en materia de diagnóstico, pronóstico y terapéutica.

- Interpreta las pruebas de laboratorio y las observaciones y mediciones clínicas con un conocimiento de las variaciones fisiológicas y de las correspondientes al observador y a los instrumentos.
- Proporciona el conocimiento y comprensión de la información acerca de la etiología y el pronóstico de las enfermedades, a fin de asesorar a los pacientes sobre la manera de evitar las enfermedades o limitar sus efectos.
- Otorga un discernimiento de los problemas sanitarios para que eficientemente se apliquen los recursos disponibles para resolverlos.

En adición a los objetivos antes citados, resalta la utilidad de la estadística en el desarrollo del pensamiento crítico, a fin de: (a) pensar críticamente acerca de los problemas de salud; (b) evaluar correctamente los datos disponibles para la toma de decisiones e (c) identificar las decisiones y conclusiones que carecen de base científica y lógica.<sup>11</sup>

### 1.3 Análisis de las soluciones existentes

Cuando se trata de analizar sistemas existentes en el campo de la salud lo primero que hay que tener en cuenta es que: el Sistema Nacional de Salud de cada país tiene sus particularidades, y diferentes políticas en lo que a atención médica se refiere y no presentan una estructura uniforme. En la mayoría, es un negocio más, por lo que casi no existen modelos estándares para la gestión de la información de salud. En los que se utilizan sistemas automatizados, generalmente satisfacen sus necesidades particulares y no es posible adaptarlos a otros países.

A nivel nacional e internacional existen varios sistemas informáticos relacionados con las Áreas de Salud, pero sus funcionalidades casi nunca están dirigidas a brindar reportes estadísticos que permitan consultar la información de estas.

Entre las aplicaciones que registran información con fines estadísticos se encuentra *Gelattina*, la cual brinda la información estadística del Hospital Universitario de la UANL. El módulo de reportes entrega información sumamente importante que permitirá al personal del Hospital Universitario llevar a cabo análisis estadísticos más a fondo sobre los pacientes que están atendiendo y sus padecimientos. Se

---

<sup>11</sup> Ídem Referencia 5

podrán realizar reportes estadísticos por diferentes parámetros (Por edad, sexo, diagnósticos, entre otros).

También se encuentra *DATASUS*, el cual es la vinculación del Sistema Único de salud de Brasil al campo de la informática, por lo que es el Ministerio de Salud en el Área del departamento informático. El *DATASUS* proporciona las informaciones de salud del Brasil, entre las que se encuentran: indicadores de salud, asistencia en salud, redes asistenciales, informaciones epidemiológicas, estadísticas vitales, informaciones demográficas y de salud suplementaria.

La necesidad de desarrollar las Estadísticas Descriptivas del Registro de Áreas de salud surge a partir de que los sistemas anteriormente expuestos no cumplen con los requisitos de integración plasmados en la estrategia de informatización de la salud en Cuba. Estos sistemas, en su mayoría no cumplen con todas las funcionalidades necesarias que permitan la consulta eficiente de la información y que ayude a mejorar los resultados en términos de salud.

### **1.4 Tecnologías, Herramientas y Metodologías utilizadas**

#### **1.4.1 Tecnologías**

##### **Internet**

Nació a finales de los años 70 en los Estados Unidos como un proyecto militar. Internet es tanto un conjunto de comunidades como un conjunto de tecnologías, y su éxito se puede atribuir a la satisfacción de las necesidades básicas de la comunidad y a la utilización de ésta de un modo efectivo para impulsar la infraestructura. Es a la vez una oportunidad de difusión mundial, un mecanismo de propagación de la información y un medio de colaboración e interacción entre los individuos y sus ordenadores, independientemente de su localización geográfica. Su estructura es descentralizada y resistente a los puntos de control. No está controlada por ninguna organización gubernamental y es de Acceso Libre.

##### **Aplicaciones Web**

Una aplicación Web es una aplicación informática que permite a los usuarios acceder a un servidor a través de la red. La popularidad de las aplicaciones Web se debe principalmente a los navegadores como clientes ligeros y que para actualizar y mantener las aplicaciones no es necesario distribuir e instalar software en miles de potenciales clientes. En términos más simples, una Aplicación Web es un



Sistema Web que permite a los usuarios ejecutar lógica de negocio a través de un Navegador o lo que es lo mismo, modificar el estado del negocio. Las aplicaciones Web hacen uso de las tecnologías que existen para permitir a los usuarios del sistema modificar la lógica del negocio en el servidor y para generar contenidos dinámicos, de no existir lógica de negocio en el servidor el sistema es considerado como sitio, no como una aplicación Web.

Las aplicaciones Web generalmente presentan una arquitectura simple, como componentes principales usan el servidor Web, la red y el navegador. El servidor es el encargado de distribuir las páginas por cada petición de cada cliente. Las peticiones se hacen a través de las conexiones de redes y para hacerlo utilizan el protocolo de comunicación http. Para mostrar la información a los usuarios y hacer las validaciones en la entrada de los datos se usa siempre un browser o navegador.

### **El Servidor Web**

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

### **El servidor Web Apache**

El servidor Web Apache es un software de código abierto que funciona sobre cualquier plataforma. Es uno de los servidores Web más potentes y de los más utilizados a nivel mundial. Se encuentra muy por encima de sus más fuertes competidores gratuitos y comerciales. Ofrece una fuerte estabilidad y sencillez. Soporta varios protocolos de comunicación. Su distribución se encuentra en todas las versiones de Linux. Apache está capacitado tanto para prestar servicios de páginas estáticas como de páginas dinámicas, su configuración se hace mediante la administración de un único archivo, permite el trabajo con las bases de datos, con los ficheros y posee Virtual Host.

El servidor Web Apache se distribuye como software libre, porque para el grupo que lo desarrolla, ese es su deseo, ellos presentan el criterio de que los productos semejantes al Apache tienen que ser accesibles para todo el mundo, y que las compañías que quieran ganar dinero deben hacerlo

mediante los soportes técnicos y la prestación de los servicios, no apoderándose de los productos, además si permiten que el software sea retroalimentado es porque creen que otras personas también tienen buenas ideas que pueden aportar y que pueden convertir al Apache en lo que actualmente es, un servidor Web potente, ágil y eficiente.

### **El Navegador Web o Browser**

Un navegador web o explorador web (del inglés, navigator o browser) es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet. Esta red de documentos es denominada World Wide Web (WWW). Cualquier navegador actual permite mostrar o ejecutar gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web). Tales documentos, comúnmente denominados páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen.

El seguimiento de enlaces de una página a otra, ubicada en cualquier computadora conectada a la Internet, se llama navegación; que es de donde se origina el nombre de navegador.

### **HyperText Transfer Protocol (HTTP)**

HTTP es un protocolo para el desarrollo de sistemas de información multimedia distribuidos, no orientado a estado, puede ser usado con varios fines, no sólo para transferir ficheros HTML. Se caracteriza por poseer un esquema de direccionamiento comprensible que utiliza Universal Resource Identifier (URI) para localizar sitios (URL) o nombres (URN) sobre los que hay que aplicar un método.

Por defecto, HTTP utiliza el puerto 80, aunque también puede trabajar con otros. Este protocolo no presenta conexión ni estado. Una vez que el servidor da respuesta a la petición del cliente se rompe la conexión entre ambos, no se guarda un historial de la conexión. HTTP acepta cualquier tipo de datos nuevos, utiliza Multipart Internet Mail Extensión para determinar el tipo de datos a transportar. El

servidor HTTP incluye una cabecera que indica al cliente los tipos de datos que presenta el documento, lo que hace posible que los navegadores muestren videos e imágenes entre otros.

### **Servicio Web**

Un **servicio web** (*Web service*) es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. **(Ver Anexo II)**

### **Patrones de Diseño e Implementación**

Los diseñadores expertos en orientación a objetos (y también otros diseñadores de software) van formando un amplio repertorio de los principios generales y de expresiones que los guían en la confección de un software. A unos y a otras se le puede asignar el nombre de patrones, si se codifican en un formato estructurado que describe el problema y la solución, y si se les asigna un nombre.

En la terminología de objetos, el patrón es una descripción de un problema y su solución, recibe un nombre y puede emplearse en otros contextos; en teoría indica la manera de utilizarlo en circunstancias diversas.

Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería de software. Todo lo contrario: intentan codificar el conocimiento, las expresiones y los principios ya existentes: cuanto más trillados y más generalizados, mejor.

Los podemos separar, aunque están íntimamente ligados, de hecho si se esfuerza en aumentar mucho la cohesión del sistema o software, se tiene un alto acoplamiento entre las clases, y por el contrario si se reduce mucho el acoplamiento, se verá mermada la cohesión:

**Alta cohesión:** La información que almacena una clase debe de ser coherente y estar en la mayor medida posible relacionada con la clase.

**Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí posible. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

### **Modelo Vista Controlador**

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

Los elementos de este patrón son:

**Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.

**Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

**Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

### **Arquitectura en 3 capas**

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. Una de las ventajas principales de la arquitectura en capas es que el desarrollo e implementación se puede realizar por capas y un cambio en una capa, solamente tendrá cambios dentro de la misma y no habrá que cambiar otras. **(Ver Anexo III)**

Esta arquitectura permite distribuir la implementación por capas, de forma tal que se puede dividir el grupo de desarrollo en equipos de trabajo por cada una de las capa que tenga el sistema, de esta forma cada equipo de desarrollo está abstraído totalmente del trabajo de los otros equipos, solamente

le interesaría conocer elementos puntuales que existen entre cada una de las capas. El diseño que más se utiliza es el de la arquitectura en tres capas (presentación, negocio, acceso a datos) donde a cada capa se le asigna una misión simple, que permite que el diseño de la arquitectura sea escalable y que pueda ampliarse con facilidad si es necesario.

**Capa de presentación:** es la que ve el usuario (hay quien la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable (entendible y fácil de usar) para el usuario.

**Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

**Capa de datos:** es donde residen los datos y es la encargada de acceder a los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

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

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

Esta arquitectura habla de cómo construir aplicaciones complejas a través de ensamblar módulos, o componentes, diseñados previamente con el fin de ser reutilizadas. Se hace énfasis en esta arquitectura en la estandarización de las interfaces de comunicación de los componentes. Estos módulos se diseñan desde un comienzo para integrarse en una gran variedad de configuraciones.

Una de las ventajas de los componentes es la reusabilidad, aunque depende en buen grado de la correcta identificación de los componentes, así como del modo en que se organicen estos. El Desarrollo Basado en Componentes (DBC) brinda el soporte para la integración de las partes en sistemas mayores, por lo cual se requiere de mucho cuidado a la hora del modelado de la arquitectura con fines de asegurar la compatibilidad entre los componentes que interactúan.<sup>12</sup>

### **Arquitectura Orientada a Servicios (SOA)**

La Arquitectura Orientada a Servicios (en inglés Service-Oriented Architecture o SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario.

SOA es una arquitectura de software que permite la creación y/o cambios de los procesos de negocio desde la perspectiva de TI de forma ágil, a través de la composición de nuevos procesos utilizando las funcionalidades de negocio que están contenidas en la infraestructura de aplicaciones actuales o futuras (expuestas bajo la forma de webservices).

SOA define las siguientes capas de software:

- aplicativa básica, sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad;
- de exposición de funcionalidades, donde las funcionalidades de la capa aplicativas son expuestas en forma de servicios (webservices);
- de integración de servicios, facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración;
- de composición de procesos, que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio;
- de entrega, donde los servicios son desplegados a los usuarios finales.

Los beneficios que puede obtener una compañía que adopte SOA son:

- Mejora en los tiempos de realización de cambios en procesos.
- Facilidad para evolucionar a modelos de negocios basados en tercerización.

---

<sup>12</sup> Ídem Referencia 7

- Facilidad para abordar modelos de negocios basados en colaboración con otros entes (socios, proveedores).
- Poder para reemplazar elementos de la capa applicativa SOA sin interrupción en el proceso de negocio.

SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación.<sup>13</sup>

### **Sistema operativo GNU/Linux**

Linux, más correctamente llamando GNU/Linux, es un sistema operativo que poco a poco, gracias a su robustez y coste gratuito, va ganando más presencia en entornos como el del ordenador personal, dejando así de ser considerado como un sistema operativo adecuado sólo para servidores. El sistema operativo es el conjunto de programas básicos y utilidades que hacen que funcione su computadora. Debian utiliza el núcleo Linux (el corazón del sistema operativo), pero la mayor parte de las herramientas básicas vienen del Proyecto GNU; de ahí el nombre GNU/Linux. Es multitarea, multiusuario, multiplataforma y multiprocesador; en las plataformas Intel corre en modo protegido; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria; usa un sistema de memoria virtual por páginas; utiliza toda la memoria libre para cache; permite usar bibliotecas enlazadas tanto estática como dinámicamente; se distribuye con código fuente; usa hasta 64 consolas virtuales; tiene un sistema de archivos avanzado pero puede usar los de los otros sistemas; soporta redes tanto en TCP/IP como en otros protocolos.

### **Plataforma de servicios PLASER**

PlaSer, acrónimo de Plataforma de Servicio, constituye una plataforma sobre la que se pueden desplegar aplicaciones XML - Web Services. Este sistema está concebido completamente sobre Arquitectura Basada en Componentes y Orientada a Servicios, usando el paradigma de XML Web Services, específicamente SOAP. En su concepción se han utilizado estándares actuales y normas abiertas. Todo el código ha sido programado en PHP. Desde el punto de vista estructural permite trabajar con cualquier base de datos que cumpla con el estándar SQL-92; pero desde el punto de vista de implementación sólo trabaja con las bases de datos soportadas por el componente DBX, ya que encapsula a dicho componente y lo utiliza para el acceso a bases de datos. El uso de PlaSer, por parte

---

<sup>13</sup> Wikipedia. [cited; Available from: [http://es.wikipedia.org/wiki/Arquitectura\\_orientada\\_a\\_servicios](http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios)

de los desarrolladores, asegura varias ventajas entre las que pudieran destacarse que el programador no tiene que preocuparse por implementar la seguridad del Sistema, ya que esta es una de las tareas que asume PlaSer, además facilita la programación y homogeneidad de los componentes.

PlaSer está integrado por una colección de clases y una "estructura de directorios" (layout) en la capa de presentación. También constituye una capa abstracta de transporte para la interoperabilidad entre los componentes. PlaSer además implementa y maneja la seguridad.

PlaSer, en lo fundamental, está integrado por varias clases desarrolladas en PHP, una librería, que puede o no ser usada para que un componente se integre, pero que de no ser usada la seguridad corre por parte del programador. PlaSer, en su configuración ideal, se distribuye en tres servidores, los cuales se encuentran conectados en cascada y solo el primero está conectado a Internet con una IP real. Este primero contiene la capa de presentación (layout), el segundo contiene a ProxPla y los demás componentes, incluido el SAAA (Single Authorization Authentication and Account) componente de seguridad, que autentifica y autoriza, además tiene registrados los nombres de los métodos. Y el tercer servidor contiene las bases de datos desarrolladas en MySQL. Para aumentar la seguridad se pudiera colocar a ProxPla solo en un cuarto servidor separado de los demás componentes.<sup>14</sup>

### **El modelo cliente – servidor**

Esta arquitectura consiste básicamente en que un programa -el cliente- realiza peticiones a otro programa -el servidor- que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. **(Ver Anexo IV)**

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

---

<sup>14</sup> León Garcés, J. 2008. Descripción de la Plataforma de Servicios PlaSer.



Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La **arquitectura cliente-servidor** sustituye a la arquitectura **monolítica** en la que no hay distribución, tanto a nivel físico como a nivel lógico.

### 1.4.2 Metodologías de Desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Definen Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Si no se emplea una metodología pueden surgir los siguientes problemas:

- Resultados impredecibles.
- Detección tardía de errores.
- La introducción de nuevas herramientas afectará perjudicialmente al proceso.
- Cambios de organización también afectarán al proceso.
- Resultados distintos con nuevas clases de productos.

### UML

UML, cuyas siglas en inglés significan Unified Modeling Language, es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Se utiliza para modelar sistemas orientados a objetos e incluye un conjunto de diagramas y notaciones estándar y su significado. UML cuenta con varios tipos de diagramas para definir sistemas de software y hardware, detallar artefactos en los sistemas y documentar la programación. Puede ser utilizado en variadas metodologías de software aunque se utilizará específicamente en el Proceso Unificado de Desarrollo (RUP) en el actual proyecto.<sup>15</sup>

---

<sup>15</sup> Ídem Referencia 7

### RUP

El Rational Unified Process (RUP) es un proceso de Ingeniería de Software que mejora la productividad del equipo de trabajo y entrega las mejores prácticas del software a todos los miembros del mismo. Proporciona una guía específica en áreas tales como la de Modelado de Negocios, Arquitecturas Web, Pruebas y Calidad.

Basado en las mejores prácticas adoptadas en miles de proyectos por todo el mundo, utiliza como lenguaje de modelación el anteriormente expuesto UML. Contiene patrones que permiten a los gestores de proyectos añadir o quitar rápidamente piezas que resuelvan problemas comunes y enfocarlas en el proyecto específico. Permite manejar contenidos en diferentes dominios como el modelado de bases de datos o control de requisitos.

Los casos de uso, o sea, lo que los usuarios futuros necesitan o desean, guían el proceso de desarrollo. Los modelos que se obtienen como resultado de los diferentes flujos de trabajo representan la realización de un Caso de Uso (CU).

La arquitectura muestra la visión común del sistema, describe los elementos del modelo que son más importantes para su construcción. El RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.

RUP propone que cada fase del proyecto se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo aunque se enfoca en unos más que en otros. **(Ver Anexo V)**

### 1.4.3 Lenguajes de Programación Web

Los lenguajes de programación Web se clasifican en dos partes fundamentales, los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor se pueden presentar algunos significativos como PHP, ASP.NET, PERL, Java, JSP, los módulos CGIs e ISAPIs entre otros. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la Información, etc. Del lado del cliente se encuentran principalmente el Java

Script y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores.

Esta distinción en los lenguajes ha sido necesaria debido a que la Web funciona en modo “Desconectado”, o sea, un usuario a través de un navegador hace una petición de una página Web a un Servidor Web (Request); el Servidor recepciona la petición, la procesa y le envía la Respuesta al Cliente (Response), este realiza la recepción y se desconecta.<sup>16</sup>

### **PHP**

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes. A diferencia de Java o JavaScript que se ejecutan en el navegador PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML (Wap). Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP. Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.

---

<sup>16</sup> Ídem Referencia 7

- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Un sitio con páginas webs dinámicas es aquel que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para propósitos particulares.<sup>17</sup>

### **Java Script**

Java Script es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM. Muchas de las funciones que tienen las aplicaciones Web se basan en Java Script. Todos los navegadores actuales para la versión 3 o superior lo admiten sin problemas. Las funciones que pueden hacerse con este lenguaje hacen más rica la navegación entre las páginas Web. En Java Script se puede trabajar con los eventos onclick, onmouseover, entre otros, que permiten interactuar con el menú, cambiar imágenes dentro de la propia página. Se pueden crear menús desplegables, herramientas de selección de color, alarmas, generar mensajes, calendarios, relojes, juegos sencillos para jugar online, efectos de imágenes y audios y muchos más.

### **XML**

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero

---

<sup>17</sup> Rodas Hinostraza, R. [cited; Available from:  
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>

su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

Entre las tecnologías XML disponibles se pueden destacar:

XSL: Lenguaje Extensible de Hojas de Estilo, cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación como puede ser una ventana de un navegador Web o un dispositivo móvil, o un conjunto de páginas de un catálogo, informe o libro.

XPath: Lenguaje de Rutas XML, es un lenguaje para acceder a partes de un documento XML.

XLink: Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos XML.

XPointer: Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.

XQL: Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web.

### **XSLT**

XSLT o Transformaciones XSL es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las hojas de estilo XSLT (aunque el término de hojas de estilo no se aplica sobre la función directa del XSLT) realizan la transformación del documento utilizando una o varias reglas de plantilla: unidas al documento fuente a transformar, esas reglas de plantilla alimentan a un procesador de XSLT, el cual realiza las transformaciones deseadas colocando el resultado en un archivo de salida o, como en el caso de una página web, directamente en un dispositivo de presentación, como el monitor de un usuario.

Actualmente, XSLT es muy usado en la edición web, generando páginas HTML o XHTML. La unión de XML y XSLT permite separar contenido y presentación, aumentando así la productividad.

### Fundamentación del lenguaje a utilizar

Hasta el momento se han analizado las características fundamentales de los lenguajes de programación candidatos para la implementación de la propuesta de este trabajo, para fundamentar la elección se hará una comparación teniendo en cuenta algunas características que influyen directamente en el ambiente de trabajo donde se va a desarrollar la propuesta. En cuanto a:

- Características multiplataforma: Menos el ASP, que es solamente soportado por la plataforma Windows, los demás lenguajes están soportados en múltiples plataformas.
- Velocidad de ejecución: La velocidad es mayor en PHP, seguidos por PERL y JSP.
- Disponibilidad de recursos: Actualmente los más utilizados en Internet son PHP y JSP, siendo más utilizado en la publicación de artículos y códigos de ejemplos. PHP tiene una de las comunidades más grandes en Internet, al igual que la de Java.
- Familiaridad con el lenguaje: En la universidad los lenguajes más utilizados por los programadores son el ASP y el PHP.

De acuerdo a estas comparaciones, PHP resulta mucho más favorecido, por tanto se piensa que es el adecuado para implementar la propuesta de sistema de este trabajo, además y principalmente por ser el lenguaje solicitado por el Cliente. También se utiliza el lenguaje XSLT para transformar los documentos XML en páginas HTML y JavaScript para aportar dinamismo a las páginas HTML.<sup>18</sup>

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

Los Sistemas de Gestión de Base de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa. El propósito general de los Sistemas de Gestión de Base de Datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos.

Algunas de las ventajas de los SGBD son:

---

<sup>18</sup> Ídem Referencia 7

- Facilidad de manejo de grandes volúmenes de información.
- Gran velocidad en muy poco tiempo.
- Independencia del tratamiento de información.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- Integridad referencial al terminar los registros.

### **MySQL**

MySQL es un sistema de gestión de bases de datos relacionales que se encuentra entre los más populares. Aunque no es el único, se considera que es el perfecto compañero de PHP para generar sitios dinámicos con grandes volúmenes de información y acceso a bases de datos. El servidor MySQL fue desarrollado con el objetivo de manipular grandes bases de datos, mucho más rápido que las soluciones existentes. Se ha usado exitosamente en ambientes de desarrollo sumamente exigentes, por varios años, aunque se encuentra en desarrollo constante. Ofrece un conjunto rico de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL un servidor apropiado para bases de datos en Internet. Una de las razones para el rápido crecimiento de la popularidad de MySQL, es que se trata de un producto Open Source. Esto quiere decir que se puede descargar de Internet y usarlo sin pagar nada. Se puede estudiar su código fuente y cambiarlo de acuerdo a sus necesidades. MySQL usa GPL (la Licencia Pública General GNU), que define que es lo que se puede y lo que no se puede hacer con el software para diferentes situaciones. Si uno no está de acuerdo con una la licencia GPL o tiene la necesidad de modificar el código de MySQL en una aplicación, es posible comprar una versión de MySQL con la licencia comercial.

Características:

Interioridades y portabilidad:

- Escrito en C y en C++.
- Probado con un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas.

### Seguridad:

- Es un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

### Escalabilidad y límites:

- Soporte a grandes bases de datos. Se usa MySQL Server con bases de datos que contienen 50 millones de registros. También se tienen usuarios que usan MySQL Server con 60.000 tablas y acerca de 5.000.000 de registros.

### Conectividad:

- Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP, o 2003), los clientes pueden usar *named pipes* para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros *socket Unix*.

### Localización:

- El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.

### Cientes y herramientas:

- MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente mysqlcheck. MySQL.

## SQL

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos. Es un lenguaje declarativo de alto nivel o de no procedimiento, que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación. De esta forma una sola sentencia puede equivaler a uno o más programas que utilicen un lenguaje de bajo nivel orientado a registro. Es un lenguaje de cuarta generación (4GL). El SQL proporciona una rica funcionalidad más allá de la simple consulta (o recuperación) de datos. Asume el papel de lenguaje de definición de datos (LDD), lenguaje de definición de vistas (LDV) y lenguaje de manipulación de datos (LMD). Además permite la concesión y denegación de permisos, la implementación de restricciones de integridad y controles de transacción, y la alteración de esquemas. Las primeras versiones del SQL incluían funciones propias de lenguaje de definición de



almacenamiento (LDA), pero fueron suprimidas en los estándares más recientes con el fin de mantener el lenguaje sólo a nivel conceptual y externo.

El SQL permite fundamentalmente dos modos de uso:

- Un uso interactivo, destinado principalmente a los usuarios finales avanzados u ocasionales, en el que las diversas sentencias SQL se escriben y ejecutan en línea de comandos, o un entorno semejante.
- Un uso integrado, destinado al uso por parte de los programadores dentro de programas escritos en cualquier lenguaje de programación anfitrión. En este caso el SQL asume el papel de sub-lenguaje de datos.

### **Fundamentación del Sistema Gestor de Bases de Datos a utilizar**

Hasta el momento se han planteado algunas de las características de los dos Sistemas Gestores de Base de Datos (SGBD) más usados en el mundo y en la Universidad, a continuación se fundamenta la selección del gestor a utilizar:

En concreto, usar MySQL tiene ventajas adicionales:

- Escalabilidad: Es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- MySQL está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes.
- Conectividad: Es decir, permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es corriente que servidores Linux o Unix, usando MySQL, sirvan datos para ordenadores con Windows, Linux, Solaris, etc. Para ello se usa TCP/IP, tuberías, o sockets Unix.
- Es multi-hilo, con lo que puede beneficiarse de sistemas multiprocesador.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable.
- FREE: MySQL es libre, o sea no hay que pagar por su utilización.

De acuerdo a estas ventajas, MySQL resulta ser el propuesto para implementar el sistema además de ser el gestor solicitado por el Cliente.

### 1.4.5 Herramientas a utilizar

#### Macromedia Dreamweaver 8

Crea aplicaciones y sitios web profesionales. Macromedia Dreamweaver es un excelente programa destinado a la edición de aplicaciones y sitios web con amplio soporte de CSS. Esta versión ofrece una integración más estrecha con otras herramientas de Macromedia, y una integración mejorada con Studio. Soporta elementos de Flash, elementos HTML de MX y posee un entorno racionalizado de diseño y desarrollo. Incluye un editor de gráficos incorporado, soporte único para el desarrollo con ColdFusion, ASP.NET, JSP, PHP, XML y servicios web. Macromedia Dreamweaver incluye una función de validación en distintos navegadores con la que podrás verificar automáticamente las etiquetas y reglas CSS para comprobar su compatibilidad con los navegadores más importantes. Además podrás codificar completamente las transferencias de archivos y evitar el acceso no autorizado a los datos y archivos.<sup>19</sup>

#### Zend Estudio

Es el único Ambiente de Desarrollo Integrado (IDE) disponible para diseñadores profesionales que abarcan todos los componentes de desarrollo necesarios para el PHP, se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. La parte del programa que nos permite escribir los scripts es bastante útil para la programación en PHP. La interfaz está compuesta por varias partes, en las que encontramos un explorador de archivos, una ventana de depuración, los menús y otra para mostrar el código de las páginas.<sup>20</sup>

#### MySQL Manager Professional

Herramienta avanzada del GUI para la administración y el desarrollo del servidor de MySQL. De gran alcance para la administración y el desarrollo del servidor de la base de datos de MySQL. El encargado de MySQL trabaja con cualquier versión de MySQL a partir del 3.2 a 5 y ayudas, todos de las últimas características de MySQL incluyendo las opiniones, los procedimientos almacenados y las

<sup>19</sup> *Macromedia Dreamweaver 8*, [cited; Available from: <http://www.zeroprogramas.com/programas/macromedia-dreamweaver-8.asp>

<sup>20</sup> Angel Alvarez, M., *Evaluando Zend Studio*. 2003, [cited; Available from: <http://www.maestrosdelweb.com/editorial/zendstudio>.

funciones, llaves extranjeras de InnoDB, funciones usuario-definibles, tablas transacción-seguras, nuevo protocolo de la autenticación de MySQL 5, MySQL 4. Extensiones del privilegio del usuario x, charsets y ayuda de las colaciones para las bases de datos y las tablas. Ofrece varias herramientas de gran alcance para los usuarios experimentados tales como diseñador visual de la base de datos, constructor visual de la pregunta, y GOTA viewer/editor para satisfacer todas sus necesidades.<sup>21</sup>

### **Stylus Studio**

Stylus Studio 2007 XML Enterprise Suite esta compuesto de una multitud de herramientas XML todas en un solo paquete editor XML. Construye aplicaciones Java utilizando el editor JSP de Stylus Studio 2007 XML Enterprise Suite, XML-Java debugger, y otras aplicaciones XML Java. El editor también provee soporte en el desarrollo de XSLT incluyendo un robusto editor XSLT. El editor XSLT incluye debugger, mapeo XSLT, perfilaje XSLT, diseño con estilos visuales desde HTML a XSLT, XSL: FO. Stylus Studio 2007 XML Enterprise Suite también incluye numerosas sincronizaciones, vistas de edición de XML (visores XML), Editor Inteligente XML, y se integra un compilador XML y un validador XML, diferenciación XML y más.<sup>22</sup>

### **Rational Rose Enterprise Edition**

Rational Rose Enterprise Edition es el producto más completo de la familia Rational Rose. Todos los productos Rational Rose incluyen soporte Unified Modeling Language™ (UML™). Rational Rose Enterprise Edition es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++® y Visual Basic®. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Tiene como características:

- Mantiene la consistencia de los modelos del sistema software.
- Chequeo de la sintaxis UML.
- Generación de Documentación automáticamente.

<sup>21</sup> *EMS MySQL Manager Professional*, [cited; Available from: <http://www.fileheaven.com/descargar/ems-mysql-manager-professional-for-windows/35281.htm>

<sup>22</sup> *Descripción de Stylus Studio 2007*, [cited; Available from: <http://www.boxdownload.com/fichas/stylus-studio-2007-xml-enterprise-suite-build-735d.asp>

- Generación de Código a partir de los Modelos.
- Ingeniería Inversa (crear modelo a partir código).

### **1.5 Conclusiones**

En este capítulo se profundizó en el conocimiento del Sistema Nacional de Salud y las Áreas de Salud. Se realizó un análisis de las tecnologías a utilizar en el desarrollo del sistema propuesto y se fundamentaron las elecciones del lenguaje, el sistema gestor de bases de datos, y la metodología a utilizar.

Con la introducción de Internet, se han abierto posibilidades de acceso a la información desde cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas. Existen herramientas potentes para realizar esto, ya que han surgido nuevas tecnologías que permiten que el acceso a una base de datos desde una aplicación Web.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### 2.1 Introducción

En este capítulo se abordan diferentes elementos relacionados con el objeto de estudio brindando la base fundamental para el desarrollo de las Estadísticas Descriptivas del RAS, quedando expuestos los diferentes aspectos que conllevan a la descripción de la propuesta de solución para dichas funcionalidades. Se hace énfasis en las características del sistema, para ello se modelan los flujos de trabajo Modelamiento del Negocio y Gestión de Requerimientos. Se modelan los artefactos propios de estos flujos. Se hace la realización del Modelo del Negocio, cuyo propósito es lograr una mejor comprensión del sistema que el software tiene que resolver. Se determina desarrollar un Modelo de Dominio, además se enumeran los requerimientos funcionales y no funcionales y como quedan agrupados los primeros en Casos de Uso, con el fin de estructurar el Diagrama de Casos de Uso del Sistema.

### 2.2 Situación Problemática

Las Áreas de Salud en Cuba son áreas geográficas delimitadas en kilómetros cuadrados, donde sus habitantes son beneficiados con los servicios médicos que se brindan en una unidad de salud que contemple el Programa del Médico y la Enfermera de la Familia. Estas unidades de salud pueden ser hospitales rurales o policlínicos y responden a la mayoría de los problemas de salud de la población en el nivel de la Atención Primaria. En las Áreas de Salud operan los Grupos Básicos de Trabajo y los Equipos Básicos de Salud, que constituyen estructuras organizativas fundamentales para la prevención y el control de los problemas de salud de la población en el Sistema Nacional de Salud.

En la actualidad la consulta de los datos que se gestionan en las Áreas de Salud no es efectiva. Existen dificultades con las acciones a desarrollar, relacionadas con la transmisión de la información generada por los Grupos Básicos de Trabajo y los Equipos Básicos de Salud. La información que generan estas estructuras organizativas, se hace mediante el uso de métodos tradicionales como llamadas telefónicas, correos convencionales y en la mayoría de los casos mediante papeles, que provocan demoras en el procesamiento de los datos.

Mucha de la información que se recibe a nivel nacional sobre las acciones que se desarrollan en las Áreas de Salud, tiende a duplicarse, se emplea gran cantidad de tiempo en la elaboración de documentos que muestran datos estadísticos, se pierde el criterio favorable de la información ofrecida, por la no actualización de la misma con respecto a su fecha de emisión. En muchas ocasiones es necesario imprimir nuevamente los datos que llegan desde las provincias para poder captarlos.

Parte de la información que se brinda en las unidades de salud se realiza de forma local y manual, esto provoca pérdida en la precisión cualitativa de los datos, mediante el flujo a través de los diferentes niveles de dirección. Solamente algunos de los datos, de carácter cuantitativo, se recogen en algunos modelos estadísticos que pertenecen al Sistema de Información Estadístico Complementaria (SIE-C) del Ministerio de Salud Pública. La descentralización de la información provoca dificultades con el transporte y la periodicidad de los datos.

Existe el Sistema de Información para la Salud (SISalud), que constituye una de las estrategias a desarrollar para la informatización del Sistema Nacional de Salud. Entre las partes que componen este sistema se encuentra el Registro Informatizado de Salud (RIS). Uno de sus componentes es el Registro de Unidades de Salud, el cual gestiona la información de las Unidades de Salud y define cuándo una unidad es un Área de Salud, pero no gestiona el resto de la información de las unidades como Áreas de Salud. El registro de Áreas de Salud otro componente del RIS, gestiona la información relacionada con las Áreas de Salud, pero no cuenta con reportes que permitan consultar la misma en los diferentes niveles de dirección de salud.

Por todas estas razones planteadas anteriormente se requiere de funcionalidades que permitan brindar los datos de las Áreas de Salud a todos los niveles de dirección.

### **2.3 Objetos de automatización**

Analizando los problemas existentes y luego de un análisis profundo del alcance de la investigación, se propone el desarrollo de una serie de funcionalidades en el Registro de Áreas de Salud que sean capaces de facilitar, de forma automática, la obtención de los diferentes reportes estadísticos solicitados por los Departamentos de Estadísticas en los diferentes niveles de dirección del SNS, con el objetivo de consultar de forma eficiente la información de las áreas, permitiendo un mejor control de las mismas; lo cual repercutirá en una mejor toma de decisiones, un buen funcionamiento de las áreas

y un mayor servicio a la población, además se podrán imprimir los resultados en formato Adobe Acrobat (PDF) ó Microsoft Excel (XLS).

### 2.4 Modelo de Dominio

Un Modelo de Dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades, el Modelo de Dominio podría considerarse como un diccionario visual de las abstracciones relevantes, vocabulario del dominio e información del dominio. Aprovechando las bondades de los diagramas UML para representar cosas y conceptos el diagrama del modelo del dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema analizado.

En el desarrollo de las Estadísticas Descriptivas del Registro de Áreas de Salud, se decide realizar Modelo de Dominio debido a que no pueden ser identificados los procesos del negocio asociados al contexto del sistema, como consecuencia de la poca estructuración y fronteras de los mismos. Además debido a que los conceptos que serán expuestos representan objetos del mundo real a los cuales el sistema debe darle un estricto seguimiento.

#### 2.4.1 Conceptos Fundamentales

Para una mejor comprensión del Diagrama del Modelo de Dominio estructurado en el siguiente epígrafe, a continuación se proporciona un marco conceptual con las definiciones identificadas en el contexto del proceso de desarrollo de las Estadísticas Descriptivas del Registro de Áreas de Salud, estas son:

**Área de Salud:** Área geográfica a la que presta sus servicios una Unidad de salud que contemple el Programa de Trabajo del Médico y la Enfermera de la Familia. Tienen el nombre del Policlínico.

**Centro:** Lugar en que se desarrolla la actividad social fundamental u ocupación (retribuida o no) de un individuo. En términos del Registro de Áreas de Salud se refiere a la clasificación de los tipos de poblaciones, limitados a estos lugares, que reciben los servicios del Equipo Básico de Salud. Se clasifica en Centro de Trabajo o Laboral y Centro Estudiantil.

**Departamento:** Se aplica a cada una de las divisiones internas de una Unidad de Salud clasificadas según preste, o no, servicio directo a la población. Por ejemplo: Departamento Directo es aquel que presta servicio directo a la población como los departamentos de consultas, información, etc. Un Departamento Indirecto es aquel que no presta servicio directo a la población como la biblioteca, la cocina, el comedor, mantenimiento, economía, recursos humanos, administración, etc.

**Estadística Descriptiva:** La Estadística Descriptiva es un paso inicial y obligatorio en cualquier tipo de análisis estadístico, analiza metódicamente los datos, simplificándolos y presentándolos en forma clara; eliminando la confusión característica de los datos preliminares. Permite la elaboración de cuadros, gráficos e índices bien calculados; suficientemente claros, como para disipar las dudas y la obscuridad de los datos masivos.

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

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

**Hospital Base:** Unidad de Salud del nivel de Atención Secundaria (Generalmente Hospitales) que se responsabilizan de manera íntegra con las Unidades de Salud de la Atención Primaria (Áreas de Salud) para el enfrentamiento de los problemas de salud de una determinada población que no se pueden manejar en el nivel primario por ser de mayor complejidad en el diagnóstico, el tratamiento y la rehabilitación, (Integración Policlínico Hospital). Existe un Hospital Base por cada especialidad básica: Medicina Interna, Ginecología y Obstetricia, y Pediatría.

**Locales:** Este término ha sido usado en este documento con similar significado a **Consultorio**. Un local es una entidad que puede ser para Consultorio (local de trabajo de los Equipos Básicos de Salud)



o para ser vivienda de los especialistas que integran el Equipo Básico de Salud, es por ello que los locales de Consulta pueden o no tener asociados hasta dos locales de viviendas.

**Plantilla:** Relación ordenada, por categorías o cargos, de los trabajadores de un centro de trabajo, que tiene como fin entre otros, el control del personal y los salarios que devenga cada uno según su cargo y horas trabajadas por mes.

## 2.4.2 Diagrama del Modelo de Dominio

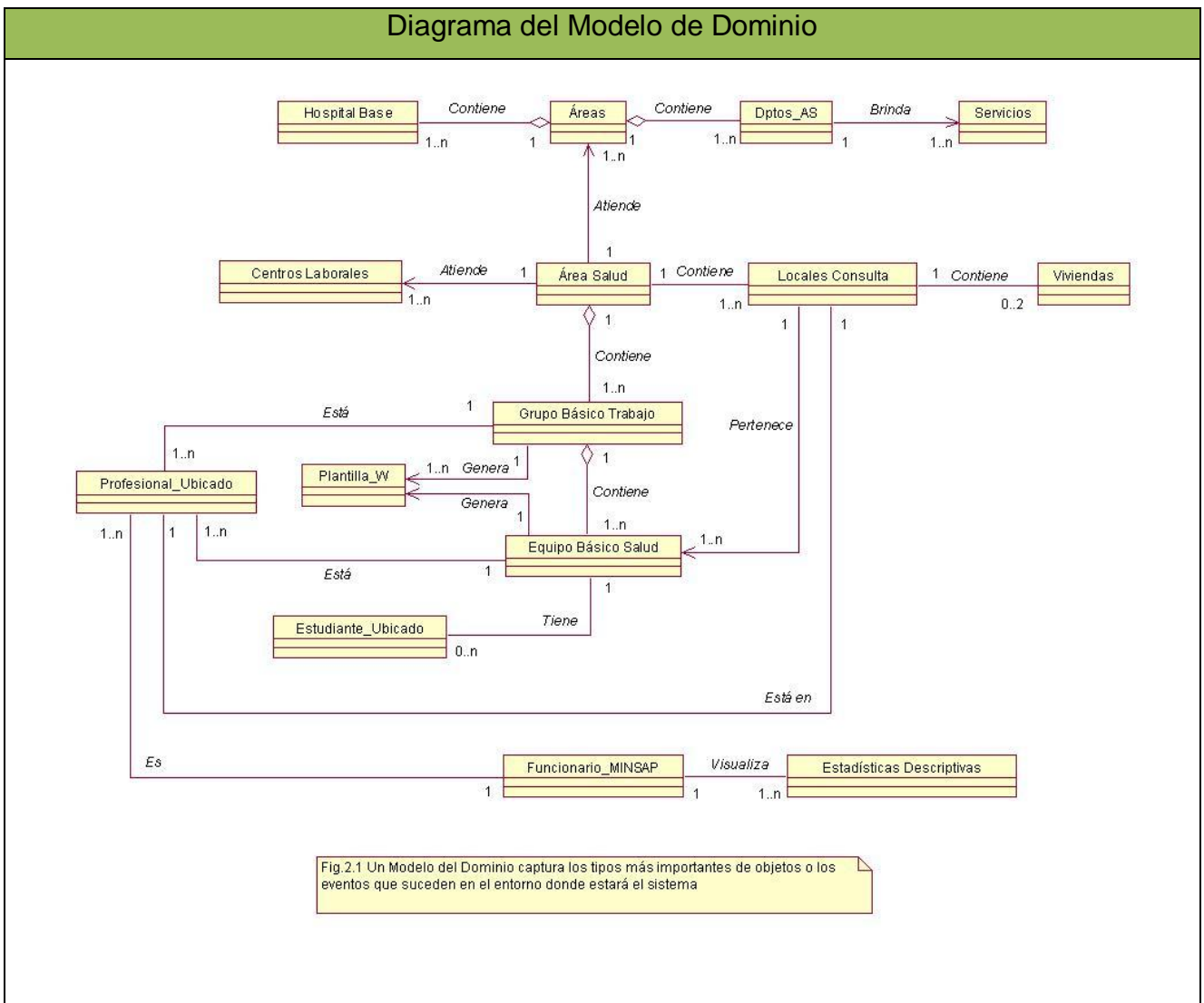


Figura 2.1 Diagrama del Modelo de Dominio.

## 2.5. Propuesta del Sistema

### 2.5.1. Especificación de Requerimientos de Software

Los requerimientos, también conocidos como requisitos, son condiciones o capacidades que tienen que ser alcanzadas o poseídas por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. Los requisitos se pueden clasificar en: funcionales y no funcionales.

#### 2.5.1.1 Requerimientos Funcionales

Los requerimientos representan condiciones, capacidades o cualidades, en dependencia de si son funcionales o no, que se necesitan para resolver un problema o para lograr un objetivo. La especificación de los requerimientos se realiza de forma completa porque al definirlos se busca establecer un común entendimiento sobre los objetivos del negocio propuesto y que reflejan todo lo que el sistema debe hacer o cumplir.

A continuación se muestra el listado de los requerimientos funcionales de las Estadísticas Descriptivas del Registro de Áreas de Salud:

- RF1.** Ofrecer información necesaria sobre los datos generales del Área de Salud.
- RF2.** Ofrecer información consolidada sobre el Área de Salud.
- RF3.** Permitir obtener el total y/o desglose lógico de la información de la estructura del Registro de Áreas de Salud, dada la provincia, municipio y profundidad de la consulta.
- RF4.** Ofrecer información sobre el Equipo Básico de Salud, dada el Área de Salud y el Grupo Básico de Trabajo.

#### 2.5.1.2 Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

### **Usabilidad**

Son aquellos que describen los niveles apropiados de usabilidad dados los usuarios finales del producto, para ello debe revisarse las especificaciones de los perfiles de usuarios y las clasificaciones de sus niveles de experiencia.

### **RNF 1**

Cada usuario que se autentique sólo tendrá acceso a la información que le corresponde, ya sea editor o visualizador a nivel nacional, provincial, municipal o de unidad de salud.

### **Rendimiento**

Son aquellos que especifican los requisitos relacionados con la carga que se espera que tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

### **RNF 2**

El sistema debe soportar la conexión simultánea de más de 60 000 usuarios.

### **RNF 3**

El sistema debe soportar un tiempo de respuesta menor o igual a 30 segundos.

### **RNF 4**

El sistema debe soportar respuestas SOAP que no deben exceder los 50 Kbyte en el tiempo de respuesta al usuario.

### **Soporte**

Son aquellos que abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de este, así como lograr su mejoramiento progresivo y evolución en el tiempo. Pueden incluir: Pruebas, Extensibilidad, Adaptabilidad, Mantenimiento, Compatibilidad, Configuración, Servicios, Instalación e Internacionalización.

### **RNF 5**

El personal que trabaja con el módulo debe contar con el nivel técnico requerido mediante adiestramiento de servicio.

### **Portabilidad**

Son aquellos que especifican los atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos de desarrollo.

### **RNF 6**

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

### **Seguridad**

Son aquellos que provocan los mayores riesgos si no se manejan correctamente. Puede ser tratada por diferentes aspectos:

### **Confidencialidad**

La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

### **RNF 7**

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

### **RNF 8**

La autenticación será la primera acción del usuario en el sistema y consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica.

### **RNF 9**

Si el usuario autenticado no se encuentra registrado se debe reportar un error de acceso.

### **RNF 10**

Si el usuario autenticado se encuentra registrado se autoriza su acceso y se crea un certificado digital.

### **RNF 11**

El Certificado Digital constará de:

- Un identificador único (token) de 32 caracteres generado de manera aleatoria en el momento de la autorización.
- El identificador del usuario.
- El nivel de acceso del usuario (Nacional, Provincial, Municipal o Unidad de Salud).
- El identificador del nivel de acceso.
- Un listado de los módulos que el usuario tiene acceso y el tipo de acceso en cada uno de ellos (Editor o Visualizador).

**RNF 11.1** Deberá registrarse el token, el identificador del usuario presente en el certificado digital y el día, mes, año, hora, minuto, segundo de la creación del certificado.

**RNF 11.2** Para cada petición que se haga al sistema, debe enviarse el token recibido en el certificado digital el cual será validado con la lista de tokens autorizados que el sistema ha registrado.

**RNF 11.3** Si el token se encuentra registrado y la petición se corresponde con los derechos del usuario se procederá a ejecutar la petición solicitada.

**RNF 11.4** Si el token no se encuentra registrado o la petición no se corresponde con los derechos del usuario se debe reportar un error de acceso.

### **RNF 12**

Cada petición de usuario, autorizada o no, será registrada, así como el día, mes, año, hora, minuto, segundo en que se registra y si fue o no autorizada.

### **Integridad**

La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Pueden incluir también mecanismos de chequeo de integridad y realización de auditorías.

### **RNF 13**

Prevenir posibles fallos y recuperarse ante ellos.

### **RNF 14**

Deberán existir mecanismos de chequeo de integridad.

### **RNF 15**

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.

### **RNF 16**

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**

Significa que a los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

### **RNF 17**

La seguridad no implicará lentitud o retaso en la respuesta dada por el sistema, por lo que se debe minimizar y reducir el tiempo de respuesta, así como optimizar el código.

### **Interfaz Externa.**

Son aquellos que describen la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle, sino que especifican cómo se pretende que sea la interfaz externa del producto.

### **RNF 18**

Tener conectividad con los procesos que gestionan los datos de las unidades de salud, el personal de salud, la población, las localidades y las ubicaciones.

### **RNF 19**

Todas las funciones se realizarán desde el portal principal y una vez realizada la opción, se vuelve a él.

### **RNF 20**

Diseñar una interfaz donde el personal que trabaja con la aplicación, seleccione y organice la información que desee mostrar con sólo hacer clic.

### **Interfaz Interna**

Son aquellos que describen la apariencia del producto internamente, cómo deben ser sus componentes, su relación entre ellos, así como el funcionamiento interno del software.

### **RNF 21**

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

### **RNF 22**

Los componentes reutilizables entre los módulos que componen el Sistema de Información para la Salud deberán ser desarrollados como servicios web XML que interactúan a través de SOAP con otros componentes.

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

Son aquellos que describen los requisitos que proveen a los usuarios y clientes la documentación necesaria para la utilización y manipulación del sistema, así como para el entendimiento del mismo.

### **RNF 23**

Disponer de instrucciones en una opción de ayuda.

### **RNF 24**

Disponer de una documentación del sistema, documentada con alguna metodología de desarrollo.

### **Software**

Son aquellos que describen los programas que se deben disponer, para que el sistema funcione correctamente. Indican si hay que integrar el producto con otros productos de software o si para su correcta utilización hay que asociar otros programas al mismo.

### **RNF 25**

Los clientes tendrán acceso al Sistema de Información para la Salud a través de cualquier navegador Web, recomendado Mozilla 1.5, Internet Explorer 5.0 o superior.

### **RNF 26**

Las aplicaciones legacy (aplicaciones basadas en tecnologías y hardware más viejos) deberán integrar los componentes que estén desarrollados según los requerimientos tecnológicos que está evaluando SOFTEL.

### **RNF 27**

Los reportes podrán ser impresos en formato .pdf y .xls.

### **Hardware**

Son aquellos que especifican las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

### **RNF 28**

Se permitirá aumentar la cantidad de servidores o adicionar componentes de hardware en función de disminuir el tráfico o balancear la carga, sin que sea necesario realizar modificaciones al software.

### **RNF 29**

Impresora local o de red para imprimir los reportes solicitados.

### **Restricciones de diseño e implementación.**

Son aquellos que especifican o restringen la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente. Ejemplos de ellas son: estándares requeridos, lenguajes de programación a ser usados para la implementación, uso obligatorio de ciertas herramientas de desarrollo, restricciones en la arquitectura o el diseño, bibliotecas de clases, etc.

### **RNF 30**

La lógica de presentación constituirá una capa independiente de la lógica de negocio, centrando su función en la interfaz de usuario y validaciones simples de los datos de entrada.



### **RNF 31**

Validar el proceso de la captación de datos para evitar entradas inadecuadas.

### **RNF 32**

Se utilizarán herramientas de desarrollo que garanticen la calidad de todo el ciclo de desarrollo del producto.

## **2.5.2 Modelo de Casos de Uso del Sistema**

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso proporciona la entrada fundamental para el análisis, el diseño y la implementación en el desarrollo de un sistema de Software. Es un modelo del sistema que contiene actores, casos de usos y las relaciones que existen entre ellos.

### **2.5.2.1 Definición de Actor del Sistema**

Los actores del sistema suelen corresponderse con los trabajadores del negocio y en algunas ocasiones con los actores del mismo. Un actor juega un papel por cada caso de uso con el que colabora representando terceros fuera del sistema, por lo cual una vez que se ha identificado todos los actores del sistema se tiene identificado el entorno externo al mismo. Los actores del sistema no son parte de él, pero pueden intercambiar información con él y ser un recipiente pasivo de información, además, pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. A continuación se justifican los actores del sistema propuesto:

Actor	Descripción
Funcionario del MINSAP	Este rol puede ser desempeñado por cualquier funcionario del MINSAP encargado de obtener los reportes estadísticos del Registro de Áreas de Salud.
RPS	Es el encargado de brindar los datos del Personal de Salud, así como su vinculación laboral.
RU	Es el encargado de brindar las Provincias y Municipios, así como las calles.
RUS	Es el encargado de brindar todos los datos de las Unidades de Salud que son Áreas de Salud.
RL	Es el encargado de brindar las Circunscripciones, CDR y Manzanas de la población que atiende un EBS determinado.

Tabla 2.1 Definición de los actores del sistema.

### 2.5.2.2 Diagrama de Casos de Uso del Sistema

Un diagrama de casos de uso del sistema representa gráficamente los procesos y su interacción con los actores, describiendo lo que hace el sistema para cada tipo de usuario. Cada tipo de usuario se representa mediante un o más actores, también se representa mediante uno o más actores cada sistema externo con el que interactúa el sistema, incluyendo los dispositivos.

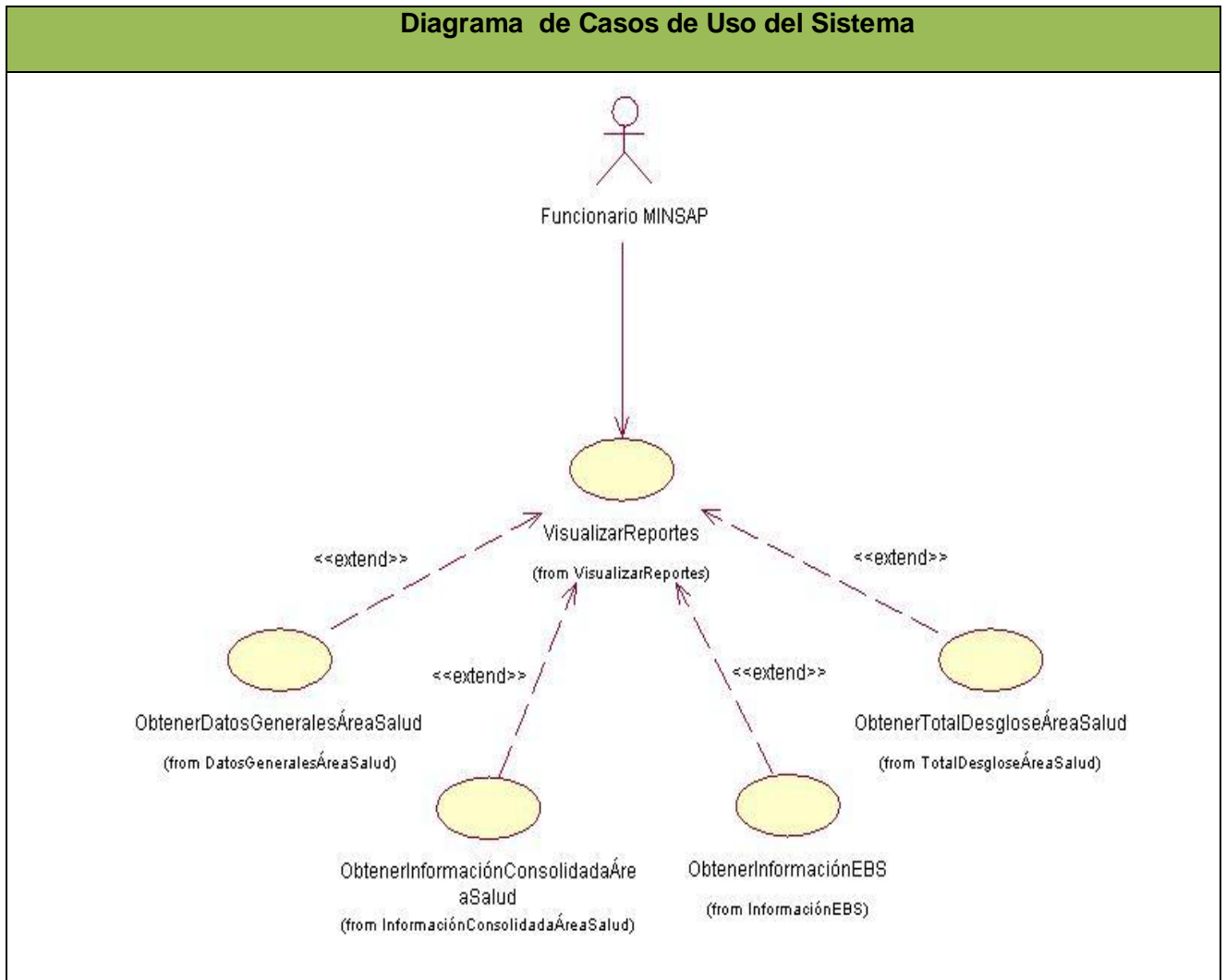


Figura 2.2 Diagrama de Casos de Uso del Sistema.

### 2.5.2.3 Descripción de los Casos de Uso del Sistema

Nombre del Caso de Uso	ObtenerDatosGeneralesÁreaSalud
<b>Actor</b>	Funcionario del MINSAP
<b>Descripción</b>	El caso de uso comienza cuando el Funcionario del MINSAP se interesa por los datos generales de una o varias AS y elige la opción Datos Generales del AS que aparece en el menú de los Reportes, el sistema muestra una interfaz dando la posibilidad de buscar las AS por provincias, municipios o ir específicamente al AS deseada, al realizar la búsqueda se muestra una interfaz con los datos que se quieren, permitiendo además imprimir dichos resultados, terminando así el caso de uso.
<b>Referencia</b>	RF 1

Tabla 2.2 Descripción del Caso de Uso Obtener Reportes Datos Generales del Área de Salud.

Nombre del Caso de Uso	ObtenerInformaciónConsolidadaÁreaSalud
<b>Actor</b>	Funcionario del MINSAP
<b>Descripción</b>	El caso de uso comienza cuando el Funcionario del MINSAP se interesa por la información consolidada de una o varias AS y elige la opción Información Consolidada del AS que aparece en el menú de los Reportes, el sistema muestra una interfaz dando la posibilidad de buscar las AS por provincias, municipios o ir específicamente al AS deseada, al realizar la búsqueda se muestra una interfaz con los datos que se quieren, permitiendo además imprimir dichos resultados, terminando así el caso de uso.
<b>Referencia</b>	RF 2

Tabla 2.3 Descripción del Caso de Uso Obtener Reporte Información Consolidada del Área de Salud.

Nombre del Caso de Uso	ObtenerInformaciónEBS
<b>Actor</b>	Funcionario del MINSAP
<b>Descripción</b>	El caso de uso comienza cuando el Funcionario del MINSAP se interesa por la información de uno o varios EBS y elige la opción Información del EBS que aparece en el menú de los Reportes, el sistema muestra una interfaz dando la posibilidad de buscar las AS por provincias, municipios o ir específicamente al AS donde se encuentran los EBS deseados, al realizar la búsqueda se muestra una interfaz con los datos que se quieren, permitiendo además imprimir dichos resultados en formatos Adobe Acrobat (pdf) o Microsoft Excel (xls), teniendo en cuenta que por problemas de arquitectura y el gran volumen de información que se recoge, sólo se muestran los datos más significativos previo acuerdo con el cliente en el caso del Adobe Acrobat (pdf), terminando así el caso de uso.
<b>Referencia</b>	RF 4

Tabla 2.4 Descripción del Caso de Uso Obtener Reporte Información del EBS.

Nombre del Caso de Uso	ObtenerTotalDesgloseAreaSalud
<b>Actor</b>	Funcionario del MINSAP
<b>Descripción</b>	El caso de uso comienza cuando el Funcionario del MINSAP se interesa por el total y/o desglose lógico de las AS y elige la opción Total y/o Desglose Lógico del las AS que aparece en el menú de los Reportes, el sistema muestra una interfaz dando la posibilidad de buscar las AS por provincias y municipios, así como crear la profundidad que desee, al realizar la búsqueda se muestra una interfaz con los datos que se quieren, permitiendo además imprimir dichos resultados, terminando así el caso de uso.
<b>Referencia</b>	RF 3

Tabla 2.5 Descripción del Caso de Uso Obtener Reporte Total y Desglose Lógico del las Aéreas de Salud.

### **2.6 Conclusiones**

Con el desarrollo del presente capítulo se obtuvo una perspectiva del sistema que se desea desarrollar, términos de requerimientos funcionales y no funcionales. Además, se identificaron y describieron los actores que interactúan con las funcionalidades previstas para cada uno de los casos de uso previstos a desarrollar en la Implementación de las Estadísticas Descriptivas del Registro de Áreas de Salud, siendo estos últimos descritos, priorizados y estableciéndose la referencia entre estos y las funcionalidades con las que debe cumplir el sistema. Con la culminación de este capítulo se sentaron las bases para el exitoso desarrollo de las Disciplinas de Análisis y Diseño, Implementación y Pruebas de las Estadísticas Descriptivas del Registro de Áreas de Salud.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

### 3.1 Introducción

Con el presente capítulo se tiene el propósito de definir la estructura y elementos del Análisis y Diseño. En el Análisis se define el diagrama de clases del análisis de la aplicación, se especifican las clases del análisis que toman parte del caso de uso, las relaciones entre ellas y las entidades. En el Diseño se modela y adquiere forma el sistema para que soporte todos los requisitos funcionales o no funcionales e inclusive cualquier otro tipo de restricción, contribuyendo a obtener una arquitectura sólida y estable para la futura implementación del sistema de software.

### 3.2 Análisis

Durante el desarrollo del Flujo de Trabajo de Análisis se obtiene una visión del sistema que preocupa de ver QUÉ es lo que hace, de modo que solo se interesa de los requerimientos funcionales.

#### 3.2.1 Modelo Análisis

El modelo de análisis es un modelo de objetos conceptual que analiza los requisitos mediante su refinamiento y estructuración y permite razonar sobre los aspectos internos del sistema, incluidos sus recursos compartidos internos. Además ofrece un mayor poder expresivo y una mayor formalización y proporciona una estructura centrada en el mantenimiento en aspectos tales como la flexibilidad ante los cambios y la reutilización. Esta estructura no solo es útil para el mantenimiento de los requisitos como tales, sino también que se utiliza como entrada en las actividades de Diseño e Implementación.

El principal artefacto que se genera durante este flujo de trabajo es la realización de los casos de uso del análisis que describen como se refinan los casos de uso en términos de colaboraciones dentro del modelo de análisis y además de cómo se ejecuta un caso de uso determinado en términos de las clases del análisis y de sus objetos del análisis en su interacción. Las realizaciones de los casos de uso aislarán los cambios en los casos de uso, debido a que si cambia un caso de uso, debe cambiarse también su realización.

### 3.2.2 Diagrama de Clases del Análisis

Un Diagrama de Clases del Análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. En el caso de los diagramas de clases del análisis de las Estadísticas Descriptivas del Registro de Áreas de Salud se representan mediante un conjunto de Clases Interfaces, Controladoras y Entidades y las relaciones que existen entre ellas. Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. RUP propone clasificar a las clases en:

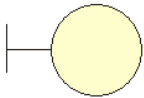


Nombre	Características	Representación
Interfaz	Estas clases modelan la interacción Actor – Sistema. Representan ventanas, formularios, comunicación con otros sistemas o dispositivos, interfaces de comunicaciones, interfaces de impresoras.	
Entidad	Modelan la información persistente del sistema y el comportamiento asociado a una información. Las clases entidad suelen mostrar una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema.	
Control	Coordinan el trabajo de las clases. Encapsulan el comportamiento de un CU. Los aspectos dinámicos del sistema se modelan con clases de control, debido a que ellas manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos, es decir objetos de interfaz y de entidad.	

Tabla 3.1 Clases del Análisis



En una aplicación cliente/servidor de tres capas, como es el caso de este trabajo, en la capa de usuario aparecen fundamentalmente clases interfaz ya que allí se ejecutan las aplicaciones del cliente. En la capa intermedia están las clases de control ya que en ella se agrupan los servicios que son compartidos por múltiples aplicaciones. En la capa servidor estarían las clases entidad porque allí se tienen las bases de datos.

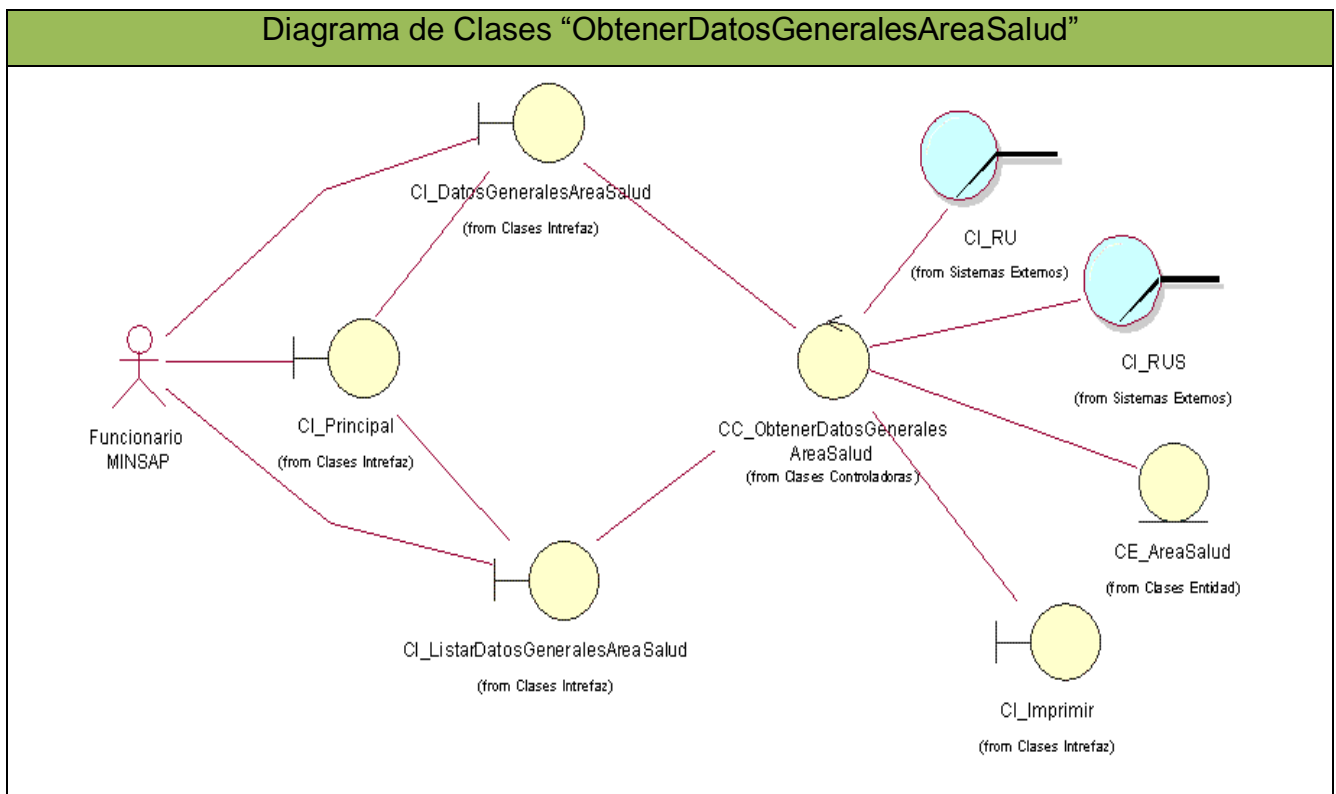


Figura 3.1 Diagrama de Clases del Análisis “ObtenerDatosGeneralesAreaSalud”

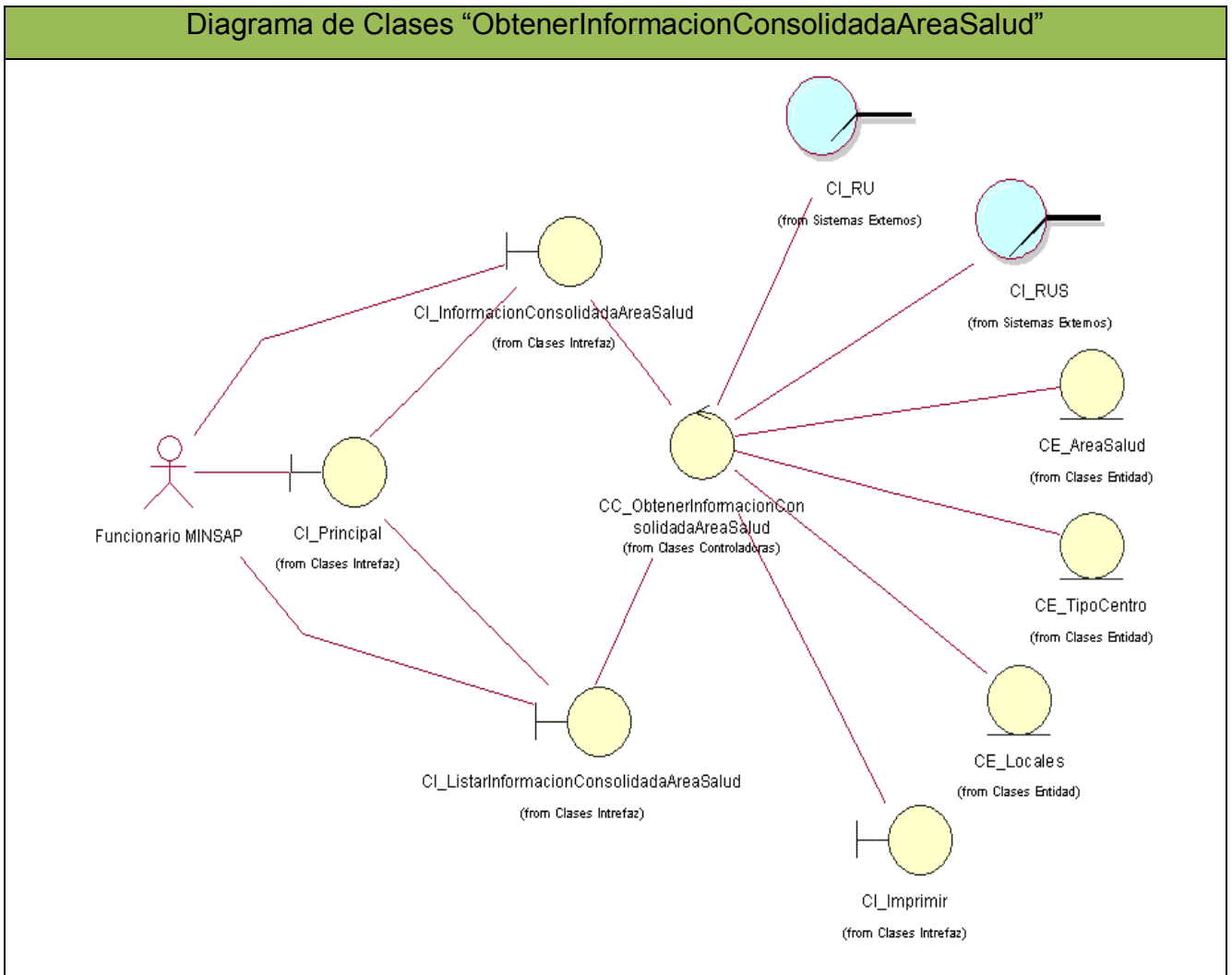


Figura 3.2 Diagrama de Clases del Análisis "ObtenerInformacionConsolidadaAreaSalud"

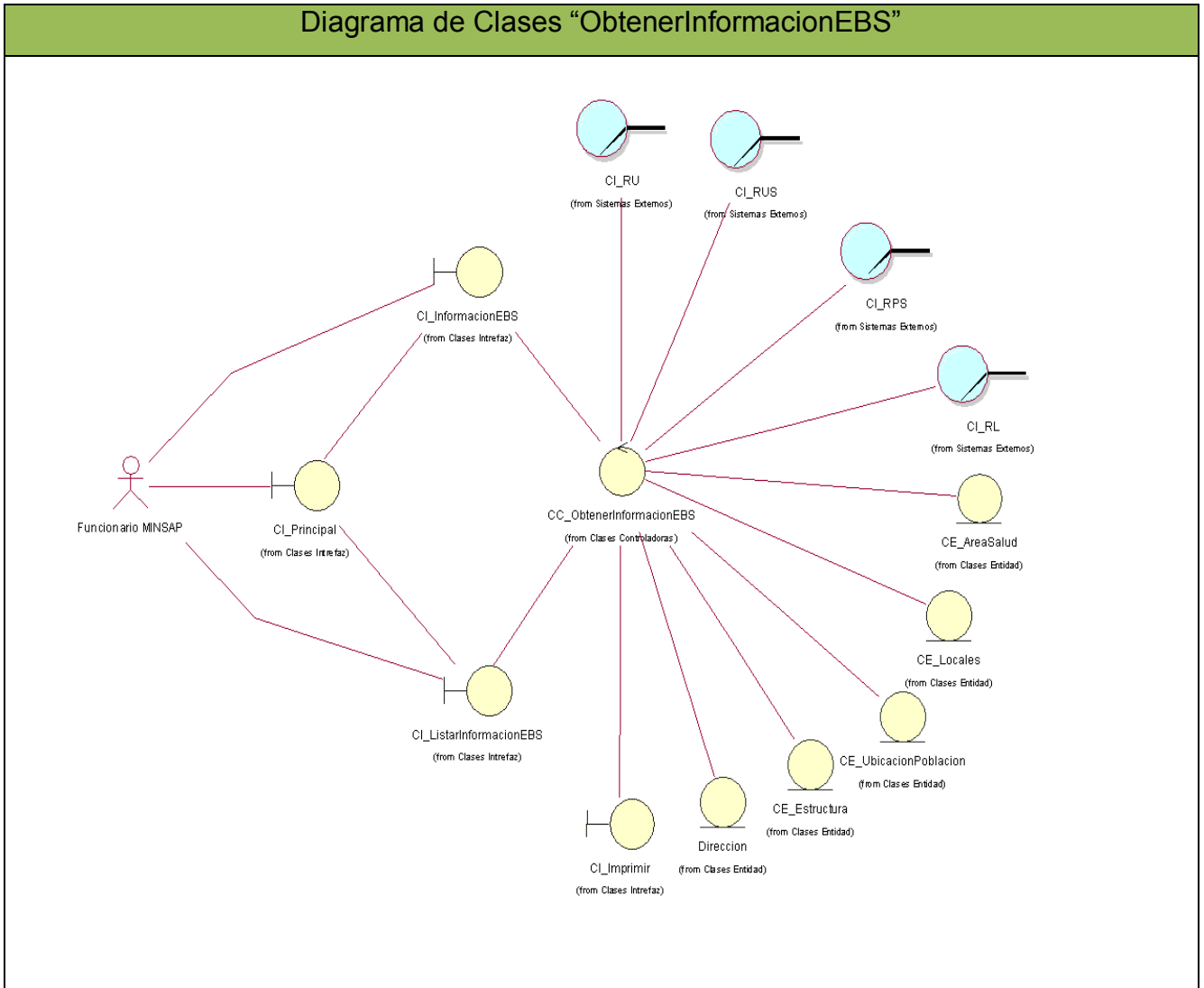


Figura 3.3 Diagrama de Clases del Análisis “ObtenerInformacionConsolidadaAreaSalud”

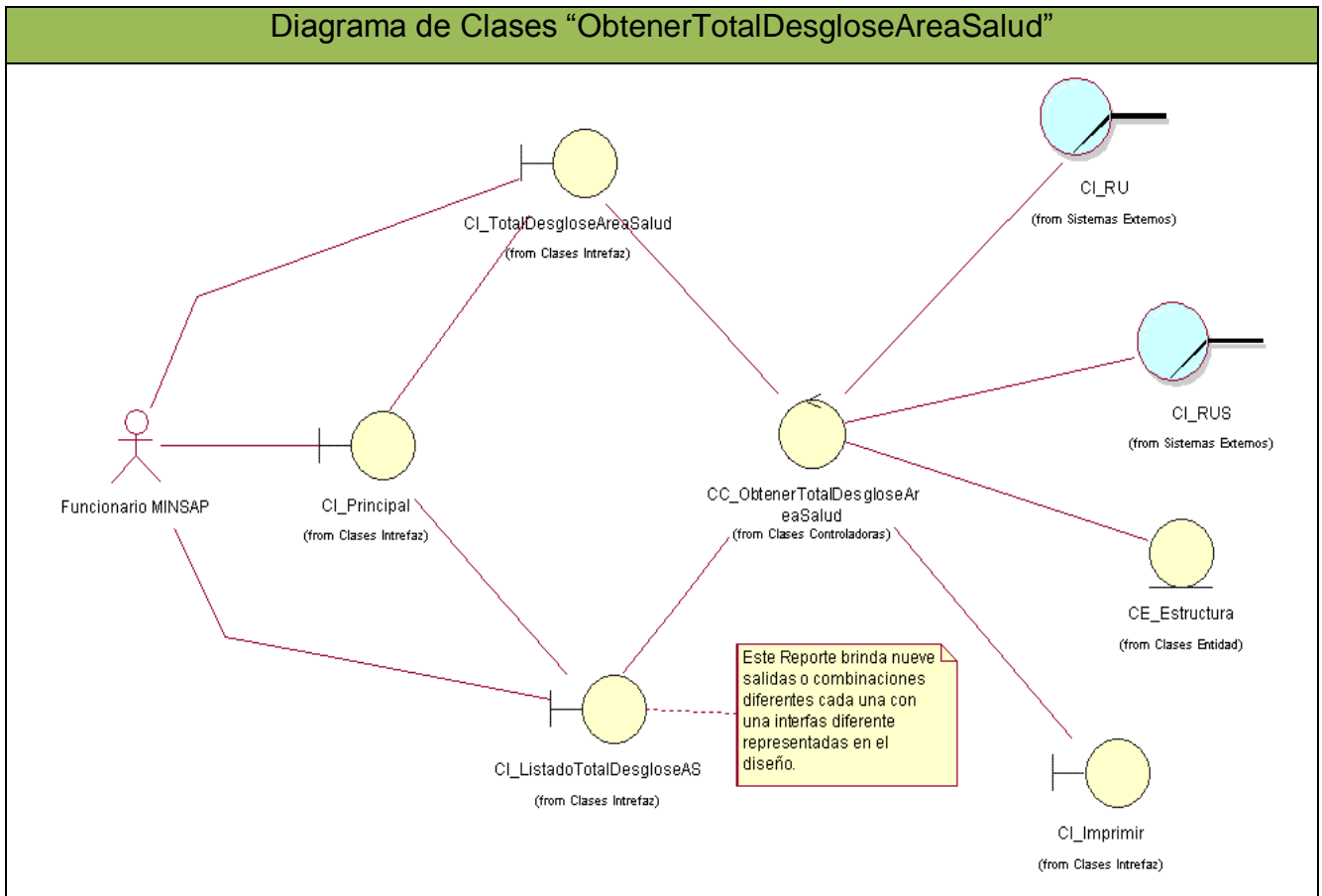


Figura 3.4 Diagrama de Clases del Análisis “ObtenerTotalDesgloseAreaSalud”

### 3.2.3 Diagramas de Interacción

Los Diagramas de Interacción muestran las interacciones entre objetos mediante transferencia de mensajes entre objetos o subsistemas. Se utilizan para modelar los aspectos dinámicos de un sistema, estos se dividen en Diagramas de Secuencia y Diagramas de Colaboración.

En el presente trabajo se definió realizar Diagramas de Colaboración ya que el objetivo fundamental es identificar requisitos y responsabilidades sobre los objetos y no identificar secuencias de interacción ordenadas cronológicamente por lo que no se realizan Diagramas de Secuencia.

En los Diagramas de Colaboración se muestran las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces, es decir, un Diagrama de Colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes. El

nombre de un mensaje debe denotar el propósito del objeto invocante en la interacción con el objeto invocado.

A continuación se muestran los diagramas de colaboración de las Estadísticas Descriptivas del Registro de Áreas de Salud.

## Diagrama de Colaboración del Análisis “ObtenerDatosGeneralesÁreaSalud”

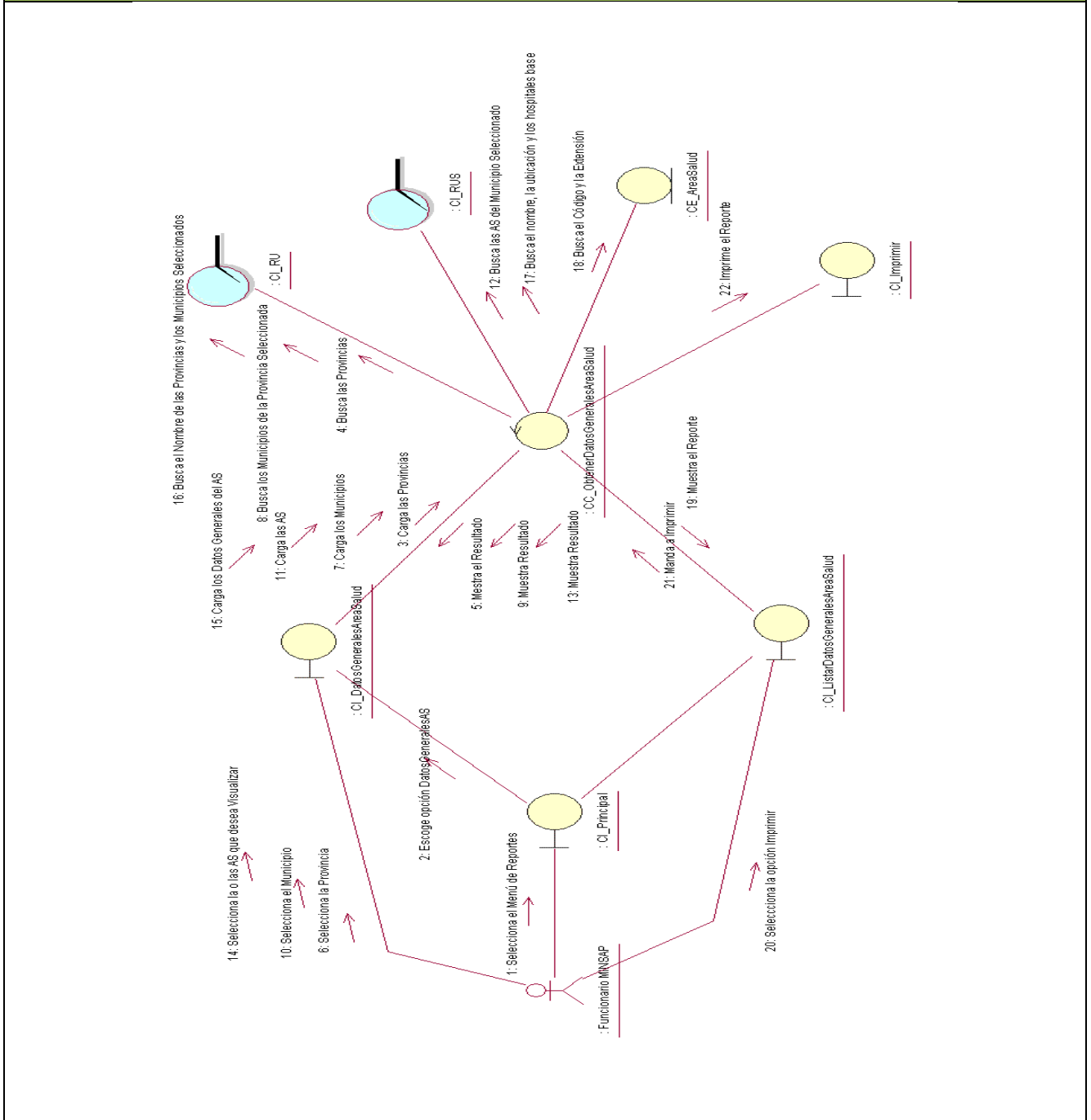


Figura 3.5 Diagrama de Colaboración del Análisis “ObtenerDatosGeneralesÁreaSalud”

## Diagrama de Colaboración del Análisis “ObtenerInformacionConsolidadaAreaSalud”

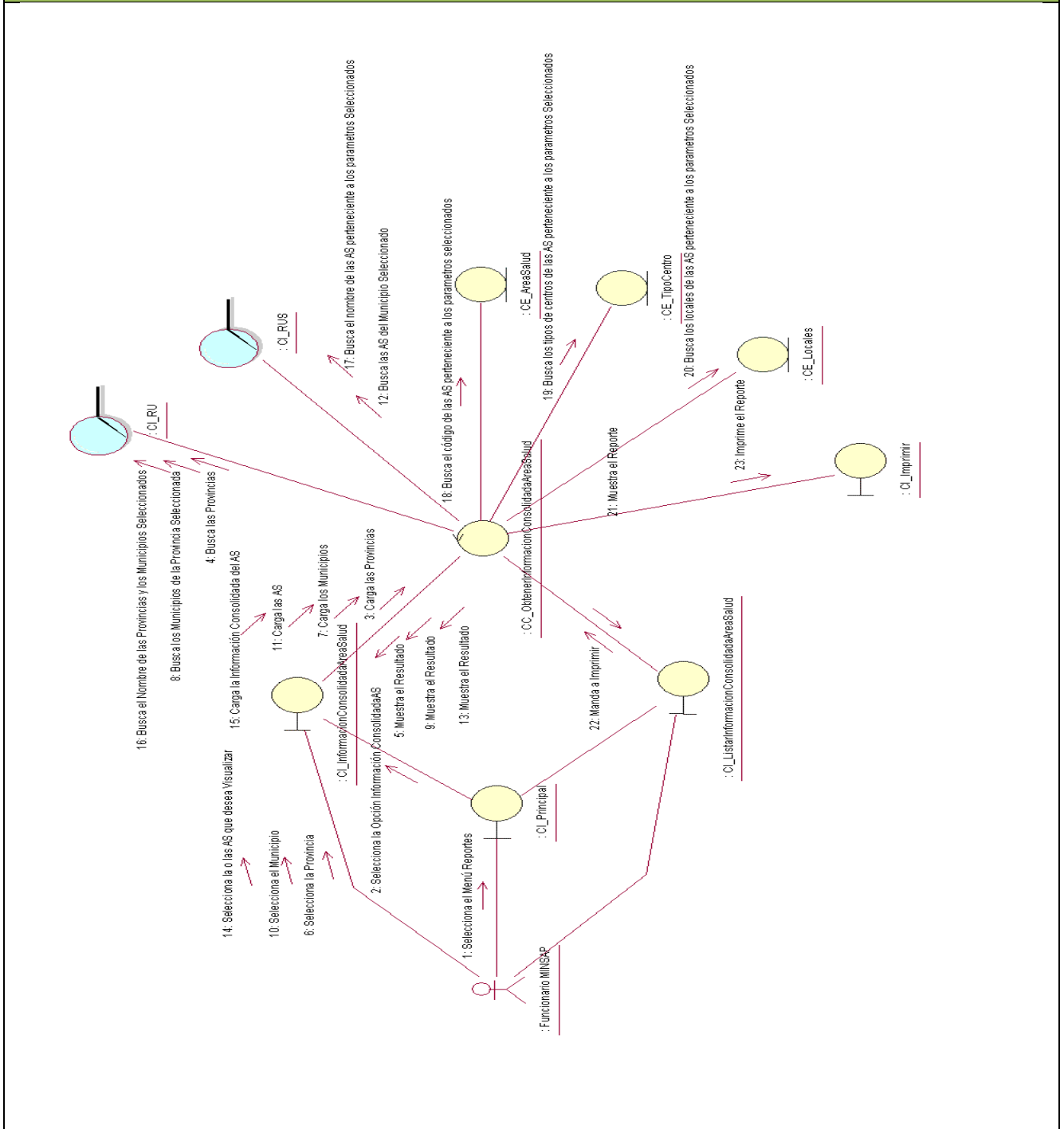


Figura 3.6 Diagrama de Colaboración del Análisis “ObtenerInformacionConsolidadaAreaSalud”

## Diagrama de Colaboración del Análisis “ObtenerInformacionEBS”

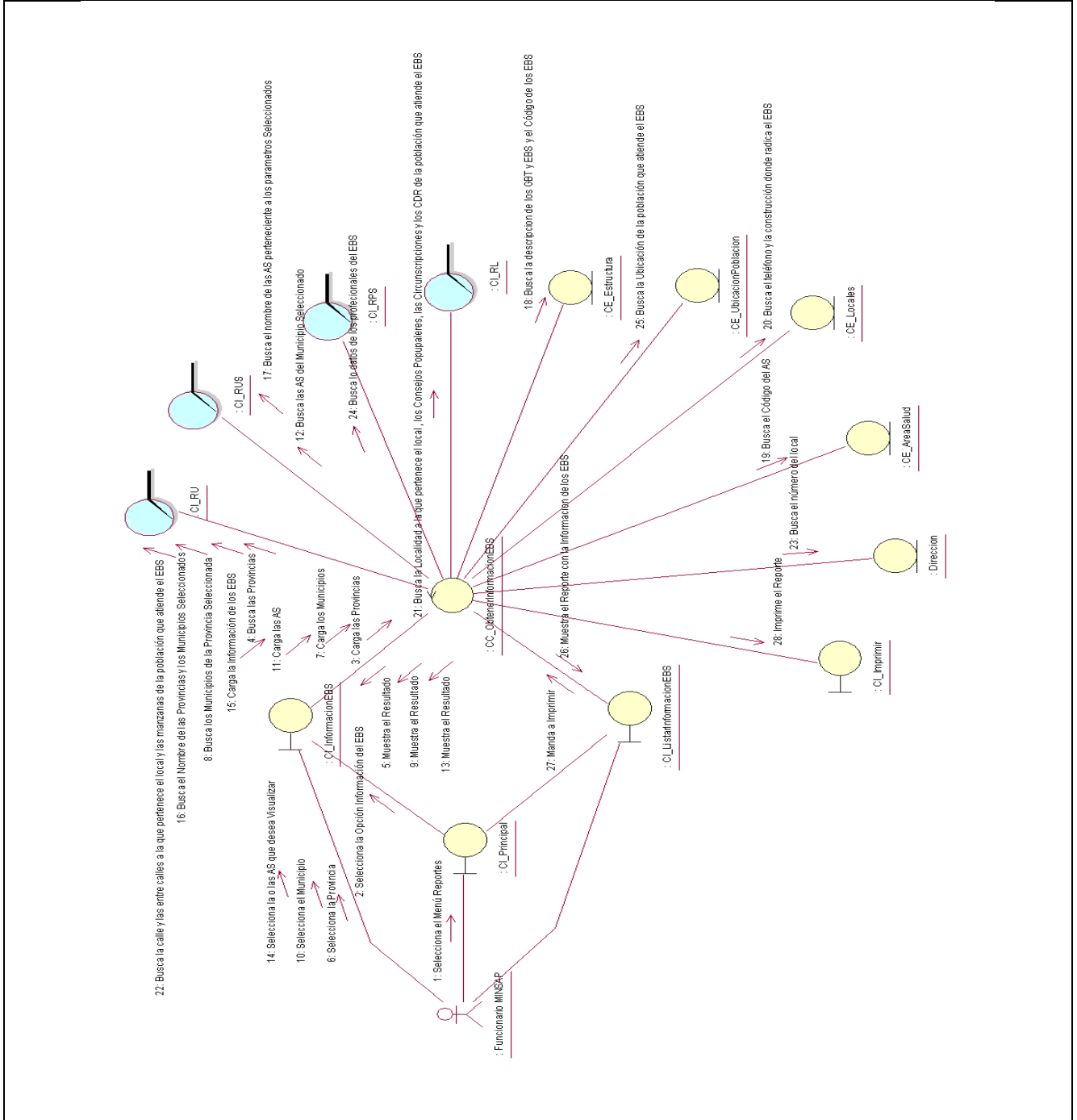


Figura 3.7 Diagrama de Colaboración del Análisis “ObtenerInformacionEBS”



## Diagrama de Colaboración del Análisis “ObtenerTotalDesgloseAreaSalud”

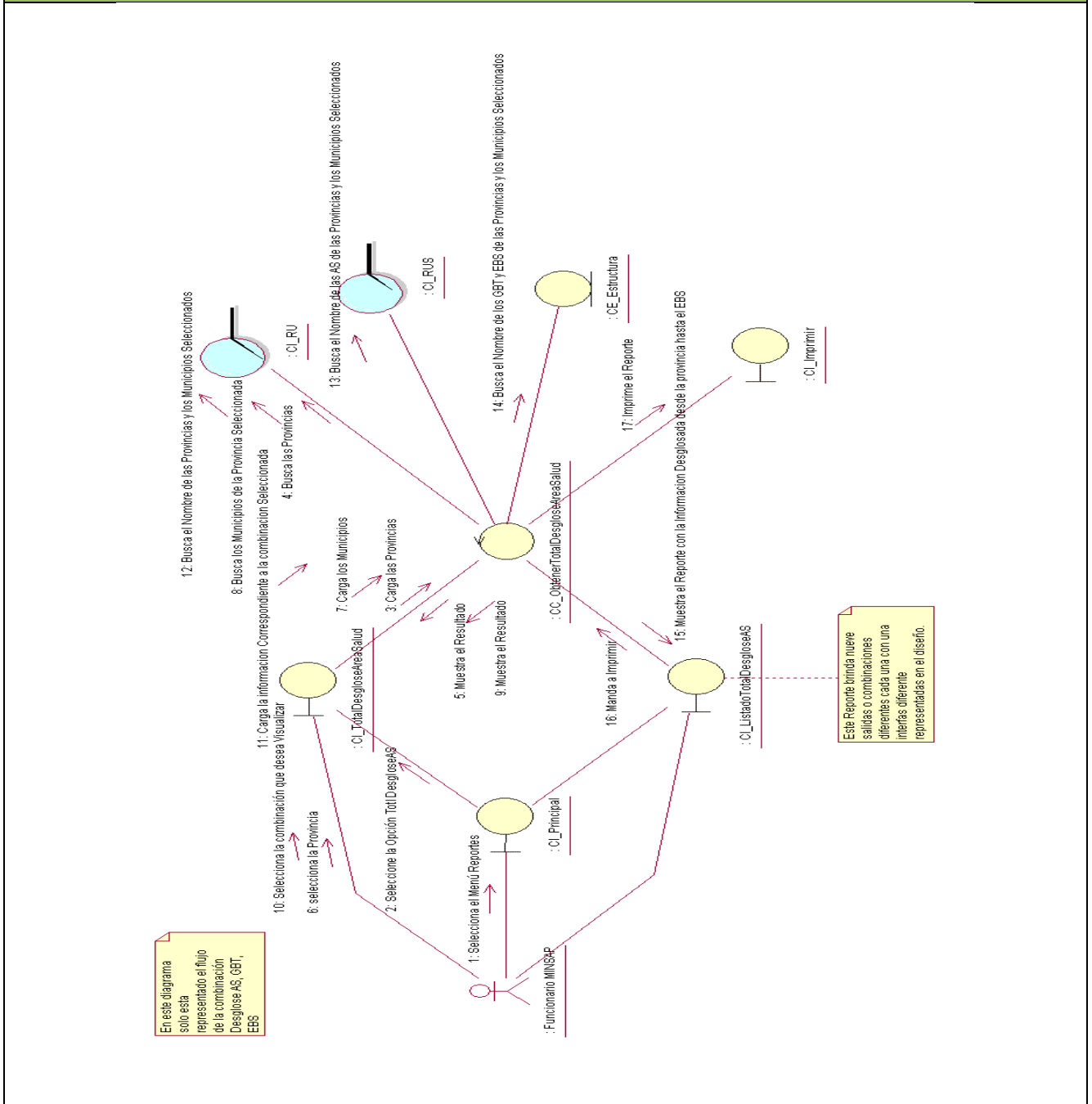


Figura 3.8 Diagrama de Colaboración del Análisis “ObtenerTotalDesgloseAreaSalud”

## 3.3 Modelo de Diseño

El Modelo de Diseño permite adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.

Descompone los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. Captura las interfaces entre los subsistemas en el ciclo de vida del software, lo cual es muy útil cuando se utilizan interfaces como elementos de sincronización entre diferentes equipos de desarrollo. Los artefactos generados en el modelo de diseño son: Modelo de Despliegue, Descripción de la Arquitectura, Realización de Casos de Uso, Clase del Diseño, Subsistema de Diseño, Interfaz.

### 3.3.1 Justificación del Uso de Patrones

Un patrón es un modelo que se puede seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar lograr ciertos objetivos promoviendo buenas prácticas.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Un patrón de diseño es una solución a un problema de diseño.

Los patrones de diseño pretenden proporcionar catálogos de elementos reusables en el diseño de sistemas software, formalizar un vocabulario común entre diseñadores, estandarizar el modo en que se realiza el diseño, facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

En el sistema desarrollado se utilizan patrones como: Proxy, Alta Cohesión y Bajo Acoplamiento, los cuales jugaron determinadas funciones que se explican a continuación:

#### **Patrón Proxy**

El Patrón Proxy trabaja como un concentrador de otro objeto para controlar el acceso a éste. Frecuentemente este patrón se usa cuando se tienen objetos que consumen muchos recursos o memoria, y se quiere instanciar dichos objetos solo en el momento que sean pedidos por el cliente, aunque también se puede usar en otros casos.

Es un patrón muy general que obliga a que las llamadas a métodos de un objeto ocurran indirectamente a través de un objeto proxy, que actúa como sustituto o intermediario del objeto original, delegando luego las llamadas a los métodos de los objetos respectivos.

### **Patrón Alta Cohesión**

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

El Patrón Alta Cohesión mejora la claridad y facilidad con que se entiende el diseño. Se simplifica el mantenimiento y las mejoras de funcionalidad. A menudo se genera un bajo acoplamiento. Soporta mayor capacidad de reutilización.

### **Patrón Bajo Acoplamiento**

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas.

Acoplamiento bajo significa que una clase no depende de muchas clases.

Acoplamiento alto significa que una clase recurre a muchas otras clases.

Con el empleo de este patrón no se afectan por cambios de otros componentes, estos serían fáciles de entender por separado y de reutilizar.

### **3.3.2 Estructura del Diseño**

La estructura del diseño permite ver la relación de los subsistemas de diseño dentro de la arquitectura definida. Un subsistema define una forma coherente de examinar un aspecto del problema. Dividir el diseño de un sistema en subsistemas de diseño da la posibilidad de organizar el modelo de diseño en porciones más manejables.

A continuación se describirá el contenido de cada uno de los subsistemas de servicios con el objetivo de posibilitar una mayor comprensión del Diagrama de Subsistemas de Servicios.

**Capa de Presentación:** Esta capa reúne todos los aspectos del software que tienen que ver con las interfaces y la interacción con los diferentes tipos de usuarios. Por lo que presenta el sistema al usuario, le comunica la información y captura la información del usuario realizando un mínimo de proceso de filtrado previo para comprobar que no hay errores de formato. Típicamente incluyen el

manejo y aspecto de las ventanas, el formato de los reportes, menús, gráficos y elementos multimedia en general. Esta capa se comunica únicamente con la capa de negocio.

**PlaSer:** Este subsistema está representado por la clase de acceso a datos `Plaser_DBz` y los ficheros `configplaser_client` y `configplaser_server`, utilizados como ficheros de configuración de un componente determinado, el primero contiene la URL para la comunicación con los componentes dependientes y el otro los métodos pertenecientes al componente desplegado. En este subsistema también se encuentran las librerías `PlaSer_XML` y `PlaSer_XSLT`, las cuales permiten la manipulación de los documentos XML y el formateo de estos para ser mostrados a los clientes a través de páginas XHTML.

**Web/General:** Este subsistema contiene los ficheros de configuración, de seguridad y de acceso a los diferentes módulos, los cuales son utilizados en la Capa de Presentación.

**Core/General:** Representa los ficheros utilizados en la Capa de Negocio para las validaciones y generación de ficheros en formato Portable Document Format (PDF) o Microsoft Office Excel (XLS).

**Capa de Negocio:** Esta capa reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio), pues es aquí donde se establecen todas las reglas que deben cumplirse y las restricciones, así como las tareas que forman parte de los procesos. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar, almacenar o recuperar datos de él.

**Capa de Datos:** Esta capa reúne todos los aspectos del software que tienen que ver con el manejo de los datos persistentes. Es donde residen los datos. Está formado por el servidor de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

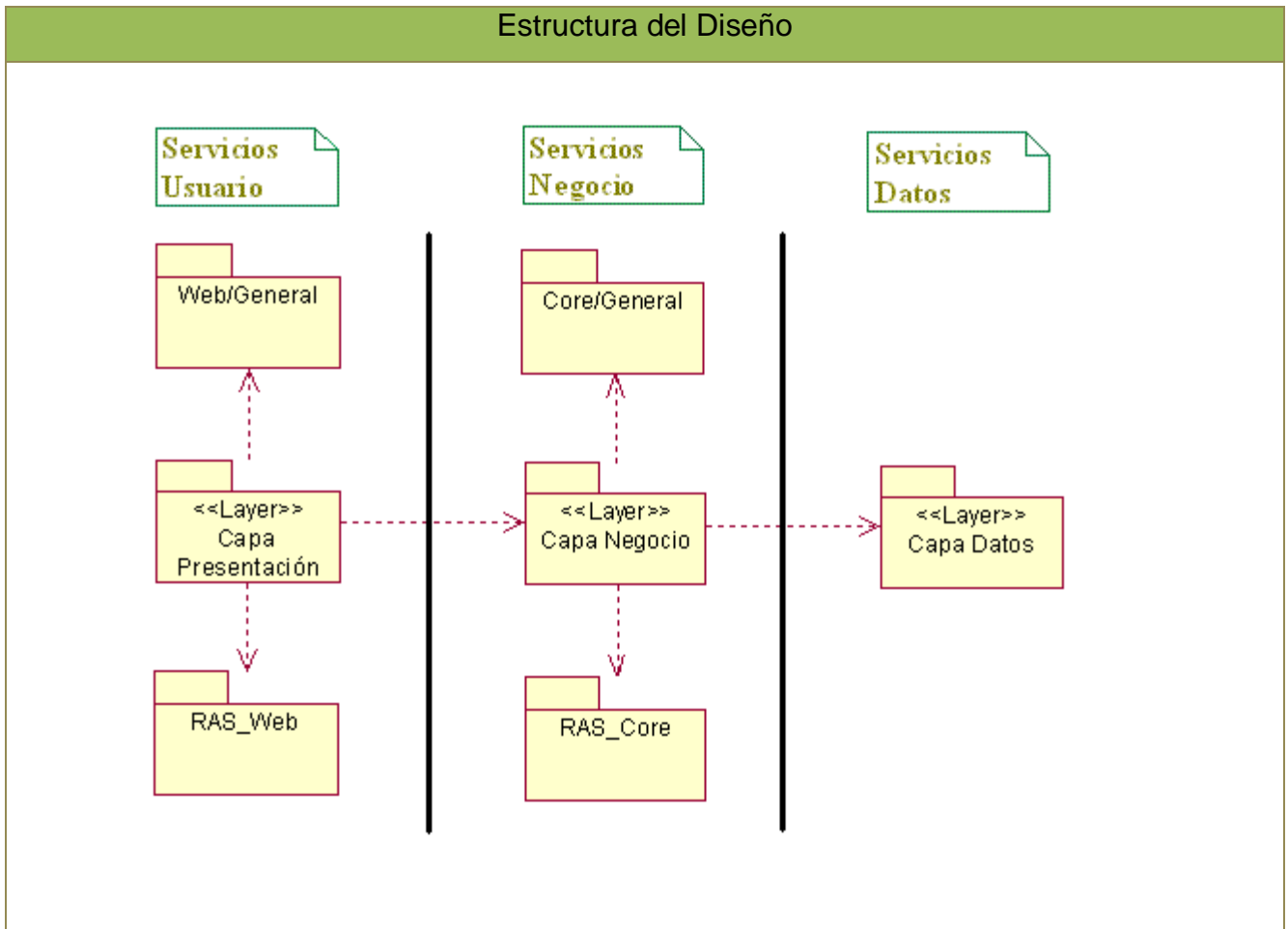


Figura 3.9 Estructura del Diseño

### 3.3.3 Definición de Elementos del Diseño

La extensión UML para Web presenta como elementos más significativos a 3 clases de UML estereotipadas con los siguientes estereotipos "Server Page", "Client Page", "Form" empleados para el código servidor, código cliente y formularios respectivamente.

A continuación se describe en qué consiste cada estereotipo y cómo se utilizan en el diseño:

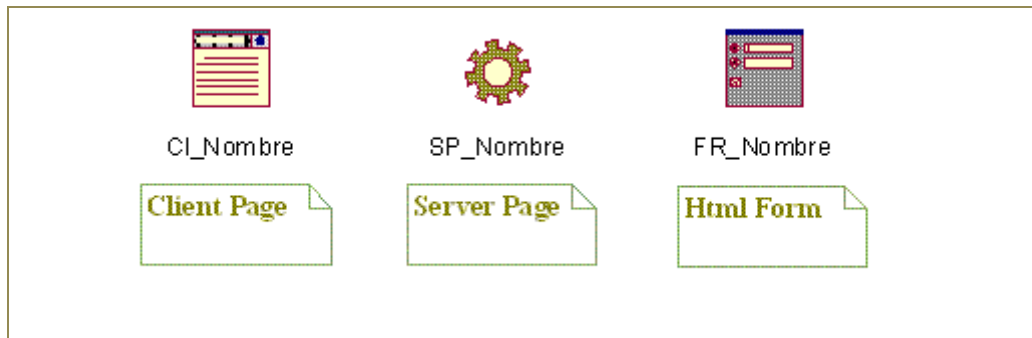


Figura 3.10 Estereotipos Web

**<<Client Page>>**: Es una página Web con formato XHTML. Mezcla de datos, presentación y lógica, puede contener scripts interpretados por el browser, Las operaciones representan funciones en el script en la página, Los atributos representan variables del *script* y puede participar en asociaciones con «Client Page»'s o «Server Page»'s

**<<Server Page>>**: Página web con scripts ejecutados por el servidor, los scripts interactúan con recursos del servidor (Base de datos, lógica del negocio, sistemas externos, etc), las operaciones representan las funciones en el script y los atributos representan variables visibles en el ámbito de la página.

**<<Form>>**: Formulario: colección de campos de entrada de datos, parte de una «Client Page». Los atributos representan campos de entrada del form HTML, no tiene operaciones y toda operación que interactúa con el form es una propiedad de la página que contiene el form.

#### Estereotipos de asociación:

**<<Build>>**: Representa la relación existente entre las páginas cliente, que de forma general expresa cómo las páginas que se encuentran en el servidor construyen las páginas en el cliente. Es una relación direccional, donde una página servidor construye una o más páginas cliente.

**<<Call>>**: Se utiliza para llamadas a páginas servidoras que representan métodos de la capa de negocio.

**<<Include>>**: Una página servidor puede incluir a otra página del mismo tipo, pudiendo utilizar todas las funciones brindadas por esta última.

**<<Link>>**: Permite ir de una página cliente a otra página cliente.

**<<Redirect>>**: Una página servidora puede redireccionar el procesamiento a otra página, es decir, enviar información para que la otra ejecute la acción.

**<<Submit>>**: Envía los valores de un formulario a una página servidora.

### 3.3.4 Diagramas de Clases del Diseño

Un Diagrama de Clases es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones. Gráficamente, un diagrama de clases es una colección de nodos y arcos. Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema principalmente, esto incluye modelar el vocabulario del sistema, las colaboraciones o los esquemas. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para sistemas ejecutables, aplicando ingeniería directa e inversa.

A continuación se presentan los diagramas de algunos de los casos de uso, modelados con la extensión de UML para aplicaciones Web.

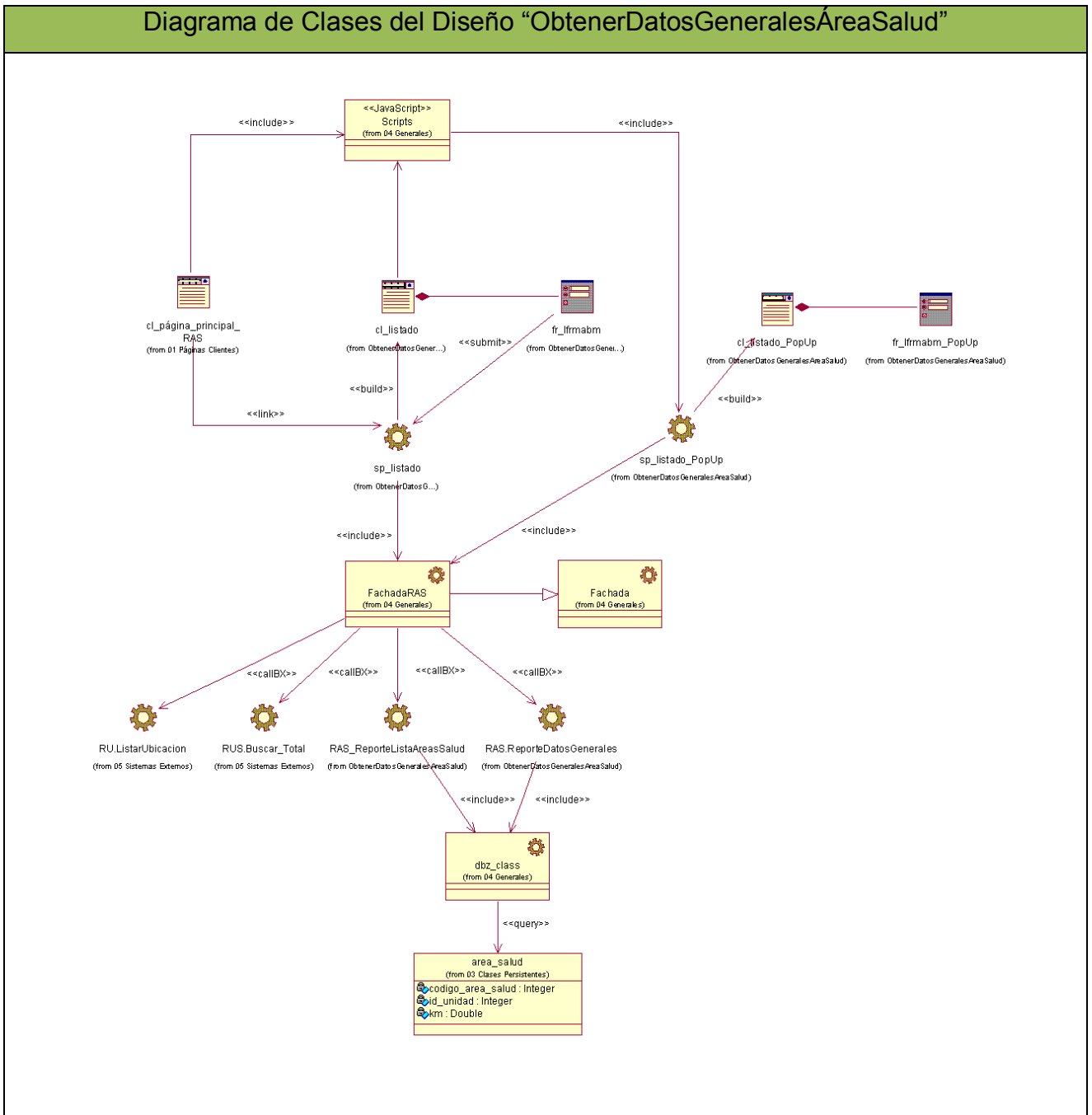


Figura 3.11 Diagrama de Clases del Diseño "ObtenerDatosGeneralesÁreaSalud"



Diagrama de Clases del Diseño "ObtenerInformacionConsolidadaAreaSalud"

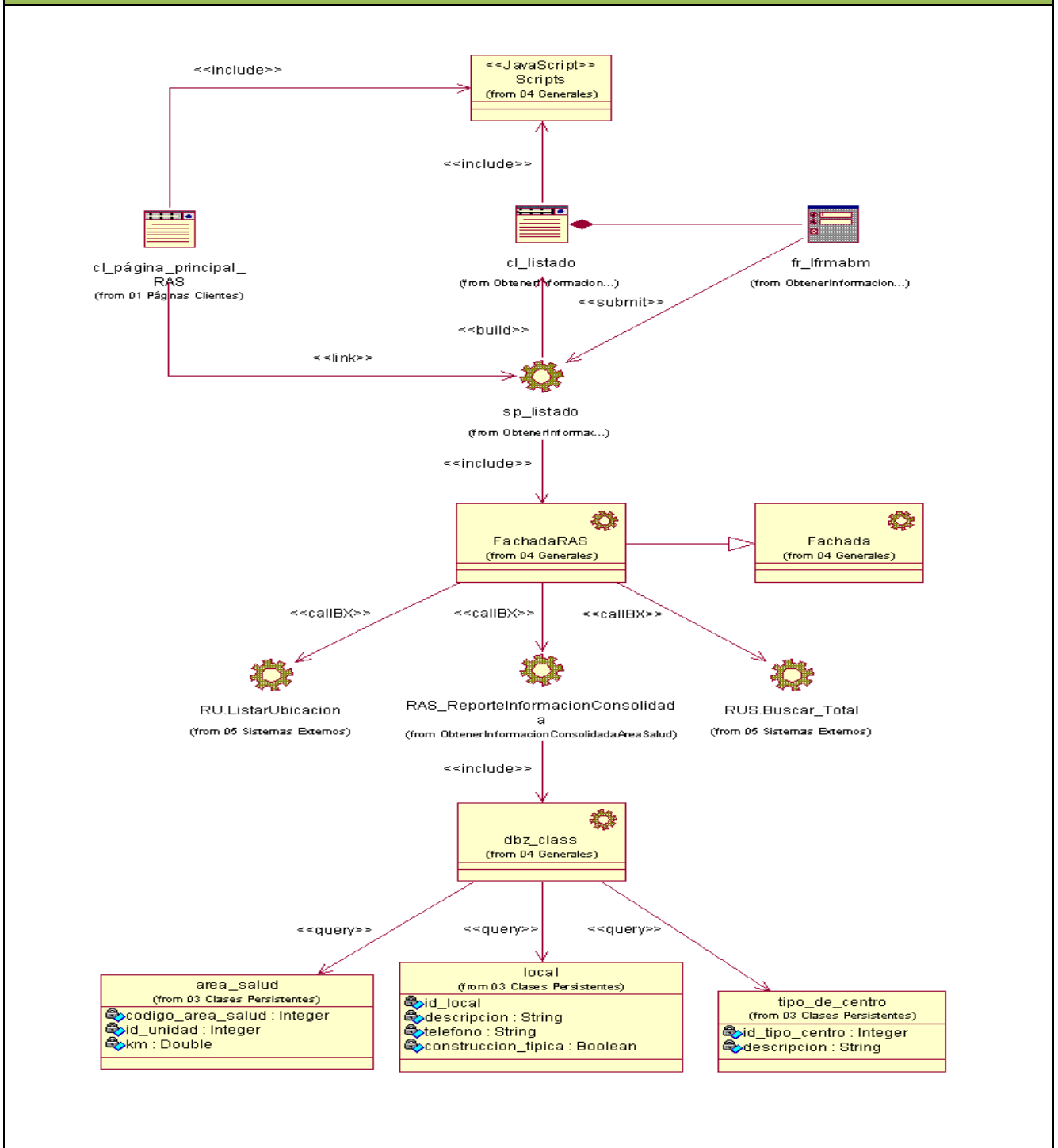


Figura 3.12 Diagrama de Clases del Diseño "ObtenerInformacionConsolidadaAreaSalud"

## Diagrama de Clases del Diseño "ObtenerInformacionEBS"

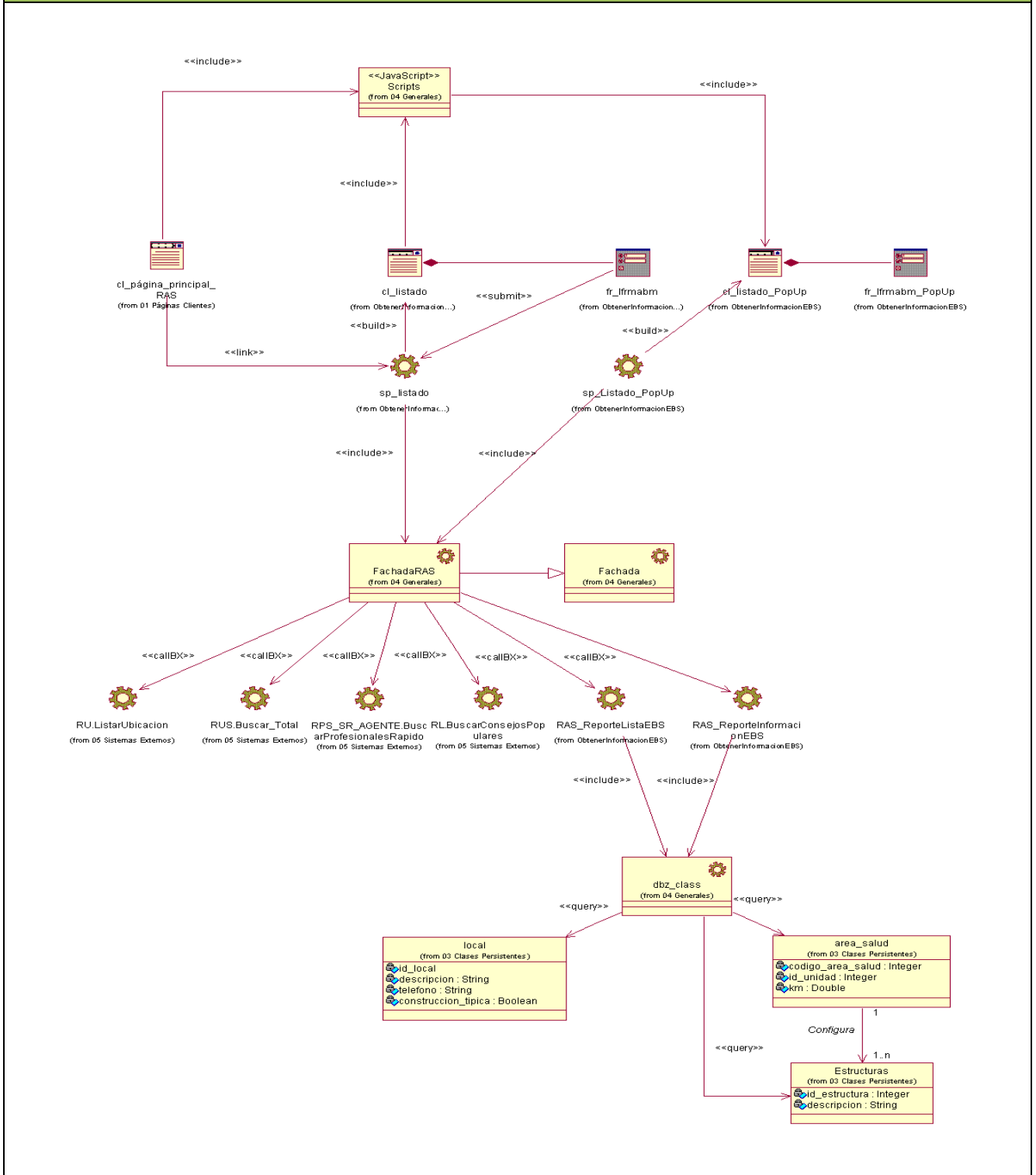


Figura 3.13 Diagrama de Clases del Diseño "ObtenerInformacionEBS"

## Diagrama de Clases del Diseño "ObtenerTotalDesgloseAreaSalud"

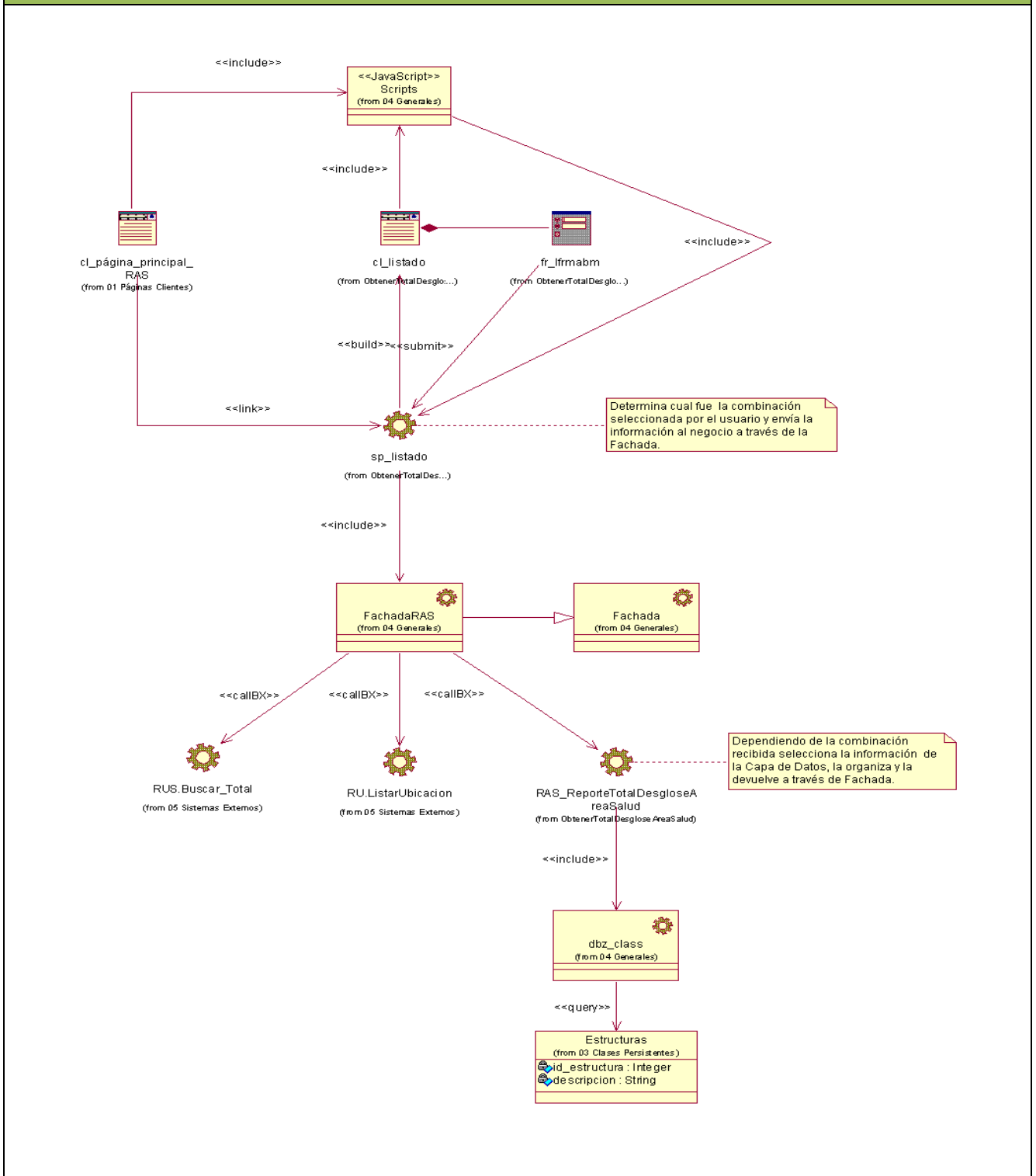


Figura 3.13 Diagrama de Clases del Diseño "ObtenerTotalDesgloseAreaSalud"

### 3.3.5 Descripción de las Clases del Diseño

A continuación se describen las clases utilizadas en los diagramas de clases presentados anteriormente:

#### Páginas Clientes (Client Page)

<b>Nombre:</b> cl_listado
<b>Tipo de Clase:</b> página cliente
<p><b>Descripción General:</b> La clase cl_listdo es una página que se ejecuta del lado del cliente sobre un navegador Web o Browser. Permite al usuario interactuar con la aplicación, realizar búsquedas de información e imprimir documentos en formato Adobe Acrobat (PDF) y Microsoft Excel (XLS). Visualiza los datos en forma de páginas como si fuera un libro, de tal manera que el usuario puede moverse de una página a otra de forma sencilla. Es utilizada en todos los casos de uso:</p> <ul style="list-style-type: none"> <li>➤ ObtenerDatosGeneralesÁreaSalud</li> <li>➤ ObtenerInformacionConsolidadaÁreaSalud</li> <li>➤ ObtenerInformacionEBS</li> <li>➤ ObtenerTotalDesgloseAreaSalud</li> </ul>

Tabla 3.2 Descripción de la clase del diseño cl\_listado

#### Páginas Servidoras (Server Page) de la Capa de Presentación

<b>Nombre:</b> sp_listado
<b>Tipo de Clase:</b> página servidora
<p><b>Descripción General:</b> La clase sp_listado es una página Web que se ejecuta del lado del servidor. Su actividad fundamental es construir la página cl_listado. Aplica un documento XSLT a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. Es utilizada en todos los casos de uso:</p> <ul style="list-style-type: none"> <li>➤ ObtenerDatosGeneralesÁreaSalud</li> <li>➤ ObtenerInformacionConsolidadaÁreaSalud</li> <li>➤ ObtenerInformacionEBS</li> <li>➤ ObtenerTotalDesgloseAreaSalud</li> </ul>

Tabla 3.3 Descripción de la clase del diseño sp\_listado

**Páginas Servidoras (Server Page) de la Capa de Negocio**

Nombre	RAS_ReporteListaAreasSalud
Tipo de clase	página servidora
Caso de Uso	ObtenerDatosGeneralesÁreaSalud
<b>Parámetros de entrada</b>	
Nombre	Tipo
id_provincia	int
id_municipio	int
id_as	int
cantidad	int
offset	int
nivel	int
tipo_salida	int
<b>Descripción General</b>	
<p>Esta clase es una página Web que se ejecuta del lado del servidor forma parte de la capa de negocio del sistema. Recibe un arreglo de datos con los parámetros de entrada. Esta clase establece una conexión a través de la clase de acceso a datos dbz_class con el servidor de bases de datos y ejecuta las distintas consultas SQL. La responsabilidad principal de esta página es retornar una taxonomía con las provincias, municipios, y aéreas de salud.</p> <p>En dependencia del valor del parámetro de entrada tipo_salida será el formato de la respuesta de este método. Su implementación permite retornar documentos en formatos Adobe Acrobat (PDF) y Microsoft Excel (XLS).</p> <p>Una vez terminadas las consultas SQL, se cierra las conexión con la base de datos</p>	
<b>Parámetros de salida</b>	
Nombre	Tipo
Resultado	Array
<ul style="list-style-type: none"> <li>○ id_unidad</li> </ul>	<ul style="list-style-type: none"> <li>○ int</li> </ul>

o nombre_area	o string
o CodigoArea	o int
o id_municipio	o int
o nombre_municipio	o string
o id_provincia	o int
o nombre_provincia	o string
offset	int
cantidad	int
total	int

Tabla 3.4 Descripción de la clase del diseño RAS\_ReporteListaAreasSalud

Nombre		RAS_ReporteDatosGenerales	
<b>Tipo de clase</b>		página servidora	
<b>Caso de Uso</b>		ObtenerDatosGeneralesÁreaSalud	
<b>Parámetros de entrada</b>			
<b>Nombre</b>		<b>Tipo</b>	
id_provincia		int	
id_municipio		int	
id_us		int	
cantidad		int	
offset		int	
nivel		int	
tipo_salida		int	
<b>Descripción General</b>			
Esta clase es una página Web que se ejecuta del lado del servidor forma parte de la capa de negocio del sistema. Recibe un arreglo de datos con los parámetros de entrada. Esta clase establece una conexión a través de la clase de acceso a datos dbz_class con el servidor de bases de datos y ejecuta			

las distintas consultas SQL. La responsabilidad principal de esta página es retornar un reporte con los datos generales de las aéreas de salud.

En dependencia del valor del parámetro de entrada tipo\_salida será el formato de la respuesta de este método. Su implementación permite retornar documentos en formatos Adobe Acrobat (PDF) y Microsoft Excel (XLS).

Una vez terminadas las consultas SQL, se cierra las conexión con la base de datos

<b>Parámetros de salida</b>	
<b>Nombre</b>	<b>Tipo</b>
Resultado	Array
o id_unidad	o int
o nombre_area	o string
o CodigoArea	o int
o id_municipio	o int
o nombre_municipio	o string
o id_provincia	o int
o nombre_provincia	o string
o ubicacion	o string
o direccion	o string
o km	o string
o nombre_hospitales_base	o array
o nombre_hopital_base	o string
o cantidad_hospitales_base	o int
offset	int
cantidad	int
total	int

Tabla 3.5 Descripción de la clase del diseño RAS\_ReporteDatosGenerales

<b>Nombre</b>		<b>RAS_ReporteInformacionConsolidada</b>
<b>Tipo de clase</b>	página servidora	
<b>Caso de Uso</b>	ObtenerInformacionConsolidadaÁreaSalud	
<b>Parámetros de entrada</b>		
<b>Nombre</b>	<b>Tipo</b>	
id_provincia	int	
id_municipio	int	
id_as	int	
cantidad	int	
offset	int	
nivel	int	
tipo_salida	int	
<b>Descripción General</b>		
<p>Esta clase es una página Web que se ejecuta del lado del servidor forma parte de la capa de negocio del sistema. Recibe un arreglo de datos con los parámetros de entrada. Esta clase establece una conexión a través de la clase de acceso a datos dbz_class con el servidor de bases de datos y ejecuta las distintas consultas SQL. La responsabilidad principal de esta página es retornar un reporte en forma de taxonomía con la información consolidada de las aéreas de salud.</p> <p>En dependencia del valor del parámetro de entrada tipo_salida será el formato de la respuesta de este método. Su implementación permite retornar documentos en formatos Adobe Acrobat (PDF) y Microsoft Excel (XLS).</p> <p>Una vez terminadas las consultas SQL, se cierra las conexión con la base de datos</p>		
<b>Parámetros de salida</b>		
<b>Nombre</b>	<b>Tipo</b>	
Resultado	Array	
○ id_unidad	○ int	
○ nombre_area	○ string	



○ CodigoArea	○ int
○ id_municipio	○ int
○ nombre_municipio	○ string
○ id_provincia	○ int
○ nombre_provincia	○ string
○ Centro	○ array
○ tipo_de_centro	○ string
○ cantidad_centros	○ int
○ Tipo_Construccion	○ array
○ tipo_de_construccion	○ string
offset	int
cantidad	int
total	int

Tabla 3.6 Descripción de la clase del diseño ObtenerInformacionConsolidadaÁreaSalud

Nombre		RAS_ReporteTotalDesglose	
<b>Tipo de clase</b>		página servidora	
<b>Caso de Uso</b>		ObtenerTotalDesgloseAreaSalud	
<b>Parámetros de entrada</b>			
Nombre		Tipo	
id_provincia		int	
id_municipio		int	
id_us		int	
cantidad		int	
offset		int	
nivel		int	

profundidad	int
tipo_salida	int
<b>Descripción General</b>	
<p>Esta clase es una página Web que se ejecuta del lado del servidor forma parte de la capa de negocio del sistema. Recibe un arreglo de datos con los parámetros de entrada. Esta clase establece una conexión a través de la clase de acceso a datos dbz_class con el servidor de bases de datos y ejecuta las distintas consultas SQL. La responsabilidad principal de esta página es retornar en dependencia del parámetro de entrada profundidad, un reporte en forma de taxonomía con la información ya sea en forma de total ó desglose de las áreas de salud. En dependencia del valor del parámetro de entrada tipo_salida será el formato de la respuesta de este método. Su implementación permite retornar documentos en formatos Adobe Acrobat (PDF) y Microsoft Excel (XLS). Una vez terminadas las consultas SQL, se cierra la conexión con la base de datos.</p>	
<b>Parámetros de salida</b>	
<b>Nombre</b>	<b>Tipo</b>
Resultado	Array
o id_unidad	o int
o nombre_area	o string
o CodigoArea	o int
o id_municipio	o int
o nombre_municipio	o string
o id_provincia	o int
o nombre_provincia	o string
o GBT	o array
o id_gbt	o int
o gbt	o string
o cant_ebs	o int
o EBS	o array
o id_ebs	o int

○ ebs	○ string
○ Total_GBT_Areas	○ int
○ Total_EBS_Areas	○ int
○ Total_GBT_Municipios	○ int
○ Total_EBS_Municipios	○ int
○ Total_GBT_Provincias	○ int
○ Total_EBS_Provincias	○ int
offset	int
cantidad	int
total	int

Tabla 3.7 Descripción de la clase del diseño ObtenerTotalDesgloseAreaSalud

### 3.3.6 Otras Clases Utilizadas

Clase	Propósito
<b>dbz_class</b>	Clase utilizada para establecer la conexión con bases de datos MySQL, usa el módulo dbx de PHP para su funcionalidad. Crea un objeto conexión que permite hacer consultas, y recuperar los resultados. Esta clase está en la capa de negocio y forma parte de la plataforma de servicios PLASER
<b>Fachada</b>	Clase general que se encuentra en la plataforma de servicios PLASER, permite aplicar un encapsulamiento a esta librería de clases, creándose de esta forma un nivel de abstracción, logrando una mejor arquitectura del sistema.
<b>FachadaRAS</b>	Clase que hereda de Fachada, se aplica 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 sólo hará las peticiones a la clase Fachada. Es la clase principal de la capa de presentación, contiene las funcionalidades del sistema que permiten acceder a los métodos del negocio. Cualquier solución general del módulo

	debe implementarse en ella.
<b>personal_salud</b>	Almacena los datos del personal de la salud que integra un GBT ó un EBS en el área de salud.
<b>estructuras</b>	Almacena los datos de las estructuras del área de salud (datos comunes a los GBT y EBS).
<b>hospitales_base</b>	Almacena los hospitales base que tiene cada área de salud.
<b>area_salud</b>	Almacena los datos de las áreas de salud.
<b>ubicación_poblacion</b>	Almacena la ubicación relacionada con un área de salud determinada.

Tabla 3.7 Descripción de otras clases

### 3.4 Conclusiones

En este capítulo se describió la estructura arquitectónica propuesta para la solución, describiéndose la realización física de los casos de uso. Se obtuvo el Diagrama de Clases del Análisis donde se representaron las relaciones que se establecen entre las clases: interfaz, control y entidad, que permiten reflejar cada una de las funcionalidades que tiene el sistema. Haciendo uso del Diagrama de Clases del Diseño, se detallaron los elementos de este, describiéndose las clases y sus relaciones. Además constituyó la entrada fundamental para el desarrollo del flujo de Trabajo de Implementación.

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

### 4.1 Introducción

Este capítulo es una secuencia del flujo de análisis y diseño, donde se definieron mediante el uso de los estereotipos Web un conjunto de clases o ficheros que serán tratados en este capítulo como componentes de la aplicación. Se justifica la integración de las Estadísticas Descriptivas del Registro de las Áreas de Salud con otros componentes del Sistema de Información para la Salud (SISalud). Se hace una breve descripción de los servicios Web, métodos o agentes más complejos de la aplicación. Además se presentan los Diagramas de Componentes, el Diagrama de Despliegue y un estudio de los estándares de diseño, codificación y tratamiento de excepciones.

### 4.2 Dependencias y Relaciones con otros Sistemas

La utilización de una arquitectura basada en componentes y orientada a servicios, donde cada módulo es un componente con funcionalidad propia y que utiliza los servicios brindados por otros componentes hacen que el funcionamiento de las funcionalidades Estadísticas Descriptivas del Registro de Áreas de Salud, dependa de otros componentes de SISalud, los cuales se detallan a continuación. La posibilidad de utilizar los servicios Web brindados por otros componentes, garantiza la reutilización de los datos y evita la duplicidad de la información.

#### 4.2.1 Componente de Seguridad (SAAA)

Single Authentication Authorization and Account (SAAA). Este componente tiene implementado un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría. La autenticación es la primera acción del usuario en el sistema y consiste en suministrar un nombre de usuario único y una contraseña. Si el usuario no se encuentra registrado se reportará un error de acceso. Si el usuario autenticado se encuentra registrado se autorizará su acceso y se crea un certificado digital que contiene un identificador único (token) de 32 caracteres, que tiene como información: el identificador del usuario, el nivel de acceso (Nacional, Provincial, Municipal o Unidad de Salud), el identificador de nivel de acceso, un listado de los módulos a los que el usuario tiene acceso y el tipo de acceso que en cada uno de ellos (Editor o Visualizador).

#### **4.2.2 Registro de Unidades de Salud (RUS)**

Este Registro tiene almacenada la información correspondiente a las unidades de salud de todo el país. Brinda un conjunto de servicios Web que facilitan la búsqueda de información relacionada con los procesos de negocio que se gestionan en este codificador. Una de las funcionalidades del RUS es la de relacionar un tipo de unidad de salud con sus atributos (cualitativos o de capacidad), esta relación incluye también la combinación de estos atributos con las unidades de salud que los posean. Los Hospitales Rurales y los Policlínicos son tipos de unidades de salud que contemplan el Programa del Médico y la Enfermera de la Familia, a las mismas se les puede agregar un atributo cualitativo que indique si son o no Áreas de Salud.

Uno de los servicios Web disponibles del RUS y que se utiliza en las Estadísticas Descriptivas del RAS, es el que permite la búsqueda de las unidades de salud. La información que ofrece este servicio se realiza teniendo en cuenta un conjunto de parámetros de entrada como el nombre, la ubicación geográfica (Provincia o Municipio) o los atributos cualitativos de las unidades de salud. El RAS se comunica con el RUS para buscar los datos de las Áreas de Salud y de sus Hospitales Bases. Estas búsquedas se realizan teniendo en cuenta el nivel en que se haya autenticado un usuario en el sistema, ejemplo, si el usuario es del nivel de unidad de salud y la unidad a la que pertenece es un área de salud, se buscarán los datos de dicha unidad en el RUS.

#### **4.2.3 Registro de Localidades (RL)**

El Registro de Localidades, de forma centralizada gestiona la información de los Consejos Populares, Circunscripciones, Zonas CDR y CDR de cada municipio del país. Los servicios que ofrece este componente se utilizan en las Estadísticas Descriptivas del RAS para solicitar los nombres de los Consejos Populares, Circunscripciones y CDR de la población que atiende un Equipo Básico de Salud (EBS) determinado.

#### **4.2.4 Registro de Ubicación (RU)**

El RU es uno de los componentes no médicos del Registro Informatizado de Salud. El mismo gestiona la información de las Provincias, Municipios, Localidades, Calles y Manzanas del país. Los servicios que ofrece este componente se utilizan en las Estadísticas Descriptivas del RAS para solicitar los criterios de búsqueda en las diferentes opciones, relacionados con la información que se gestiona en

RU. También se emplean para introducir las direcciones de los Locales de Consulta que atiende un EBS, además de que dicho EBS podrá conocer las Manzanas que tienen sus poblaciones.

### **4.2.5 Registro de Personal de la Salud (RPS)**

Este componente tiene la funcionalidad de gestionar la información de todos los profesionales de la salud del país, ya sean médicos o no médicos. En las Áreas de Salud se encuentran los Grupos Básicos de Trabajo y Equipos Básicos de Salud. Los mismos están compuestos por especialistas y profesionales de la salud, que ejercen sus funciones para prevenir y resolver los problemas de la salud de la población. El RPS tiene implementado un conjunto de servicios Web que brindan la información que se procesa en este componente. Estos servicios Web son utilizados para las Estadísticas Descriptivas del RAS para obtener la información de los profesionales integrantes de un EBS que opera en las Áreas de Salud.

## **4.3 Implementación**

Para comenzar a desarrollar el flujo de trabajo de Implementación se comienza con el resultado del Diseño y se define como se organizan las clases y objetos en términos de componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación. Uno de los principales propósitos de este flujo de trabajo consiste en desarrollar la arquitectura y el sistema como un todo.

### **4.3.1 Modelo de Implementación**

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

El modelo de implementación de las Estadísticas Descriptivas del RAS expone una organización en capas y jerarquías de paquetes y subsistemas de implementación que contienen componentes y sus relaciones. A continuación se muestra una vista global de la estructura y organización de la implementación del sistema:

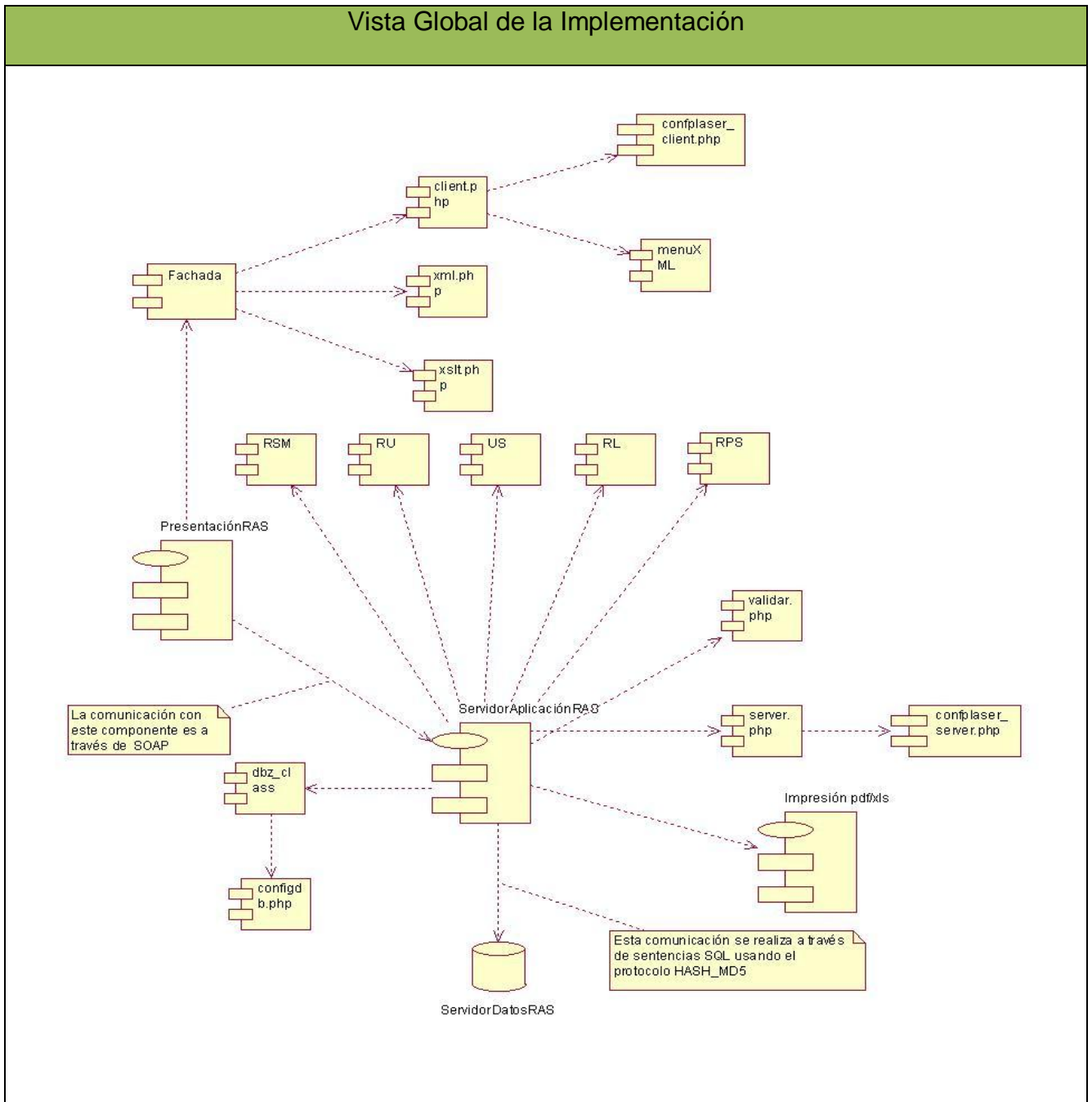


Figura 4.1 Vista Global de la Implementación



#### 4.3.1.1 Diagrama de Componentes

Un componente es una parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces, proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.).

Un diagrama de componentes muestra un conjunto de elementos tales como componentes, subsistemas de implementación y sus relaciones. Permite modelar la gestión de la configuración de los archivos de código fuente y las versiones, así como las bibliotecas.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Un subsistema de implementación es una colección de componentes y otros subsistemas de implementación usados para estructurar el modelo de implementación y dividirlos en pequeñas partes que pueden ser integradas y probadas de forma separada.

El uso más importante de los diagramas de componentes es mostrar la estructura de alto nivel del modelo de implementación, especificando:

- Los subsistemas de implementación y sus dependencias a la hora de importar código.
- Organizar los subsistemas de implementación en capas.

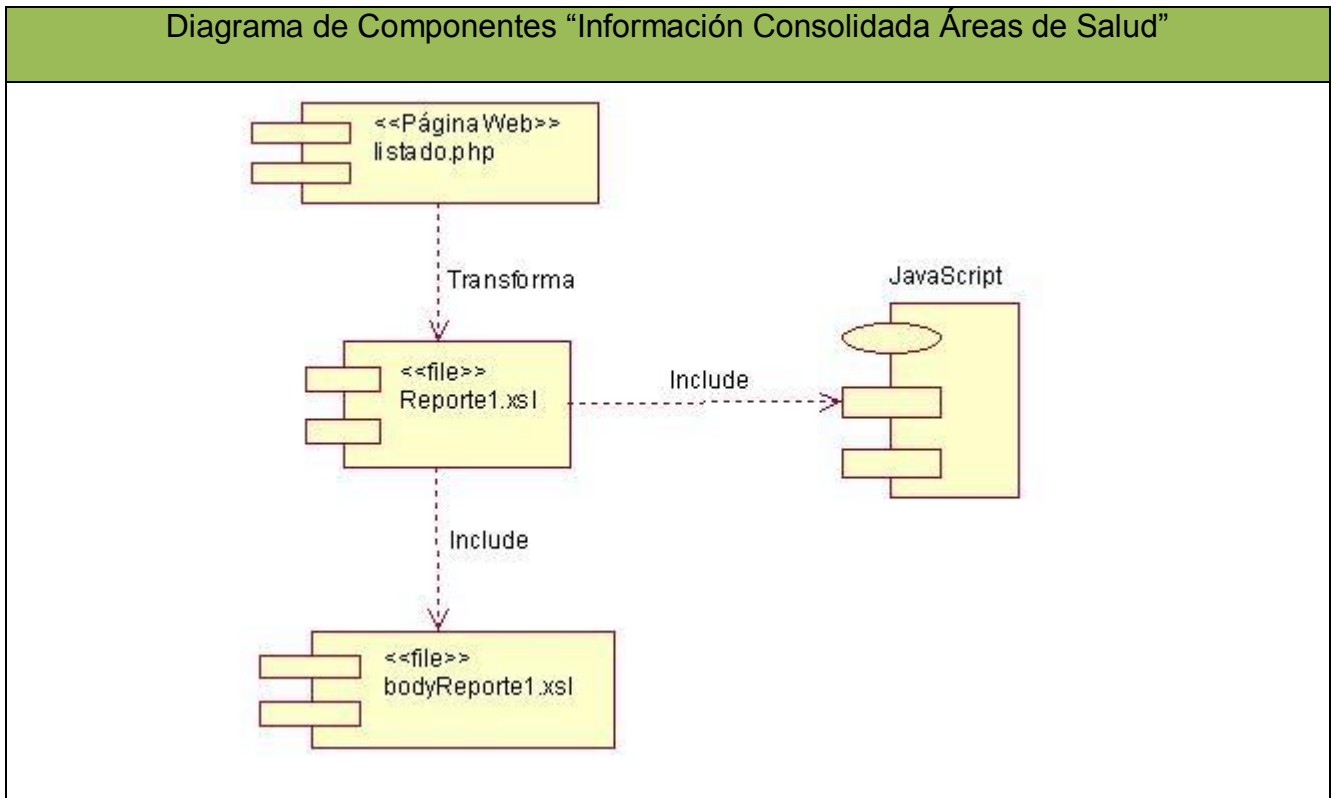


Figura 4.2 Diagrama de Componentes "Información Consolidada Áreas de Salud"

#### 4.3.2 Diagrama 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.

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 topología de esos nodos mediante un diagrama de despliegue.<sup>23</sup>

<sup>23</sup> Piker, M, José. DNS. (2005), [cited; Available from: <http://www.dcc.uchile.cl/~jpiquer/Internet/DNS/node1.html>].

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.

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.

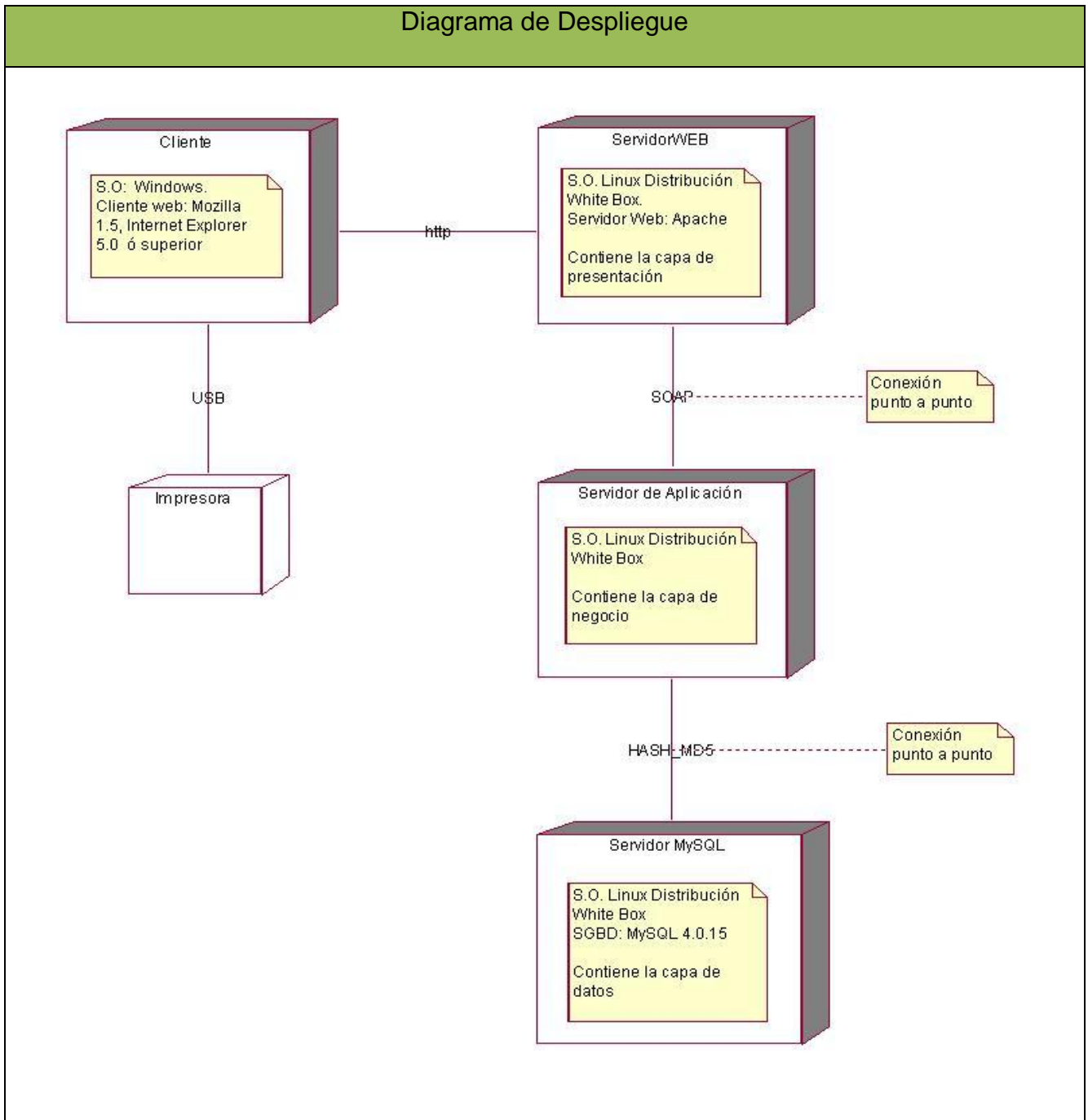


Figura 4.3 Diagrama de Despliegue

### 4.3.3 Descripción de los Métodos del Negocio

A continuación se presentan algunos de los métodos más complejos del negocio de las Estadísticas Descriptivas del Registro de Áreas de Salud:

**RAS\_ReporteInformacionConsolidada:** Este método muestra un reporte con la información consolidada de las áreas de salud. Para su ejecución, el Funcionario del MINSAP interesado en el reporte debe seleccionar los siguientes parámetros: la provincia, el municipio y el área de salud y luego oprimir el botón Mostrar ó Mostrar Todas. El método retorna una taxonomía con los datos en dependencia de los parámetros de búsqueda especificados. Pueden acceder a este, usuarios editores y visualizadores a todos los niveles.

**RAS\_ReporteTotalDesglose:** Este método muestra un reporte con la información de las áreas de salud, ya sea total, desglose ó la combinación de ellas. Para su ejecución, el Funcionario del MINSAP interesado en el reporte debe seleccionar los siguientes parámetros: la provincia y el municipio, además debe crear la profundidad (9 combinaciones) que desee y oprimir el botón Mostrar. El método retorna los datos en dependencia de los parámetros de búsqueda especificados y muestra una pantalla distinta en dependencia de la profundidad creada. Pueden acceder a este, usuarios editores y visualizadores a todos los niveles.

### 4.3.4 Estándares de diseño, codificación y tratamiento de errores

Con el propósito de que exista homogeneidad entre las aplicaciones que se encuentran integradas al Sistema Informatizado de Salud en el proyecto APS se han definido una serie de estándares tanto para el diseño como para la codificación y tratamiento de errores para que los códigos fuentes de las aplicaciones y los mensajes que se emitan mantengan una uniformidad y sean legibles.

#### 4.3.4.1 Estándares de Diseño

Para el diseño de las Estadísticas Descriptivas del Registro de Áreas de Salud se siguieron las pautas y el mismo diseño que el Sistema de Información para la Salud (SISalud); trayendo como principal ventaja la uniformidad en la estructura de las páginas web de todos los módulos integrados a SISalud. Para lograr esta uniformidad se debe de aplicar una hoja de estilo en cascada (Cascading Style Sheets CSS), todas las páginas deben tener una resolución de 800 x 600 px, incluir una serie de imágenes estándares que define el SISalud y seguir una serie de pautas entre las que se encuentran:

- 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 a la izquierda de la página (como usualmente se hace) para ampliar el espacio de trabajo, pues estará reservado lo más amplio posible para la inserción de grandes tablas y formularios que constituyen la base fundamental de estas aplicaciones.
- El logo siempre estará ubicado en el extremo superior izquierdo de la página, es una imagen que cuenta con un ancho de 270 px y se corresponde con el nombre de cada módulo. Estará constituido por un juego tipográfico en Frankling Gothic Medium, y en el caso de las aplicaciones propias del Proyecto APS, estando especificado dentro del logo como una especie de genérico.
- 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.
- 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 ptos 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.

#### **4.3.4.2 Estándares de Codificación**

Con el propósito de distribuir los esfuerzos y mejorar los rendimientos de la aplicación, se utilizó la programación orientada a servicios, creando clases genéricas que permiten la definición y distribución de las llamadas a los procedimientos de los módulos distribuidos y la definición de los métodos propios de cada módulo.

Actualmente se encuentran estándares de codificación para la mayoría de los lenguajes existentes. El uso de los mismos, 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 CODIG Standard), que mayormente está basada en el estándar de código para aplicaciones en C++ (C++ CODIG Standard).

- Las etiquetas de apertura y cierre del lenguaje serán de la forma `<?php ?>`, ya que siempre están disponibles en cualquier configuración.
- Se hará 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 lenguaje español, utilizando como separador para las palabras el carácter “\_”, tratando de usar nombres sugerentes a la acción de la variable.
- Todos los campos identificadores van a comenzar con el identificador (id) seguido del nombre del campo. Ejemplo: `id_unidad`.
- Los arreglos empezarán con el identificador array y las palabras no se separarán con el carácter “\_”. Ejemplo: `Arrayunidades`.
- Las estructuras se identificarán poniendo al final del nombre struct. Ejemplo: `paginadostruct`.
- En el caso de las clases se pondrá delante la letra C. Ejemplo: `CFachada`.
- El idioma de las clases auxiliares como sesión y error, será el inglés para garantizar la homogeneidad con las programadas en este ámbito en el mundo, en el caso de los Servicios Web y la interface de administración se usará el español para esclarecer los objetivos de cada método o script a utilizar.
- En los métodos no se usarán abreviaturas y las palabras continuas deben comenzar con mayúsculas. Ejemplo: `ReporteInformacionEBS`.
- Cada método que se defina debe seguir la siguiente estructura, acrónimo del módulo. Nombre del método. Ejemplo: `RAS_ReporteInformacionConsolidada`.
- Para comentar el código se utilizará, en el caso de una línea, al final de la misma el carácter `//` y seguido el comentario y en el caso de un bloque se utilizará los caracteres `/* */`.



- Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas. Ejemplo: `SELECT * FROM tb_area_salud`.
- Las páginas HTML se harán sin incluir código y todas las funciones JavaScript que se usarán se escribirán dentro de ficheros “.js”.
- Los controles seguirán el siguiente tratamiento:

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

Tabla 4.1 Estándares para los controles

#### 4.3.4.3 Tratamiento de Errores

El tratamiento de los errores es un paso fundamental para el buen funcionamiento de un sistema, por tanto, desde inicio de desarrollo del mismo se realizan operaciones y se cumplen tareas para evitar la ocurrencia de estos.

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 de error. Cuando se verifica un error se pone en marcha una excepción que, si se recibe enseguida, permite gestionar un error.

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

Proporcionan una manera de verificar los errores y poder controlarlos si fuera el caso si abortar el código.

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 mas estructurado, legible, robusto y fácil de mantener.

Para emitir los diferentes reportes se tiene en cuenta que la información debe estar bien actualizada en la Base de Datos y se crean las condiciones en el sistema para que la información que no esté disponible no pueda ser emitida. Otros errores pueden ser generados por el gestor de Base de Datos, estos se capturan antes de mostrarse al usuario y una vez tratados por el sistema, son mostrados al usuario en una manera entendible para él.

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

#### 4.4 Prueba

El desarrollo de las pruebas es una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

El Flujo Trabajo de Prueba tiene entre sus principales objetivos encontrar y documentar los defectos que puedan afectar la calidad del software, validar que el software trabaje como fue diseñado, validar y probar los requisitos que debe cumplir el software, validar que los requisitos fueron implementados correctamente. Algunos de los métodos de pruebas mas importantes son: el método de caja negra y el de caja blanca.

En la primera actividad *Planificar Pruebas* se obtuvo como artefacto de salida el Plan de Pruebas para la primera iteración, definiéndose en cada fase del desarrollo las pruebas que iban a ser diseñadas y ejecutadas. En la segunda actividad, *Diseñar Pruebas* se diseñaron pruebas fundamentalmente de caja negra teniendo como punto de partida los requerimientos funcionales y su descripción a través del modelo de sistema, logrando así un correcto y completo diseño de los casos de pruebas. Además se diseñaron pruebas con el fin de probar el correcto y estricto cumplimiento de los estándares definidos

por el proyecto tanto de diseño como de implementación. Por último se ejecutaron todos los diseños de casos de pruebas.

Durante el desarrollo de este flujo de trabajo se verificó el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a los terceros.

A continuación se describen algunos casos de prueba de integración el resto de las pruebas realizadas se encuentran en el expediente de proyecto:

<b>Caso de Uso: ObtenerDatosGeneralesÁreaSalud</b>	
<b>Entrada</b>	Es necesaria la realización del Caso de Prueba porque a la hora de seleccionar los parámetros de entrada para realizar el Reporte, deben de aparecer solamente todas las provincias en un combobox, luego al elegir una de estas provincias se deben cargar en otro combobox los municipios que pertenezcan a esta, y al elegir un municipio se deben cargar en un último combobox las áreas de salud correspondientes a este. Además de que se debe verificar que la opción de Exportar a: funcione correctamente tanto para las extensiones pdf como xls.
<b>Resultados</b>	Que se carguen los combobox correctamente, en dependencia de lo que el usuario vaya seleccionando.  Realizar la impresión de acorde a las solicitudes realizadas por el usuario.
<b>Condiciones</b>	Entre las condiciones que deben cumplirse cuando se ejecute el caso de prueba es que tanto el llenado de los combobox como la impresión se realicen de forma correcta.

Tabla 4.2 Descripción Caso de Prueba ObtenerDatosGeneralesÁreaSalud

<b>Caso de Uso: ObtenerTotalDesgloseAreaSalud</b>	
<b>Entrada</b>	Es necesaria la realización del Caso de Prueba porque a la hora de seleccionar los parámetros de entrada para realizar el Reporte, deben de aparecer solamente todas las provincias en un combobox, luego al elegir una de estas provincias se deben cargar en otro combobox los municipios que pertenezcan a esta. También existe otro parámetro de entrada llamado profundidad el cual se crea seleccionando varios checkboxes y radiobuttons, donde esta profundidad debe ser una combinación de checkboxes y radiobuttons válida. Además de que se debe verificar que la opción de Exportar a: funcione correctamente tanto para las extensiones pdf como xls.
<b>Resultados</b>	Que se carguen los combobox correctamente, en dependencia de lo que el usuario vaya seleccionando.  Mostrar un mensaje de alerta indicando que: <ul style="list-style-type: none"> <li>✓ Debe crearse una profundidad.</li> </ul> Realizar la impresión de acorde a las solicitudes realizadas por el usuario.
<b>Condiciones</b>	Entre las condiciones que deben cumplirse cuando se ejecute el caso de prueba es que tanto el llenado de los combobox como la impresión se realicen de forma correcta. Además de que se muestre la ventana de alerta.

Tabla 4.3 Descripción Caso de Prueba ObtenerTotalDesgloseAreaSalud

## 4.5 Conclusiones

En el capítulo se mostraron los resultados de la etapa de construcción del sistema. En el proceso de implementación se cumplieron los principios de diseño y estándares de interfaz, elementos que ayudan a una mejor comunicación con el usuario, así como los estándares de implementación. Como culminación al diseño se presentó el modelo de implementación describiendo la distribución física del sistema y sus componentes. Además se realizaron las pruebas al sistema para evaluarlo y determinar la calidad del mismo.

### CONCLUSIONES

Luego de dar cumplimiento a los objetivos propuestos y tareas trazadas, se obtuvieron las siguientes conclusiones:

- Se demostró que la gestión de la información estadística es una funcionalidad necesaria del Registro de Áreas de Salud, que es útil para la toma de decisiones en los diferentes niveles de dirección.
- Se realizó un estudio detallado del proceso de gestión de la información estadística en el Sistema Nacional de Salud.
- Se modelaron los flujos de trabajo: Modelamiento de Negocio, Requerimientos, Diseño e Implementación; quedando completa la documentación de Ingeniería de Software.
- Se investigó sobre sistemas existentes a nivel nacional e internacional, que tuvieran relación con el problema planteado y pudieran resolverlo, realizando un análisis comparativo entre los mismos.
- Se estudiaron las tendencias, tecnologías y herramientas a utilizar para el desarrollo de la solución propuesta.
- Se diseñaron e implementaron las funcionalidades del Registro de Áreas de Salud que brindan reportes estadísticos, cumpliendo con los requerimientos trazados y con la arquitectura definida por el MINSAP.

### RECOMENDACIONES

Al concluir el presente trabajo y habiendo cumplido los objetivos propuestos, se recomienda:

- Desplegar las Estadísticas Descriptivas del Registro de Áreas de Salud, en SISalud para que sus servicios puedan ser utilizados por cualquiera que los necesite.
- Implementar el resto de los reportes estadísticos correspondientes al Registro de Áreas de Salud.
- Continuar el perfeccionamiento constante de las Estadísticas Descriptivas del Registro de Áreas de Salud, para mejorar el rendimiento y rapidez del procesamiento de la información.
- Conformar y aplicar un plan de adiestramiento para el personal involucrado en el trabajo con las Estadísticas Descriptivas del Registro de Áreas de Salud.

**BIBLIOGRAFÍA**

JAMES RUMBAUGH, G. B., IVAR JACOBSONI. *El Proceso Unificado de Desarrollo de Software*. 2000. La Habana, Editorial Félix Varela, 2004.

Sistema de Información Estadística Complementario de Salud SIEC. Modelos e Instrucciones.

De la OSA, J. A. D. L. La semilla del desarrollo de la salud pública en Cuba. [Disponible en: <http://www.granma.cubaweb.cu/marti-moncada/pro07.html>].

ORTEGA VALDÉS, A., RAMÍREZ RODRÍGUEZ, Y., Análisis del Registro de Áreas de Salud de la Atención Primaria del Sistema Nacional de Salud. Universidad de las Ciencias Informáticas: Ciudad de La Habana. 2007.

GUDÁS, D. M.N. Infomed, red de Salud de Cuba. [Disponible en: <http://www.infomed.sld.cu>].

PRESMAN, R. S. Ingeniería del software. Un Enfoque práctico. La Habana, Editorial Félix Varela, 2005.

Anuario Estadístico de Salud 2004.

EKKA. Tecnología a su alcance. Aplicaciones Web. 2007. [Disponible en: <http://www.ekka.cl/aplicaciones.htm>].

DELGADO RAMOS, A. Presentación Informatización del Sistema Nacional de Salud, Dirección Nacional de Registros Médicos y Estadísticas de Salud MINSAP, 2006. p.

Web Services Activity. [Disponible en: <http://www.w3.org/2002/ws>].

VIDAL LEDO, M. Información, informática y estadísticas de salud: un perfil de la tecnología de la salud, 2004. Disponible en: [http://www.bvs.sld.cu/revistas/aci/vol12\\_4\\_04/aci08404.htm](http://www.bvs.sld.cu/revistas/aci/vol12_4_04/aci08404.htm).

MARÍN DÍAZ, M. E. Manual de Fundamentos del Sistema Nacional de Salud Pública en Cuba. 2006b.

MARÍN DÍAZ, M. E. Consideraciones sobre el Proyecto de Informatización de la Atención Primaria de Salud. Revista Habanera de Ciencias Médicas. 3. 2004.

LARGMAN, C. UML y patrones, 2004.

Informatización de la sociedad cubana. [Disponible en:

<http://www.monografias.com/trabajos14/informatica-social/informatica-social.shtml#inf>.

RODRÍGUEZ OSORIA, D., PANEQUE MESCHENKOV, A. L., SARMIENTO BORRERO, Y. R., Diseño y Servicios Web para el Registro de Áreas de Salud de la Atención Primaria del Sistema Nacional de Salud. Universidad de las Ciencias Informáticas: Ciudad de La Habana. 2007.

CANTÚ MARTÍNEZ, P. C. El Valor de la Estadística para la Salud Pública. [Disponible en:

<http://www.respyn.uanl.mx/iv/1/ensayos/bioestadistica.html>.

Wikipedia. Arquitectura Orientada a Servicios. [Disponible en:

[http://es.wikipedia.org/wiki/Arquitectura\\_orientada\\_a\\_servicios](http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios).

LEÓN GARCÉS, J. Descripción de la Plataforma de Servicios PlaSer. 2008.

RODAS HINOSTROZA, R. Características de PHP. [Disponible en:

<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.

LÓPEZ GONZÁLEZ, T. Análisis del Registro de Enfermedades de Declaración Obligatoria. Universidad de las Ciencias Informáticas: Ciudad de La Habana. 2007.

ANGEL ALVAREZ, M., Evaluando Zend Studio. 2003. [Disponible en:

<http://www.maestrosdelweb.com/editorial/zendstudio>.

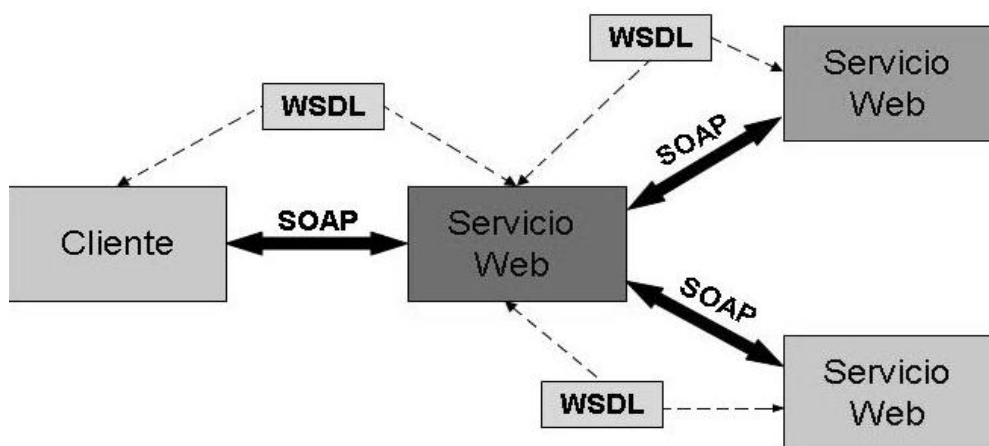


## ANEXOS

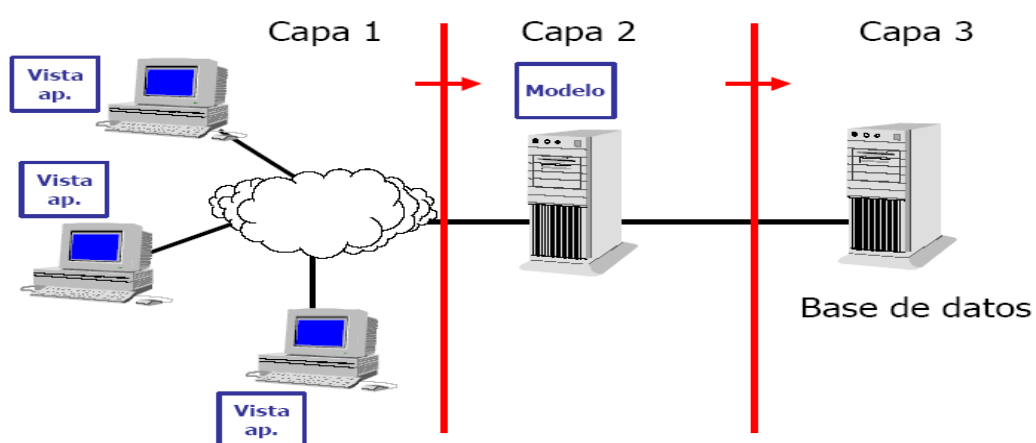
## Anexo I: Organigrama del Sistema Nacional de Salud Cubano.



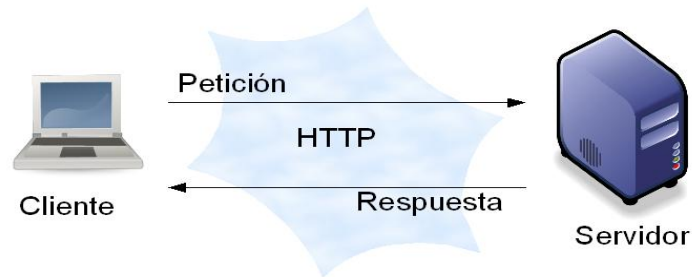
Anexo II: Esquema que representa la interacción de un conjunto de Servicios Web.



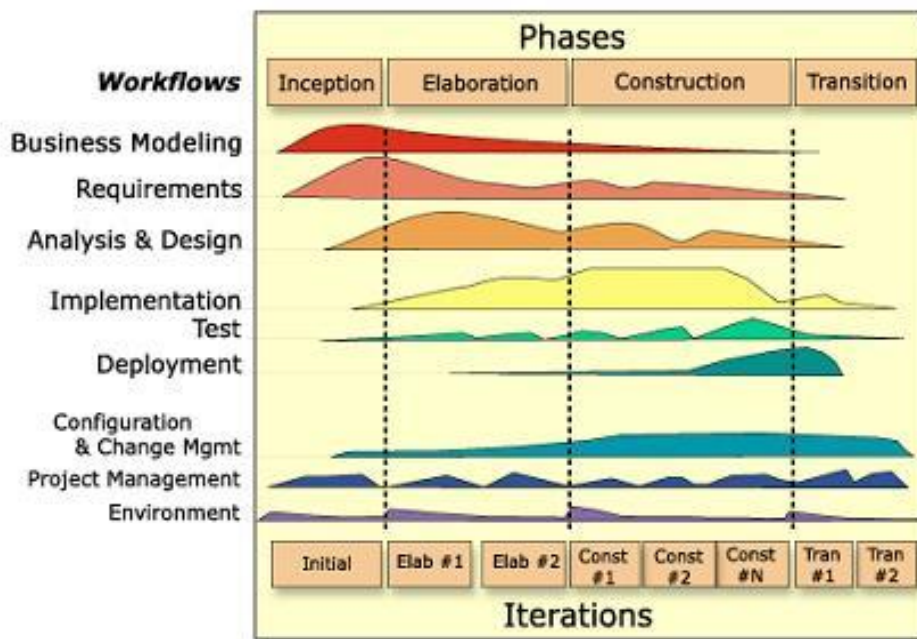
Anexo III: Esquema que representa la arquitectura en tres capas.



#### Anexo IV: Esquema que representa el Modelo Cliente-Servidor.



#### Anexo V: Fases e Iteraciones de la Metodología RUP



### GLOSARIO DE TÉRMINOS

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

**Aplicaciones legacy (Aplicaciones legadas, heredadas):** Cualquier aplicación basada en tecnologías y hardware más viejo que continúa brindando servicios esenciales a una organización.

**APS (Atención Primaria de Salud): Atención Primaria de Salud:** Nivel asistencial que constituye la puerta de entrada del paciente al Sistema Nacional de Salud, donde debe darse solución alrededor del 90% de los problemas que afectan a la población. En este nivel se realizan acciones educativas, curativas y de rehabilitación.

**Área de Salud:** Área geográfica a la que presta sus servicios una Unidad de salud que contemple el Programa de Trabajo del Médico y la Enfermera de la Familia. Tienen el nombre del Policlínico.

**Cargo:** Se refiere a la labor que desempeña un trabajador de un centro de trabajo.

**Centro:** Lugar en que se desarrolla la actividad social fundamental u ocupación (retribuida o no) de un individuo. En términos del Registro de Áreas de Salud se refiere a la clasificación de los tipos de poblaciones, limitados a estos lugares, que reciben los servicios del Equipo Básico de Salud. Se clasifica en Centro de trabajo o laboral y Centro estudiantil.

**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.

**CSS (Cascading Style Sheets):** Las hojas de estilo en cascada (Cascading Style Sheets) son un lenguaje formal de ordenador usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

**Departamento:** Se aplica a cada una de las divisiones internas de una Unidad de Salud clasificadas según preste, o no, servicio directo a la población.

**DHTML (Dynamic HTML):** Posibilidad que permitía el Internet Explorer de modificar la página actual desde JavaScript.

**Estadística Descriptiva:** Es un paso inicial y obligatorio en cualquier tipo de análisis estadístico, analiza metódicamente los datos, simplificándolos y presentándolos en forma clara; eliminando la confusión característica de los datos preliminares.

**Estructura:** Organización del personal del Área de Salud que atiende a la población. Puede ser Equipo Básico de Salud, Grupo Básico de Trabajo.

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

**Grupo Básico de Trabajo (GBT):** Equipo de trabajo multidisciplinario integrado por un grupo de EBS (entre 15 y 20), por especialistas de Medicina Interna, Gineco-obstetricia, Pediatría, Licenciado en Psicología y de MGI, todos en función de interconsulta y de profesores, por una enfermera supervisora, un técnico de higiene y epidemiología y un técnico en Trabajo Social.

**Hospital Base:** Unidad de Salud del nivel de Atención Secundaria (generalmente Hospitales) que se responsabiliza de manera integra con las Unidades de Salud de la Atención Primaria (Áreas de Salud) para el enfrentamiento de los problemas de salud de una determinada población que no se pueden manejar en el nivel primario por ser de mayor complejidad en el diagnóstico, el tratamiento y la rehabilitación, (integración Policlínico-Hospital). Existe un Hospital Base por cada especialidad básica: Medicina Interna, Ginecología y Obstetricia y Pediatría.

**Infomed:** Red telemática del MINSAP.

**Locales:** Este término ha sido usado en este documento con similar significado a **Consultorio**. Un local es una entidad que puede ser para Consultorio (local de trabajo de los Equipos Básicos de Salud) o para ser vivienda de los especialistas que integran el Equipo Básico de Salud, es por ello que los locales de Consulta pueden o no tener asociados hasta dos locales de viviendas.

**Minsap:** Ministerio de Salud Pública de Cuba.

**Población:** En términos del Registro de Áreas de Salud, se refiere al tipo de grupos de individuos que reciben los servicios de un Equipo Básico de Salud (pacientes) delimitado por su situación geográfica donde realiza sus interacciones con otros individuos. Se clasifica en Comunidad, Centro (Laboral o Educativo), etc.

**Policlínico:** Es la unidad básica de la APS que brinda servicios médicos ambulatorios a una población territorial definida, con servicios cercanos hacia el lugar que reside, trabaja y estudia esa población y acorde a las necesidades de salud identificadas.

**Registro Área de Salud (RAS):** Es un registro del Sistema Integrado de Gestión del Sistema Nacional de Salud Pública, el cual gestiona y controla los Tipos de Estructura, Tipos de Locales, Tipo de Población, Ubicación de Población, Tipo de Centros, Tipos de Departamentos, Cargos, Plantillas y además las Área de Salud país.

**Servicio Web:** Colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

**SOAP (Simple Object Access Protocol):** Estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los Servicios Web.

**Software:** Conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

**Subsistema:** Agrupación de elementos, de los que algunos constituyen una especificación del comportamiento ofrecido por los elementos contenidos.

**WSDL (Web Services Description Language):** El lenguaje de descripción de servicios Web WSDL permite aprovechar las ventajas derivadas de SOAP al proporcionar un medio de colaboración sencillo entre los proveedores de servicios Web y los usuarios de los mismos.

**XML (Extensible Markup Language):** Es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados.