

Universidad de las Ciencias Informáticas

Facultad 7



**Título: Diseño e Implementación del
Registro de Fallecidos**

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

Autores: Yenny Leal González
Ariuska Hernández Sablón
Fidel Enrique Castro Dieguez

Tutores: Lic. Yamilka Gómez León
Ing. Yosvani Turruelles Tejeda

Asesora: Lic. Juana I. Pérez Rodríguez

Ciudad de La Habana, Julio del 2008

“Año 50 de la Revolución”

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 firmamos la presente a los 7 días del mes de Julio del año 2008.

Yenny Leal González

Firma de la Autora

Ariuska Hernández Sablón

Firma de la Autora

Fidel Enrique Castro Dieguez

Firma del Autor

Lic. Yamilka Gómez León

Firma de la Tutora

Ing. Yosvani Turruelles Tejeda

Firma del Tutor

“Crear es la palabra de pase de esta generación.”

José Martí

“El revolucionario verdadero está guiado por grandes sentimientos de amor.”

Ernesto Che Guevara

*“Yo parto de una confianza grande que este ser humano,
con todos sus defectos y limitaciones, tiene capacidades suficientes para preservarse
y tiene inteligencia suficiente para mejorarse.”*

Fidel Castro Ruz

Datos de Contacto

Tutores:

Lic. Yamilka Gómez León: Profesor graduado de Licenciatura en Ciencias de la Computación en el año 2004 en la UCLV. Ha impartido las asignaturas Programación 2, Sistemas de Bases de Datos y Gestión de Software. Posee la categoría docente de Instructor y cursa la maestría de Ciencias de la Computación en la UCI. Forma parte del proyecto de Atención Primaria de Salud de la facultad 7.

Correo electrónico: yamilkagl@uci.cu

Ing. Yosvani Turruelles Tejeda: Graduado como Ingeniero en Ciencias Informáticas en el año 2007 en la Universidad de Ciencias Informáticas (UCI). Posee la categoría docente de Instructor Recién Graduado. Forma parte del proyecto Atención Primaria de Salud de la facultad 7.

Correo electrónico: yturruelles@uci.cu

Asesor:

Lic. Juana Isabel Pérez Rodríguez: Graduado de Licenciado en Educación en la Especialidad de Lengua Inglesa en el año 2000 en el ISPEJV en la Facultad de Lenguas Extranjeras. Posee categoría docente de Profesor Instructor Principal y cursa en estos momentos una maestría en la Universidad de La Habana. Ha impartido las asignaturas Idioma Extranjero I y II en la Facultad 7 desde el curso 2004-2005. Ha presentado ponencias en eventos científicos nacionales e internacionales. Se desempeña como Jefa de la Asignatura en la propia facultad.

Correo electrónico: jane@uci.cu

Agradecimientos

Por darnos la oportunidad de formarnos como ingenieros dignos de esta época, conectados al futuro, pero por sobre todo, conectados a la Revolución, gracias revolucionarios todos de esta isla y del mundo entero por creer y luchar por un mundo mejor.

A nuestro líder indiscutible, Fidel Castro, por dedicar más de medio siglo a una causa que para algunos es imposible, utópica, pero para muchos es confianza, altruismo, respeto y libertad, por ser el creador de esta maravillosa universidad y por depositar su confianza en cada uno de nosotros, a nuestro Comandante en Jefe, este trío integrará por siempre su Tropa de Futuro.

A nuestros padres. Sin ustedes no existiéramos, como no puede crecer una planta sin agua y sol, por apoyarnos en cada momento y por estar ahí cuando los necesitamos, gracias.

No solo fueron transmisores de contenidos ni programas de estudios, fueron padres, amigos, compañeros, eternos compartidores de conocimientos, motivadores por excelencia por extraer lo máximo que podíamos dar en nuestros quehaceres, gracias profesores.

Aunque ha estado lejos, continuó supervisando y velando por cada uno de nuestros movimientos, gracias tutora Yamilka.

Aunque ha tenido muchas responsabilidades en este curso, nos acogió como si hubiera sido tutor desde el principio, por ser uno de los recién graduados, ejemplo nuestro, gracias tutor Yosvani.

Ese hombre siempre dispuesto a apoyar nuestra frente, esa mano cálida que se tendía cada vez que se necesitaba, sus ayudas constantes, sus ojos y alma transmitiendo confianza y valor, gracias eternos amigos.

A nuestros compañeros del proyecto, gracias por su ayuda y apoyo incondicional en estos años de trabajo: Yoelvis, Ariel, Yanet, Mairénys, Aymé, Daylén, Yosle y los profesores de este equipo de trabajo que es APS. Además a Vanesa, Rubén, Alexis y Jose por contribuir en el desarrollo de esta aplicación. A los especialistas de Softel, gracias por su apoyo incondicional y su influencia día a día en nuestra correcta formación: Mirna Cabrera, al doctor Denis, el profesor Juan Carlos Pujols y Carmenchun.

Yenny: Por su profunda y eterna amistad, gracias Yutmila, Yasu, Yanoska, Yazmín, Yeny, Marisel, Naty, Yanet, Yasmín Caridad, Yanio, Portales, Adamis, Amet y a todos con los que he compartido estos años de estudio.

Ariuska: A mis amigas de la infancia y siempre: Dania y Arasay. Yaima y Mildrey a quienes considero mis hermanas. También a las niñas: Yeni, Ori, Lissy, Yako y Yairi y con los que he vivido mis momentos de estudios.

Fidel: De manera muy especial a mis queridos amigos durante mi vida estudiantil: Félix, Dixson, Yoanky, Annerys, Kenia, Mare, Yanet, Alex, Ana y Wilkie.

Dedicatoria

Por ser mi faro máspreciado y la razón de mi existir, a ti mi mani adorada, Esperanza, va dedicado este trabajo.

A mi estrella, por tanta entrega, dedicación y al que le debo la fortaleza en mi carácter, a mi padre Braulio.

A mi abuelita Carmen querida por tanto amor depositado. Para ti Jorge todo mi cariño. A toda mi familia por velar y confiar plenamente en mí.

A mis amigos que han sido como hermanos todos estos años. A mis compañeros de tesis por dedicar horas en el trabajo colectivo.

A profesores abnegados y a todos los que creen en esta universidad.

A todos los que sueñan y luchan por un mundo de paz.

Yenny

A mi abuela Tita y a mi mamá Mirian que han sido padre y madre al mismo tiempo.

A mi papá Enrique, que desde niño mi sueño siempre fue ser como él.

A Yamira, que es mi mejor amiga y compañera de los 5 años de carrera. No sólo eres mi esposa, también me das las fuerzas para enfrentar este largo camino que es la vida.

A mis hermanas Mabel y Susana, a mi madrastra Elena, a mi familia, a la de mi esposa que me han ayudado mucho.

A mis dos compañeras de tesis que sin ellas no hubiese sido posible esta tesis.

Fidel

A mis padres: Rafael y Milagros, por su amor y dedicación. Por enseñarme a luchar cada día, por mis sueños. Porque son y serán mis modelos a seguir en la vida. Por sus consejos y paciente escucha.

A Rosa, a quien quiero como una madre y de quien siempre he tenido apoyo y cariño.

A mis seis hermanos, por el amor y el apoyo que siempre he encontrado en ellos.

A toda mi familia. A todos los quiero mucho.

A mis dos compañeros de tesis, porque sin ellos este trabajo no hubiese sido posible.

Ariuska

Resumen

El presente trabajo, persigue diseñar y desarrollar la aplicación web, Registro de Fallecidos (RF), permitiendo la gestión de los Certificados Médicos de Defunción (CMD) y facilitar el flujo de información de la mortalidad en Cuba. La gestión de los CMD se realiza de forma manual, los datos son recogidos en papel y el procesamiento de la información es muy engorroso, cuando el volumen de documentos es considerable. Como consecuencia, los Departamentos de Estadísticas de Salud, principalmente a nivel nacional, no posean una información completa y actualizada a corto plazo. El registro desarrollado es un módulo del Sistema de Información para la Salud (SISalud).

En el sistema se modelaron los flujos de trabajo de Análisis y Diseño, Implementación y Prueba utilizando la metodología de desarrollo brindada por el Proceso Unificado de Desarrollo (RUP) y el lenguaje de modelado visual (UML). El registro se implementó con tecnologías actuales como: PHP, XML, XSL, MySQL y Apache. Además de utilizar herramientas como el Zend Studio, SQL Manager 2007 for MySQL, Rational Rose, Dreamweaver y Style Studio; siguiendo las decisiones sobre arquitectura y tecnologías propuestas por el MINSAP.

El Registro de Fallecidos permitirá informatizar la gestión de los tres tipos de Certificados Médicos de Defunción, así como de las Solicitudes de Reparos y los Reparos correspondientes. Así como, disponer de un registro para el control de la información sobre los fallecidos en los diferentes niveles de dirección. Además de brindar y obtener información de otros componentes de SISalud.

PALABRAS CLAVE

Registro de Fallecidos, Certificado Médico de Defunción, Departamentos de Estadísticas de Salud, niveles de dirección del Sistema Nacional de Salud, Solicitudes de Reparos y Reparos.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	7
1.1 Informatización del Sistema Nacional de Salud.....	7
1.2 Informatización del Sistema Nacional de Salud.....	9
1.3 Situación Problemática.....	11
1.4 Objetivo de Automatización.....	13
1.5 Estado del Arte.....	14
1.6 Tecnologías y Tendencias a considerar.....	16
1.6.1 Arquitectura de Software.....	16
1.6.2 Servidor Web Apache.....	20
1.6.3 Plataforma de Servicios PlaSer.....	20
1.6.4 Lenguajes de Programación Web.....	20
1.6.5 Sistemas de Gestión de Bases de Datos.....	25
1.6.6 Metodologías y lenguajes de Desarrollo de Software.....	25
1.6.7 Definición de Tecnología y Herramientas a utilizar.....	27
1.6.8 Herramientas a utilizar.....	27
1.6.9 Servicios Web.....	28
CAPÍTULO 2. DISEÑO DEL SISTEMA.....	30
2.1 Workflow (Flujo de Trabajo).....	30
2.1.1 Workflow del proceso de gestión de los CMD.....	30
2.1.2 Diagrama de Estado.....	31

2.2 Modelo del Diseño.....	34
2.3 Justificación del uso de los Patrones.....	34
2.3.1 Alta Cohesión.....	34
2.3.2 Bajo Acoplamiento.....	35
2.3.3 Fachada.....	35
2.3.4 Proxy.....	35
2.3.5 Data Access Objects (DAO: Objetos de Acceso a Datos).....	36
2.3.6 Modelo Vista Controlador (MVC).....	36
2.4 Definición de la estructura del diseño.....	36
2.4.2 Estereotipos Web.....	38
2.5 Diagrama de Clases del Diseño.....	40
2.6 Descripción de las Clases del Diseño.....	56
2.6.1 Descripción de páginas clientes.....	56
2.6.2 Descripción de páginas servidoras.....	58
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA.....	66
3.1 Dependencias y Relaciones con otros Sistemas.....	66
3.1.1 Registro de Ubicación (RU) y Registro de Localidades (RL).....	66
3.1.2 Registro de Partos y Nacimientos (RPN).....	66
3.1.3 Registro de Población (RPOB).....	66
3.1.5 Registro de Unidades de Salud (RUS).....	67
3.1.6 Registro de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud (RCIE).....	67
3.1.7 Sistema Autorización, Autenticación y Auditoría (SAAA).....	67

3.1 Implementación.....	68
3.1.1 Diagrama de Despliegue.....	68
3.1.2 Modelo de Implementación.....	69
3.1.3 Diagrama de Componentes.....	70
3.2 Descripción de los Métodos (agentes o métodos más complejos).....	73
3.3 Estándares del Diseño, Codificación y Tratamientos de Errores.....	75
3.3.2 Estándares de Diseño.....	75
3.3.2 Estándares de Codificación.....	79
3.3.3 Tratamiento de Errores.....	81
3.4 Pruebas.....	82
CONCLUSIONES.....	86
RECOMENDACIONES.....	87
REFERENCIAS BIBLIOGRÁFICAS.....	88
BIBLIOGRAFÍA.....	90
ANEXOS.....	92
GLOSARIO DE TÉRMINOS.....	98

INTRODUCCIÓN

La informática en el mundo ha tenido un crecimiento vertiginoso, lo cual, unido al auge de las Tecnologías de la Información y las Comunicaciones, ha garantizado que en ninguna esfera económica o social se pueda pensar en el desarrollo sin la utilización de esta herramienta. Los avances en la informática y las comunicaciones han conducido a la 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 toma de decisiones. El sector de la Salud no está ajeno a ello y se trabaja en la informática médica, la telemedicina y la automática, con fórmulas que mejoran la calidad de los servicios brindados a la población.

Desde hace varias décadas la informática ha encontrado en la medicina una de sus aplicaciones más comunes e importantes. Esto ha permitido al sector de la salud; no sólo contar con métodos novedosos, sencillos y eficaces de gestión administrativa en consultas, hospitales y centros de investigación biomédica, sino también disponer de recursos informáticos de gran valor en las exploraciones con tratamiento de imágenes, señales bioeléctricas y otras que reducen la posibilidad de error en el diagnóstico de las enfermedades, facilitan el trabajo de las estadísticas médicas y aumenta la confiabilidad de la misma.

La utilización de estos recursos ahorra tiempo al personal, que puede ser empleado en el análisis de la información. Así como en la toma de decisiones y la ejecución de las mismas, en vez de limitarse a la preparación y compilación de los datos. También ofrecen una gran ayuda en el campo de la investigación epidemiológica, farmacéutica, biológica, química, etc., aspectos todos ellos relacionados con la lucha para conseguir un mejor nivel de salud en las personas y la excelencia en los servicios a la población.

En Cuba el Estado ha asumido totalmente la responsabilidad de la atención a la salud de sus ciudadanos y en tal sentido lleva a cabo acciones económicas y médicas para asegurar la protección de la salud, partiendo de los principios de gratuidad y accesibilidad total a los servicios con la calidad requerida.

Como aspecto importante de todo el proceso de informatización en que se encuentra sumergida la Revolución Cubana, el Ministerio de Salud Pública (MINSAP), órgano rector del Sistema Nacional de

Salud (SNS), unido a la dirección de la Revolución, han trazado los primeros pasos para la reorientación del SNS en la Atención Primaria de Salud (APS), proyecto con el que se persigue implementar un Sistema Integral de Informatización del SNS que abarque todos los niveles. (1)

El SNS está considerado un sistema complejo, en cuyas relaciones internas y externas descansan los procesos que apoyan la aplicación de planes, programas y acciones que garantizan la consecución de los objetivos encaminados a garantizar la salud del pueblo. (2) El mismo está estructurado en tres niveles básicos de atención relacionados a continuación:

- Atención Primaria de Salud: constituye el nivel donde se efectúa el primer contacto del sistema de salud con los individuos, la familia y la comunidad. Abarca a todos los Equipos Básicos de Salud (EBS), los policlínicos y hospitales rurales a través del Programa de Medicina Familiar.
- Atención Secundaria de Salud: se brinda a nivel de las instituciones hospitalarias que por lo general son de carácter provincial, y por tanto atienden a toda la población de una provincia determinada. En el mismo se llevan a cabo acciones de salud más complejas y especializadas de acuerdo a las distintas especialidades existentes.
- Atención Terciaria de Salud: es aquella que por su condición muy especializada brinda servicios de gran complejidad, para lo cual dispone de recursos humanos muy capacitados y especializados, con tecnologías más desarrolladas para diagnósticos y tratamientos. A este nivel pertenecen los Institutos y Hospitales Especializados, por ejemplo: Servicios de Neurocirugía, Nefrología, Cirugía Cardiovascular, etc. Incluye además los Centros e Institutos de Investigaciones.

Paralelamente al SNS, el país cuenta con un Sistema de Información Estadística Complementaria del Sector de la Salud (SIEC de la salud). Este está constituido por 73 subsistemas de información, entre los que se encuentra el Subsistema de Defunciones; cuenta además con 23 reportes de información adelantada o partes como el de Mortalidad Materna y Mortalidad Infantil.

Uno de los objetivos trazados por el MINSAP es mejorar la calidad y seguridad en la gestión de la información que se genera en los departamentos de estadísticas en las diferentes direcciones del país.

Esto es posible, perfeccionando los sistemas estadísticos que se utilizan en la actualidad, con el desarrollo de nuevas aplicaciones informáticas que brinden reportes estadísticos con información oportuna.

El MINSAP y el Ministerio de la Informática y las Comunicaciones (MIC), en el año 2003 le asignan a SOFTEL, empresa cubana dedicada al desarrollo aplicaciones informáticas para la salud, la misión de informatizar los procesos y servicios del Sistema Nacional de Salud. Para lograr este objetivo se han creado varios grupos de trabajo, con el propósito de informatizar los tres niveles de atención médica, definiéndose las estrategias a seguir y asumiendo el diseño y puesta en marcha de manera gradual, de un sistema de gestión de la información y el conocimiento de nuevas dimensiones: el Sistema de Información para la Salud (SISalud).

La organización de este proyecto ha sido concebida mediante el empleo de una estrategia que antes no se había utilizado en un proceso de desarrollo, donde participan médicos y trabajadores de la salud en su calidad de expertos funcionales en estrecho vínculo con los especialistas de informática. Además participa la Universidad de las Ciencias Informáticas (UCI), quien a través de una organización de trabajo en proyectos productivos, ha vinculado a estudiantes y profesores al desarrollo de software de todo tipo en conjunto con empresas y otras entidades.

Como parte de este proyecto se concibe el SISalud, infraestructura en la cual deberán integrarse todos los sistemas o soluciones que se conciban, con el fin de dar cumplimiento al programa general para informatizar el SNS. Mediante esta infraestructura se pretende abordar los problemas del sector de forma integral e innovadora; utilizando la Arquitectura Orientada a Servicios (SOA) y la Arquitectura Basada en Componentes (CBA), satisfaciendo las necesidades de los clientes en todos los niveles de salud.

SISalud está compuesto por los módulos de los diferentes niveles de atención médica: primaria, secundaria y especializada, y por el Registro Informatizado de Salud (RIS) que incluye los nomencladores nacionales que brindan servicios a las demás aplicaciones informáticas como el Registro de Unidades de Salud (RUS), Registro de Áreas de Salud (RAS), entre otros, para obtener una información integral en tiempo real que facilite la toma de decisiones en los diferentes niveles de dirección, la docencia, la investigación y la gestión en salud. Uno de estos módulos es el Registro de Fallecidos.

La muerte misma no puede ser evitada, sin embargo, se puede postergar. La importancia para la salud pública de este hecho ha motivado que uno de sus aspectos a velar con atención es el referente a la mortalidad de la población; de ahí el interés de mejorar la gestión de la información que actualmente se realiza en el SNS. El dato primario dentro de este proceso lo constituye el Certificado Médico de Defunción (CMD), el cual es el modelo oficial y el registro básico del sistema de información estadístico de mortalidad del país, además de ser el documento jurídico que avala el fallecimiento y los derechos civiles que de ello se derivan.

Actualmente la gestión de los Certificados Médicos de Defunción se realiza de forma manual y semiautomática resultando en ocasiones engorroso este proceso para el personal médico que lo realiza. La información es recogida en modelos confeccionados en papel, material que con el paso del tiempo se deteriora, implicando pérdida de información importante para futuros estudios personalizados y estadísticos que se deseen obtener en el país.

Debido a la cantidad de CMD que se generan anualmente, la acumulación de todos estos documentos implica que el proceso de análisis sea complejo cuando el volumen de documentos es considerable. Esta situación conlleva a que los Departamentos de Estadísticas en todos niveles, principalmente a nivel nacional, no posean una información completa y a corto plazo de cada una de sus estructuras inferiores.

El flujo de la información de los CMD resulta muy complicado, pues esta debe transitar por los diferentes niveles del Sistema Nacional de Salud, desde las unidades de salud donde son registrados los CMD hasta el nivel nacional. Este no es un proceso que pueda realizarse en las condiciones actuales instantáneamente para garantizar la disponibilidad de la información de manera inmediata en los niveles superiores, trayendo consigo que a la hora de realizar estudios de mortalidad u otras acciones en el país no se disponga de la información actualizada en cualquier momento del año de manera rápida.

En el país actualmente se cuenta con un sistema automatizado capaz de procesar la información de los CMD, el Sistema Automatizado para el Control de la Mortalidad. Este sistema está dividido en dos subsistemas, el MORTAPro para el registro a nivel provincial y el MORTANac en el nacional. También es utilizado para las publicaciones estadísticas y con fines investigativos en las diferentes especialidades médicas del sector de la salud. Su implementación está realizada en FoxPro 2.6^a, no permite el trabajo en

red, y como los registros de la mortalidad sólo pueden ser digitalizados en la provincia o en la nación, los datos obtenidos en los reportes estadísticos no son los más actualizados. La política de seguridad del sistema no está bien diseñada para las necesidades de un sistema confiable, porque no permite inspeccionar y controlar el acceso de los usuarios que acceden a la aplicación.

Teniendo en cuenta lo expuesto anteriormente y la necesidad de buscar una respuesta a estas dificultades que frenan el desempeño de los especialistas de la salud encargados de estudiar las causas de muerte en un grupo poblacional o en una región determinada, surge el tema que da origen a este trabajo donde los esfuerzos estarán encaminados a solucionar el siguiente **Problema a Resolver**: ¿Cómo mejorar la gestión de los Certificados Médicos de Defunción en el Sistema Nacional de Salud?

Se define como **Objeto de Estudio** el Proceso de gestión de la información en el Sistema Nacional de Salud, enfocándose el **Campo de Acción** en el Proceso de gestión de la información de los Certificados Médicos de Defunción en el Sistema Nacional de Salud.

Como **Objetivo de la Investigación** para solucionar los problemas mencionados se plantea el siguiente: Implementar el Registro de Fallecidos para mejorar la gestión de los Certificados Médicos de Defunción en el Sistema Nacional de Salud, garantizando la disponibilidad inmediata de la información en todos los niveles de atención médica del país.

Para dar cumplimiento al Objetivo General planteado se realizarán las siguientes **Tareas de la Investigación**:

1. Analizar los sistemas o aplicaciones relacionados con el tema existente en la actualidad en Cuba y el mundo.
2. Asimilar el estudio realizado para el Registro de Fallecidos.
3. Asimilar la Arquitectura definida por el MINSAP orientada a servicios y basada en componentes para el desarrollo de sus aplicaciones.
4. Asimilar la Plataforma de Servicios PlaSer.

5. Identificar arquitectónicamente la integración con otros componentes del Sistema de Información para la Salud.
6. Realizar el Modelo de Diseño a partir del Modelo de Análisis ya existente.
7. Modelar el Flujo de Trabajo “Implementación” y “Prueba”.
8. Implementar el Registro de Fallecidos.

Como antecedente fundamental a esta investigación, se desarrolló el Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas “Análisis del Registro de Fallecidos para el Sistema de Información para la Salud” realizado en el curso 2006-2007, de los autores Yunaisi Renté Vázquez y Alain García de la Paz. En este se modeló el Flujo de Trabajo “Análisis y Diseño”, realizando el Modelo de Análisis, del que se parte para realizar el Modelo de Diseño de esta investigación, culminando así el Flujo de Trabajo descrito por RUP.

El presente documento está estructurado en tres capítulos:

Capítulo 1: Fundamentación Teórica. En este capítulo se brinda una breve explicación de la fundamentación del tema, justificándose las tecnologías, metodologías y herramientas actuales que fueron utilizadas.

Capítulo 2: Diseño del Sistema. Describe los aspectos relacionados con el diseño de la solución propuesta. Haciendo uso de estereotipos Web, se presentan los diagramas de clases del diseño de algunos casos de uso del sistema, refinando el análisis existente.

Capítulo 3: Implementación y Prueba. Se presenta la integración entre los componentes, se modela el diagrama de componentes y despliegue y se describen los estándares de diseño, codificación y tratamiento de excepciones en la solución del sistema, así como los casos de prueba realizados.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se abordan los distintos aspectos que se utilizan como soporte técnico del sistema a diseñar e implementar. Se expone la organización del Sistema Nacional de Salud a través de conceptos descritos, así como el proceso de informatización de la salud. Se incluye también la situación problemática, los antecedentes, y la descripción de las herramientas, la metodología y los lenguajes de programación que se valoraron como propuesta y los que finalmente se utilizaron.

1.1 Informatización del Sistema Nacional de Salud.

El Sistema Nacional de Salud en Cuba desde su creación a principios de la Revolución, ha resultado un proyecto social en sí mismo para garantizar la equidad, accesibilidad y universalidad que requieren todos los ciudadanos.

Está estructurado en tres niveles, que se corresponden con la estructura político-administrativa del país:

- Nivel Nacional, representado por el MINSAP, como órgano rector con funciones metodológicas, normativas, de coordinación y de control, al cual se le subordinan directamente los centros universitarios, los institutos de investigaciones, centros hospitalarios de asistencia médica altamente especializados, centros de distribución y comercializadoras de suministros y tecnologías médicas, así como otros centros y entidades nacionales destinados a actividades técnicas y de apoyo.
- Nivel Provincial, representado por las Direcciones Provinciales de Salud, directamente subordinadas administrativa y financieramente a la Asamblea Provincial del Poder Popular.
- Nivel Municipal, representado por las Direcciones Municipales de Salud dependientes, administrativa y financieramente, de la Asamblea Municipal del Poder Popular. (3)

Los servicios médicos se prestan en forma escalonada según el grado de complejidad de las unidades y atendiendo a los tres niveles de atención médica se clasifican en: Atención Primaria, Atención Secundaria y Atención Terciaria.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Dentro de los principales objetivos del SNS están: perfeccionar la capacidad del personal médico de la salud, garantizar la introducción de tecnologías de punta en los programas de salud y elevar la calidad de los servicios que brindan a la población, mediante estrategias dirigidas al fortalecimiento de la Atención Primaria.

Por la importancia que representa para la sociedad, es que se decide llevar a cabo la informatización del SNS, teniendo como principal objetivo implementar un Programa General de Informatización, cuyo centro sería el policlínico, apoyando las estrategias y políticas trazadas por la dirección del país y el MINSAP; de forma que se logre la incorporación progresiva y sistemática de las Tecnologías de la Información y las Comunicaciones (TIC) en función de la adquisición y gestión del conocimiento y los servicios de salud.

Como parte de este proceso de informatización, el sector de la salud se propone una solución informática integral que permita la gestión de una información confiable en tiempo real y unificado, y que aporte la rapidez y fiabilidad necesarias para la toma de decisiones en los diferentes niveles del SNS. Para lograrlo se pretende que todos los registros estén incluidos en un conjunto de aplicaciones que formarán parte del Sistema de Información para la Salud: el Registro Informatizado de Salud (RIS), el Sistema Informatizado de Atención Especializada (SIAE), el Sistema Informatizado de Atención Primaria (SIAP) y el Sistema Informatizado de Gestión Hospitalaria (SIGH).

El Registro Informatizado de Salud (RIS), está integrado por un conjunto de componentes independientes que se relacionan según las necesidades del flujo de información, permitiendo además la interoperabilidad y el acceso de la información desde los diferentes niveles de dirección. Contiene los nomencladores nacionales que brindan información para el funcionamiento del resto de las aplicaciones, citando como ejemplo: el Registro de Ubicación Geográfica (RU), el Registro de Equipos Médicos (REM), entre otros.

En el Sistema Informatizado de Atención Especializada se agruparán los módulos que pertenecen al Nivel de Atención Terciaria o Especializada.

En el Sistema Informatizado de Gestión Hospitalaria se agruparán los módulos que pertenecen al Nivel de Atención Secundario u Hospitalario.

El Sistema Informatizado de Atención Primaria, contempla los módulos propios de la Atención Primaria de

la Salud (APS), tales como el Registro de Actividades Diarias (RAD), el Registro de Población (RPOB), entre otros.

Todo este proyecto de informatización fue encomendado a SOFTEL, empresa que en el año 2003 reorienta su trabajo y se especializa en Soluciones Informáticas para la Salud con la misión de proporcionar 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. (4) La empresa es ubicada en la Infraestructura Productiva (IP) de la Universidad de las Ciencias Informáticas (UCI) y se vincula a estudiantes y profesores de la Facultad 7 que se especializa en el desarrollo de software relacionado con la salud. De esta integración surge el Proyecto Atención Primaria de la Salud que encamina sus objetivos a analizar, diseñar y desarrollar un producto de software para dar respuesta a los procedimientos establecidos para el Nivel de Atención Primaria.

1.2 Informatización del Sistema Nacional de Salud.

La informatización de la Atención Primaria de Salud es la aplicación gradual e integral de las Nuevas Tecnologías de la Información y las Comunicaciones en la gerencia de los procesos relacionados con las funciones del SNS a ese nivel de atención. Ya desde hace años se han estado produciendo y poniendo en práctica soluciones informáticas para la automatización de algunos procesos administrativos, docentes, investigativos y asistenciales. (5)

En el proceso de informatizar la APS ocurren cambios que favorecen la transmisión rápida y efectiva de la información clínica y administrativa, relativa a todas las actividades de los policlínicos y las unidades de la Atención Primaria. Se pretende además, que las instituciones del país alcancen un elevado nivel de informatización de las actividades que brindan, y así lograr un incremento en la calidad de los servicios que se presten a la población. Todo esto posibilita aplicar eficientemente las acciones de control y vigilancia y los programas de salud existentes.

El conocimiento del volumen de una determinada población y de su composición en función de unas determinadas características, fundamentalmente edad y sexo, en un momento dado, proporciona una información estática, es decir, una instantánea de la misma. Ahora bien, la población está en continua

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

evolución, originada por una serie de factores de cambio que inciden sobre ella: nacimientos, defunciones, matrimonios y movimientos migratorios. La estadística del movimiento natural de la población, que proporciona información sobre los principales fenómenos demográficos -nacimientos, defunciones y matrimonios-, junto con la estadística de variaciones residenciales, permite estudiar la evolución y las características de una población a lo largo del tiempo.

El Registro de Fallecidos es uno de los módulos del Sistema Informatizado de la Atención Primaria y contribuye a conocer el aspecto sanitario del movimiento natural de la población, y la calidad o desarrollo alcanzado en la salud pública. Este registro se realiza captando y almacenando los datos según las necesidades de información del Sistema Nacional de Salud y de la Organización Mundial de la Salud (OMS) y para la codificación de la mortalidad, se tiene en cuenta los criterios establecidos por la Clasificación Internacional de Enfermedades (CIE) vigente, dictada por la OMS. Los datos son almacenados en los Certificados Médicos de Defunción, documento oficial y jurídico, además de ser el registro básico del sistema de información estadístico de mortalidad del país.

Existen tres tipos de certificados, relacionados a continuación:

Certificado Médico de Defunción Fetal: para defunciones de 500 gramos o más, o 22 semanas y más de gestación. Este original es destinado al MINSAP y solamente posee fines estadísticos. **Ver Anexo # 1:** Modelo 8 - 110 (18-38-2).

Certificado Médico de Defunción Neonatal: se confecciona para nacidos vivos que fallezcan entre 0 y 27 días de vida. Este modelo consiste en dos originales, uno destinado al MINSAP, con fines estadísticos y otro destinado a las oficinas del Registro del Estado Civil. **Ver Anexo # 2:** Modelo 8 – 110 (18-38).

Certificado Médico de Defunción para fallecidos de 28 días y más: este modelo consiste en dos originales, uno destinado al MINSAP, con fines estadísticos y otro destinado a las oficinas del Registro del Estado Civil. Es el de mayor utilización en el país y donde se registran la mayor cantidad de errores en su llenado, principalmente en la sección referida a los datos sobre la certificación de las causas de muerte. **Ver Anexo # 3:** Modelo 8 – 111 (18-39-2).

Al llenar correctamente estos modelos permite obtener información fiable y actualizada en los informes

estadísticos, facilitando el trabajo de los funcionarios que se ocupan del procesamiento informativo para la producción de estadísticas sanitarias del Sistema de Información Estadística Complementaria (SIE-C) del sector de la salud que comienzan en el policlínico hasta la Dirección Nacional de Estadísticas del MINSAP.

Con la gestión de estos datos, se pueden lograr establecer líneas de conducta para la reducción de la morbilidad y mortalidad, la realización de acciones para diagnósticos precoces, trabajar en grupos de riesgos en prevención primaria y hacer comparaciones con períodos anteriores y con situaciones de otras regiones del país o en el exterior.

Indudablemente la informática ha venido a jugar un papel importante en la medicina, tanto en la gestión administrativa de consultas, hospitales, centros de investigaciones como en la recopilación y procesamiento de los datos estadísticos, encontrándose lo suficientemente extendida en los sistemas de información existentes, lo que ha permitido darle un vuelco cualitativo a estas actividades, fundamentalmente en la prestación de servicios sobre la labor estadística.

1.3 Situación Problémica.

Cuba cuenta con un sólido Sistema de Información Estadístico de Salud que brinda información numérica oportuna y confiable. Una de las informaciones estadísticas vitales que provee es la mortalidad, usadas en el proceso de toma de decisiones de diferentes sectores del país. En las tasas de mortalidad, el sexo, la edad, la causa de muerte, el lugar de residencia y otras características socio-económicas de la persona fallecida, continúan siendo la piedra angular de esta información, pero el análisis de estas tasas aún es laborioso. Se debe tener en cuenta que la Calidad de las Estadísticas de Mortalidad depende en primer término de la calidad del llenado del Certificado Médico de Defunción, cualquier error o tendencia negativa en torno a su llenado se traduciría en una desviación en el comportamiento de las estadísticas.

Existe en el país un sistema informático capaz de procesar la información de los CMD, el Sistema Automatizado para el Control de la Mortalidad en su versión 1.0, utilizado en los niveles Provincial y Nacional para validar, registrar, y emitir los resultados referentes a la mortalidad del país. Este sistema está dividido en dos subsistemas: el MORTAPro, donde se registran los Certificados Médicos de Defunción de las diferentes unidades asistenciales que pertenecen a la provincia, y proporciona al nivel nacional estas informaciones mediante el correo electrónico. Las defunciones ocurridas en una provincia,

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

que tengan como residencia del fallecido otra provincia diferente, también son enviadas a las provincias correspondientes, a través del correo electrónico.

El otro subsistema es el MORTANac, que aprueba y procesa todas las informaciones enviadas por las provincias, editándola según lo demanda el nivel nacional para posteriores publicaciones e investigaciones.

Independientemente de las facilidades que presentan estos dos sistemas informáticos, no satisfacen las necesidades que en la actualidad demanda el SNS: disponibilidad y acceso inmediato de la información relacionada con los fallecimientos ocurridos.

En la gestión estadística de mortalidad aún existen inconvenientes que no permiten obtener una información exacta:

- ✓ La gestión de la información asentada en los Certificados Médicos de Defunción se realiza de forma manual y semiautomática resultando en ocasiones engorroso este proceso para el personal médico que lo realiza. La información es recogida en modelos confeccionados en papel material que con el paso del tiempo se deteriora y puede implicar una pérdida de información importante para futuros estudios personalizados y estadísticos que se quieran obtener en el país.
- ✓ Debido a la cantidad de CMD que se generan anualmente, la acumulación de todos estos documentos implica que el proceso de análisis sea complejo cuando el volumen de documentos es considerable. Esta situación conlleva a que los Departamentos de Estadísticas en todos niveles, principalmente a nivel Nacional, no posean una información completa y a corto plazo de cada una de sus estructuras inferiores.
- ✓ El flujo de la información de los Certificados Médicos de Defunción resulta muy complicado pues esta debe transitar por los diferentes niveles del Sistema Nacional de Salud, desde las unidades hasta el nivel Nacional. Este no es un proceso que pueda realizarse en las condiciones actuales instantáneamente para garantizar la disponibilidad de la información de manera inmediata en los niveles superiores, trayendo consigo que a la hora de realizar estudios de mortalidad u otras acciones en el país no se disponga de la información actualizada en cualquier momento del año

de manera inmediata.

- ✓ El registro de la mortalidad solo puede ser digitalizado en los niveles Provincial (Dirección Provincial de Salud) con el MORTAPro y Nacional (Dirección Nacional de Estadísticas del MINSAP) con el MORTANac, no siendo esto posible en las unidades de salud o Departamentos Municipales de Estadísticas.
- ✓ En los niveles inferiores, díganse las unidades de salud o Departamentos Municipales de Estadísticas la información es trasladada por una persona del Departamento de Estadísticas (lo más común) o un mensajero, mientras que en la provincia y la nación los datos del MORTAPro son enviados al MORTANac en tablas por medio del correo electrónico.

1.4 Objetivo de Automatización.

Analizando los problemas existentes en el país referente a la gestión de los Certificados Médicos de Defunción (28 días y más, Neonatal y el Fetal), se propone el desarrollo de una aplicación web que contenga las funcionalidades necesarias que permitan procesar los datos del fallecido una vez creado el certificado. Esta aplicación será capaz de:

- ✓ Gestionar la información referente a los codificadores necesarios en el proceso de gestión de los CMD, es decir, Listar, Agregar, Modificar o Eliminar grupos de edades, vida conyugal, confirmación de causas, certificación, muertes violentas y sitio de defunción.
- ✓ Gestionar un Certificado Médico de Defunción de cualquiera de los tres tipos existentes, posibilitando también la realización de búsquedas dinámicas por diferentes criterios de los que ya fueron creados, en función del nivel y permiso del usuario de la aplicación.
- ✓ Gestionar una Solicitud de Reparación a un CMD una vez detectado un error en el mismo después de realizarse una búsqueda y revisión de este, en cualquiera de los niveles a partir del municipal.
- ✓ Realizar un Reparación a un CMD al cual se le haya realizado una Solicitud de Reparación por parte de las instancias superiores, estando disponible esta opción solo en el nivel unidad de salud que es donde se gestiona el dato primario.

- ✓ La aplicación brindará servicios a las aplicaciones integradas a SISalud que necesiten información de los fallecidos, así mismo consumirá servicios de las aplicaciones que contienen los datos necesarios para gestionar de forma completa la información de los Certificados Médicos de Defunción.
- ✓ La aplicación simulará el recorrido de cada CMD por todo el flujo establecido en los diferentes niveles, desde la unidad de salud hasta el nivel nacional.

1.5 Estado del Arte.

Ámbito Internacional

En las búsquedas realizadas se ha constatado la existencia en España de una red de vigilancia epidemiológica. Mediante el Decreto 184/1996, del 19 de diciembre, se crea en ese país la Red de Vigilancia Epidemiológica de la Comunidad de Madrid, que se define como el conjunto de sistemas de información que, desde una óptica poblacional y de Salud Pública son imprescindibles para la vigilancia epidemiológica de la morbilidad, mortalidad y factores de riesgo. Posee alrededor de 12 sistemas que componen la red, entre ellos se encuentra el Sistema de Información sobre Mortalidad.

El Sistema de Información de Mortalidad se incluye en lo que se denominan las estadísticas del Movimiento Natural de la Población (MNP), que contiene la información de nacimientos, matrimonios, y defunciones procedente del aprovechamiento estadístico de las inscripciones practicadas por los individuos en el Registro Civil de cada municipio. Este los envía mensualmente a la Delegación Provincial del Instituto de Estadística (INE) que a su vez se los proporciona al Instituto de Estadística de la Comunidad de Madrid, donde se ubica la gestión del registro de Mortalidad.

Este sistema a pesar de que permite recoger datos correspondientes a las causas de muerte que provienen de los Certificados Médicos de Defunción, el proceso de gestión no se corresponde con lo establecido por el Sistema Nacional de Salud de Cuba, pues esta red fue desarrollada siguiendo las características del SNS de España. Además de que los modelos de certificado de defunción no incluyen en su contenido datos como: fecha del último embarazo, suicidio, homicidio, accidente, entre otros datos que sí son contemplados por el Ministerio de Salud Pública y el Sistema de Información Estadístico

Nacional de Cuba. La secuencia para registrar las causas de muerte en los CMD es diferente a como los contemplan los modelos de los certificados españoles. **Ver Anexo # 4:** Certificado Médico Defunción (España).

Ámbito Nacional

Está establecido en Cuba el uso del Sistema Automatizado para el Control de la Mortalidad en su versión 1.0, el cual está dividido en dos subsistemas, MORTAPro y MORTANac ubicados en la Direcciones Provinciales de Salud y la Dirección Nacional de Estadísticas del MINSAP respectivamente. La aplicación permite almacenar todos los datos de los Certificados Médicos de Defunción con o sin reparos asociados, codificar las causas de muertes según la Codificación Internacional de Enfermedades en el nivel provincial, validar los errores, así como generar un gran volumen de reportes estadísticos cada vez que el usuario lo crea necesario.

A pesar de las funcionalidades que presenta tiene algunas limitaciones, que provocan la necesidad de este trabajo. Es una aplicación de escritorio sin conectividad, y está implementado en Microsoft FoxPro 2.6^a para Windows, herramienta para crear y administrar componentes y aplicaciones de bases de datos. La interfaz de las pantallas es poco creativa y atractiva, no permite controlar el acceso y mantener una traza de los usuarios que interactúan con la aplicación, haciéndolo un sistema poco seguro. No trabaja en régimen multiusuario existiendo solo un usuario definido para acceder a la misma.

Debido a la importancia que tiene la gestión de los Certificados Médicos de Defunción en el proceso de informatización de la información estadística de mortalidad, se estableció como objetivo del trabajo de diploma que precede al presente (curso 2006-2007), el desarrollo de las fases del modelamiento del negocio, requerimientos y análisis del registro de fallecido. Permitiendo conocer el ambiente o entorno en que el negocio se enmarca, determinando los procesos que desarrolla y las personas involucradas.

Además, de identificar las funcionalidades requeridas y las restricciones que se imponen para el funcionamiento del sistema a realizar. Partiendo del análisis modelado anteriormente, el presente trabajo afronta la tarea de diseñar e implementar el sistema propuesto, el cual permitirá agilizar el proceso de gestión de dichos certificados, de los cuales depende la calidad con que se lleve a cabo la Información de Estadísticas de Mortalidad en el país.

1.6 Tecnologías y Tendencias a considerar.

1.6.1 Arquitectura de Software.

La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones. (6)

1.6.1.1 Arquitectura Basada en Componentes.

La arquitectura basada en componentes consiste en uno o un número pequeño de componentes específicos de la aplicación (que se diseñan específicamente para ella), que hacen uso de otros muchos componentes prefabricados que se ensamblan entre sí para proporcionar los servicios que se necesitan en la aplicación.

En la arquitectura de componentes la interfaz constituye el elemento básico de interconectividad. Sus componentes deben describir de forma completa las interfaces que ofrece, así como las interfaces que requiere para sus operaciones. Sus características más relevantes son la modularidad, la reusabilidad y compatibilidad y en todos ellos se manifiesta la tecnología orientada a objetos. El desarrollo de software basado componentes es la evolución natural de la ingeniería software para mejorar la calidad, disminuir los tiempos de desarrollo y gestionar la creciente complejidad de los sistemas.

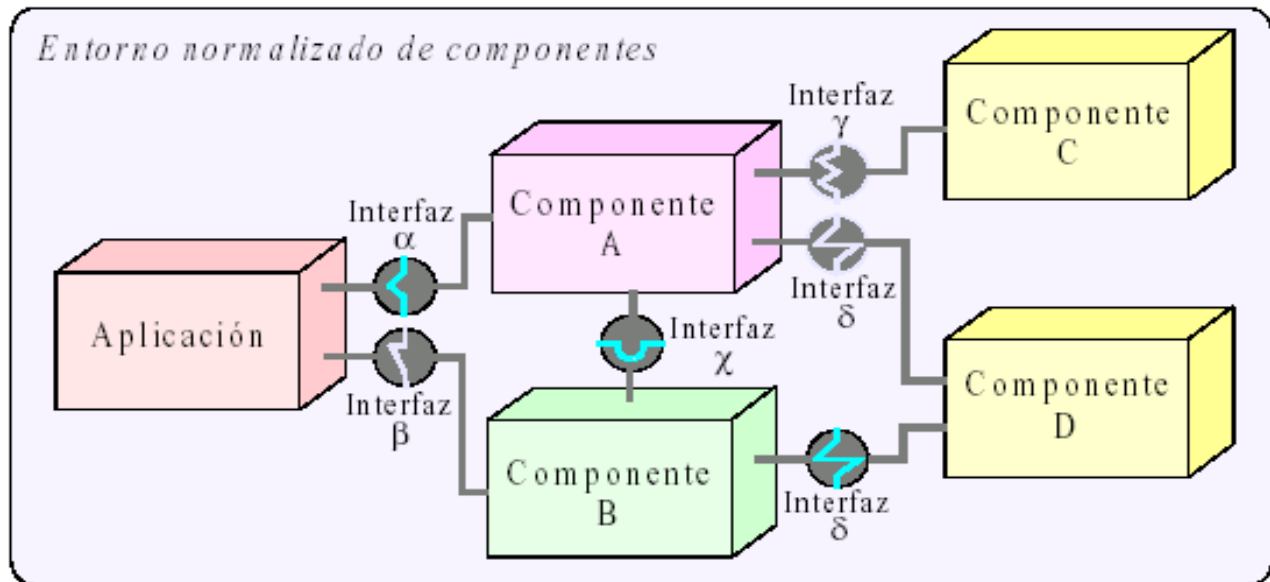


Figura 1. Arquitectura Basada en Componentes.

1.6.1.2 Arquitectura Orientada a Servicios.

SOA (Service Oriented Architecture) nace como una estrategia de integración, exponiendo servicios con funcionalidades bien definidas a la aplicación que lo requiere. De esta manera, una aplicación final, simplemente se provee de un conjunto de estos servicios, añade su lógica particular y le presenta una interfaz al usuario final. Una visión interna de los servicios es que los mismos son como aplicaciones independientes, teniendo sus propias reglas del negocio, datos, procedimientos, operaciones y administración. Exponen toda su funcionalidad basada en una interfaz a través de mensajes, lo que implica la carencia de una interfaz de usuario. Uno de sus elementos básicos es su flexibilidad. Se puede reemplazar un servicio sin tener que preocuparse por la tecnología fundamental; la interfaz es lo que importa y está definida en un estándar universal en servicios Web y XML.

1.6.1.3 Arquitectura en Tres Capas.

Toda aplicación contiene código de presentación, de procesamiento de datos y de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código.

➤ **La capa de presentación o interfaz de usuario.**

Está formada por los formularios y los controles que se encuentran en los formularios y es la capa con la que interactúa el usuario. Es la responsable de presentar la información del sistema al usuario. Reúne todos los aspectos para interactuar y construir las interfaces de usuarios, generalmente involucran el trabajo con ventanas, reportes, gráficos, menús y elementos de multimedia. Con la única capa que puede relacionarse es con la capa de negocio a la cuál le envía los datos que son recogidos en los formularios.

➤ **La capa de negocio.**

Se le conoce como capa de lógica de negocio, donde se establece la comunicación con la capa de datos y la capa de presentación. Recibe y responde cada petición que los usuarios realizan. Es la parte del sistema donde se establecen las reglas de negocio que deben cumplirse. Al recibir una petición del cliente, sus programas son los encargados de comunicarse con la capa de datos para almacenar, actualizar o recuperar información, emitiendo una respuesta.

➤ **La capa de acceso a datos.**

Contiene clases que interactúan con la base de datos, y permiten, utilizando los procedimientos almacenados generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio. Ella recibe las peticiones de la capa de Negocio.

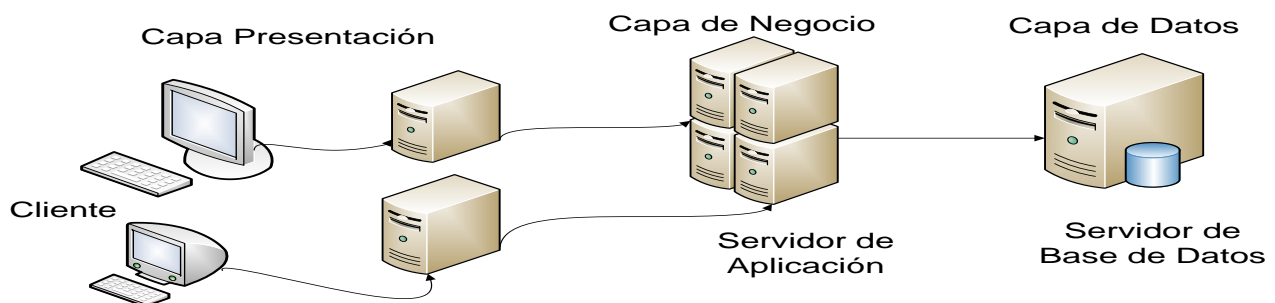


Figura 2. Arquitectura en tres capas.

1.6.1.4 Arquitectura Cliente/Servidor.

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

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

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo. Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.

- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

1.6.2 Servidor Web Apache.

Es un servidor web potente y flexible que funciona en la más amplia variedad de plataformas y entornos de desarrollo. Se ha adaptado a una gran variedad de entornos mediante su diseño modular, el cual permite a los administradores de sitios web elegir qué características van a incluir en el servidor seleccionando los módulos que se deben cargar, al compilar o ejecutar este servidor. Un servidor es una computadora que entrega a otras computadoras una información que se requiera bajo un lenguaje común, denominado protocolo. En el Servidor Web Apache es donde se almacena la información estática accedida o las aplicaciones que la generan. Puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos y puede personalizarse en dependencia a las necesidades de cada sitio web.

1.6.3 Plataforma de Servicios PlaSer.

Es 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 se encuentra 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 de los desarrolladores, asegura muchas 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.

1.6.4 Lenguajes de Programación Web.

Los lenguajes de programación Web se clasifican en dos grupos 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 JavaScript y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores.

1.5.4.1 ASP.NET.

Es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP). Esta tecnología genera páginas dinámicas en el servidor y las envía al cliente (navegador Web) que las ha solicitado, ejecutando previamente el código que contienen (código Visual Basic, C#, etc.) y convirtiendo el resultado a código HTML, que es el único que puede interpretar adecuadamente el cliente.

La clave de este proceso es que, a diferencia de la navegación clásica en que el servidor espera (*escucha*) peticiones del cliente y, cuando las recibe, envía automáticamente la respuesta (generalmente un documento HTML que es mostrado en el navegador Web), aquí el servidor ejecuta algún tipo de procesamiento a raíz de la petición del cliente y elabora dinámicamente la respuesta que devuelve. En esquema sería: el cliente (explorador Web) se conecta al servidor Web gracias a su URL y le envía una petición de (solicitud de una página Web). El cliente puede ejecutar código con secuencias de comandos que se utilizan para generar efectos dinámicos de cara al usuario (DHTML), código que no se procesa en el servidor. El servidor busca la página solicitada y ejecuta el código contenido en ella la cual se escribe en secuencias de comandos pero también en lenguajes compilados (Visual Basic, C#, etc., para tareas complejas como son cálculos matemáticos, recuperación de datos, gestión de sitios, comercio electrónico, etc.). El servidor traduce el resultado de la ejecución del código a HTML y lo envía al cliente, muestra al usuario el documento recibido. Si el usuario consulta el código fuente del documento que visita, no verá el código original del archivo que reside en el servidor, sino solamente código HTML en que la tecnología ASP.NET ha convertido el resultado del procesamiento. (7)

1.5.4.2 PERL.

Lenguaje de programación de scripts multiplataforma desarrollado por Larry Wall en 1987. Toma características del lenguaje C, del shell, de AWK, Lisp, etc. Se utiliza para manipular textos y ciertos procesos, y especialmente para la creación de CGI. Perl es un lenguaje imperativo, con variables, expresiones, asignaciones, bloques de código delimitados por llaves, estructuras de control y subrutinas. El diseño de Perl puede ser entendido como una respuesta a tres amplias tendencias de la industria informática: rebaja de los costes en el hardware, aumento de los costes laborales y las mejoras en la tecnología de compiladores. Anteriormente, muchos lenguajes de ordenador como el Fortran y C, fueron diseñados para hacer un uso eficiente de un hardware caro. En contraste, Perl es diseñado para hacer un uso eficiente de los costosos programadores de ordenador. (8)

Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows.

1.5.4.3 JavaScript.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con JavaScript se puede crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. (9)

Entre las acciones típicas que se pueden realizar en JavaScript se tienen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambio de color o cualquier otro dinamismo. Por el otro, este lenguaje permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se puede crear páginas interactivas con programas como calculadoras, agendas o tablas de cálculo. (10)

Brinda la posibilidad de programar pequeños scripts, pero también programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. (11)

1.5.4.4 Cascade Style Sheets (CSS).

Las hojas de estilo en cascada describen como se va a mostrar un documento, permitiendo darle estilos a los documentos HTML y XML. Mediante CSS es posible separar el contenido de la presentación, lo que posibilita que se pueda simplificar y limpiar el HTML de los documentos, haciendo estos más accesibles.

1.5.4.5 Hyper Text Markup Language (HTML).

HTML es el lenguaje que permite definir una página Web mediante la estructuración de textos y presentando los mismo en forma de hipertexto. Este lenguaje constituye en sí un conjunto de etiquetas utilizadas para estructurar dicho documento en forma de texto, imágenes, vínculos y otros elementos que componen la página. Unas de sus principales ventajas es su facilidad de uso porque puede ser editado desde cualquier editor, ya sea de textos o WYSIWYG.

1.5.4.6 Extensible Markup Language XML (XML).

XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Desarrollado por el World Wide Web Consortium (W3C), tiene como objetivo principal desarrollar una Web más semántica. XML no solo se usa en Internet, también se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas, pudiéndose usar en bases de datos, editores de texto, hojas de cálculo y mucho más. Está complementado por otras tecnologías que los hacen tener mayores posibilidades. También permite la compatibilidad entre sistemas para el intercambio de información de una forma fiable y segura.

1.5.4.7 Hypertext Preprocessor (PHP).

Es un lenguaje de programación interpretado que se ejecuta en el servidor y que permite dar las características dinámicas a la Web. Se puede embeber muy fácil dentro del HTML y crear aplicaciones Web muy robustas con un código fácil y legible. Es totalmente gratis y su uso puede ser en varias plataformas.

Mediante PHP también es posible crear aplicaciones con Interfaz Gráfica de Usuario (GUI), utilizando las extensiones PHP-GTK. Permite un potente trabajo con la generación de archivos como PDF, Flash, así como el trabajo con diferentes formatos de imágenes. Soporta diferentes tipos de bases de datos como son: MySQL, PostgreSQL, Oracle, SQL Server, entre otras. PHP ha transcurrido por varias generaciones, desde la 3.x hasta la 5.x, que es en esta última donde se incluyen las mayores ventajas y mejoras del mismo:

- ✓ Soporte sólido para Programación Orientada a Objetos.
- ✓ Mejoras de rendimiento.
- ✓ Mejor soporte para MySQL con extensión de rewrite completa.
- ✓ Mejor soporte a XML (XPath, DOM, entre otros).
- ✓ Soporte nativo para SQLite.
- ✓ Soporte integrado para SOAP.
- ✓ Iteradores de datos.
- ✓ Excepciones de errores.

PHP cuenta con una potente librería de acceso a datos como es ADOdb que permite implementar una capa de abstracción de acceso a datos facilitando la conexión a servidores de bases de datos soportados por el lenguaje. También posee las librerías para el trabajo con plantillas, denominadas Smarty; las mismas dan la posibilidad de separar el negocio de la presentación o sea el código PHP del HTML, haciendo más legible la sintaxis del sitio.

1.5.4.8 XSL (Extended Style Language).

Es un lenguaje que permite definir un formato para un documento XML. Básicamente es el lenguaje que define una transformación entre un documento XML de entrada y otro XML de Salida. XSL está formada por:

- ✓ XSLT: Significa transformaciones XSL y es la parte más importante de XSL. Puede transformar un documento XML en otro documento XML, en una salida XHTML o en texto sencillo.
- ✓ XPath: es una sintaxis (no basada en XML) para acceder o referirse a porciones de un documento XML.
- ✓ XSL-FO: (lenguaje de hojas extensibles de formateo de objetos), que permite especificar el formato visual con el cual se quiere presentar un documento XML, es usado principalmente para generar documentos PDF.

1.6.5 Sistemas de Gestión de Bases de Datos.

Los Sistemas de Gestión de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos y el usuario. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

1.6.5.1 MySQL.

Sistema multiplataforma que ha atraído a muchos programadores por su sencillez. Distribuido bajo la licencia GNU GPL es uno de los más usados en la actualidad. Posee un amplio subconjunto del lenguaje SQL, se usa en aplicaciones tanto pequeñas como enormes. Se conecta de forma segura proporcionando sistemas de almacenamientos transaccionales y no transaccionales. Usa tablas B-tree, muy rápidas con compresión de índice y tablas hash en memoria, que son usadas como tablas temporales; además es soportado por varios lenguajes de programación como: PHP, Java, Perl, Python, entre otros.

1.6.6 Metodologías y lenguajes de Desarrollo de Software.

1.6.6.1 RUP.

El RUP (Rational Unified Process) es el proceso de desarrollo que garantiza el análisis, la documentación y elaboración de un producto de software orientado a objetos. Es un marco de trabajo genérico, adaptable a cualquier variedad de sistemas, contexto y necesidades de cada proyecto de software. Utiliza el UML como lenguaje de presentación visual. Cuenta con un ciclo de vida que organiza las tareas en fases (inicio, elaboración, construcción, transición) que finalizan con un hito. El ciclo de vida de RUP se caracteriza por ser: dirigido por casos de usos, centrado en la arquitectura, iterativo e incremental. Contiene muchas de las mejores prácticas en el desarrollo de software.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo los Flujos de Trabajo de Desarrollo y Flujos de Trabajo de Soporte:

Flujos de trabajo de Desarrollo: Modelo de Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue.

Flujos de Trabajo de Soporte: Gestión de la Configuración y Cambios, Entorno de Desarrollo, Gestión de Proyecto. (12)

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. (13)

1.6.6.2 UML.

El Lenguaje Unificado de Modelado (UML: Unified Modeling Language), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. (14)

El modelo gráfico de UML tiene un vocabulario que identifica los elementos, las relaciones, y los diagramas. No define un proceso de desarrollo específico, solo se trata de una notación. A través de UML, se unifican las mejores técnicas para modelar y documentar los elementos de un sistema de software. Ofrece diferentes diagramas que son utilizados para visualizar los distintos aspectos del sistema. Estos diagramas se agrupan en: diagramas de estructura estática (diagramas de clases,

diagramas de casos de usos, y diagramas de objetos), diagramas dinámicos (diagramas de secuencia y colaboración, diagrama de estado, diagrama de actividad), y diagramas de implementación (diagramas de componentes, y diagrama de despliegue).

1.6.7 Definición de Tecnología y Herramientas a utilizar.

Con los argumentos descritos en epígrafes anteriores, dado el conocimiento del problema en cuestión y de las políticas de informatización establecidas por el MINSAP, se puede definir la tecnología y las herramientas a utilizar para el desarrollo del sistema que se propone. En primer lugar se utilizarán las arquitecturas mencionadas, basada en componente, orientada a servicios, en tres capas y con un modelo cliente servidor. Como librería a utilizar estará la plataforma de servicios PlaSer y lenguajes se emplearán: XSL, PHP 4, XML, HTML, JavaScript, CSS y UML. Como metodología de desarrollo se utilizará RUP y como Gestor de Base de Datos se empleará el MySQL 4. Se utilizará el Apache 1.0 como servidor web y los servicios web como premisa en la implementación. Las herramientas a utilizar son las que se comentan a continuación.

1.6.8 Herramientas a utilizar.

1.6.8.1 Stylus Studio 5.1.

Stylus Studio es un completo entorno de desarrollo integrado que incluye un potente editor de XML, un "debugger" XSLT y otras muchas herramientas pensadas especialmente para facilitar y mejorar la productividad en el desarrollo de sitios web y aplicaciones. Permite la edición de XML en modo visual y sincronizado, y un completo set de herramientas para desarrollo en XSLT entre las que se incluyen un debugger, un mapeador y una utilidad de diseño de hojas de estilo de HTML a XSLT.

1.6.8.2 Zend Studio 5.0.

Es un editor de texto para páginas PHP que brinda un buen número de ayudas que permiten la creación, gestión de proyectos y la depuración del código. Esta herramienta de desarrollo de PHP consta de dos partes, en las que se dividen las funcionalidades del lado del cliente y las del lado del servidor. Existen versiones tanto para Windows como para Linux.

1.6.8.3 Macromedia Dreamweaver 8.

Es un editor gráfico de HTML para la creación de sitios web y uno de los programas líderes para la programación web tanto para profesionales, como para aficionados. Permite realizar sitios web complejos y bien estructurados. Se pueden diseñar las páginas con una interfaz completamente gráfica mientras se observa simultáneamente el código generado. Esta versión incorpora numerosas herramientas para el tratamiento de tablas y otros componentes avanzados así como para la inserción de objetos no HTML como películas flash o JavaScript. Si se programa en PHP y se colorea el código, permitirá testear las páginas con un servidor configurado. Se ha ampliado su funcionalidad en ámbitos esenciales como XML y CSS, y su flujo de trabajo se ha agilizado para que los usuarios puedan hacer más cosas en menos tiempo. Permite convertir un documento XML a uno XHTML y crear y validar documentos XSL.

1.6.8.5 SQL Manager 2007 for MySQL.

EMS MySQL Manager brinda una interfaz gráfica sencilla para poder administrar las bases de datos creadas en MySQL. Con esta utilidad se puede administrar de una manera sencilla los perfiles de usuario, correr scripts de SQL, peticiones y escritura en tu base de datos. Su interfaz gráfica es altamente comfortable.

1.6.8.6 Rational Rose Enterprise Edition.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML. Rational Rose Enterprise Edition es una herramienta que se podría encuadrar dentro del grupo de herramientas más técnicas debido a que se encarga de llevar a cabo tanto la automatización de los sistemas para la posterior generación de código, como para labores de ingeniería inversa. Tiene como ventaja que permite generar código en distintos lenguajes de programación a partir de un diseño UML.

1.6.9 Servicios Web.

Existen múltiples definiciones sobre lo que son los servicios Web (Web Services), por lo que a la hora de dar una adecuada definición resulta un poco compleja. Una posible sería, que estos utilizan un conjunto de estándares y protocolos entre diferentes aplicaciones para intercambiar datos por la red, aunque estén desarrolladas en diferentes plataformas y cualquier lenguaje de programación, presentando información dinámica al usuario, y propiciando interoperabilidad y extensibilidad entre aplicaciones. Por un lado estaría

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

SOAP (Protocolo Simple de Acceso a Objetos) protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja.

Los datos pueden ser transmitidos a través de HTTP, SMTP, etc. Además de que especifica el formato de los mensajes, los que están compuesto por un envelope, cuya estructura está formada por los siguientes elementos: header y body. Por otro lado están los WSDL (Lenguaje de Descripción de Servicios Web), este es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios web; y especifica la sintaxis y los mecanismos de intercambio de mensajes.

En este capítulo se profundizó en el conocimiento de conceptos y definiciones necesarios para una mejor comprensión del trabajo. Se realizó un análisis de las tecnologías involucradas en el desarrollo del sistema propuesto, fundamentándose también la elección del lenguaje de programación, el sistema gestor de bases de datos y la metodología a utilizar. Además, se describen las herramientas a utilizar; teniendo en cuenta las políticas definidas por el MINSAP para el desarrollo de sus sistemas en el país.

CAPÍTULO 2. DISEÑO DEL SISTEMA

En el presente capítulo se describe el Modelo de Diseño, refinando lo modelado en el Análisis. Se modelan los diagramas de clases del diseño por cada caso de uso y se describen todas las clases involucradas. Además, de justificar los distintos patrones que se utilizan en la arquitectura y diseño del sistema.

2.1 Workflow (Flujo de Trabajo).

La definición de un “Workflow es la automatización de un proceso de negocio, total o parcialmente, durante la cual documentos, información o tareas, son pasados de un participante a otro por una acción conforme a un conjunto de reglas procedimentales”. (15)

Un workflow comprende un número de pasos lógicos, conocidos como actividad. Una actividad puede involucrar interacción manual con el usuario o ser ejecutada por una máquina. Un ejemplo de uso de workflow es en el Proceso Unificado de Desarrollo donde se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales: Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Instalación, Administración del proyecto, Administración de configuración y cambios y Ambiente. En esta investigación el workflow automatiza el proceso de negocio de la gestión de los Certificados Médicos de Defunción, así como de las Solicitudes de Reparación y los Reparos.

2.1.1 Workflow del proceso de gestión de los CMD.

Un Certificado Médico de Defunción una vez creado y registrado por el médico que recibe el fallecido, transita por todos los niveles de dirección del SNS y en cada uno de ellos es revisado y aprobado en caso de no tener ningún error en su confección. Uno de los objetivos primordiales de esta investigación es proporcionar esta funcionalidad al sistema que se elaboró. Para lograrlo se diseñó el workflow que a continuación se explica.

Una vez que se inserta el CMD en el sistema, se determina la subordinación de la unidad de salud donde

este ha sido creado, que puede ser de subordinación municipal, provincial o nacional. Al incorporarse al flujo se inserta además en un historial, acción que permitirá llevar constantemente una traza del certificado. Las acciones en las que pueden estar involucrados, estando en el flujo son: Listar, Buscar y Revisar. Si al revisarlo no se detecta ningún error, se simula el transporte del documento hacia la dirección de salud del nivel superior. Actualizando el nuevo nivel donde se encuentra y el historial; esto se realiza solo cuando el CMD no se encuentra en el nivel nacional, porque de encontrarse en este nivel es archivado y se elimina del flujo.

En caso de detectarse un error, se crea una Solicitud de Reparación, que es enviada junto al certificado, a la unidad de salud donde fue expedido. Cuando esto sucede, cambia el nivel en que se encuentra el certificado y su estado, pasando de “Normal” a “En reparación”, así como su actualización en el historial. Al realizarse el reparo vuelve a cambiar el estado a “Reparado”, actualizándose nuevamente en el historial y se somete a las acciones explicadas anteriormente.

2.1.2 Diagrama de Estado.

Un diagrama de estado representa la secuencia por la que pasa un caso de uso, un objeto a lo largo de su vida, o bien el sistema completo. En él se indican que eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que generan. Se modela para todas las clases que se consideran con un comportamiento dinámico. También se modela las secuencias de estado que un objeto de la clase atraviesa durante su vida en respuesta a los estímulos recibidos, junto con sus propias respuestas y acciones.

Los estados representan las condiciones de objetos en ciertos puntos en el tiempo. Los eventos representan incidentes que hacen que los objetos pasen de un estado a otro. Las líneas de transición describen el movimiento desde un estado hasta otro. Cada línea de transición se nombra con el evento que causa esa transición. Las acciones ocurren cuando un objeto llega a un estado.

El siguiente diagrama aporta una visión de la funcionalidad del sistema a diseñar en cuanto al workflow de los certificados por los diferentes niveles del SNS, además de las solicitudes de reparación y los respectivos reparos que estos pueden tener.

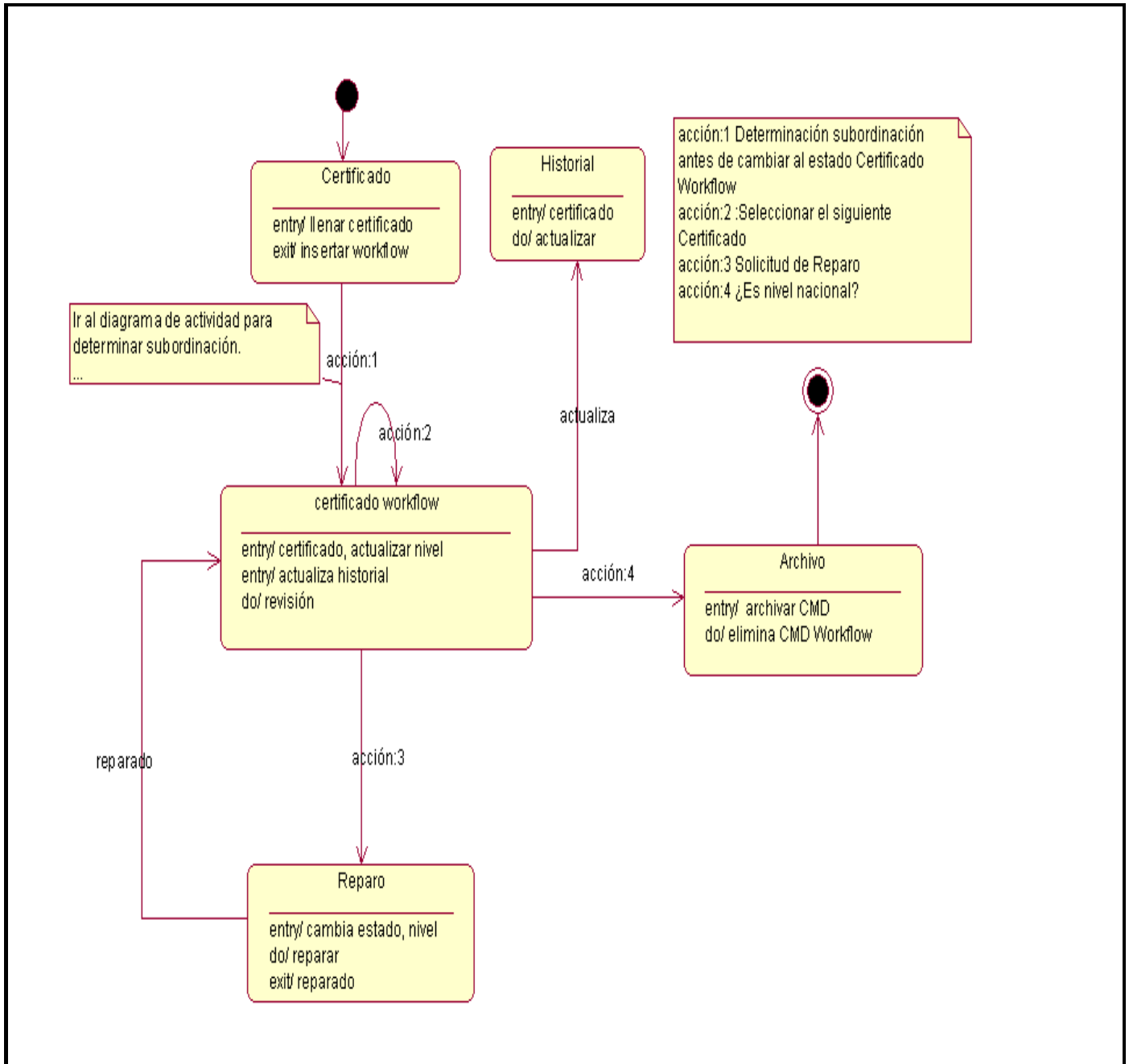


Figura 3. Diagrama de Estado, Workflow del CMD.

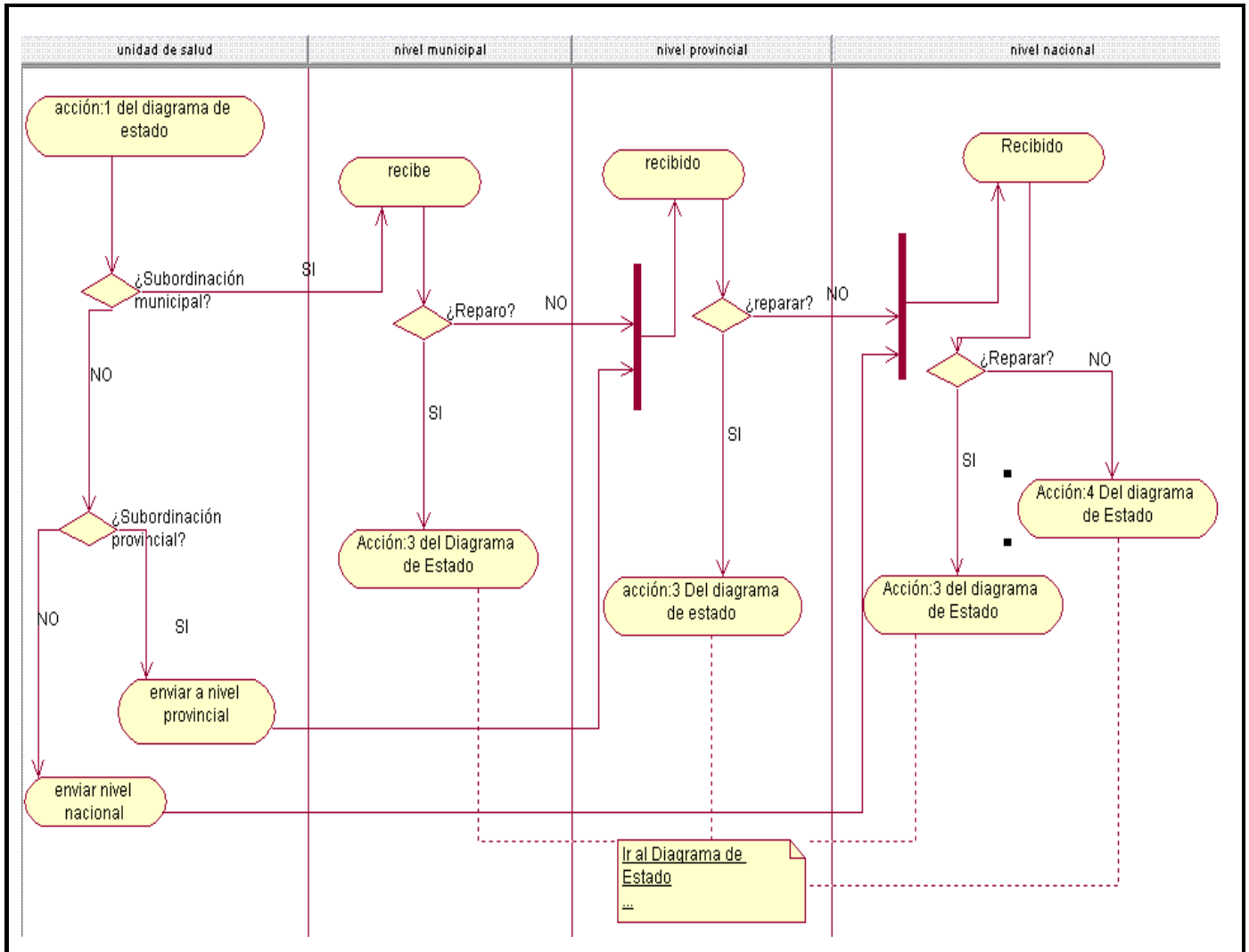


Figura 4. Diagrama de Actividad para determinar la subordinación de la unidad de salud.

Una vez llenado el Certificado Médico de Defunción en el sistema, por parte del técnico de estadísticas de la unidad de salud correspondiente, se ejecuta la acción de salida del estado **Certificado** exit/ insertar Workflow. Llevando el objeto al estado **certificado Workflow** con una transición, cuyo evento determina la subordinación de la unidad de salud en la fue registrado el certificado y así enviarlo al nivel en que debe ser revisado, ya sea nivel **Municipal**, **Provincial**, o **Nacional**. Al entrar al estado **certificado Workflow** se ejecutan las acciones de entrada actualizar nivel donde se encuentra el certificado y el **historial**, llevando de esta forma la traza de todas las acciones por las que pasa el certificado.

2.2 Modelo del Diseño.

El modelo de diseño es un modelo que contribuye a establecer una arquitectura estable y sólida, y crear un plano del modelo de implementación. Es un modelo de objetos que describe la realización de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema que desarrolla. Se representa por un sistema de diseño que denota el subsistema de nivel más alto del modelo. La utilización de otro subsistema es, entonces, una forma de organización del modelo de diseño en porciones más manejables. (16)

2.3 Justificación del uso de los Patrones.

Los patrones de diseño son una solución estándar para un problema común de diseño dentro de un contexto dado y constituyen una manera más práctica de describir ciertos aspectos de la organización de un programa. Estos permiten dar flexibilidad y extensibilidad al diseño en cuestión. Son una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. Los patrones contribuyen a reutilizar diseño, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones.

2.3.1 Alta Cohesión.

Una clase está en alta cohesión cuando el objeto tiene delimitadas sus responsabilidades, es decir, no se le asignan responsabilidades que no estén limitadas dentro de sus funciones. Las responsabilidades de un objeto se traducen después en métodos que realizan acciones.

El Registro de Fallecidos se diseñó asignando responsabilidades de modo que la cohesión sea alta, ejemplo de esto lo representan los métodos del negocio, pues estos tienen bien delimitadas sus responsabilidades, no recargando en ningún caso sus funcionalidades, permitiendo mayor eficiencia y que el tiempo de respuesta y ejecución no exceda lo estimado, generando por ende bajo acoplamiento.

2.3.2 Bajo Acoplamiento.

El bajo acoplamiento hace referencia a que un objeto tenga la mínima dependencia posible con el resto del sistema. Esto posibilita que se puedan realizar modificaciones en partes del programa sin necesidad de que los objetos relacionados se vean afectados en gran medida.

2.3.3 Fachada.

El patrón de diseño Fachada sirve para proveer de una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. La idea principal es la de ocultar todo lo posible la complejidad de un sistema, el conjunto de clases o componentes que lo forman, de forma que solo se ofrezca un (o unos pocos) punto de entrada al sistema tapado por la fachada. El proceso consiste en asignar la responsabilidad a una clase directa para que colabore con una clase indirecta, de modo que no se necesite saber nada de la clase indirecta. Una ventaja de usar una clase fachada para comunicar dos partes o componentes, es la de aislar los posibles cambios que se puedan producir en alguna de las partes. Si cambias el medio de comunicación o de almacenamiento de una de las partes, la otra, que hace la presentación, no tiene porque enterarse, y viceversa.

En la solución de este sistema se utiliza este patrón Fachada para abstraer la capa de presentación de la librería de clases PlaSer. Este patrón está implementado en una clase que lleva el mismo nombre de fachada, permitiendo que la comunicación se establezca siempre con la clase fachada y no con las clases de PlaSer directamente, muchas veces no conocidas en su totalidad por los desarrolladores. Además permite que el sistema sea portable a cualquier cambio en la librería de clases de PlaSer, por tanto reduce el impacto al cambio y permite la escalabilidad del sistema.

2.3.4 Proxy.

Los patrones estructurales se ocupan de cómo se combinan las clases y los objetos para formar estructuras más grandes. El Proxy es un patrón de tipo estructural asociado a objetos. Este patrón, proporciona un sustituto o representante de otro objeto para controlar el acceso a éste. Un objeto Proxy, recibe llamadas a métodos que pertenecen a otros objetos. El objeto Proxy, comparte interfaz o superclase común con el objeto que realmente provee el servicio. El patrón obliga a que las llamadas a

métodos de un objeto ocurran indirectamente a través de un objeto Proxy.

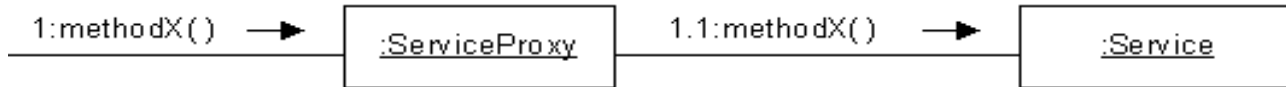


Figura 5. Llamadas a través de un objeto proxy.

2.3.5 Data Access Objects (DAO: Objetos de Acceso a Datos).

El propósito del patrón DAO es, en pocas palabras, abstraer y encapsular todos los accesos a la fuente de datos. Con esto se obtiene varias ventajas: se tiene un paliativo al problema del diferencial de impedancia (transparencia); se baja en nivel de acoplamiento entre clases, reduciendo la complejidad de realizar cambios; se aísla las conexiones a la fuente de datos en una capa fácilmente identificable y mantenible.

2.3.6 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.

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.

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.

2.4 Definición de la estructura del diseño.

Se ha definido que la aplicación posea una estructura centrada en la flexibilidad ante los cambios y la reutilización. Mediante la conservación de dicha estructura durante la implementación, se obtendrá un sistema mantenible como un todo: será flexible a los cambios en los requerimientos e incluirá elementos

que podrán ser reutilizados cuando se construyan sistemas similares. El siguiente diagrama muestra el diseño de la estructura de RF:

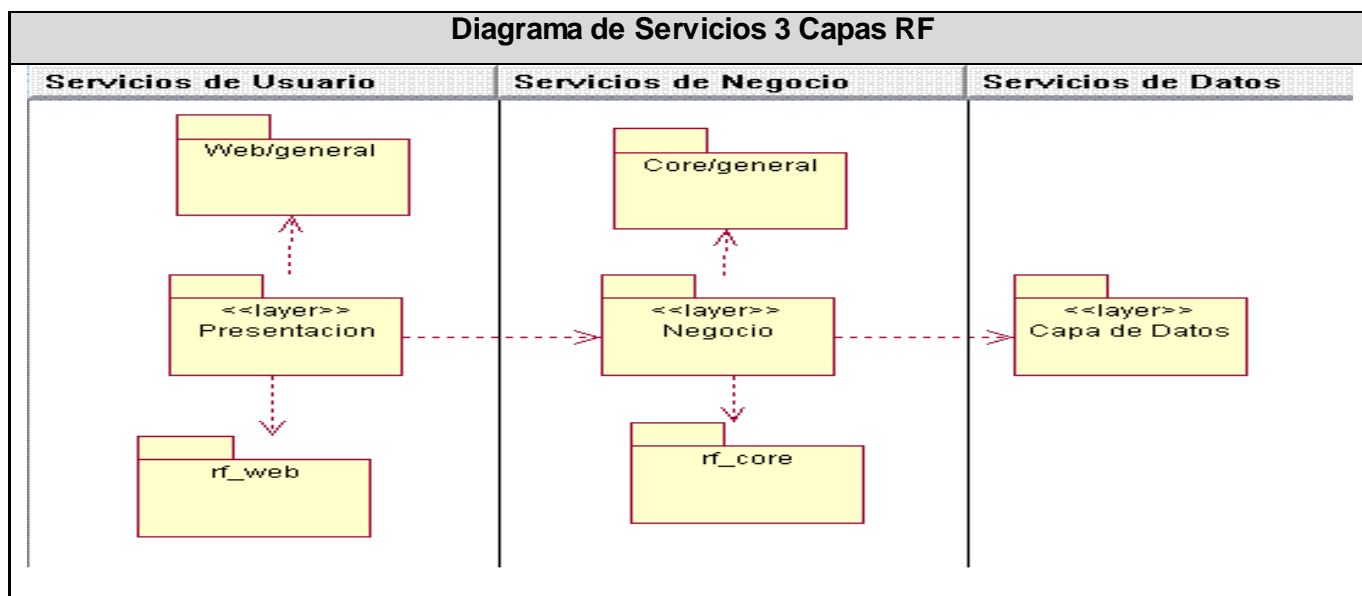


Figura 6. Diagrama de Servicios 3 Capas RF.

Descripción de cada uno de los subsistemas que intervienen en la estructura del sistema:

Capa de Presentación: Esta capa encapsula la lógica de navegación y controla los eventos de las interfaces. Es la encargada de interactuar con el usuario y hacer transparente la complejidad del sistema a los mismos, estando conformada por: DHTML, XHTML, XML, PHP y XSL.

Capa de Negocio: Esta capa encapsula la lógica de negocio. Establece comunicación entre la capa de presentación y la de datos, recibiendo y respondiendo peticiones. Los métodos que la conforman reciben solicitudes de la capa superior, se comunican con los ficheros de acceso a datos y actualizan o retornan información.

Core General: Este paquete contiene los ficheros utilizados por la capa de negocio para las validaciones y mensajes de error, así como los que brindan la posibilidad de generar ficheros en formato Portable Document Format (pdf) o Microsoft Office Excel (xls). Este paquete contiene la clase de acceso a datos

DBz y la clase encargada de brindar la dirección física de los componentes, PLASER_Server, la que incluye para su funcionamiento al fichero confplaser_server.

Capa de Datos: Contiene las tablas de la base de datos de RF, las que son gestionadas por la capa de negocio.

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.

rf_core: Incluye todos los métodos del negocio de RF.

rf_web: Incluye todo el código de presentación de RF.

2.4.2 Estereotipos Web.

En el diseño aplicaciones Web, Jim Conallen propone en 1998, una extensión al UML, a esta extensión se le conoce hoy en día como los estereotipos web, conformadas por Server page, Client Page, Form, Frame set, y Target. De todas las extensiones, presenta como elementos más significativos tres clases UML, que son los estereotipos Server Page, Client Page, y Form. En la Tabla 1 están representadas las relaciones que se establecen entre los tres elementos claves mencionados anteriormente.

Al ser RF una aplicación web, la misma se modelará con estereotipos definidos para este tipo de aplicaciones, los estereotipos web, lo cual proporcionará una mayor comprensión de las funcionalidades del sistema y el poder distinguir qué atributos, operaciones y relaciones están activas en el servidor y cuáles están activas cuando el usuario está interactuando con la página en el navegador cliente.

A continuación se brinda una explicación de cómo son usados estos estereotipos en el diseño de la propuesta del sistema y qué representa cada cual:



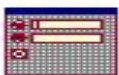
sp_<NombreClaseServidora>

<<Server Page>>: Representa la clase que tiene código que se ejecuta en el servidor, la cual se encarga de construir (build) o generar el resultado HTML y/o realizar peticiones a la capa inferior.



cl_<NombreClaseCliente>

<<Client Page>>: Es una página Web con formato XHTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del *script* que son accesibles para cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.



<nombre>

<<FormHTML>>: Es una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario (input boxes, text areas, radio buttons, check boxes, hidden fields, entre otros). No tienen operaciones, el método para el paso de los parámetros es `$_POST` y se comunican con las páginas servidores mediante *submit*.

Relaciones utilizadas entre clases:

Hasta	Client Page	Form	Server Page
Client Page	<<link>>, <<redirect>>	Contiene	<<link>>, <<redirect>>
Form		--	<<submit>>
Server Page	<<build>>, <<redirect>>, <<link>>	--	<<redirect>>, <<include>>, <<call>>, <<callBx>>, <<callB>>

Tabla 1. Relaciones entre las clases principales que conforman la extensión de UML para Web.

Las páginas clientes son construidas (build) por una sola página servidora. Estas páginas servidoras pueden completar su funcionamiento al incluir funciones o códigos existentes en otras páginas servidoras, a partir de sentencias como include ('.../general/FachadaRF.php'), esta relación se puede representar en el diagrama de clases. También una página servidora al terminar su función puede ejecutar la sentencia (redirect) hacia otra página servidora que es la encargada de construir la página cliente.

Las sentencias (call), (callB) y (callBx) son invocadas desde una página servidora a otra para realizar llamadas a cualquier componente y así utilizar sus servicios. Una página cliente puede participar en asociaciones con «Client Page»'s o «Server Page». También las páginas clientes contienen formularios, los cuales pueden ser más de uno. Los formularios pueden enviar sus datos (submit) a las distintas páginas servidoras encargadas de procesarlos y forman parte del resultado HTML que se muestra; de ahí que la relación página cliente- formulario es de composición, es decir una página cliente está compuesta por uno o más formularios. Toda operación que interactúa con el Form es una propiedad de la página que contiene el mismo.

2.5 Diagrama de Clases del Diseño.

Un diagrama de clases del diseño se utiliza para modelar la vista de diseño estática de un sistema. Estos contienen clases, interfaces, colaboraciones y relaciones de dependencia, generalizaciones y asociación. Son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa. Brindan además un mayor acercamiento a la forma y al contenido de la solución propuesta. El Registro de Fallecidos es una aplicación web, por lo que para realizar estos diagramas de clases se tuvo en cuenta la extensión UML para Web.

A continuación se muestran los diagramas de clases de diseño de los casos de uso arquitectónicamente significativos; para profundizar más en el tema, ver el documento RF_Modelo de Diseño v1.1 en el expediente de proyecto. En el caso de la gestión de los codificadores, información que se gestiona por ser imprescindible para el llenado de los certificados médicos, solo se mostrará como ejemplo uno de ellos.

Casos de Uso arquitectónicamente significativos:

- ✓ Gestionar Codificador Certificación.
- ✓ Gestionar Certificado Médico Defunción (Escenarios: Certificado Fetal, Certificado Mayor, Certificado Neonatal).
- ✓ Gestionar Reparos.
- ✓ Gestionar Solicitud de Reparos.
- ✓ Buscar Solicitudes de Reparos.

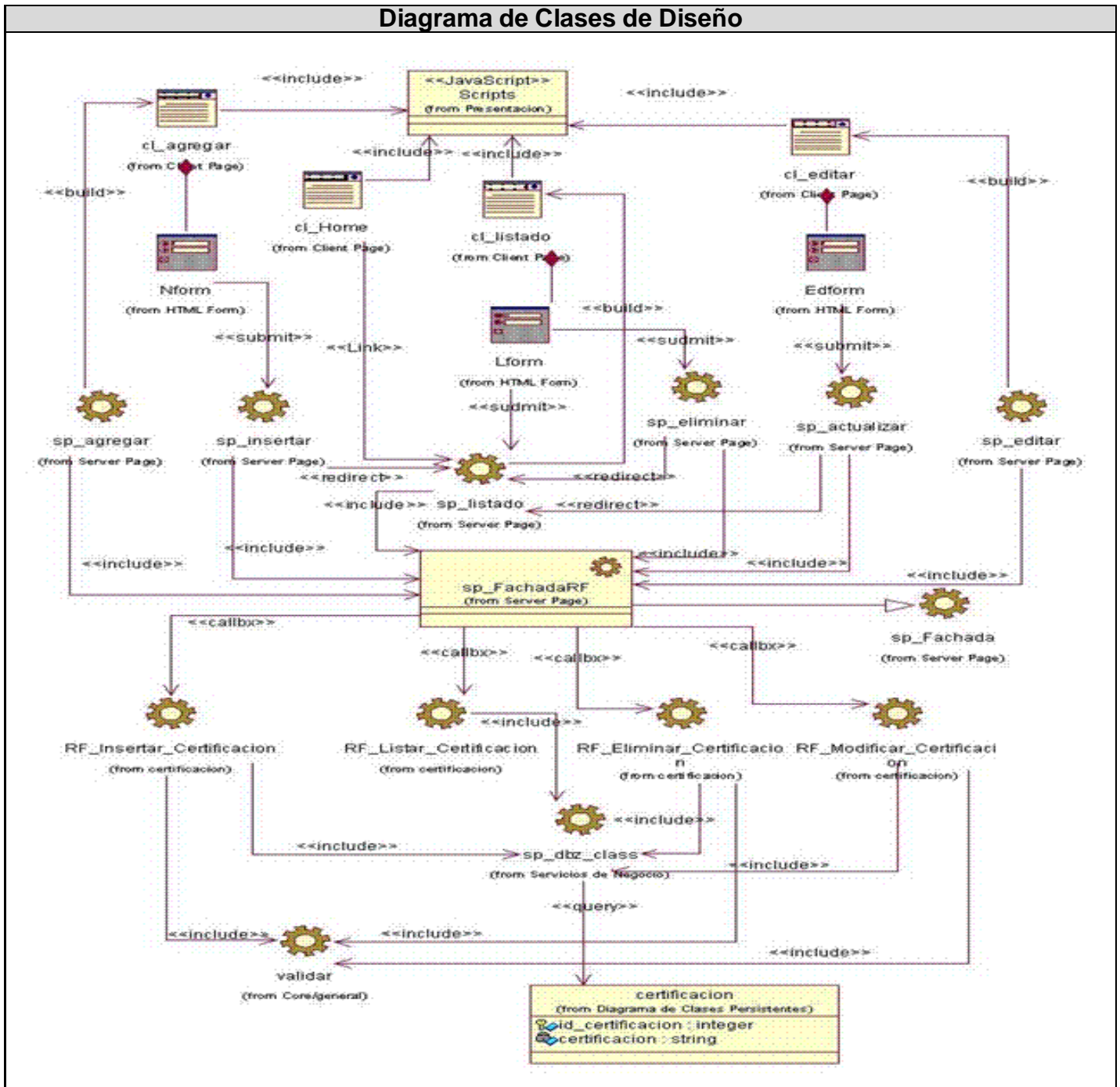


Figura 7. Diagrama de Clases del Diseño del Caso de Uso Gestionar Codificador Certificación.

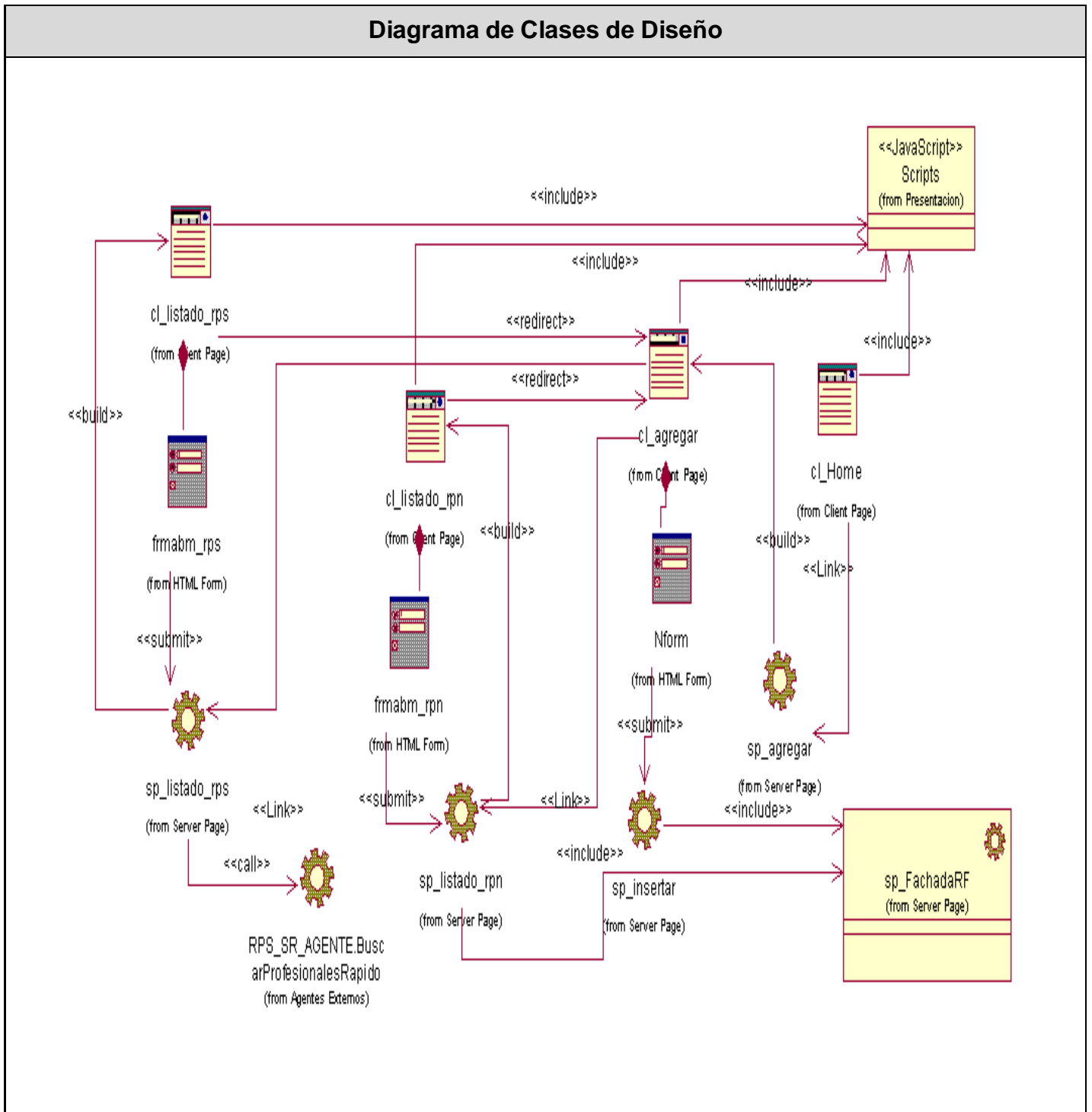


Figura 8. Diagrama de Clases del Diseño del Escenario Certificado Fetal [1/3].

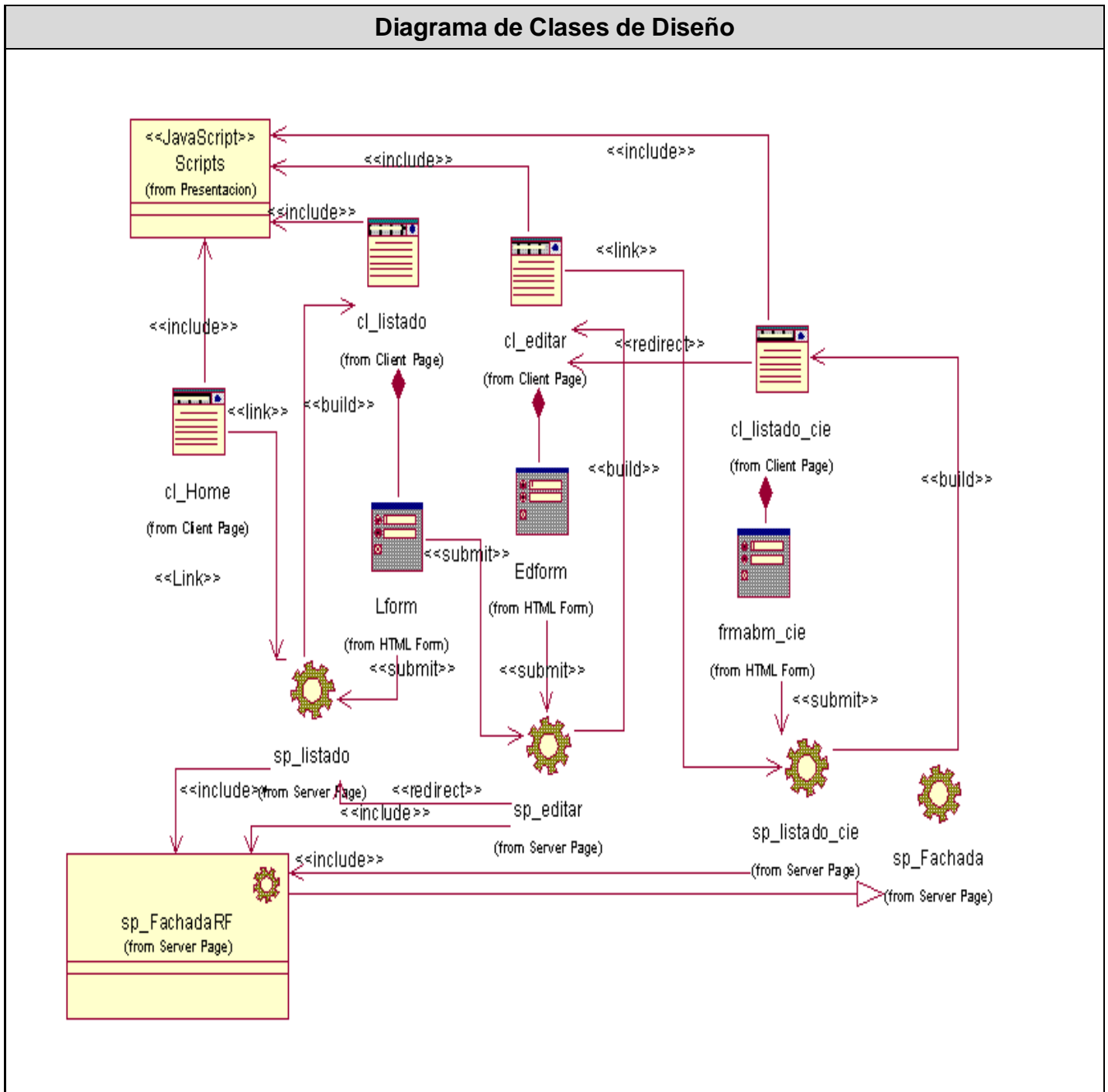


Figura 9. Diagrama de Clases del Diseño del Escenario Certificado Fetal [2/3].

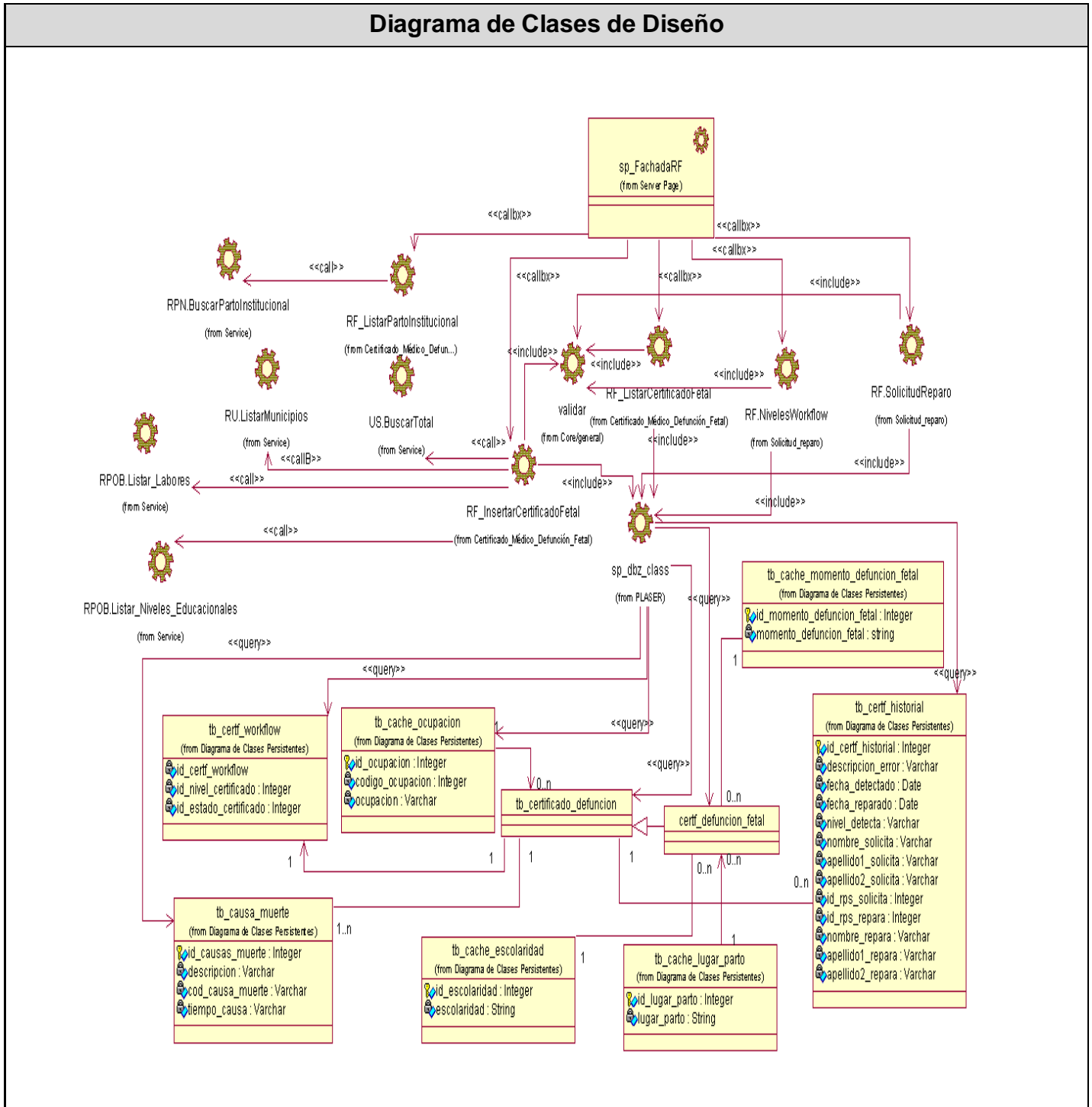


Figura 10. Diagrama de Clases del Diseño del Escenario Certificado Fetal [3/3].

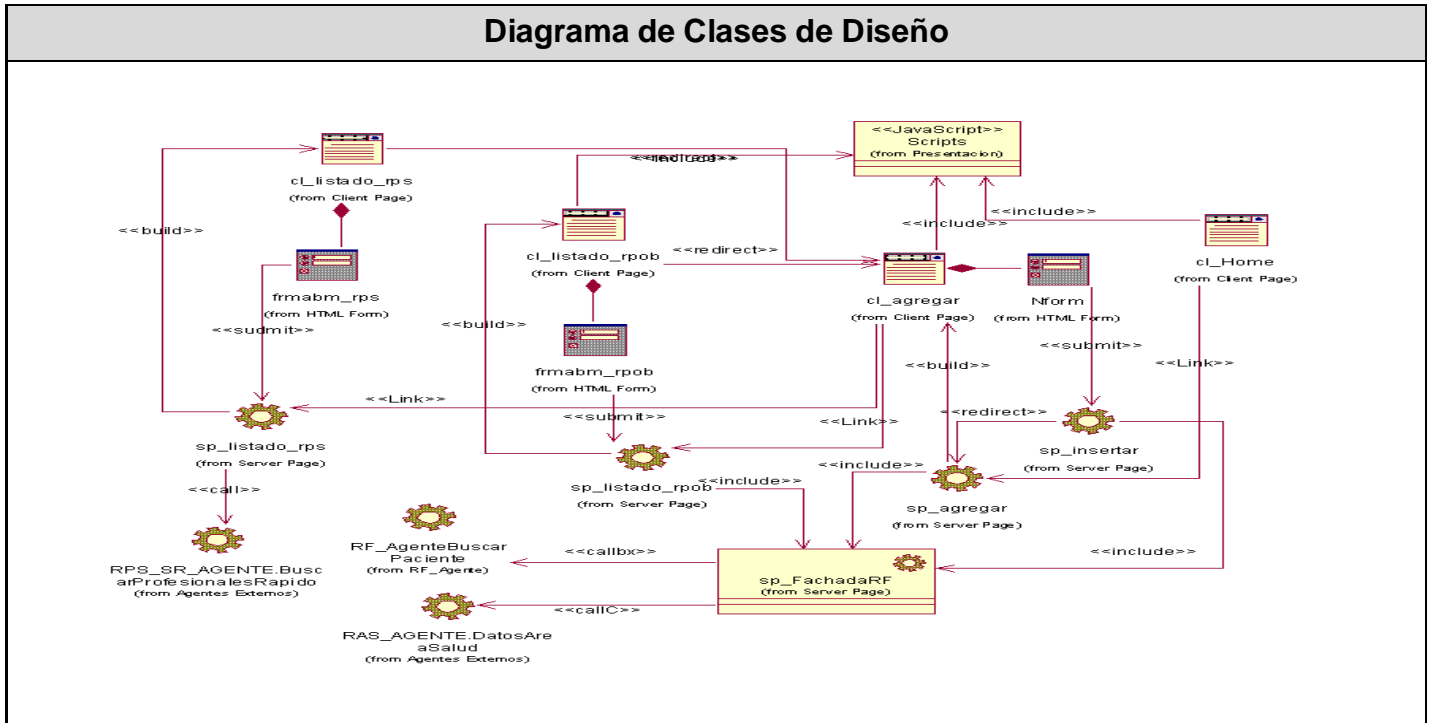


Figura 11. Diagrama de Clases del Diseño Escenario Certificado Mayor [1/3].

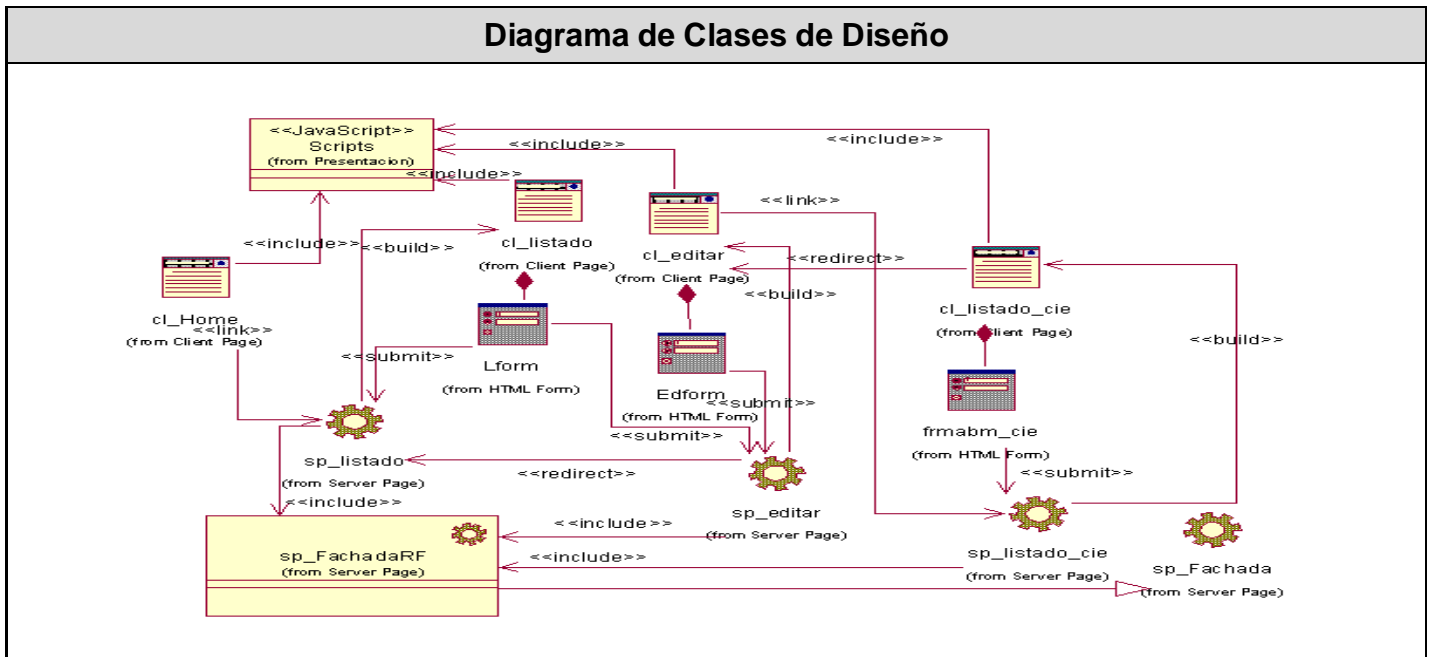


Figura 12. Diagrama de Clases del Diseño del Escenario Certificado Mayor [2/3].

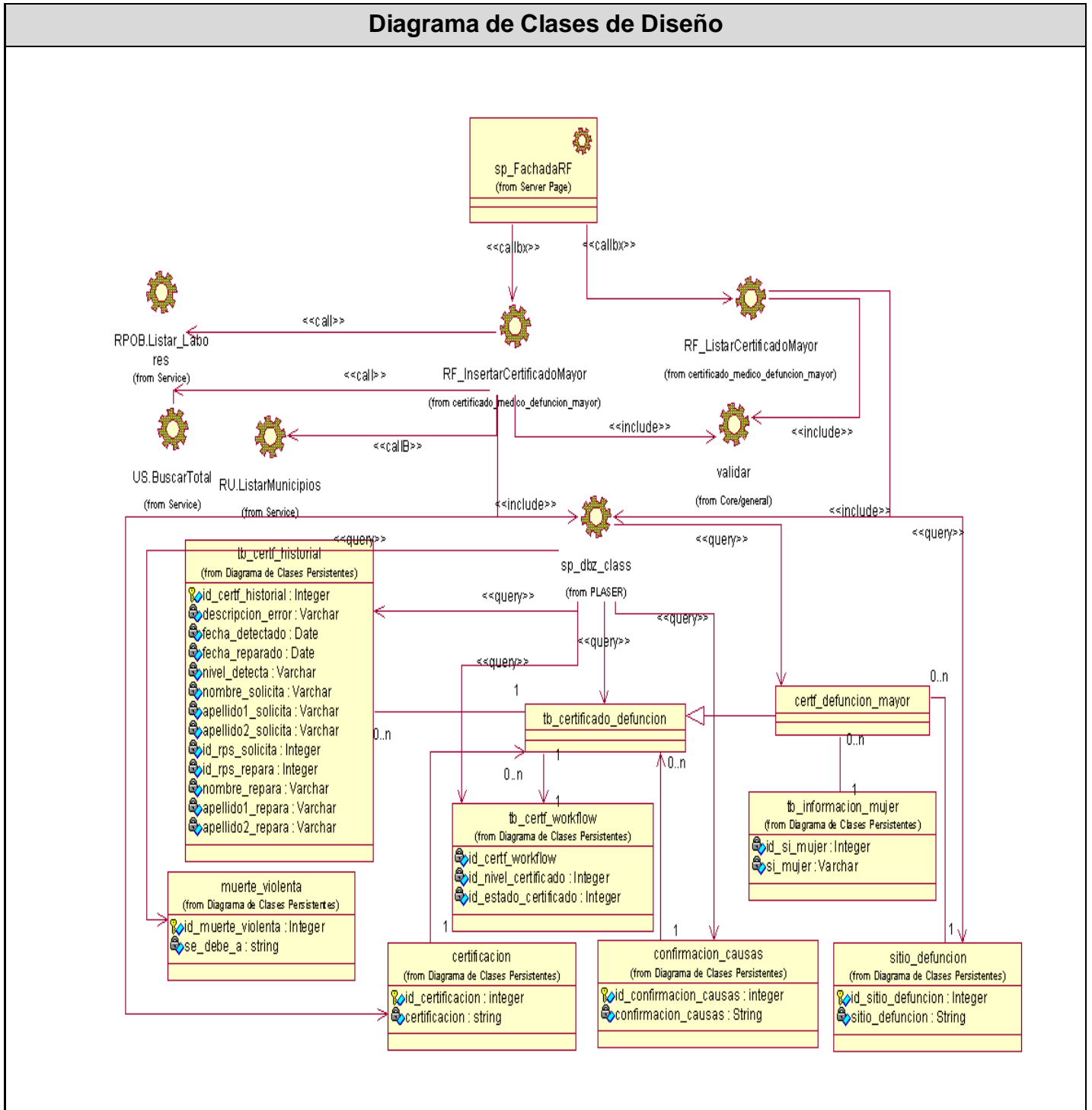


Figura 13. Diagrama de Clases del Diseño del Escenario Certificado Mayor [3/3].

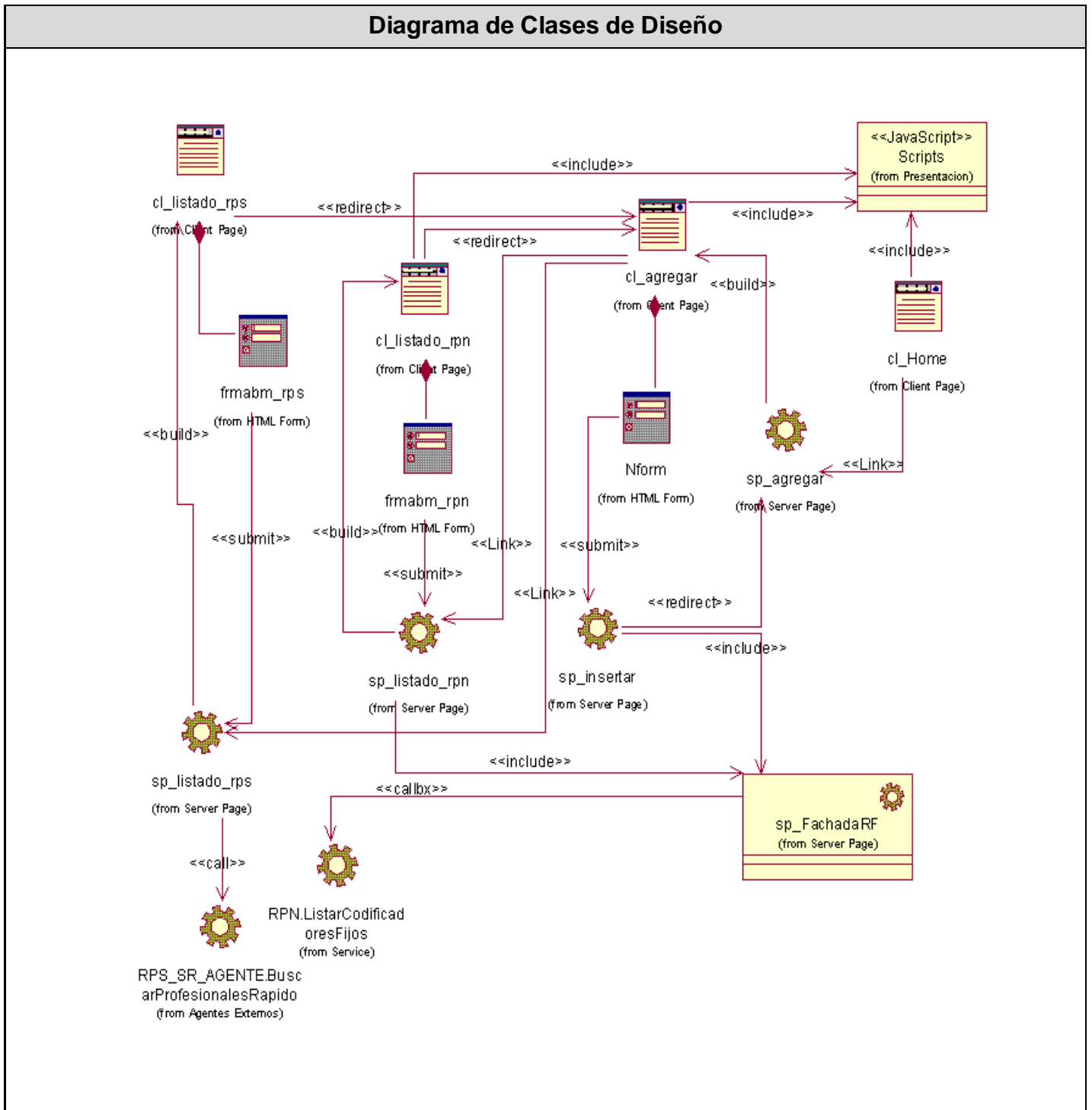


Figura 14. Diagrama de Clases del Diseño del Escenario Certificado Neonatal [1/4].

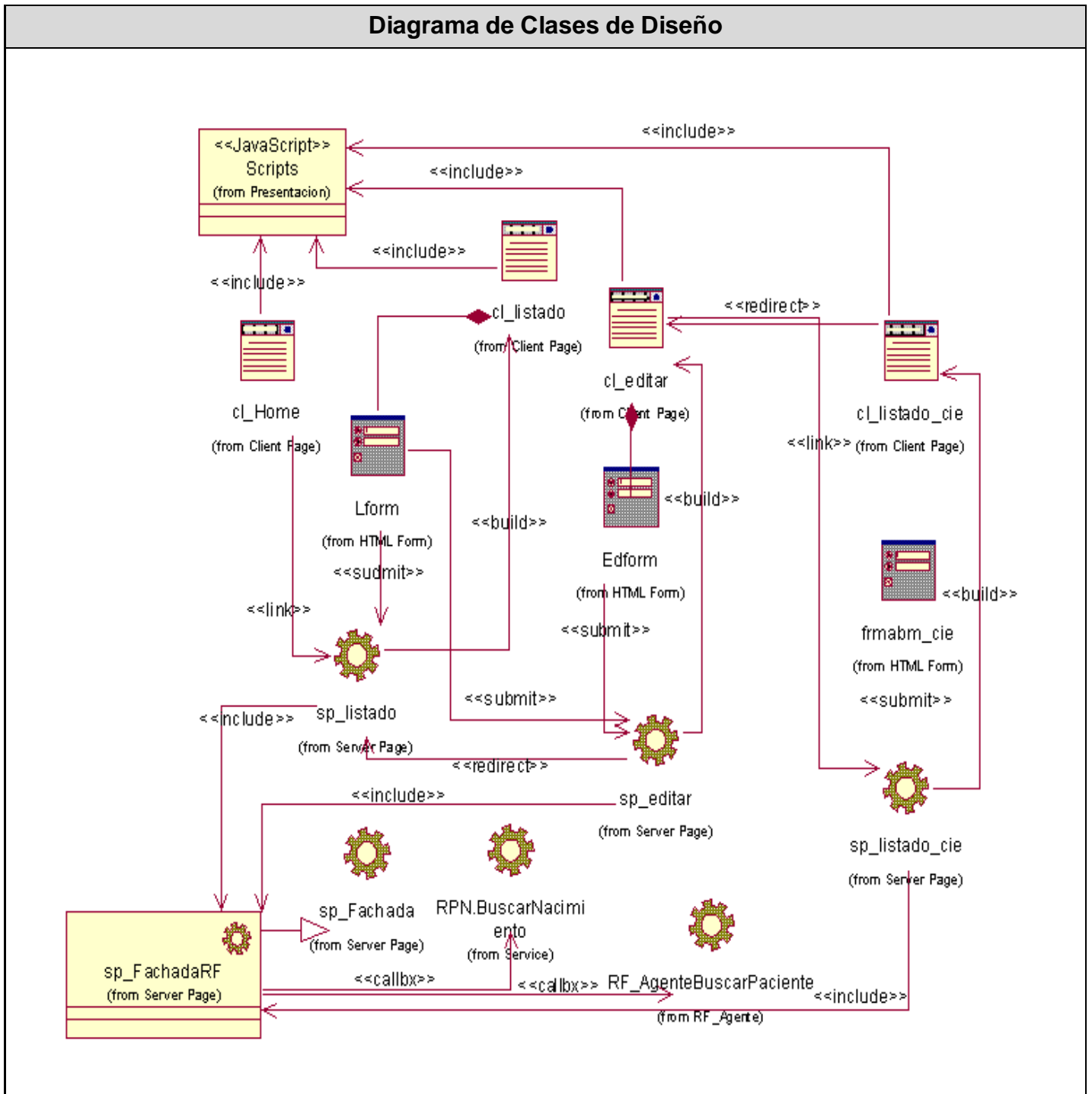


Figura 15. Diagrama de Clases del Diseño del Escenario Certificado Neonatal [2/4].

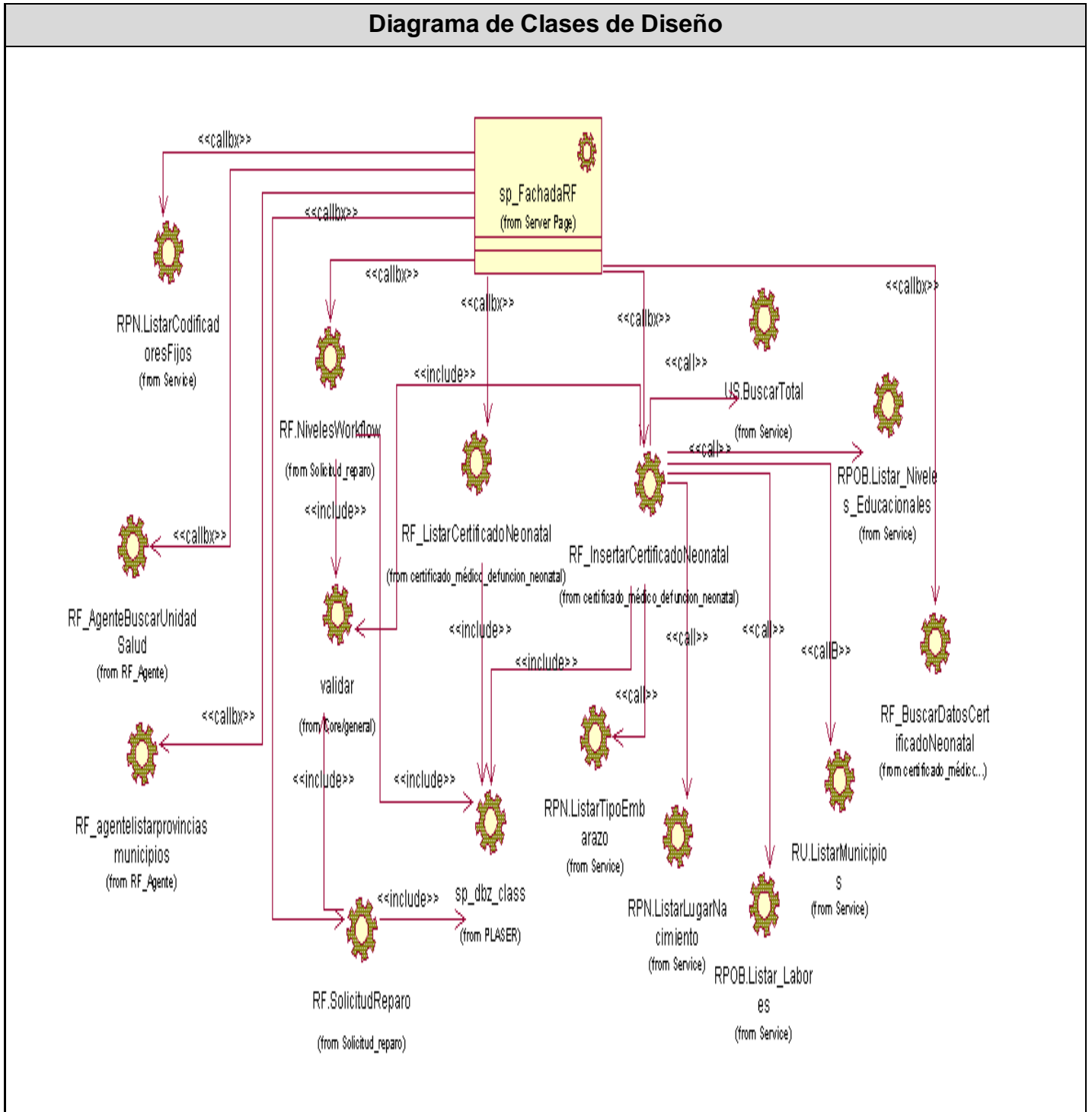


Figura 16. Diagrama de Clases del Diseño del Escenario Certificado Neonatal [3/4].

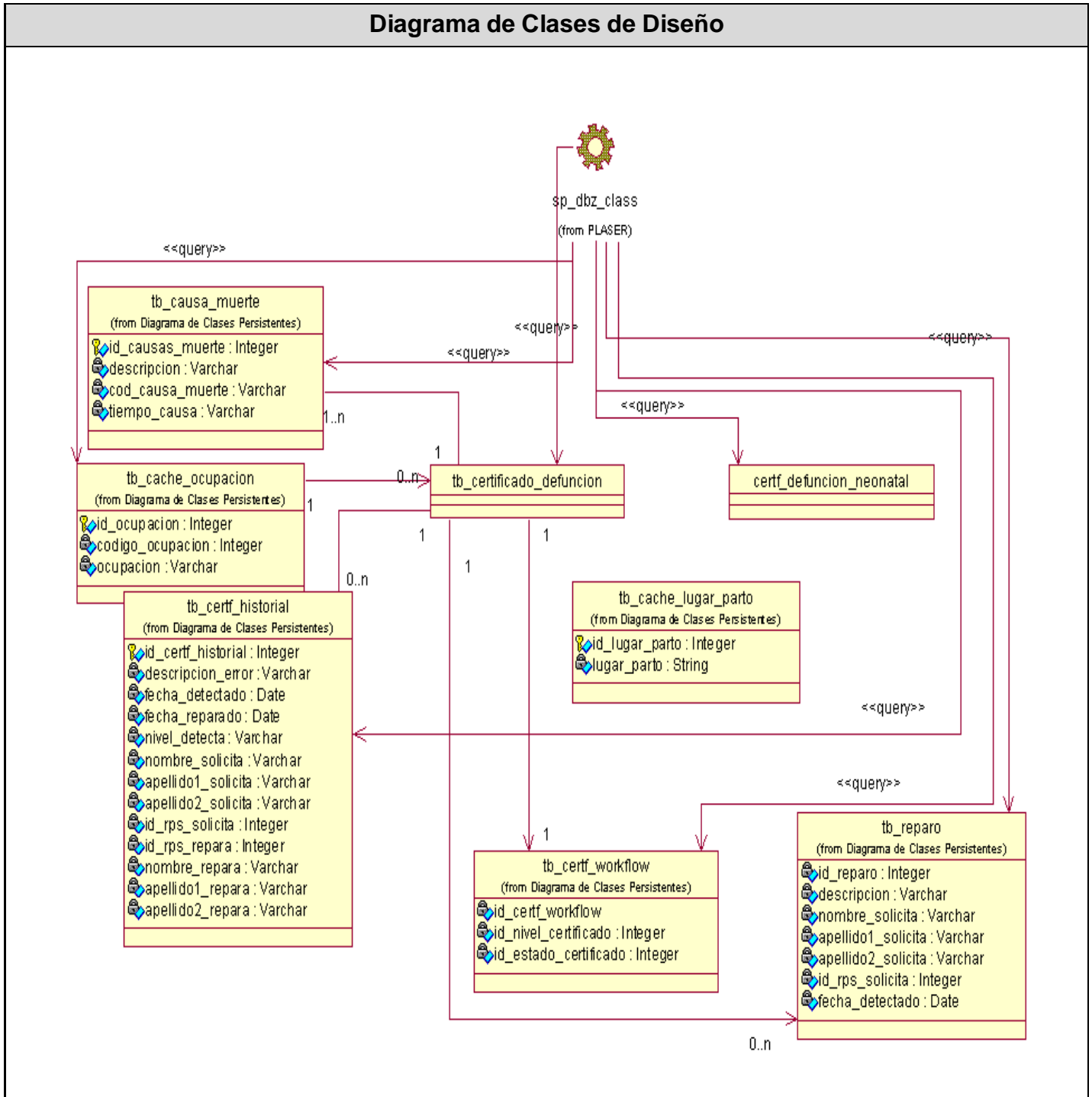


Figura 17. Diagrama de Clases del Diseño del Escenario Certificado Neonatal [4/4].

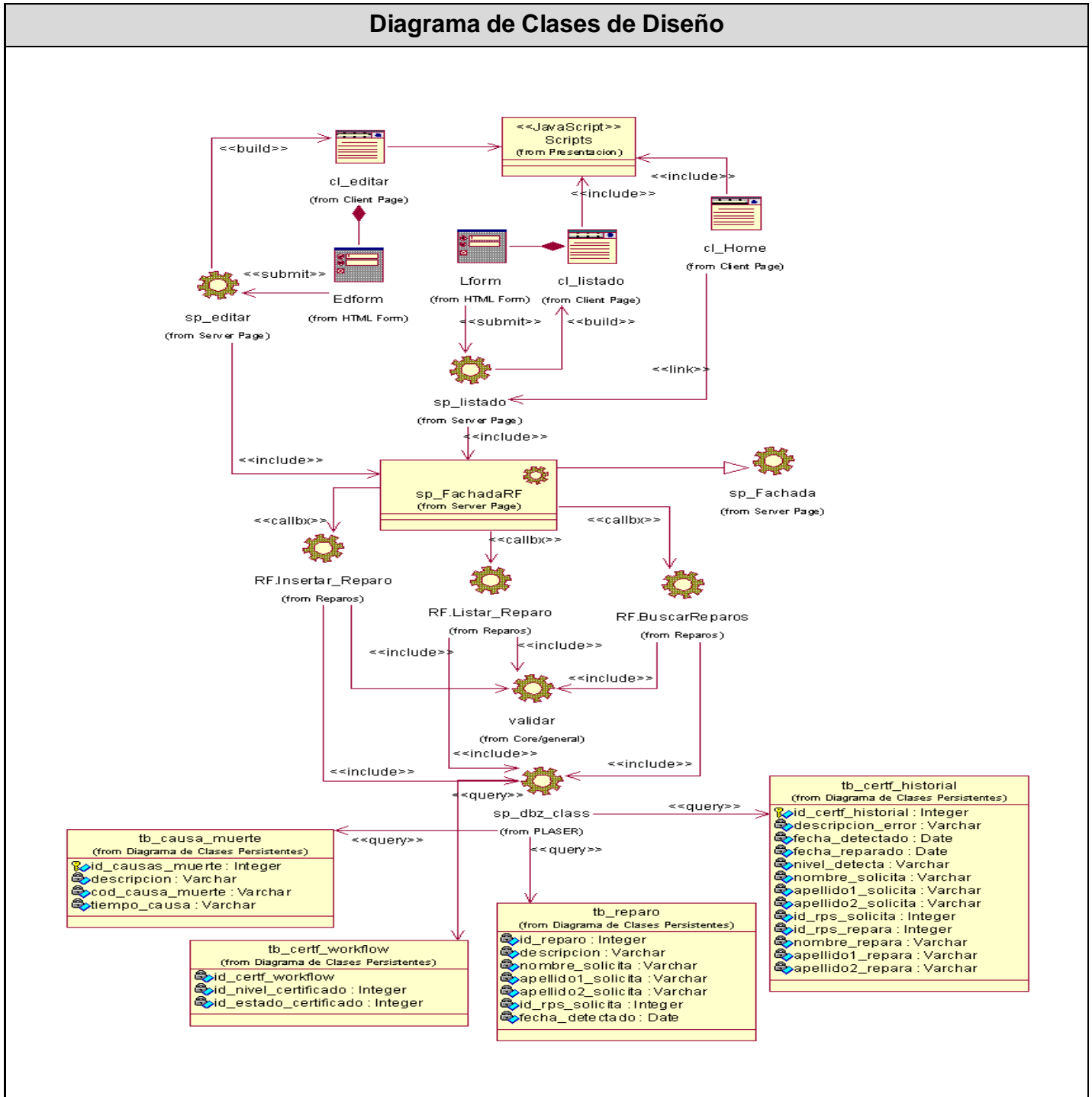


Figura 18. Diagrama de Clases del Diseño del Caso de Uso Gestionar Reparos.

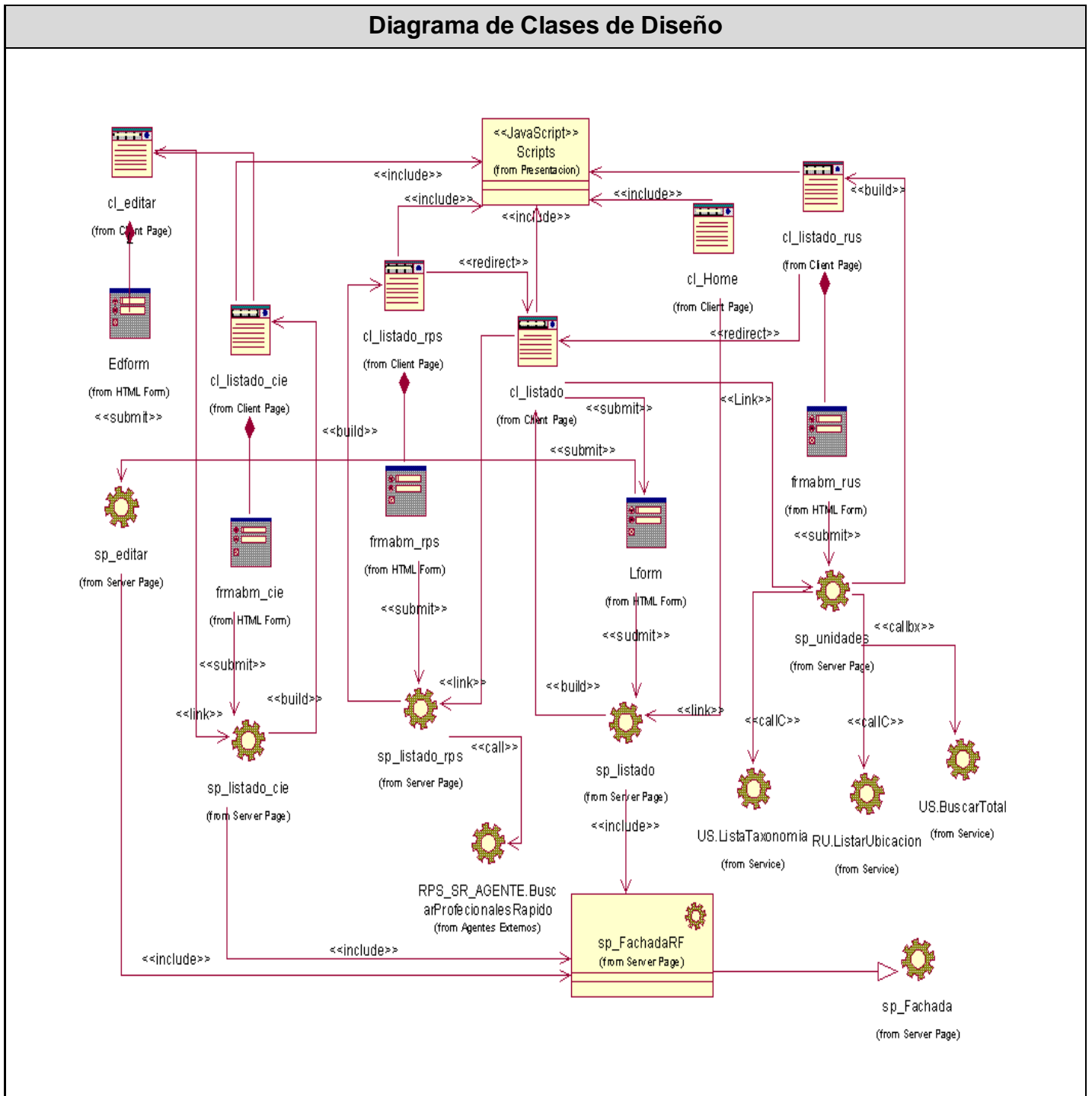


Figura 19. Diagrama de Clases del Diseño del Caso de Uso Gestionar Solicitud de Reparación [1/2].

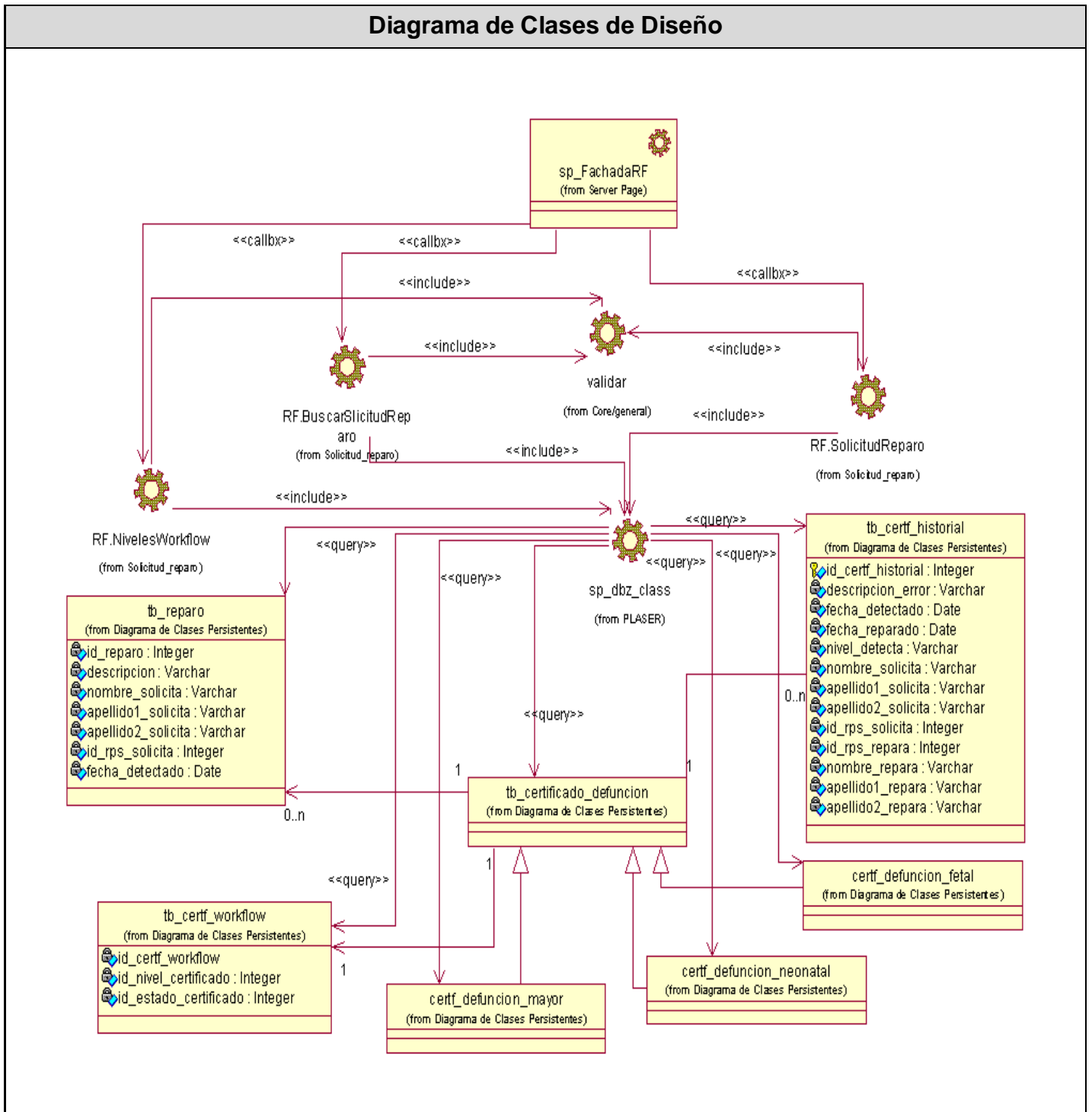


Figura 20. Diagrama de Clases del Diseño del Caso de Uso Gestionar Solicitud de Reparación [2/2].

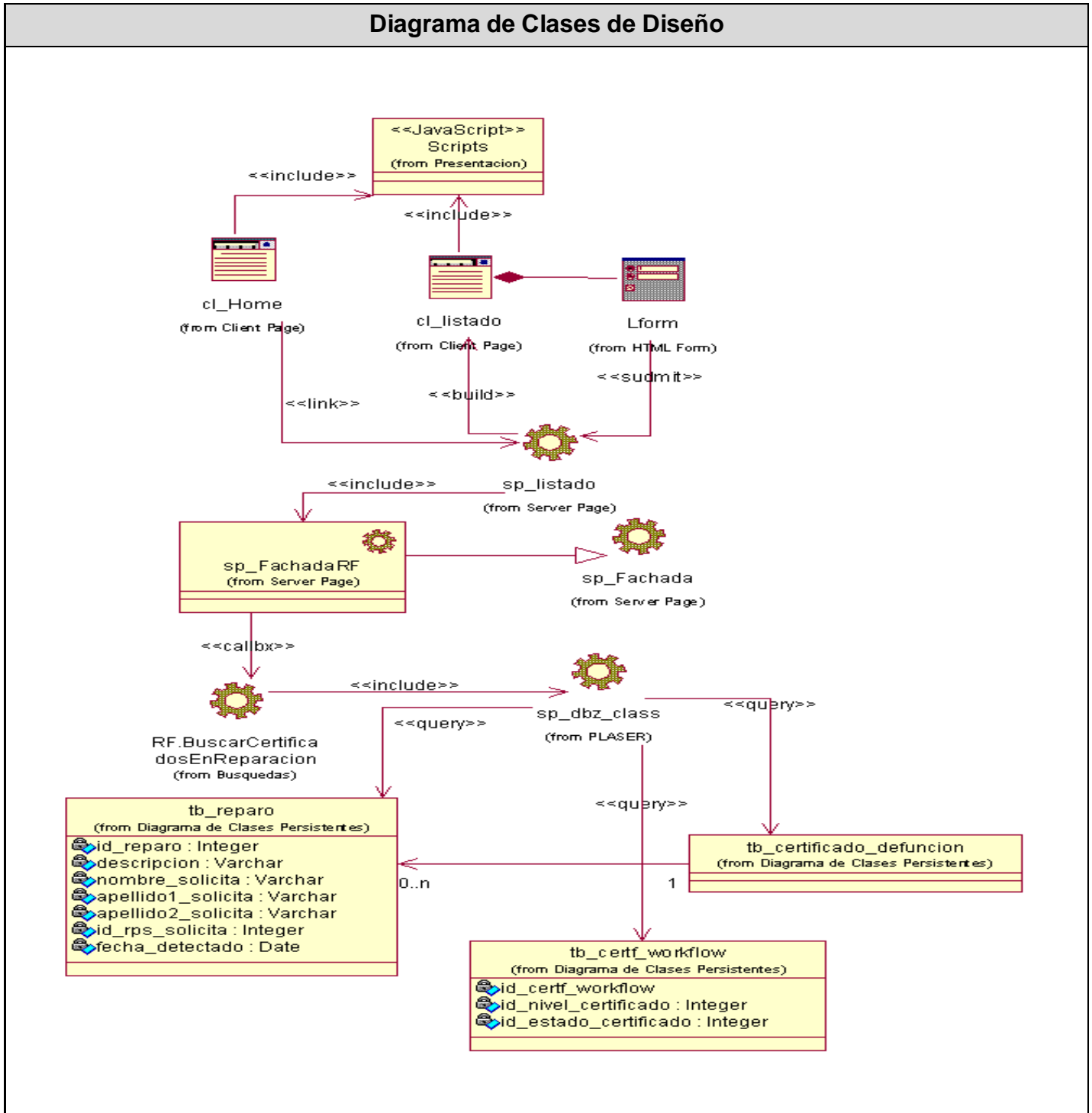


Figura 21. Diagrama de Clases del Diseño del Caso de Uso Buscar Solicitudes de Reparos.

2.6 Descripción de las Clases del Diseño.

2.6.1 Descripción de páginas clientes.

Nombre: cl_listado
<p>Descripción General: La clase cl_listado es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite a los Visualizadores o Editores, la realización de listados y búsquedas de información específica permitiendo generar documentos Portable Document Format (.pdf) y Microsoft Office Excel (.xls). A través de esta página se puede visualizar la información y el resultado de las búsquedas es paginada cada 10 elementos, permitiendo la movilidad por tales resultados, además de ir directamente a la última página. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> Gestionar Codificador Certificación Gestionar Codificador Muerte Violenta Gestionar Codificador Sitio Defunción Gestionar Codificador Confirmación Causas Gestionar Codificador Grupos de Edades Gestionar Codificador Vida Conyugal Gestionar Certificados Médicos de Defunción Gestionar Solicitud de Reparación Gestionar Reparación

Tabla 2. Descripción de la página cliente cl_listado.

Nombre: cl_editar
<p>Descripción General: La clase cl_editar es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite a los Editores Nacionales de Estadísticas luego de una búsqueda previa, realizar a través de esta página modificaciones de la información almacenada o simplemente visualizar información. El resultado de las búsquedas es paginada cada 10 elementos, permitiendo la movilidad por tales resultados, además de ir directamente a la última página. Este fichero es también utilizado a la hora de consultar los datos específicos de un certificado. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> Gestionar Codificador Certificación

<p>Gestionar Codificador Muerte Violenta Gestionar Codificador Sitio Defunción Gestionar Codificador Confirmación Causas Gestionar Codificador Grupos de Edades Gestionar Codificador Vida Conyugal Gestionar Certificados Médicos Defunción Gestionar Solicitud de Reparación Gestionar Reparación Buscar Certificados Médicos Defunción</p>

Tabla 3. Descripción de la página cliente cl_editar.

Nombre: cl_agregar
<p>Descripción General: La clase cl_agregar es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite capturar los datos que serán insertados en la base de datos del Registro y posee un conjunto de validaciones en JavaScript que no permiten realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Es utilizada en los siguientes casos de uso:</p> <p>Gestionar Codificador Certificación Gestionar Codificador Muerte Violenta Gestionar Codificador Sitio Defunción Gestionar Codificador Confirmación Causas Gestionar Codificador Grupos de Edades Gestionar Codificador Vida Conyugal Gestionar Certificados Médico de Defunción</p>

Tabla 4. Descripción de la página cliente cl_agregar.

Nombre: script
<p>Descripción General: La clase script representa un fichero Java Script que se ejecuta del lado del cliente, su código es interpretado por cualquier navegador Web. las funciones implementadas dentro de este fichero se utilizan en la validación de los parámetros de entrada, para validar que se introduzcan los datos obligatorios y evitar la entrada de caracteres especiales. Se emplean en combinación del código HTML de</p>

la página, para brindarle al usuario una interacción dinámica con la página Web (DHTML).

Tabla 5. Descripción de la clase script.

2.6.2 Descripción de páginas servidoras.

Nombre: sp_CFachada

Descripción General: La clase sp_CFachada es una clase que se encuentra dentro de PlaSer que su función principal es la comunicación de sistemas complicados y brinda los puntos de acceso entre estos. Se utiliza para separar la capa de presentación de la capa de acceso a datos.

Tabla 6. Descripción de la página servidora sp_CFachada.

Nombre: sp_dbz_class

Descripción General: 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; insertar, eliminar y actualizar datos. Esta clase está en la capa de negocio y forma parte de la plataforma de servicios PlaSer.

Tabla 7. Descripción de la página servidora sp_dbz_class.

Nombre: sp_CFachadaRF

Descripción General: La clase sp_CFachadaRF es una clase que representa a la clase sp_CFachada dentro del módulo y hereda de ella. Su función es modelar la comunicación con la capa de negocio.

Tabla 8. Descripción de la página servidora SP_CFachadaRF.

Nombre: sp_listado

Descripción General: La clase sp_listado es una clase que se ejecuta del lado del servidor en la Capa de

Presentación. Su actividad es construir una página cliente. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. Es utilizada en los siguientes casos de uso:

- Gestionar Codificador Certificación
- Gestionar Codificador Muerte Violenta
- Gestionar Codificador Sitio Defunción
- Gestionar Codificador Confirmación Causas
- Gestionar Codificador Grupos de Edades
- Gestionar Codificador Vida Conyugal
- Gestionar Certificados Médico de Defunción
- Gestionar Solicitud de Reparación
- Gestionar Reparación

Tabla 9. Descripción de la página servidora sp_listado.

Nombre: sp_agregar

Descripción General: La clase sp_agregar es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su actividad es construir la página cliente cl_agregar. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. Es utilizada en los siguientes casos de uso:

- Gestionar Codificador Certificación
- Gestionar Codificador Muerte Violenta
- Gestionar Codificador Sitio Defunción
- Gestionar Codificador Confirmación Causas
- Gestionar Codificador Grupos de Edades
- Gestionar Codificador Vida Conyugal
- Gestionar Certificados Médicos Defunción

Tabla 10. Descripción de la página servidora sp_agregar.

Nombre: sp_editar
<p>Descripción General: La clase sp_editar es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su actividad es construir la página cliente cl_editar. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. En el caso de no utilizarse la clase sp_actualizar, la clase sp_editar se encarga además de recibir los valores y enviarlos hacia el negocio a través de la clase sp_CFachadaRF la cual invoca al método del negocio correspondiente. Una vez concluida la ejecución de las responsabilidades de la clase sp_editar redirecciona su ejecución a la clase sp_listado. Es utilizada en los siguientes casos de uso:</p> <ul style="list-style-type: none"> Gestionar Codificador Certificación Gestionar Codificador Muerte Violenta Gestionar Codificador Sito Defunción Gestionar Codificador Confirmación Causas Gestionar Codificador Grupos de Edades Gestionar Codificador Vida Conyugal Gestionar Certificados Médicos Defunción Gestionar Solicitud de Reparación Gestionar Reparación Buscar Certificado Médico Defunción Buscar Solicitud de Reparación

Tabla 11. Descripción de la página servidora sp_editar.

Nombre: sp_insertar
<p>Descripción General: La clase sp_insertar es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Recibe los valores que se envían desde la página cliente cl_agregar en un submit y se encarga de enviarlos hacia la Capa de Negocio a través de la clase sp_CFachadaRF. Esta última invoca al método del negocio para la inserción de los datos involucrados y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase sp_insertar, la cual redirecciona su ejecución a la clase Sp_listado o Sp_agregar, en dependencia de la gestión que se esté realizando. Es utilizada en los siguientes casos de uso:</p>

Gestionar Codificador Certificación Gestionar Codificador Muerte Violenta Gestionar Codificador Sito Defunción Gestionar Codificador Confirmación Causas Gestionar Codificador Grupos de Edades Gestionar Codificador Vida Conyugal Gestionar Certificados Médicos Defunción
--

Tabla 12. Descripción de la página servidora sp_insertar.

Nombre: sp_actualizar

<p>Descripción General: La clase sp_actualizar es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Recibe los valores introducidos desde la página cliente cl_editar en un submit y se encarga de enviarlos hacia la Capa de Negocio a través de la clase sp_CFachadaRF la cual invoca al método del negocio para la actualización de los datos involucrados y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase sp_actualizar, la cual redirecciona su ejecución a la clase Sp_listado. Es utilizada en los siguientes casos de uso:</p>
--

Gestionar Codificador Certificación Gestionar Codificador Muerte Violenta Gestionar Codificador Sito Defunción Gestionar Codificador Confirmación Causas Gestionar Codificador Grupos de Edades Gestionar Codificador Vida Conyugal Gestionar Certificados Médicos Defunción
--

Tabla 13. Descripción de la página servidora sp_actualizar.

Nombre: sp_eliminar

<p>Descripción General: La clase sp_eliminar es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su actividad es recibir los valores de la página cliente cl_listado para posteriormente enviarlos hacia la Capa de Negocio a través de la clase sp_CFachadaRF. Esta última invoca al método del negocio para la eliminación de los datos involucrados y una vez concluida la ejecución de sus</p>

responsabilidades devuelve una respuesta a la clase `sp_eliminar` la cual redirecciona su ejecución a la clase `sp_listado`. Es utilizada en los siguientes casos de uso:

- Gestionar Codificador Certificación
- Gestionar Codificador Muerte Violenta
- Gestionar Codificador Sitio Defunción
- Gestionar Codificador Confirmación Causas
- Gestionar Codificador Grupos de Edades
- Gestionar Codificador Vida Conyugal

Tabla 14. Descripción de la página servidora `sp_eliminar`.

Nombre: `sp_listado_rps`

Descripción General: La clase `sp_listado_rps` es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su actividad es construir una página cliente. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. En esta clase server page se ejecutan todos los nomencladores del Registro Personal de Salud que son necesarios a la hora de realizar una búsqueda para obtener los datos del personal de salud. Es utilizada en los siguientes casos de uso:

- Gestionar Solicitud de Reparó
- Gestionar Certificados Médicos Defunción
- Gestionar Reparó
- Buscar Certificados Médicos Defunción

Tabla 15. Descripción de la página servidora `sp_listado_rpn`.

Nombre: `sp_listado_rpn`

Descripción General: La clase `sp_listado_rpn` es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su actividad es construir una página cliente. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. Esta clase server page brinda los datos de los partos Institucionales. Es utilizada en el siguiente caso de uso:

Gestionar Certificados Médicos Defunción

Tabla 16. Descripción de la página servidora sp_listado_rpn.

Nombre: sp_listado_cie

Descripción General: Descripción General: La clase sp_listado_cie es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su actividad es construir una página cliente. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. Esta clase server page permite realizar una búsqueda de los códigos y sus descripciones. Es utilizada en los siguientes casos de uso:

Gestionar Certificados Médicos Defunción
Gestionar Solicitud de Reparó

Tabla 17. Descripción de la página servidora sp_listado_cie.

Nombre: sp_listado_rpob

Descripción General: La clase sp_listado_rpob es una clase que se ejecuta del lado del servidor en la Capa de Presentación. Su actividad es construir una página cliente. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML. Esta clase server page permite realizar una búsqueda de los datos del paciente. Es utilizada en el siguiente caso de uso:

Gestionar Certificados Médicos Defunción

Tabla 18. Descripción de la página servidora sp_listado_rpob.

Nombre: RF_EliminarCertificación

Descripción General: Es una Clase que se ejecuta al lado del servidor en la Capa del Negocio. Valida el parámetro de entrada que le llega desde el cliente que solicita ese servicio web. Si el valor es un entero

positivo establece una conexión directa con el servidor de base de datos a través de la clase de acceso a datos `dbz_class`. Este método tiene la función de ejecutar las consultas necesarias para eliminar los datos de la certificación, una vez verificada su existencia en la base de datos. Terminada la consulta SQL, se cierra la conexión con la base de datos para no sobrecargar el servidor. Si este método detecta algún error relacionado con los parámetros de entrada o las consultas a la base de datos, retorna una excepción que indica el error cometido, de lo contrario retorna `true` que indica que se eliminaron correctamente los datos.

Tabla 19. Descripción de la página servidora RF_EliminarCertificación.

Nombre: RF_InsertarCertificación

Descripción General: Es una Clase que se ejecuta al lado del servidor en la Capa del Negocio. Valida el parámetro de entrada que le llega desde el cliente que solicita ese servicio web. Si el valor de los parámetros de entrada es correcto, establece una conexión con el servidor de base de datos a través de la clase de acceso a datos `dbz_class`. Este método tiene la función de ejecutar las consultas necesarias para insertar los datos de una certificación. Terminada la consulta SQL, se cierra la conexión con la base de datos para no sobrecargar el servidor. Si este método detecta algún error relacionado con los parámetros de entrada o las consultas a la base de datos, retorna una excepción que indica el error cometido, de lo contrario retorna `true`, que indica que se insertaron correctamente los datos.

Tabla 20. Descripción de la clase servidora RF_InsertarCertificación.

Nombre: RF_ListarCertificación

Descripción General: Es una Clase que se ejecuta al lado del servidor en la Capa del Negocio. Establece una conexión con el servidor de base de datos a través de la clase de acceso a datos `dbz_class`. Este método tiene la función de ejecutar las consultas necesarias para retornar un listado con los datos de las certificaciones existentes. Su implementación permite retornar documentos en formatos XML, Acrobat Reader (PDF) y Microsoft Excel (XSL). Terminada la consulta SQL, se cierra la conexión

con la base de datos para no sobrecargar el servidor.

Tabla 21. Descripción de la clase servidora RF_ListarCertificación.

Nombre: RF_ModificarCertificación

Descripción General: Es una Clase que se ejecuta al lado del servidor en la Capa del Negocio. Valida los parámetros de entrada que le llega desde el cliente que solicita ese servicio web. Si el valor de los parámetros de entrada es correcto, establece una conexión con el servidor de base de datos a través de la clase de acceso a datos dbz_class. Este método tiene la función de ejecutar las consultas necesarias para modificar los datos de una certificación. Terminada la consulta SQL, se cierra la conexión con la base de datos para no sobrecargar el servidor. Si este método detecta algún error relacionado con los parámetros de entrada o las consultas a la base de datos, retorna una excepción que indica el error cometido, de lo contrario retorna true que indica que se modificaron correctamente los datos.

Tabla 22. Descripción de la clase servidora RF_ModificarCertificación.

En este capítulo, se abordaron algunos elementos del Modelo de Diseño del Registro de Fallecidos, como continuación del análisis realizado. Se hace una especificación más detallada de la propuesta del sistema, haciendo uso de las clases del diseño y las descripciones de las clases utilizadas en los diagramas de clases del diseño, correspondientes a los casos de uso que se explican en el documento. Además constituyó la entrada fundamental para el desarrollo del Flujo de Trabajo de Implementación.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

En este capítulo, se argumenta la integración del Registro de Fallecidos con otros componentes del SISalud. Se muestra el modelo de implementación, los diagramas de componentes, el diagrama de despliegue, permitiendo comprender mejor el sistema implementado. Describe de forma breve los algoritmos más complejos que intervienen en la funcionalidad del módulo; así como explicar los estándares, codificaciones y tratamientos de errores utilizados. Además, se elabora el modelo de prueba.

3.1 Dependencias y Relaciones con otros Sistemas.

El módulo RF para su funcionamiento necesita consumir servicios de otros componentes de SISalud porque la mayor parte de la información necesaria para llenar un certificado médico, así como los datos para su respectivo flujo, ya se gestionan en dichos componentes. Los módulos son los que a continuación se mencionan.

3.1.1 Registro de Ubicación (RU) y Registro de Localidades (RL).

El RU y RL gestionan la información de las Provincias, Municipios, Localidades, Calles y Manzanas del país. Estos módulos permiten utilizar los servicios que brinda para registrar la provincia y municipio del sitio de defunción así como el municipio del futuro enterramiento.

3.1.2 Registro de Partos y Nacimientos (RPN).

Los Certificados Médicos de Defunción Fetal y Neonatal, requieren de una serie de datos de la madre y del feto para conformar los mismos. El RPN provee al RF de los datos del parto y del nacimiento si se produjo, así como las características del recién nacido. La información que brinda este módulo al Registro de Fallecidos es vital a la hora de extraer reportes, lo que conlleva a datos estadísticos relevantes y determinantes para el Sistema Nacional de Salud.

3.1.3 Registro de Población (RPOB).

El Registro de Población permite gestionar la información de las Historias de Salud Familiar de la

población, permitiendo la dispensarización de cada paciente y la planificación de las acciones de salud. Este registro permite obtener los datos del fallecido, información que es importante en llenado de un CMD. Como el nombre, sus apellidos, dirección particular, fecha de nacimiento, Carné de Identidad, si es extranjero su tipo de residencia y otros como el color de la piel y su ocupación.

3.1.4 Registro de Personal de la Salud (RPS).

Los CMD son confeccionados por los especialistas y profesionales de la salud, los cuales deben plasmar en ellos sus datos personales y profesionales. Este módulo gestiona la información de todos los profesionales de salud del país, ya sean médicos o no médicos posibilitando obtener los datos necesarios de los certificantes y del funcionario actuante en los Certificados Médicos de Defunción.

3.1.5 Registro de Unidades de Salud (RUS).

Los Certificados Médicos de Defunción son registrados en las unidades de salud a las que pertenecen los médicos que certifican el fallecimiento o donde ocurre este, la información necesaria de estas unidades se obtiene del RUS, como el nombre de la unidad y su respectivo código.

3.1.6 Registro de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud (RCIE).

Este registro controla la información de las enfermedades que se encuentran en el libro del CIE-10, agrupándolas según criterios específicos. El RF recibe la codificación de las enfermedades que causaron la muerte de dicho registro, facilitando la creación de futuros reportes estadísticos vitales para el sistema nacional de Salud.

3.1.7 Sistema Autorización, Autenticación y Auditoría (SAAA).

Es el encargado de garantizar la seguridad. Está basado en el modelo de Autenticación, Autorización y Auditoría (AAA). La autenticación debe ser la primera acción del usuario en el sistema y consiste en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica. Si el usuario autenticado no se encuentra registrado se reporta un error de

acceso. En caso contrario, se autoriza su acceso y se crea un certificado digital y se retornan todos los datos y permisos del usuario, desglosado por módulos. Cada petición de usuario, autorizada o no, es registrada, así como el día, mes, año, hora, minuto, segundo en que se registra y si fue o no autorizada. (17)

El certificado digital del SAAA brinda el nivel en que se autentica un usuario, en el caso del RF se utiliza esta información para garantizar la seguridad pues se va a almacenar en un token de 32 caracteres en el que se guardan el usuario, la contraseña, los módulos a los que puede acceder ese usuario, los niveles (Nacional, Unidad de Salud) y los privilegios (Editor, Visualizador).

3.1 Implementación.

Con el desarrollo del Flujo de Trabajo de Implementación se comienza con el resultado del diseño y se define cómo se organizan las clases y objetos en términos de componentes, cuáles nodos se utilizarán y la estructura de capas de la aplicación. Uno de los principales propósitos de este flujo consiste en desarrollar la arquitectura y el sistema como un todo.

3.1.1 Diagrama de Despliegue.

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene. Cada Hardware se representa como un nodo. Un nodo es un elemento donde se ejecutan los componentes, representan el despliegue físico de estos componentes.

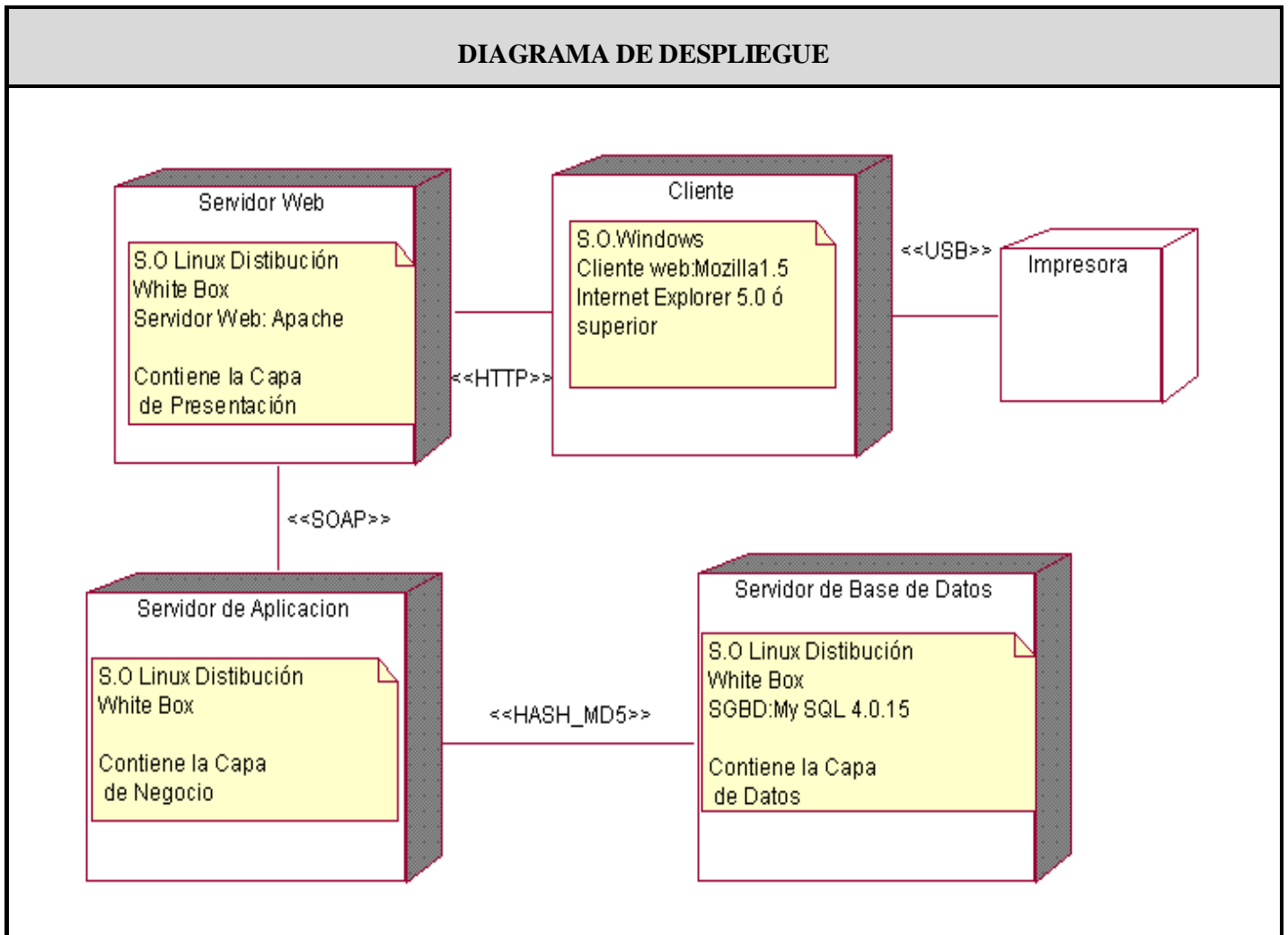


Figura 22. Diagrama de Despliegue.

3.1.2 Modelo de Implementación.

Define cómo se organizan las clases y objetos en componentes (ficheros fuentes, binarios, ejecutables y otros), cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura en capas de la aplicación.

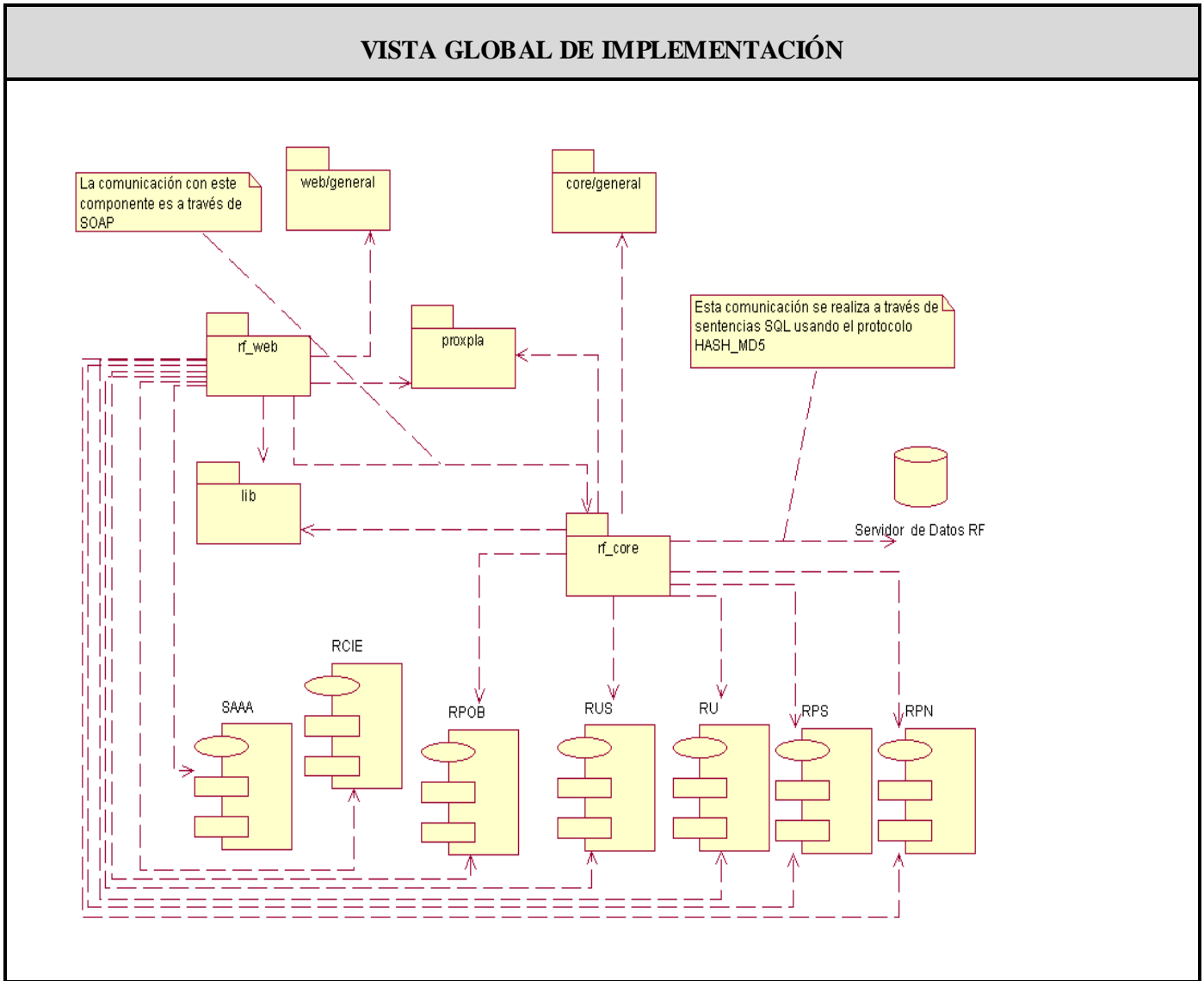


Figura 23. Vista Global de Implementación del RF.

3.1.3 Diagrama de Componentes.

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases del modelo de diseño; los mismos son creados, modificados o eliminados en el proceso de implementación y constituyen la versión del producto. Un diagrama de componentes se representa como un grafo de elementos, tales como componentes y subsistemas unidos por medio de relaciones de dependencia. Cada

diagrama describe un apartado del sistema. Uno de los usos principales es que puede servir para ver que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema. A continuación se muestran diagramas de componentes asociados a los subsistemas de implementación identificados para el RF, los demás se encuentran en el expediente de proyecto.

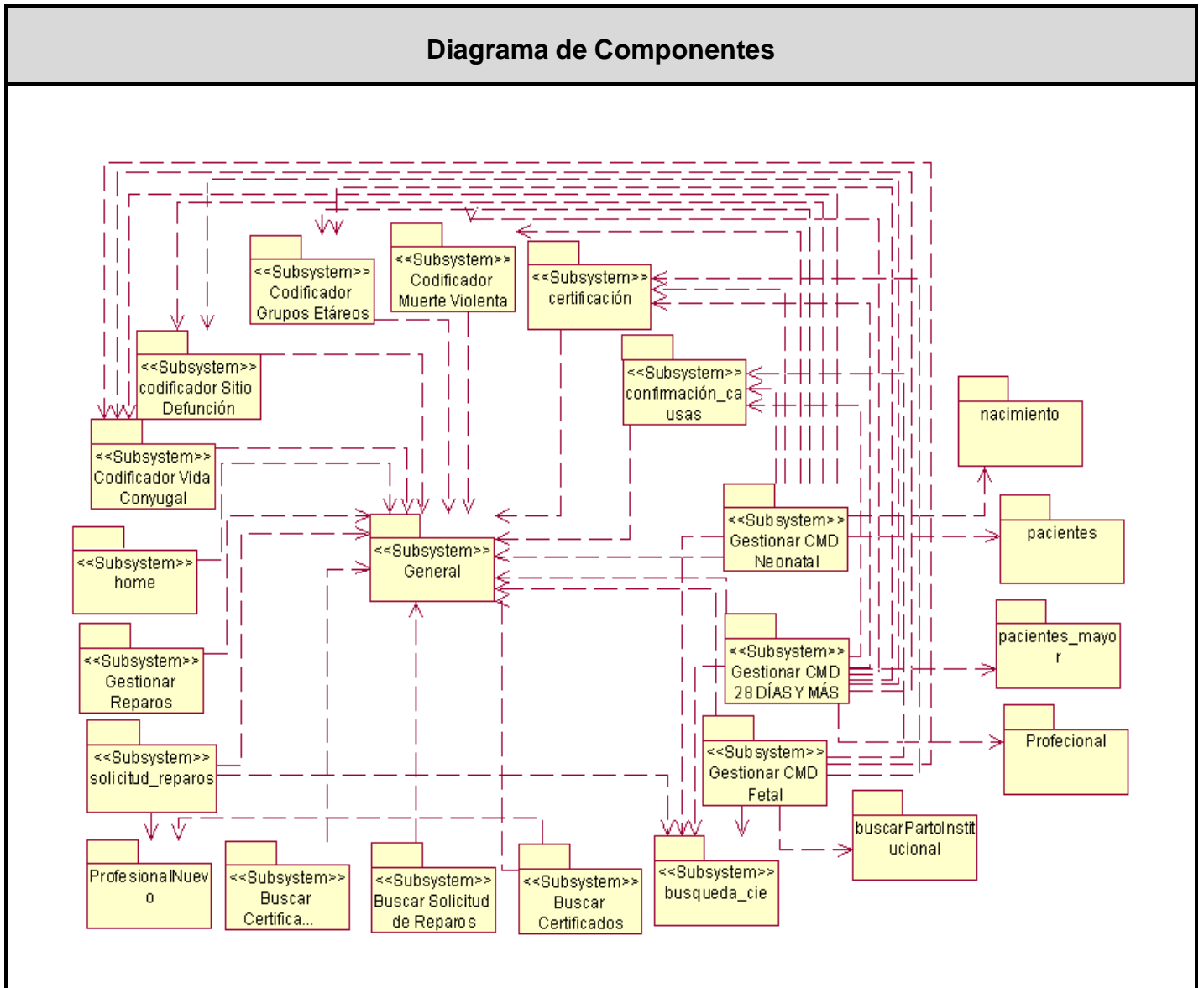


Figura 24. Diagrama de Componentes del Subsistema `rf_web`.

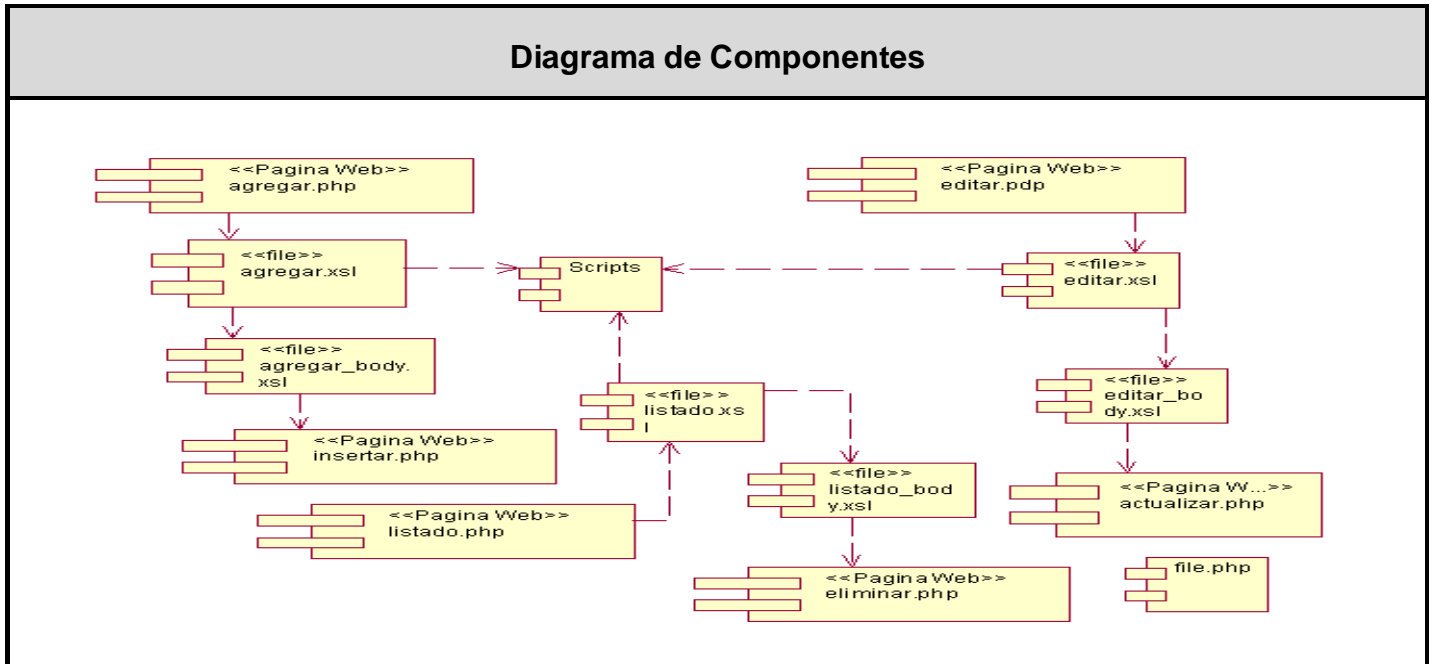


Figura 25. Diagrama de Componentes del Subsistema Certificación.

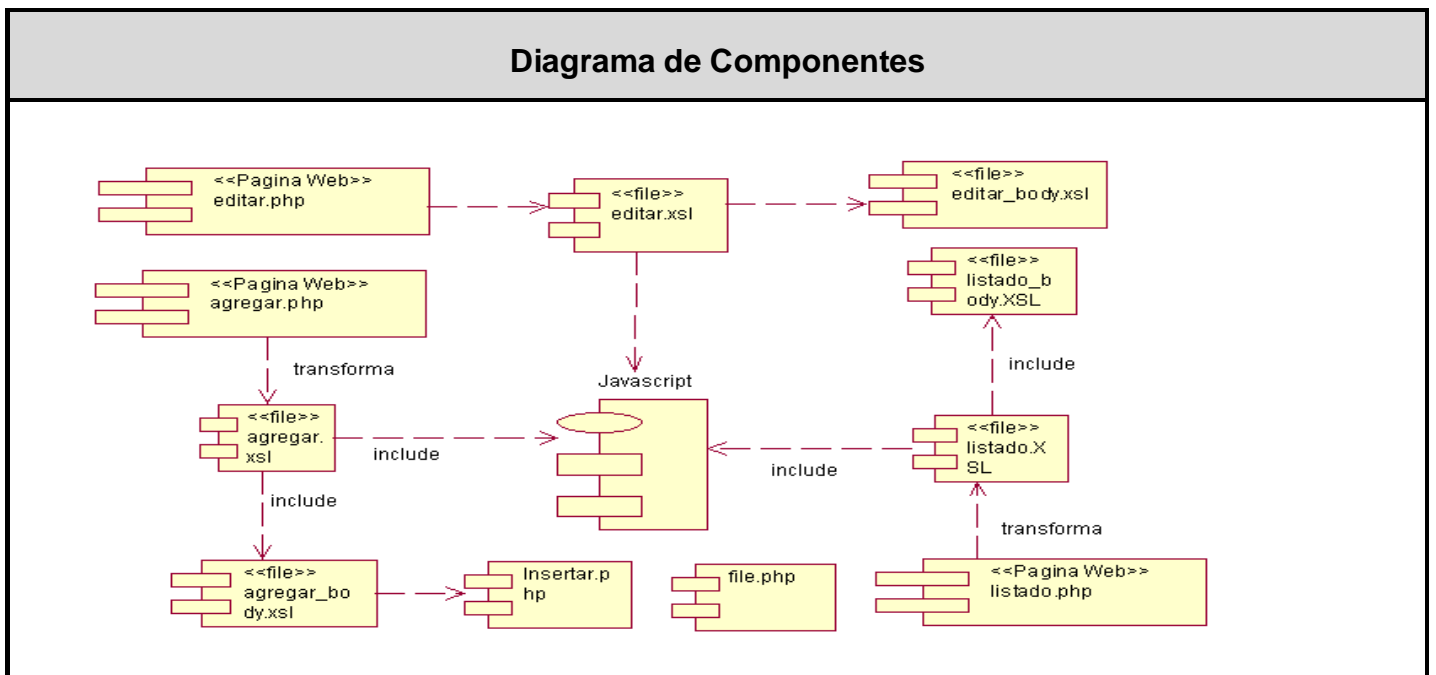


Figura 26. Diagrama de Componentes del Subsistema 28 días y más.

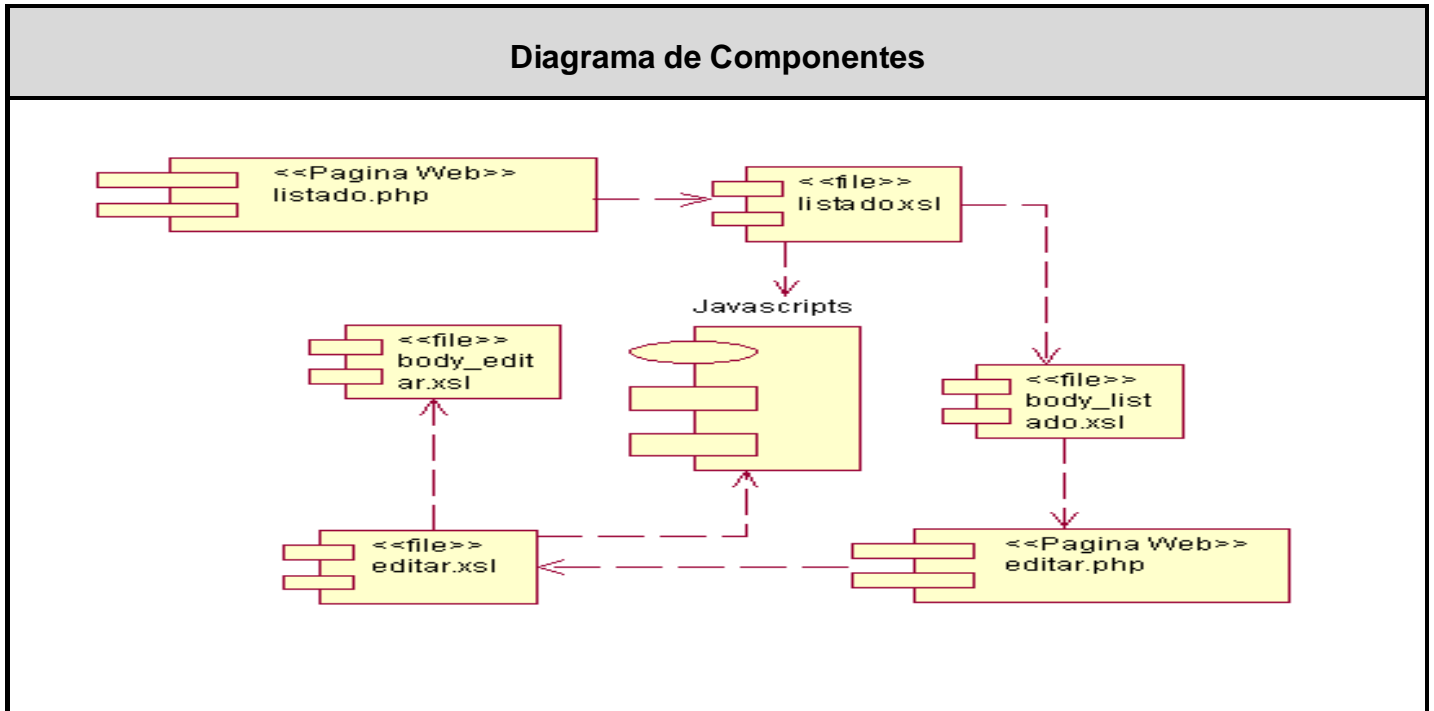


Figura 27. Diagrama de Componentes del Subsistema Gestionar Reparos.

3.2 Descripción de los Métodos (agentes o métodos más complejos).

RF_ NivelesWorkflow.

Es el encargado de actualizar el nuevo nivel del Certificado en el Workflow, para eso recibe una serie de parámetros de entrada, como son el tipo de certificado, estado del certificado, el identificador del certificado, el nivel en el que se encuentra, entre otros.

En dependencia del nivel que posee el certificado que está en el flujo, el método realiza diferentes acciones; si el nivel es Nacional, archiva el certificado eliminándolo del workflow lo que equivale a borrar su ocurrencia en la tabla tb_certf_workflow; en caso de no suceder la primera acción el nivel se actualiza en la tabla antes mencionada convirtiendo el CMD apto para revisarse en el nivel superior; si su nivel es provincial, inserta los códigos necesario para la ocupación, sitio de defunción y causas de muerte ya insertadas una vez creado el CMD.

RF_SolicitudReparo.

Se encarga de las Solicitudes de Reparación que se le puedan realizar a un certificado, donde recibe una serie de parámetros de entrada para llevar a cabo esta funcionalidad. Para realizar la solicitud de reparación se actualiza el nivel del certificado, enviándolo a la unidad de salud donde fue registrado, además de cambiar el estado del certificado, siendo el nuevo estado “En reparación”. Si el certificado ya ha sido reparado, se inserta la nueva traza del certificado en la tabla `tb_cert_historial`, sino se actualizan los datos del mismo en dicha tabla. Antes de insertar los datos del reparo busca los datos del técnico que realizó la solicitud y son insertados junto con la descripción del error que se le encontró al certificado en la tabla `tb_reparo`.

RF_InsertarReparo.

Este método es el responsable de realizar los reparos a los CMD que lo requieren, la primera acción es actualizar el estado de las causas de muerte que fueron reparadas, pasando su estado de activa a inactiva e inserta la nueva descripción de las causas de muerte con su estado en activa. Luego actualiza los datos del certificado reparado en la tabla `historial` y vuelve a introducir el certificado al flujo actualizando el nivel y el estado el cual será “Reparado”, además de eliminar la solicitud de reparo de la base de datos.

RF_AgenteBuscarPaciente.

El método realiza una búsqueda de todos los pacientes que cumplen con los parámetros de entrada. Recibe como parámetros el carnet de identidad, nombre, primer apellido, segundo apellido, sexo, municipio y provincia, si los parámetros son pasados vacíos el método devuelve todos los pacientes. Para realizar esta búsqueda se hace a través del método `BuscarMadreRPN` que se encuentra en el Registro de Población, el cual invoca a un servicio web del Registro de Ciudadano, `BuscarCiudadanoTotal` donde se obtienen los datos personales del paciente fallecido. Para conformar la respuesta final se concatenan las repuestas del Registro de Población con la del Registro de Ciudadano y se retornan los datos del paciente necesarios para el llenado del CMD.

RF_AgenteBuscarUnidadSalud.

Este agente devuelve un arreglo con el nombre de la unidad de salud, su identificador, el identificador de

la provincia y el municipio al cual pertenecen estas unidades. Recibe como parámetros de entrada el identificador de la provincia, del municipio y de la unidad de salud.

RF_AgenteListarProvinciasMunicipios.

Se encarga de buscar toda la información referente a las Provincias, Municipios y Localidades en dependencia de la profundidad. Para mostrar esta información, dado la profundidad que se pasa como parámetro, lo primero que hace el método es invocar un servicio web del Registro de Ubicación, ListarProvincias para obtener los datos de las Provincias, Municipios y Localidades.

Este agente es de gran importancia en la capa de presentación porque la información que retorna es utilizada en múltiples ocasiones y por muchas opciones del sistema.

RF_BuscarDatosCertificadoNeonatal.

Es utilizado para obtener información correspondiente a los datos del nacimiento de un niño, recibe como parámetros de entrada el identificador del ciudadano de la madre, el del niño y el identificador propio del niño que le da el Registro de Partos y Nacimientos.

Este método invoca varios servicios del RPN, mediante los cuales se logra conformar un XML con los datos necesarios, tanto de la madre como del recién nacido, los cuales son importantes en el llenado del Certificado de 0 a 27 días (Neonatal). Ahorra tener que invocar varias veces estos servicios.

3.3 Estándares del Diseño, Codificación y Tratamientos de Errores.

Para lograr que exista homogeneidad entre las aplicaciones del Sistema de Información para la Salud, se han definido una serie de elementos comunes tanto para el diseño como para la codificación y tratamiento de errores en los códigos fuentes de los módulos y los mensajes que se emitan, facilitando su reconocimiento y el uso que se haga de ellos.

3.3.2 Estándares de Diseño.

La estructura base de las aplicaciones es la misma para todos los módulos: las pantallas más usadas, los modelos establecidos, las rutas de navegación, las utilidades básicas, la organización de los elementos en

pantalla y el diseño de identificadores serán comunes para todos. Para el diseño del Módulo RF se siguieron las pautas y el mismo diseño que el SISalud; trayendo como principal ventaja la uniformidad en la estructura de las páginas web de todos los módulos integrados. Para lograr esta uniformidad se debe 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 pts y los submenús a 6 pts. 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 pts de cada edit.
- ✓ En el caso de tablas generadas por búsquedas, que ordenan una serie de elementos, y necesiten selección, se harán a través de checkboxes justificados a la izquierda de la tabla. Siempre habrá un checkbox en la fila de título, también a la izquierda, que facilite seleccionar todos. Es necesario destacar que estas tablas pueden tener una cantidad grande de líneas generadas por la búsqueda, por lo que debe quedar pautado que hasta 10 resultados la tabla funcione con scroll, pero más de esta cantidad será entonces por paginado, con 10 resultados por página.

Se muestra la pantalla principal del Registro de Fallecidos a continuación como ejemplo de que se ha seguido con los estándares de diseño mencionados.

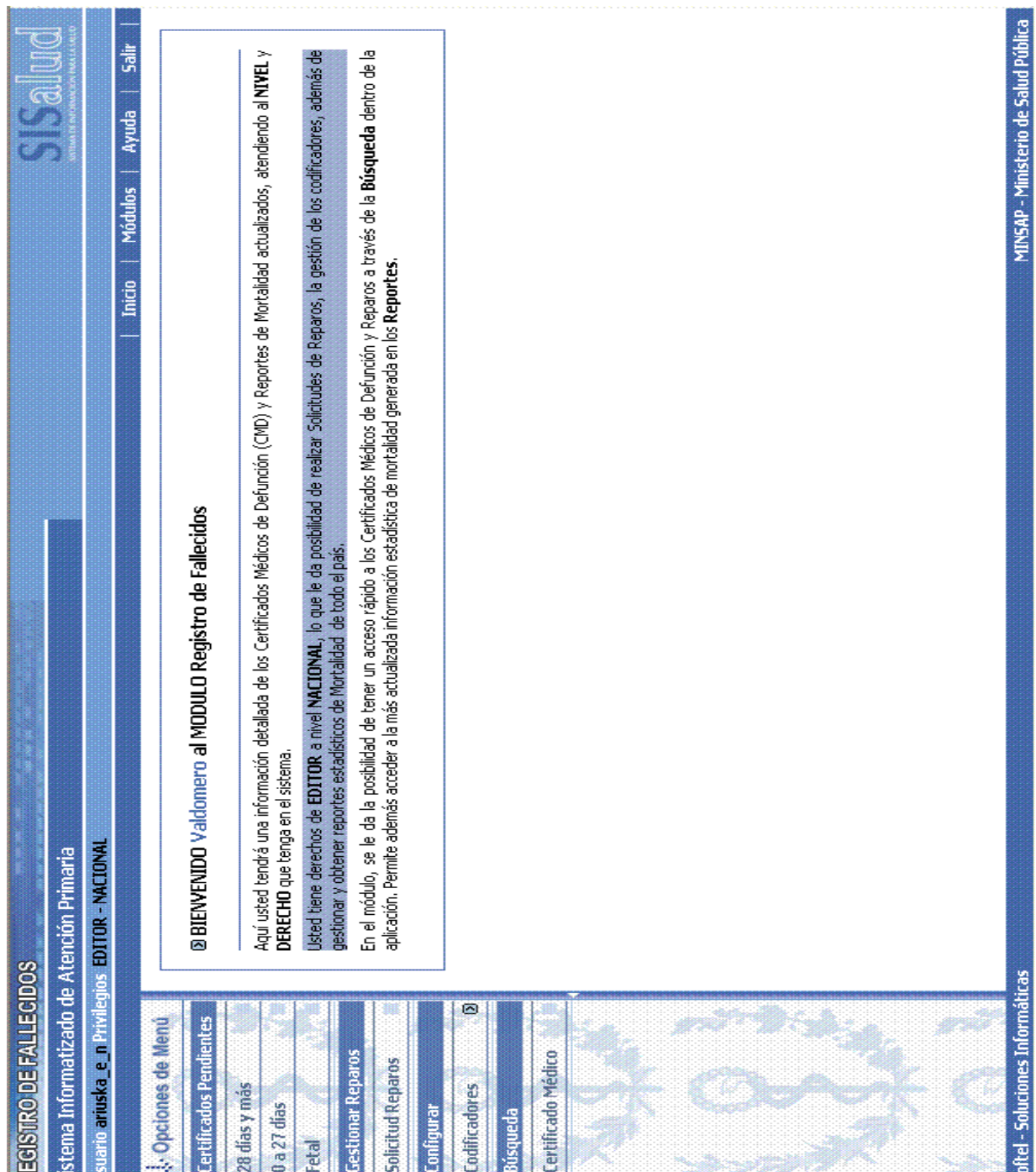


Figura 28. Pantalla principal del módulo Registro de Fallecidos.

3.3.2 Estándares de Codificación

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

Las etiquetas de apertura y cierre del lenguaje serán de la forma `<?php ¿>`, ya que siempre están disponible en cualquier configuración. Se harán uso de los arreglos predefinidos para el manejo de los valores enviados por el usuario `$_GET`, `$_POST`, `$_FILES` evitando el uso de `$_REQUEST`.

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

Todos los campos id van a comenzar con el identificador (id) seguido del nombre del campo. Ejemplo id_79efunción79ión Los arreglos empezarán con el identificador array y las palabras no se separarán con el 79efunció “_”. Ejemplo: arraybuscar.

En el caso de las clases se pondrá delante la letra C. Ejemplo: CFachadaRF y en el de los métodos no se usarán abreviaturas y las palabras continuas deben comenzar con mayúsculas. Ejemplo: BuscarCertificadosReparados.

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 utilizarán los caracteres “/* */”.

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

El idioma de las clases auxiliares como sesión y error, será el inglés para garantizar la homogeneidad con

las programadas en este ámbito en el mundo, en el caso de los Servicios Web y la interface de administración se usará el español para esclarecer los objetivos de cada método o script a utilizar.

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

El almacenamiento de la información será en scripts SQL para construir la base de datos e interactuar con ella desde las aplicaciones. Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas Ej: SELECT * FROM tb_certificado_defuncion.

Los nombres de las tablas deben ir en minúsculas y cada palabra separada por línea abajo “_”. (Ejemplo: tb_muerte_violenta) En el caso de los XSL será con el mismo nombre que el fichero de la capa de presentación.

Los controles seguirán el siguiente tratamiento:

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

Tabla 23. Estándares para los controles.

Las páginas HTML se harán sin incluir código y todas las funciones JavaScript que se usarán se escribirán dentro de ficheros “.js”. Para la capa de datos tienen que nombrar la base de datos poniendo el identificador del proyecto “APS” seguido del carácter “_” y del nombre del módulo. Ejemplo: bd_rf Los campos de la base de datos se nombrarán igual que las variables. Cada método que se defina debe seguir la siguiente estructura, acrónimo del módulo. Nombre del método. Ej: RF_BuscarCertificadosEnReparacion.

3.3.3 Tratamiento de Errores

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. Son una forma clara para controlar los errores sin confundir el código con muchas instrucciones de control del error. Cuando se verifica un error se pone en marcha una excepción que, si se recibe enseguida, permite gestionar un error. Para depurar los errores se utiliza JavaScript, validando los formularios y evitando consultas a la base de datos sin sentido o que tenga malas intenciones como son las inyecciones SQL, garantizando además que los datos introducidos por los usuarios sean validos, o les sea posible corregirlos en caso contrario.



Figura 29. Mensaje de error (JavaScript).



Figura 30. Mensaje de error (JavaScript).

Otros errores en la capa de negocio serán tratados devolviendo un SOAP_FAULT, cuyos elementos FaultCode, FaultString, FaultActor describiremos a continuación:

FaultCode.

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

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

FaultString.

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

FaultActor.

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

3.4 Pruebas.

El flujo de pruebas le presta servicios a los demás flujos. Su principal objetivo es evaluar o valorar la calidad del producto a través de:

- ✓ Buscar y documentar errores
- ✓ Validar el cumplimiento de requerimientos
- ✓ Validar desempeño
- ✓ Dar una indicación de calidad.

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

Caso de Uso: Gestionar Certificados Médicos Defunción

Entrada	<p>El caso de prueba se ejecuta dada la necesidad de validar el correcto funcionamiento de la aplicación dado un conjunto de acciones a realizar por el usuario final del sistema, para la gestión de los CMD, específicamente la opción de insertar el CMD de 28 días y más.</p> <p>- Para insertar un CMD se exigen un conjunto de datos obligatorios en un determinado formato. Estos datos se dividen en seis sesiones identificadas por pestañas y para pasar de una pestaña a otra tiene que haber llenado de forma correcta los datos en la pestaña anterior.</p> <p>En la primera pestaña el usuario (técnico de estadística) debe seleccionar el paciente fallecido e insertar los datos requeridos.</p> <p>En la segunda pestaña se requieren los datos en caso de ser una mujer el fallecido, sobre posibles abortos o partos antes del fallecimiento.</p> <p>En la tercera pestaña se insertan las causas de muerte registradas por el médico certificante.</p> <p>En la cuarta pestaña solo si es caso de ser una muerte violenta.</p> <p>En la quinta pestaña se recogerán los datos de los médicos certificantes.</p> <p>Y por último en la sexta pestaña se podrán insertar datos con fines estadísticos.</p>
Resultados	<p>Se muestra un mensaje de alerta indicando que:</p> <ul style="list-style-type: none"> ✓ Faltan Datos por llenar del Fallecido.

	<ul style="list-style-type: none"> ✓ Faltan parámetros por llenar (campos obligatorios). ✓ No debe introducir caracteres extraños. <p>Se muestra un mensaje indicando que ya existe el certificado (si verifica que ya existe el registro de dicho fallecimiento).</p> <p>Si no se detectan errores se inserta el CMD en la base de datos.</p>
Condiciones	El usuario debe estar autenticado en el sistema y tener los permisos necesarios para ejecutar la acción.

Tabla 24. Descripción Caso de Prueba Gestionar CMD.

Caso de Uso: Gestionar Solicitud de Reparación

Entrada	<p>El caso de prueba se ejecuta dada la necesidad de validar el correcto funcionamiento de la aplicación dado un conjunto de acciones a realizar por el usuario final del sistema, para la gestión de los CMD, específicamente la opción de realizar una solicitud de reparación a un CMD que lo requiere.</p> <p>- En los niveles superiores al de unidad de salud, se requiere revisar las causas de muertes descritas por el médico, si se le encuentra un error, se realiza una descripción del error y se ejecuta la acción Solicitud Reparación, en caso de no se le da Aceptar y el certificado continúa su flujo.</p>
Resultados	<p>Se muestra un mensaje de alerta indicando que:</p> <ul style="list-style-type: none"> ✓ Por favor, debe describir el error para realizar el reparación. <p>Si no se detectan errores se ejecuta correctamente el método correspondiente que actualiza el nivel y el estado del certificado.</p>
Condiciones	<p>El usuario debe estar autenticado en el sistema y tener los permisos necesarios para ejecutar la acción.</p> <p>El caso de uso se ejecuta en todos los niveles excepto en el de unidad de salud.</p>

Tabla 25. Descripción Caso de Prueba Gestionar Solicitud de Reparación.

Caso de Uso: Gestionar Reparos

Entrada	<p>El caso de prueba se ejecuta dada la necesidad de validar el correcto funcionamiento de la aplicación dado un conjunto de acciones a realizar por el usuario final del sistema, para la gestión de los CMD, específicamente la opción de realizar un reparo a un CMD que se le ha solicitado uno.</p> <p>- En el nivel de unidad de salud cuando se buscan todas las solicitudes de reparo de dicha unidad y se escoge un CMD, se le incorpora la enmienda señalada insertando en el campo de texto correspondiente y se ejecuta la acción Aceptar.</p>
Resultados	<p>Se muestra un mensaje de alerta indicando que:</p> <ul style="list-style-type: none"> ✓ No debe introducir caracteres extraños. ✓ Seleccione la fecha de reparo. ✓ Escriba el número de orden del certificado. ✓ Debe escribir el tiempo aproximado de esta causa de muerte. <p>Si no se detectan errores se ejecuta correctamente el método correspondiente que actualiza el nivel y el estado del certificado.</p>
Condiciones	<p>El usuario debe estar autenticado en el sistema y tener los permisos necesarios para ejecutar la acción.</p> <p>El caso de uso solo se ejecuta en el nivel de unidad de salud.</p>

Tabla 26. Descripción Caso de Prueba Gestionar Reparos.

En el capítulo se hizo un estudio de los estándares de diseño, la codificación empleada en la implementación y el tratamiento de errores. Se expuso una muestra de los diagramas componentes, y el diagrama de despliegue. También se describieron los métodos que impulsaban complejidad a la hora de entenderlos. De igual forma se realizó una descripción de los casos de prueba verificando la integridad del sistema.

CONCLUSIONES

Al culminar el estudio sobre el control de fallecidos que se realiza actualmente en el SNS, con la visión clara y segura de mejorar a gran escala la gestión de dicho control y valorando las tareas y objetivos propuestos, se llegó a las siguientes conclusiones:

- ✓ Se realizó un estudio detallado de los sistemas y aplicaciones relacionados con el control de los fallecidos existentes en la actualidad en Cuba y el mundo.
- ✓ Se realizó el Modelo de Diseño a partir del resultado del Modelo de Análisis ya existente en el trabajo de diploma del curso 2006-2007.
- ✓ Se diseñó el Flujo de Trabajo de Implementación y el flujo de Prueba quedando completa la documentación de Ingeniería de Software.
- ✓ Se implementó el RF, aplicando la arquitectura definida por el MINSAP y la plataforma de servicios PlaSer.
- ✓ Se identificaron y utilizaron las relaciones de integración con otros componentes del SISalud.

RECOMENDACIONES

Se recomienda:

- ✓ Desplegar el Registro de Fallecidos para su uso en SISalud y que sus servicios puedan ser consumidos por cualquier aplicación informática que necesite de los mismos.
- ✓ Hacer uso de la aplicación insertando los CMD en las unidades de salud correspondientes y las demás funcionalidades en los diferentes niveles.
- ✓ Implementar los reportes estadísticos propios del Registro de Fallecidos.
- ✓ Incorporar a los servicios del Registro de Personal de la Salud, la facilidad de obtener el cargo del personal de la salud que se gestiona porque en el proceso actual este dato debe insertarse.
- ✓ Promover una investigación para realizar la impresión y el escaneo del CMD, facilitando procesos que en varios sectores de la sociedad se realizan como en Servicio Comunales, para el enterramiento y en el Registro Civil para trámites de herencia.

REFERENCIAS BIBLIOGRÁFICAS

1. *Consideraciones sobre el Proyecto de Informatización de la Atención Primaria de Salud*. **Díaz, Marín**. 2004, Revista Habanera de Ciencias Médicas.
2. Infomed. *Sitio Web de la Red de Salud de Cuba*. [En línea] Escuela Nacional de Salud Pública, 2004. [Citado el: 13 de noviembre de 2007.] http://www.sld.cu/galerias/doc/sitios/infodir/25_proyeccio_estrategica.doc.
3. **Softel**. *Propuesta de Esquema Sistema Integral de Salud (SISalud)*. 2006.
4. **Tesis**. *Análisis del Registro de Fallecidos para el Sistema de Información para la Salud*. 2007.
5. **Marín**. *Estado actual, pilares y debilidades del proceso de informatización de la Atención Primaria de Salud (APS)*. Ciudad de La Habana : Inédito. Taller de Informática en Salud. Universidad de las Ciencias Informáticas, 2004.
6. **Reynoso, Carlos Billy**. *Introducción a la Arquitectura de Software*. [En línea] UNIVERSIDAD DE BUENOS AIRES, 2005. [Citado el: 15 de Febrero de 2008.] <http://www.willydev.net/descargas/prev/IntroArq.pdf>.
7. **Gassmann, Alejandro**. *Artículos de Programación Web | .NET*. [En línea] [Citado el: 15 de febrero de 2008.] http://www.gamarod.com.ar/articulos/introduccion_a_aspnet.asp.
8. *Lenguajes de Programación (PERL)*. [En línea] [Citado el: 15 de febrero de 2008.] <http://es.wikipedia.org/wiki/Perl>.
9. **Alvarez, Miguel Angel**. *Desarrollo Web*. [En línea] [Citado el: 16 de febrero de 2008.] <http://www.desarrolloweb.com/articulos/25.php>.
10. **Ídem**. Referencia 9.
11. —. Referencia 9.
12. **Jacobson, I. B., Grady y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*.
13. **Ídem**. Referencia 12.
14. **Autores, C**. *UML_Tutorial*. [En línea] 2008. [Citado el: 13 de Marzo de 2008.] <http://www.tripod.lycos.es/>.
15. **Riesco, Daniel y Romero, Daniel**. *Un Workflow que Automatice los*. [En línea] Mayo de 2004. <http://www.dsic.upv.es/~dromero/papers/wicc2004.pdf>.
16. **UCI**. *Flujo de trabajo Análisis & Diseño (Modelo de diseño)*. Ciudad de la Habana : s.n., 2006.
17. **Ídem**. Referencia 3.

18. **Grady, Rumbaugh, Jacobson, I. B., James.** *El Lenguaje Unificado de Modelado. Manual de Referencia.*
19. **Bermón Angarita, L.** LP11. Ingeniería de software. [En línea] 2008. [Citado el: 13 de Marzo de 2008.] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060038/lecciones/modulo%201/capitulo%204/conallen.htm>.
20. **Lage Davila, A., Molina García, J. R., & Bascos Fuentes, E.** La investigación en salud como elemento integrador entre la universidad y los servicios de salud. [En línea] Revista Cubana Educación Médica, Enero de 1995. [Citado el: 27 de Noviembre de 2007.] http://scielo.sld.cu/scielo.php?pid=S0864-21411995000100003&script=sci_arttext.
21. **Rojas Ochoa, F.** La atención primaria en Cuba, 1959-1984. [En línea] Revista Cubana Salud Pública, Abril de 2005. [Citado el: 28 de Noviembre de 2007.] http://scielo.sld.cu/scielo.php?pid=S0864-34662005000200011&script=sci_arttext.
22. **Alonso, Santiago Crespo.** *Análisis comparativo del Certificado Médico de Defunción en los períodos 1989-1992 y 1996-1997 en Sabadell. Interpretación según legislación vigente.* Barcelona : s.n., 2004.
23. Portal de Salud de la Comunidad de Madrid. [En línea] [Citado el: 24 de Enero de 2008.] http://www.madrid.org/cs/Satellite?cid=1142329549247&language=es&pagename=PortalSalud%2FPage%2FP TSA_pintarContenidoFinal&vest=1156827010188.
24. **Martí, Consuelo Ibañez.** Weblogs. [En línea] [Citado el: 24 de Enero de 2008.] http://weblogs.madrimasd.org/salud_publica/archive/2007/08/22/72301.aspx.
25. **Zacca Peña, E y Martínez Morales, M. A.** ¿Cómo llenar un certificado médico de defunción? [En línea] Supercurso. [Citado el: 15 de Mayo de 2008.] <http://supercurso.sld.cu/supercursos/plonearticlemultipage.2006-05-18.7979805252/colectivo-de-autores>.

BIBLIOGRAFÍA

Alonso, Santiago Crespo. *Análisis comparativo del Certificado Médico de Defunción en los períodos 1989-1992 y 1996-1997 en Sabadell. Interpretación según legislación vigente.* Barcelona : s.n., 2004.

Alvarez, Miguel Angel. Desarrollo Web. [En línea] [Citado el: 16 de febrero de 2008.]
<http://www.desarrolloweb.com/articulos/25.php>.

Autores, C. UML_Tutorial. [En línea] 2008. [Citado el: 13 de Marzo de 2008.] <http://www.tripod.lycos.es/>.

Bermón Angarita, L. LP11. Ingeniería de software. [En línea] 2008. [Citado el: 13 de Marzo de 2008.]
<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060038/lecciones/modulo%201/capitulo%204/conallen.htm>.

Consideraciones sobre el Proyecto de Informatización de la Atención Primaria de Salud. **Díaz, Marín.** 2004, Revista Habanera de Ciencias Médicas.

Gassmann, Alejandro. Artículos de Programación Web | .NET. [En línea] [Citado el: 15 de febrero de 2008.]
http://www.gamarod.com.ar/articulos/introduccion_a_aspnet.asp.

Grady, Rumbaugh, Jacobson, I. B., James. *El Lenguaje Unificado de Modelado. Manual de Referencia.*

Infomed. *Sitio Web de la Red de Salud de Cuba.* [En línea] Escuela Nacional de Salud Pública, 2004. [Citado el: 13 de noviembre de 2007.] http://www.sld.cu/galerias/doc/sitios/infodir/25_proyeccio_estrategica.doc.

Jacobson, I. B., Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.*

Lage Davila, A., Molina García, J. R., & Bascos Fuentes, E. La investigación en salud como elemento integrador entre la universidad y los servicios de salud. [En línea] Revista Cubana Educación Médica, Enero de 1995. [Citado el: 27 de Noviembre de 2007.] http://scielo.sld.cu/scielo.php?pid=S0864-21411995000100003&script=sci_arttext.

Lenguajes de Programación (PERL). [En línea] [Citado el: 15 de febrero de 2008.] <http://es.wikipedia.org/wiki/Perl>.

Marín. *Estado actual, pilares y debilidades del proceso de informatización de la Atención Primaria de Salud (APS).* Ciudad de La Habana : Inédito. Taller de Informática en Salud. Universidad de las Ciencias Informáticas, 2004.

Martí, Consuelo Ibañez. Weblogs. [En línea] [Citado el: 24 de Enero de 2008.]
http://weblogs.madrimasd.org/salud_publica/archive/2007/08/22/72301.aspx.

Portal de Salud de la Comunidad de Madrid. [En línea] [Citado el: 24 de Enero de 2008.]

http://www.madrid.org/cs/Satellite?cid=1142329549247&language=es&pagename=PortalSalud%2FPage%2FPTSA_pintarContenidoFinal&vest=1156827010188.

Reynoso, Carlos Billy. Introducción a la Arquitectura de Software. [En línea] UNIVERSIDAD DE BUENOS AIRES, 2005. [Citado el: 15 de Febrero de 2008.] <http://www.willydev.net/descargas/prev/IntroArq.pdf>.

Riesco, Daniel y Romero, Daniel. Un Workflow que Automatice los. [En línea] Mayo de 2004. <http://www.dsic.upv.es/~dromero/papers/wicc2004.pdf>.

Rojas Ochoa, F. La atención primaria en Cuba, 1959-1984. [En línea] Revista Cubana Salud Pública, Abril de 2005. [Citado el: 28 de Noviembre de 2007.] http://scielo.sld.cu/scielo.php?pid=S0864-34662005000200011&script=sci_arttext.

Softel. *Propuesta de Esquema Sistema Integral de Salud (SISalud)*. 2006.

Tesis. *Análisis del Registro de Fallecidos para el Sistema de Información para la Salud*. 2007.

UCI. *Flujo de trabajo Análisis & Diseño (Modelo de diseño)*. Ciudad de la Habana : s.n., 2006.

Zacca Peña, E y Martínez Morales, M. A. ¿Cómo llenar un certificado médico de defunción? [En línea] Supercurso. [Citado el: 15 de Mayo de 2008.] <http://supercurso.sld.cu/supercursos/ploneartidemultipage.2006-05-18.7979805252/colectivo-de-autores>.

ANEXOS

Anexo 1. Modelo 8-110(18-38-2).

MINISTERIO DE SALUD PUBLICA SISTEMA DE INFORMACION ESTADISTICA NACIONAL MOD. 8-110 (18-38-2)		CERTIFICADO MEDICO DE DEFUNCION FETAL Para defunciones de 500 gramos y más o 22 semanas y más de gestación			No	
DEL FETO	1. SEXO: <input type="checkbox"/> 1 Masculino <input type="checkbox"/> 2 Femenino <input type="checkbox"/> 3 Indeterminado	2. PESO AL NACER: (EN GRAMOS)	3. SEMANAS COMPLE- TAS DE GESTACION:	4. TIPO DE EMBARAZO: <input type="checkbox"/> 1 SENCILLO <input type="checkbox"/> 2 MELLIZO <input type="checkbox"/> 3 OTRO <input type="checkbox"/> 9 IGNORADO	5. ¿EN QUE MOMENTO OCURRIO LA DEFUNCION? <input type="checkbox"/> 1 ANTES DEL PARTO <input type="checkbox"/> 2 DURANTE EL PARTO	
	6. LUGAR DEL PARTO: <input type="checkbox"/> 1 HOSPITAL <input type="checkbox"/> 2 OTRO CENTRO.MED <input type="checkbox"/> 3 DOMICILIO <input type="checkbox"/> 4 OTRO LUGAR <input type="checkbox"/> 9 IGNORADO	7. HORA Y FECHA DE NACIMIENTO: _____ <input type="checkbox"/> A.M <input type="checkbox"/> P.M DIA MES AÑO				
	LOCALIDAD:	MUNICIPIO:	PROVINCIA:	CODIGO:		
DE LA MADRE	8. NOMBRE (S) Y APELLIDOS:			9. No. DE IDENTIDAD O FECHA DE NAC:		
	10. EDAD: _____ <input type="checkbox"/> 4 Años	11. VIDA CONYUGAL: <input type="checkbox"/> 1 Sin Pareja Estable <input type="checkbox"/> 2 Con Pareja Estable <input type="checkbox"/> 9 Ignorado	12. SI ES EXTRANERO LA RESIDENCIA ES: <input type="checkbox"/> 1 Permanente <input type="checkbox"/> 2 Temporal <input type="checkbox"/> 9 Ignorado			
	13. ESCOLARIDAD: <input type="checkbox"/> 1 Nigua <input type="checkbox"/> 2 Primaria Incompleta <input type="checkbox"/> 3 Primaria Terminada <input type="checkbox"/> 4 Secundaria Terminada <input type="checkbox"/> 5 Pre-Univ. o Tec. Terminado <input type="checkbox"/> 6 Universit. Terminado <input type="checkbox"/> 9 Ignorado	14. OCUPACION:		CODIGO:		
	15. SI PARTO ANTERIOR, FECHA: DIA MES AÑO	16. NUMERO DE EMBARAZOS (INCLUYE EL DE ESTE CERTIFICADO): ABORTOS MUERTOS NACIDOS MUERTOS NACIDOS VIVOS TOTAL (DE ELLOS) { ACTUALMENTE VIVOS ACTUALMENTE MUERTOS } TOTAL DE EMBARAZOS				
	17. RESIDENCIA HABITUAL. (Calle, carretera, finca, etc.): ENTRE CALLES, BLOQUE, ZONA, REPARTO:		NUMERO O Km:	EDIFICIO:	APTO:	
MUNICIPIO:	PROVINCIA:	AREA DE SALUD:	CODIGO:			
SOBRE LA CERTIFICACION DE LAS CAUSAS	18. CAUSAS DE MUERTE:				TIEMPO aprox. Entre el inicio de la causa y la muerte.	CODIGO
	A) PRINCIPAL ENFERMEDAD O CONDICION DEL FETO					
	B) OTRAS ENFERMEDADES O CONDICIONES DEL FETO					
	C) PRINCIPAL ENFERMEDAD DE LA MADRE QUE AFECTO AL FETO					
D) CIRCUNSTANCIAS RELEVANTES (ACLARE SI ES DE LA MADRE O DEL FETO, CUANDO SEA NECESARIO)						
PARA MEDICOS VOLUNTARIOS	19. CONFIRMACION DE LAS CAUSAS: <input type="checkbox"/> 1 Clínica <input type="checkbox"/> 5 Necropsia <input type="checkbox"/> 6 Reconocimiento		20. CERTIFICACION HECHA POR: <input type="checkbox"/> 1 NEONATOLOGO <input type="checkbox"/> 2 PEDIATRA <input type="checkbox"/> 3 OBSTETRA <input type="checkbox"/> 4 LEGISTA <input type="checkbox"/> 5 OTRO			
	21. A) PARECE DEBERSE A: <input type="checkbox"/> Accidente <input type="checkbox"/> Homicidio <input type="checkbox"/> Otro <input type="checkbox"/> Por investigar		B) FECHA DE LA LESION: DIA MES AÑO		C) LUGAR DONDE OCURRIO LA LESION:	
	D) DESCRIPCION DE COMO SE PRODUJO LA LESION:					
	E) AUTORIDAD QUE SOLICITA LA ACTUACION:				No. DE ACTA:	
	NOMBRE (S) APELLIDOS DEL MEDICO:		No. REG. PROFESIONAL:		FIRMA:	
NOMBRE (S) APELLIDOS DEL MEDICO:		No. REG. PROFESIONAL:		FIRMA:		
PARA USO DEL DEPARTAMENTO DE ESTADISTICA						
CUÑO DE LA UNIDAD:	NOMBRE DE LA UNIDAD:		FECHA: DIA MES AÑO		CODIGO:	
	NOMBRE (S) Y APELLIDOS DEL FUNCIONARIO ACTUANTE:		CARGO:		No. REG. UNIDAD:	
	MUNICIPIO DE ENTE RRAMIENTO:		FIRMA:		No. PROVINCIA:	
ORIGINAL DESTINADO AL MINSAP PARA FINES ESTADISTICOS						

Anexo 2. Modelo 8-110(18-38).

MINISTERIO DE SALUD PUBLICA SISTEMA DE INFORMACION ESTADISTICA NACIONAL MOD. 8-110 (18-38)		CERTIFICADO MEDICO DE DEFUNCION NEONATAL (Para fallecidos de 0-27 días)				No
DEL FALLECIDO DE 0-27 DIAS	1. 1er APELLIDO:	2do APELLIDO:	NOMBRE (S):		2. No DE IDENTIDAD:	
	3. SEXO: <input type="checkbox"/> 1 Masculino <input type="checkbox"/> 2 Femenino	4. PESO AL NACER (EN GRAMOS)	5. SEMANAS COMPLETAS DE GESTACION:	6. TIPO DE EMBARAZO: <input type="checkbox"/> 1 SENCILLO <input type="checkbox"/> 2 MELLI-OTRO <input type="checkbox"/> 3 OTRO <input type="checkbox"/> 9 IGNORADO	7. LUGAR DEL PARTO: <input type="checkbox"/> 1 HOSPITAL <input type="checkbox"/> 2 OTRO CENTRO MEDICO <input type="checkbox"/> 3 DOMICILIO <input type="checkbox"/> 4 OTRO LUGAR <input type="checkbox"/> 9 IGNORADO	
	8. HORA Y FECHA DE NACIMIENTO: _____ <input type="checkbox"/> A.M. <input type="checkbox"/> P.M. DIA MES AÑO		9. HORA Y FECHA DE DEFUNCION: _____ <input type="checkbox"/> A.M. <input type="checkbox"/> P.M. DIA MES AÑO		10. EDAD: <input type="checkbox"/> 1 MENOS DE 1 HORA <input type="checkbox"/> 2 1-11 HORAS <input type="checkbox"/> 3 12-23 HORAS <input type="checkbox"/> 4 Días	
	11. SITIO DE LA DEFUNCION: <input type="checkbox"/> 1 C. Guard. Hospital <input type="checkbox"/> 2 Ingresado Hospital <input type="checkbox"/> 3 Otro Hospital <input type="checkbox"/> 4 Centro Medico <input type="checkbox"/> 5 Domicilio <input type="checkbox"/> 9 Otro Lugar		LOCALIDAD:	MUNICIPIO:	PROVINCIA:	CODIGO:
12. NOMBRE (S) Y APELLIDOS:					13. No DE IDENTIDAD O FECHA DE NAC:	
14. EDAD: _____ Años <input type="checkbox"/> 4		15. VIDA CONYUGAL: <input type="checkbox"/> 1 Sin Pareja Estable <input type="checkbox"/> 2 Con Pareja Estable <input type="checkbox"/> 9 Ignorado		16. SI ES EXTRANERO LA RESIDENCIA ES: <input type="checkbox"/> 1 Permanente <input type="checkbox"/> 2 Temporal <input type="checkbox"/> 9 Ignorado		
17. ESCOLARIDAD: <input type="checkbox"/> 1 Niguna <input type="checkbox"/> 2 Primaria Incompleta <input type="checkbox"/> 3 Primaria Terminada <input type="checkbox"/> 4 Secundaria Terminada <input type="checkbox"/> 5 Pre-Univ. o Tec. Terminado <input type="checkbox"/> 6 Universit. Terminado <input type="checkbox"/> 9 Ignorado			18. OCUPACION:		CODIGO:	
19. SI PARTO ANTERIOR, FECHA: DIA MES AÑO		20. NUMERO DE EMBARAZOS (INCLUYE EL DE ESTE CERTIFICADO): ABORTOS MUERTOS NACIDOS MUERTOS NACIDOS VIVOS (DE ELLOS) TOTAL ACTUALMENTE VIVOS ACTUALMENTE MUERTOS } TOTAL DE EMBARAZOS				
21. RESIDENCIA HABITUAL. (Calle, carretera, finca, etc.): ENTRE CALLES, BLOQUE, ZONA, REPARTO:			NUMERO O Km:	EDIFICIO:	APTO:	
MUNICIPIO:		PROVINCIA:	AREA DE SALUD:	LOCALIDAD:		
22. CAUSAS DE MUERTE:					TIEMPO aprox. Entre el inicio de la causa y la muerte.	
A) PRINCIPAL ENFERMEDAD O CONDICION DEL RECIEN NACIDO					CODIGO	
B) OTRAS ENFERMEDADES O CONDICIONES DEL RECIEN NACIDO						
C) PRINCIPAL ENFERMEDAD DE LA MADRE QUE AFECTO AL RECIEN NACIDO						
D) CIRCUNSTANCIAS RELEVANTES (ACLARE SI ES DE LA MADRE O DEL RECIEN NACIDO, CUANDO SEA NECESARIO)						
23. CONFIRMACION DE LAS CAUSAS: <input type="checkbox"/> 1 Clínica <input type="checkbox"/> 2 Investigación (excluye biopsia) <input type="checkbox"/> 3 Operación <input type="checkbox"/> 4 Biopsia <input type="checkbox"/> 5 Necropsia <input type="checkbox"/> 6 Reconocimiento			24. CERTIFICACION HECHA POR: <input type="checkbox"/> 1 NEONATOLOGO <input type="checkbox"/> 2 PEDIATRA <input type="checkbox"/> 3 OBSTETRA <input type="checkbox"/> 4 LEGISTA <input type="checkbox"/> 5 OTRO			
25. A) PARECE DEBERSE A: <input type="checkbox"/> Accidente <input type="checkbox"/> Homicidio <input type="checkbox"/> Otro <input type="checkbox"/> Por investigar		B) FECHA DE LA LESION: DIA MES AÑO		C) LUGAR DONDE OCURRIO LA LESION:		
D) DESCRIPCION DE COMO SE PRODUJO LA LESION:						
E) AUTORIDAD QUE SOLICITA LA ACTUACION:					No DE ACTA:	
NOMBRE (S) APELLIDOS DEL MEDICO:			No. REG. PROFESIONAL:	FIRMA:		
NOMBRE (S) APELLIDOS DEL MEDICO:			No. REG. PROFESIONAL:	FIRMA:		
PARA USO DEL DEPARTAMENTO DE ESTADISTICA						
CUÑO DE LA UNIDAD:		NOMBRE DE LA UNIDAD:		FECHA: DIA MES AÑO	CODIGO:	
		NOMBRE (S) Y APELLIDOS DEL FUNCIONARIO ACTUANTE:		CARGO:	No. REG. UNIDAD:	
		MUNICIPIO DE ENTE RRAMIENTO:		FIRMA:	No. PROVINCIA:	
ORIGINAL DESTINADO AL MINSAP PARA FINES ESTADISTICOS						

Anexo 3. Modelo 8-111(18-39-2).

MINISTERIO DE SALUD PUBLICA SISTEMA DE INFORMACION ESTADISTICA NACIONAL MOD. 8-111 (18-39-2)		CERTIFICADO MEDICO DE DEFUNCION (Para fallecidos de 28 días y más)			No
1. 1er APELLIDO:		2do APELLIDO:		NOMBRE (S):	
2. No DE IDENTIDAD O FECHA DE NAC:		3. EDAD: a) Si es -1 año: <input type="checkbox"/> 4 <input type="checkbox"/> 5 Días Meses		b) Si es de un 1 año o más: <input type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/> 100 Años 100 Años y más	
4. SEXO: <input type="checkbox"/> 1 Masculino <input type="checkbox"/> 2 Femenino		5. COLOR DE LA PIEL: <input type="checkbox"/> 1 Blanca <input type="checkbox"/> 2 Negra <input type="checkbox"/> 3 Mestiza		6. VIDA CONYUGAL: <input type="checkbox"/> 1 Sin Pareja <input type="checkbox"/> 2 Con Pareja <input type="checkbox"/> 9 Ignorado Estable Estable	
7. OCUPACION:		CODIGO:			
8. SI ES EXTRANERO LA RESIDENCIA ES: <input type="checkbox"/> 1 Permanente <input type="checkbox"/> 2 Temporal <input type="checkbox"/> 9 Ignorado		9. RESIDENCIA HABITUAL. (Calle, carretera, finca, etc.):			
EDIFICIO:		APTO:		ENTRE CALLES, BLOQUE, ZONA, REPARTO:	
LOCALIDAD:		MUNICIPIO:			
PROVINCIA:		AREA DE SALUD:		CODIGO:	
10. HORA Y FECHA DE DEFUNCION: ____ <input type="checkbox"/> A.M ____ <input type="checkbox"/> P.M DIA MES AÑO		11. SITIO DE LA DEFUNCION: <input type="checkbox"/> 1 Cuerpo de Guardia Hospital <input type="checkbox"/> 2 Ingresado Hospita <input type="checkbox"/> 3 Otro Centro Médico <input type="checkbox"/> 4 Domicilio <input type="checkbox"/> 5 Otro Lugar <input type="checkbox"/> 6 Extranjero <input type="checkbox"/> 9 Ignorado			
LOCALIDAD:		MUNICIPIO:		PROVINCIA:	
CODIGO:					
12. Sólo para mujeres de entre 12 a 49 años de edad: ¿En los 12 meses antes de la defunción tuvo un embarazo? Si <input type="checkbox"/> No <input type="checkbox"/> En caso de "Si" responde:		El embarazo terminó en: <input type="checkbox"/> Aborto <input type="checkbox"/> Parto (Incluye Cesárea)		Fecha del Evento: DIA MES AÑO	
13. CAUSAS DE MUERTE:		PARTE I Enfermedades o estado patológico que finalmente produjo la muerte. Causas que antecedieron a la causa directa si existen. La última causa anotada en b, c ó d será la que inició todo el proceso patológico.		Tiempo aprox. Entre el inicio de la causa y la muerte.	
PARTE II Otras enfermedades significativas que contribuyeron a la muerte, pero no relacionadas con la causa directa (1 a).		A) CAUSA DIRECTA		CODIGO	
		B) CAUSA QUE OCASIONO LO ANOTADO EN PARTE I a)			
		C) CAUSA QUE OCASIONO LO ANOTADO EN PARTE I b)			
		D) CAUSA QUE OCASIONO LO ANOTADO EN PARTE I c)			
14. CONFIRMACION DE LAS CAUSAS: Clínica <input type="checkbox"/> 1 Investigación (excluye biopsia) <input type="checkbox"/> 2 Operación <input type="checkbox"/> 3		Biopsia <input type="checkbox"/> 4 Necropsia <input type="checkbox"/> 5 Reconocimiento <input type="checkbox"/> 6		15. CERTIFICACION HECHA POR EL MEDICO (de): <input type="checkbox"/> 1 Guardia <input type="checkbox"/> 2 Acistencial <input type="checkbox"/> 3 Familia <input type="checkbox"/> 4 Legista <input type="checkbox"/> 5 Otro	
16. A) SI ES MUERTE VIOLENTA PARECE DEBERSE A: <input type="checkbox"/> Accidente <input type="checkbox"/> Suicidio <input type="checkbox"/> Homicidio <input type="checkbox"/> Otro <input type="checkbox"/> Por investigar		B) FECHA DE LA LESION: DIA MES AÑO		C) LUGAR DONDE OCURRIO LA LESION:	
D) DESCRIPCION DE COMO SE PRODUJO LA LESION:					
E) AUTORIDAD QUE SOLICITA LA ACTUACION:		No DE ACTA:			
NOMBRE (S) APELLIDOS DEL MEDICO:		No. REG. PROFESIONAL:		FIRMA:	
NOMBRE (S) APELLIDOS DEL MEDICO:		No. REG. PROFESIONAL:		FIRMA:	
PARA USO DEL DEPARTAMENTO DE ESTADISTICA					
CUÑO DE LA UNIDAD:		NOMBRE DE LA UNIDAD:		FECHA: DIA MES AÑO	
		NOMBRE (S) Y APELLIDOS DEL FUNCIONARIO ACTUANTE:		CARGO:	
		MUNICIPIO DE ENTE RRAMIENTO:		FIRMA:	
				No. REG. UNIDAD:	
				No. PROVINCIA:	
ORIGINAL DESTINADO AL MINSAP PARA FINES ESTADISTICOS					

Anexo 4. Certificado Médico de Defunción (España).



CONSEJO GENERAL DE COLEGIOS OFICIALES DE MEDICOS DE ESPAÑA

464 Ptas.
Derechos autorizados.
I.V.A. incluido

Clase 3.^a
Serie B
Nº 5841751

CERTIFICADO MEDICO DE DEFUNCION

Colegio de _____

C.I.F. Q-288-0010-11



D. _____
en Medicina y Cirugía, colegiado en _____, con
el número _____ y con el ejercicio profesional en _____

CERTIFICO la defunción de _____

que ocurrió a las _____ horas del día _____ de _____
de _____ en la _____ de _____
nº. _____, cuarto _____, población de _____

Murió a consecuencia de _____
(Causa inmediata del fallecimiento)

y son manifiestas en el finado las señales de descomposición _____
(Causa fundamental)

El finado tenía la edad de _____ años
de estado _____ Era natural de _____
hijo de _____ y de _____, cuya identidad (1) _____
(Ciudad y Provincia)

Observación especial (2) _____

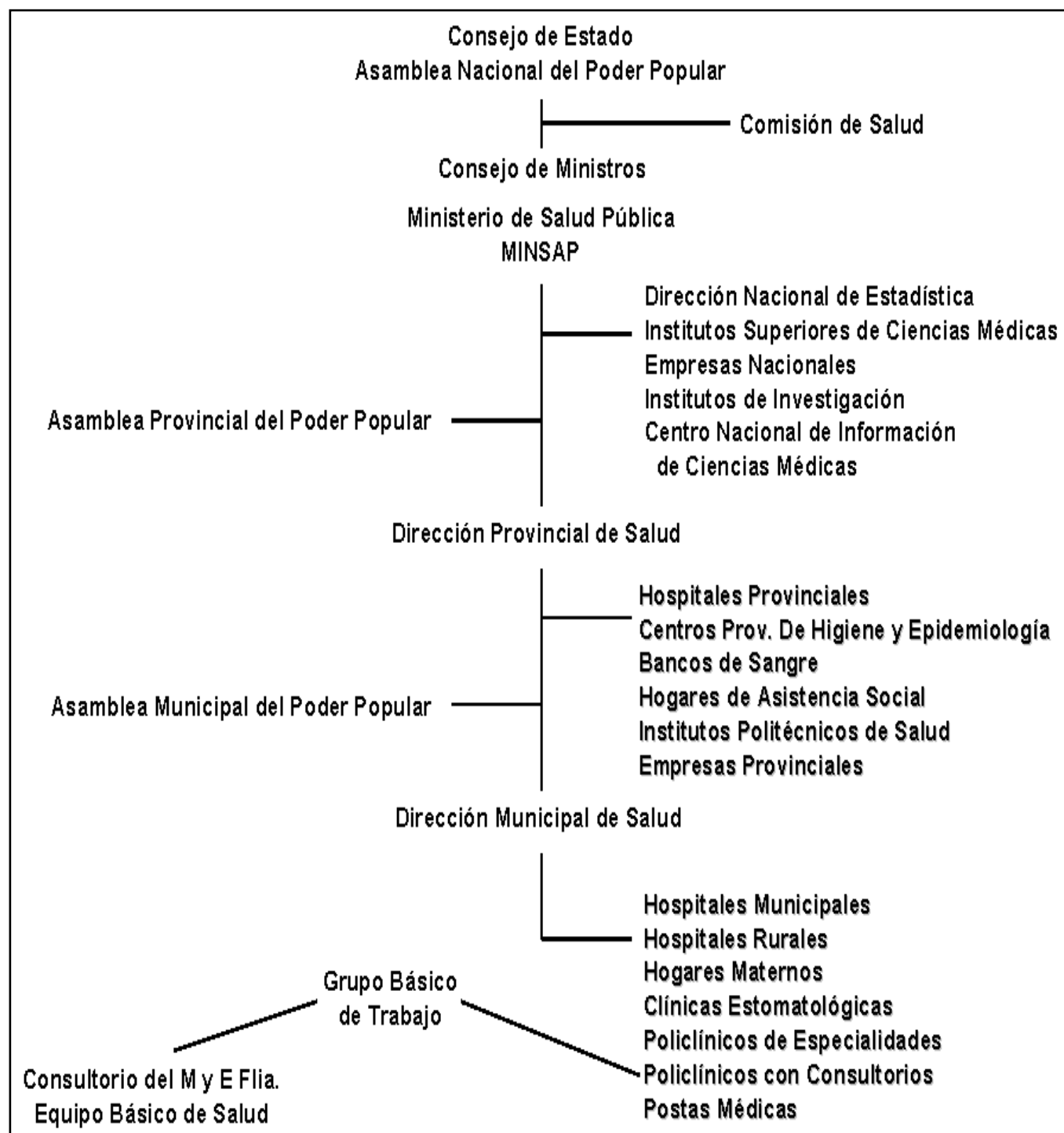
En _____, a _____ de _____
de dos mil _____

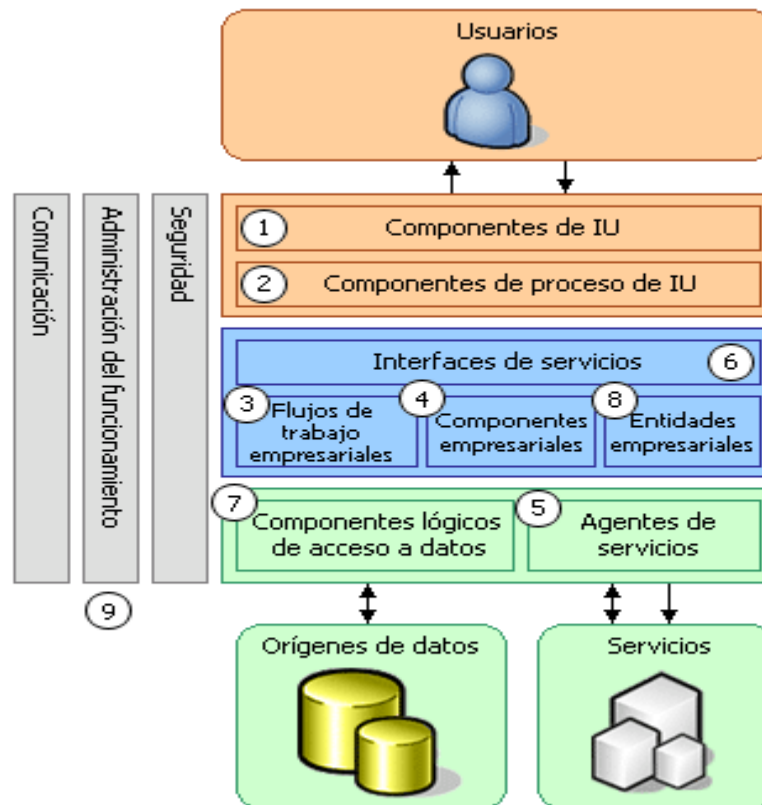
(1) «Se me acreditó con documento de identidad _____» o «se me asegura por don _____
domiciliado en _____, cuya identidad me acredita y firma
este parte», o «Conozco de experiencia propia».

(2) Si hubiera indicios de muerte violenta se comunicará urgente y especialmente al Encargado del Registro Civil.

NOTAS.- Ningún Parte de Defunción podrá ser expedido por el Médico si no va extendido en este impreso, editado por el Consejo General de los Colegios Oficiales de Médicos, debiendo llevar estampado el sello oficial del Colegio Médico Provincial. Los derechos autorizados por la Dirección General de Sanidad son independientes de los Timbres que exigen las disposiciones vigentes.
Corresponde al modelo 8 aprobado por la Orden del Ministerio de Justicia de 24 de diciembre de 1958.

Anexo 5. Organigrama del Sistema Nacional de Salud Cubano.



Anexo 5. Arquitectura en tres capas.

GLOSARIO DE TÉRMINOS

Aplicación (Sistema): Sistema que ofrece a un usuario final un conjunto coherente de casos de uso.

Aplicación Web: Es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de un navegador o browser. Estas son muy populares debido a la habilidad para actualizar y mantener la información manipulada sin distribuir e instalar el software en miles de potenciales clientes.

Arquitectura: Conjunto de decisiones significativas acerca de la organización de un sistema de software, la selección de los elementos estructurales a partir de los cuales se componen el sistema. La misma se interesa no solo por la estructura y el comportamiento, sino también por las restricciones y compromisos de uso, funcionalidad, funcionamiento, flexibilidad al cambio, reutilización, comprensión, economía y tecnología, así como por aspectos estéticos.

Asentada: inscrita, anotada, registrada, apuntada.

Causas de defunción: Todas aquellas enfermedades, estados morbosos o lesiones que causaron la muerte o que contribuyeron a ella, y las circunstancias del accidente o de la violencia que produjeron dichas lesiones.

Codificador: Servicios web creados para estandarizar la información común recogida y procesada, contienen la nomenclatura de una ciencia o un proceso determinado.

Defunción fetal: La muerte de un producto de la concepción, antes de su expulsión o su extracción completa del cuerpo de su madre, independientemente de la duración del embarazo; la muerte está indicada por el hecho de que después de la separación, el feto no respira ni da ninguna otra señal de vida, como latidos del corazón, pulsaciones del cordón umbilical o movimientos efectivos de los músculos de contracción voluntaria.

Defunción: Muerte (Del lat. mors, mortis). Cesación o término de la vida. La desaparición total y permanente de todo signo de vida en un momento cualquiera posterior al nacimiento, sin posibilidad de resurrección.

Neonatal: Referido al recién nacido antes de cumplir 28 días de edad.

PlaSer: Constituye una plataforma sobre la que se pueden desplegar aplicaciones “XML_WebServices“. Este sistema está concebido completamente sobre Arquitectura Basada en Componentes y Orientada a Servicios, usando el paradigma de “XML_WebServices“, específicamente “SOAP“. En su concepción se han utilizado estándares actuales y normas abiertas además de un enriquecido lenguaje de patrones entre los que podemos encontrar MVC, Fachada, Alta Cohesión, Bajo Acoplamiento, SAAA, SSO, Proxy, etc.

Registro de Áreas de Salud (RAS): Es el registro que gestiona la información de las Áreas de Salud en Cuba, su estructura organizativa formada por Grupos Básicos de Trabajo y Equipos Básicos de Salud, así como las Poblaciones que estos atienden.

Registro de Ciudadanos (RC): Es el registro que gestiona los datos generales de los ciudadanos del país.

Registro de Enfermedades de Declaración Obligatoria (REDO): Es el registro que gestiona aquellas enfermedades que por su importancia y repercusión se consideran como de constante vigilancia epidemiológica por tratados internacionales o intereses del país.

Registro de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud (RCIE): Es el registro que gestiona la estructura de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud y permite realizar búsquedas dinámicas de acuerdo a criterios seleccionados por el usuario. Ofrece un lenguaje unificado y estándar para la información de la morbilidad del país.

Registro de Personal de Salud (RPS): Es el registro que gestiona los datos del personal de salud del país.

Registro de Población (RPOB): Es el registro que gestiona la información de la Historia de Salud.

Registro de Problemas de Salud de la Atención Primaria (RPSAP): Es el registro que gestiona una clasificación especial para los problemas de salud que se presentan en el nivel de Atención Primaria tratando que resulte una clasificación más breve que la CIE-10, de manejo más simple dada la naturaleza

de tales problemas. Permite realizar búsquedas dinámicas de acuerdo a criterios seleccionados por el usuario. Ofrece un lenguaje unificado y estándar para la información de la morbilidad en la APS.

Reparo: Enmienda que se realiza a algún campo del Certificado Médico de Defunción, en caso que fuese necesario.

SAAA: Se encarga de los permisos dentro del sistema. Está basado en el modelo de Autenticación, Autorización y Auditoria (AAA).

SISalud: Plataforma única para la gestión, procesamiento y transmisión de la información clínica en el Sistema Nacional de Salud.

Sitio de muerte: Define el lugar en el cual ocurre un fallecimiento: Cuerpo de guardia de hospital, Ingresado en hospitales, Otro centro médico, Domicilio, Otro lugar, Extranjero, Ignorado.

Softel: Entidad del Ministerio de Informática y las Comunicaciones (MIC).

Solicitud de Reparo: Documento creado por el técnico en estadísticas con las posibles enmiendas que pueda tener un Certificado Médico de Defunción, que va transitando por todos los niveles correspondientes hasta llegar al médico certificador.

Subordinación: Dependencia de una unidad de salud respecto al nivel de dirección del SNS, por las que están regidas o sometidas.

W3C: El World Wide Web Consortium, abreviado W3C, es una organización que produce estándares para la World Wide Web (o Telaraña Mundial). Está dirigida por Tim Berners-Lee, el creador original de URL (Uniform Resource Locator, Localizador Uniforme de Recursos), HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto) y HTML (Lenguaje de Marcado de HiperTexto) sobre el que se basa la Web, y por extensión su inventor.

WSDL: WSDL son las siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios Web. La versión 1.1 está en estado de "propuesta de recomendación" por parte del W3C. WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de

comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligán después al protocolo concreto de red y al formato del mensaje.

WYSIWYG: es el acrónimo de What You See Is What You Get (en inglés, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso. Se dice en contraposición a otros procesadores de texto, hoy en día poco frecuentes, en los que se escribía sobre una vista que no mostraba el formato del texto, hasta la impresión del documento. En el caso de editores de HTML este concepto se aplica a los que permiten escribir la página sobre una vista preliminar similar a la de un procesador de textos, ocupándose en este caso el programa de generar el código fuente en HTML.

XHTML: Es el acrónimo en inglés de Extensible HyperText Markup Language. Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo y JavaScript su aspecto y diseño en distintos medios (ordenadores, PDAs, teléfonos móviles, impresoras).

XML: XML es la sigla del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C). Se ha convertido en un formato estándar en Internet y está diseñado para representar datos estructurados, es más que un lenguaje de marcado como su nombre lo indica; es un metalenguaje para definir otros lenguajes de marcados adecuados a un uso en específico, éste es la base de los servicios web. XML, al que algunos consideran el esperanto de los sistemas de información, se emplea principalmente para separar el contenido de la presentación de forma total, o sea, permite representar datos de forma homogénea en entornos heterogéneos, lo que facilita la interoperabilidad entre distintos sistemas.

XSL: Es el acrónimo en inglés de Extensible Stylesheet Language. Ha surgido como una solución para transformar documentos XML a otro formato como: HTML, WML, texto simple, PDF e inclusive en otro documento XML con diferentes parámetros. El lenguaje XSL está dividido en tres partes: XSLT (Extensible Stylesheet Lenguaje Transformation), el cual define la transformación de documentos en XML, XPath (XML Path Language), donde se especifica como acceder cierto punto de la estructura del documento XML y XSLF (Extensible Stylesheet Lenguaje Format), que precisa el formato que deben tomar los objetos dentro del documento en XML. Generalmente, se necesita un procesador de hojas de estilo, o stylesheet processor para procesarlas, aplicándolas a un fichero XML.