Universidad de las Ciencias Informáticas
Facultad 7

Título: Registro de Ubicación Geográfica y Registro de Localidades para el Sistema de Información para la Salud.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autores: Yosvani Turruelles Tejeda
Maikel Gonzalez Diaz

Tutores: Lic. Caridad Guzmán Vitón.
Ing. Luis A. Cobo Espinosa.

Asesor: Ing. David Barreto Medina.

La Habana, Julio 2007

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 6 días del mes de julio del año 2007.

_____	_____
Yosvani Turruelles Tejeda	Maikel González Díaz
Firma del Autor	Firma del Autor
_____	_____
Lic. Caridad Guzmán Vitón	Ing. Luis A. Cobo Espinosa
Firma del Tutor	Firma del Tutor

DATOS DE CONTACTO

Caridad Guzmán Vitón (Tutora): Especialista A en Sistemas Organizativos e Informativos de la empresa Softel. Graduada en Cibernética Matemática en el año 1986 en la Universidad de La Habana. Posee 19 años de experiencia en el desarrollo de software desempeñando diferentes roles. Ha trabajado en las empresas Textilera Rubén Martínez Villena (1987 – 1993) y Softel (1993 hasta la actualidad). Trabajó en la empresa mixta BC BIOCON Internacional S.A. España (1996-2001). Ha pasado varios cursos de superación y postgrado. Ha sido tutora de tesis. Su dirección de correo electrónico es cary@softel.cu.

Luis Abel Cobo Espinosa (Tutor): Ingeniero Informático, graduado en el ISPJAE en 1997. Jefe de Proyecto (desarrollo) de Softel. Actualmente curso la maestría Gestión de Proyectos Informáticos en la UCI. Participé en el Fórum Provincial del 2005. Su dirección de correo electrónico es luiscobo@softel.cu

Ing. David Barreto Medina (Asesor): Profesor graduado de Ing. Industrial en el año 2004. Ha impartido asignaturas como Administración de Empresa, Contabilidad y Comercio Electrónico. Se desempeña como líder del proyecto APS y actualmente esta cursando el postgrado de Costo de Proyectos. Su dirección de correo electrónico es dbarreto@uci.cu.

AGRADECIMIENTOS

A Fidel y la Revolución, por darnos la oportunidad de estudiar en esta Universidad…

A nuestras familias, por siempre confiar en nosotros...

A Lili y Lianet…

A Cary, Cobo y a Pura, por todo el tiempo dedicado…

A todos los especialistas de la empresa Softel y los profes: Cary, Mirna, Cobo, Alfredo, Denis, El Chino, Lucy, Daniel, Yamilka, Yosvanotti, David…

A todos los integrantes del Proyecto APS, en especial al team de 5to año, Ayme y Mairenys…

A todos los que han sido participes de la realización de este trabajo de diploma…

DEDICATORIA

A mis padres, Bárbaro y Maritza por ser esa clase de personas que todo lo comprenden y dan lo mejor de sí mismos sin esperar nada a cambio…

A mi hermano Alejandro, que siempre ha seguido mis pasos…

A Lianet, por perdonarme tantas horas ausente, no hubo alternativa, pero ya se acabó!!!

A mis amigos de toda la vida (no se fijen en el orden): Karel, Dayner, Ruber, Fernando y los nuevos, ellos saben quienes son…

A mi familia, por ayudarme siempre, en todo momento…

Al proyecto APS…

A todos, les dedico este trabajo…

Yosvani Turruelles Tejeda

Dedico este trabajo especialmente a mi madre, porque siempre confió ciegamente en mí y gracias a la educación me inculcó y a su amor incondicional, hoy logré ser lo que soy.

A mi tía Miriam, por darme su ejemplo y por guiarme en este difícil camino, de la vida profesional.

A mi padre por confiar en mí y a mi familia en general, porque me apoyaron y me aconsejaron siempre.

A aquellos que no son mi sangre pero que los quiero como si lo fueran, por su dedicación y la confianza depositada en mí.

A mi novia y a su familia, los cuales son tan especiales en mi vida por aceptarme entre ellos sin recibir nada a cambio y darme siempre el apoyo que necesité para tomar las decisiones correctas.

A mis amigos de antes y a los que he conocido en esta maravillosa carrera, que me han aceptado como soy y me han brindado su amistad sincera e incondicional.

A todos, va dedicado este trabajo.

Maikel González Díaz

RESUMEN

Actualmente se desarrollan componentes que forman parte del Sistema de Información para la Salud (SISalud). En este proceso, surge la necesidad de contar con información no perteneciente al dominio de Salud, la ubicación geográfica y localidades. Se decide desarrollar el Registro de Ubicación Geográfica, que gestiona la información de las Provincias, Municipios, Localidades, Calles y Manzanas del país; y el Registro de Localidades que gestiona la información de los Consejos Populares, Circunscripciones, Zonas y CDR.

El presente trabajo, persigue como objetivo desarrollar una aplicación web que automatice la gestión de la información de la ubicación geográfica y localidades para el Sistema de Información para la Salud (SISalud).

Para cumplir el objetivo se utilizan tecnologías actuales como: PHP, XSL, XML, MySQL, y el apache. Además de algunas herramientas como fueron el Zend Development Environment, el MySQL-Front, Rational Rose. Para la documentación de dicho trabajo se utilizó la metodología de desarrollo RUP, con el lenguaje de modelado UML. Estas son utilizadas por todas las facilidades que brindan y porque responden a las políticas trazadas por el MINSAP de utilizar sistemas abiertos, tecnologías basadas en INTERNET, entre otras.

Con la creación de los Registros se espera un eficiente manejo de los datos que estos gestionaran, así como poder contar con información actualizada, precisa y centralizada acerca de ubicaciones geográficas y localidades. Otro de los resultados esperados, es que como se presentan el análisis, diseño e implementación, y las recomendaciones, se permitirán futuras mejoras a los sistemas ya desarrollados.

PALABRAS CLAVE

Informatización, componente, ubicación geográfica, localidades.

Para ser exitoso

no tienes que hacer cosas extraordinarias.

Haz cosas ordinarias, extraordinariamente bien.

Jim Rohn

TABLA DE CONTENIDOS

Introducción

“La salud es una de las condiciones más importantes de la vida humana y un componente fundamental de las posibilidades humanas que tenemos motivos para valorar. Ninguna concepción de la justicia social que acepte la necesidad de una distribución equitativa y de una formación eficiente de las posibilidades humanas puede ignorar el papel de la salud en la vida humana y en las oportunidades de las personas para alcanzar una vida sana, sin enfermedades y sufrimientos evitables ni mortalidad prematura. La equidad en la realización y distribución de la salud queda así incorporada y formando parte integral de un concepto más amplio de la justicia”. [1]

La salud en Cuba antes del triunfo de la revolución no era un servicio para toda la sociedad, a ella sólo tenían acceso los que podían pagar por recibirlo. La atención médica especializada se centraba sólo en la capital y el precio de los medicamentos estaba muchas veces por encima de su costo. Después del triunfo de la Revolución estas limitaciones se erradicaron por completo ya que el gobierno llevó a cabo grandes transformaciones en esta esfera, por ejemplo los servicios médicos pasaron a ser absolutamente gratuitos, se rebajó considerablemente el precio en casi la totalidad de los medicamentos, se creó el servicio médico rural, además de la construcción de hospitales, policlínicos y postas médicas.

El Sistema Nacional de Salud (SNS) en Cuba ha sido en los últimos años un proyecto social en sí mismo para garantizar la equidad, accesibilidad y universalidad que requieren todos los ciudadanos. Además de presentar una Estructura Jerárquica Administrativo - Territorial muy bien definida.

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 de nuestro pueblo. [2]

El mismo está compuesto por tres niveles básicos de atención. La Atención Primaria de Salud (APS) el cual representa el primer nivel de contacto del SNS con los individuos, la familia y la comunidad. La Atención Secundaria se brinda a nivel de las instituciones hospitalarias, por lo general son de carácter provincial, o sea atienden a toda la población de una provincia determinada. La Atención Terciaria es

aquella que por su condición muy especializada, sólo se brinda en determinados centros de especialidades ambulatorias.

El Ministerio de Salud Pública (MINSAP), es el organismo de la administración central del estado cubano, encargado de ejecutar las políticas y trazar las estrategias metodológicas, reguladoras y de control, necesarias para obtener los resultados propuestos. [3]

En el esfuerzo realizado por la Revolución en busca del bienestar absoluto de la sociedad, el país ha asignado diferentes recursos tanto humanos como económicos, los cuales incluyen personal altamente calificado procedente casi en su totalidad de las diversas instituciones educativas de la isla, formando el capital humano necesario para asegurar el desarrollo de la humanidad. En todo este desarrollo es de vital importancia la presencia de las tecnologías informáticas las cuales tienen el rol más importante en el futuro desarrollo de la informatización en las diferentes esferas de la sociedad, un ejemplo de esto es la informatización de la Salud Cubana.

La informatización del Sistema Nacional de Salud Pública (SNS) está dada por el conjunto de métodos, técnicas, procederes y actividades gerenciales dirigidas al manejo de la información en salud, la cual comprende la información sobre el estado de salud de la población, la información sobre el conocimiento de las ciencias de la salud y la información en general para la toma de decisiones, clínico-epidemiológicas, operativas y estratégicas. [4]

En el año 2003 el Ministerio de Salud Publica como organismo rector del Sistema Nacional de Salud define como una prioridad su informatización, para esto el MINSAP en conjunto con la alta dirección del país ha trazado políticas y estrategias para lograr incorporar de forma ambiciosa las Tecnologías de la Informática y las Comunicaciones (TIC) en la Salud Cubana, todo esto persiguiendo un único objetivo, la búsqueda de nuevas formas que brinden una atención con mejor calidad al pueblo, incrementando la eficiencia y calidad en los servicios.

Para esta gran y compleja tarea, en algunos casos se ha tomado como punto de partida sistemas ya desarrollados en diferentes instituciones del Sistema de Salud Pública, los cuales gestionaban la información de forma aislada y no garantizaban diferentes aspectos como la disposición de información única y confiable para la toma de decisiones en los diferentes niveles de dirección, la integridad de la

información y la interconexión entre las diferentes aplicaciones para poder lograr un flujo de información lógico.

Por estos motivos se decide que la Empresa Softel perteneciente al Ministerio de la Informática y las Comunicaciones (MIC), desarrollara un sistema que permitiera integrar la información de diferentes áreas de una manera sencilla y eficiente. Para ello se utilizó una arquitectura que permite la utilización de los servicios brindados por los diferentes componentes, da soporte a los requerimientos de software del usuario y facilita la integración de los diversos componentes de este sistema. Es en este momento que comienza el desarrollo del Registro Informatizado de Salud (RIS), implementándose cinco módulos iníciales.

Posteriormente a esta informatización, surgió el proyecto Atención Primaria de la Salud (APS), el cual se lleva a cabo por especialistas de la empresa Softel, estudiantes y profesores pertenecientes a la Universidad de las Ciencias Informáticas (UCI) y expertos funcionales del MINSAP. Este equipo de trabajo también desarrolla otros componentes del Sistema de Información para la Salud (SISalud). Surge así la necesidad de ubicar a las personas (pacientes y personal de la salud), configurar las áreas de salud, asignar una dirección a las unidades de salud, centros laborales, viviendas de una población y a los locales de consulta de un área. Para ello son necesarios datos como Localidad, Calle, Manzana, Consejo Popular, Circunscripción, CDR, para lograr que los módulos necesitados de estos datos funcionen de forma eficiente.
Teniendo en cuenta lo anteriormente descrito se presenta como **Situación Problémica** la siguiente afirmación:

El país no cuenta con un sistema automatizado a nivel nacional disponible, que proporcione los datos referentes a la ubicación geográfica y localidades.

Después de haber analizado la situación que existe respecto a este tema, se puede definir como **Problema Científico** la siguiente interrogante ¿Cómo desarrollar un sistema automatizado para gestionar de forma eficiente toda la información referente a la ubicación geográfica y localidades en el Sistema Nacional de Salud?

Se define como **Objeto de Estudio** el Proceso de gestión de la información para el Sistema Nacional de Salud.

El **campo de acción** se enfoca en el proceso de gestión de la información de la Ubicación Geográfica y Localidades para el Sistema Nacional de Salud.

Como **Objetivo de la Investigación** para solucionar los problemas mencionados se presenta lo siguiente: Desarrollar una aplicación web que automatice la gestión de la información de la ubicación geográfica y localidades para el Sistema de Información para la Salud (SISalud).

Para guiar la investigación se propone la siguiente **idea a defender**:

El desarrollo de una aplicación web que automatice la gestión de la información de la Ubicación Geográfica y Localidades del país, brindará información homogénea para el Sistema de Información para la Salud.

Para dar cumplimiento al objetivo es necesario llevar a cabo las siguientes **tareas**:

1- Identificar la información necesaria para el Sistema de Información para la Salud, relacionada con la Ubicación Geográfica y las Localidades.

2- Aplicar de la Arquitectura definida por el MINSAP, Basada en Componentes y Orientada a Servicios, teniendo en cuenta:

-Plataforma de Servicios PlaSer.

-Registro Informatizado de Salud (RIS).

3- Analizar la integración con otros componentes del Sistema de Información para la Salud (SISalud).

4- Modelar los flujos de trabajo “Negocio”, “Requerimientos”, “Análisis”, “Diseño” e “Implementación” mediante Proceso Unificado de Desarrollo (RUP) utilizando el Lenguaje de Modelado Unificado (UML) con su extensión para aplicaciones Web.

5- Implementar los métodos del negocio y capa de presentación del Registro de Ubicación Geográfica y del Registro de Localidades.

Después que se cumplan las tareas anteriormente mencionadas se esperan un conjunto de **beneficios** para la Informatización del Sistema Nacional de Salud, entre los que se encuentra:

- Se dispondrá de los registros que gestionan de manera homogénea la información del Registro de Ubicación Geográfica y el Registro de Localidades del país
- Se dispondrá de información de la Ubicación Geográfica para asignar una dirección a un personal de salud, paciente o ciudadano.
- Ambos registros, brindarán información para ubicar las viviendas y centros laborales que forman parte de las poblaciones en las áreas de salud.

El presente documento está compuesto por cuatro capítulos, que incluyen todo lo relacionado con el trabajo investigativo realizado, así como el diseño e implementación de los sistemas en cuestión. Además cuenta con introducción, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, anexos y glosario de términos.

Capítulo 1. Fundamentación Teórica: Se describen los conceptos fundamentales, antecedentes y la situación problémica relacionada con el dominio del problema. Se hace una valoración crítica de las soluciones existentes tanto en el ámbito nacional como internacional. Además, se hace un estudio de las técnicas, tendencias y tecnologías a considerar, en las que se apoya la solución del problema.

Capítulo 2. Características del sistema: Se exponen los conceptos fundamentales de las entidades involucradas, mediante un modelo de dominio. Se presentan los requerimientos funcionales y no funcionales. Se describen los casos de uso del sistema y se justifican los actores.

Capítulo 3. Diseño del sistema: Se justifican los patrones de diseño a utilizar en el desarrollo de la aplicación. Se muestran los diagramas de clases por casos de uso utilizando estereotipos Web. Se presenta el diagrama de clases persistentes y el modelo de datos. Como resultado del flujo de trabajo Diseño se generan los artefactos mencionados anteriormente, que son el punto de partida para la implementación del sistema.

Capítulo 4. Implementación: En este capítulo se justifica la integración con otros componentes. Se muestra el modelo de implementación con el diagrama de componentes y el diagrama de despliegue. Se muestra una descripción detallada de los métodos más complejos o agentes en el proceso de implementación, así como los estándares de diseño, codificación y tratamientos de errores utilizados.

Capítulo 1: Fundamentación Teórica.

Introducción.

El objetivo de este capítulo es abordar distintos aspectos que se utilizan como soporte teórico del sistema diseñado y un análisis exhaustivo de las principales tendencias y tecnologías que van a ser utilizadas para el desarrollo del sistema. Se explica la estructura del Sistema Nacional de Salud (SNS), el Registro Informatizado de la Salud (RIS) y el Sistema de Información para la Salud (SISalud) como la nueva etapa de la informatización del Sistema Nacional de Salud. También se presenta los antecedentes y la situación problémica relacionada con el dominio del problema. Se hace una valoración crítica de la arquitectura, lenguajes de programación, gestor de bases de datos, metodología de desarrollo de software y herramientas a utilizar.

1.1 Sistema Nacional de Salud.

El Ministerio de Salud Pública (MINSAP) es el Organismo rector del Sistema Nacional de Salud. Encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la Salud Pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica. [5]

Desde el propio triunfo revolucionario se comenzó a trabajar por la creación del Sistema Nacional de Salud que llevó la acción del trabajador de la salud a los lugares más apartados. El sistema creado comenzó a realizar importantes reformas a partir de los años 60, como parte fundamental de las transformaciones del período revolucionario y en respuesta al respeto más absoluto de uno de los derechos humanos fundamentales de todo ciudadano. [6]

El Sistema Nacional de Salud cubano tiene un carácter estatal y social, permitiendo una accesibilidad de forma gratuita a estos servicios que se extienden a todo lo largo del país, con una activa participación de la comunidad. También como parte de sus principios rectores se encuentra la colaboración internacional permitiendo extender estos servicios no sólo a los lugares más intrincados de la geografía nacional, sino a cualquier parte del mundo que lo necesite, muestra de ello fue la creación del Contingente Internacional

de Médicos Especializados en Situaciones de Desastre y Graves Epidemias “Henry Reeve” el pasado 19 de Septiembre del 2005.

El Sistema Nacional de Salud se organiza en 3 niveles de atención a la población: Atención Primaria, Atención Secundaria y Atención Terciaria.

El Dr Cosme Ordoñez profesor y científico cubano dedicado a la Atención Primaria, definió a la Atención Primaria de Salud como:

El conjunto de actividades planificadas de atención médica integral que tienen como objetivo alcanzar un mayor nivel de salud en el individuo y la comunidad, aplicando la metodología científica con la óptima utilización de los recursos disponibles y la participación activa de las masas organizadas. [7]

La Atención Primaria de Salud (APS) representa el primer nivel de contacto del SNS con los individuos, la familia y la comunidad. La APS como modelo para desafiar los problemas de salud, vistos en las dimensiones biopsicosocial y ligadas al desarrollo político económico de los países es hoy, discutida como estrategia efectiva. [8]

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

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

El Sistema Nacional de Salud de Cuba presenta otra clasificación atendiendo a su distribución administrativa:

El Sistema Nacional de Salud se estructura en tres niveles que se corresponden con la estructura político-administrativa del país.**(Ver Anexo I)** El nivel nacional está representado por el Ministerio de Salud Pública y es el órgano rector con funciones metodológicas, normativas y de coordinación y control al cual se le

subordinan directamente los centros universitarios, institutos de investigaciones, centros hospitalarios de asistencia médica altamente especializados, centros de distribución y comercializadoras de suministros y tecnologías médicas así como otros centros y entidades nacionales destinados a actividades técnicas y de apoyo. Los otros dos niveles están representados por las direcciones provinciales y municipales de salud que agrupan las instituciones de salud a su respectivo nivel y que, al igual que en el nivel central, se subordinan desde el punto de vista administrativo a las estructuras de Gobierno en los distintos niveles organizativos, representando sus intereses ante ellos y dando respuesta a las demandas y necesidades de la población. [9]

1.2 Informatización del Sistema Nacional de Salud.

A partir del año 2003 el Ministerio de Salud Pública de la República de Cuba (MINSAP) ha definido como una prioridad su informatización convocando para ello a un grupo de instituciones del sector de la salud y del Ministerio de Informática y Comunicaciones para, de manera conjunta definir los proyectos a desarrollar, tomando como punto de partida en algunos casos los sistemas ya desarrollados en el país. [10]

El principal objetivo de la informatización del Sistema Nacional de Salud es acercar eficientemente y con calidad, la prestación de los servicios de salud a la población. Para cumplir este objetivo se pretende implementar un Programa General de Informatización del Sistema Nacional de Salud teniendo como centro del mismo al Policlínico, que apoye las Estrategias y Políticas trazadas por la dirección del país y el MINSAP; de manera que se logre la incorporación progresiva y sistemática de las Tecnologías de la Información y las Comunicaciones (TIC) en función de la adquisición y gestión del conocimiento y los servicios de salud.

Este proceso de informatización tiene como perspectiva que las Instituciones del país alcancen un elevado nivel de informatización de las actividades que brindan, partiendo del Sistema de Atención Primaria y tomando como eje al policlínico. De manera que estas redunden en un incremento de la calidad, efectividad y eficiencia de los servicios que se presten a la población contribuyendo al logro de la satisfacción de los prestadores y usuarios del Sistema Nacional de Salud.

Al concebirse de manera integrada, los datos generados en los distintos niveles de atención en el SNS, tienen un proceso de captura, registro, procesamiento, validación y análisis de la información inestimable, lo cual incrementa su consistencia, veracidad y oportunidad. Lo anterior redunda finalmente en el mejoramiento de la actividad administrativa, asistencial, docente y de investigación. De ahí que es importante repetir que es esta integración la que permite hablar de informatización en el sector de la salud pública cubana, no de proyectos aislados. [11]

Muchos son los beneficios esperados de este proceso de informatización tanto para la población en general como para el fortalecimiento del Sistema Nacional de Salud. A continuación se mencionan algunos de estos beneficios. [12]

Para la población:

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

Para el SNS:

- Gestión oportuna de una información confiable y actualizada que propiciará una optimización considerable de recursos con su lógico resultado en la reducción significativa de costos de operación de las entidades que conforman el SNS.
- Elevación de la capacidad y calidad de las tomas de decisiones asistenciales y gerenciales por la disposición oportuna de información actualizada para todos los niveles del SNS.

- Disponer de un soporte y herramientas poderosos para la formación y actualización constante de sus miembros desde sus propios escenarios de desempeño con la consecuente equidad de conocimientos independientemente de áreas geográficas y nivel de atención.
- Potenciaría la investigación científica multi-céntrica nacional e internacional.

1.2.1 Registro Informatizado de Salud (RIS).

Por definición, el RIS es la solución informática integral para la Salud Pública, acorde con los objetivos de la informatización de la sociedad cubana. Constituido por un conjunto de aplicaciones independientes (módulos del sistema) que se interconectan según las necesidades del flujo de información. Es además la herramienta que permite a los usuarios autorizados combinar la información de los diferentes módulos que lo componen, para obtener una información integral en tiempo real para la toma de decisiones en los diferentes niveles de dirección, la docencia, investigación y la gestión en salud. [13]

Para el desarrollado del RIS se utilizó la metodología de la dirección integrada de proyecto se organizó el proceso en 4 grupos de trabajo en la Empresa Softel del Ministerio de Informática y Comunicaciones, integrados por especialistas de la propia Empresa, de la empresa ESATEL de Santiago de Cuba, del Centro para el Desarrollo Informático de la Salud (CEDISAP) e Infomed, estos dos últimos pertenecientes al Ministerio de Salud Pública. [14]

El Registro Informatizado de Salud constituye la materialización de la estrategia metodológica de Informatización del Sistema Nacional de Salud como parte de la Informatización de la Sociedad. Fue desarrollado e implementado siguiendo las políticas para el desarrollo informático aprobadas por el sector a finales del 2003. Es una aplicación orientada a servicios bajo una arquitectura moderna y versátil, es un producto portable que no depende del motor de base de datos escogido para su desarrollo y funciona tanto con el Sistema Operativo “Windows” como en “Linux”. [15]

En el año 2004 se crea el Proyecto Atención Primaria de la Salud (APS) el cual está integrado por especialistas de la Empresa de Soluciones Informáticas Softel, expertos del Ministerio de Salud Púbica (MINSAP), estudiantes y profesores de la Facultad número 7 de la Universidad de Ciencias Informáticas. El proyecto tiene como misión, perfeccionar componentes existentes en el RIS y desarrollar otros que

complementen la nueva propuesta de informatización del Sistema Nacional de Salud, el Sistema de Información para la Salud (SISalud).

1.2.2 Sistema de Información para la Salud.

El Sistema de Información para la Salud (SISalud) es la propuesta que hace la Empresa de Soluciones Informáticas Softel para dar continuidad al proceso de informatización del Sistema Nacional de Salud (SNS). Esta nueva aplicación permitirá organizar e integrar los componentes ya desarrollados y los que se van a desarrollar como parte de la informatización del SNS. SISalud estará compuesto por el Registro Informatizado de Salud (RIS), el Sistema Automatizado de Atención (SIAP), Sistema Informatizado de Gestión Hospitalaria (SIGH) y el Sistema Informatizado de Atención Especializada (SIAE). [16]

Registro Informatizado de Salud (RIS): Formado por el Registro Informatizado de Salud No Médico (RISNM), el cual está propuesto por los registros que son administrados o gestionados a nivel nacional o central y el Registro Informatizado de Salud Médico (RISM) integrado por los registros que pueden ser accedidos desde cualquier nivel de atención o institución de salud para lograr la continuidad en el seguimiento del paciente. En otras palabras estará formado por todos los módulos o componentes que no son del dominio de Atención Primaria propiamente, pero procesan y generan información que se obtiene de este nivel comunitario y además lo retroalimenta.

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

Sistema Informatizado de Gestión Hospitalaria (SIGH): Se agruparán aquí los módulos que pertenecen al nivel de atención secundario u hospitalario.

Sistema Informatizado de Atención Especializada (SIAE): Se agruparán aquí los módulos que pertenecen al nivel de atención terciario o especializado.

En esta nueva propuesta de informatizar el Sistema Nacional de Salud surge la necesidad de contar con información no perteneciente al dominio de Salud y de vital importancia, como es, la ubicación geográfica y localidades del país. El análisis de este contexto lleva a la siguiente situación problémica.

1.3 Situación Problémica y problema a resolver.

El país no cuenta con un sistema automatizado a nivel nacional disponible, que proporcione los datos referentes a la ubicación geográfica y localidades. Entiéndase por ubicación geográfica la organización física del país, que responde a una distribución lógica del territorio nacional, que contempla entidades como: Provincias, Municipios, Localidades, Calles y Manzanas. Se entiende por Localidades, la organización más bien estratégica del estado cubano para la toma de decisiones, que contempla entidades como: Consejos Populares, Circunscripciones, Zonas y Comités de Defensa de la Revolución (CDR). Para el Sistema de Información para la Salud es de vital importancia contar con la información anteriormente mencionada, necesaria para:

- Asignar una dirección a una persona (personal de la salud, paciente o un ciudadano).
- Asignar una dirección a una unidad de salud, centro laboral, vivienda o local de consulta.
- Configurar las Áreas de Salud y garantizar que funcione correctamente el Registro de Población.
- Permitir al módulo de Administración gestionar los usuarios por los distintos niveles del Sistema Nacional de Salud (SNS).

Al no existir una solución centralizada y disponible que brinde este tipo de información, se dio la necesidad de simular estos sistemas.

1.4 Análisis de las soluciones existentes.

Muchas son las soluciones existentes vinculadas al problema a resolver a nivel internacional. Una de ellas es el Callejero de España o también conocido como Códigos Postales de España. El Callejero de España brinda información actualizada de los Códigos Postales en diferentes formatos, como son, aplicaciones Web y de ventana o escritorio con posibilidades de filtrado de información.

Otra solución existente vinculada al ámbito nacional, es un componente llamado Registro de Ubicación Geográfica desplegado en el Registro Informatizado de la Salud (RIS).

Seguidamente se explican y analizan las características de estas soluciones de manera detallada para una mejor comprensión de las soluciones existentes.

Códigos Postales de España

Códigos Postales de España es un software que cuenta con tres bases de datos: Ciudades, Callejero y Poblaciones, contiene información de España y Andorra. Estas bases de datos manejan información de Personas, Provincias, Localidades, Calles, Tipos de vías, Direcciones, Códigos Postales, etc. Con este software se puede:

- Buscar la Provincia, Localidad, Calle y el Código Postal de una o varias Localidades, por la Localidad, Dirección ó Número. (Ver Anexo II)
- Buscar la Localidad, Calle y Código Postal de una o varias Localidades, por el País (España o Andorra), Provincia, Localidad, Dirección ó Número. (Ver Anexo III)
- Buscar la Provincia, Localidad y Código Postal de una o varias Localidades por el Código Postal. (Ver Anexo IV)

Este sistema implementado en España, ofrece enormes ventajas permitiendo gestionar y brindar información de vital importancia para cualquier usuario o sistema. Algunas de las características del Callejero de España se ajustan a los conceptos de los componentes a desarrollar como son: Provincias, Localidades y Calles. Sin embargo, la información que se maneja no es de Cuba, la estructura organizativa de España no se corresponde con la del país y la información que proporciona no son Servicios Web que puedan ser consumidos por sistemas externos; esta situación hace que el software Código Postales de España sea inadecuado para dar respuesta a la situación existente en el Sistema de Información para la Salud (SISalud).

Registro de Ubicación Geográfica del RIS

En el proceso de implementación del Registro Informatizado de la Salud (RIS) en sus inicios se realizaron los módulos Administración, Registro de Unidades de Salud, Registro de Equipos Médicos, Registro de Equipos No Médicos y Registro de Personal de la Salud. Para el correcto funcionamiento de dichos módulos era necesario un Registro que gestionara las Provincias y Municipios que hasta ese momento, era la información que se necesitaba. Por esta razón se crea el Registro de Ubicación Geográfica para brindar información que permita gestionar la dirección de un personal de salud, una unidad de salud o permitir al módulo de Administración la gestión de los usuarios en los distintos niveles del Sistema Nacional de Salud (SNS); ofreciendo la información de las Provincias y Municipios.

Este sistema resuelve parcialmente el problema existente, ya que brinda información de las Provincias y gestiona información de los Municipios al nivel nacional. Sin embargo, es de vital importancia gestionar la información referente a las Localidades, Calles y Manzanas; indispensable para los módulos de la Atención Primaria, cuya funcionalidad básicamente parte de esta información. Además, la aplicación debe dar la posibilidad de gestionar la información en los niveles provincial y municipal, que serán básicamente los encargados de introducir esta información. Por las razones anteriormente expuestas este Registro de Ubicación Geográfica no responde a las exigencias del Sistema de Información para la Salud (SISalud).

1.5 Política para el desarrollo de la informatización del Sistema Nacional de Salud.

En esta nueva etapa se ha definido por el Ministerio de Salud Pública (MINSAP) un grupo de premisas y requisitos que incorporan los últimos adelantos en el área de las tecnologías de la información y las comunicaciones y que garantizan la plataforma de integración de las aplicaciones, la compatibilidad y sostenibilidad de los productos a desarrollar, tales como: empleo de tecnologías basadas en Internet (XML, Web Services), software libre (PHP, MySql, Linux), documentación de todo el proceso productivo, requisitos de seguridad del software, independencia de la base de datos, desarrollo en multiplataforma y empleo de estándares internacionales para los productos relacionados con la salud. El soporte de infraestructura en todos los aspectos mencionados es la Red Telemática de la Salud. [17]

Todos los productos y servicios se integrarán a la ciberinfraestructura del sector y se realizarán en lo fundamental sobre sistemas abiertos, arquitectura orientada a los servicios y basadas en componentes, utilizando software libre y de calidad. Deben constituirse en componentes modulares y estables, que compartan normas y cooperen entre si. [18]

Por las razones anteriormente expuestas se decidió realizar la investigación de las tendencias y tecnologías actuales vinculadas a las políticas de la informatización del SNS, así como los estándares definidos por la empresa Softel.

1.6 Tendencias y tecnologías actuales a considerar.

En este epígrafe se tratarán los conceptos fundamentales relacionados con los lenguajes de programación, gestor de bases de datos, arquitectura, tecnologías, herramientas y metodologías que se considerarán para el proceso de desarrollo de los sistemas a desarrollar.

1.6.1 Internet.

Definición de Internet: Un consorcio *(pool)* global de información y servicios al que se puede acceder por medio de un programa interfaz ejecutado de modo local. [19]

Internet, interconexión de redes informáticas que permite a las computadoras u ordenadores conectados comunicarse directamente, es decir, cada computadora de la red puede conectarse a cualquier otra computadora de la red. El término suele referirse a una interconexión en particular, de carácter planetario y abierto al público, que conecta redes informáticas de organismos oficiales, educativos y empresariales. También existen sistemas de redes más pequeños llamados intranets, generalmente para el uso de una única organización, que obedecen a la misma filosofía de interconexión. La tecnología de Internet es una precursora de la llamada “superautopista de la información”, un objetivo teórico de las comunicaciones informáticas que permitiría proporcionar a escuelas, bibliotecas, empresas y hogares acceso universal a una información de calidad que eduque, informe y entretenga. A finales de 1998 estaban conectados a Internet unas 148 millones de computadoras, y la cifra sigue en aumento. [20]

1.6.2 Software libre.

El Software Libre es un asunto de libertad, no de precio. Para entender el concepto, debe pensarse en *libre* como en libertad de expresión, no como en cerveza gratis. Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software: [21]

- La libertad de usar el programa, con cualquier propósito (libertad 0).

- La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puede ayudar a otros (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3). De igual forma que la libertad 1 el acceso al código fuente es un requisito previo.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, deberías tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos. [22]

1.6.3 Sistema Operativo Linux.

Linux, sistema operativo derivado de UNIX que, manteniendo la generalidad de sus características, como el ser multitarea y basado en bibliotecas dinámicas, puede ser ejecutado en computadoras u ordenadores personales aunque su potencia sea limitada. En sus orígenes fue desarrollado, en 1990, por el informático finlandés Linus Torvalds, que publicó su código como un denominado código abierto, esto es, accesible para toda la comunidad, sin restricciones para modificarlo y ampliarlo. Este planteamiento, favorecido por su estructura modular (basado en la instalación de diversos paquetes), generó una nueva visión de desarrollo informático, ya que su expansión fue debida a la aportación, generalmente voluntaria y sin ánimo de lucro, de multitud de desarrolladores independientes. En la actualidad este sistema operativo ha obtenido un cierto apoyo por parte de la industria, de forma que empresas como IBM o Hewlett-Packard lo integran en algunas de sus computadoras y prestan el soporte técnico correspondiente, normalmente como parte de los sistemas servidores. Su implantación en sistemas para usuarios finales, aún no ha alcanzado la extensión que tiene en algunos de los ámbitos más profesionales, muy especialmente en servidores de Internet. [23]

1.6.4 Servicios Web.

Un Servicio Web es cualquier servicio que esté disponible en Internet o en las redes privadas (Intranet), utiliza un sistema estandarizado de la mensajería XML (del inglés Extensible Markup Language, Lenguaje de Marcas Extensible), y no se ata a ningún sistema operativo o lenguaje de programación.

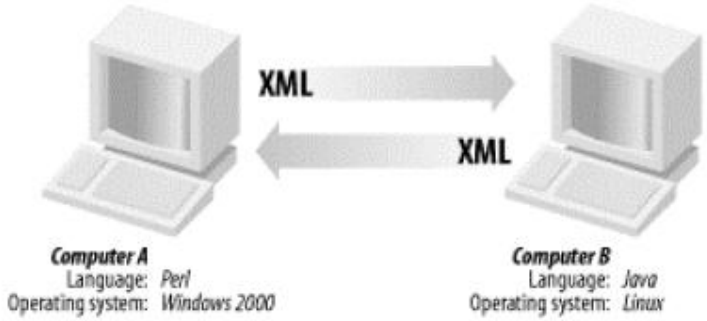


Figura 1.1 Servicio Web básico.

Los Servicios Web poseen un conjunto de protocolos y servicios que son utilizados para el formato, comunicación y transporte de la información. XML es el formato estándar para la información que se vaya a intercambiar. El protocolo SOAP (del inglés Simple Object Access Protocol, Protocolo de Acceso Simple a Objetos) o XML-RPC (del inglés Remote Procedure Call, Llamada a Procedimiento Remoto) es el protocolo de comunicación que establecen el intercambio de información, y otros como HTTP (del inglés, Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto), FTP (del inglés, File Transfer Protocol, Protocolo de Transferencia de Ficheros), o SMTP (del inglés, Simple Mail Transfer Protocol, Protocolo Simple de Transferencia de Correo Electrónico) que facilitan el transporte de la información. WSDL (del inglés, Web Services Description Language, Lenguaje de Descripción de Servicios Web) es el lenguaje de la interfaz pública para los Servicios Web, es una descripción basada en XML de los requisitos funcionales necesarios para establecer la comunicación. UDDI (del inglés Universal Description, Discovery, and Integration, Universal Descripción, Descubrimiento e Integración) se utiliza para publicar la información de los Servicios Web, permite a las aplicaciones comprobar qué servicios web están disponibles. [24]

1.6.5 Arquitectura de Software.

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad. [25]

La arquitectura de software se centra tanto en los elementos estructurales significativos del sistema, como subsistemas, clases, componente y nodos, como en las colaboraciones que tienen lugar entre estos elementos a través de las interfaces.

Arquitectura Cliente/Servidor.

Arquitectura cliente/servidor, arquitectura *hardware* y *software* adecuada para el proceso distribuido, en el que la comunicación se establece de uno a varios. Un proceso es un programa en ejecución. Proceso cliente es el que solicita un servicio. Proceso servidor es el capaz de proporcionar un servicio. Un proceso cliente se puede comunicar con varios procesos servidores y un servidor se puede comunicar con varios clientes. Los procesos pueden ejecutarse en la misma máquina o en distintas máquinas comunicadas a través de una red. Por lo general, la parte de la aplicación correspondiente al cliente se optimiza para la interacción con el usuario, ejecutándose en su propia máquina, a la que se denomina terminal o cliente, mientras que la parte correspondiente al servidor proporciona la funcionalidad multiusuario centralizada y se ejecuta en una máquina remota, denominada de forma abreviada, simplemente, servidor. [26]

Arquitectura tres capas.

La arquitectura de tres capas es la más común en sistemas de información que además de tener una interfaz de usuario contemplan la persistencia de los datos. Cada capa o nivel es un proceso separado y bien definido corriendo en plataformas separadas. Las tres capas o niveles son: presentación, negocio y almacenamiento. **(Ver Anexo V)**

Capa de Presentación: Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. [27]

Capa de Negocio: Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la

capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. [28]

Capa de Datos: Es donde residen los datos. Está formada por uno o más gestores de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. [29]

Desarrollo de Software Basado en Componentes (DSBC).

El Desarrollo de Software Basado en Componentes (DSBC) se centra en el desarrollo de grandes sistemas de software mediante la integración de componentes de software ya existentes. Realiza y mejora la flexibilidad y la manutención proporcionando la reducción de costos en el desarrollo de software, incremento en la velocidad de ensamble de nuevos sistemas y reducción de carga de mantenimiento para correcciones y mejoras. El DSBC introduce los términos de **Alta Cohesión** y **Bajo Acoplamiento**.
Alta Cohesión: Dice que la información que almacena un componente debe ser coherente y está en la mayor medida de lo posible relacionada con otro componente.
Bajo Acoplamiento: Es tener los componentes lo menos ligados entre sí que se pueda. De tal forma, en caso de producirse una modificación en alguno de ellos, se tendrá la mínima repercusión posible en el resto de los componentes, potenciando la reutilización, y disminuyendo la dependencia entre los componentes.

Arquitectura Orientada a Servicio (SOA).

La SOA Tradicional es aquella que utiliza los principios y tecnologías básicos de los Servicios Web. Esto significa utilizar SOAP como lenguaje de intercambio, WSDL como lenguaje para la descripción de los servicios y UDDI para la publicación o registro de los mismos. En el dibujo que se muestra a continuación, se puede ver la estructura básica de funcionamiento de una SOA tradicional. Ver figura 1.3. [30]

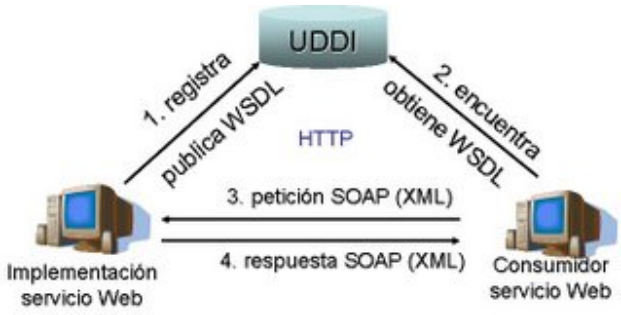


Figura 1.2 SOA tradicional.

En la figura anterior, se puede observar la existencia de tres roles claramente diferenciados:

Cliente del servicio: Es el que solicita la ejecución del servicio web, y por lo tanto el que lo consume.

Proveedor del servicio: Es el encargado de implementar el servicio web y ofrecerlo a los clientes.

Registro del servicio: Es un repositorio donde se almacenan las descripciones de los servicios, para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades.

La secuencia de ejecución es la siguiente:

1. El proveedor del servicio da alta al servicio web en el registro. Para realizar esto, el proveedor almacena en el registro el documento de descripción de este.

2. El solicitante del servicio busca en el registro un servicio web que pueda adaptarse a sus necesidades.

3. Una vez seleccionado el servicio, el solicitante lo invoca mediante el envío de un mensaje SOAP, en el cual se indica la acción a realizar y los datos de entrada.

4. El servicio web recibe la petición y ejecuta la funcionalidad. Para finalizar envía un mensaje SOAP al solicitante con los resultados obtenidos.

Por lo tanto, una SOA tradicional estará compuesta por un conjunto de servicios que reciben y envían mensajes SOAP en base a una descripción WSDL. Este tipo de arquitecturas hoy en día es muy utilizado, pero no es el más óptimo debido a que no proporciona una serie de características esenciales a la hora de crear una Arquitectura profesional. [31]

1.6.6 Plataforma de Servicio (PLASER).

La Plataforma de Servicios (PLASER) está conformada fundamentalmente por varias clases en PHP, una librería, que puede ser usada opcionalmente para que un componente se integre al Sistema de Información para la Salud, pero de no ser usada, la seguridad del sistema corre a cuenta del programador. En esta versión PLASER sólo soporta como llamada RCP el protocolo SOAP, pero en futuras versiones se incorporarán otros protocolos de transporte o incluso el acceso local a código a nivel de File System, de forma tal que para el programador sea totalmente transparente si la invocación del proceso es remota, local, por SOAP, directamente a código, etc. [32]

Este sistema está concebido completamente sobre Arquitectura Basada en Componentes y Orientada a Servicios, usando el paradigma de XML Web Services específicamente SOAP. En su concepción se han utilizado estándares actuales y normas abiertas. PLASER constituye una plataforma sobre la que se pueden desplegar aplicaciones XML – Web Services, con la ventaja de que el programador no tiene que preocuparse por implementar la seguridad del Sistema, ya que esta es una de las tareas que asume PLASER, además facilita la programación y homogeneidad de los componentes. PLASER desde el punto de vista estructural permite trabajar con cualquier base de datos que cumpla con la norma SQL ANSI 92; pero desde el punto de vista de implementación sólo trabaja con las bases de datos soportadas por el componente DBX, ya que PLASER encapsula a dicho componente y lo utiliza para el acceso a bases de datos. [33]

1.6.7 Servidor Web Apache.

Apache es un servidor web gratuito, Open Source (Código Abierto), potente y flexible, funciona en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades, o que una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia.

Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor, seleccionando que módulos se van a cargar, ya sea al compilar o ejecutar el servidor.

1.6.8 Lenguajes de Programación.

Un lenguaje de programación es cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora. Es complicado definir qué es y qué no es un lenguaje de programación. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Normalmente es la computadora la que realiza la traducción. [34]

PHP.

PHP (del inglés PHP Hypertext Pre-processor, también conocido por sus nombres anteriores PHP Tools, o, *P*ersonal *H*ome *P*age Tools). Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend-2, desarrollado con mayor robustez para cubrir las necesidades de las aplicaciones Web actuales y se ha anunciado la aparición del la versión 6.

Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML.

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

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

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

Resumiendo, el PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte para unos 20 Gestores de Bases de Datos y contiene unas 40 extensiones estables.

XML.

XML (del inglés, Extensible Markup Lenguaje, Lenguaje de Marcas Extensible) se ha convertido en un formato estándar en Internet y está diseñado para representar datos estructurados. Como su nombre lo indica, no es un lenguaje de marcado, si no que es un metalenguaje para definir otros lenguajes de marcados adecuados a un uso específico. Este 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.

Ventajas XML:

- Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan bugs y se acelera el desarrollo de la aplicación.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

XSL.

XSL (del inglés, Extensible Stylesheet Language, Lenguaje Extensible de Hojas de Estilo) es una familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML cualquiera debe ser transformada o formateada para su presentación en un medio específico. La familia de lenguajes XSL está formada por:

- **XSLT** (del inglés, Extensible Stylesheet Language Transformations, Lenguaje de Hojas Extensibles de Transformación) permite convertir documentos XML de una sintaxis a otra (por ejemplo, de un XML a otro o a un documento HTML).
- **XSL-FO** (del inglés, Extensible Stylesheet Language Formatting Objects, Lenguaje de Hojas Extensibles de Formateo de Objetos) permite especificar el formato visual con el cual se quiere presentar un documento XML, es usado principalmente para generar documentos PDF.
- **XPath**, o **XML Path Language**, es una sintaxis (no basada en XML) para acceder o referirse a porciones de un documento XML.

JavaScript.

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores web comerciales.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.

1.6.9 Sistema de Gestión de Bases de Datos (SGBD): MySql.

Los Sistemas de Gestión de Bases de Datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Tiene como propósito manejar de manera clara, sencilla y ordenada un conjunto de datos.

MySql es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySql. Todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones. Y así ocurre. MySql ha pasado de ser una "pequeñita" base de datos a una completa herramienta y ha conseguido superar a una gran cantidad de bases de datos comerciales (lo que ha asustado a la mayor parte de los proveedores comerciales de bases de datos). Por lo tanto, su rápido desarrollo se debe a la contribución de mucha gente al proyecto, así como a la dedicación del equipo de MySql. [35]

MySQL es un SGBD multihilo (permite dividir un programa en dos o más tareas que corren simultáneamente), es multiusuario, desarrollado como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL (del inglés, General Public License, Licencia Pública General, licencia creada por la Free Software Foundation a mediados de los 80), pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. MySQL acaba de lanzar el pasado 27 de febrero 2007 su versión 5.0.37.

1.6.10 Metodologías de Desarrollo de Software.

Una Metodología de Desarrollo de Software se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente. En la actualidad las 3 Metodologías de Desarrollo más utilizadas son RUP (del inglés, Rational Unified Process, Proceso Unificado de Desarrollo), XP (del inglés, Extreme Programing, Programación Extrema) y MSF (del inglés, Microsoft Solution Framework, Framework de Soluciones de Microsoft).

La metodología de desarrollo de software usada es el Rational Unified Process (RUP), la cual utiliza UML como lenguaje para realizar los modelos. La empresa Softel, decidió utilizarla por las ventajas que ofrece y por los conocimientos adquiridos en la asignatura Ingeniería de Software por los estudiantes que laboran en el proyecto. De esta forma se pone en práctica, se reafirman estos conocimientos y se agiliza el proceso de producción.

Lenguaje Unificado de Modelado (UML).

UML (del inglés, Unified Modeling Language, Lenguaje Unificado de Modelado) es un lenguaje para visualizar, especializar, construir y documentar los artefactos de un sistema o sistemas de software. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. La finalidad de los diagramas es presentar diversas perspectivas la aplicación, a las cuales se les conoce como modelo. El modelo UML de un sistema es similar a un modelo a escala de un edificio junto con la interpretación del artista del edificio. Es importante destacar que un modelo UML describe lo que supuestamente hará la aplicación, pero no dice cómo implementarla. En la actualidad existe una extensión de UML para modelar aplicaciones Web.

Extensión de UML para Web.

A finales de los 90, cuando el desarrollo de aplicaciones Web se hizo más importante, Jim Conallen hace uso de las facilidades de extensión brindadas por el UML para basado en este lenguaje modelar aplicaciones Web. Publica hacia junio del 99 varios artículos incluido “Modeling Web Applications Architectures with UML” donde describe la extensión de UML para el modelado de aplicaciones Web. [36]

La extensión de UML para modelar las aplicaciones Web, tan populares en la actualidad, puede resultar útil, para idear y documentar y generar la solución final de software.

Proceso Unificado de Desarrollo (RUP).

RUP (del inglés, Rational Unified Process, Proceso Unificado de Rational) es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo.

RUP define "quién" (trabajadores) está haciendo "qué" (artefactos), "cuándo" (flujo de trabajos) y "cómo" (actividades) para alcanzar un determinado objetivo. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

1.6.11 Herramientas a utilizar.

Después de haber definido la arquitectura, servidor web, lenguajes de programación, sistema de gestión de bases de datos y la metodología de desarrollo de software a seguir, sólo queda seleccionar las herramientas que son las encargadas de dar un mayor rendimiento a las tecnologías anteriormente mencionadas. Las herramientas seleccionadas son: Zend Studio para la programación en PHP, MySQL Front 3.2 como administrador de la base de datos, Stylus Studio como editor de XML y "debugger" de XSLT, para el diseño Dreamweaver 8 y para modelar los diferentes artefactos generados de los flujos de trabajos se utilizó el Rational Rose.

Zend Studio.

Zend Studio es un ambiente integrado del desarrollo (IDE) disponible para los desarrolladores profesionales de PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa está escrito en Java, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más. También posee herramientas para el manejo de bases de bases de datos, apresura ciclos de desarrollo y simplifica proyectos complejos.

MySQL Front 3.2.

MySQL-Front es un administrador de bases de datos MySQL gratuito con una sencilla pero útil aplicación diseñada especialmente para desarrolladores que trabajan con MySQL. Desde el primer momento en que se empieza a usar este administrador, se descubre su facilidad para obtener información sobre las bases de datos, tanto de sus tablas como de su estructura y contenido.

Stylus Studio.

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. Soporta además edición visual de XQuery, e incluye un editor DTD, utilidades para XPath, y mucho más.

Macromedia Dreamweaver 8.

Macromedia Dreamweaver 8 creado por Macromedia (actualmente Adobe Systems) permite a los desarrolladores Web, desde la creación y mantenimiento de sitios Web básicos hasta aplicaciones avanzadas compatibles con las mejores prácticas y las tecnologías más recientes.

Ventajas:

- Incluye herramientas para trabajar aplicaciones que manejan XML.
- Usa un editor de diseño y código de primera calidad en una sola herramienta.
- El panel unificado de CSS ofrece una representación sencilla y directa de la cascada de estilos aplicados al contenido y ofrece acceso rápido para realizar cambios sin necesidad de realizar búsquedas en el código probando por ensayo y error.

Rational Rose.

Rational Rose es una herramienta CASE (del inglés, Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadoras) desarrollada por los creadores de UML (Booch, Rumbaugh y

Jacobson), propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas, creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Rational cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Ventajas:

- Permite que hayan varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.
- También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades.
- Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.
- Proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

Conclusiones

En este capítulo se realizó una sistematización de los elementos teóricos que sustentan el desarrollo propuesto. Se profundizó en los conceptos y definiciones necesarias para comprender la situación que llevó a la realización del presente trabajo. También se realizó una profunda y crítica investigación de las tendencias, tecnologías, arquitectura, lenguajes, metodologías de desarrollo y herramientas a utilizar en todo el proceso de desarrollo. Siempre siguiendo las políticas definidas por el Ministerio de Salud Pública (MINSAP) para la informatización del Sistema Nacional de Salud (SNS).

Capítulo 2: Características del Sistema.

Introducción

En el capítulo se realiza una presentación de las características del sistema. Se expone el modelo de domino o Modelo Conceptual el cual sirve de apoyo para la especificación de las condiciones, capacidades y cualidades que el sistema debe tener, los conceptos fundamentales de las entidades involucradas en el dominio y el diagrama de modelo de dominio del Registro de Ubicación Geográfica y Registro de Localidades. Se presentan los requisitos de software tanto los funcionales como los no funcionales, así como el modelo de casos de uso del sistema, la definición de los actores, diagrama de caso de uso del sistema y la descripción textual de los casos de uso de ambos registros.

2.1 Modelo de Dominio.

Al no identificarse negocio en el Registro de Ubicación Geográfica y el Registro de Localidades, se emplea un modelo de dominio el cual permite captar los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. En el modelo de dominio se describen los conceptos fundamentales, permitiendo mostrar al cliente el entorno de la información que se maneja y de esta manera contribuir a la comprensión del contexto del sistema. También se representa el diagrama del modelo de dominio, el cual permite de forma gráfica ver la relación entre los objetos o clases.

2.1.1 Conceptos Fundamentales.

Conceptos fundamentales del Registro de Ubicación Geográfica.

Provincia: Cada una de las divisiones de la República de Cuba. El país cuenta con 14 provincias y el municipio especial Isla de la Juventud considerado una provincia. Las Provincias están constituidas por Municipios.

Municipio: Divisiones territoriales que posee una Provincia. Un Municipio pertenece a una Provincia. Los Municipios están constituidos por varias Localidades.

Localidad: Nombre de un pueblo, barrio, reparto o finca. Una Localidad pertenece a un Municipio.

Calle: Vía urbana. Tramo de una vía urbana comprendido entre dos esquinas. Puede tener como nombre un número, una letra, el nombre de un santo o de un mártir de la revolución, un personaje histórico o famoso. Si es un tramo de calle (cuadra) incluye las entrecalles.

Manzana: Espacio urbano, edificado o destinado a la edificación, generalmente cuadrangular, delimitado por Calles en sus lados. Se enumeran para su identificación.

Conceptos fundamentales del Registro de Localidades.

Consejo Popular: El Consejo Popular es un órgano del Poder Popular, local, de carácter representativo, investido de la más alta autoridad para el desempeño de sus funciones. Comprende una demarcación territorial dada, apoya a la Asamblea Municipal del Poder Popular en el ejercicio de sus atribuciones y facilita el mejor conocimiento y atención de las necesidades e intereses de los pobladores de su área de acción. Esta constituido por varias Circunscripciones.

Circunscripción: Órgano del gobierno en una zona geográfica determinada que se subordina al Consejo Popular.

Zona: Es la estructura superior de organización de los CDR que se subordina a la dirección municipal de los CDR.

CDR: Organización de masa que se crea en cuadras o comunidades, a la cual pertenecen las personas de la población mayores de 16 años.

2.1.2 Diagrama del modelo de dominio.

El diagrama del modelo de dominio permite visualizar las distintas asociaciones que tienen las entidades, y la cardinalidad de sus relaciones. A continuación se presenta el diagrama del modelo de dominio del Registro de Ubicación Geográfica y Registro de Localidades respectivamente.

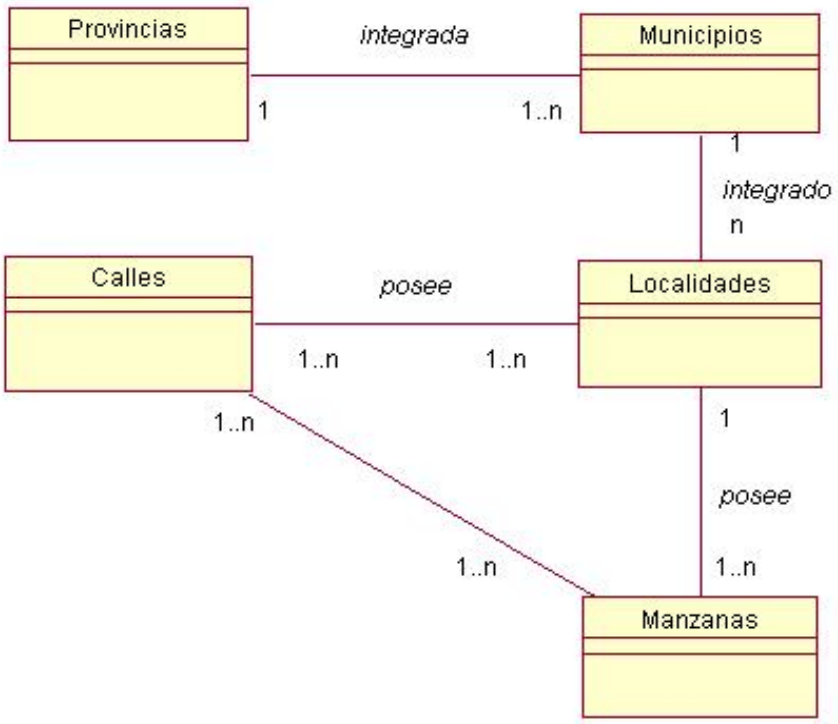


Figura 2.1 Diagrama del Modelo de Dominio del Registro de Ubicación Geográfica.

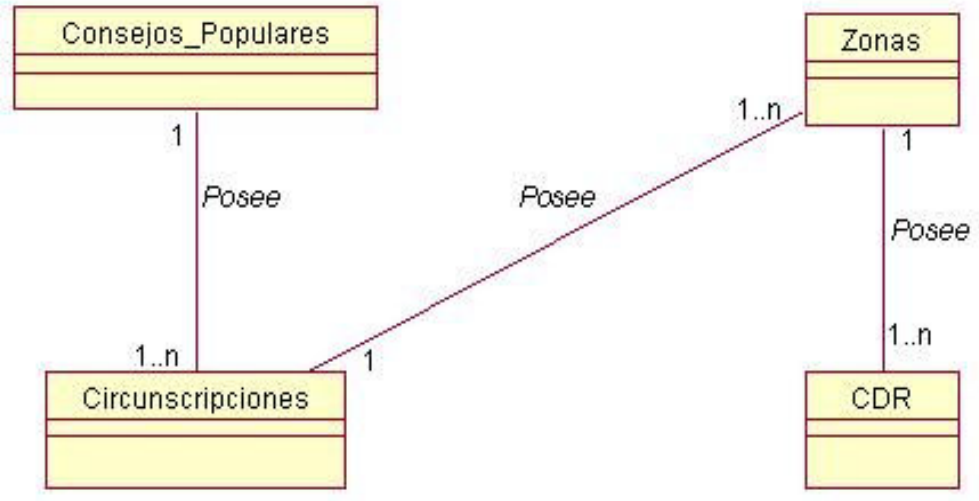


Figura 2.2 Diagrama del Modelo de Dominio del Registro de Localidades.

2.2 Propuesta del Sistema

Para dar solución al problema planteado, se decide hacer el Registro de Ubicación Geográfica y el Registro de Localidades.

El Registro de Ubicación Geográfica gestiona información de las Provincias, Municipios, Localidades, Calles y Manzanas; teniendo en cuenta la organización física del país. Las Provincias están constituidas por Municipios, haciendo un paréntesis en la Isla de la Juventud que es considerada una Provincia y está integrada por un solo Municipio de igual nombre. Los Municipios están constituidos por Localidades. Las Localidades poseen Calles y Manzanas y las Manzanas a su vez, están constituidas por Calles.

El Registro de Localidades organiza el país de forma estratégica, permitiendo al estado cubano la toma de decisiones; y gestiona información de los Consejos Populares, Circunscripciones, Zonas y Comités de Defensa de la Revolución (CDR). Los Consejos Populares están constituidos por Circunscripciones. Las Circunscripciones por Zonas y las Zonas por Comités de Defensa de la Revolución.

Se debe destacar la dependencia que tiene el Registro de Localidades del Registro de Ubicación Geográfica, ya que los Consejos Populares pertenecen a un Municipio y abarcan varias Localidades, y los Comités de Defensa de la Revolución pueden abarcar una o varias Manzanas.

Ambos sistemas permitirán la gestión y la realización de búsquedas eficientes de la información, así como la impresión de la misma. Brindarán servicios a las demás aplicaciones del Sistema de Información para la Salud. Además, contarán con un diseño de interfaces agradables y de fácil navegación, con ayuda y manual de usuario en línea.

2.2.1 Especificación de Requerimientos de Software.

Un Requerimiento de Software es la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. Los requerimientos de Software se dividen en requerimientos funcionales y requerimientos no funcionales.

2.2.1.1 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

Requerimientos Funcionales del Registro de Ubicación Geográfica

RF1: Listar las Provincias del país.

RF2: Listar los Municipios del país y los Municipios de una Provincia seleccionada previamente.

RF3: Listar las Localidades del país y las Localidades que se correspondan con los criterios introducidos previamente (Provincia, Municipio o Localidad).

RF4: Listar las Calles del país y las Calles que se correspondan con los criterios introducidos previamente (Provincia, Municipio, Localidad o Calle).

RF5: Listar las Manzanas del país y las Manzanas que se correspondan con los criterios introducidos previamente (Provincia, Municipio, Localidad o Manzana).

RF6: Gestionar la información de los Municipios.

RF6.1: Insertar Municipio (nombre del Municipio y Provincia a la que pertenece).

RF6.2: Actualizar Municipio (nombre del Municipio).

RF6.3: Eliminar Municipio. No se puede eliminar un Municipio que tenga Localidades asociadas.

RF7: Gestionar la información de las Localidades.

RF7.1: Insertar Localidad (nombre de la Localidad, Provincia y Municipio a la que pertenece).

RF7.2: Actualizar Localidad (nombre de la Localidad).

RF7.3: Eliminar Localidad. No se puede eliminar una Localidad que tenga Calles asociadas.

RF8: Gestionar la información de las Calles.

RF8.1: Insertar Calle (nombre de la Calle, Provincia, Municipio y Localidad a la que pertenece).

RF8.2: Actualizar Calle (nombre de la Calle).

RF8.3: Eliminar Calle. No se puede eliminar una Calle que tenga Manzanas asociadas.

RF9: Gestionar la información de las Manzanas.

RF9.1: Insertar Manzana (nombre de la Manzana, Provincia, Municipio, Localidad y Calles asociadas).

RF9.2: Actualizar Manzana (nombre de la Manzana, Calles asociadas).

RF9.3: Eliminar Manzana.

RF10: Imprimir Información de los Municipios, Localidades, Calles, Manzanas.

Requerimientos Funcionales del Registro de Localidades.

RF1: Listar los Consejos Populares del país que se correspondan con los criterios de búsquedas introducidos previamente (Provincia, Municipio, Localidad o Consejo Popular).

RF2: Listar las Circunscripciones del país que se correspondan con los criterios de búsquedas introducidos previamente (Provincia, Municipio, Localidad, Consejo Popular o Circunscripción).

RF3: Listar las Zonas del país que se correspondan con los criterios introducidos previamente (Provincia, Municipio, Localidad, Consejo Popular, Circunscripción o Zona).

RF4: Listar los CDR del país que se correspondan con los criterios introducidos previamente (Provincia, Municipio, Localidad, Consejo Popular, Circunscripción, Zona, CDR o No de CDR).

RF5: Gestionar la información de los Consejos Populares.

RF5.1: Insertar Consejo Popular (nombre del Consejo Popular, Municipio y Localidades que abarca).

RF5.2: Actualizar Consejos Populares (nombre del Consejo Popular y Localidades que abarca).

RF5.3: Eliminar Consejo Popular. No se puede eliminar un Consejo Popular que tenga Circunscripciones asociadas.

RF6: Gestionar la información de las Circunscripciones.

RF6.1: Insertar Circunscripción (nombre de la Circunscripción, Consejo Popular y Municipio al que pertenece).

RF6.2: Actualizar Circunscripción (nombre de la Circunscripción, Consejo Popular al que pertenece).

RF6.3: Eliminar Circunscripción. No se puede eliminar una Circunscripción que tenga Zonas asociadas.

RF7: Gestionar la información de las Zonas.

RF7.1: Insertar Zona (nombre de la Zona, Circunscripción, Consejo Popular y Municipio al que pertenece).

RF7.2: Actualizar Zona (nombre de la Zona, Circunscripción y Consejo Popular al que pertenece).

RF7.3: Eliminar Zona. No se puede eliminar una Zona que tenga CDR asociados.

RF8: Gestionar la información de los CDR.

RF8.1: Insertar CDR (nombre del CDR, Zona, Circunscripción, Consejo Popular, Municipio al que pertenece y Manzanas que abarca).

RF8.2: Actualizar CDR (nombre del CDR, Zona, Circunscripción, Consejo Popular al que pertenece y Manzanas que abarca).

RF8.3: Eliminar CDR.

RF9: Imprimir Información de los Consejos Populares, Circunscripciones, Zonas y CDR.

2.2.1.2 Requerimientos no Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Para el Registro de Ubicación Geográfica y el Registro de Localidades los requerimientos no funcionales son los mismos, ya que ambos registros forman parte del Sistema de Información para la Salud.

Usabilidad.

RNF1: El sistema debe garantizar un acceso fácil y rápido, podrá ser usado por cualquier usuario que posea pocos conocimientos informáticos y de un ambiente Web en sentido general.

Rendimiento.

RNF2: El sistema debe tener una similitud en sus páginas y estar poco cargado, posibilitando que el sistema devuelva las respuestas de una manera eficiente, siendo más sencillo de entender y usar por el usuario.

Soporte.

RNF3: El personal que trabaja con el módulo debe contar con el nivel técnico requerido mediante adiestramiento de servicio.

Portabilidad.

RNF4: Permitir que el sistema se ejecute sobre el Sistema Operativo Linux, Windows 98 o superior.

Seguridad.

RNF5: Disponer de un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoria (AAA).

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

Integridad: Que la información sea modificada (incluyendo su creación y borrado) sólo por personal autorizado. Se permitirá la creación de copias de respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.

Disponibilidad: Los usuarios autorizados tendrán acceso a la información en todo momento, se debe lograr balancear la carga de acceso entre múltiples servidores, disminuyendo los tiempos de respuesta.

Apariencia o Interfaz Externa.

RNF6: La interfaz debe ser sencilla y amigable ya que el usuario no es experto en el uso de las aplicaciones Web.

Ayuda y Documentación en Línea.

RNF7: Disponer de instrucciones en una opción de ayuda.

Software.

RNF8: Los clientes tendrán acceso a ambos registros a través de cualquier navegador Web. Recomendados Mozilla 1.5, Internet Explorer 5.0 o superior. El servidor debe tener PHP Versión 4.3.4 (aunque debe trabajar también con el 4.3.2), Biblioteca PEAR-SOAP 0.8RC3, Módulo XSLT (Sablotron) en PHP, Módulo DBX en PHP, Servidor de Base de Datos MySQL Versión 4 , Servidor HTTP (preferiblemente Apache) , Web Browser que soporte DHTML y CSS2.

Hardware.

RNF9: Requerimientos mínimos:

- Ordenador Pentium o superior.
- 64 MB de Memoria RAM.
- Monitor VGA o superior.
- Teclado y Mouse.
- Procesador 486DX / 66 MHZ o superior.
- Disco duro de 20 GB.
- Impresora de puntos.

- Insumos. (Disquetes, CD RW, Papel continuo y cintas de impresora).
- La PC de trabajo debe estar conectada a una Red de Área Local (LAN).
- Conectividad con el nodo local de INFOMED.

2.2.2 Modelo de Casos de Uso del Sistema.

El modelado de casos de uso es una técnica efectiva y a la vez simple para modelar los requerimientos del sistema desde la perspectiva del usuario. Presenta el sistema desde la perspectiva de su uso y esquematiza cómo proporcionará valor a sus usuarios. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores para limitar las funciones con que dispondrá el sistema luego de ser implementado, además proporciona la entrada fundamental para el análisis, el diseño, la implementación y las pruebas.

El modelo de caso de uso del sistema está compuesto por la definición de actores, los diagramas de caso de uso y la descripción textual de los casos de uso que aparecen a continuación.

2.2.2.1 Definición de actores.

Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

A continuación se presenta los actores del sistema con su justificación, del Registro de Ubicación Geográfica y Registro de Localidades respectivamente.

Actores del sistema	Justificación
Funcionario (Editor)	Es el encargado de la gestión de toda la información del sistema, puede insertar, modificar y eliminar todos los datos, además de hacer las búsquedas y listados que un usuario visualizador puede realizar. El Funcionario está presente en los niveles Nacional, Provincial y Municipal. Generaliza el rol de Funcionario Nacional, que es el único que puede gestionar la información de los Municipios.

Funcionario Nacional	Es el encargado de la gestión de toda la información del sistema, además de hacer las búsquedas y listados que un usuario visualizador puede realizar. Es el único que puede gestionar la información de los Municipios.
Usuario (Visualizador)	Sólo puede listar y hacer búsquedas de toda la información.

Tabla 2.1 Definición de los actores del sistema del Registro de Ubicación Geográfica.

Actores del sistema	**Justificación**
Funcionario (Editor)	Es el encargado de la gestión de toda la información del sistema, puede insertar, modificar y eliminar todos los datos, además de hacer las búsquedas y listados que un usuario visualizador puede realizar. El Funcionario está presente en el nivel Municipal, ya que es el único nivel donde se pueden gestionar los datos.
Usuario (Visualizador)	Sólo puede listar y hacer búsquedas de toda la información.
Registro de Ubicación	Componente que contiene y brinda los datos de las Provincias, Municipios, Localidades, Calles y Manzanas del país.

Tabla 2.2 Definición de los actores del sistema del Registro de Localidades.

2.2.2.2 Diagrama de Casos de Uso.

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas.

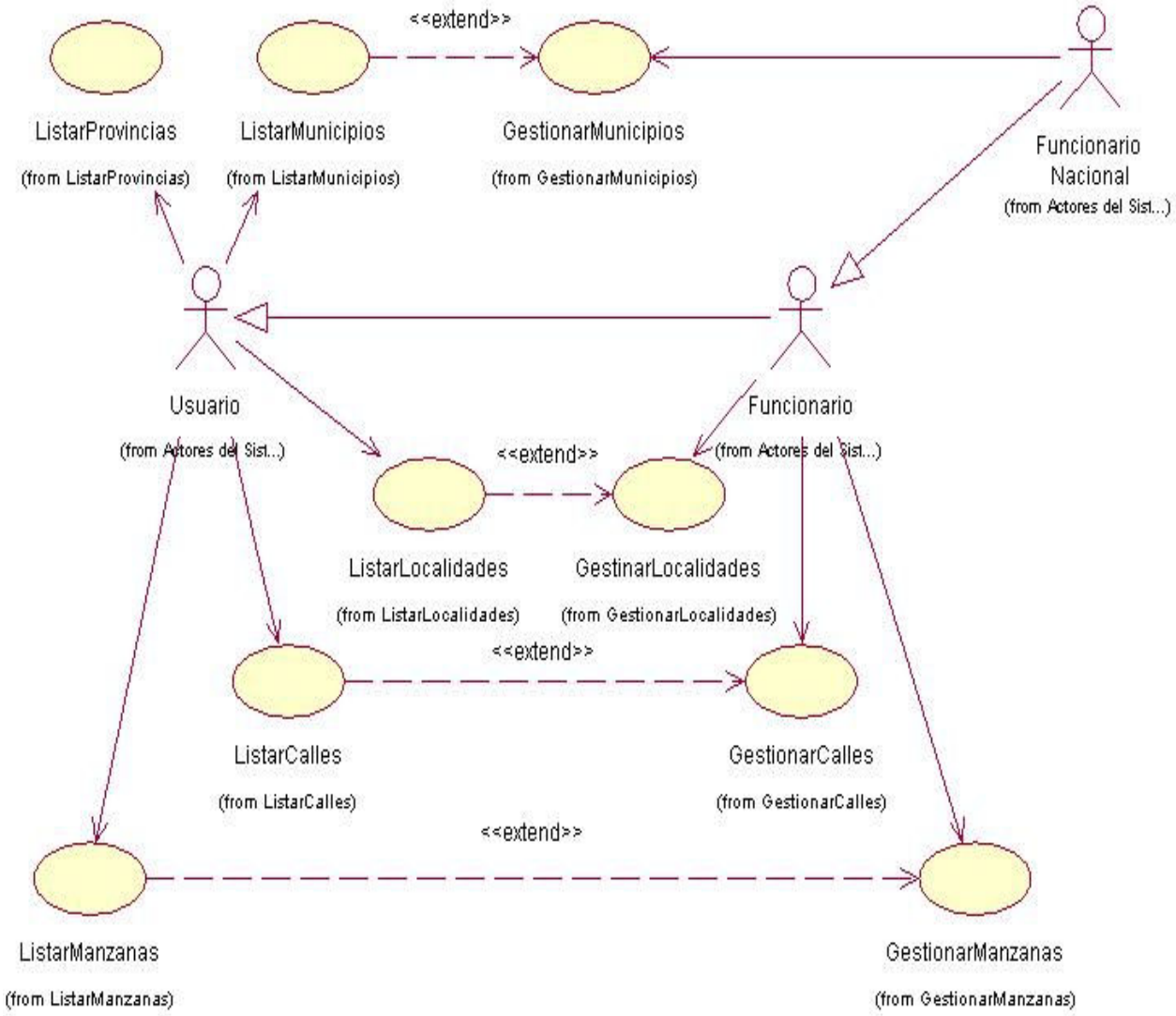


Figura 2.3 Diagrama de casos de uso del Registro de Ubicación.

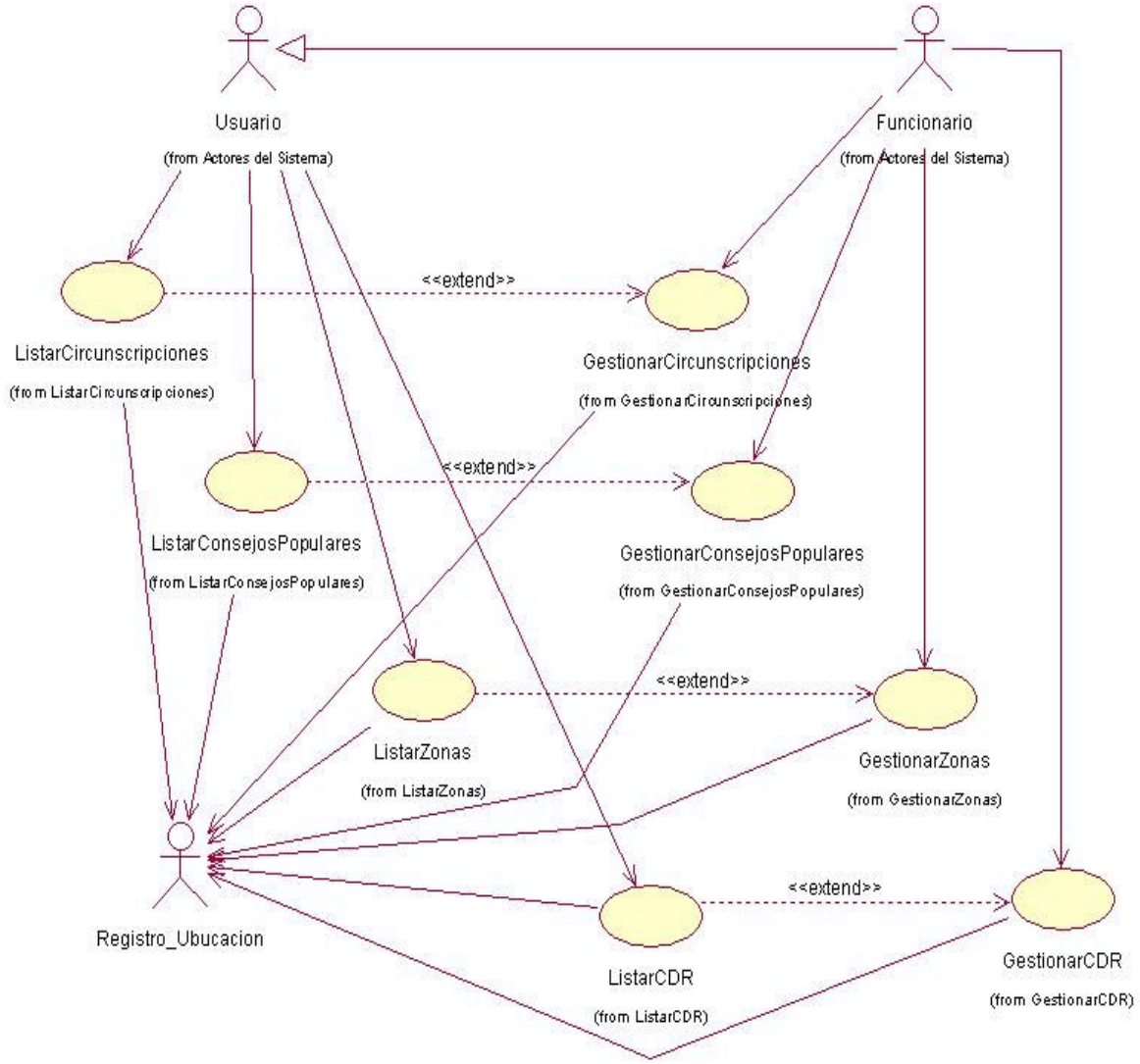


Figura 2.4 Diagrama de casos de uso del Registro de Localidades.

2.2.2.3 Descripción textual de los Casos de Uso.

Descripción textual de los Casos de Uso del Registro de Ubicación Geográfica.

CU-01	ListarProvincias
Actores:	Usuario
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción "Provincias", el sistema muestra en la página el listado de las 15 Provincias con que cuenta el país, culminando así el Caso de Uso.
Requisitos:	**RF1,RF10**
Prioridad:	**Crítico**

Tabla 2.3 Descripción del caso de uso ListarProvincias.

CU-02	ListarMunicipios
Actores:	Usuario
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción "ListarMunicipios", el sistema permite que el usuario introduzca los criterios de selección de los Municipios que desea listar. Una vez introducidos el sistema hace una búsqueda (según estos criterios) y muestra la página, con un listado de los Municipios encontrados, mostrando su nombre y la Provincia a la que pertenece, culminando así el Caso de Uso.
Requisitos:	**RF2,RF10**
Prioridad:	**Crítico**

Tabla 2.4 Descripción del caso de uso ListarMunicipios.

CU-03	ListarLocalidades
Actores:	Usuario
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción "ListarLocalidades", el sistema permite que el usuario introduzca los criterios de selección de las Localidades que desea listar. Una vez introducidos el sistema hace una búsqueda (según estos criterios) y muestra la página, con un listado de las Localidades encontradas, mostrando su nombre, Municipio y la Provincia a la que pertenece, culminando así el Caso de Uso.
Requisitos:	**RF3,RF10**
Prioridad:	**Crítico**

Tabla 2.5 Descripción del caso de uso ListarLocalidades.

CU-04	ListarCalles
Actores:	Usuario
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción "ListarCalles", el sistema permite que el usuario introduzca los criterios de selección de las Calles que desea listar. Una vez introducidos el sistema hace una búsqueda (según estos criterios) y muestra la página, con un listado de las Calles encontradas, mostrando su nombre, Localidad, Municipio y la Provincia a la que pertenece, culminando así el Caso de Uso.
Requisitos:	**RF4,RF10**
Prioridad:	**Crítico**

Tabla 2.6 Descripción del caso de uso ListarCalles.

CU-05	ListarManzanas
Actores:	Usuario
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción "Listar Manzanas", el sistema permite que el usuario introduzca los criterios de selección de las manzanas que desea listar. Una vez introducidos el sistema hace una búsqueda (según estos criterios) y muestra la página, con un listado de las Manzanas encontradas, mostrando su nombre, Localidad, Municipio y la Provincia a la que pertenece, culminando así el Caso de Uso.
Requisitos:	**RF5,RF10**
Prioridad:	**Crítico**

Tabla 2.7 Descripción del caso de uso ListarManzanas.

CU-06	GestionarMunicipios
Actores:	Funcionario Nacional
Descripción:	El caso de uso se inicia cuando el Funcionario Nacional elige en el menú, la opción "Municipios". Al seleccionarla el sistema permite insertar, actualizar y eliminar un Municipio. Para actualizar o eliminar un Municipio primero debe buscarlo (extiende el CU ListarMunicipio). Una vez confirmada la inserción, actualización o eliminación por parte del usuario se actualiza la base de datos y finaliza el Caso de Uso.
Requisitos:	**RF6,RF10**
Prioridad:	**Crítico**

Tabla 2.8 Descripción del caso de uso GestionarMunicipios.

CU-7	GestionarLocalidades
Actores:	Funcionario
Descripción:	El caso de uso se inicia cuando el Funcionario elige en el menú, la opción “Localidades”. Al seleccionarla el sistema permite insertar, actualizar y eliminar una Localidad. Para actualizar o eliminar una Localidad primero debe buscarla (extiende el CU ListarLocalidades). Una vez confirmada la inserción, actualización o eliminación por parte del usuario se actualiza la base de datos y finaliza el Caso de Uso.
Requisitos:	**RF7,RF10**
Prioridad:	**Crítico**

Tabla 2.9 Descripción del caso de uso GestionarLocalidades.

CU-8	GestionarCalles
Actores:	Funcionario
Descripción:	El caso de uso se inicia cuando el Funcionario elige en el menú, la opción “Calles”. Al seleccionarla el sistema permite insertar, actualizar y eliminar una Calle. Para actualizar o eliminar una Calle primero debe buscarla (extiende el CU ListarCalles). Una vez confirmada la inserción, actualización o eliminación por parte del usuario se actualiza la base de datos y finaliza el Caso de Uso.
Requisitos:	**RF8,RF10**
Prioridad:	**Crítico**

Tabla 2.10 Descripción del caso de uso GestionarCalles.

CU-9	GestionarManzanas
Actores:	Funcionarios
Descripción:	El caso de uso se inicia cuando el Funcionario elige en el menú, la opción “Manzanas”. Al seleccionarla el sistema permite insertar, actualizar y eliminar una Manzana. Para actualizar o eliminar una Manzana primero debe buscarla (extiende el CU ListarManzanas). Una vez confirmada la inserción, actualización o eliminación por parte del usuario se actualiza la base de datos y finaliza el Caso de Uso.
Requisitos:	**RF9, RF10.**
Prioridad:	**Crítico**

Tabla 2.11 Descripción del caso de uso GestionarManzanas.

Descripción textual de los Casos de Uso del Registro de Localidades.

CU-11	ListarConsejosPopulares
Actores:	Usuario.
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción “Consejos Populares”, el sistema permite que el usuario introduzca los criterios de selección de los Consejos Populares que desea listar. Una vez introducidos, el sistema hace una búsqueda (según estos criterios) y muestra la página, con un listado de los Consejos Populares, mostrando su nombre y Municipio al que pertenece, culminando así el Caso de Uso.
Requisitos:	**RF1,RF9**
Prioridad:	**Crítico**

Tabla 2.12 Descripción del caso de uso ListarConsejosPopulares.

CU-12	ListarCircunscripción
Actores:	Usuario.
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción “Circunscripciones”, el sistema permite que el usuario introduzca los criterios de selección de la Circunscripción que desea listar. Una vez introducidos el sistema hace una búsqueda (según estos criterios) y muestra la página, con un listado de las Circunscripciones, mostrando su nombre, Consejo Popular y Municipio al que pertenece, culminando así el Caso de Uso.
Requisitos:	**RF2,RF9**
Prioridad:	**Crítico**

Tabla 2.13 Descripción del caso de uso ListarCircunscripción.

CU-13	ListarZona
Actores:	Usuario.
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción “Zona”, el sistema permite que el usuario introduzca los criterios de selección de las Zonas que desea listar. Una vez introducidos el sistema hace una búsqueda (según estos criterios) y muestra la página, con un listado de las Zonas, mostrando su nombre, Circunscripción, Consejo Popular y Municipio al que pertenece, culminando así el Caso de Uso.
Requisitos:	**RF3,RF9**
Prioridad:	**Crítico**

Tabla 2.14 Descripción del caso de uso ListarZona.

CU-14	ListarCDR
Actores:	Usuario.
Descripción:	El Caso de Uso comienza cuando el usuario selecciona en el menú, la opción "CDR", el sistema permite que el usuario introduzca los criterios de selección de los CDR que desea listar. Una vez introducidos el sistema hace una búsqueda (según estos criterios) y muestra la página, con un listado de los CDR, mostrando el número y el nombre del CDR así como la Zona, Circunscripción, Consejo Popular y Municipio al que pertenece, culminando así el Caso de Uso.
Requisitos:	**RF4,RF9**
Prioridad:	**Crítico**

Tabla 2.15 Descripción del caso de uso ListarCDR.

CU-15	GestionarConsejosPopulares
Actores:	Funcionario Municipal
Descripción:	El caso de uso se inicia cuando el Funcionario Municipal elige en el menú, la opción "Consejos Populares". Al seleccionarla el sistema permite insertar, actualizar y eliminar un Consejo Popular. Para actualizar o eliminar un Consejo Popular primero debe buscarlo (extiende el CU ListarConsejosPopulares). Una vez confirmada la inserción, actualización o eliminación por parte del usuario se actualiza la base de datos y finaliza el Caso de Uso.
Requisitos:	**RF5,RF9**
Prioridad:	**Crítico**

Tabla 2.16 Descripción del caso de uso GestionarConsejosPopulares.

CU-16	GestionarCircunscripción
Actores:	Funcionario Municipal
Descripción:	El caso de uso se inicia cuando el Funcionario Municipal elige en el menú, la opción "Circunscripción". Al seleccionarla el sistema permite insertar, actualizar y eliminar una Circunscripción. Para actualizar o eliminar una Circunscripción primero debe buscarla (extiende el CU ListarCircunscripción). Una vez confirmada la inserción, actualización o eliminación por parte del usuario se actualiza la base de datos y finaliza el Caso de Uso.
Requisitos:	**RF6,RF9**
Prioridad:	**Crítico**

Tabla 2.17 Descripción del caso de uso GestionarCircunscripción.

CU-17	GestionarZona
Actores:	Funcionario Municipal
Descripción:	El caso de uso se inicia cuando el Funcionario Municipal elige en el menú, la opción “Zona”. Al seleccionarla el sistema permite insertar, actualizar y eliminar una Zona. Para actualizar o eliminar una Zona primero debe buscarla (extiende el CU ListarZona). Una vez confirmada la inserción, actualización o eliminación por parte del usuario se actualiza la base de datos y finaliza el Caso de Uso.
Requisitos:	**RF7,RF9**
Prioridad:	**Crítico**

Tabla 2.18 Descripción del caso de uso GestionarZona.

CU-18	GestionarCDR
Actores:	Funcionario Municipal
Descripción:	El caso de uso se inicia cuando el Funcionario Municipal elige en el menú, la opción “CDR”. Al seleccionarla el sistema permite insertar, actualizar y eliminar un CDR. Para actualizar o eliminar un CDR primero debe buscarlo (extiende el CU ListarCDR). Una vez confirmada la inserción, actualización o eliminación por parte del usuario se actualiza la base de datos y finaliza el Caso de Uso.
Requisitos:	**RF8,RF9**
Prioridad:	**Crítico**

Tabla 2.19 Descripción del caso de uso GestionarCDR.

Conclusiones.

En el capítulo se presentaron las características del sistema a desarrollar. Empleando un modelo de dominio se han definido los principales conceptos del entorno y sus relaciones, contribuyendo a la comprensión del contexto del sistema. Se describieron las condiciones o capacidades que debe cumplir el sistema. También se describieron y justificaron los actores del sistema así como el diagrama de casos de uso y la descripción textual de los casos de uso.

Capítulo 3: Diseño del Sistema.

Introducción.

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

En el presente capítulo se realiza el diseño de la solución propuesta. Se realiza la justificación del uso de patrones y subsistemas de servicios. Se presentan los diagramas de clases de diseño y se describen las clases y atributos. Además se presentan los diagramas de clases persistentes, los modelos de datos y las descripciones de las tablas y atributos del Registro de Ubicación Geográfica y el Registro de Localidades.

3.1 Modelo de Diseño.

El modelo de diseño permite adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario. Crea una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

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

3.1.1 Justificación del uso de Patrones.

Un patrón es un modelo que se puede seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar lograr ciertos objetivos promoviendo buenas prácticas. Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular.

Los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo. Por tanto, están basados en la recopilación del conocimiento de los expertos en desarrollo de software. Son una experiencia real, probada y que funciona.

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

Patrón Fachada

El patrón Fachada se utiliza para proveer de una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. Provee una solución de software en capas y simplifica la interfaz entre dos sistemas o componentes.

En el sistema desarrollado se utiliza para abstraer la capa de presentación de interactuar con las clases de PlaSer, este patrón puede ser útil cuando se decida utilizar otra forma de interactuar con Servicios Web que no sea a través de la librería PlaSer, como por ejemplo en la migración a php5.

Patrón Proxy

El patrón Proxy se utiliza como intermediario para acceder a un objeto, permitiendo controlar el acceso a él. El proxy guarda una referencia a un objeto, que es quien realmente se encargará de ejecutar la acción. El proxy crea el objeto, por lo que puede crear una instancia de una u otra clase dependiendo de ciertas condiciones en tiempo de ejecución.

En el sistema fue implementado para que controle todas las peticiones en un único punto y esto sirve para hacer chequeos que garantizan la seguridad y el tratamiento de la integridad referencial.

Patrón Alta Cohesión.

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

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

Patrón Bajo Acoplamiento

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

El patrón Bajo Acoplamiento permite que al realizar cambios en un componente no se afecten otros. Son diseños fáciles de entender por separado y fáciles de reutilizar.

3.1.2 Estructura del Diseño.

La estructura del diseño permite ver la relación de los subsistemas de diseño dentro de la arquitectura definida. Dividir el diseño de un sistema en subsistemas de diseño da la posibilidad de organizar el modelo de diseño en porciones más manejables. A continuación se presenta de forma gráfica los subsistemas de diseño.

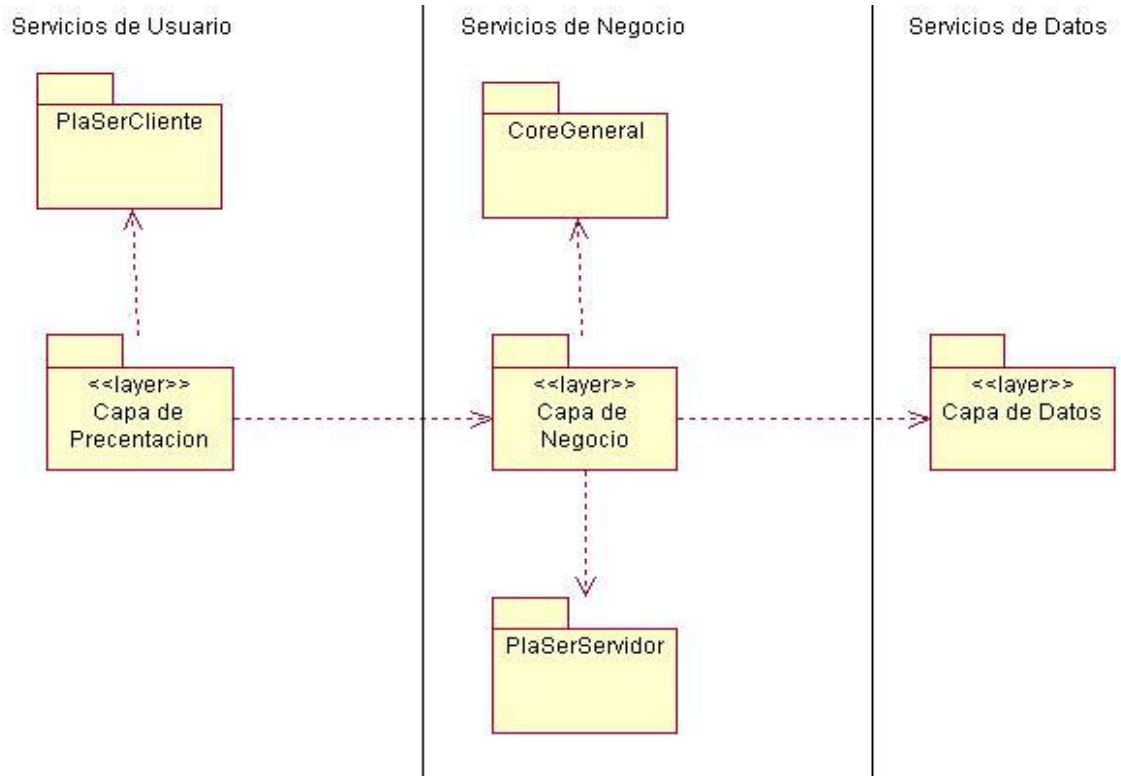


Figura 3.1 Diagrama de Subsistemas de Servicios.

A continuación se describen los subsistemas de servicios:

Capa de Presentación: Contiene 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.

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.

Capa de Datos: Contiene las tablas de la base de datos del Registro de Ubicación Geográfica y el Registro de Localidades, las que son gestionadas por la capa de negocio.

Core General: Este paquete contiene los ficheros de validación utilizados por la capa de negocio. También contiene las clases que generan ficheros en formato Portable Document Format (pdf) o Microsoft Office Excel (xls) para la impresión de información.

PlaSerWeb: Este paquete contiene la clase CFachada de la cual hereda la clase FachadaRL ubicada en la capa de presentación del módulo Registro de Localidades, la misma es la encargada de transformar las páginas XSL en HTML mediante la función transforma, para esto incluye las clases PlaSer_xml, PlaSer_xslt y de acceder al negocio incluyendo a PlaSer_Client.

PlaSerCore: Este paquete contiene la clase de acceso a datos DBz_class y la clase encargada de brindar la dirección física de los componentes, PlaSer_Server, la que incluye para su funcionamiento al fichero confplaser_server.

3.1.3 Definición de Elementos de Diseño.

La extensión UML para Web presenta como elementos más significativos a 3 clases de UML estereotipadas con los siguientes estereotipos “Server Page”, “Client Page”, “Form” empleados para el código servidor, código cliente y formularios respectivamente. [37]

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

<<Client Page>>: Es una página Web con formato XHTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Cada página cliente es construida por una sola página de servidor.

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

<<Form>>: Es una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario. No tienen operaciones y se comunican con las páginas servidores mediante submit.

Las relaciones posibles a establecerse entre los tres elementos claves son:

Hasta / Desde	**Client Page**	**Form**	**Server Page**
Client Page	<<Link>> , <<redirect>>	---	<<redirect>>
Form	---	---	<<Submit>>
Server Page	<<Build>>, <<redirect>>		<<Redirect>>. <<Call>>, <<include>>

Tabla 3.1 Relaciones entre las clases principales que conforman la extensión de UML para Web.

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

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

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

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

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

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

3.1.4 Diagrama de Clases del Diseño.

Un diagrama de clases es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones. Gráficamente, un diagrama de clases es una colección de nodos y arcos. Contiene clases, interfaces, colaboraciones y relaciones de dependencia, generalización y asociación. Los diagramas de clases brindan un mayor acercamiento a la forma y al contenido de la solución propuesta. El Registro de Ubicación Geográfica y el Registro de Localidades son aplicaciones web, para realizar estos diagramas de clases se tuvo en cuenta la extensión UML para Web.

A continuación se presentan algunos de estos diagramas de clases, se escogieron aquellos que por sus características reúnen todas las funcionalidades del sistema diseñado:

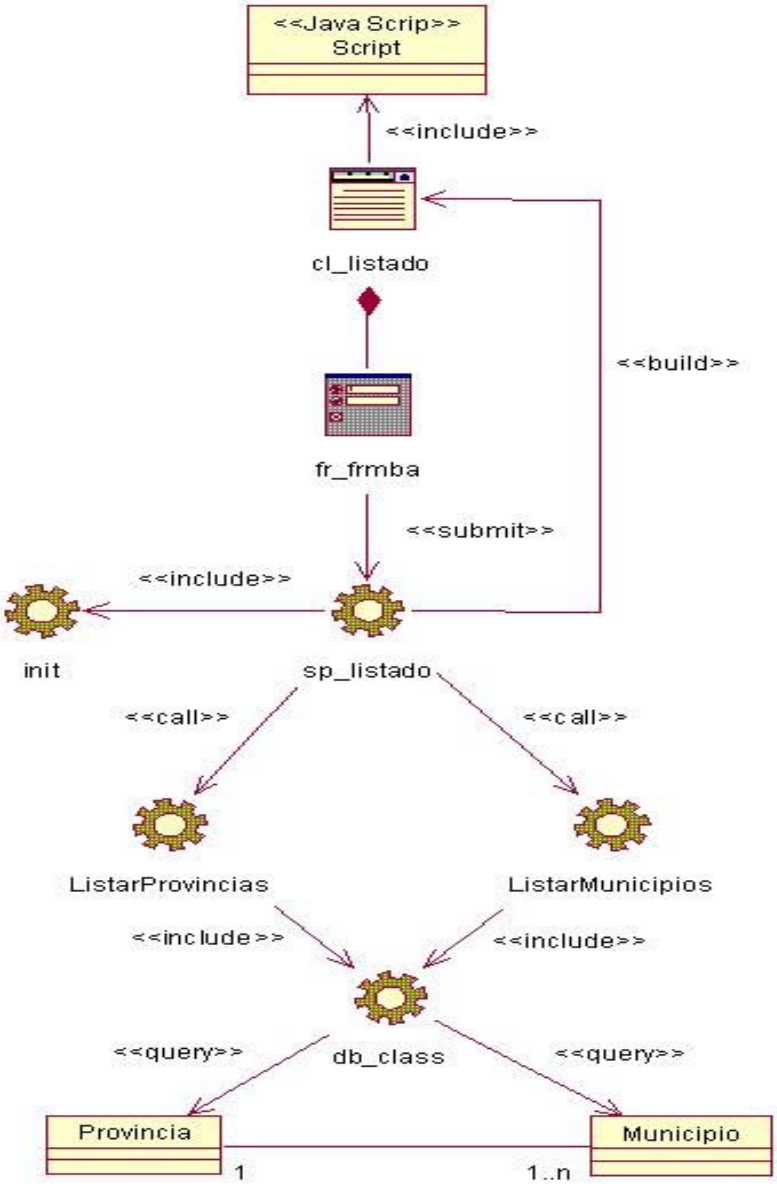


Figura 3.2 Diagrama de Clases del Diseño del CU-02: ListarMunicipios.

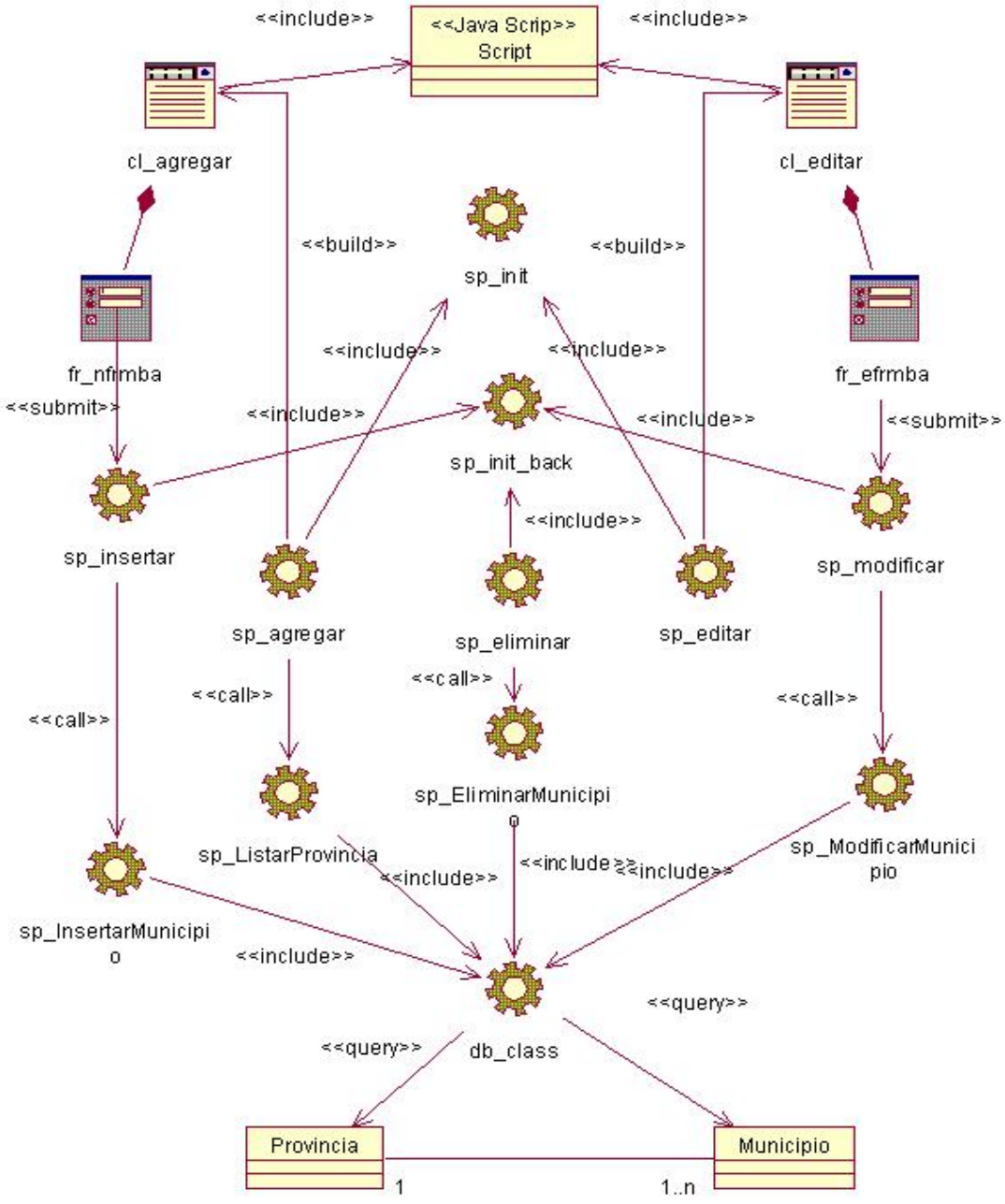


Figura 3.3 Diagrama de Clases del Diseño del CU-06: GestionarMunicipios.

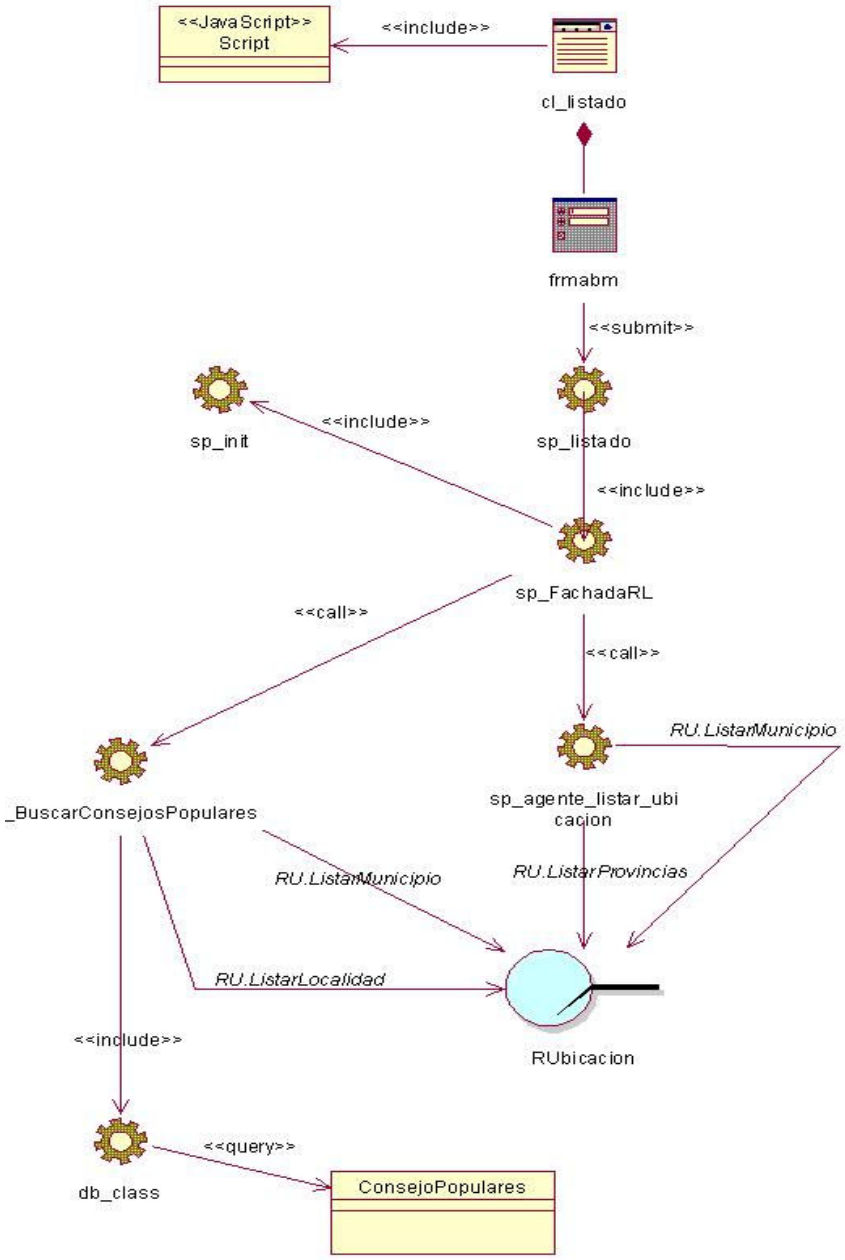


Figura 3.4 Diagrama de Clases del Diseño del CU-11: ListarConsejosPopulares.

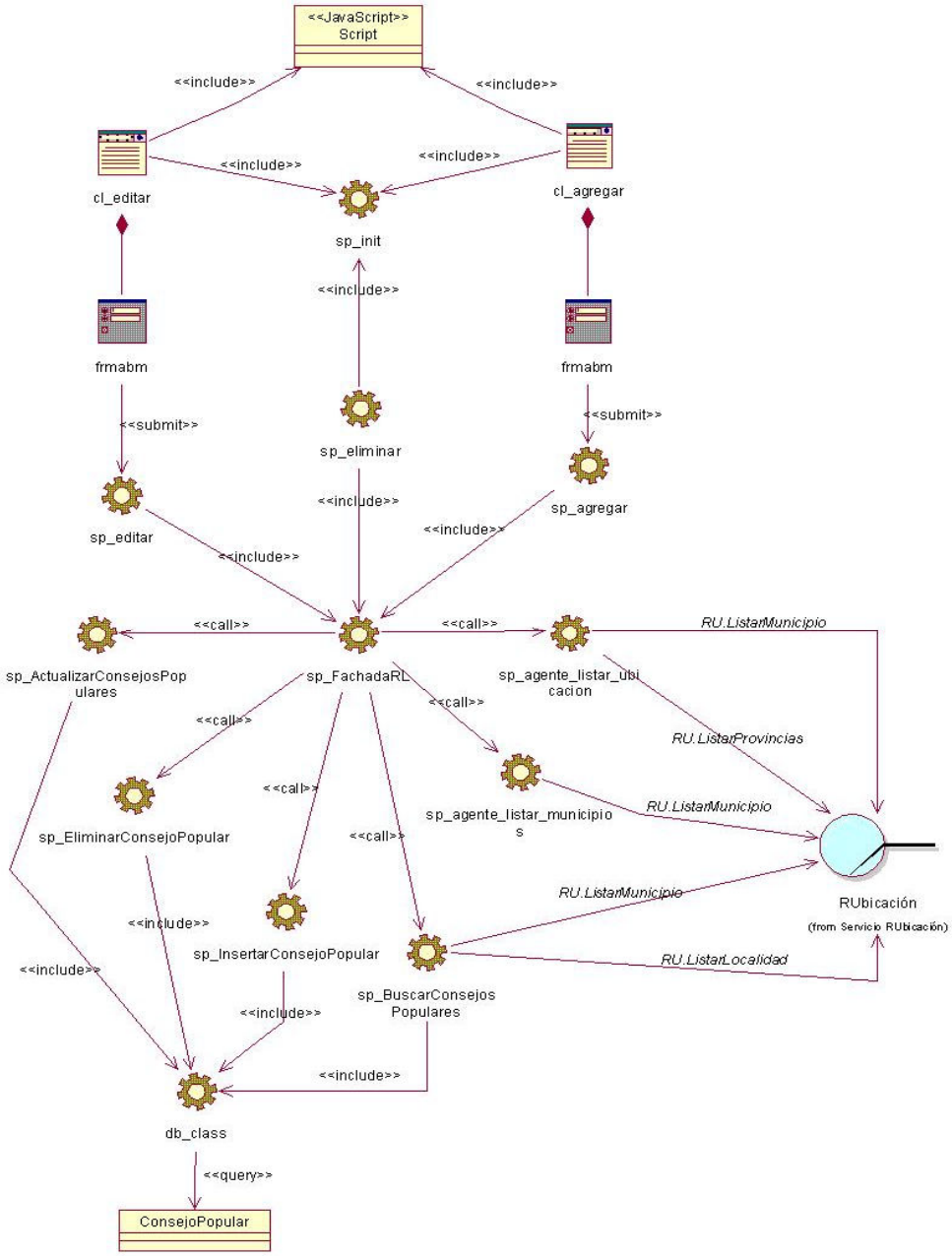


Figura 3.5 Diagrama de Clases del Diseño del CU-15: GestionarConsejosPopulares.

3.1.5 Descripción de las clases y atributos.

A continuación se describen las clases del diseño representadas en los diagramas mostrados anteriormente. Se separan las clases servidoras de los Registro de Ubicación Geográfica y el Registro de Localidades porque el primero usa la clase servidora Fachada y el segundo no.

Páginas cliente.

Nombre: cl_listado
Tipo de clase: página cliente
Descripción General: La clase *cl_listado* es una página web que se ejecuta del lado del cliente sobre un browser o navegador web. Permite a los visualizadores o editores nacionales, provinciales y municipales la realización de listado y búsquedas de información específica permitiendo generar documentos Portable Document Format (.pdf) y Microsoft Office Excel (.xls) para imprimir. A través de esta página se puede visualizar la información. El resultado de las búsquedas es paginada, permitiendo la movilidad por las diferentes páginas, inclusive ir directamente a la última. Es utilizada en los siguientes casos de uso: • ListarMunicipios. • ListarConsejosPopulares.

Tabla 3.2 Descripción página cliente cl_listado.

Nombre: cl_editar
Tipo de clase: página cliente
Descripción General: La clase *cl_editar* es una página web que se ejecuta del lado del cliente sobre un browser o navegador web. Recibe los valores de la página cliente *cl_listado* .Permite a los editores nacionales, provinciales y municipales modificar o visualizar información. • GestionarMunicipios. • GestionarConsejosPopulares.

Tabla 3.3 Descripción página cliente cl_editar.

Nombre: cl_agregar
Tipo de clase: página cliente
Descripción General: La clase *cl_agregar* es una página web que se ejecuta del lado del cliente sobre un browser o navegador web. Permite a los editores nacionales, provinciales y municipales insertar información. • GestionarMunicipios. • GestionarConsejosPopulares.

Tabla 3.4 Descripción página cliente cl_nuevo.

Páginas servidoras del Registro de Ubicación Geográfica.

Nombre: sp_listado
Tipo de clase: página servidora
Descripción General: La clase *sp_listado* es una clase que se ejecuta del lado del servidor en la capa de presentación. Su responsabilidad es construir la página cliente *cl_listado*. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML para posteriormente recibir los parámetros de búsqueda. Es utilizada en el siguiente caso de uso: • ListarMunicipio.

Tabla 3.5 Descripción página servidora sp_listado.

Nombre: sp_agregar
Tipo de clase: página servidora
Descripción General: La clase *sp_agregar* es una clase que se ejecuta del lado del servidor en la capa de presentación. Su responsabilidad es construir la página cliente *cl_agregar*. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML para posteriormente recibir los valores introducidos. Es utilizada en el siguiente caso de uso: • GestionarMunicipios.

Tabla 3.6 Descripción página servidora sp_agregar.

Nombre: sp_insertar
Tipo de clase: página servidora
Descripción General: Recibe los valores de la clase sp_agregar, los valida y los envía hacia la capa de negocio al método de inserción de los datos involucrados y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase *sp_insertar*. Es utilizada en el siguiente caso de uso: • GestionarMunicipios.

Tabla 3.7 Descripción página servidora sp_insertar.

Nombre: sp_eliminar
Tipo de clase: página servidora
Descripción General: La clase sp*_eliminar* es una clase que se ejecuta del lado del servidor en la capa de presentación. Su responsabilidad es recibir los valores de la página cliente *cl_listado* y posteriormente enviarlos hacia la capa de negocio al método del negocio para eliminar los datos involucrados. Una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase *sp_eliminar.* Es utilizada en el siguiente caso de uso: • GestionarMunicipios.

Tabla 3.8 Descripción página servidora sp_eliminar.

Nombre: sp_editar
Tipo de clase: página servidora
Descripción General: La clase *sp_editar* es una clase que se ejecuta del lado del servidor en la capa de presentación. Su responsabilidad es construir la página cliente *cl_editar*. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML para posteriormente recibir los valores modificados. Es utilizada en el siguiente caso de uso: • GestionarMunicipios.

Tabla 3.9 Descripción página servidora sp_editar.

Nombre: sp_modificar
Tipo de clase: página servidora
Descripción General: Recibe los valores de la clase sp_editar, los valida y los envía hacia la capa de negocio al método de modificación de los datos involucrados. Una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase *sp_modificar*. Es utilizada en el siguiente caso de uso: • GestionarMunicipios.

Tabla 3.10 Descripción página servidora sp_modificar.

Nombre: sp_ListarUbicación	
Tipo de clase: página servidora	
Descripción General: La clase *sp_ListarUbicación* se ejecuta del lado del servidor en la capa de negocio. Su función es redireccionar a la clase servidora sp_ListarProvincias o sp_ListarMunicipio en dependencia de los parámetros introducidos.	
Parámetros de Entrada:	
Nivel	int
Profundidad	int
offset	int
cantidad	int
arregloid	array

Tabla 3.11 Descripción página servidora sp_ListarUbicación.

Nombre: sp_ListarProvincias	
Tipo de clase: página servidora	
Descripción General: La clase *sp_ListarProvincias* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es devolver al usuario un listado de las Provincias, Municipios y Localidades según los parámetros de entrada.	
Parámetros de Entrada:	
Profundidad	int
offset	int
cantidad	int
arregloid	array

Tabla 3.12 Descripción página servidora sp_ListarProvincias.

Nombre: sp_ListarMunicipio	
Tipo de clase: página servidora	
Descripción General: La clase *sp_ListarMunicipio* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es devolver al usuario un listado de los Municipios y Localidades según los parámetros de entrada.	
Parámetros de Entrada:	
Profundidad	int
offset	int
cantidad	int
tipo_salida	int
arregloidprov	array
arregloid	array

Tabla 3.13 Descripción página servidora sp_ListarMunicipio.

Nombre: sp_ModificarMunicipio	
Tipo de clase: página servidora	
Descripción General: La clase *sp_ModificarMunicipio* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es modificar en la base de datos un Municipio según los parámetros de entrada.	
Parámetros de Entrada:	
municipio	string
id_municipio	int

Tabla 3.14 Descripción página servidora sp_ModificarMunicipio.

Nombre: sp_EliminarMunicipio	
Tipo de clase: página servidora	
Descripción General: La clase *sp_EliminarMunicipio* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es eliminar de la base de datos un Municipio según el parámetro de entrada.	
Parámetros de Entrada:	
id_municipio	int

Tabla 3.15 Descripción página servidora sp_EliminarMunicipio.

Páginas servidoras del Registro de Localidades

Nombre: sp_listado
Tipo de clase: página servidora
Descripción General: La clase *sp_listado* es una clase que se ejecuta del lado del servidor en la capa de presentación. Su responsabilidad es construir la página cliente *cl_listado*. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML para posteriormente recibir los parámetros de búsqueda. Es utilizada en el siguiente caso de uso: • ListarConsejosPopulares.

Tabla 3.16 Descripción página servidora sp_listado.

Nombre: sp_agregar
Tipo de clase: página servidora
Descripción General: La clase *sp_agregar* es una clase que se ejecuta del lado del servidor en la capa de presentación. Su responsabilidad es construir la página cliente *cl_agregar*. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML para posteriormente recibir los valores introducidos. Es utilizada en el siguiente caso de uso: • GestionarConsejosPopulares.

Tabla 3.17 Descripción página servidora sp_agregar.

Nombre: sp_eliminar
Tipo de clase: página servidora
Descripción General: La clase sp_*eliminar* es una clase que se ejecuta del lado del servidor en la capa de presentación. Su responsabilidad es recibir los valores de la página cliente *cl_listado* para posteriormente enviarlos hacia la capa de negocio al método del negocio para eliminar los datos involucrados. Una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a la clase *sp_eliminar.* Es utilizada en el siguiente caso de uso: • GestionarConsejosPopulares.

Tabla 3.18 Descripción página servidora sp_eliminar.

Nombre: sp_editar
Tipo de clase: página servidora
Descripción General: La clase *sp_editar* es una clase que se ejecuta del lado del servidor en la capa de presentación. Su responsabilidad es construir la página cliente *cl_editar*. Aplica un documento (XSL) a otro documento XML para transformarlo y mostrarlo al usuario en formato XHTML para posteriormente recibir los valores modificados. Es utilizada en el siguiente caso de uso: • GestionarConsejosPopulares.

Tabla 3.19 Descripción página servidora sp_editar.

Nombre: sp_ActualizarConsejosPopulares	
Tipo de clase: página servidora	
Descripción General: La clase *sp_ActualizarConsejosPopulares* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es modificar en la base de datos un Consejo Popular según los parámetros de entrada.	
Parámetros de Entrada:	
id_localidad	array
descripción	string
id_consejo	int

Tabla 3.20 Descripción página servidora sp_ActualizarConsejosPopulares.

Nombre: sp_EliminarConsejoPopular	
Tipo de clase: página servidora	
Descripción General: La clase *sp_EliminarConsejoPopular* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es eliminar en la base de datos un Consejo Popular según el parámetro de entrada.	
Parámetros de Entrada:	
id_consejo	int

Tabla 3.21 Descripción página servidora sp_EliminarConsejoPopular.

Nombre: sp_InsertarConsejoPopular	
Tipo de clase: página servidora	
Descripción General: La clase *sp_InsertarConsejoPopular* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es insertar en la base de datos un Consejo Popular determinado con los datos pasados como parámetros.	
Parámetros de Entrada:	
descripción	string
id_localidad	array
municipio	int

Tabla 3.22 Descripción página servidora sp_InsertarConsejoPopular.

Nombre: sp_BuscarConsejosPopulares	
Tipo de clase: página servidora	
Descripción General: La clase *sp_BuscarConsejosPopulares* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es realizar una búsqueda en la base de datos sobre los datos de un determinado Consejo Popular según parámetros introducidos previamente.	
Parámetros de Entrada:	
id_consejo	Int ó array
descripción	string
id_localidad	array
municipio	int
offset	int
cantidad	int
ordenar_por	string
ordena	string
todos	int
limit	int
tipo_salida	int

Tabla 3.23 Descripción página servidora sp_BuscarConsejosPopulares.

Nombre: sp_agente_listar_municipios	
Tipo de clase: página servidora	
Descripción General: La clase *sp_agente_listar_municipios* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es brindar información sobre los Municipios y sus Localidades correspondientes, según una Provincia determinada. Para obtener estos datos se invoca al método ListarMunicipio perteneciente al Registro de Ubicaciones.	
Parámetros de Entrada:	
idpro	Int

Tabla 3.24 Descripción página servidora sp_agente_listar_municipios.

Nombre: sp_agente_listar_ubicacion	
Tipo de clase: página servidora	
Descripción General: La clase *sp_agente_listar_ubicacion* es una clase que se ejecuta del lado del servidor en la capa de negocio. Su función es brindar información sobre las Provincias según el parámetro de entrada. Si el parámetro de entrada tiene el valor 2 esta clase brinda información sobre las Provincias y sus Municipios. Si este parámetro tiene valor 3 se brinda información acerca de las Provincias, sus Municipios y las Localidades correspondientes a cada Municipio. Para esto se invocan los métodos ListarProvincias y ListarMunicipio, ambos pertenecientes al Registro de Ubicación.	
Parámetros de Entrada:	
profundidad	int

Tabla 3.25 Descripción página servidora sp_agente_listar_municipios.

Otras páginas servidoras.

Clase	Propósito
sp_init	Clase encargada de la seguridad. Extiende clases de la librería PlaSer como: client, xml, xls y utiles_gui.
sp_init_back	Verifica los derechos del usuario. Extiende clases de la librería PlaSer como: client y utiles_gui.

sp_dbz_class	Clase que realiza la conexión con las bases de datos MySql, usa el modulo dbx de PHP para su funcionalidad. Además crea un objeto conexión que permite hacer consultas, y recuperar los resultados; insertar, eliminar y actualizar datos. Esta clase se encuentra en la capa de negocio, la misma está dentro de la librería PlaSer.
sp_fachada	Clase que hereda de Fachada, que aplica una fachada en la capa de presentación, disminuyendo así la carga de negocio en gran medida. Esta clase implementa el patrón Fachada, de esta forma la aplicación solo le hará las peticiones a esta clase.
sp_fachada_rl	La clase *sp_fachada_rl* es una clase que se ejecuta del lado del servidor en la capa de presentación del Registro de Localidades**,** disminuyendo así la carga de negocio en gran medida, esta clase hereda de Fachada. La misma implementa el patrón Fachada, de esta forma la aplicación sólo le hará las peticiones a la clase Fachada. Es la clase principal de la capa de presentación, contiene las funcionalidades del sistema que permiten acceder a los métodos del negocio. Cualquier solución general del módulo debe implementarse en ella.

Tabla 3.26 Descripción de otras páginas servidoras.

3.1.6 Diagrama de clases persistentes.

En el diagrama de clases persistentes sólo aparecen las clases persistentes. Estas son las clases capaces de mantener su valor en el espacio y en el tiempo. A continuación se muestra el diagrama de clases persistentes del Registro de Ubicación Geográfica y el Registro de Localidades respectivamente:

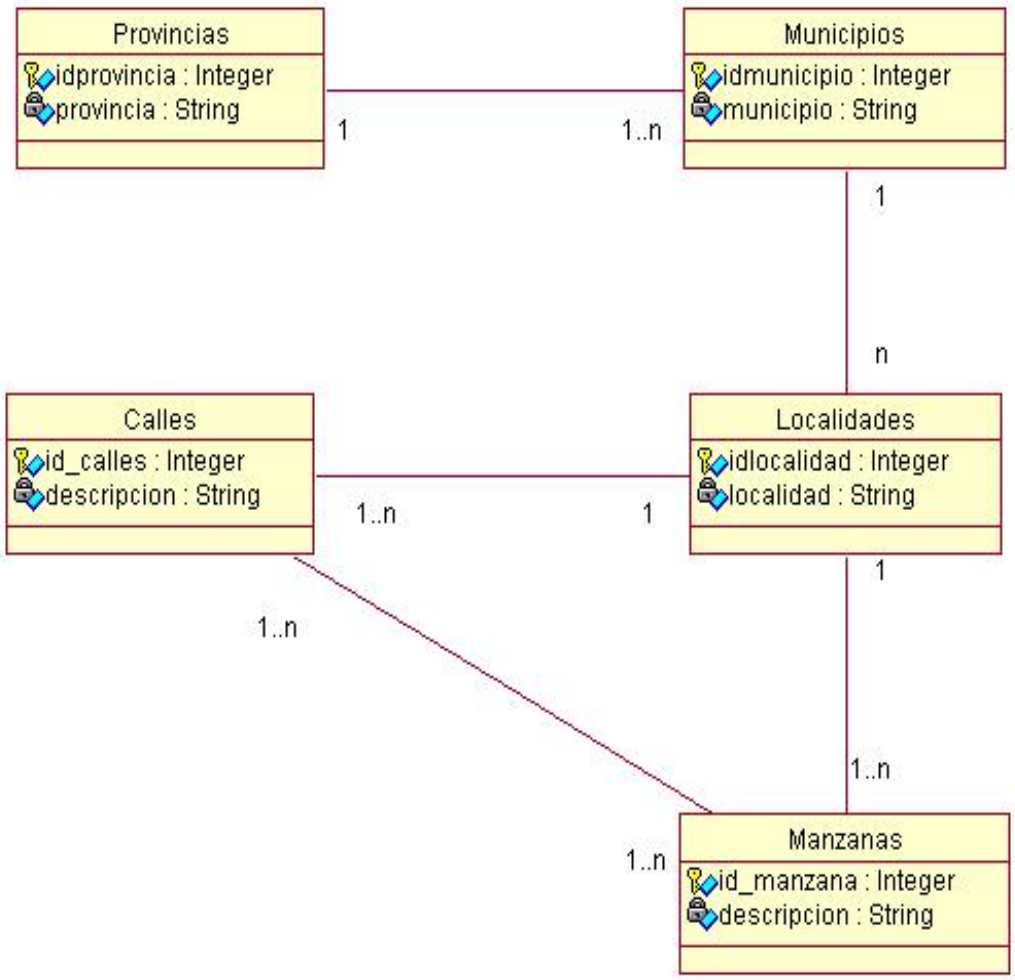


Figura 3.6 Diagrama de Clases Persistes del Registro de Ubicación Geográfica.

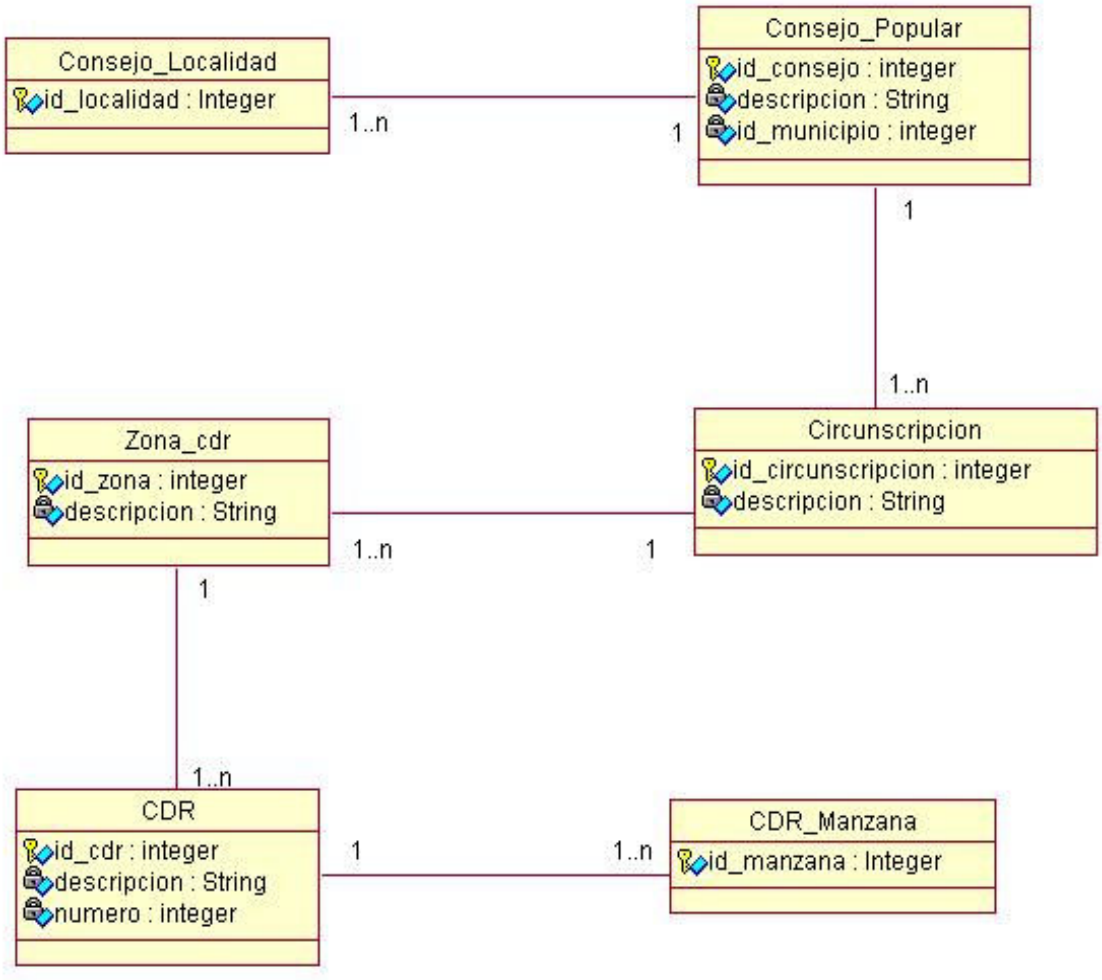


Figura 3.7 Diagrama de Clases Persistes del Registro de Localidades.

3.1.7 Modelo de datos.

El modelo de datos describe la representación lógica y física de datos persistentes en el sistema, es generado a partir del diagrama de clases persistentes. A continuación se muestra el modelo de datos del Registro de Ubicación Geográfica y el Registro de Localidades respectivamente:

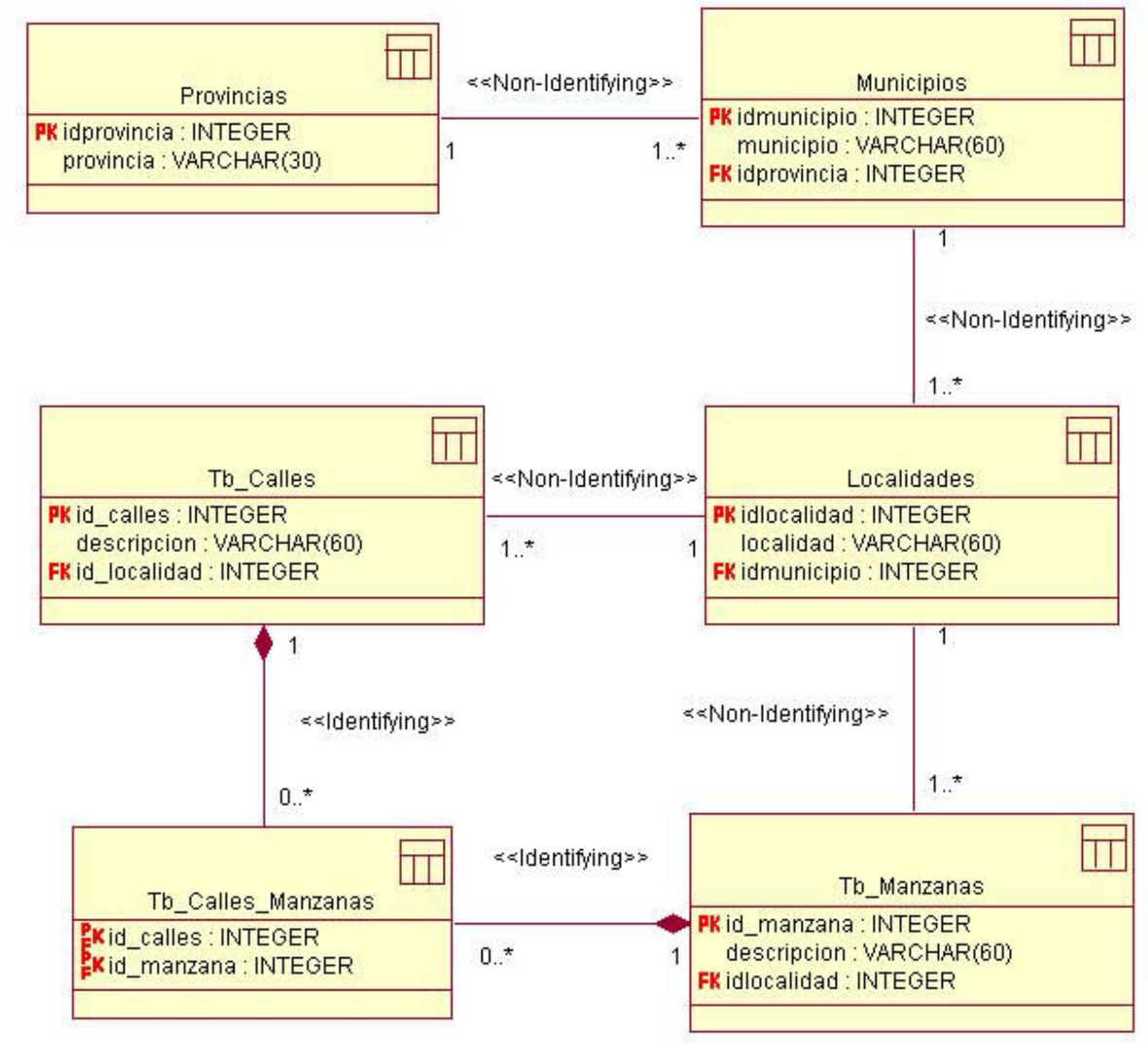


Figura 3.8 Diagrama de Modelo de Datos del Registro de Ubicación Geográfica

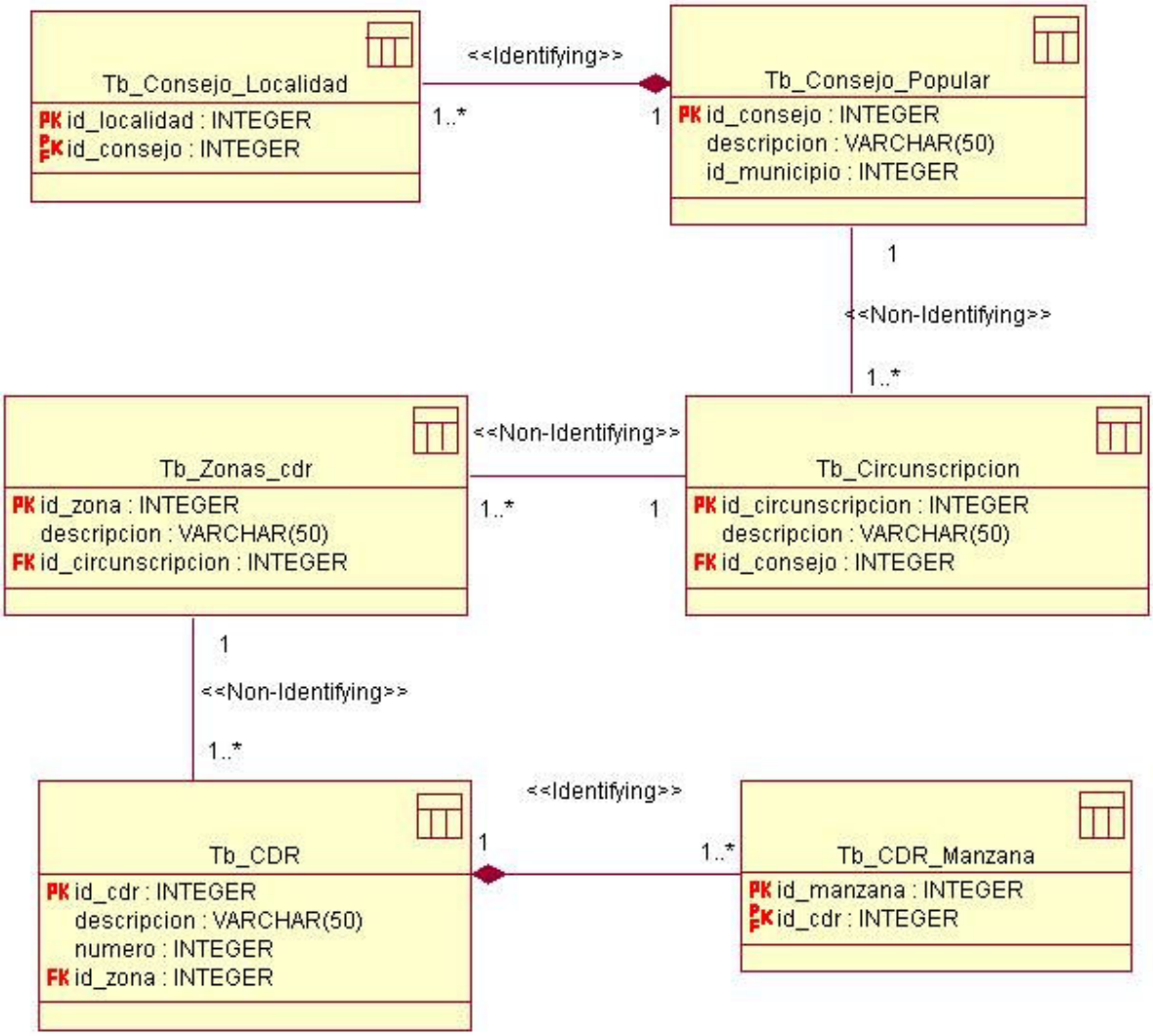


Figura 3.9 Diagrama de Modelo de Datos del Registro de Localidades.

3.1.7 Descripción de tablas y atributos.

Descripción de las tablas de la Base de Datos del Registro de Ubicación Geográfica.

Nombre: Provincias		
Descripción: Almacena las Provincias del país.		
Atributo	**Tipo**	**Descripción**
idprovincia	Integer	Identificador de la Provincia.(PK)
provincia	Varchar(30)	Nombre de la Provincia.

Tabla 3.27 Descripción de la tabla Provincias.

Nombre: Municipios		
Descripción: Almacena los Municipios del país.		
Atributo	**Tipo**	**Descripción**
idmunicipio	Integer	Identificador del Municipio.(PK)
municipio	Varchar(60)	Nombre del Municipio.
idprovincia	Integer	Identificador de la Provincia en la tabla Provincias.(FK)

Tabla 3.28 Descripción de la tabla Municipios.

Nombre: Localidades		
Descripción: Almacena las Localidades del país.		
Atributo	**Tipo**	**Descripción**
Idlocalidad	Integer	Identificador de la Localidad. (PK)
localidad	Varchar(60)	Nombre de la Localidad.
idmunicipio	Integer	Identificador del Municipio en la tabla Municipios. (FK)

Tabla 3.29 Descripción de la tabla Localidades.

Nombre: tb_Calles		
Descripción: Almacena las Calles del país por Localidad.		
Atributo	**Tipo**	**Descripción**
Id_calles	Integer	Identificador de la Calle. (PK)
descripción	Varchar(60)	Nombre de la Calle.
Id_localidad	Integer	Identificador de la Localidad en la tabla Localidades. (FK)

Tabla 3.30 Descripción de la tabla Tb_Calles.

Nombre: tb_Manzanas		
Descripción: Almacena las Manzanas del país por Localidad.		
Atributo	**Tipo**	**Descripción**
id_manzana	Integer	Identificador de la Manzana. (PK)
descripción	Varchar(60)	Nombre de la Manzana.
id_localidad	Integer	Identificador de la Localidad en la tabla Localidades. (FK)

Tabla 3.31 Descripción de la tabla Tb_Manzanas.

Nombre: tb_Calles_Manzanas		
Descripción: Almacena la relación de las calles de cada manzana, guardando sus identificadores.		
Atributo	**Tipo**	**Descripción**
Id_calle	Integer	Identificador de la Calle en la tabla Tb_Calles. (PK)(FK)
Id_manzana	Integer	Identificador de la Manzana en la tabla Tb_Manzanas. (PK)(FK)

Tabla 3.32 Descripción de la tabla Tb_Calles_Manzanas.

Descripción de las tablas de la Base de Datos del Registro de Localidades.

Nombre: tb_Consejo_Localidad		
Descripción: Almacena los identificadores de todas las Localidades que abarcan los Consejos Populares.		
Atributo	**Tipo**	**Descripción**
id_localidad	Integer	Identificador de la Localidad en el Registro de Ubicación.(PK)
id_consejo	Integer	Nombre del Consejo Popular en la tabla Tb_Consejo_Popular. (FK)

Tabla 3.33 Descripción de la tabla Tb_Consejo_Localidad.

Nombre: Tb_Consejo_Popular		
Descripción: Almacena los Consejos Populares del país por Municipio.		
Atributo	**Tipo**	**Descripción**
id_consejo	Integer	Identificador del Consejo Popular.(PK)
descripción	Varchar(50)	Nombre del Consejo Popular.
id_municipio	Integer	Identificador del Municipio en el Registro de Ubicación.

Tabla 3.34 Descripción de la tabla Tb_Consejo_Popular.

Nombre: Tb_Circunscripción		
Descripción: Almacena las Circunscripciones del país por Consejo Popular.		
Atributo	**Tipo**	**Descripción**
id_circunscripción	Integer	Identificador de la Circunscripción. (PK)
descripción	Varchar(50)	Nombre de la Circunscripción.
id_consejo	Integer	Identificador del Consejo Popular en la tabla Tb_Consejo_Popular. (FK)

Tabla 3.35 Descripción de la tabla Tb_Circunscripción.

Nombre: Tb_Zona_cdr		
Descripción: Almacena las Zonas del país por Circunscripción.		
Atributo	**Tipo**	**Descripción**
id_zona	Integer	Identificador de la Zona. (PK)
descripción	Varchar(50)	Nombre de la Zona.
id_circunscripción	Integer	Identificador de la Circunscripción en la tabla Tb_Circunscripción. (FK)

Tabla 3.36 Descripción de la tabla Tb_Zona_cdr.

Nombre: Tb_CDR		
Descripción: Almacena los CDR del país por Zona.		
Atributo	**Tipo**	**Descripción**
id_CDR	Integer	Identificador del CDR. (PK)
No CDR	Integer	Número del CDR
descripción	Varchar(50)	Nombre del CDR.
id_zona	Integer	Identificador de la Zona en la tabla Tb_Zona_cdr. (FK)

Tabla 3.37 Descripción de la tabla Tb_CDR.

Nombre: Tb_CDR_ Manzanas		
Descripción: Almacena los identificadores de todas las Manzanas que abarcan los CDR.		
Atributo	**Tipo**	**Descripción**
Id_CDR	Integer	Identificador del CDR en la tabla Tb_CDR (FK)
Id_manzana	Integer	Identificador de la Manzana en el Registro de Ubicación. (PK)

Tabla 3.38 Descripción de la tabla T_CDR_Manzanas.

Conclusiones

En el capítulo se ha descrito la propuesta de solución al problema planteado. Se describieron y justificaron los patrones de diseño utilizados. Se definen la estructura y los elementos de diseño. Se representaron los diagramas de clases del diseño, se describieron las clases y atributos involucrados en los diagramas de clases. También se expone el diagrama de clases persistentes, el modelo de datos y la descripción de todas las tablas y atributos de ambos registros.

Capítulo 4: Implementación

Introducción.

En este capítulo se describe el proceso de implementación a partir del diseño de la solución propuesta. Se justifica la integración con otros componentes del Registro Informatizado de la Salud (RIS). Se presentan los diagramas de componentes del Registro de Ubicación Geográfica y el Registro de Localidades, así como el diagrama de despliegue. Se describen los métodos del negocio más complejos o agentes de ambos registros. Se detallan las pautas que se definieron para estandarizar las interfaces, así como los estándares de codificación y tratamientos de errores.

4.1 Justificación de integración con otros sistemas.

El Registro de Ubicación Geográfica y el Registro de Localidades al igual que todos los componentes que se integren al Sistema de Información para la Salud (SISalud) tienen relación con el Módulo de Administración, el cual tiene implementado el Componente de Seguridad (SAAA).

Componente de Seguridad (SAAA):

Está basado en el modelo de Autenticación, Autorización y Auditoría (AAA). La autenticación debe ser la primera acción del usuario en el sistema y consiste en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica. Si el usuario autenticado no se encuentra registrado se reporta un error de acceso. En caso contrario, se autoriza su acceso y se crea un certificado digital y se retornan todos los datos y permisos del usuario, desglosado por módulos. Cada Petición de usuario, autorizada o no, es registrada, así como el día, mes, año, hora, minuto, segundo en que se registra y si fue o no autorizada. [38]

El certificado digital del SAAA brinda el nivel en que se autentica un usuario. Los Registros de Ubicación Geográfica y Localidades utilizan esta información para saber la Provincia (si el nivel es provincial) o el Municipio (si el nivel es municipal). De esta forma se le permite al usuario gestionar la información propia de su ubicación (Nación, Provincia o Municipio).

Registro de Ubicación:

El Registro de Localidades también se relaciona con el Registro de Ubicación, del cual toma la información de las Provincias, Municipios, Localidades y Manzanas. Se utilizan los Municipios para ubicar en ellos a los Consejos Populares, las Localidades para decir las Localidades que abarcan los Consejos y

las Manzanas para decir las Manzanas que abarca un CDR. La información de las Provincias se utiliza para realizar búsquedas en las diferentes opciones de este registro.

4.2 Modelo de Implementación.

Los diagramas de despliegue y componentes conforman lo que se conoce como modelo de implementación. Este describe los componentes a construir, su organización y la dependencia entre nodos físicos, en los que funcionará el sistema o aplicación.

El modelo de implementación del Registro de Ubicación Geográfica y el Registro de Localidades expone una organización en capas, jerarquías de paquetes y subsistemas de implementación. Los mismos contienen componentes y sus relaciones, dividiendo al sistema en trozos más manejables. Esto posibilita la reutilización, que se pueda implementar por separado y disminuye el impacto que pueda traer consigo un cambio. A continuación se presenta una vista global de la estructura y organización de la implementación del sistema:

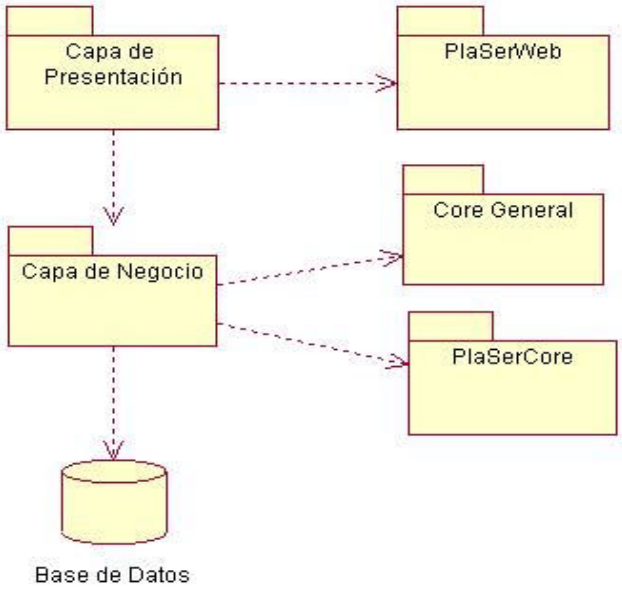


Figura 4.1 Vista Global de la Implementación.

4.2.1 Diagramas de Componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. También se utilizan para mostrar las dependencias de compilación de los ficheros, relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseño que son implementados.

A continuación se presentan los diagramas de componentes de los Casos de Uso, ListarMunicipios y ListarConsejosPopulares, separados por capas:

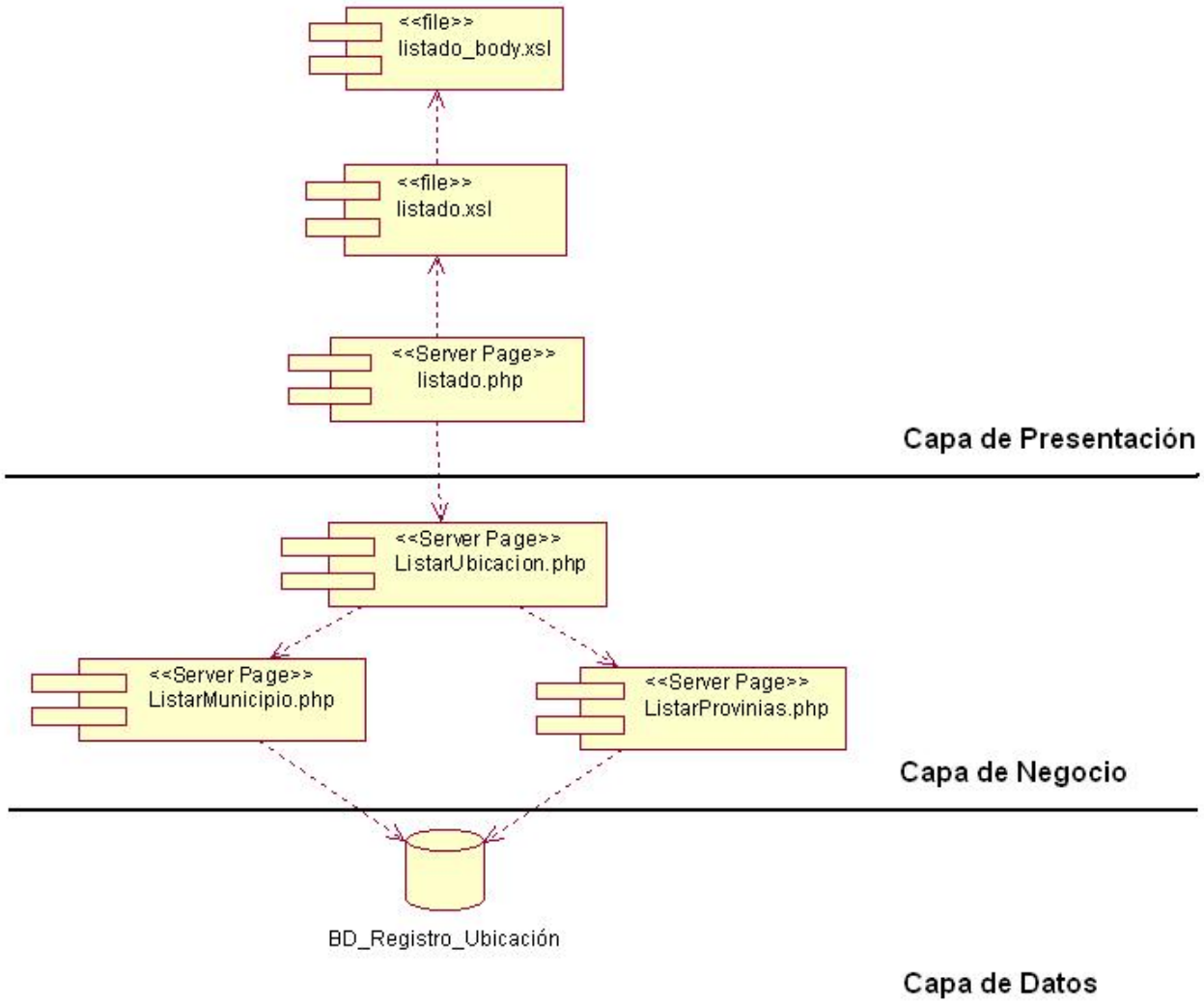


Figura 4.2 Diagrama de Componentes del Caso de Uso ListarMunicipios.

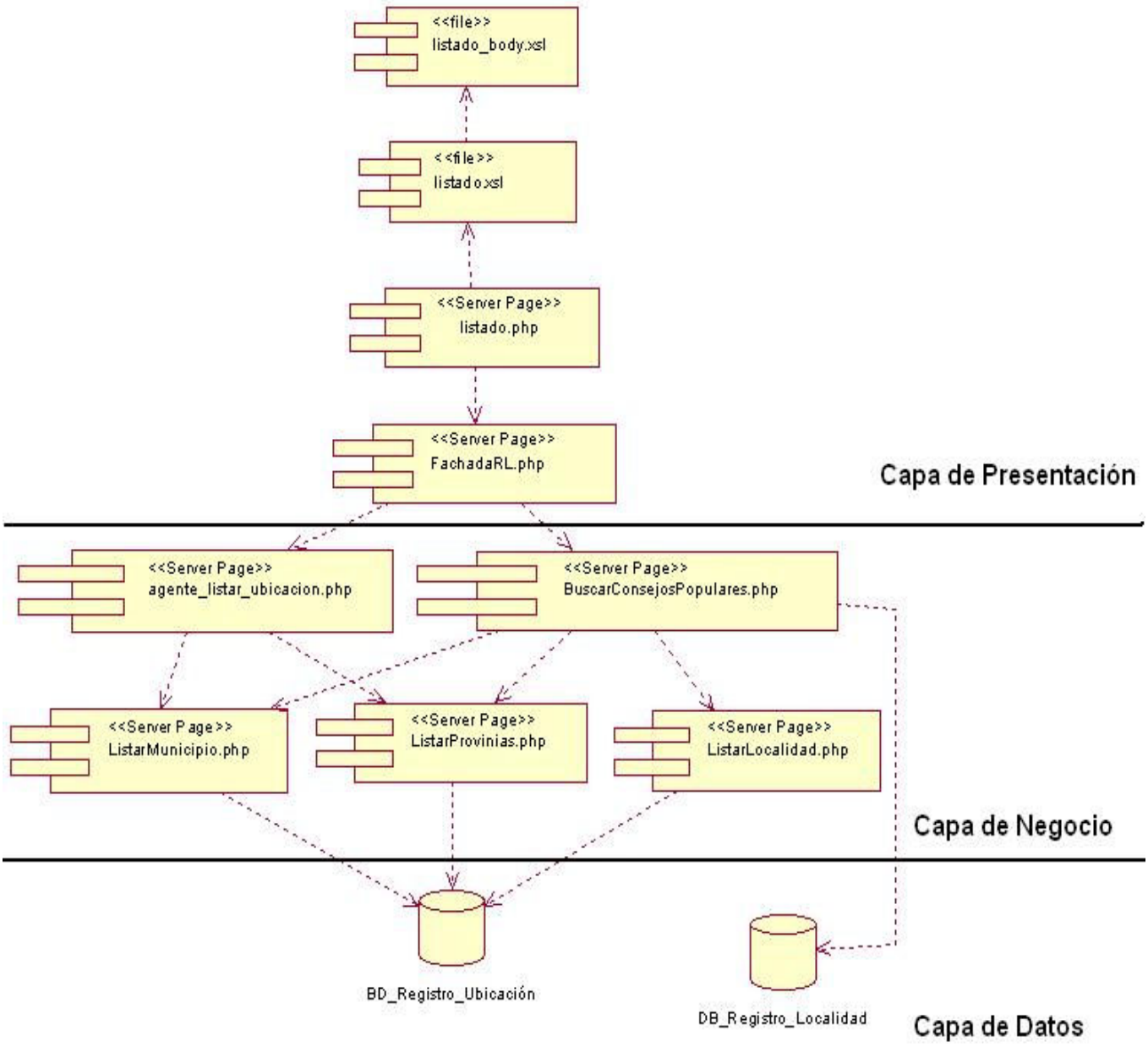


Figura 4.3 Diagrama de Componentes del Caso de Uso ListarConsejosPopulares.

4.2.2 Diagrama de Despliegue.

El diagrama de despliegue muestra cómo están distribuidos los componentes de software entre los distintos nodos de cómputo. El diagrama está compuesto por dispositivos, procesadores y protocolos. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware. Como el diagrama de despliegue describe la distribución física del sistema y ambos Registros, Ubicación y Localidades, se encuentran en el mismo sistema, el diagrama es igual y se representa a continuación:

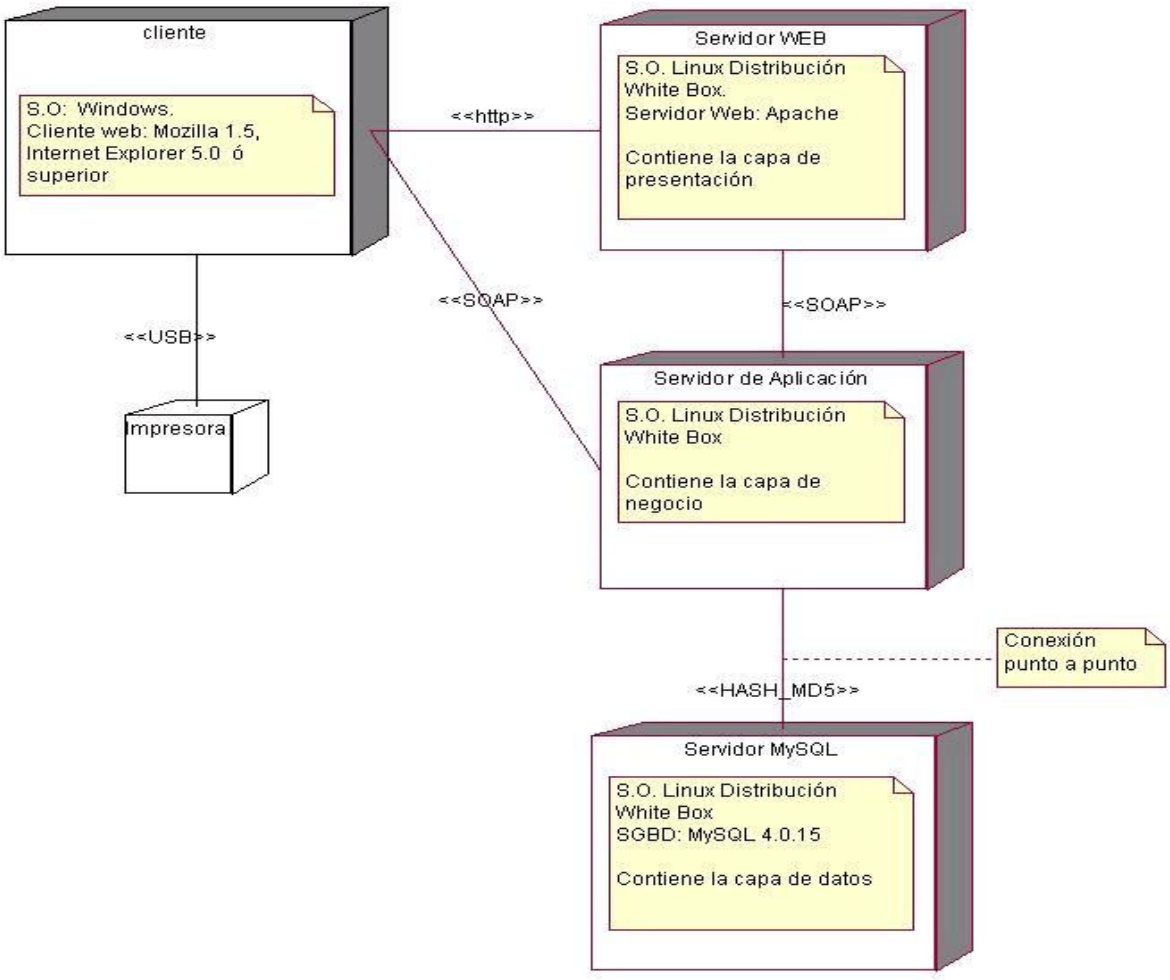


Figura 4.4 Diagrama de Despliegue.

4.3 Descripción de los Métodos.

A continuación se presenta la descripción de algunos de los métodos del negocio más complejos o agentes del Registro de Ubicación Geográfica y Registro de Localidades.

Métodos del Registro de Ubicación Geográfica.

ListarProvincias.

Este método lista la información de los Municipios según los parámetros de entrada.

Recibe los siguientes parámetros: profundidad (puede ser 1, 2 ó 3), offset, cantidad, arregloid (arreglo de Municipios). El método retorna los datos de las Provincias que cumplan con los criterios de búsqueda especificados. De acuerdo con la profundidad especificada en los parámetros de entrada el método devolverá: las Provincias si la profundidad es 1, las Provincias con sus Municipios si la profundidad es 2 y si es 3 devuelve además las Localidades de cada Municipio. Pueden acceder a él usuarios editores y visualizadores de cualquier nivel.

ListarMunicipios.

Este método lista la información de los municipios según los parámetros de entrada.

Recibe los siguientes parámetros: profundidad (puede ser 1, 2 ó 3), offset, cantidad, arregloid (arreglo de Municipios), arregloidprov (arreglo de Provincias), tipo_salida (1 ó 2). El método retorna los datos de los Municipios que cumplan con los criterios de búsqueda especificados. Si el tipo_salida tiene valor 1 se generará la salida en un archivo con extensión pdf y el método retornará, en el elemento URL, la localización del archivo generado para ser descargado, si tiene valor 2 ocurre lo mismo pero el fichero se genera con extensión xls. El parámetro tipo_salida es opcional, en caso de no existir, el método retornará el mensaje SOAP correspondiente. De acuerdo con la profundidad especificada el método devolverá los Municipios con sus Provincias si la profundidad es 1, si la profundidad es 2 devuelve el Municipio y las Localidades de cada Municipio, y si es 3 devuelve la Provincia, Municipio y las Localidades. Pueden acceder a él usuarios editores y visualizadores de cualquier nivel.

ListarUbicacion.

Su función es redireccionar al método ListarProvincias o ListarMunicipio en dependencia de los parámetros de entrada.

Recibe los siguientes parámetros: profundidad (puede ser 1, 2 ó 3), nivel (1 si es nacional, 2 si es provincial y 3 si es municipal). Este método verifica el nivel del usuario, si el nivel es 1 ó 2, llama al método ListarProvincias con profundidad 3, si el nivel es 3, llama al método ListarMunicipios con profundidad 3 y si el nivel es 4 llama al ListarProvincias con profundidad 3 también.

Métodos del Registro de Localidades.

BuscarConsejosPopulares

Este método lista la información de los Consejos Populares según los parámetros de entrada.

Recibe los siguientes parámetros: id_consejo (puede ser un arreglo de id consejos), descripción, id_localidad (arreglo de identificadores de localidades), offset, cantidad, ordenar_por, ordena, todos (cuando se hace una búsqueda por un Mostrar Todos), tipo_salida (1 ó 2), municipio y limit. El método retorna los datos de los Consejos Populares que cumplan con los criterios de búsqueda especificados, además de un arreglo de Id y descripciones de Localidades, los cuales son obtenidos mediante el método ListarLocalidad perteneciente al Registro de Ubicación.

Además utiliza el método ListarMunicipio para obtener la descripción del Municipio al cual pertenece el Consejo Popular retornado en la búsqueda. Si el tipo_salida tiene valor 1 se generará la salida en un archivo con extensión pdf y el método retornará, en el elemento URL, la localización del archivo generado para ser descargado, si tiene valor 2 ocurre lo mismo pero el fichero se genera con extensión xls. El parámetro tipo_salida es opcional, en caso de no existir el método retornará el mensaje SOAP correspondiente.

agente_listar_municipios

Este método lista la información de los Municipios y las Localidades correspondientes según los parámetros de entrada.

Recibe el idpro como parámetro. El método retorna los datos de los Municipios, con las Localidades correspondientes, que cumplan con los criterios de búsqueda especificados. Estos datos son obtenidos mediante el método ListarMunicipio perteneciente al Registro de Ubicación.

agente_listar_ubicacion

Este método brinda la información de las Provincias según el parámetro de entrada.

Recibe la profundidad como parámetro. El método retorna los datos de las Provincias con sus Municipios si el parámetro de entrada tiene valor 2, si el valor de este parámetro es 3 pues entonces este método retornaría la información de las Provincias con sus Municipios correspondientes además de las localidades correspondiente a cada Municipio. Estos datos son obtenidos mediante los métodos ListarProvincias y ListarMunicipio pertenecientes al Registro de Ubicación.

4.4 Estándares de diseño, codificación y tratamiento de errores.

Estándares de diseño.

El Registro de Ubicación Geográfica y el Registro de Localidades siguen las pautas de diseño del Registro Informatizado de la Salud (RIS). Estas pautas permiten lograr una mayor eficiencia en el proceso de trabajo al existir una coherencia formal entre todos los módulos y páginas del sistema, y que estos sean identificados como parte de un todo. Se han pautado una serie de elementos comunes que facilitarán su reconocimiento y el uso que se haga de ellos.

La resolución utilizada para el diseño de la aplicación es de 800 x 600 pixel. Se definió un estilo CSS (del inglés, Cascading Style Sheets, Hoja de Estilo en Cascada) para permitir separar el contenido de la presentación y lograr una homogeneidad en las letras, imágenes, tablas, controles, encabezados, etc. El CCS permite manejar los cambios que se hagan al diseño de forma centralizada ya que un cambio en él modifica el estilo de todos los implicados.

Para el diseño de cualquier interfaz que esté en el RIS se debe tener en cuenta que su diseño va a estar dividido en 4 partes, el header o cabecera, menú, región editable y footer o pie.

En el header se encuentra el nombre de la aplicación que aparece en la parte superior izquierda (A); tres vínculos: Páginas de Inicio, Acerca de y Ayuda, los cuales permiten entrar a las páginas de Inicio, una explicación del RIS y la ayuda de cada página respectivamente (B). Cuenta con una imagen que identifica al módulo en la parte izquierda (C); en la parte central el nombre y una breve descripción del módulo (D). En la parte derecha aparece el nombre del Usuario, Derechos (Administrador, Visualizador o Editor), Nivel (Nacional, Provincial, Municipal o Unidades de Salud) y módulos a los cuales el usuario que se autenticó en el RIS tiene privilegios (E).

En la parte izquierda se encuentra el menú donde se listan un conjunto de vínculos a los cuales el usuario tiene permisos para entrar (F).

En la parte central se encuentra la región editable en la cual se realiza el proceso de gestión de la información, ya sea, un listado, agregar o editar (G).

En la parte inferior se encuentra el footer el cual describe los derechos y dice el ministerio al que pertenece la aplicación (H).

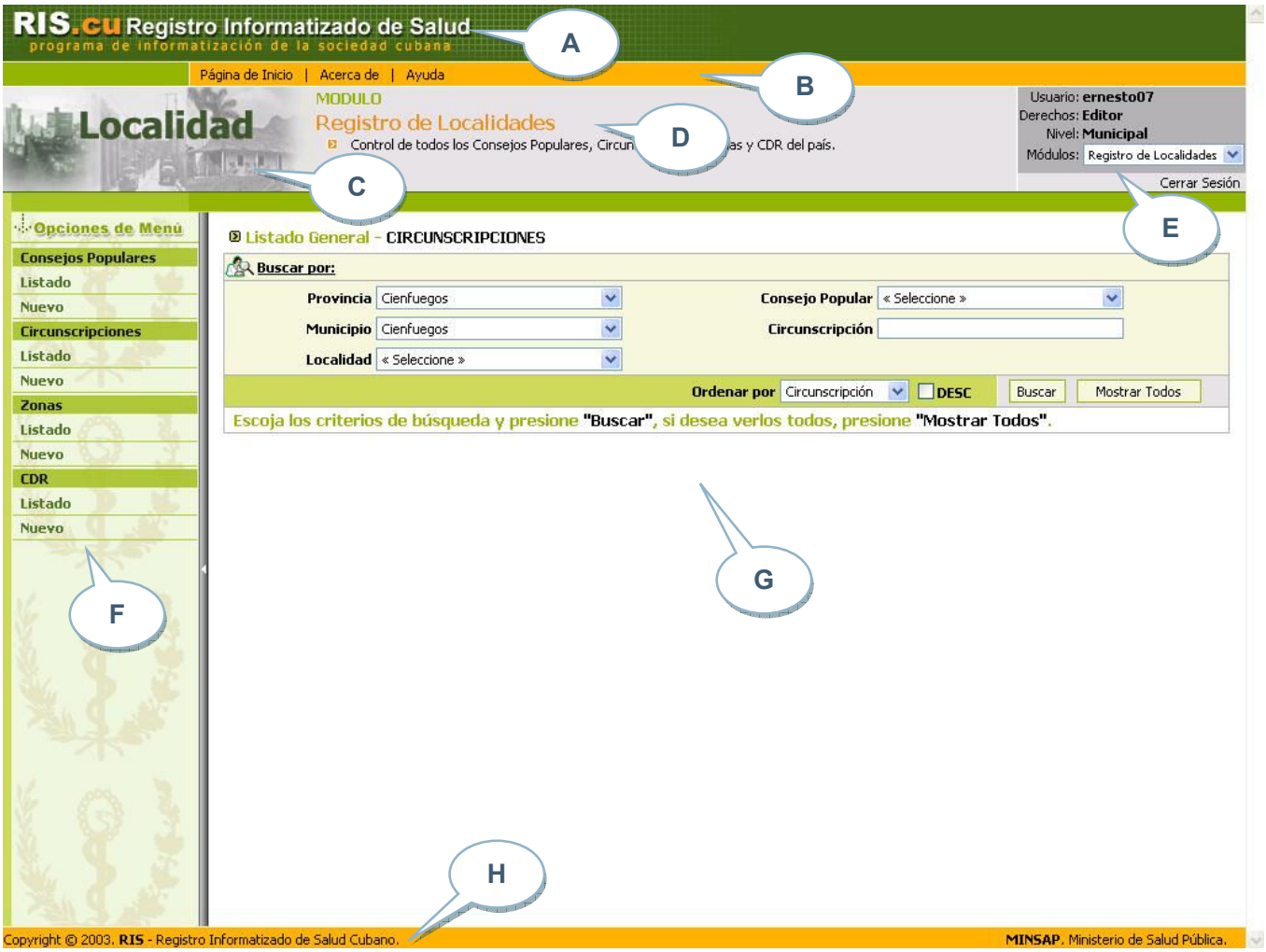


Figura 4.5 Ejemplo de interfaz de usuario del Registro de Localidades.

Estándares de codificación.

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

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

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

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

Todos los campos identificadores van a comenzar con el identificador (id) seguido del nombre del campo. Ejemplo: id _ calle.

Los arreglos empezarán con el identificador array y las palabras no se separarán con el carácter "_". Ejemplo: Arrayidcalles.

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

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

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

Se usará una 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_calles

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

Los controles seguirán el siguiente tratamiento:

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

Tabla 4.1 Estándares para los controles.

Las páginas HTML se harán sin incluir código y todas las funciones JavaScript que se usarán y 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_RU

Las tablas se identificarán con el acrónimo tb_Nombre, ejemplo: tb_calles.

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

Cada módulo definirá un espacio de nombre (namespace), siguiendo la siguiente estructura: SOFTEL_APS_RU.

Cada método que se defina debe seguir la siguiente estructura, acrónimo del módulo. Nombre del método. Ej.: RU_InsertarCalle.

Tratamiento de errores.

Se utiliza JavaScript para depurar los errores de parte del cliente validando los formularios y evitando consultas a la base de datos sin sentido o que tengan malas intenciones como son las inyecciones SQL, en cualquiera de estas situaciones el JavaScript funciona como una barrera y no deja pasar estos valores lanzando alertas o excepciones.

Figura 4.6 Ejemplo de una alerta en JavaScript.

Otros errores en la capa de negocio serán tratados devolviendo un SOAP_FAULT, cuyos elementos FaultCode, FaultString, FaultAutor se describen 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-RU: 1.5 que indica error 5 en el método 1 del módulo Registro de Ubicación Geográfica 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.

FaultAutor:

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

Detail:

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

Conclusiones.

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

Conclusiones

Los Registros de Ubicación Geográfica y Localidades se encuentran en proceso de liberación en la empresa de Soluciones Informáticas (Softel), para posteriormente ser desplegados en Infomed. Partiendo de este hecho se puede concluir que:

- Se realizó un estudio detallado de la información de la ubicación geográfica y localidades, necesaria para el Sistema de Información para la Salud.
- Se modelaron los flujos de trabajo: Modelamiento de Negocio, Requerimientos, Diseño e Implementación; quedando completa la documentación de Ingeniería de Software de ambos registros.
- Se implementó el Registro de Ubicación Geográfica y el Registro de Localidades, siguiendo las políticas de informatización del Sistema Nacional de Salud, la arquitectura definida por el MINSAP y la integración con otros componentes.

Por todo lo expuesto anteriormente, se cumplieron los objetivos del trabajo, logrando implementar el Registro de Ubicación Geográfica y el Registro de Localidades, los cuales permiten hacer búsquedas dinámicas, gestionar la información y brindar servicios a las demás aplicaciones integradas al Sistema de Información para la Salud.

Recomendaciones

Al concluir el trabajo y habiendo cumplido el objetivo propuesto, se recomienda:

- Desplegar ambos registros para su uso en el Sistema de Información para la Salud.
- Introducir la información de estos registros en los niveles que corresponda, dada su importancia para las demás aplicaciones de SISalud.
- Implementar el nivel de unidades de salud, permitiendo supervisar la información correspondiente a la ubicación geográfica y las localidades en cada unidad.

Referencias Bibliográficas

[1] **Fernando Antezana, Aranibar.** Ministerio de Salud y Deportes. [En línea] [Citado el: 16 de Marzo de 2007.] http://www.sns.gov.bo/polinasa.htm

[2] **Infomed.** *sitio Web de la Red de Salud de Cuba.* [En línea] Escuela Nacional de Salud Pública, 2004. [Citado el: 15 de Marzo de 2007.] http://www.sld.cu/galerias/doc/sitios/infodir/25_proyeccion_estartegica.doc

[3] Idem al 2.

[4] **Delgado Ramos, Ariel y Vidal Ledo, María.** Revista Cubana Salud Pública. [En línea] 2006. [Citado el: 12 de Marzo de 2007.] http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm#cargo

[5] Idem al 2.

[6] CUBAMinRex. *sitio Web del Ministerio de Relaciones Exteriores de la República de Cuba.* [En línea] [Citado el: 13 de Febrero de 2007.] http://www.cubaminrex.cu/Enfoques/ddhh_salud_tc.htm

[7] **Colaboradores, Rigol y.** *Medicina General Integral,*La Habana : Ciencias Sociales, 1986, Vols. 2-10.

[8] **Rosa, Lemus Elia, Radamés, Borroto Eugenio y R, Aneiros-Riba.** Infomed. *sitio Web Red Salud en Cuba.* [En línea] 1998. [Citado el: 20 de Marzo de 2007.] http://www.sld.cu/galerias/doc/sitios/infodir/17_borroto_eugenio_radames_1.doc

[9] **Ramírez Márquez, Dr. Abelardo; Castell-Florit Serrate; Dr. Pastor y Mesa, Dr. Guillermo.** *EL SISTEMA NACIONAL DE SALUD DE CUBA.* [Documento Word] La Habana : ENSAP, 2003.

[10] **Delgado Ramos, Dr. Ariel, Cabrera Mirna, Ing. Mirna y Juncal, Dra Virginia.** *Registro Informatizado de Salud (RIS).* La Habana : Temas Estadísticos de Salud, 2005.

[11] Idem al 8.

[12] **Cobas Guilarte, Osmel.** *Registro de Población para el Sistema Informatizado de Atención Primaria.* Instituto Superior Politécnico "Jose Antonio Echeverría". La Habana, 2005.página 9.

[13] Idem al 10.

[14] Idem al 10.

[15] Idem al 10.

[16] **Cabrera Hernández, Mirna. 2006.** *Propuesta de Esquema para el Sistema Ingral de Salud.* La Habana : Softel, 2006.

[17] Idem al 4.

[18] **Delgado Ramos, Ariel.** *Informatización Sistema Nacional de Salud.* La Habana : MINSAP, 2003.

[19] **Barrett, Neil.** *El estado de la Cibernación, Consecuencias culturales, políticas y económicas de la Internet.* España : Liberduplex, Octubre 1998.

[20] "Internet." **Microsoft® Encarta® 2007** [CD]. Microsoft Corporation, 2006.

[21] **GNU, Proyecto.** GNU. *sitio Web de GNU.* [En línea] Free Software Foundation. [Citado el: 13 de Marzo de 2007.] http://www.gnu.org/philosophy/free-sw.es.html

[22] Idem al 21.

[23] "Linux." **Microsoft® Encarta® 2007** [CD]. Microsoft Corporation, 2006.

[24] **Cerami, Ethan.** *Web Services Essentials.* s.l. : O'Reilly, 2002. ISBN: 0-596-00224-6.

[25] **Gallegos, Rafael.** Maestría en Sistemas de Información Gerencial. *sitio Web de la Escuela Superior Politécnica del Litoral .* [En línea] [Citado el: 20 de Junio de 2007.] http://www.msig.espol.edu.ec/recursos/9.Service_Oriented_Architecture_Resumen.pdf

[26] "Arquitectura cliente/servidor." **Microsoft® Encarta® 2007** [CD]. Microsoft Corporation, 2006.

[27] **Softel.** *Documento sobre la Arquitectura de Software para los componentes a emplear por el Sistema de Información para la Salud.* La Habana, 2006.

[28] Idem al 27.

[29] Idem al 27.

[30] **Barco, Antonio.** *SOA y los Servicios Web (I).* [En línea] [Citado el: 9 de Junio de 2007.] http://arquitecturaorientadaaservicios.blogspot.com/

[31] Idem al 30.

[32] **Cobas Guilarte, Osmel.** *Registro de Población para el Sistema Informatizado de Atención Primaria.* Instituto Superior Politécnico "Jose Antonio Echeverría". La Habana, 2005.página 40.

[33] Idem al 32.

[34] "Lenguaje de programación." **Microsoft® Encarta® 2007** [CD]. Microsoft Corporation, 2006.

[35] **Gilfillan, Ian.** *La Biblia MySQL.* ISBN: 8441515581.

[36] **Franco Navarro, Ing. Jose Angel.** *UML en acción. Modelando Aplicaciones Web.* Instituto Superior Politécnico "Jose Antonio Echeverría".La Habana, 2005.

[37] Idem al 36.

[38] Idem al 27.

Bibliografía

1. Alarcón de Quesada, Ricardo. LEY No. 91 De los Consejos Populares. *sitio Web de la Asamblea Nacional del Poder Popular de la República de Cuba.* [En línea] [Citado el: 10 de Mayo de 2007.] http://www.parlamentocubano.cu/espanol/ley91.htm.

2. Barrett, Neil. *El estado de la Cibernación, Consecuencias culturales, políticas y económicas de la Internet.* España, Liberduplex, Octubre 1998.

3. Cerami, Ethan. *Web Services Essentials.* s.l. O'Reilly, 2002. ISBN: 0-596-00224-6.

4. Delgado Ramos, Ariel. *Informatización Sistema Nacional de Salud.* La Habana, MINSAP, 2003.

5. Delgado Ramos, Ariel y Vidal Ledo, María. Revista Cubana Salud Pública. [En línea] 2006. [Citado el: 12 de Marzo de 2007.] http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm#cargo

6. Delgado Ramos, Dr. Ariel, Cabrera Mirna, Ing. Mirna y Juncal, Dra Virginia. *Registro Informatizado de Salud (RIS).* La Habana : Temas Estadísticos de Salud, 2005.

7. Franco Navarro, Ing. Jose Angel. *UML en acción. Modelando Aplicaciones Web.* Instituto Superior Politécnico "Jose Antonio Echeverría".La Habana, 2005.

8. Gilfillan, Ian. *La Biblia MySQL.* ISBN: 8441515581.

9. Infomed. *sitio Web de la Red de Salud de Cuba.* [En línea] Escuela Nacional de Salud Pública, 2004. [Citado el: 15 de Marzo de 2007.] http://www.sld.cu/galerias/doc/sitios/infodir/25_proyeccion_estartegica.doc

10. Jacobson, Ivan, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software, Volumen I.* La Habana, Editorial Félix Varela, 2004.

11. Jacobson, Ivan, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software, Volumen II.* La Habana, Editorial Félix Varela, 2004.

12. Kruchten, Philippe. *Architectural Blueprints--The 4+1 View Model of Software Architecture. .* s.l. : Institute of Electrical and Electronics Engineers, r 1995, pp. 42-50.

13. Larman, Craig. UML y Patrones. La Habana, Editorial Félix Varela, 2004.

14. Microsoft ® Encarta ® 2007. © 1993-2006 Microsoft Corporation. Reservados todos los derechos.

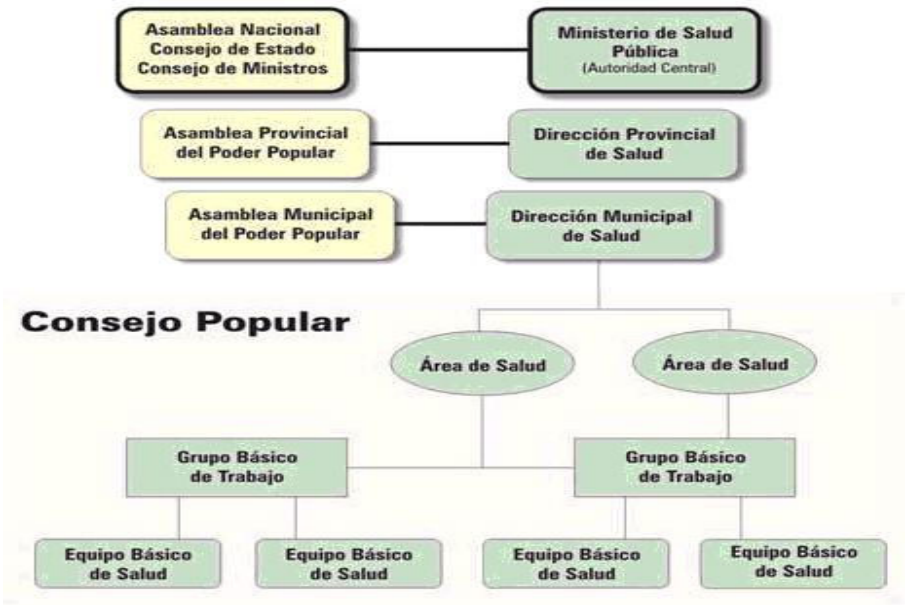
15. Pressman, Roger. *Ingeniería del Software un enfoque práctico, Parte 1.* La Habana, Editorial Félix Varela, 2005.

16. Pressman, Roger. *Ingeniería del Software un enfoque práctico, Parte 2.* La Habana, Editorial Félix Varela, 2005.

17. Ramírez Márquez, Dr. Abelardo, Castell-Florit Serrate, Dr. Pastor y Mesa, Dr. Guillermo. *EL SISTEMA NACIONAL DE SALUD DE CUBA.* La Habana , ENSAP, 2003.

Anexos

Anexo I. Estructura administrativa del Sistema Nacional de Salud.



Anexo II. Códigos Postales de España: Búsqueda por Localidad.

Búsqueda por localidad

Por favor, introduzca localidad y dirección para obtener su código postal:

Localidad : Valles San Juan

Dirección*: Nº:

* Escriba la dirección sin incluir el tipo de vía

Buscar

España					
Provincia	Localidad	Calle	Impar	Par	C.P.
MADRID	VALLES SAN JUAN (URBANIZACION)				28380

Realizar nueva búsqueda

Anexo III. Códigos Postales de España: Búsqueda por Provincia.

Búsqueda por provincia

Por favor, seleccione País, provincia y localidad:

País: España
Andorra

Provincia : MADRID

Localidad : VALLES SAN JUAN

Dirección*: Nº:

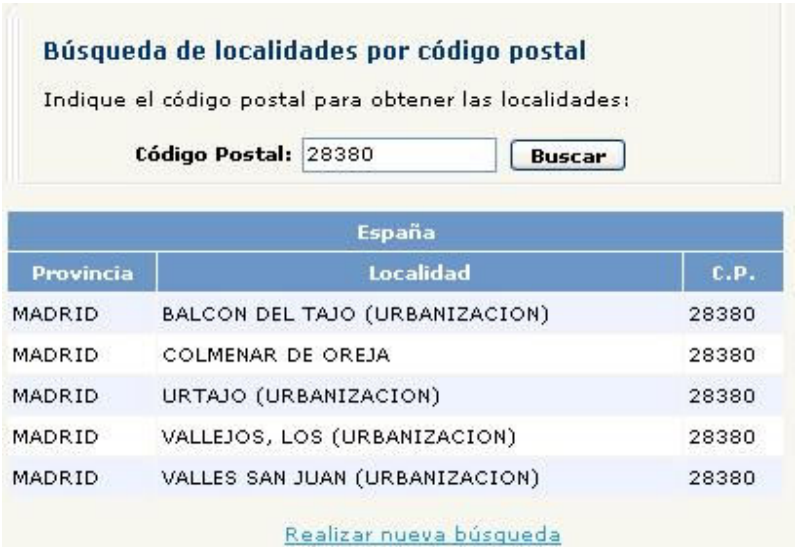
* Escriba la dirección sin incluir el tipo de vía

Buscar

ESPAÑA - MADRID				
Localidad	Calle	Impar	Par	C.P.
VALLES SAN JUAN (URBANIZACION)				28380

Realizar nueva búsqueda

Anexo IV. Códigos Postales de España: Búsqueda por Código Postales.



Búsqueda de localidades por código postal

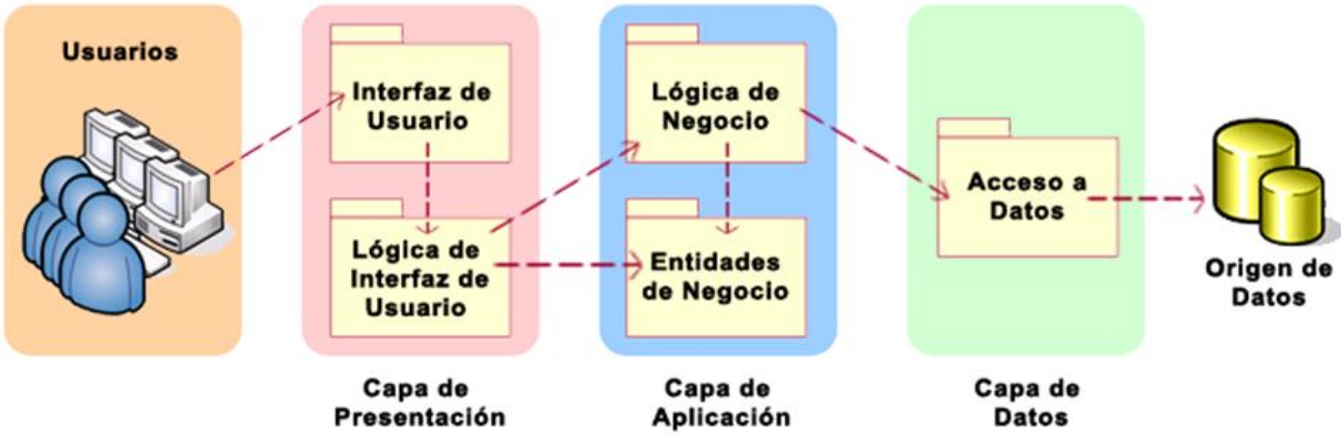
Indique el código postal para obtener las localidades:

Código Postal: 28380 Buscar

España		
Provincia	Localidad	C.P.
MADRID	BALCON DEL TAJO (URBANIZACION)	28380
MADRID	COLMENAR DE OREJA	28380
MADRID	URTAJO (URBANIZACION)	28380
MADRID	VALLEJOS, LOS (URBANIZACION)	28380
MADRID	VALLES SAN JUAN (URBANIZACION)	28380

Realizar nueva búsqueda

Anexo V. Arquitectura de 3 capas.



Glosario de Términos

Active Server Pages (ASP): Es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente.

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 son 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 sólo 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.

Base de Datos: Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

Bugs: En informática, error en el software o en el hardware.

Caso de Uso: Descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

Componente: Parte física y reemplazable de un sistema que se ajusta a, y proporciona la realización de, un conjunto de interfaces.

COM (Component Object Model): Es una plataforma de Microsoft para componentes de software.

Concurrencia: Ejecución simultánea de dos o más actividades durante el mismo intervalo de tiempo.

Dependencia: Relación semántica entre dos elementos, en la cual un cambio en uno puede afectar al otro.

Diagrama: Presentación gráfica de un conjunto de elementos y sus relaciones.

Dominio: Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminología comprendidos por los practicantes de ese dominio.

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

IBM o International Business Machines: Conocida coloquialmente como el Gigante Azul, es una empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

Interoperabilidad: Condición necesaria para que los usuarios (humanos o mecánicos) tengan un acceso completo a la información disponible. Entre las iniciativas recientes más destacadas para dotar a la Web de interoperabilidad se encuentran los servicios Web y la Web semántica.

INFOMED: Red telemática del Sistema Nacional de Salud (SNS) de Cuba que funciona como una división del Centro Nacional de Información de Ciencias Médicas (CNICM), y parte de la existencia de una red nacional especializada de centros de información.

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

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

Informix: Familia de productos RDBMS de IBM.

Internet: Es un método de interconexión de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red (lógica) única.

Java: Un lenguaje de programación de alto nivel, orientado a objetos.

Mac OS: Sistema operativo utilizado por computadoras Apple. En la actualidad se comercializa la versión 10.4 de Mac OS X, como la última versión del sistema.

Metalenguaje: Es un lenguaje usado para hacer referencia a otros lenguajes.

ODBC (Open Database Connectivity): Estándar de acceso a Bases de Datos.

Open Source (Código abierto): es el término con el que se conoce al software distribuido y desarrollado libremente.

Oracle: Sistema de gestión de base de datos relacional.

Paquete: Mecanismo de propósito general para organizar elementos en grupos.

PDF (del inglés, Portable Document Format, Formato de Documento Portátil): Es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

Policlínico: Unidad de salud donde se brindan servicios médicos a una población geográficamente determinada perteneciente al nivel asistencial de Atención Primaria de Salud.

Proyecto: Esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida.

RPC (del inglés, Remote Procedure Call, Llamada a Procedimiento Remoto): Es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

SOFTEL (Empresa de Soluciones Informáticas): Entidad del Ministerio de Informática y las Comunicaciones (MIC).

Servicio: Unidad de software que encapsula alguna funcionalidad de negocio y proporciona estas a otros servicios a través de interfaces públicas bien definidas.

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

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

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

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

SQL ANSI 92: Estándar SQL.

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

UDDI (Universal Description, Discovery and Integration): Catálogo independiente, basado en XML, que lista los negocios de Internet de todo el mundo.

Unidad de Salud: Centro de trabajo del Ministerio de Salud Pública (MINSAP).

UNIX: Es un sistema operativo portable, multitarea y multiusuario.

UDDI (del inglés, Universal Description, Discovery, and Integration, Universal Descripción, Descubrimiento e Integración): Un formato XML que se utiliza para describir servicios Web.