

Universidad de las Ciencias Informáticas
Facultad 7



**Título: Estadísticas Descriptivas del Registro de
Enfermedades de Declaración Obligatoria**

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

Autores: Alexander Fonseca Cardoso
Yandy Rojas Barrios

Tutores: Dr. Denis Derivet Thaireaux
Ing. Yoenny Pérez Romero

Asesora: Lic. Damaris Figueredo Benítez

Ciudad de La Habana, Julio 2008
"Año 50 de la Revolución"

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 7 días del mes de julio del año 2008.

Alexander Fonseca Cardosa

Firma del Autor

Yandy Rojas Barrios

Firma del Autor

Dr. Denis Derivet Thaireaux

Firma del Tutor

Ing. Yoenny Pérez Romero

Firma del Tutor

DATOS DE CONTACTO

Dr. Denis Derivet Thureaux (denis@softel.uci): Graduado en la Facultad de Ciencias Médicas de Guantánamo como Médico General Básico en 1994 y como Especialista 1er Grado en Medicina General Integral (MGI) en 1999. Es Profesor Instructor en la Universidad de Ciencias Informáticas (UCI) y se desempeña como Especialista Funcional del Ministerio de Salud Pública (MINSAP) en el Proyecto de Informatización de este sector.

Ing. Yoenny Pérez Romero (yoenny@uci.cu): Graduado como Ingeniero Informático en el Instituto Superior Politécnico José Antonio Echeverría en el 2005. Posee 3 años de experiencia laboral, se ha desempeñado como Profesor Instructor en la Universidad de las Ciencias Informáticas (UCI), vinculado siempre a las asignaturas de la especialidad. Durante este período ha estado desarrollando conjuntamente con un equipo de trabajo de la UCI y la empresa SOFTEL una solución informática que automatiza y gestiona los procesos inherentes a la Atención Primaria de la Salud en Cuba (APS), donde ha fungido como Líder de Proyecto. Actualmente cursa la parte lectiva de la Maestría Informática Aplicada que se desarrolla en la UCI.

"El ave canta, aunque la rama cruja, porque conoce lo que son sus alas..."

José Santos Chocano

AGRADECIMIENTOS

A nuestros compañeros y amigos, a Arianna por compartir cinco de nuestros mejores años, a Yaremís y Yanet por su gran ayuda.

Al colectivo de profesores por formarnos académica y profesionalmente, a Mírna por querernos tanto, por ayudarnos siempre y por confiar.

A Yoenny y Denis, por su acertada guía, a Leo por su incondicionalidad.

A nuestros padres, nuestros abuelos, nuestros hermanos. A todos nuestros seres queridos, por encaminarnos en la vida, por apoyarnos.

A nuestro Comandante Fidel, por su fe en nuestra valía, por darnos la posibilidad de ser estudiantes de la Tropa de Futuro.

DEDICATORIA

A mi compañero de tesis, Alexander, por su amistad inquebrantable.

A Libeth, mi niña, por estar siempre a mi lado, por permitirme quererte.

A mi hermano Yaciel, por los pequeños detalles, por estar pendiente de mí.

A mis abuelos, por vivir para su niño, para verlo crecer y graduarse.

A mi mamá y a mi papá por brindarme ese apoyo espiritual que hasta el hombre más fuerte, una vez necesita.

Yandy Rojas Barrios

A mi novia, por estar a mi lado cuando más la necesité.

A Yandy, por la amistad de todos estos años.

A mi hermano, por todo lo atento que ha sido conmigo.

A Aly, quien nunca ha dejado de estar pendiente de mí.

A mi madre y a mi padre por sus siempre buenos consejos, por hacer de mí el hombre que soy y por el constante apoyo a mi superación personal.

Alexander Fonseca Cardoso

RESUMEN

El proceso de desarrollo llevado a cabo en el presente trabajo está dirigido a la obtención de una nueva funcionalidad en el Registro de Enfermedades de Declaración Obligatoria (REDO) que permita visualizar los reportes estadísticos de las Enfermedades de Declaración Obligatoria (EDO) en todos los niveles de dirección, perfeccionando así el actual proceso de gestión de las Estadísticas Descriptivas de las EDO en el Sistema Nacional de Salud (SNS). Para tal propósito ha sido utilizada tecnología LAMP (Linux, Apache, MySQL y PHP), puesto que así lo ha definido el MINSAP en las políticas de desarrollo de sus aplicaciones.

La Empresa de Soluciones Informáticas SOFTEL ha concebido para el desarrollo de sus aplicaciones el uso de la Plataforma de Servicios PlaSer, framework en PHP que resulta ventajoso para el desarrollo de los componentes del Sistema de Información para la Salud (SISalud), porque está basado en el paradigma de Arquitectura Orientada a Servicios y Basada en Componentes (SOA-CBA), específicamente de Servicios Web XML.

Las Estadísticas Descriptivas del REDO brindan un conjunto de beneficios entre los que se destacan: la sustitución del proceso manual de captación y coordinación de la información relativa a las EDO en los departamentos de estadística de las unidades de salud y las direcciones municipales de salud, la erradicación de las imprecisiones de cálculos durante el procesamiento de los datos, así como la reducción del desfasaje existente entre la notificación de los datos y la toma de decisiones por parte del sistema de vigilancia y control epidemiológico.

PALABRAS CLAVE: Sistema Nacional de Salud, Registro de Enfermedades de Declaración Obligatoria, Informatizar, Estadísticas Descriptivas.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
1.1 Sistema Nacional de Salud	7
1.2 Sistemas para la Vigilancia y Control Epidemiológico	10
1.3 Tendencias y Tecnologías Actuales a considerar	13
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	43
2.1 Situación Problemática y Problema a Resolver	43
2.2 Modelo de Negocio	44
2.3 Propuesta del Sistema	54
2.4 Modelo de Casos de Uso del Sistema	62
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	67
3.1 Modelo de Análisis	67
3.2 Modelo de Diseño	70
CAPÍTULO 4: IMPLEMENTACIÓN	94
4.1 Justificación de la Integración con otros Sistemas Externos	94
4.2 Modelo de Implementación	96
CONCLUSIONES	100
RECOMENDACIONES	101
REFERENCIAS BIBLIOGRÁFICAS	102
BIBLIOGRAFÍA	105
ANEXOS	110
GLOSARIO DE TÉRMINOS	112

INTRODUCCIÓN

En los tiempos actuales, en los cuales existe una gran brecha digital entre Cuba y los países del primer mundo, la Revolución ha dirigido los mayores esfuerzos a disminuir esta sustancial diferencia tecnológica, mediante la informatización de sectores como la educación, la salud y la sociedad cubana en general.

Desde mediados de la década del 80 se han venido desarrollando sistemas encaminados a fomentar la informatización de la salud. Sin embargo, aquellas aplicaciones informáticas no poseían un estándar que permitiera la integración entre ellas, además no existían los recursos tecnológicos necesarios para su despliegue en el Sistema Nacional de Salud (SNS). Es entonces cuando en 1997, el sector de la salud, concibe una primera estrategia, fundamentada según los lineamientos establecidos para la informatización de la sociedad cubana, cuya finalidad consistía en aunar esfuerzos en el proceso de desarrollo de la informática en el SNS.

Este proceso, desde comienzos del año 2000, tiene como objetivo fundamental la implementación y puesta en marcha de un programa general de gestión de la información para la salud, donde cada policlínico constituye la célula fundamental, cuyo basamento estratégico se apoya en las políticas trazadas por la dirección del país y el Ministerio de Salud Pública (MINSAP), incorporando gradualmente las Tecnologías de la Información y las Comunicaciones (TIC's) en el SNS.

El SNS está estructurado en los niveles de dirección: Nacional, Provincial, Municipal y Unidad de Salud y en 3 niveles de atención médica de salud: Primaria, Secundaria y Terciaria, de acuerdo con la división político – administrativa del país, la complejidad de las acciones preventivas, curativas y de rehabilitación y el grado de especialización de los servicios médicos.

La Atención Primaria de Salud (APS) es, en este proceso de consolidación del Sistema Nacional de Salud, el centro de las transformaciones dirigidas a optimizar los servicios médicos brindados a la población, en el nivel municipal, de forma tal que los problemas de salud de la comunidad reciban una solución eficiente. Para ello se concibió el desarrollo de una aplicación única que permita la toma de decisiones por parte de los directivos del MINSAP.

Para dar cumplimiento a este propósito, en el año 2003, la Empresa de Soluciones Informáticas SOFTEL, entidad que pertenece al Ministerio de Informática y las Comunicaciones (MIC), comienza a desarrollar aplicaciones y productos de software para incrementar la eficiencia en la gestión de datos del Sistema Nacional de Salud, vinculándose a la Infraestructura Productiva (IP) de la Universidad de las Ciencias Informáticas (UCI).

INFOMED, como Red Telemática de la Salud, que fue creada en 1992 y hoy en día es la columna vertebral de la red de transmisión de datos del SNS, desplegó a fines de 2003, el Registro Informatizado de la Salud (RIS) y creó las condiciones para el desarrollo de un sistema de aplicaciones independientes capaces de establecer comunicación entre ellas, utilizando Servicios Web basados en XML e implementadas sobre una arquitectura Orientada a Servicios y Basada en Componentes (SOA-CBA), permitiendo que cada aplicación gestionara solo la información correspondiente a su negocio, obteniendo de esta manera una mayor eficiencia en el flujo de datos.

En el año 2004 se concibe el Sistema de Información para la Salud (SISalud), como plataforma generalizable para la gestión, procesamiento y transmisión de la información médica. Esta infraestructura permite la integración de todas las aplicaciones para el Sistema Nacional de Salud.

La gestión de la información en la plataforma tiene por finalidad recoger y coordinar numerosos datos, de modo que permita obtener relaciones numéricas sensiblemente independientes de las anomalías del azar y que ponen de manifiesto la existencia de causas regulares cuya acción se combina con las causas fortuitas.

Desde los años 30 del siglo XX, tres grandes estadistas (Ronald Fisher, Sewall G. Wright y J. B. S. Haldane) introdujeron la bioestadística. De forma general, es la aplicación de la estadística a la biología. Debido a que las cuestiones a investigar en biología son de naturaleza muy variada, por ejemplo, la medicina, ciencias agropecuarias y forestales, la bioestadística ha expandido sus dominios para incluir cualquier modelo cuantitativo, no solo estadístico, que pueda ser usado para responder a estas necesidades.

La bioestadística puede ser considerada como una rama, altamente especializada, de la informática médica que puede ser, a su vez, complementada por la bioinformática. La aplicación resulta hoy en día necesaria, en campos tales como ecología, poblaciones genéticas, medicina y salud pública, que incluye epidemiología, nutrición, salud ambiental e investigación de servicios sanitarios. [1]

El Primer Congreso Internacional de Estadística, se celebró en Bruselas en 1853, y en él se propuso introducir la unidad en las oficinas de estadística y sentar bases uniformes para los trabajos realizados en las mismas, ya que eran muy difíciles de llevar las estadísticas internacionales debido principalmente a la diversidad de conceptos que formaban parte de la información.

La estadística aplicada a las ciencias médicas aparece muy rudimentariamente desde los días del renacimiento. Ya desde la antigüedad se recogían números de casos sin agruparlos sistemáticamente, pero estas estadísticas obtienen valor desde el momento en que comienzan a relacionarse con la demografía, informándose los resultados de morbilidad, natalidad y mortalidad en relación con la población; las dos últimas forman parte de las llamadas estadísticas vitales. En el

transcurso de los años se ha insistido en la unificación de la nomenclatura, tanto para la mortalidad como para la morbilidad.

En Cuba las estadísticas médicas tienen alrededor de cuatro siglos de evolución, desde las noticias aisladas en época de la colonia y encontradas en las Actas Capitulares, hasta llegar al estado actual de reconocimiento internacional en la calidad y confiabilidad de los datos ofrecidos.

En el año 1959 se inician las grandes transformaciones políticas, sociales y administrativas en el país. El triunfo de la Revolución conmociona la República y se comienzan a aplicar medidas que conllevan a grandes cambios tales como la denominación del Ministerio de Salubridad y Asistencia Social por la de Ministerio de Salud Pública.

Hasta 1960 no existía en el país un verdadero sistema de estadística de salud. La información era incompleta y la que se obtenía se le daba poca o ninguna utilización. Las deficiencias eran múltiples, por ejemplo en el sub-registro de defunciones, no se observaba la definición estadística de “nacidos vivos”, recomendada por organismos internacionales; el sub-registro de enfermedades transmisibles era tan marcado que en algunas enfermedades se reportaban mayor número de muertos que el número de enfermos; no existía personal calificado dedicado a las estadísticas de salud y se elaboraban los datos en el Nivel Central.

En 1962 se utilizaba la vía de telegramas para la notificación de enfermedades transmisibles. En esa fecha se comienza con el método de listado que se usa hasta 1963 en que se comienza a aplicar el sistema de tarjetas y dicho sistema se mantiene hasta hoy. En 1962 se informa en modelos separados las actividades de hospitales y policlínicos; hasta esta fecha el único modelo existente para dicha información era el No. 10. En 1966, la atención brindada por urgencia, se desglosa del resto de las actividades médicas de las unidades asistenciales. En 1971 se comenzó la docencia de médicos y estomatólogos en la especialidad de bioestadística.

Las estadísticas sanitarias forman parte de un Sistema Nacional que a su vez depende de la Junta Central de Planificación. Las funciones principales de este sistema de información de salud son promover al sistema de salud la información para el diagnóstico, desarrollo y evaluación de los programas sanitarios; brindar asesoramiento técnico de la especialidad en investigaciones de salud y realizar investigaciones administrativas para conocer la situación de los sistemas de información estadística.

A partir de 1964, es que existen normas de organización, funciones, métodos y procedimientos para los departamentos de estadísticas. Los sistemas estadísticos actualmente utilizados son: [2]

- ❖ **De información de actividades de salud:** abarca a los hospitales, policlínicos, unidades docentes y otras unidades. Este sistema comprende movimiento hospitalario, servicios externos, actividades obstétricas, servicios de recién nacidos y actividades docentes.

- ❖ **De natalidad:** a este sistema corresponden todas las inscripciones de nacimientos incluyendo las institucionales, dentro de las propias unidades del MINSAP.
- ❖ **De mortalidad:** se responsabiliza del registro y recolección de los certificados médicos de defunción y defunción fetal, controlando su calidad y codificándolos.
- ❖ **De morbilidad:** se utilizan dos sistemas el de notificación de enfermedades de declaración obligatoria y el de egresos de hospitales.

En resumen, a lo largo de todo el periodo revolucionario, el país ha alcanzado logros significativos en el sector de las estadísticas médicas tales como la formación, organización y desarrollo de las estadísticas de salud y el reconocimiento por los organismos internacionales de la calidad que presentan las Estadísticas de Salud en Cuba.

Sin embargo, el Sistema Estadístico de Notificación de Enfermedades de Declaración Obligatoria posee actualmente una serie de deficiencias que afectan el proceso de vigilancia y control epidemiológico.

La información estadística de las EDO se recoge y coordina manualmente en los departamentos de estadísticas de las unidades de salud y de las direcciones municipales, solamente se gestiona de forma automatizada en los departamentos de estadísticas de las direcciones provinciales y de la dirección nacional.

Precisamente el proceso obtención de relaciones numéricas mediante tablas y gran cantidad de cálculos, a partir de la información estadística capturada, en los niveles de atención de unidad de salud y municipal es sumamente engorroso y está propenso a imprecisiones como consecuencia de los errores humanos.

Para el envío de la información estadística desde las unidades de salud a los niveles superiores se emplea considerable tiempo que retrasan la toma de decisiones por parte de la dirección nacional de salud, de ahí el desfase entre la notificación de las EDO y la planificación de las medidas de control. En este transcurso además, la información se recoge en diferentes modelos, lo que conlleva en ocasiones a la pérdida o duplicidad de los datos.

En los niveles de atención provincial y nacional se utiliza un software implementado en Foxpro 2.6 para Windows. El sistema se encuentra dividido en dos subsistemas, una versión provincial (EDO Prov) y una versión nacional (EDO Nac). En el subsistema provincial se procesa la información enviada de los municipios correspondientes y se recopilan los modelos de entrada de la información, los cuales se confeccionan en las unidades asistenciales. Se reciben dos versiones del Modelo 241-403, en el que se reporta el total de casos de EDO. Este sistema genera un fichero que se compacta y se envía por correo electrónico al nivel nacional, donde en EDO Nac se recoge la información enviada

desde las provincias y se generan tablas de salida similares a las generadas por EDO Prov, en este caso con información referente a las provincias.

El Registro de Enfermedades de Declaración Obligatoria, aplicación desarrollada con herramientas y tecnologías actuales para la gestión de la información relativa a las EDO en el sistema de salud, no dispone de una funcionalidad que permita visualizar los reportes estadísticos.

Para dar solución a la situación existente el mayor esfuerzo está dirigido en **resolver el problema** siguiente: ¿Cómo obtener la información estadística de las Enfermedades de Declaración Obligatoria en el Sistema Nacional de Salud?

Se ha identificado como **objeto de estudio** el proceso de obtención de la información estadística en el Sistema Nacional de Salud y como **campo de acción** el proceso de obtención de la información estadística de las Enfermedades de Declaración Obligatoria en el Sistema Nacional de Salud.

Se ha concebido como **objetivo general** obtener los reportes estadísticos del Registro de Enfermedades de Declaración Obligatoria, favoreciendo el proceso de toma de decisiones en los diferentes niveles del Sistema Nacional de Salud.

Para dar cumplimiento al objetivo definido se trazaron las siguientes **tareas de la investigación**:

- ❖ Analizar los antecedentes sobre el tema a nivel nacional e internacional.
- ❖ Asimilar la arquitectura definida por el MINSAP para el desarrollo de sus aplicaciones: Arquitectura en 3 Capas y Orientada a Servicios y Basada en Componentes (SOA-CBA).
- ❖ Asimilar la Plataforma de Servicios PlaSer, framework sobre el cual la Empresa SOFTEL desarrolla los procesos de implementación.
- ❖ Asimilar el negocio actual del RIS, profundizando en el Registro de Personal de la Salud (RPS), el cual dispone de una funcionalidad que permite visualizar los reportes generales del módulo.
- ❖ Identificar los elementos de integración del REDO con otros componentes del RIS.
- ❖ Analizar las características de la base de datos utilizada por el Registro de Enfermedades de Declaración Obligatoria.
- ❖ Describir las herramientas y tecnologías informáticas para la implementación de los Reportes Estadísticos del Registro de Enfermedades de Declaración Obligatoria.
- ❖ Modelar los Flujos de Trabajo “Modelamiento del Negocio”, “Gestión de Requerimientos”, “Análisis y Diseño” e “Implementación” de los Reportes del REDO.
- ❖ Implementar los Reportes del Registro de Enfermedades de Declaración Obligatoria, utilizando estándares de diseño y codificación definidos por el proyecto SISalud.

En sentido general se podría decir de los Reportes Estadísticos del REDO se esperan un grupo de **beneficios** para la informatización del SNS y en la obtención y procesamiento de las estadísticas de las EDO por el MINSAP, entre ellos se pueden mencionar los siguientes:

- ❖ Gestión de forma automatizada las estadísticas de las EDO en todos los niveles de la dirección de salud.
- ❖ Minimización de las pérdidas y/o de la duplicidad de los datos durante el traslado de los mismos.
- ❖ Erradicación de los errores de cálculo durante el procesamiento de la información estadística.
- ❖ Eliminación del envío de datos mediante modelos y vía correo electrónico desde las unidades de salud hacia los niveles superiores, disponiendo de una base de datos centralizada con la información relacionada a las EDO.
- ❖ Visualización de los Reportes Estadísticos en el REDO, a todos los niveles de la dirección de salud.
- ❖ Reducción del desfase existente entre la notificación de los datos y la toma de decisiones por parte del sistema de vigilancia y control epidemiológico.

Para lograr una organización factible del contenido expuesto en el presente documento, el mismo se encuentra estructurado en cuatro capítulos fundamentales. El Capítulo 1, “**Fundamentación Teórica**”, donde se enmarca al lector en el ambiente de desarrollo de las Estadísticas Descriptivas del Registro de Enfermedades de Declaración Obligatoria, justificándose las tendencias, tecnologías, metodologías y herramientas actuales utilizadas. A continuación el Capítulo 2, “**Características del Sistema**”, contiene el entorno conceptual asociado a la información manipulada por el sistema, llegándose a un acuerdo con el cliente sobre las funcionalidades y requerimientos deseados. El Capítulo 3, “**Análisis y Diseño del Sistema**”, se centra en la modelación detallada y la construcción de la estructura de la aplicación y el Capítulo 4, “**Implementación**”, describe cómo será desplegado la nueva funcionalidad del REDO y se detalla la solución propuesta al cliente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El presente capítulo tiene como objetivo abordar los diferentes elementos que brindan la base teórico conceptual para el desarrollo de las Estadísticas Descriptivas del Registro de Enfermedades de Declaración Obligatoria. Se exponen además los antecedentes e intentos de integración entre sistemas informáticos en el área de la salud pública. Además se realiza un análisis de las técnicas, tecnologías, metodologías y herramientas de software empleado en el proceso de desarrollo.

1.1 Sistema Nacional de Salud

Se conoce como Sistema Nacional de Salud (SNS) a la forma y los métodos que sirven de base para la organización de la atención a la salud en un país determinado. Además es el conjunto de unidades administrativas, de producción, investigación y servicios, responsabilizado con la atención integral de la salud de una población.

El SNS cubano está estructurado en tres niveles de dirección, los cuales se encuentran identificados con la estructura político – administrativa, estos niveles son los siguientes:

- ❖ **Nivel Nacional:** representado por el Ministerio de Salud Pública (MINSAP), como órgano rector con funciones metodológicas, normativas, de coordinación y de control en la aplicación de las políticas del Estado y el Gobierno en cuanto a la salud pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica.
- ❖ **Nivel Provincial:** representado por las Direcciones Provinciales de Salud, directamente subordinadas, administrativa y financieramente a la Asamblea Provincial del Poder Popular.
- ❖ **Nivel Municipal:** representado por las Direcciones Municipales de Salud y dependientes, administrativa y financieramente de la Asamblea Municipal del Poder Popular.

Los diferentes niveles de atención médica se han organizado De acuerdo con la complejidad de las acciones preventivas, curativas y de rehabilitación, así como la especialización de los servicios de salud prestados en:

- ❖ **Atención Primaria de Salud (APS):** Soluciona los problemas de salud de la población que correspondan con las acciones de promoción y protección de la salud. Aunque sus actividades se realizan en cualquier unidad del SNS, están relacionados fundamentalmente con las que se realizan en el Policlínico y en los Consultorios del Médico de la Familia, Hospitales Rurales, Dispensarios y Postas Médicas.

- ❖ **Atención Médica Secundaria:** A este tipo de atención se llega generalmente mediante remisión por parte del médico en la atención primaria. Su función fundamental es tratar al hombre ya enfermo, tanto desde el punto de vista individual como colectivo, pero también desempeña funciones de rehabilitación, promoción y prevención de la salud. Se llevan a cabo acciones de salud más complejas y especializadas. Comprende la atención médica brindada en los distintos Hospitales.
- ❖ **Atención Médica Terciaria:** Abarca los problemas de salud relacionados con secuelas o aumento de las complicaciones de determinadas dolencias. Se brindan servicios de muy alta complejidad, con la óptima utilización de los recursos y medios existentes en los mismos y el desarrollo de la investigación. A este nivel pertenecen los Institutos y Hospitales especializados. Ejemplos de esta atención pueden ser Servicios de Neurocirugía, Nefrología, Cirugía Cardiovascular o Trasplante Renal.

Informatización del Sistema Nacional de Salud

La informatización del SNS es una tarea de primordial importancia, la misma está apoyada en estrategias y políticas trazadas por la dirección del país y el MINSAP. Este proceso está encaminado a crear una infraestructura informática para este sector, al que se integrarán todos los productos y servicios, respondiendo a una Arquitectura en 3 Capas y Orientada a Servicios y Basada en Componentes (SOA-CBA), que permita que todas las unidades de salud del país alcancen un nivel de informatización elevado en las actividades que realizan, influyendo directamente en el aumento gradual de la eficiencia del personal de salud y en la calidad de los servicios que se brindan a la población. Entre las estrategias y políticas a tener en cuenta para la informatización del SNS se encuentran:

- ❖ Adecuar a las condiciones económicas del país las tecnologías de punta y los estándares de calidad desarrollados en el mundo.
- ❖ Desarrollar aplicaciones de forma integrada a una plataforma generalizable para la gestión, procesamiento y transmisión de datos médicos que garanticen la viabilidad, sostenibilidad y mantenimiento dicho sistema de aplicaciones.
- ❖ Los productos se desarrollarán basados en tecnología LAMP (Linux, Apache, MySQL y PHP) para garantizar sus sostenibilidad en el tiempo y se emplearán estándares internacionales para productos relacionados con la salud pública.

Esta solución integral para la informatización significa la articulación de un nuevo paradigma para la prestación de servicios de salud, regido por el principio básico de lograr la descentralización,

permitiendo que estos puedan ser accesibles por la población en su propio escenario social. Los principales impactos esperados son los siguientes:

- ❖ 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, que permitirá una rápida transferencia de la información sanitaria de un paciente.
- ❖ Disponer de un soporte y herramientas poderosos para la formación y actualización constante de los miembros del SNS, desde sus propios escenarios de desempeño, con la consecuente equidad de conocimientos independientemente de áreas geográficas y nivel de atención.
- ❖ Equidad en el acceso de la población a servicios, tecnologías e información de salud, independientemente del área geográfica y nivel de atención.
- ❖ Atención a la población por un personal médico mejor preparado y actualizado, entre los cuales se podrá intercambiar información, criterios y diagnósticos; elevándose la confianza hacia el sistema de atención.
- ❖ Reducción del número de desplazamientos innecesarios entre instituciones de salud y del tiempo de espera para el acceso de la población a servicios especializados.

Registro Informatizado de Salud

En sus inicios la informatización del SNS no ofrecía un mecanismo único para la integración de aplicaciones. Las instituciones del sector de la salud pública poseían un conjunto de aplicaciones que brindaban solución a determinados problemas, pero estos sistemas se comportaban de forma aislada al no interactuar entre sí para obtener un flujo lógico y coherente de la información.

En 1997, empresas como SOFTEL, CENTERSOFT y VIRTUS, pertenecientes al Ministerio de la Informática y las Comunicaciones (MIC) y dedicadas al desarrollo de aplicaciones para el sector de la salud pública, vieron en la integración de sus soluciones una alternativa para satisfacer exigencias para la comercialización de este tipo de producto. El principal exponente de integración de aplicaciones para la salud es el Registro Informatizado de Salud (RIS).

El surgimiento de Servicios Web basados en XML implicó la aparición de variantes mucho más factibles para la integración. El RIS constituyó la aplicación pionera en la utilización de esta tecnología de avanzada, debido a las ventajas que ofrecía para implementar cómputo distribuido, arquitectura orientada a servicios y basada en componentes. Además permitiría la reutilización, la no duplicación de esfuerzos y el logro de la confiabilidad e integridad de la información clínica que se gestionaría. El RIS

fue implementado a finales del año 2003 por cuatro grupos de trabajo en la Empresa SOFTEL, integrados por especialistas de esta entidad, de la empresa ESATEL de Santiago de Cuba, del Centro para el Desarrollo Informático de la Salud (CEDISAP) e INFOMED.

Los módulos que fueron desarrollados se denominaron Registro de Equipos Médicos (REM), Registro de Equipos no Médicos (RENM), Registro de Personal de la Salud (RPS), Registro del Ciudadano (RC), Registro de Unidades de Salud (RUS), Registro de Ubicación Geográfica (RU) y el módulo de Seguridad y Administración (SAAA).

Sistema de Información para la Salud

En el año 2004 se concibe el Sistema de Información para la Salud (SISalud), infraestructura que permitirá la integración de los componentes, servicios y sistemas que se desarrollen para darle cumplimiento al programa de informatización del SNS. Este aborda los problemas del sector de forma integral e innovadora, satisfaciendo las necesidades de los clientes en todos los niveles de salud, desde la Atención Primaria hasta las redes de Especialidades en los diferentes Institutos y Hospitales.

1.2 Sistemas para la Vigilancia y Control Epidemiológico

Los primeros sistemas automatizados para la detección e intervención urgente en situaciones de emergencia surgen a finales del siglo pasado, con el fin de desarrollar las estrategias de investigación y establecer las medidas de control necesarias, proporcionando a las autoridades sanitarias información oportuna y retroalimentación periódica. La característica más importante de estos sistemas lo constituye la rapidez en la detección. Muchos de ellos funcionan en tiempo real, o sea, al mismo tiempo que se recibe la información se van disparando los mecanismos para la alerta.

Antecedentes Internacionales

En el ámbito internacional no son pocos los sistemas automatizados de alerta desarrollados para la vigilancia y control de enfermedades específicas, entre los más importantes: [3]

LEADERS (Lighthweight Epidemiology Advanced Detection & Emergency Response System): Uno de los primeros sistemas automatizados de alerta fue desarrollado por la Agencia de Proyectos de Investigación para la Defensa de los Estados Unidos en 1993, el sistema LEADERS, al cual en años posteriores se le añadieron aplicaciones para extenderlo al sector civil. Este sistema utiliza información de los registros hospitalarios de varios hospitales de emergencia seleccionados y rastrea el reporte de síntomas inusuales.

RODS (Real Time Outbreak and Disease Surveillance): La primera versión de este sistema fue creado por investigadores norteamericanos de la Universidad de Pittsburg en 1999, con la finalidad de mejorar las capacidades de respuesta de las autoridades sanitarias frente a una amenaza muy específica: el *Bacillus anthracis*. Este sistema recolecta información de las principales causas de admisión en los servicios de urgencia de hospitales seleccionados y en tiempo real clasifica estas causas en categorías de síndromes basado en la Novena Revisión de la CIE y alerta cuando existen anomalías que pueden ser brotes epidémicos.

GPHIN (Global Public Health Intelligence Network): Red de Información Mundial en Salud Pública, un sistema de alerta temprana, seguro, basado en Internet que busca reportes preliminares significativos de problemas de salud, en tiempo real, 24 horas al día, los 7 días de la semana. Este sistema único y multilingüe busca y provee información importante no verificada acerca de brotes de enfermedades y otros eventos en salud pública, monitoreando los medios de comunicación a escala mundial en seis idiomas, árabe, chino, inglés, francés, ruso y español. Este proceso automático incluye la filtración y categorización de la información por importancia que finalmente es completado por análisis humano. Fue creado en Canadá y es utilizado por la OMS desde 1997.

Virgil: Este proyecto de Vigilancia de la Resistencia Viral fue presentado por el Centro de Investigaciones del Instituto Nacional de Investigaciones Médicas Científicas (INSERM) de Francia en Junio del 2004, producto de la colaboración de investigadores y académicos de toda Europa y constituye una red de información dirigida a combatir las nuevas pandemias con tecnologías que pueden monitorear la droga resistencia de las cepas virales existentes y la susceptibilidad de las nuevas cepas emergentes y desencadenar alertas. Este proyecto comenzó con la vigilancia de la hepatitis viral y la influenza y tiene el propósito de extenderse a otras infecciones virales.

Sentiweb: En 1995 Francia hizo público el sitio Sentiweb, con información de su Sistema Centinela de Vigilancia que funciona desde 1984, con una muestra de médicos generales (1%) del país conectados mediante una computadora personal y un MODEM, los que transmiten la incidencia de varias enfermedades transmisibles seleccionadas a un servidor, donde se almacenan automáticamente en una base de datos, que puede ser consultada y que ofrece dos tipos de salidas: series de tiempo de enfermedades solicitadas para el período y región de Francia expresado como mapas y curvas de incidencia.

DengueNet: Sistema automatizado en tiempo real basado en Internet creado por la OMS para la vigilancia global del dengue y el dengue hemorrágico en mayo de 2003, permite fomentar y fortalecer la capacidad de los sistemas de salud para la vigilancia, la prevención, control y tratamiento. Las principales características de este sistema de vigilancia son: servicio protegido por contraseña que permite la introducción de datos a distancia, inclusión de subdivisiones por estado o provincia de los

países, para los que se introducen o se calculan indicadores, dispositivo dinámico de consulta y preguntas, con el análisis y la presentación de los datos en gráficos, tablas, textos y utilización de herramientas SIG para elaborar mapas. Dispone de series de datos recopilados desde 1955.

FluSurge: Los Centros para el Control y Prevención de Enfermedades (CDC) de Estados Unidos propusieron en julio del 2004 para contribuir a la preparación contra la pandemia de influenza, el software FluSurge 1.0, que estima el número de muertes y hospitalizaciones que pueden producirse en una epidemia cuya duración, virulencia y extensión puede determinar el usuario, así como el número de personas que requerirán ingreso en unidades de cuidados intensivos y necesitarán ventilación asistida.

Sistema de Información en el Sistema Nacional de Vigilancia de Salud: Surge en Argentina con el propósito de automatizar totalmente el ingreso de datos, el análisis epidemiológico básico (tablas y gráficos) de comportamiento de los daños bajo vigilancia y monitorea el comportamiento de los atributos (aceptabilidad, oportunidad, cobertura) del sistema. [4]

Antecedentes en Cuba

En Cuba se han desarrollado también sistemas destinados para la vigilancia y control de determinadas enfermedades: [5]

SAVT (El Sistema Automatizado de la Vigilancia de la Tuberculosis): este sistema fue uno de los primeros confeccionados en el país por investigadores del Instituto de Medicina Tropical "Pedro Kourí" en 1994.

VIGILA: (Software para la vigilancia epidemiológica de enfermedades transmisibles): permite el registro y mantenimiento de datos con sencillez, creado en 1998 en el Instituto de Medicina Tropical "Pedro Kourí". Permite pronosticar y detectar con una alta sensibilidad la aparición de situaciones epidémicas, convirtiéndose en un instrumento para la alerta y la acción; puede elaborar reportes y gráficos para interpretar datos y comparar los eventos de salud entre los territorios de la región que se vigila, con la incorporación de técnicas de análisis geográfico; permite observar varias enfermedades de manera simultánea y transferir reportes, gráficos, mapas y pronósticos resultantes del trabajo hacia otras aplicaciones del entorno operativo Windows.

WINEDOS (Sistema automatizado para el análisis de las bases de datos de las enfermedades de declaración obligatoria): fue desarrollado en la UATS de la provincia Las Tunas, en el año 2000. Fue programado en Microsoft Access 97 para Windows 95. La aplicación entre sus funciones tiene captar datos preliminares, importar datos que se oficialicen en los departamentos de estadísticas, captar poblaciones para el cálculo de las tasas, generar canales por diferentes tipos de métodos de

cálculos, tablas o reportes de salida con tasas, proporciones y el diagnóstico de una epidemia por municipios y provincia, con frecuencia semanal.

SIVD (Sistema Integrado de Vigilancia de Dengue): Es un sistema que se utiliza desde el año 2005 en los municipios Cotorro y en Centro Habana. Permite agilizar en gran medida el procesamiento de la información obtenida mediante la recogida de información clínico-epidemiológica de los pacientes que eran diagnosticados como sospechosos de dengue, además de información ambiental recogida por parte de los compañeros de la Campaña de Lucha Antivectorial y de datos entomológicos provenientes del laboratorio a tal efecto. [6]

1.3 Tendencias y Tecnologías Actuales a considerar

En el epígrafe se tratarán los conceptos principales relacionados con la arquitectura, herramientas, tecnologías y metodología que se considerarán para el proceso de desarrollo de los Reportes del Registro de Enfermedades de Declaración Obligatoria. Los temas más significativos son los patrones o estilos, tanto de diseño como arquitectónicos. Estos patrones no serán expuestos agrupados según su clasificación natural, diseño o arquitectura, sino como una secuencia lógica que permita justificar el por qué de su utilización.

Modelos Arquitectónicos. Modelo de Arquitectura WEB

El Modelo de Arquitectura WEB (Intranet o Extranet) es una nueva tecnología concebida para cubrir las necesidades de organismos públicos que utilizan información territorial. Este modelo arquitectónico se puede utilizar para explotar y mantener sistemas corporativos con ventajas económicas, funcionales y de difusión de la información, sin embargo esta tecnología ha de evolucionar para ser plenamente operativa aunque es probable que se trate de un periodo relativamente corto. [7]

Sistemas distribuidos. Modelo Cliente-Servidor

Actualmente, entre las estrategias concebidas para el proceso de informatización de la salud se destaca la inserción de los sistemas informáticos para la salud en un ambiente donde coexisten numerosos sistemas heterogéneos y distribuidos en diferentes ubicaciones físicas. Por ello es inevitable la integración de datos, que no es más que combinar información que se encuentra almacenada en componentes con estas características, proporcionado al usuario final una vista unificada de todos estos datos.

En el contexto actual de las tecnologías, se no hace obligatorio el uso de aplicaciones distribuidas, aunque ha de tenerse en cuenta en este proceso la utilización de políticas de

compatibilidad, estándares, escalabilidad y tecnologías. Cabe destacar que la disminución de los precios de los computadores y el desarrollo acelerado de las redes de ordenadores trajo consigo la amplia difusión y aceptación de estos sistemas distribuidos. Algunas ventajas de los sistemas distribuidos sobre los centralizados de mayor impacto son:

- ❖ **Modularidad y Reutilización en el desarrollo:** Esto permite la agilización de los procesos de desarrollo pues se tendrá en cuenta para la implementación de un nuevo componente y que funcionalidades podrá obtener de otro ya desplegado.
- ❖ **Fiabilidad:** En caso de fallas de un sistema, la aplicación continuaría brindando los servicios que no fueron afectados, y no dejaría de funcionar.
- ❖ **Rapidez de respuesta y rendimiento:** Recursos compartidos, de manera que se pueda procesar una petición de manera mucho mas rápida.
- ❖ **Escalabilidad:** Crece de una manera sencilla, basta solo con incrementar el número de servidores que conforman el sistema.

Estos no descartan ciertas desventajas en su uso, una de las más llamativas, sin dudas, es la existencia de incompatibilidades en la información distribuida y la falta de integridad referencial en la misma, al no considerarse en el proceso de establecimiento y definición de la arquitectura la forma de solucionar esta situación. Por tanto el proceso de administración es mucho más complejo en este tipo de aplicaciones.

Los sistemas distribuidos más populares son las que se basan en la Arquitectura Cliente – Servidor, en la que una parte de estos, los servidores, proveen los servicios y los clientes acceden a estos. Lo que es lo mismo decir, que el Cliente es el encargado de permitirle al usuario formular los requerimientos y enviarlos al Servidor y este a su vez se encarga de atender los múltiples clientes que le hacen las peticiones de recursos administrados por él. El servidor es el encargado, de forma general, de las reglas del negocio y los accesos a datos. Entre las principales ventajas de la Arquitectura Cliente – Servidor se pueden apreciar: [8]

- ❖ Reducción considerable del tráfico de datos en la red, ya que los clientes se conectan cuando requieren un servicio y luego se desconectan, dejando libre la conexión.
- ❖ La estructura independiente facilita la integración entre los sistemas heterogéneos y el crecimiento continuo de la infraestructura tecnológica, favoreciendo la escalabilidad de las aplicaciones.
- ❖ Rápida construcción y fácil mantenimiento de las aplicaciones, pues se hace uso de componentes y servicios ya implementados.

La desventaja fundamental de esta Arquitectura radica en que si la congestión en la red y el tráfico en la misma son elevados puede provocar un pobre desempeño del sistema.

Patrones de Arquitectura y Diseño

Los patrones arquitectónicos y de diseño han sido definidos para crear un lenguaje común, entre desarrolladores, que generalice sobre determinado problema. Existen varios tipos de patrones, dependiendo del contexto particular en el cual se aplican o de la etapa en el proceso de desarrollo, algunos de estos tipos son: de Arquitectura, de Diseño, de Negocios, de Análisis, para Ambientes Distribuidos, entre otros.

Arquitectura en 3 Capas

Esta arquitectura constituye una generalización de la Cliente – Servidor, donde la carga ha de dividirse en tres partes con un reparto de las funciones: la Capa de Presentación, la Capa de Negocio y la Capa de Datos. Esto permite un alto nivel de abstracción, logrando que los desarrolladores la fragmentación del problema complejo en pasos a seguir. También se alcanza una amplia reutilización, pues los datos abstractos se pueden emplear por diferentes implementaciones.

Arquitectura Orientada a Servicios y Basada en Componentes

El surgimiento de la SOA (Service Oriented Architecture) se produce como una estrategia de integración, exponiendo funcionalidades a las aplicaciones que lo requieran, como respuesta a la exigencia creciente de empresas por la creación de aplicaciones mucho más complejas, en menor tiempo y costos. Cada aplicación nueva creada funciona de forma independiente, brinda servicios y los recibe, cada una tiene sus propias reglas del negocio, datos, procedimientos, administración y operaciones. Las ventajas de la SOA más significativas son: [9]

- ❖ **Interoperabilidad de las aplicaciones:** Disminuye de forma considerable la complejidad de los procesos de integración, pues interactúan con elementos que se abstraen de la tecnología y la ubicación de los servicios.
- ❖ **Reusabilidad:** Reducción considerable de costos y tiempo de desarrollo de las aplicaciones, pues ya están implementados algunos de los servicios necesarios. Aumenta de esta forma la robustez del nuevo sistema, por el uso de software ya probado anteriormente.

En tanto, el desarrollo de software basado en componentes se centra en el desarrollo de aplicaciones complejas mediante el ensamblado de otros sistemas, que han sido previamente diseñados por otras personas a fin de ser reusados en múltiples aplicaciones. Cada componente debe

describir de forma completa la interfaz que ofrece, así como las interfaces que requiere para su operación y debe funcionar correctamente con independencia de los mecanismos internos que utilice para su funcionalidad. Entre las principales ventajas de la CBA se destacan: [10]

- ❖ **Mayor Calidad:** Los componentes son construidos y luego mejorados por expertos y/u organización, esto hará que la aplicación mejore continuamente.
- ❖ **Reutilización de Software:** La Arquitectura lleva a un mayor nivel de reutilización.
- ❖ **Simplificación de las Pruebas:** Permite realizar pruebas por separado a cada una de las aplicaciones, antes de probar el sistema completamente ensamblado.
- ❖ **Simplificación del Mantenimiento:** Cuando se esta en presencia de de un débil acoplamiento el desarrollador se siente libre de actualizar y/o agregar componentes según sea necesario, sin afectar las otras partes.

Servicios Web basados en XML

La comunicación mediante un servicio, es realizada utilizando mensajes y no llamadas a métodos, como se realiza en el paradigma de la Programación Orientada a Objetos. Estos mensajes deben contener o referenciar toda la información necesaria para ser comprendidos internamente por la aplicación que lo ha solicitado. Un Servicio Web es un sistema de software diseñado para soportar interacción máquina a máquina los cuales pueden ser descritos, publicados, localizados e invocados a través de la red utilizando protocolos estándar.

Los Servicios Web utilizan SOAP (Simple Object Access Protocol) como protocolo para el invoke de llamadas remotas. Los mensajes SOAP son identificados como documentos XML (eXtensible Markup Language) conformado por envoltura obligatoria, el encabezamiento y el cuerpo del mensaje. SOAP, además, permite la comunicación entre aplicaciones diferentes sin importar las diferencias de Lenguaje de Programación o Plataformas usadas.

Alta Cohesión y Bajo Acoplamiento

Con la alta cohesión se garantiza que la información que se gestione con un servicio web, debe ser coherente y estar lo más posible relacionada con la información proporcionada por este. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos. Un ejemplo de baja cohesión son servicios que realizan demasiadas tareas. En todas las metodologías de desarrollo se considera la factorización, proceso que permite la creación de los denominados paquetes de servicio.

El bajo acoplamiento consiste en tener los componentes y servicios que estos brindan lo menos ligados entre sí, de tal manera en caso de que se produzca alguna modificación en alguno, tenga la misma repercusión en el resto de los servicios que conforman el componente, ganando en reutilización y disminuyendo la dependencia de otros.

Modelo Vista Controlador

El Modelo Vista Controlador (MVC) permite la separación de los datos de una aplicación, la interfaz de usuario y la lógica del negocio en tres componentes distintos, esto proporciona múltiples vistas del mismo modelo de datos. Es muy usado en aplicaciones Web donde se emplean diferentes interfaces de usuario y el código que provee los datos a la página es dinámico. La responsabilidad de cada uno de sus elementos fundamentales: [11]

- ❖ **Modelo:** Se encarga de la administración del comportamiento y los datos del dominio de la aplicación, responde a los requerimientos de información sobre su estado, usualmente formulados desde la vista, respondiendo a instrucciones de cambio, habitualmente desde el controlador.
- ❖ **Vista:** Presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.
- ❖ **Controlador:** Responde a eventos, generalmente acciones del usuario e invoca a cambios en el modelo y probablemente en la vista.

Tanto la vista como el controlador dependen del modelo, el cual no depende de estos conceptos o clases. Entre las ventajas de mayor impacto del MVC: [12]

- ❖ **Soporte de múltiples vistas:** No hay dependencia de la vista con respecto al modelo, dado que la vista se halla separada del modelo. La interfaz del usuario puede mostrar múltiples vistas de datos de manera simultánea.
- ❖ **Adaptación al cambio:** Agregar nuevas opciones y/o modificar la representación, de las ya existentes, generalmente no provoca afectaciones al modelo pues este no depende de la vista.

Sus desventajas esenciales radican en: [13]

- ❖ **Complejidad:** Introduce nuevos niveles de indirección y esto aumenta la complejidad de la solución. También se profundiza la orientación a eventos del código de interfaz de usuario, que puede llegar a ser difícil de depurar.

- ❖ **Costo de actualizaciones frecuentes:** Desacoplar el modelo de la vista no significa que los desarrolladores del modelo puedan ignorar la naturaleza de las vistas.

Frameworks en PHP

Los Frameworks ayudan en el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez y a la hora de realizar el mantenimiento del sitio gracias a la organización durante el desarrollo de la aplicación. Son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos. Se utiliza la Programación Orientada a Objetos (POO), permitiendo la reutilización de código. A diferencia de ASP.NET, los Frameworks en PHP no son desarrollados y comercializados por Microsoft. A continuación se mencionan algunos entre los más populares y se describen las características de los más utilizados: [14]

Zend Frameworks: es simple, no necesita instalación especial, requiere PHP 5 e incorpora el patrón MVC.

Symfony: diseñado con el objetivo de optimizar la creación de las aplicaciones web, con el uso de sus características. Posee una librería de clases que permiten reducir el tiempo de desarrollo. Está desarrollado en PHP5, se puede utilizar en plataformas Unix, Linux y Windows. Requiere de una instalación, configuración y líneas de comando, incorpora el patrón MVC, soporta AJAX, plantillas y un gran número de bases de datos.

CodeIgniter: CodeIgniter es un buen framework, utilizado por una gran comunidad de usuarios. Construido para codificadores PHP que necesitan una herramienta de desarrollo fácil para crear aplicaciones web simples y elegantes. Entre sus características se puede encontrar su compatibilidad con PHP 4 y PHP 5, incorpora el modelo MVC, soporte para múltiples bases de datos, plantillas, validaciones, no requiere instalación, se puede encontrar una librería con un gran número de clases.

Plataforma de Servicios

Plaser: Acrónimo de Plataforma de Servicios, es una librería para el desarrollo de componentes utilizando el paradigma de los Servicios Web basados en XML. Este sistema está concebido completamente sobre Arquitectura Basada en Componentes y Orientada a Servicios, usando el paradigma de XML Web Services, específicamente SOAP. En su concepción se han utilizado estándares actuales y normas abiertas. Todo el código ha sido programado en PHP 4 y entre los componentes que lo conforman pueden ser mencionados:

- ❖ **Plaser_Client:** Manipula las llamadas mediante SOAP mediante la biblioteca Pear SOAP.

- ❖ **Plaser_XML:** Manipula el documento XML que devuelve el Plaser_Client e interconecta este con Plaser_XSLT.
- ❖ **Plaser_XSLT:** Transforma el XML hacia HTML aplicando para ello el XSLT correspondiente.
- ❖ **Plaser_DBz:** Manipula las operaciones con la Base de Datos utilizando la extensión DBX para la abstracción del acceso a la información.

Desde el punto de vista estructural permite trabajar con cualquier base de datos que cumpla con el estándar SQL-92; pero desde el punto de vista de implementación solo trabaja con las bases de datos soportadas por el componente DBX, ya que encapsula a dicho componente y lo utiliza para el acceso a bases de datos.

PlaSer, en su configuración ideal, se distribuye en tres servidores, los cuales se encuentran conectados en cascada y solo el primero esta conectado a Internet con una IP real. Este primero contiene la capa de presentación (layout), el segundo contiene a ProxPla y los demás componentes, incluido el SAAA (Single Authorization Authentication and Account) componente de seguridad, que autentifica y autoriza, además tiene registrados los nombres de los métodos. El tercer servidor contiene las bases de datos desarrolladas en MySQL. Para aumentar la seguridad se pudiera colocar a ProxPla solo en un cuarto servidor separado de lo demás componentes. También PlaSer permite ser desplegado totalmente en un solo servidor, aunque esto no es recomendable por motivos de seguridad. El Sistema de Seguridad de PlaSer comprende 3 capas o “rejillas”:

- ❖ Primera Rejilla: Niveles
 - Nivel 1: Nacional
 - Nivel 2: Provincial
 - Nivel 3: Municipal
 - Nivel 4: Unidades de Salud
- ❖ Segunda Rejilla: Tipos de Usuarios
 - Administrador: Manipula los usuarios
 - Editor: Modifica información en el Sistema.
 - Visualizador: Solo puede visualizar información.
- ❖ Tercera Rejilla: Especifica a que módulo se pertenece

Lenguajes de Programación Utilizados

En la sección siguiente se realizará un análisis de los lenguajes candidatos a ser utilizados para el proceso de implementación de las capas de negocio y presentación de los Reportes Estadísticos del REDO. En todos los casos se abordará la definición de estos y se expondrán algunas de sus características así como las ventajas que influyeron en la elección de los finalmente empleados.

Lenguaje de Programación del lado del Cliente. JavaScript

JavaScript es un lenguaje de programación interpretado, es decir no requiere compilación. Su código es embebido directamente en el código HTML o similares, haciendo fácil la creación de páginas Web con contenido dinámico. Está diseñado para controlar la apariencia y manipular los eventos de los objetos contenidos en un formulario Además de ser soportado por la mayoría de los navegadores existentes. El mismo comparte muchos elementos con otros lenguajes de alto nivel siendo muy semejante a PHP, tanto en su formato como en su sintaxis. También se debe tener en cuenta que los programas escritos en este lenguaje tienden a ser pequeños y compactos, lo cual influye en que no se requiera ni memoria ni tiempo adicional para su transmisión.

Lenguajes de Programación del lado del Servidor

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET esta construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework. Esta tecnología presenta facilidades importantes que la convierten en un framework ventajoso:

- ❖ **Velocidad:** ASP.NET es un lenguaje compilado lo que hace que su ejecución sea mucho más rápida. PHP por ejemplo es un lenguaje interpretado lo cual significa que el servidor interpreta el código cada vez que lo va a utilizar.
- ❖ **Controles:** Visual Studio.NET posee validadores para los campos de entradas, tablas que se pueden conectar de manera automática a las bases de datos y hasta un control que permite configurar, también de manera automática, el login de la página. Además, todos estos controles se pueden adornar de manera muy sencilla haciendo clic con el botón derecho sobre ellos y seleccionando "Estilo", todo en una elegante ventana que permite controlar los tipos de letra, bordes, fondos, etc.

Por otro lado, ASP.NET posee significativas desventajas:

- ❖ **Componentes:** necesita integrarse en Windows casi al mismo nivel que un programa de ejecución local, para entonces utilizar cualquier componente hecho para Windows, las librerías dll, las APIs de Windows, etc. Registrar las librerías en el sistema operativo es un gran

problema para una aplicación, la alternativa es copiar las dll en el directorio bin de la aplicación para activar las funciones.

- ❖ **Seguridad:** es totalmente dependiente de los Services Packs de Microsoft y la actualización de los servidores por parte de los webmasters.
- ❖ **Acceso a datos:** es necesario emplear Open DataBase Connectivity (ODBC), opción desarrollada por Microsoft para la conexión a bases de datos.
- ❖ **Escalabilidad:** está dada por el concepto de granja de servidores que tiene integrado IIS.

Hypertext Pre-processor (PHP): es un lenguaje de programación interpretado, creado en 1994 por Rasmus Lerdof, utilizado generalmente para la creación de sitios, contenido dinámico para aplicaciones Web y aplicaciones para servidores.

Generalmente los scripts PHP se embeben en otros códigos como HTML, ampliando las posibilidades del diseñador de páginas Web enormemente. La interpretación y ejecución de estos scripts se hacen en el servidor, el cliente (solicitud realizada desde un navegador Web) solo recibe el resultado y jamás ve el código PHP. Permite conexión con todo tipo de bases de datos como MySQL, Postgre SQL, Oracle, DB2, Microsoft SQL Server, Firebird y SQLite. PHP corre sobre 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte para varios Sistemas Gestores de Bases de Datos (SGBD) y contiene unas 40 extensiones estables. Actualmente se encuentra en su versión 5 y posee un número de ventajas importantes:

- ❖ Integración con varias bibliotecas externas, lo que permite generar documentos Portable Document Format (PDF) y Microsoft Office Excel (XLS).
- ❖ Ofrece una solución simple y universal para las paginaciones dinámicas de fácil programación.
- ❖ Soportado por una gran comunidad de desarrolladores, como producto de código abierto, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente, implicando menos costos.
- ❖ Gran número de funciones predefinidas. A diferencia de otros lenguajes de programación, PHP fue diseñado especialmente para el desarrollo de páginas Web dinámicas. Por ello, está dotado de un gran número de funciones que simplificará enormemente tareas habituales como descargar documentos, envío de correo electrónico, creación dinámica de imágenes y gráficos en el servidor, procesamiento de información en formularios, manipulación de cookies y sesiones, transporte de información mediante HTTP y análisis de documentos XML.
- ❖ Análisis léxico para reconocer el tipo de dato almacenado en una variable haciéndose automáticamente, permitiéndole al usuario no tener que separar las variables de sus valores.

- ❖ Posee un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos desde el cliente evitando por ejemplo, la ocurrencia de la conocida inyección de código SQL.

Aunque, también posee algunas desventajas, entre ellas se pueden mencionar que: todo el trabajo se realiza en el lado del servidor, no delegando responsabilidades al cliente, puede que en un momento determinado la capacidad de respuesta sea ineficiente en la medida en que las solicitudes al servidor aumenten considerablemente. Además si el código PHP se incluye en código HTML, puede que su legibilidad se vea afectada. La Programación Orientada a Objetos (POO) en PHP es aún deficiente para aplicaciones de gran envergadura influyendo sobre el rendimiento de estos sistemas.

Lenguajes de Marcas

En sentido general estos son una forma de codificar un documento que, junto con el texto, incorpora marcas que contienen información adicional acerca de la estructura del texto o su presentación. Proporcionan un conjunto de etiquetas o tags, que se entremezclan con el propio contenido en un único archivo o flujo de datos. Se suele diferenciar entre tres clases de lenguajes de marcado, aunque en la práctica pueden combinarse varias clases en un mismo documento siendo estas el marcado de presentación, el marcado procedimental y el marcado descriptivo o semántico, los cuales se comentan a continuación:

- ❖ **Marcado de Presentación:** Es aquel que indica el formato del texto. Este tipo de marcado es útil para maquetar la presentación de un documento para su lectura, pero resulta insuficiente para el procesamiento automático de la información. El marcado de presentación resulta más fácil de elaborar, sobre todo para cantidades pequeñas de información. Sin embargo resulta complicado de mantener o modificar, por lo que su uso se ha ido reduciendo en proyectos grandes en favor de otros tipos de marcado más estructurados.
- ❖ **Marcado de Procedimientos:** Está enfocado hacia la presentación del texto, sin embargo, también es visible para el usuario que edita el texto. El programa que representa el documento debe interpretar el código en el mismo orden en que aparece. Por ejemplo, para formatear un título, debe haber una serie de directivas inmediatamente antes del texto en cuestión, indicándole al software instrucciones tales como centrar, aumentar el tamaño de la fuente, o cambiar a negrita. Inmediatamente después del título deberá haber etiquetas inversas que reviertan estos efectos. En sistemas más avanzados se utilizan macros o pilas que facilitan el trabajo.

- ❖ **Marcado descriptivo o semántico:** Utiliza etiquetas para describir los fragmentos de texto, pero sin especificar cómo deben ser representados o en qué orden, lo cual le da al mismo una gran flexibilidad. El marcado descriptivo también simplifica la tarea de reformatear un texto, debido a que la información del formato está separada del propio contenido.

HTML es el acrónimo de **Hypertext Markup Language** (Lenguaje de marcado de hipertexto), creado en 1989 por Tim Berners-Lee. Es originalmente un subconjunto de SGML (Standard Generalized Markup Language), especializado en la descripción de documentos en pantalla.

El proyecto inicial se basaba en una colección de etiquetas que permitían describir documentos de texto y vínculos de hipertexto que hacían posible el desplazamiento en forma jerárquica entre diferentes documentos. La facilidad de su uso y la particularidad de no ser propiedad de nadie, hizo de HTML el sistema idóneo para compartir información a través de Internet. Inicialmente su intención era que las etiquetas fueran capaces de marcar la información de acuerdo a su significado, pero por diversos motivos los creadores de los navegadores Web fueron añadiendo más etiquetas HTML, dirigidas a controlar la representación de la información contenida en el documento.

Extensible Markup Language (XML), establecido en febrero 1998, no es más que un metalenguaje de marcado descriptivo, que posee un grupo de reglas que le permite crear sus propios elementos de marcación, los cuales pueden usarse después para describir su contenido. XML se desarrolló porque HTML no estaba diseñado para describir algunos tipos de datos que las personas querían enviar a través de la red, pues este último tenía sus etiquetas predefinidas. Proporciona la flexibilidad y consistencia que no se podía alcanzar con HTML, pues no se tiene que forzar al contenido para que se ajuste al grupo limitado de elementos proporcionados por este. En pocas palabras, XML proporciona un mecanismo estándar para describir cualquier tipo de datos en documentos que son altamente transportables y reutilizables. [15]

HTML carece de grandes cantidades de marcas semánticas por lo que dificulta la descripción y el intercambio de datos y la mayoría de los documentos especificados en este lenguaje, en la actualidad, están plegados de código innecesario lo que hace que la velocidad de descarga aumente considerablemente. Desde finales de 1998, ha habido una migración hacia tecnologías relacionadas con XML. Su éxito no fue una sorpresa, pues este lenguaje permite a los desarrolladores la capacidad de crear un vocabulario personalizado, a la vez que separa la presentación de la estructura, lo que hace que el mantenimiento y el análisis gramatical sea más fácil.

XHTML es el acrónimo de **Extensible Hypertext Markup Language** (Lenguaje de marcado de hipertexto extensible). No es más que una reformulación de HTML 4.01 de acuerdo a las reglas de XML 1.0, en realidad está diseñado para que HTML vuelva a sus raíces y lo convierta en un grupo de

marcación que describa una clase en particular de contenido sin importar cómo se visualizará este. Debido a que está especificado de acuerdo a XML, puede beneficiarse de muchas tecnologías y herramientas que están desarrolladas para trabajar con XML. XHTML 1.0 se convirtió en la recomendación de World Wide Web Consortium (W3C) en enero del 2000.

Aunque XHTML es compatible regresivamente con los navegadores Web actuales y no es muy diferente a HTML 4.01, muchos desarrolladores todavía se resisten a cambiar porque no ven el valor agregado que ofrece una herramienta con un grupo más estricto de especificaciones de sintaxis.

XHTML podrá de una manera natural beneficiarse con infraestructuras desarrolladas para XML, como por ejemplo XSL (**Extensible Stylesheet Language**, Lenguaje de hojas de estilo extensible) y XSLT (**XSL Transformations**, Transformaciones XSL). XSL es al igual que CSS (Cascading Style Sheets, Hojas de estilo en cascada) un lenguaje de hojas de estilo, pero a diferencia de CSS permite realizar transformaciones complejas o simples de documentos, a la vez que les da formato. Es decir, XSL consta de dos etapas, la primera de ellas un proceso de transformación, esbozado en la especificación de XSLT y en segundo lugar, un proceso para dar formato, definido en la especificación principal de XSL. [16]

Gestor de Bases de Datos

Un Sistema de Gestión o Manejador de Bases de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una Base de Datos. Por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Existen muchas formas de organizar las bases de datos, pero hay un conjunto de objetivos generales que deben cumplir todos los SGBD, de modo que faciliten el proceso de diseño de aplicaciones y que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios. Sus objetivos fundamentales son:

- ❖ **Abstracción de la información:** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- ❖ **Independencia:** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- ❖ **Redundancia mínima:** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no

obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

- ❖ **Consistencia:** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- ❖ **Seguridad:** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- ❖ **Integridad:** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- ❖ **Respaldo y recuperación:** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- ❖ **Control de la concurrencia:** En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
- ❖ **Tiempo de respuesta:** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en dar la información solicitada y en almacenar los cambios realizados.

Los SGBD poseen grandes ventajas entre las que se destacan:

- ❖ Facilidad de manejo de grandes volúmenes de información.
- ❖ Gran velocidad de ejecución de las consultas.
- ❖ Independencia del tratamiento de información.
- ❖ Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.

- ❖ No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- ❖ Integridad referencial al terminar los registros.

Aunque se pueden tornar desfavorables, entre otras, por las siguientes causas:

- ❖ El costo de actualización del hardware y software son muy elevados.
- ❖ El mal diseño de esta puede originar problemas a futuro.
- ❖ Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
- ❖ Generan campos vacíos en exceso.
- ❖ El mal diseño de seguridad genera problemas en esta.

Microsoft SQL Server: es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL o MySQL.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en su versión 2005 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita. Es muy común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa una potente base de datos (Microsoft SQL Server), con un entorno de desarrollo cómodo y de alto rendimiento (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows. Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor solo está disponible para Sistemas Operativos Windows.

PostgreSQL: es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indizables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS. Otras características interesantes son: las Claves

ajenas también denominadas Llaves ajenas o Llaves Foráneas (foreign keys) y los Disparadores (triggers).

MySQL: es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. En enero de 2008, MySQL AB fue adquirida por Sun Microsystems, y por tanto MySQL. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C. Al contrario que proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Características (versión 4.0):

- ❖ Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.
- ❖ Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:
 - Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
 - Disponibilidad en gran cantidad de plataformas y sistemas.
 - Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
 - Transacciones y claves foráneas.
 - Conectividad segura.
 - Replicación.
 - Búsqueda e indexación de campos de texto.
- ❖ MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando

grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

- ❖ MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.
- ❖ Es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

Características distintivas, las siguientes son implementadas únicamente por MySQL:

- ❖ Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- ❖ Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Servidores Web

Internet Information Services, (IIS): es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web). El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

El servidor HTTP **Apache:** es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo

HTTP/1.1[1] y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "emparchado"). El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. [17]

- ❖ Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.
- ❖ Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft[2]).
- ❖ La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan solo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

Ventajas:

- | | |
|---------------------|--|
| ❖ Modular. | ❖ Extensible. |
| ❖ Open source. | ❖ Popular (fácil conseguir ayuda/soporte). |
| ❖ Multi-plataforma. | ❖ Gratuito. |

Metodologías de Desarrollo de Software

Una Metodología de Desarrollo de Software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. Una metodología puede seguir uno o varios modelos de ciclo de vida, indicando cada ciclo de vida qué es lo que hay que obtener a lo largo del desarrollo del proyecto, pero no cómo hacerlo, en otras palabras, la metodología indica cómo hay que obtener los distintos productos parciales y finales en el desarrollo de un software.

Metodologías Ágiles

Las Metodologías Ágiles son estrategias de desarrollo de software que promueven prácticas que son adaptativas en vez de predictivas, centradas en la gente o en los equipos, iterativas, orientadas hacia prestaciones y hacia la entrega, de comunicación intensiva y que requieren que el negocio se involucre en forma directa. Entre las metodologías ágiles más destacadas hasta el momento se pueden nombrar:

- ❖ XP (eXtreme Programming).
- ❖ Scrum.
- ❖ Evolutionary Project Management (Evo).
- ❖ Crystal Methods.
- ❖ FDD (Feature Driven Development).
- ❖ DSDM (Dynamic Systems Development Method).
- ❖ ASD (Adaptive Software Development).
- ❖ Agile Modeling.

Extreme Programing (XP): Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo y escaso equipo y cuya entrega era ayer. La metodología consiste en una programación rápida o extrema, su particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

La literatura de **Scrum**: se enfoca principalmente en la planeación iterativa y el seguimiento del proceso. Este divide un proyecto en iteraciones de 30 días, llamados carreras cortas. Antes de que comience una carrera se define la funcionalidad requerida para esa carrera y entonces se deja al equipo para que la entregue.

El punto es estabilizar los requisitos durante la carrera. La gerencia no se desentiende durante la carrera corta pues todos los días el equipo sostiene una junta corta (quince minutos), llamada scrum, donde el equipo discute lo que hará al día siguiente. En particular muestran a los bloques de la gerencia los impedimentos que se atraviesan para progresar y que la gerencia debe resolver, también informan lo que se ha hecho para que la gerencia tenga una actualización diaria de dónde va el proyecto. Al final de cada iteración hay una demostración a cargo del Scrum Master. Las presentaciones en PowerPoint están prohibidas. En los encuentros diarios los observadores externos deben estar fuera del círculo y todos tienen que ser puntuales, si alguien llega tarde se le cobra una multa que se destinará a obras de caridad.

Metodologías Orientadas a Objetos

El Método de **Diseño Orientado a Objetos** (Object Oriented Design, OOD) fue definido por Grady Booch a principios de los años 80 y ha conocido varias versiones sucesivas. Desde el punto de

vista de las aplicaciones industriales, es probablemente uno de los métodos precursores de la aproximación orientada a objetos. Se trata de un método de desarrollo (especificación técnica e implementación) y no de diseño (análisis de las necesidades y especificación formal). Lo esencial del método está dedicado a la elaboración del modelo estático (describir los objetos del sistema); el modelo dinámico (cambios de estado de los objetos frente a ciertos eventos) solamente se aborda muy parcialmente y el modelo funcional (describe los procesos de transformación de los usuarios) no se tiene en cuenta.

Object Modeling Technique (OMT): es una de las metodologías (de análisis y diseño) orientadas a objetos, más maduras y eficientes que existen en la actualidad. Fue desarrollada por James Rumbaugh. La gran virtud que aporta esta metodología es su carácter libre (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software. Se hace cargo de todo el ciclo de vida del software y durante ese tiempo mantiene la misma notación, dividiéndolo en las siguientes cuatro fases consecutivas.

Proceso Unificado de Desarrollo (RUP): La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo 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, actualmente se le conoce como Proceso Unificado de Desarrollo. Elementos del RUP:

- ❖ **Trabajadores:** Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- ❖ **Actividades:** Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- ❖ **Artefactos:** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- ❖ **Flujo de actividades:** Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

Ciclo de vida de RUP: RUP cuenta con cuatro fases de desarrollo de software en su ciclo de vida y se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales, los seis primeros son conocidos como flujos de ingeniería o flujos ingenieriles y los tres últimos como de apoyo. Fases de RUP:

- ❖ **Conceptualización** (Concepción o Inicio): Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema. El objetivo en esta etapa es determinar el alcance, la viabilidad y la visión del proyecto.
- ❖ **Elaboración**: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido. El objetivo de esta fase es determinar la línea base de la arquitectura de forma sólida, óptima y estable.
- ❖ **Construcción**: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varias liberaciones (release) del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios. El objetivo de esta fase es obtener la capacidad operativa inicial del producto.
- ❖ **Transición**: El release ya está listo para su instalación en las condiciones reales, esto puede implicar reparación de errores. El objetivo de esta fase es la liberación del producto.

Flujos de trabajo de RUP:

- ❖ **Modelamiento del Negocio**: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- ❖ **Requerimientos**: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ❖ **Análisis y Diseño**: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas, por lo que indica con precisión lo que se debe programar.
- ❖ **Implementación**: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación, creando un software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- ❖ **Prueba (Testeo)**: Busca los defectos a lo largo del ciclo de vida del software, asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- ❖ **Instalación**: Produce release del producto y realiza actividades de empaque, instalación, asistencia a usuarios, etc. para entregar el software a los usuarios finales.
- ❖ **Administración del Proyecto**: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes, administrando horarios y recursos.

- ❖ **Administración de Configuración y Cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a utilización y actualización concurrente de elementos, guardando todas las versiones del proyecto.
- ❖ **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto, así como el procedimiento para implementar el proceso en una organización.

Características del ciclo de vida de RUP:

- ❖ **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- ❖ **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo, en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
- ❖ **Iterativo e Incremental:** Aunque el Anexo 4 puede sugerir que los flujos de trabajo se desarrollan en cascada, la lectura de este gráfico tiene que ser vertical y horizontal. RUP propone que cada fase se desarrolle en iteraciones, una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

En el presente trabajo la metodología que se utilizará es RUP, pues es la que definió la empresa SOFTEL que se debía utilizar, debido a que fue la metodología de desarrollo que los estudiantes del Proyecto APS recibieron en la asignatura Ingeniería de Software I y II. La dirección de la empresa concibió que utilizar otra metodología era poner a los estudiantes a estudiar algo nuevo que les llevaría mucho tiempo aprender y por las potencialidades de RUP, se determinó que no era necesario. RUP es una metodología que usa el Lenguaje Unificado de Modelado (UML) para preparar todos los artefactos de un software, por lo cual se indagará en este trabajo sobre las principales características del mismo.

El **Lenguaje Unificado de Modelado** o UML (Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico

para visualizar, especificar, construir y documentar los elementos que forman un sistema de software orientado a objetos. UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos y empezó como una consolidación del trabajo de Grady Booch, James Rumbaugh e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares (Método de Diseño Orientado a Objetos, OMT y RUP).

UML prescribe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí. Es importante remarcar que UML es un "lenguaje" para especificar y no un método o un proceso. UML ofrece nueve diagramas con los cuales modelar sistemas: [18]

- ❖ Diagramas de Casos de Uso para modelar los procesos 'business'.
- ❖ Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- ❖ Diagramas de Colaboración para modelar interacciones entre objetos.
- ❖ Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- ❖ Diagramas de Actividad para modelar el comportamiento de los casos de uso, objetos u operaciones.
- ❖ Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- ❖ Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- ❖ Diagramas de Componentes para modelar componentes. Diagramas de Implementación para modelar la distribución del sistema.

Herramientas a utilizar en el proceso de desarrollo. Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. Entre sus principales objetivos se encuentran:

- ❖ Mejorar la productividad en el desarrollo y mantenimiento del software.
- ❖ Aumentar la calidad del software.
- ❖ Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- ❖ Mejorar la planificación de un proyecto.

- ❖ Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- ❖ Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- ❖ Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- ❖ Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- ❖ Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Visual Paradigm 2.3: es una herramienta propietaria en su versión libre para Linux. El Visual Paradigm es una herramienta CASE que utiliza “UML”: como lenguaje de modelaje, con el uso del acercamiento orientado al objeto. Esta herramienta apoya los estándares más altos de las notaciones de Java y de UML. Las transiciones del análisis al diseño y la puesta en práctica se integran dentro de la herramienta. Genera productos de calidad, soporta aplicaciones web y es fácil de instalar y actualizar. [19]

Rational es un agrupamiento de metodologías y herramientas que abarca todos los aspectos del desarrollo de software, desde su concepción hasta la elaboración del producto. Para ayudar en la implementación del proceso, Rational ha desarrollado un conjunto de herramientas integradas que permiten desarrollar las actividades del proceso y obtener una sinergia al integrarse entre las distintas herramientas, permitiendo a todo el equipo de desarrollo compartir y utilizar la información necesaria en el momento adecuado. Una de estas herramientas es el Rational Rose.

Rational Rose es la herramienta líder en el mundo para el modelado de sistemas complejos y de tiempo real. Tiene todas las características que los desarrolladores, analistas, y arquitectos están exigiendo, como por ejemplo: un soporte UML incomparable, ingeniería round-trip multi lenguaje, completo soporte al equipo, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria, facilidad de uso, integración optimizada, y mucho más.

Para los analistas de negocios, Rational Rose ofrece la capacidad de modelar y visualizar sus procesos de negocios y destacar oportunidades para aumentar la eficiencia; mientras que para los analistas de datos, el modelado de su diseño de base de datos mejora la comunicación entre el cliente y los desarrolladores y cuando se modela los casos de uso, asegura que la solución sea creada con el usuario en mente.

Esta herramienta es una ayuda invaluable en los esfuerzos de desarrollo porque unifica a todos los equipos a través del modelamiento, el cual está basado en UML; con Rational Rose todo el equipo corporativo puede comunicarse con un lenguaje y una herramienta. Permite visualizar, entender y refinar los requerimientos y arquitectura antes de enfrentarse al código. Esto evita desperdiciar

esfuerzos en el ciclo de desarrollo. Usar una sola herramienta de modelamiento a través del ciclo de vida de desarrollo del software permite asegurar que se está construyendo el sistema correcto.

Rational Rose Enterprise Edition (2003): es el producto más completo de la familia Rational Rose. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado, sin embargo posee características adicionales incluidas: [20]

- ❖ Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
- ❖ Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- ❖ Soporte de ingeniería Forward y reversa para los conceptos más comunes de Java 1.5.
- ❖ La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- ❖ Soporte Enterprise Java Beans 2.0.
- ❖ Capacidad de análisis de calidad de código.
- ❖ El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- ❖ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- ❖ Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
- ❖ Integración con otras herramientas de desarrollo de Rational.
- ❖ Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.
- ❖ Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

Entornos de desarrollo para la edición de código PHP

NuSphere PhpED: es un potente editor de códigos que permite a los desarrolladores prevenir errores y generar códigos más seguros, esta herramienta se caracteriza principalmente por su facilidad de uso y por el soporte de sintaxis que ofrece. Es un programa sencillo y funcional, incluye soporte para un amplio abanico de lenguajes de programación, entre los más destacados se encuentran: PHP, XML, HTML, Perl, CSS, etc. Además es una aplicación robusta, que se integra perfectamente con los sistemas de bases de datos relacionales más utilizados en la actualidad como MySQL. Incluye un asistente que guía paso a paso a los usuarios novatos, minimizando los periodos de adaptación y facilitando las tareas de programación.

Entre sus características más interesantes se pueden relacionar: es un entorno de desarrollo profesional, con depurador, análisis de errores, ayudas para la localización de cuellos de botella en el código, publicación segura de código en servidores e integración con herramientas de terceros

Zend Studio o Zend Development Environment: es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux. Junto con su contraparte Zend Platform, son la propuesta de Zend Technologies para el desarrollo de aplicaciones Web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre. Está disponible también una versión de Zend Studio para Eclipse. Entre sus características de más impacto:

- ❖ No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- ❖ Soporte para PHP 4 y PHP 5.
- ❖ Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- ❖ phpDoc integrado.
- ❖ Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).
- ❖ Inserción automática de paréntesis y corchetes de cierre.
- ❖ Sangrado automático y otras ayudas de formato de código.
- ❖ Emparejamiento (matching) de paréntesis y corchetes (si se sitúa el cursor sobre un paréntesis (corchete) de apertura (cierre), Zend Studio localiza el correspondiente paréntesis (corchete) de cierre (apertura)).
- ❖ Detección de errores de sintaxis en tiempo real.
- ❖ Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP. Permite también la depuración en servidores remotos (requiere Zend Platform).
- ❖ Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox (opcional).
- ❖ Soporte para gestión de grandes proyectos de desarrollo.
- ❖ Manual de PHP integrado.
- ❖ Soporte para control de versiones usando CVS o Subversion (a elección del desarrollador).
- ❖ Cliente FTP integrado.
- ❖ Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Javascript y XML. Zend Studio 5.5 se ha diseñado para una amplia gama de programadores y existen dos ediciones: Standard y Professional. Es concebido con el fin de crear aplicaciones altamente fiables, proporciona una facilidad de uso inigualable, escalabilidad, fiabilidad, y la extensión que los programadores profesionales y de empresas requieren para desarrollar, distribuir, depurar y administrar aplicaciones PHP críticas de negocios, constituye el entorno mas poderoso para PHP, entre las facilidades que brinda se encuentran las siguientes:[21]

- ❖ Integra Java fácilmente en su código utilizando las características del completamiento de Código y define Jars adicionales o carpetas de Clase que pueden utilizarse para el completamiento de códigos.
- ❖ Integración del uso y completamiento de código personalizado de Zend Framework y vista de la lista de funciones del framework para PHP.
- ❖ Aumentar la productividad con: Soporte PHP 5 completo, Analizador de Código, carpeta de Código, completamiento de Código, coloreado de Sintaxis, Administrador de Proyecto, Editor de Código.
- ❖ Documentación del código de forma más sencilla, aplicaciones, y proyectos con PHPDocumentor, la herramienta de documentación standard para PHP.
- ❖ Simplificar el despliegue con la integración FTP y SFTP de forma tal que permita a los programadores en forma segura subir y descargar archivos de proyectos de modo transparente hacia y desde servidores remotos.
- ❖ Conectarse directamente con la bases de datos profesionales más utilizadas tales como IBM-DB2/Cloudscape/Derby/, MySQL, Oracle, Microsoft SQL Server, PostgreSQL y SQLite.
- ❖ Facilitar el desarrollo y colaboración en equipo mediante la administración efectiva de su código fuente utilizando CVS o Subversión directamente desde Zend Studio.
- ❖ Obtener una respuesta inmediata en forma de código o en rendimiento script cuando utiliza Zend Studio 5 Control de Calidad y herramientas de prueba.

Herramienta de Desarrollo Web. Dreamweaver

Adobe Dreamweaver es una aplicación en forma de estudio (Basada por supuesto en la forma de estudio de Adobe Flash) pero con más parecido a un taller destinado para la edición WYSIWYG de páginas web, creado inicialmente por Macromedia (actualmente es propiedad de Adobe Systems). Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su

soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Expression Web. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas.

Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la web, ya que el código que generaba era con frecuencia solo válido para Internet Explorer, y no validaba como HTML estándar. Esto se ha ido corrigiendo en las versiones recientes.

Dreamweaver 8: [22]

- ❖ Proporciona una combinación potente de herramientas visuales de disposición, características de desarrollo de aplicaciones y soporte para la edición de código.
- ❖ Permite que los diseñadores y desarrolladores web creen y manejen cualquier sitio web con toda facilidad.
- ❖ Incluye potentes controles basados en normas para asegurar un diseño de alta calidad. Un entorno de diseño construido en torno a las hojas de estilo en cascada (CSS) hace posible un desarrollo más rápido y más eficiente de sitios profesionales creados con código limpio.
- ❖ Permite el trabajo con diferentes lenguajes para el desarrollo de aplicaciones web, entre ellos: HTML, XHTML, XML, ASP, ASP.NET, JSP, PHP y JavaScript.

Entorno Gráfico para el manejo de Bases de Datos

EMS SQL Manager (2005): es una herramienta que permite un alto rendimiento para Servidores de Base de Datos MySQL para versiones superiores a la 3.23. Permite la realización de Vistas, Procedimientos Almacenados, así como de funciones SQL, llaves foráneas InnoDB, datos Unicode, entre otros. SQL Manager permite crear y editar todo tipo de objetos de las bases de datos MySQL; posee un excelente diseño visual y tiene la posibilidad de ejecutar los script SQL, importar y exportar datos desde y hacia Bases de Datos MySQL. Características claves:

- ❖ Soporta versiones superiores a la 3.23 de los Servidores de Base de Datos MySQL.
- ❖ La rápida navegación y gestión de bases de datos.
- ❖ Simple gestión de todos los objetos de MySQL.
- ❖ Avanzadas herramientas de manipulación de datos.
- ❖ Potente gestión de la seguridad.
- ❖ Excelente texto visual y herramientas para la construcción de consulta.
- ❖ Importante posibilidades de exportación e importación de los datos.
- ❖ Potente Visual Designer para la base de datos.
- ❖ Fácil de uso de asistentes en el desempeño de los servicios de MySQL.

- ❖ La conexión al servidor MySQL a través de HTTP.
- ❖ La conexión al servidor MySQL a través de SSH.

Más novedades: [23]

- ❖ Un Año de Mantenimiento ya incluido.
- ❖ Actualizaciones de software y actualizaciones durante el período de mantenimiento, todo de forma gratis.
- ❖ Libre e ilimitado apoyo técnico durante el período de mantenimiento.

Editor de Hojas de Estilo

Stylus Studio (2007): es un excelente entorno de desarrollo integrado que incluye un potente editor de XML, un "debugger" XSLT y otras muchas herramientas pensadas especialmente para facilitar y mejorar tu 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.

Stylus Studio es el primer y único Ambiente Integrado de Desarrollo que soporta XML en todas sus tecnologías principales: XML, XSL, XSLT, XML Schema, DTD, SOAP, WSDL, SQL/XML y XQuery, y es utilizado por los desarrolladores de aplicaciones en XML para Web a nivel mundial. Stylus Studio 5.1 agrega una nueva y poderosa funcionalidad y facilidad de uso al galardonado Ambiente Integrado de Desarrollo XML que simplifica la programación en XML incrementando la productividad de los desarrolladores a través de la innovación.

Stylus Studio es reconocido ampliamente como la herramienta de desarrollo XSLT más avanzada del mercado al ofrecer funcionalidades como edición, mapeo, creación de perfiles y corrección de bugs en XSLT. Para mantener su tradición de excelencia, Stylus Studio mejoró la velocidad de su "mapeador gráfico" y la creación de instrucciones y valores en XSLT (tales como los valores de xsl, contacto y sum) y las funciones y expresiones condicionales de XSLT (string, contact y sum). Al utilizar Stylus Studio para generar transformaciones de XSLT en conjunto con la capacidad para acceder a datos relacionales a través de SQL/XML, los directivos de TI pueden ahora reducir en forma importante el esfuerzo requerido para agregar y transformar datos para distribución y reporte en Web.

Stylus Studio 5.1 permite la integración de Bases de Datos Relacionales con servicios Web, lo que habilita a las organizaciones para ofrecer una serie de servicios que van desde consultas hasta comercio electrónico a través de Internet.

Una de las características más poderosas de la Versión 5.1 es el Editor de Código Fuente entre Bases de Datos y XML. Al balancear el nuevo estándar SQL/XML, Stylus Studio permite a sus usuarios conectarse en forma directa a cualquier Base de Datos y utilizar la información contenida en éstas como XML. La función de Base de Datos a XML también hace posible que se ejecuten búsquedas complejas XQuery a través de las bases de datos y de la fuente en XML. También soporta la generación de reportes visuales tipo SELECT y UPDATE para Bases de Datos relacionales utilizando el estándar SQL/XML. Las Bases de Datos que se soportan incluyen Oracle, DB2, SQL Server e Informix. [24]

Editor de Web Services Description Language

WSDL son las siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios Web (se lee wisdel). La versión 1.0 fue la primera recomendación por parte del W3C y la versión 1.1 no alcanzó nunca tal estatus. La versión 2.0 se convirtió en la recomendación actual por parte de dicha entidad.

WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

Así, WSDL se usa a menudo en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar que funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

Altova XMLSpy Professional Edition (2007): es el entorno de desarrollo XML líder del mercado, proporcionando intuitivas vistas y potentes utilidades para modelar, editar, transformar y depurar tecnologías relacionadas con XML de forma rápida y fácil. Además del editor XML más popular del mundo XMLSpy 2007 Professional Edition incluye: [25]

- ❖ El original diseñador gráfico de Esquemas XML, permitiéndole diseñar y documentar complejos esquemas fácilmente. Su avanzado manejo de errores y sus capacidades de hiperenlace dinámico en el validador XML hacen de la corrección de documentos XML una tarea sencilla.
- ❖ Un procesador con soporte para XSLT y un depurador XSLT 1.0/2.0 que le permite revisar el proceso de transformación línea por línea. Un práctico analizador XPath 1.0/2.0 le ayuda en la construcción y verificación de expresiones XPath.

- ❖ Un procesador con soporte para XQuery y un robusto depurador XQuery le facilitan la consulta inteligente de datos XML. XMLSpy 2007 Professional Edition soporta las principales bases de datos relacionales: Microsoft Access, SQL Server, Oracle, IBM DB2, MySQL y Sybase, entre otras. Usted podrá conectarse a una base de datos relacional, generar un Esquema XML a partir de la misma, importar y exportar datos basándose en esquemas de bases de datos, y generar bases de datos a partir de Esquemas XML.
- ❖ APIs COM y Java además de controles OLE y ActiveX, que le permitirán acceder programáticamente a las potentes capacidades de XMLSpy. La integración opcional de XMLSpy 2007 Professional Edition con Microsoft Visual Studio.NET o Eclipse le permite acceder directamente a las vistas de edición y herramientas de desarrollo de XML de XMLSpy 2007 desde estos populares IDEs. Altova XMLSpy 2007 Professional Edition está dirigida a desarrolladores profesionales que trabajen con XML, XML Schema, XSLT, XPath, XQuery y bases de datos.

En el capítulo se realizó una profundización en los principales conceptos y definiciones para la comprensión del proceso de desarrollo de la implementación de las Estadísticas Descriptivas del Registro de Enfermedades de Declaración Obligatoria. Este es un sistema de para obtener los Reportes de estas enfermedades y que permita la toma de decisiones por parte del MINSAP y cumplir con la arquitectura definida por dicho organismo. Se realizó, además, un análisis de las tecnologías, patrones, lenguaje y herramientas a tener en cuenta a la hora del desarrollo.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se abordan los temas referentes al objeto de estudio constituyendo el basamento del desarrollo de los Reportes Estadísticos del Registro de Enfermedades de Declaración Obligatoria. De esta forma quedan definidos los principales aspectos para el desarrollo de una propuesta de solución. Para ello se determina la creación de un Modelo de Negocio, debido a la claridad de los procesos de negocio actual y se identifican los actores y trabajadores del mismo.

Para describir el sistema propuesto se enumeran los requisitos, tanto funcionales como no funcionales, identificando a partir de los primeros los Casos de Uso y de esta forma construir el Diagrama de Casos de Uso del Sistema.

2.1 Situación Problémica y Problema a Resolver

El Ministerio de Salud Pública de Cuba, en su Vice-dirección de Higiene y Epidemiología, tiene establecido un control de todas las Enfermedades de Declaración Obligatorias (enfermedades que por su relevancia son consideradas de obligatoria vigilancia epidemiológica por tratados nacionales e internacionales). Por medio de la Vice-dirección de Estadísticas del MINSAP, se lleva el cómputo de los datos referentes a estas enfermedades, permitiéndole al país la posibilidad de la toma de decisiones en correspondencia con la situación determinada en la que se encuentre, desde el punto de vista epidemiológico.

Un médico de familia o de cualquier unidad de salud, puede realizar la notificación de una enfermedad sometida a vigilancia que sea detectada en un lugar determinado, ya sea por medio de una tarjeta de modificación (donde son recogidos los datos generales del paciente, médico que la notificó, lugar donde ocurre, lugar donde reside y de la enfermedad en particular) o por medio de un consolidado que, al igual que la tarjeta, recoge un grupo de datos referentes a un grupo de pacientes que presenten una EDO.

La información estadística de las EDO se recoge y coordina manualmente en los departamentos de estadísticas de las unidades de salud y de las direcciones municipales, solamente se gestiona de forma automatizada en los departamentos de estadísticas de las direcciones provinciales y de la dirección nacional.

Precisamente el proceso obtención de relaciones numéricas mediante tablas y gran cantidad de cálculos, a partir de la información estadística capturada, en los niveles de atención de unidad de salud

y municipal es sumamente engorroso y está propenso a imprecisiones como consecuencia de los errores humanos.

Para el envío de la información estadística desde las unidades de salud a los niveles superiores se emplea considerable tiempo que retrasan la toma de decisiones por parte de la dirección nacional de salud, de ahí el desfasaje entre la notificación de las EDO y la planificación de las medidas de control. En este transcurso además, la información se recoge en diferentes modelos, lo que conlleva en ocasiones a la pérdida o duplicidad de los datos.

En los niveles de atención provincial y nacional se utiliza un software implementado en Foxpro 2.6 para Windows, lenguaje de programación que no se encuentra entre las tendencias tecnológicas actuales.

El sistema se encuentra dividido en dos subsistemas, una versión provincial (EDO Prov) y una versión nacional (EDO Nac). En el subsistema provincial se procesa la información enviada de los municipios correspondientes y se recopilan los modelos de entrada de la información, los cuales se confeccionan en las unidades asistenciales. Se reciben dos versiones del Modelo 241-403, en el que se reporta el total de casos de EDO. Este sistema genera un fichero que se compacta y se envía por correo electrónico al nivel nacional, donde en EDO Nac se recoge la información enviada desde las provincias y se generan tablas de salida similares a las generadas por EDO Prov, en este caso con información referente a las provincias.

El Registro de Enfermedades de Declaración Obligatoria, aplicación desarrollada con herramientas y tecnologías actuales para la gestión de la información relativa a las EDO, no dispone de una funcionalidad que permita visualizar los reportes estadísticos.

Para dar solución a la situación existente el mayor esfuerzo está dirigido en **resolver el problema** siguiente: ¿Cómo obtener la información estadística de las Enfermedades de Declaración Obligatoria en el Sistema Nacional de Salud?

2.2 Modelo de Negocio

El modelamiento del negocio es el primer flujo de trabajo ingenieril que propone RUP. Con este flujo de trabajo se pretende llegar a un mejor entendimiento de la organización donde se implantará el producto. Los principales motivos para esto son los siguientes:

- ❖ Asegurarse de que el producto será algo útil, no un obstáculo.
- ❖ Conseguir que encaje de la mejor forma posible en la organización.
- ❖ Tener un marco común para los desarrolladores, los clientes y los usuarios finales.

Este flujo de trabajo no será siempre necesario. Si solo se añaden funcionalidades que no verán los usuarios directamente, no hará falta. Para modelar el negocio se usarán las mismas técnicas que para modelar software, facilitando que ambas partes entiendan los modelos. En concreto se tendrán casos de uso de negocio, actores de negocio, etc. Los diagramas también tendrán su equivalente de negocio.

Dependiendo del tipo de software que se esté construyendo, el modelado del negocio cambiará. No se trata de modelar la organización de arriba a abajo, solo la parte que corresponde. En la fase de inicio de RUP habrá que hacer una valoración del negocio. Tras hacerla se decidirá cómo hacer el modelado del negocio. Se usará una extensión de UML para modelar el negocio. En concreto habrá modelos de casos de uso de negocio y modelos de objetos de negocio. Con los primeros se captura qué se hace y quién lo hace, con los segundos cómo se hace. Una gran ventaja de ésta aproximación al modelado del negocio es que es una forma clara y concisa de mostrar las dependencias entre el negocio y el sistema que se está construyendo.

Conceptos Fundamentales

Previo a la realización de la descripción general del proceso de negocio actual es pertinente puntualizar varios términos de vital importancia para una mejor comprensión de dicho proceso, a saber:

Estadísticas Descriptivas: es la rama de la Estadística cuyo objetivo es examinar a todos los individuos de un conjunto para describir e interpretar numéricamente la información obtenida.

Enfermedades de Declaración Obligatoria (EDO): son aquellas enfermedades que por su importancia y repercusión se consideran de constante vigilancia epidemiológica por tratados internacionales o nacionales. Ejemplos: SIDA, dengue, tuberculosis, hepatitis, entre otras.

Registro de Enfermedades de Declaración Obligatoria (REDO): es una aplicación web integrada al Sistema de Información para la Salud (SISalud) que permite la gestión de la información relacionada con las Enfermedades de Declaración Obligatoria (EDO).

Descripción General del Proceso de Negocio

El proceso negocio está dirigido a la obtención de las Estadísticas Descriptivas del REDO, este se identifica en las direcciones de unidades de salud, municipales y provinciales y en la dirección nacional. El personal de salud con privilegios para visualizar la información estadística relativas a las EDO (un directivo de la salud, un médico o un técnico estadístico) realizan la solicitud al departamento de estadística correspondiente a su nivel de dirección y a su ubicación geográfica. Con el objetivo de analizar el comportamiento epidemiológico de un grupo poblacional para en caso de alerta

epidemiológica tomar las medidas preventivas o encaminadas a mitigar los daños según las circunstancias en el menor tiempo posible.

El departamento de estadística consulta la serie cronológica de las EDO y realiza una selección de los datos en correspondencia con la solicitud realizada, es decir, el departamento de estadística no debe entregar el paquete de datos completo si éste no fue solicitado sino que deberá hacer un filtrado por criterios tales como lugar, edad, periodo, sexo, fecha, etc. Finalmente al personal de salud solicitante se le hace entrega de la información requerida agrupada convenientemente en tablas que contienen además cálculos importantes como algunos totales, etc.

Reglas del Negocio

Este proceso de negocio actual tiene definido un determinado número de reglas que son de estricto cumplimiento, tanto para el propio proceso como para el sistema propuesto, a continuación se relacionan las Reglas del Negocio identificadas:

- ❖ Los atributos generales definidos para cada reporte serán los mismos pero se tendrá en cuenta según el tipo de salida aquellos que no serán modificables por el actor del negocio.
- ❖ Atributos Generales:
 - Enfermedades
 - Lugar.
 - Periodo.
 - Sexo.
 - Edad.
 - Tipo de Enfermedad.
 - Tipo de Diagnóstico.
 - Tipo de Notificación.
 - Tipo de Fecha.
- ❖ Tipos de Salida:
 - EDO por Lugar.
 - EDO por Grupos de Edades.
 - EDO por Semanas Estadísticas.
 - EDO por Meses.
 - EDO por Semestres.
 - EDO por Lugar y Grupos de Edades.
 - EDO por Lugar y Semanas Estadísticas.
- ❖ El atributo general “Enfermedades” no será modificable nunca pues los reportes son basados en la información de las enfermedades.
- ❖ Obtener Reportes de EDO por Lugar:
 - No se podrá modificar el atributo “Lugar” pues constituye parte de la salida.
- ❖ Obtener Reporte de EDO por Grupos de Edades.
 - No se podrá modificar el atributo “Edad” pues constituye parte de la salida.
- ❖ Obtener Reporte de EDO por Semanas Estadísticas (Meses, Semestres).
 - No se podrá modificar el atributo “Período” pues constituye parte de la salida.
- ❖ Obtener Reporte de EDO por Lugar y Grupos de Edades.

- No se podrán modificar los atributos “Lugar” y “Edad” pues constituyen parte de la salida.
- Solo se podrán relacionar los datos correspondientes a la salida del reporte para una enfermedad, negándose la posibilidad de escoger más de una enfermedad para este tipo de reporte.
- ❖ Obtener Reporte de EDO por Lugar y Semanas Estadísticas.
 - No se podrán modificar los atributos “Lugar” y “Período” pues constituyen parte de la salida.
 - Solo se podrán relacionar los datos correspondientes a la salida del reporte para una enfermedad, negándose la posibilidad de escoger más de una enfermedad para este tipo de reporte.
- ❖ El nivel de dirección del actor del negocio controlará la visibilidad de los datos cuando se desea realizar un reporte que incluya el atributo Lugar de la siguiente manera:
 - Nivel de Dirección Nacional – visualiza información relacionada con:
 - ✓ Provincias
 - ✓ Municipios
 - ✓ Unidades de Salud
 - ✓ GBT
 - ✓ EBS
 - Nivel de Dirección Provincial – visualiza información relacionada con:
 - ✓ Municipios
 - ✓ Unidades de Salud
 - ✓ GBT
 - ✓ EBS
 - Nivel de Dirección Municipal – visualiza información relacionada con:
 - ✓ Unidades de Salud
 - ✓ GBT
 - ✓ EBS
 - Nivel de Dirección de Área de Salud – visualiza información relacionada con:
 - ✓ GBT
 - ✓ EBS
- ❖ Si se selecciona solo una Provincia se podrá escoger de la misma un conjunto de municipios específicos para desglosar la información en la salida del reporte. En caso contrario se tomará en la salida todos los municipios correspondientes a cada una de las provincias seleccionadas.
- ❖ Si se selecciona solo un Municipio se podrá escoger del mismo un conjunto de unidades de salud específicas para desglosar la información en la salida del reporte. En caso contrario se tomará en la salida todas las unidades de salud correspondientes a cada uno de los municipios seleccionados.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

- ❖ Si se selecciona solo una Unidad de Salud se podrá escoger de la misma un conjunto de GBT específicos para desglosar la información en la salida del reporte. En caso contrario se tomará en la salida todos los GBT correspondientes a cada una de las unidades de salud seleccionadas.
- ❖ Si se selecciona solo un GBT se podrá escoger del mismo un conjunto de EBS específicos para desglosar la información en la salida del reporte. En caso contrario se tomará en la salida todos los EBS correspondientes a cada uno de los GBT seleccionados.
- ❖ Si el Lugar es un atributo modificable y no es seleccionado se asume por defecto todo el territorio correspondiente al nivel de dirección y ubicación geográfica del actor del negocio.
- ❖ En caso de seleccionar el Período dentro de un rango de fechas específicos se cumplirá que la fecha inicial nunca superará a la final y la fecha final no podrá ser superior a la actual.
- ❖ El Período podrá ser seleccionado solo para el año en curso si se escogen alguno de los siguientes criterios de selección: Semestres, Meses, Semanas Estadísticas.
- ❖ En caso de seleccionar el Período por semestres se tendrá en cuenta el semestre actual, de modo tal que si se está en el curso del primer semestre será negada la posibilidad de mostrar los datos del segundo semestre.
- ❖ En caso de seleccionar el Período por meses se tendrá en cuenta el mes en actual, de modo tal que no se podrá seleccionar un mes superior al mes en curso.
- ❖ Si el Periodo es un atributo modificable y no es seleccionado se asume por defecto el periodo entre el 1º de enero del presente año hasta la fecha actual.
- ❖ El Sexo podrá ser seleccionado por 4 criterios distintitos: Femenino, Masculino, Ambos (representa la suma total de los sexos femenino y masculino) o Desglosado (representa los sexos femenino y masculino y la suma total de ambos).
- ❖ Si el Sexo no es seleccionado se asume por defecto la suma total de ambos sexos, femenino y masculino.
- ❖ La Edad podrá ser seleccionada por Grupos de Edades o Edad si es un atributo modificable, en caso contrario solo podrá ser seleccionada por Grupos de Edades.
- ❖ Si la Edad es un atributo modificable y no es seleccionado se asume por defecto el rango de edad de 0 a la edad máxima registrada.
- ❖ El Tipo de Enfermedad podrá ser seleccionada por 2 criterios: Tarjeta o Consolidado.
- ❖ Si se selecciona el atributo cualitativo Tarjeta entonces será posible seleccionar el atributo modificable Tipo de Diagnóstico. Además, el atributo modificable Tipo de Notificación podrá ser seleccionado por 4 criterios distintos: Ocurrencia, Residencia, Unidad o Unidad que Informa.

- ❖ Si se selecciona el atributo cualitativo Consolidado entonces no será posible seleccionar el atributo modificable Tipo de Diagnóstico. Además, el atributo modificable Tipo de Notificación podrá ser seleccionado solo por 2 criterios distintos: Unidad o Unidad que Informa.
- ❖ Si el Tipo de Enfermedad no es seleccionado se asume por defecto las enfermedades notificadas por consolidado.
- ❖ El Tipo de Diagnóstico podrá ser seleccionado por 2 criterios distintos: Presuntivo o Confirmado.
- ❖ Si no se selecciona el Tipo de Diagnóstico se asume por defecto el diagnóstico presuntivo.
- ❖ El Tipo de Notificación podrá ser seleccionado por 2 ó 4 criterios distintos según el Tipo de Enfermedad seleccionada.
- ❖ Si no se selecciona el Tipo de Enfermedad (se asume Consolidado por defecto) entonces el Tipo de Notificación podrá ser seleccionado por 2 criterios.
- ❖ Si no se selecciona el Tipo de Notificación se asume por defecto que es notificado desde la unidad correspondiente al actor del negocio.
- ❖ El Tipo de Fecha podrá ser seleccionado por 2 criterios distintos: Notificación o Introducción.
- ❖ Si el Tipo de Fecha no es seleccionado se asume por defecto la fecha de notificación.

Justificación de los Actores y Trabajadores del Negocio

Un actor del negocio es una persona, grupo de personas, organización, u ordenador que interactúa con un negocio determinado y siempre permanece fuera de las fronteras del mismo. El término actor del negocio se le atribuye al rol que alguien o algo desempeña cuando interactúa con el negocio, por tanto no representa un usuario físico ya que varios usuarios físicos pueden realizar el mismo papel en el negocio, además de que un mismo usuario puede encarnar diferentes roles. El nombre de un actor del negocio debe expresar su rol dentro del negocio, además cada actor del negocio debe definirse brevemente con su responsabilidad y por qué interactúa con el negocio. Entre los aspectos más significativos a considerar durante el proceso de definición de los actores del negocio se encuentran:

- ❖ Un actor de negocio expresa un rol no una persona. Una persona puede jugar varios roles.
- ❖ Cada actor del negocio debe estar asociado correctamente con el caso de uso de negocio en el que participa, sino deber ser retirado.
- ❖ El actor del negocio o el proceso de negocio pueden iniciar la comunicación.
- ❖ Cada actor de negocio puede ser sub-clasificado mediante la relación de generalización / especialización.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

- ❖ Existen situaciones en los que existan sistemas de información externos que podrían ser actores del negocio.

ACTOR DEL NEGOCIO	JUSTIFICACION
Visualizador de la Información Estadística	Rol que pueden desempeñar los Directivos de la Salud, los Médicos y los Técnicos Estadísticos, quienes tienen la capacidad de visualizar la Información Estadística relativa a las Enfermedades de Declaración Obligatoria correspondiente a su nivel de dirección y su ubicación geográfica y se benefician de la relación de datos obtenidos a la hora de realizar un análisis del comportamiento epidemiológico del territorio que les corresponde para la posterior toma de decisiones y medidas de vigilancia y control.

Tabla 2.1 Justificación del Actor del Negocio *Visualizador de la Información Estadística*

Un trabajador del negocio es una persona, grupo de personas, organización, sistema (software) o hardware dentro del negocio que realiza las actividades que están comprendidas dentro de un caso de uso. Los trabajadores del negocio siempre permanecen dentro de las fronteras del mismo y posteriormente se convertirán en usuarios del sistema a desarrollar. Cada trabajador del negocio debe definirse brevemente con su responsabilidad dentro del negocio. Los aspectos fundamentales a tener en cuenta durante la definición de los trabajadores del negocio son los que se relacionan a continuación:

- ❖ Un Trabajador del Negocio representa a un ser humano, organización, software o hardware que desempeña un rol dentro de las Realizaciones del Caso de Uso del Negocio.
- ❖ El trabajador interactúa con entidades y otros trabajadores para que el negocio funcione.
- ❖ Los trabajadores de negocio son roles y no posiciones organizacionales, ya que una persona puede desempeñar varios roles pero solo tiene una posición en la organización.

TRABAJADOR DEL NEGOCIO	JUSTIFICACION
Departamento de Estadística	Es la unidad encargada del registro y clasificación de enfermedades en cada centro de salud del país. Tiene la responsabilidad de emitir la Información Estadística de las Enfermedades de Declaración Obligatoria solicitada por un Directivo de la

	Salud, un Médico o un Técnico Estadístico, para ello debe consultar la Serie Cronológica de las EDO y filtrar la información según los criterios solicitados.
--	---

Tabla 2.2 Justificación del Trabajador del Negocio *Departamento de Estadística*

Diagrama de Casos de Uso del Negocio

El diagrama de casos de uso del negocio es un diagrama que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como clientes y socios. Su principal objetivo es describir las funciones que realiza el negocio y cómo el mismo es utilizado por sus clientes y socios. Entre las ventajas más interesantes de realizar un diagrama de casos de uso del negocio se pueden citar:

- ❖ Permite tener una visión general de los diferentes procesos de negocio de la organización, en el cual aparece cada proceso del negocio como un caso de uso.
- ❖ Permite mostrar los límites y el entorno de la organización bajo estudio.
- ❖ Solo muestra los actores del negocio correspondientes a los roles externos al sistema, de forma que los procesos de negocio en los que solo tomen parte roles internos a la organización no estarán conectados a ningún actor.

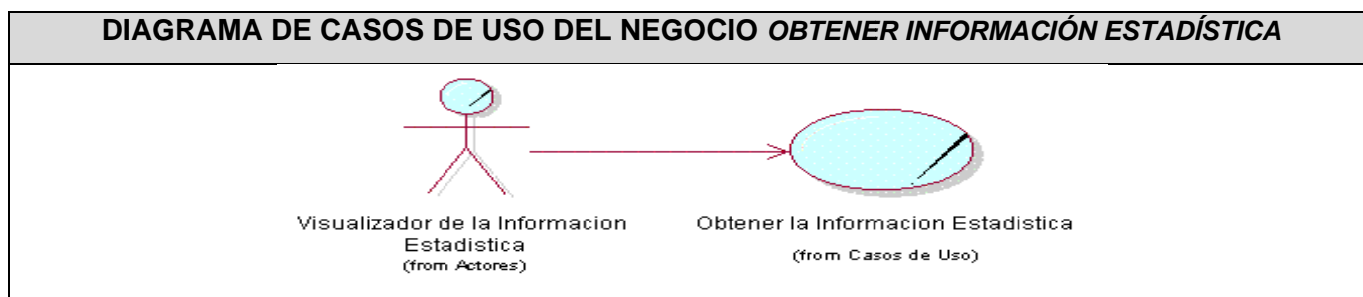


Figura 2.1 Diagrama del Caso de Uso del Negocio *Obtener Información Estadística*

Descripción de los Casos de Uso del Negocio

Un caso de uso del negocio es una secuencia de actividades o acciones dentro de un proceso de negocio que produce un beneficio para un individuo, grupo de individuos, organización u ordenador que interactúa con el negocio.

CASO DE USO	Obtener Información Estadística
ACTOR	Visualizador de la Información Estadística (Directivo de la Salud, Técnico Estadístico, Médico)

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

TRABAJADOR	Departamento de Estadística	
PROPOSITO	Obtener la Información Estadística relativa a las Enfermedades de Declaración Obligatoria para analizar el comportamiento epidemiológico y tomar las medidas de control pertinentes.	
RESUMEN	El caso de uso se inicia cuando un Visualizador de la Información Estadística solicita al Departamento de Estadística correspondiente a su nivel de dirección y ubicación geográfica la Información Estadística relativa a las Enfermedades de Declaración Obligatoria. El Departamento de Estadística consulta la Serie Cronológica de las EDO, filtra la información solicitada teniendo en cuenta los criterios de búsqueda precisados en la petición y la retorna en forma de Reporte Estadístico con subtotales y total calculados.	
FLUJO NORMAL DE LOS EVENTOS		
	ACCION DEL ACTOR	RESPUESTA DEL NEGOCIO
	1. Un Visualizador de la Información Estadística solicita al Departamento de Estadística correspondiente a su nivel de dirección y ubicación geográfica la Información Estadística relativa a las Enfermedades de Declaración Obligatoria.	1.1 El Departamento de Estadística consulta la Serie Cronológica de las EDO. 1.2 El Departamento de Estadística filtra la información solicitada teniendo en cuenta los criterios de búsqueda precisados en la petición. 1.3 El Departamento de Estadística realiza los cálculos de los subtotales horizontales y verticales y del total. 1.4 El Departamento de Estadística retorna la Información Estadística solicitada en forma de Reporte Estadístico.
FLUJO ALTERNATIVO DE LOS EVENTOS		
	ACCION DEL ACTOR	RESPUESTA DEL NEGOCIO
		-
PRE-CONDICIONES	La existencia de la Información Estadística relativa a las EDO actualizada en la Serie Cronológica de las EDO en los Departamentos de Estadística.	
POS-CONDICIONES	-	
PRIORIDAD	Crítico	
MEJORAS	Realizar de forma automática el filtrado de la Información Estadística según los	

	criterios contenidos en la solicitud realizada por el Visualizador de la Información Estadística.
--	---

Tabla 2.3 Descripción del Caso de Uso del Negocio *Obtener Información Estadística*

Diagrama de Actividades del Caso de Uso del Negocio

Un diagrama de actividades representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. En este artefacto las actividades representan las acciones que se realizan en el proceso de negocio y las transiciones el orden lógico en que se suceden estas acciones.

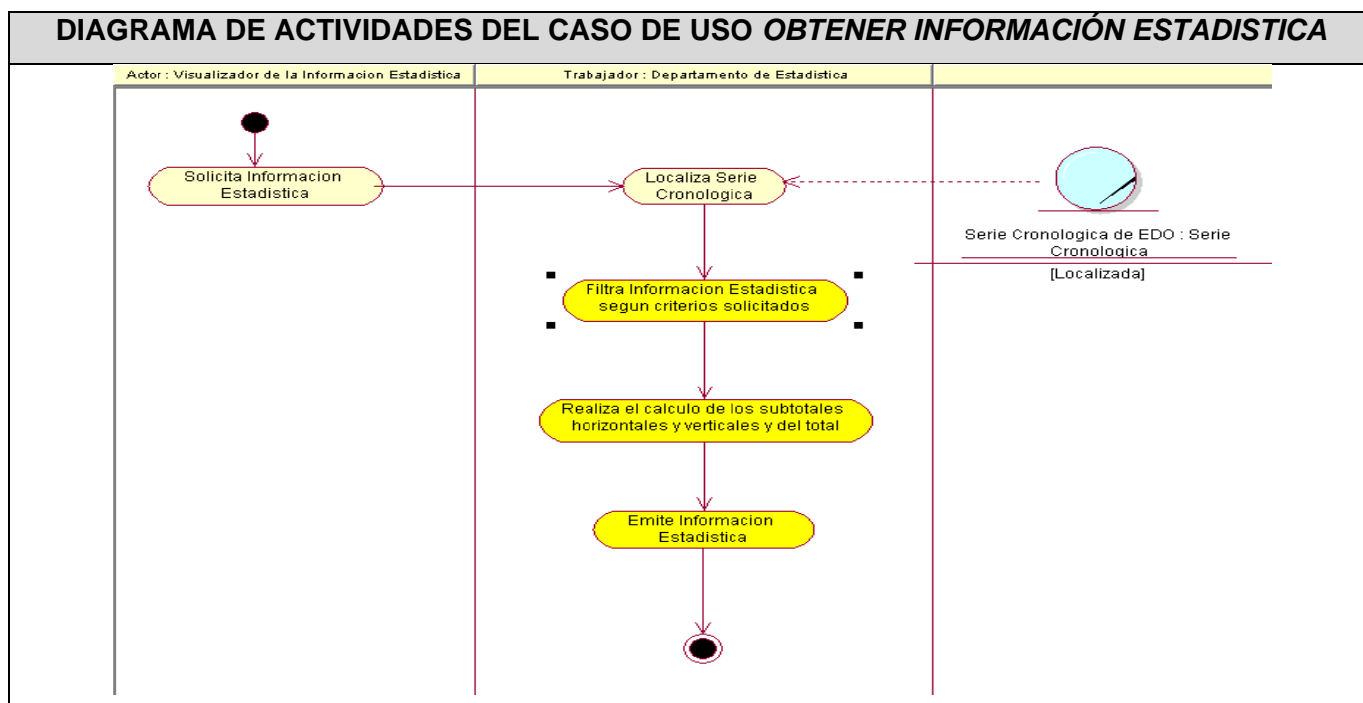


Figura 2.2 Diagrama de Actividades del Caso de Uso *Obtener Información Estadística*

Diagrama de Clases del Modelo de Objetos

Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. Se emplean para modelar la vista de diseño estática o la vista de procesos estática de un sistema al igual que se hace con los diagramas de clases, pero desde la perspectiva de instancias reales o prototípicas. Los diagramas de objetos se utilizan para visualizar, especificar, construir y documentar la existencia de ciertas instancias en el sistema, junto a las relaciones entre ellas.

Se puede considerar un caso especial de un diagrama de clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de

objetos utilizan un subconjunto de los elementos de un diagrama de clase. Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase.

Un modelo de objetos del negocio describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo. Las entidades de negocio representan a los objetos que los trabajadores del negocio manipulan o producen durante la realización de los casos de uso del negocio.

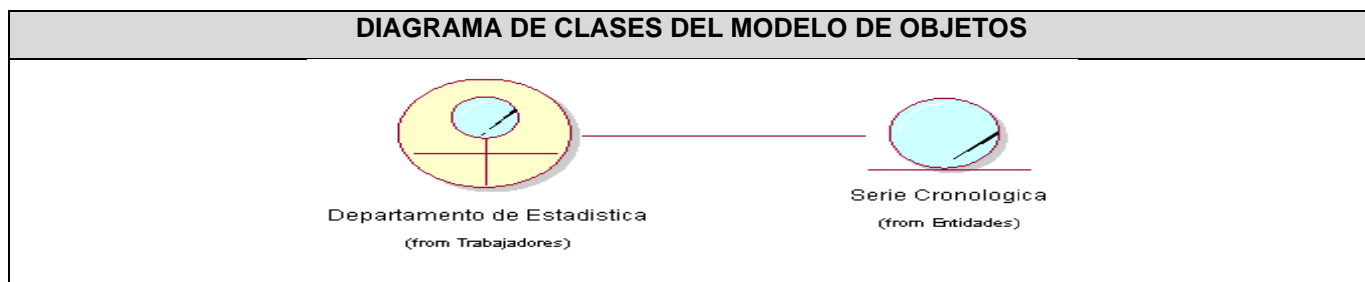


Figura 2.3 Diagrama de Clases del Modelo de Objetos

2.3 Propuesta del Sistema

Especificación de los Requerimientos del Software

Los requerimientos o requisitos son condiciones o capacidades que tienen que ser alcanzadas o por una aplicación o sistema de aplicaciones para satisfacer un contrato legal entre una persona o entidad y un equipo de desarrollo de software. Las necesidades de los clientes y usuarios finales así como las sugerencias de los miembros del equipo de proyecto acerca de qué debe hacer el sistema y cómo lo debe hacer deben ser consideradas como candidatas a requisitos. Los requisitos se pueden clasificar en funcionales y no funcionales.

Requerimientos Funcionales

En la realización de los casos de uso del negocio se obtienen las actividades que serán objeto de automatización, estas actividades no son exactamente los requerimientos funcionales pero si son el punto de partida para identificar qué debe hacer el sistema. Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, un requerimiento funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica.

Relación de los Requisitos Funcionales (RF) del Producto:

- ❖ **RF1:** Obtener Reporte EDO por Lugar.

- **RF1.1:** Obtener Reporte EDO Tarjeta por Lugar.
 - RF1.1.1:** Obtener Reporte EDO Tarjeta por Provincias.
 - RF1.1.1.1:** Imprimir Reporte EDO Tarjeta por Provincias.
 - RF1.1.2:** Obtener Reporte EDO Tarjeta por Municipios.
 - RF1.1.2.1:** Imprimir Reporte EDO Tarjeta por Municipios.
 - RF1.1.3:** Obtener Reporte EDO Tarjeta por Unidades de Salud.
 - RF1.1.3.1:** Imprimir Reporte EDO Tarjeta por Unidades de Salud.
 - RF1.1.4:** Obtener Reporte EDO Tarjeta por GBT y EBS.
 - RF1.1.4.1:** Imprimir Reporte EDO Tarjeta por GBT y EBS.
- **RF1.2:** Obtener Reporte EDO Consolidado por Lugar.
 - RF1.2.1:** Obtener Reporte EDO Consolidado por Provincias.
 - RF1.2.1.1:** Imprimir Reporte EDO Consolidado por Provincias.
 - RF1.2.2:** Obtener Reporte EDO Consolidado por Municipios.
 - RF1.2.2.1:** Imprimir Reporte EDO Consolidado por Municipios.
 - RF1.2.3:** Obtener Reporte EDO Consolidado por Unidades de Salud.
 - RF1.2.3.1:** Imprimir Reporte EDO Consolidado por Unidades de Salud.
 - RF1.2.4:** Obtener Reporte EDO Consolidado por GBT y EBS.
 - RF1.2.4.1:** Imprimir Reporte EDO Consolidado por GBT y EBS.
- ❖ **RF2:** Obtener Reporte EDO por Grupos de Edades.
 - **RF2.1:** Obtener Reporte EDO Tarjeta por Grupos de Edades.
 - RF2.1.1:** Imprimir Reporte EDO Tarjeta por Grupos de Edades.
 - **RF2.2:** Obtener Reporte EDO Consolidado por Grupos de Edades.
 - RF2.2.1:** Imprimir Reporte EDO Consolidado por Grupos de Edades.
- ❖ **RF3:** Obtener Reporte EDO por Semanas Estadísticas.
 - **RF3.1:** Obtener Reporte EDO Tarjeta por Semanas Estadísticas.
 - RF3.1.1:** Imprimir Reporte EDO Tarjeta por Semanas Estadísticas.
 - **RF3.2:** Obtener Reporte EDO Consolidado por Semanas Estadísticas.
 - RF3.2.1:** Imprimir Reporte EDO Consolidado por Semanas Estadísticas.
- ❖ **RF4:** Obtener Reporte EDO por Meses.
 - **RF4.1:** Obtener Reporte EDO Tarjeta por Meses.
 - RF4.1.1:** Imprimir Reporte EDO Tarjeta por Meses.
 - **RF4.2:** Obtener Reporte EDO Consolidado por Meses.
 - RF4.2.1:** Imprimir Reporte EDO Consolidado por Meses.
- ❖ **RF5:** Obtener Reporte EDO por Semestres.

- **RF5.1:** Obtener Reporte EDO Tarjeta por Semestres.
 - RF5.1.1:** Imprimir Reporte EDO Tarjeta por Semestres.
- **RF5.2:** Obtener Reporte EDO Consolidado por Semestres.
 - RF5.2.1:** Imprimir Reporte EDO Consolidado por Semestres.
- ❖ **RF6:** Obtener Reporte EDO por Lugar y Grupos de Edades.
 - **RF6.1:** Obtener Reporte EDO Tarjeta por Lugar y Grupos de Edades.
 - RF6.1.1:** Obtener Reporte EDO Tarjeta por Provincias y Grupos de Edades.
 - RF6.1.1.1:** Imprimir Reporte EDO Tarjeta por Provincias y Grupos de Edades.
 - RF6.1.2:** Obtener Reporte EDO Tarjeta por Municipios y Grupos de Edades.
 - RF6.1.2.1:** Imprimir Reporte EDO Tarjeta por Municipios y Grupos de Edades.
 - RF6.1.3:** Obtener Reporte EDO Tarjeta por Unidades de Salud y Grupos de Edades.
 - RF6.1.3.1:** Imprimir Reporte EDO Tarjeta por Unidades de Salud y Grupos de Edades.
 - RF6.1.4:** Obtener Reporte EDO Tarjeta por GBT y EBS y Grupos de Edades.
 - RF6.1.4.1:** Imprimir Reporte EDO Tarjeta por GBT y EBS y Grupos de Edades.
 - **RF6.2:** Obtener Reporte EDO Consolidado por Lugar y Grupos de Edades.
 - RF6.2.1:** Obtener Reporte EDO Consolidado por Provincias y Grupos de Edades.
 - RF6.2.1.1:** Imprimir Reporte EDO Consolidado por Provincias y Grupos de Edades.
 - RF6.2.2:** Obtener Reporte EDO Consolidado por Municipios y Grupos de Edades.
 - RF6.2.2.1:** Imprimir Reporte EDO Consolidado por Municipios y Grupos de Edades.
 - RF6.2.3:** Obtener Reporte EDO Consolidado por Unidades de Salud y Grupos de Edades.
 - RF6.2.3.1:** Imprimir Reporte EDO Consolidado por Unidades de Salud y Grupos de Edades.
 - RF6.2.4:** Obtener Reporte EDO Consolidado por GBT y EBS y Grupos de Edades.
 - RF6.2.4.1:** Imprimir Reporte EDO Consolidado por GBT y EBS y Grupos de Edades.
- ❖ **RF7:** Obtener Reporte EDO por Lugar y Semanas Estadísticas.
 - **RF7.1:** Obtener Reporte EDO Tarjeta por Lugar y Semanas Estadísticas.
 - RF7.1.1:** Obtener Reporte EDO Tarjeta por Provincias y Semanas Estadísticas.
 - RF7.1.1.1:** Imprimir Reporte EDO Tarjeta por Provincias y Semanas Estadísticas.
 - RF7.1.2:** Obtener Reporte EDO Tarjeta por Municipios y Semanas Estadísticas.
 - RF7.1.2.1:** Imprimir Reporte EDO Tarjeta por Municipios y Semanas Estadísticas.
 - RF7.1.3:** Obtener Reporte EDO Tarjeta por Unidades de Salud y Semanas Estadísticas.
 - RF7.1.3.1:** Imprimir Reporte EDO Tarjeta por Unidades de Salud y Semanas Estadísticas.
 - RF7.1.4:** Obtener Reporte EDO Tarjeta por GBT y EBS y Semanas Estadísticas.

- **RF9.2:** Obtener Reporte EDO Consolidado por Grupos de Edades y Sexo.
RF9.2.1: Imprimir Reporte EDO Consolidado por Grupos de Edades y Sexo.
- ❖ **RF10:** Obtener Reporte EDO por Semanas Estadísticas y Sexo.
 - **RF10.1:** Obtener Reporte EDO Tarjeta por Semanas Estadísticas y Sexo.
RF10.1.1: Imprimir Reporte EDO Tarjeta por Semanas Estadísticas y Sexo.
 - **RF10.2:** Obtener Reporte EDO Consolidado por Semanas Estadísticas y Sexo.
RF10.2.1: Imprimir Reporte EDO Consolidado por Semanas Estadísticas y Sexo.
- ❖ **RF11:** Obtener Reporte EDO por Meses y Sexo.
 - **RF11.1:** Obtener Reporte EDO Tarjeta por Meses y Sexo.
RF11.1.1: Imprimir Reporte EDO Tarjeta por Meses y Sexo.
 - **RF11.2:** Obtener Reporte EDO Consolidado por Meses y Sexo.
RF11.2.1: Imprimir Reporte EDO Consolidado por Meses y Sexo.
- ❖ **RF12:** Obtener Reporte EDO por Semestres y Sexo.
 - **RF12.1:** Obtener Reporte EDO Tarjeta por Semestres y Sexo.
RF12.1.1: Imprimir Reporte EDO Tarjeta por Semestres y Sexo.
 - **RF12.2:** Obtener Reporte EDO Consolidado por Semestres y Sexo.
RF12.2.1: Imprimir Reporte EDO Consolidado por Semestres y Sexo.
- ❖ **RF13:** Obtener Reporte EDO por Lugar y Grupos de Edades y Sexo.
 - **RF13.1:** Obtener Reporte EDO Tarjeta por Lugar y Grupos de Edades y Sexo.
RF13.1.1: Obtener Reporte EDO Tarjeta por Provincias y Grupos de Edades y Sexo.
RF13.1.1.1: Imprimir Reporte EDO Tarjeta por Provincias y Grupos de Edades y Sexo.
RF13.1.2: Obtener Reporte EDO Tarjeta por Municipios y Grupos de Edades y Sexo.
RF13.1.2.1: Imprimir Reporte EDO Tarjeta por Municipios y Grupos de Edades y Sexo.
RF13.1.3: Obtener Reporte EDO Tarjeta por Unidades de Salud y Grupos de Edades y Sexo.
RF13.1.3.1: Imprimir Reporte EDO Tarjeta por Unidades de Salud y Grupos de Edades y Sexo.
 - **RF13.1.4:** Obtener Reporte EDO Tarjeta por GBT y EBS y Grupos de Edades y Sexo.
RF13.1.4.1: Imprimir Reporte EDO Tarjeta por GBT y EBS y Grupos de Edades y Sexo.
 - **RF13.2:** Obtener Reporte EDO Consolidado por Lugar y Grupos de Edades y Sexo.
RF13.2.1: Obtener Reporte EDO Consolidado por Provincias y Grupos de Edades y Sexo.
RF13.2.1.1: Imprimir Reporte EDO Consolidado por Provincias y Grupos de Edades y Sexo.
 - **RF13.2.2:** Obtener Reporte EDO Consolidado por Municipios y Grupos de Edades y Sexo.

RF13.2.2.1: Imprimir Reporte EDO Consolidado por Municipios y Grupos de Edades y Sexo.

RF13.2.3: Obtener Reporte EDO Consolidado por Unidades de Salud y Grupos de Edades y Sexo.

RF13.2.3.1: Imprimir Reporte EDO Consolidado por Unidades de Salud y Grupos de Edades y Sexo.

RF13.2.4: Obtener Reporte EDO Consolidado por GBT y EBS y Grupos de Edades y Sexo.

RF13.2.4.1: Imprimir Reporte EDO Consolidado por GBT y EBS y Grupos de Edades y Sexo.

❖ **RF14:** Obtener Reporte EDO por Lugar y Semanas Estadísticas y Sexo.

▪ **RF14.1:** Obtener Reporte EDO Tarjeta por Lugar y Semanas Estadísticas y Sexo.

RF14.1.1: Obtener Reporte EDO Tarjeta por Provincias y Semanas Estadísticas y Sexo.

RF14.1.1.1: Imprimir Reporte EDO Tarjeta por Provincias y Semanas Estadísticas y Sexo.

RF14.1.2: Obtener Reporte EDO Tarjeta por Municipios y Semanas Estadísticas y Sexo.

RF14.1.2.1: Imprimir Reporte EDO Tarjeta por Municipios y Semanas Estadísticas y Sexo.

RF14.1.3: Obtener Reporte EDO Tarjeta por Unidades de Salud y Semanas Estadísticas y Sexo.

RF14.1.3.1: Imprimir Reporte EDO Tarjeta por Unidades de Salud y Semanas Estadísticas y Sexo.

RF14.1.4: Obtener Reporte EDO Tarjeta por GBT y EBS y Semanas Estadísticas y Sexo.

RF14.1.4.1: Imprimir Reporte EDO Tarjeta por GBT y EBS y Semanas Estadísticas y Sexo.

▪ **RF14.2:** Obtener Reporte EDO Consolidado por Lugar y Semanas Estadísticas y Sexo.

RF14.2.1: Obtener Reporte EDO Consolidado por Provincias y Semanas Estadísticas y Sexo.

RF14.2.1.1: Imprimir Reporte EDO Consolidado por Provincias y Semanas Estadísticas y Sexo.

RF14.2.2: Obtener Reporte EDO Consolidado por Municipios y Semanas Estadísticas y Sexo.

RF14.2.2.1: Imprimir Reporte EDO Consolidado por Municipios y Semanas Estadísticas y Sexo.

RF14.2.3: Obtener Reporte EDO Consolidado por Unidades de Salud y Semanas Estadísticas y Sexo.

RF14.2.3.1: Imprimir Reporte EDO Consolidado por Unidades de Salud y Semanas Estadísticas y Sexo.

RF14.2.4: Obtener Reporte EDO Consolidado por GBT y EBS y Semanas Estadísticas y Sexo.

RF14.2.4.1: Imprimir Reporte EDO Consolidado por GBT y EBS y Semanas Estadísticas y Sexo.

Requerimientos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, un requerimiento no funcional es un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Si un software cumple con las funcionalidades requeridas, las propiedades no funcionales, (como cuán usable, seguro, conveniente y agradable es), pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Los requerimientos funcionales se pueden clasificar en múltiples categorías:

Usabilidad: Estos requerimientos describen los niveles apropiados de usabilidad dados los usuarios finales del producto, para ello debe revisarse las especificaciones de los perfiles de usuarios y definir las clasificaciones de sus niveles de experiencia.

RNF 1: Cada usuario que se autentique solo podrá visualizar la información correspondiente a su nivel de dirección (nacional, provincial, municipal, de unidad de salud) y su ubicación geográfica.

Rendimiento: Especifican los requisitos relacionados con la carga que se espera que tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, el número de transacciones por segundo que deberá soportar el sistema, etc. Todos estos requisitos deben ser medibles y debe indicarse el porcentaje (%) relacionado con su tiempo.

RNF 2: El sistema debe permitir la visualización de los Reportes Estadísticos de las EDO en un tiempo de respuesta menor o igual a 30 segundos.

RNF 3: Para obtener la Información Estadística de las EDO se deben enviar respuestas SOAP que no deben exceder los 50 Kbyte en el tiempo de respuesta al usuario.

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

RNF 4: El personal que interactúa con el módulo debe contar con el nivel técnico requerido mediante capacitación de servicio para manipular la nueva funcionalidad adquirida por REDO.

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

RNF 5: La seguridad no implicará lentitud o retraso en la obtención de los Reportes Estadísticos, por lo que se debe minimizar y reducir el tiempo de respuesta, así como optimizar el código.

Apariencia o Interfaz Externa: este tipo de requerimiento describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle, sino que especifican cómo se pretende que sea la interfaz externa del producto.

RNF 6: Todos los Reportes Estadísticos se solicitarán desde una página principal y la información correspondiente a cada uno se mostrará en páginas independientes.

Apariencia o Interfaz Interna: este tipo de requerimiento describe la apariencia del producto internamente, como deben ser sus componentes, la relación entre ellos, así como el funcionamiento interno del software.

RNF 7: Todos los componentes del sistema deben desarrollarse siguiendo el principio de máxima cohesión y mínimo acoplamiento.

RNF 8: Los componentes reutilizables entre los módulos que componen el Sistema de Información para la Salud (SISalud) deberán ser desarrollados como servicios web XML que interactúan a través de SOAP con otros componentes.

Ayuda y Documentación en Línea: Describen los requisitos que provee a los usuarios y clientes documentación para la utilización y manipulación del sistema, así como para entender el mismo.

RNF 9: Disponer de instrucciones acerca del modo correcto de interacción con la nueva funcionalidad de REDO en una opción de ayuda.

Hardware: Estos requisitos especifican las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

RNF 19: Disponer de una impresora local o de red para imprimir los reportes solicitados.

Restricciones en el diseño y la implementación: Este tipo de requerimiento especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente. Ejemplos de ellas son: estándares requeridos, lenguajes de programación a ser usados en la implementación, uso obligatorio de ciertas herramientas de desarrollo, restricciones en la arquitectura o el diseño, bibliotecas de clases, etc.

RNF 20: La lógica de presentación constituirá una capa independiente de la lógica de negocio, centrando su función en la interfaz de usuario y validaciones simples de los datos de entrada.

RNF 21: Validar el proceso de la captación de datos para evitar entradas inadecuadas.

2.4 Modelo de Casos de Uso del Sistema

El Modelo de Casos de Uso del Sistema se basa en la descripción de elementos o usuarios externos al sistema (actores del sistema) y de las funcionalidades del sistema (casos de uso del sistema). Un Modelo de Casos de Uso describe los requerimientos funcionales de un actor (usuario, sistema externo, etc.) en términos de las interacciones que el actor ejecuta con el 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 como 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.

Definición de los Actores del Sistema

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

ACTOR DEL SISTEMA	JUSTIFICACION
Departamento	Es el encargado de emitir los Reportes Estadísticos de las EDO para el análisis del

de Estadística	comportamiento epidemiológico y la posterior toma de decisiones y medidas de control.
----------------	---

Tabla 2.4 Justificación del Actor del Sistema *Departamento de Estadística*

Diagrama de Casos de Uso del Sistema

Un Diagrama de Casos de Uso es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Todo sistema tiene como mínimo un Diagrama de Casos de Uso, que es una representación gráfica del entorno del sistema (actores) y sus funcionalidades principales, por tanto, muestra, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

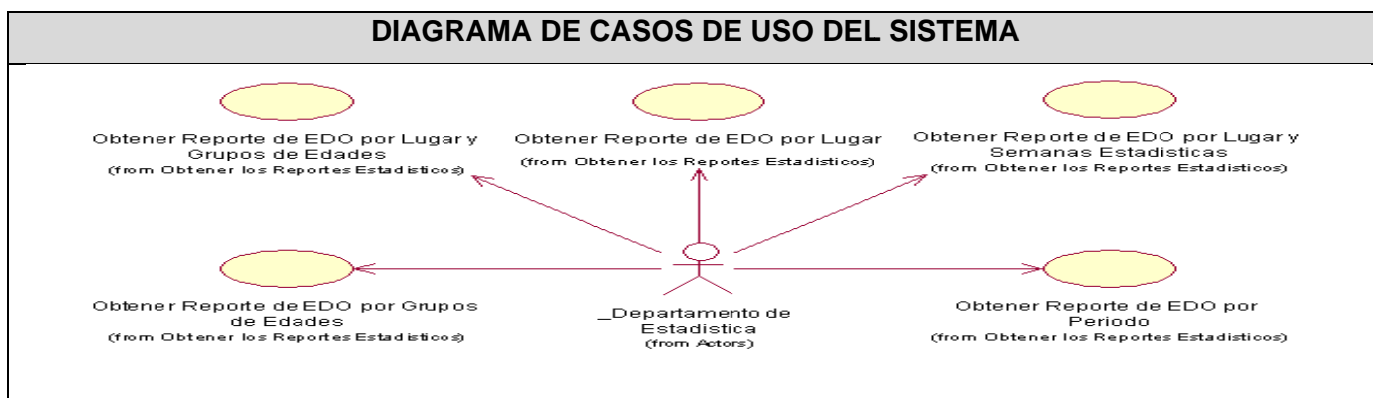


Figura 2.4 Diagrama de Casos de Uso del Sistema

Descripción Textual de los Casos de Uso del Sistema

CASO DE USO	CUS1: Obtener Reporte de EDO por Lugar
ACTORES	Departamento de Estadística
PROPOSITO	Obtener la Información Estadística de las EDO relativa solo a individuos de uno o varios lugares específicos o en todo el territorio nacional para analizar el comportamiento epidemiológico en determinadas zonas geográficas.
RESUMEN	El caso de uso se inicia cuando el Departamento de Estadística selecciona la opción Reportes Totales / EDO por Lugar. El sistema muestra la página principal para filtrar la información según criterios restrictivos. El Departamento de Estadística puede seleccionar cero, uno, varios o todos los criterios de filtrado y solicitar la información deseada. El sistema devuelve los datos organizados en una tabla resultado con subtotales y totales calculados finalizando así el caso de uso.
PRE-CONDICIONES	Debe existir conectividad entre el sistema y la base de datos. El Departamento de Estadística debe estar registrado en la base de datos como usuario del sistema. La

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	información estadística de las EDO debe estar almacenada y actualizada en la base de datos.
POS-CONDICIONES	-
REQUISITOS	RF1, RF8, CUS2, CUS3

Tabla 2.5 Descripción del Caso de Uso *Obtener Reporte de EDO por Lugar*

CASO DE USO	CUS2: Obtener Reporte de EDO por Grupos de Edades
ACTORES	Departamento de Estadística
PROPOSITO	Obtener la Información Estadística de las EDO relativa solo a individuos de uno o varios grupos de edades específicos o de todos los grupos de edades definidos para analizar el comportamiento epidemiológico en determinadas etapas de desarrollo.
RESUMEN	El caso de uso se inicia cuando el Departamento de Estadística selecciona la opción Reportes Totales / EDO por Grupos de Edades. El sistema muestra la página principal para filtrar la información según criterios restrictivos. El Departamento de Estadística puede seleccionar cero, uno, varios o todos los criterios de filtrado y solicitar la información deseada. El sistema devuelve los datos organizados en una tabla resultado con subtotales y totales calculados finalizando así el caso de uso.
PRE-CONDICIONES	Debe existir conectividad entre el sistema y la base de datos. El Departamento de Estadística debe estar registrado en la base de datos como usuario del sistema. La información estadística de las EDO debe estar almacenada y actualizada en la base de datos.
POS-CONDICIONES	-
REQUISITOS	RF2, RF9, CU1, CU3

Tabla 2.6 Descripción del Caso de Uso *Obtener Reporte de EDO por Grupos de Edades*

CASO DE USO	CU3: Obtener Reporte de EDO por Periodo
ACTORES	Departamento de Estadística
PROPOSITO	Obtener la Información Estadística de las EDO relativa a todos individuos en un margen de tiempo específico o en todo el tiempo transcurrido durante el año hasta

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	la fecha actual para analizar el comportamiento epidemiológico en determinados momentos del año en curso.
RESUMEN	El caso de uso se inicia cuando el Departamento de Estadística selecciona la opción Reportes Totales / (EDO por Semanas Estadísticas, EDO por Meses o EDO por Semestres). El sistema muestra la página principal para filtrar la información según criterios restrictivos. El Departamento de Estadística puede seleccionar cero, uno, varios o todos los criterios de filtrado y solicitar la información deseada. El sistema devuelve los datos organizados en una tabla resultado con subtotales y totales calculados finalizando así el caso de uso.
PRE-CONDICIONES	Debe existir conectividad entre el sistema y la base de datos. El Departamento de Estadística debe estar registrado en la base de datos como usuario del sistema. La información estadística de las EDO debe estar almacenada y actualizada en la base de datos.
POS-CONDICIONES	-
REQUISITOS	RF3, RF4, RF5, RF10, RF11, RF12, CU1, CU2

Tabla 2.7 Descripción del Caso de Uso *Obtener Reporte de EDO por Periodo*

CASO DE USO	CU4: Obtener Reporte de EDO por Lugar y Grupos de Edades
ACTORES	Departamento de Estadística
PROPOSITO	Obtener la Información Estadística de una EDO combinando el CU1 y el CU2.
RESUMEN	El caso de uso se inicia cuando el Departamento de Estadística selecciona la opción Reportes Totales / EDO por Lugar y Grupos de Edades. El sistema muestra la página principal para filtrar la información según criterios restrictivos. El Departamento de Estadística puede seleccionar cero, uno, varios o todos los criterios de filtrado y solicitar la información deseada. El sistema devuelve los datos organizados en una tabla resultado con subtotales y totales calculados finalizando así el caso de uso.
PRE-CONDICIONES	Debe existir conectividad entre el sistema y la base de datos. El Departamento de Estadística debe estar registrado en la base de datos como usuario del sistema. La información estadística de las EDO debe estar almacenada y actualizada en la base de datos.
POS-	-

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

CONDICIONES	
REQUISITOS	RF6, RF13, CU1, CU2, CU3

Tabla 2.8 Descripción del Caso de Uso *Obtener Reporte de EDO por Lugar y Grupos de Edades*

CASO DE USO	CU5: Obtener Reporte de EDO por Lugar y Semanas Estadísticas
ACTORES	Departamento de Estadística
PROPOSITO	Obtener la Información Estadística de una EDO combinando el CU1 y el CU3.
RESUMEN	El caso de uso se inicia cuando el Departamento de Estadística selecciona la opción Reportes Totales / EDO por Lugar y Semanas Estadísticas. El sistema muestra la página principal para filtrar la información según criterios restrictivos. El Departamento de Estadística puede seleccionar cero, uno, varios o todos los criterios de filtrado y solicitar la información deseada. El sistema devuelve los datos organizados en una tabla resultado con subtotales y totales calculados finalizando así el caso de uso.
PRE-CONDICIONES	Debe existir conectividad entre el sistema y la base de datos. El Departamento de Estadística debe estar registrado en la base de datos como usuario del sistema. La información estadística de las EDO debe estar almacenada y actualizada en la base de datos.
POS-CONDICIONES	-
REQUISITOS	RF7, RF14, CU1, CU2, CU3

Tabla 2.9 Descripción del Caso de Uso *Obtener Reporte de EDO por Lugar y Semanas Estadísticas*

En este capítulo se describieron las características del sistema propuesto. Para ello se realizó el Modelo del Negocio, donde se explicó de forma detallada el proceso de obtención de la Información Estadística de las EDO mediante la generación de los artefactos definidos en este flujo de trabajo. Además, se realizó el levantamiento de requisitos, tanto funcionales como no funcionales, lo cual permitió adentrarse en el modelo de sistema, definiendo los actores y casos de uso del mismo, así como sus relaciones, conformándose así el Diagrama de Casos de Uso.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo se expone el análisis del sistema propuesto mediante la realización de diferentes artefactos incluidos en el modelo de análisis, ya sean los diagramas de clases como la definición de las mismas.

En el diseño se modela el sistema para que soporte todos los requisitos funcionales y no funcionales, contribuyendo a obtener una arquitectura sólida y estable para la futura implementación del software. Entre los artefactos que incluye el modelo de diseño a mostrar se encuentran la definición de los elementos que este posee, diagramas de clases y descripción de las mismas.

3.1 Modelo de Análisis

El modelo de análisis permite refinar y a estructurar los requisitos y proporciona una estructura centrada en el mantenimiento en aspectos como la flexibilidad ante los cambios y la reutilización de componentes, útil además para el comienzo de las actividades de diseño e implementación. Un modelo de análisis se describe utilizando el lenguaje de los desarrolladores y puede, por tanto, introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.

Aún cuando en el modelo del análisis hay un refinamiento de los requisitos, no se toman en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, los componentes reutilizables de otras aplicaciones, entre otras características que afectan al sistema, puesto que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará el producto.

Diagramas de Clases del Análisis

El diagrama de clases del análisis se realiza para cada caso de uso del sistema y muestra las clases participantes en dicho caso de uso y sus relaciones. En los diagramas de clases se identifican tres tipos de clases: Interfaz, Controladora y Entidad.

Las clases interfaz se utilizan para modelar la interacción entre el sistema y sus actores. Esta interacción implica recibir información y peticiones de los usuarios y sistemas externos. Las clases interfaz representan abstracciones de ventanas, formularios, etc. Cada clase de interfaz debe asociarse con al menos un actor y viceversa.

Las clases controladoras se usan con para encapsular el control de cada caso de uso. Los aspectos dinámicos del sistema se modelan con clases controladoras, debido a que ellas manejan y coordinan las acciones y los flujos de control principales y delegan trabajo a otros objetos, objetos de interfaz y de entidad.

Las clases entidades se utilizan para modelar la información que posee una vida persistente. Las clases entidad suelen mostrar una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema.

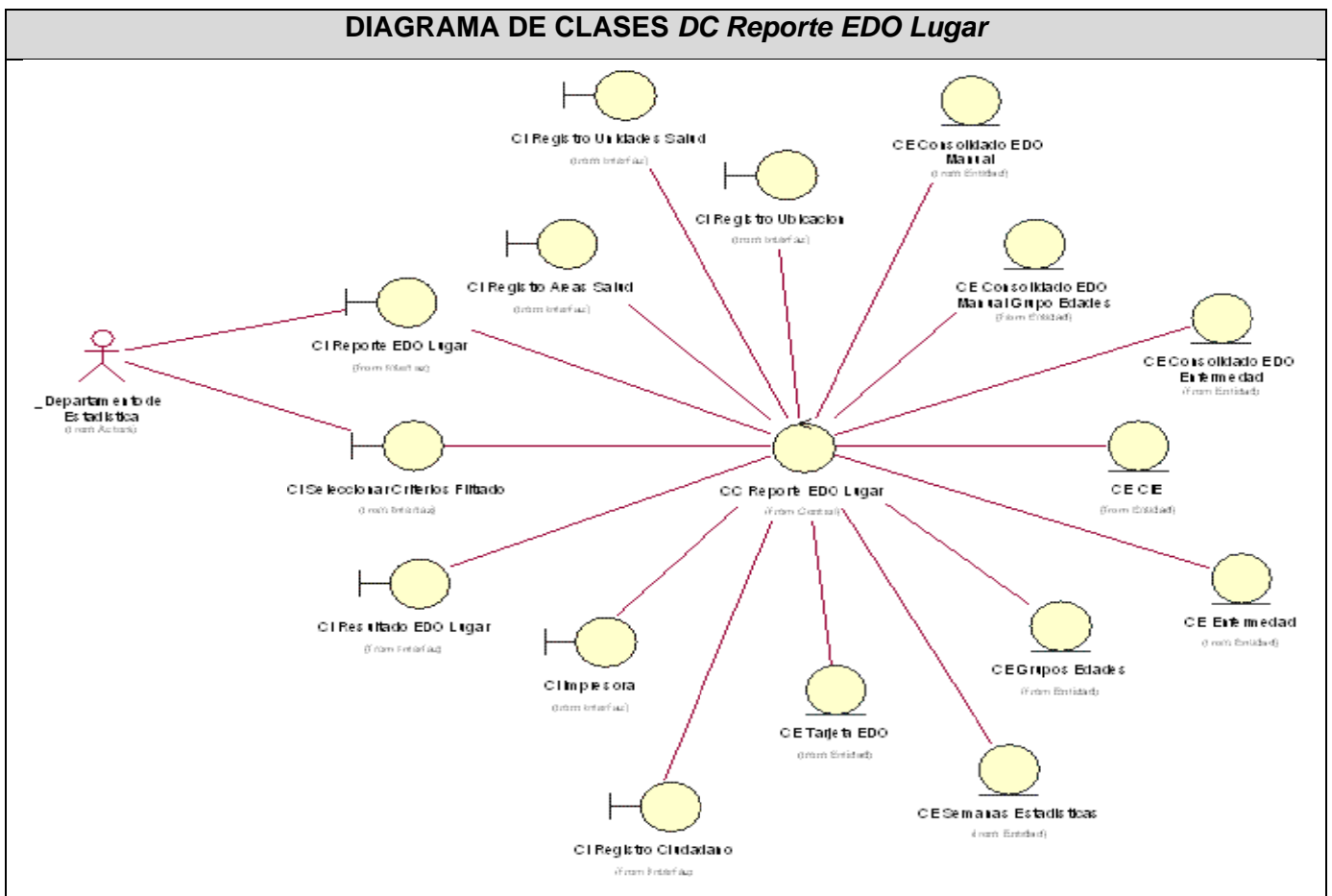


Figura 3.1 Diagrama de Clases del Análisis DC Reporte EDO Lugar

Descripción de las Clases del Análisis

❖ Clases Interfaz:

- CI Reporte EDO Lugar: busca la cantidad de casos de determinadas EDO en las zonas geográficas seleccionadas.

- CI Reporte EDO Semanas Estadísticas: busca la cantidad de casos de determinadas EDO en las semanas del año en curso seleccionadas.
 - CI Reporte EDO Semanas Meses: busca la cantidad de casos de determinadas EDO en los meses del año en curso seleccionados.
 - CI Reporte EDO Semanas Semestres: busca la cantidad de casos de determinadas EDO en los semestres del año en curso seleccionados.
 - CI Reporte EDO Grupos Edades: busca la cantidad de casos de determinadas EDO en las etapas de desarrollo seleccionadas.
 - CI Reporte EDO Lugar y Semanas Estadísticas: busca la cantidad de casos de una EDO específica en los lugares y semanas del año en curso seleccionadas.
 - CI Reporte EDO Lugar y Grupos Edades: busca la cantidad de casos de una EDO específica en los lugares y etapas de desarrollo seleccionadas.
 - CI Resultado EDO Lugar: muestra la cantidad de casos de determinadas EDO en las zonas geográficas seleccionadas.
 - CI Resultado EDO Periodo: muestra la cantidad de casos de determinadas EDO en las semanas del año en curso seleccionadas.
 - CI Resultado EDO Grupos Edades: muestra la cantidad de casos de determinadas EDO en las etapas de desarrollo seleccionadas.
 - CI Resultado EDO Lugar y Semanas Estadísticas: muestra la cantidad de casos de una EDO específica en los lugares y semanas del año en curso seleccionadas.
 - CI Resultado EDO Lugar y Grupos Edades: muestra la cantidad de casos de una EDO específica en los lugares y etapas de desarrollo seleccionadas.
 - CI Seleccionar Criterios Filtrado: muestra los criterios de búsqueda opcionales por los que puede filtrarse la información estadística, ej.: sexo, tipo de enfermedad, tipo de diagnóstico, tipo de notificación, tipo de fecha.
 - CI Registro Ubicación: muestra las provincias y municipios correspondientes a una provincia específica.
 - CI Registro Áreas Salud: muestra las áreas de salud de un municipio específico.
 - CI Registro Unidades Salud: muestra las unidades de salud de un área de salud específica.
 - CI Registro Ciudadano: muestra el sexo de un individuo determinado.
- ❖ **Clases Control:**
- CC Reporte EDO Lugar: coordina las actividades de los objetos que implementan el caso de uso Obtener Reporte de EDO por Lugar.

- CC Reporte EDO Periodo: coordina las actividades de los objetos que implementan el caso de uso Obtener Reporte de EDO por Periodo.
 - CC Reporte EDO Grupos Edades: coordina las actividades de los objetos que implementan el caso de uso Obtener Reporte de EDO por Grupos de Edades.
 - CC Reporte EDO Lugar y Semanas Estadísticas: coordina las actividades de los objetos que implementan el caso de uso Obtener Reporte de EDO por Lugar y Semanas Estadísticas.
 - CC Reporte EDO Lugar y Grupos Edades: coordina las actividades de los objetos que implementan el caso de uso Obtener Reporte de EDO por Lugar y Semanas Estadísticas.
- ❖ **Clases Entidad:**
- CE CIE: contiene la información sobre los datos que relacionan a una EDO con una CIE.
 - CE Enfermedad: contiene la información sobre los datos de las EDO.
 - CE Grupos Edades: contiene la información sobre los datos de los intervalos de edades de cada grupo de edad.
 - CE Semanas Estadísticas: contiene la información sobre los datos de las semanas del año.
 - CE Tarjeta EDO: contiene la información sobre los datos de las personas con EDO captada individualmente.
 - CE Consolidado EDO Manual: contiene la información sobre los datos de un grupo de personas con EDO captadas colectivamente.
 - CE Consolidado EDO Enfermedad: contiene la información sobre los datos que relacionan una EDO con un Consolidado.
 - CE Consolidado EDO Manual Grupo Edades: contiene la información sobre los datos que relacionan un Grupo de Edades con un Consolidado.

3.2 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso especificando cómo los requisitos funcionales y no funcionales tienen impacto en el sistema. Este artefacto constituye la entrada fundamental utilizada para el correcto desarrollo de las actividades de implementación. Entre los propósitos del Modelo de Diseño se encuentran:

- ❖ 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.
- ❖ Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- ❖ Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

Patrones Estructurales de Diseño

Los patrones de diseño son búsquedas de soluciones a problemas comunes en el desarrollo de software. Un patrón es una solución efectiva de un problema, que lo ha resuelto en otras ocasiones y que puede ser reusada en otras circunstancias.

La función de los patrones estructurales es fundamentalmente modelar las relaciones entre los objetos para crear estructuras con cierto grado de complejidad. Aunque existen también dentro de este grupo otros que se encargan de las relaciones de herencia y composición entre las clases.

Fachada: El objetivo de este patrón es simplificar la interface entre sistemas o componentes de software ocultando tras una clase fachada todo un sistema de interacción complejo. Se trata de ocultar todo lo posible la complejidad del sistema, conjunto de componentes que lo forman, para así ofrecer el mínimo posible de puntos de entrada al sistema oculto por fachada.

El patrón Fachada ofrece además la ventaja de aislar los posibles cambios que tengan lugar en algunas de las partes que interactúa, un ejemplo de lo anterior sería migrar la capa de negocio hacia php5 sin afectar el correcto funcionamiento de la capa de presentación.

Proxy: Este patrón obliga a que las llamadas a un objeto ocurran indirectamente a través de un objeto proxy, el cual actúa sustituyendo al original y luego delegando las llamadas a los métodos de las clases respectivas.

Este patrón se usa frecuentemente incluso en otros patrones. El mismo resuelve el problema del alto consumo de memoria o recursos de la máquina que algunos objetos realizan, ya que se instancian estos objetos solo en el momento que se solicita por el cliente.

Los objetos del cliente llaman a los métodos del objeto proxy con lo cual invoca los métodos del objeto específico que brinda ese servicio. La razón fundamental para usar un proxy es la transparencia en la administración de los servicios de otros objetos. Con este patrón se obliga a que todos los accesos a servicios provistos por objetos pasen a través de un objeto proxy sin que los objetos clientes

se den cuenta de que en realidad no están invocando a los métodos directamente en el objeto respectivo.

Los métodos en la aplicación no se invocan directamente sino a través de proxpla (Proxy-PlaSer), este componente es el encargado de administrar los servicios y acciones.

Definición de los Elementos de Diseño

Para el desarrollo del Modelo de Diseño es necesario establecer una descomposición del mismo en subsistemas, con sus interfaces y las dependencias. Esta representación es muy significativa para la arquitectura en general, debido a que los subsistemas y sus interfaces constituyen la estructura fundamental del producto de software.

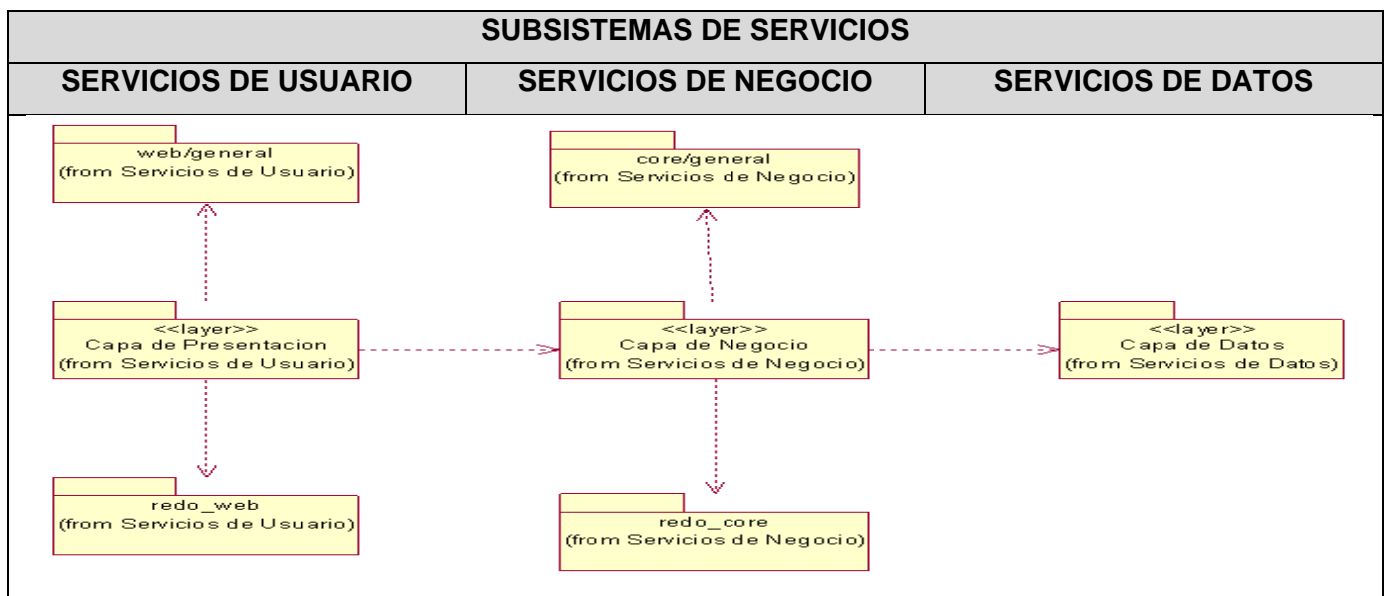


Figura 3.2 Subsistemas de Servicios

Para la adecuada comprensión de los Diagramas de Clases del Diseño se relaciona a continuación el contenido de cada uno de los subsistemas de servicios:

- ❖ **Capa de Presentación/redo_web:** Las funcionalidades de la capa de presentación consisten en intercambiar información con los actores, por lo tanto es la única capa con la que interactúa directamente un usuario. Contiene los ficheros mediante los cuales son invocados los métodos de la Capa de Negocio, entre ellos pueden ser mencionados los documentos XSL utilizados para la transformación de los XML y ficheros JavaScript utilizados para las validaciones y tratamiento de errores o excepciones.

- **web/general:** En este subsistema se encuentran las librerías PlaSer_XML y PlaSer_XSLT, permiten la manipulación de los documentos XML y el formateo de estos para ser mostrados a los clientes a través de páginas XHTML respectivamente.
- ❖ **Capa de Negocio/redo_core:** Establece la comunicación entre la capa de presentación y la capa de datos, encargada de recibir y responder cada petición de los usuarios. Los ficheros que la conforman reciben las solicitudes de los clientes, se comunican con la capa de datos, actualizando o recuperando información emitiendo una respuesta. Constituye la parte del sistema donde se establecen todas las reglas de negocio que deben cumplirse.
 - **core/general:** Representa los ficheros utilizados en la Capa de Negocio para las validaciones y generación de ficheros en formato Portable Document Format (PDF) o Microsoft Office Excel (XLS).
- ❖ **Capa de Datos:** Representa las tablas de la Base de Datos del REDO y las Bases de Datos Externas, es el repositorio físico de la información gestionada por el sistema, la cual puede ser procesada por los Reportes Estadísticos a través de la Capa de Negocio.

Modelado mediante Estereotipos Web

Para la realización de los Diagramas de Clases del Diseño será utilizada la extensión de UML para el modelado de aplicaciones Web, publicada por Jim Conallen en 1999 en el artículo *Modelling Web Applications Architectures with UML*. Esta extensión presenta como elementos más significativos a tres clases UML: Server Page, Client Page y Form empleadas para el código servidor, código cliente y formularios respectivamente, permitiendo además representar ficheros contenedores de sentencias script.

RELACIONES ENTRE LOS ELEMENTOS DE DISEÑO			
DESDE/HASTA	Client Page	Form	Server Page
Client Page	<<link>> <<redirect>>	aggregation	<<link>>
Form	aggregation by		<<submit>>
Server Page	<<build>>		<<include>>

Tabla 3.1 Relaciones entre Elementos del Diseño

- ❖ Entre las páginas servidoras pueden existir relaciones de inclusión (<<include>>).
- ❖ Las páginas servidoras construyen el resultado XHTML que conforma el código cliente (<<build>>).

- ❖ Los formularios forman parte del resultado XHTML (<<aggregation / aggregation by>>).
- ❖ Los formularios envían los datos al código servidor para su procesamiento (<<submit>>).
- ❖ Entre las páginas clientes pueden existir vínculos (<<link>>) o re-direccionamientos (<<redirect>>).
- ❖ Las páginas clientes pueden incluir ficheros script (<<include>>).

Diagramas de Clases del Diseño

Un Diagrama de Clases del Diseño es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de diseño de un sistema para crear el diseño conceptual de la información que se manejará en el mismo y los componentes que se encargaran su funcionamiento así como la relación entre ellos.

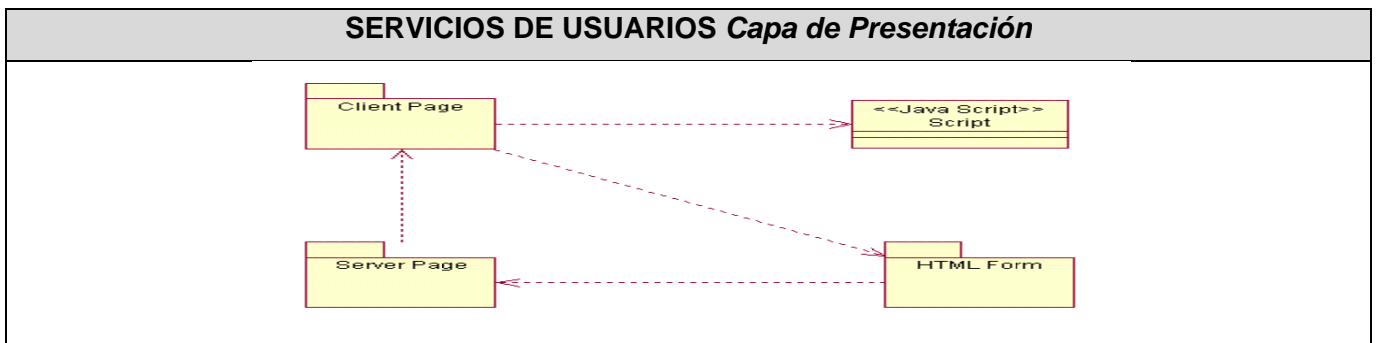


Figura 3.3 Servicios de Usuario *Capa de Presentación*

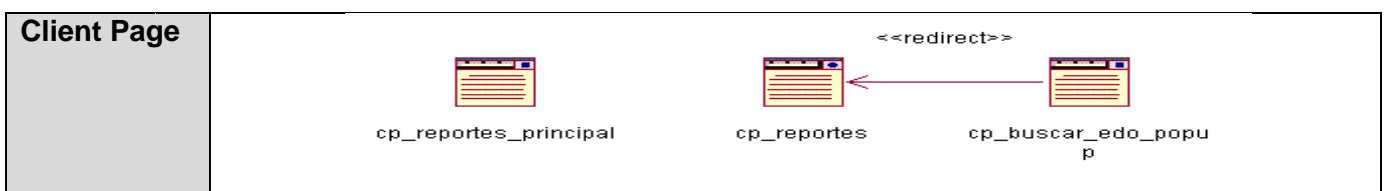


Figura 3.4 Capa de Presentación *Client Page*

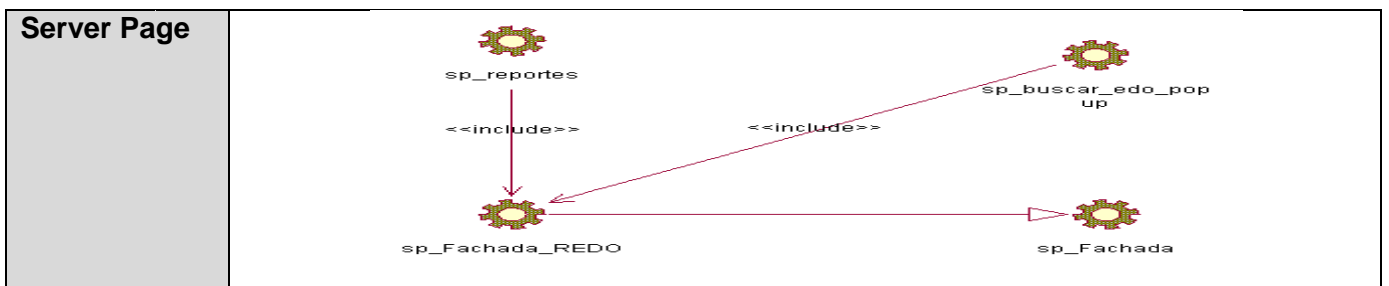


Figura 3.5 Capa de Presentación *Server Page*

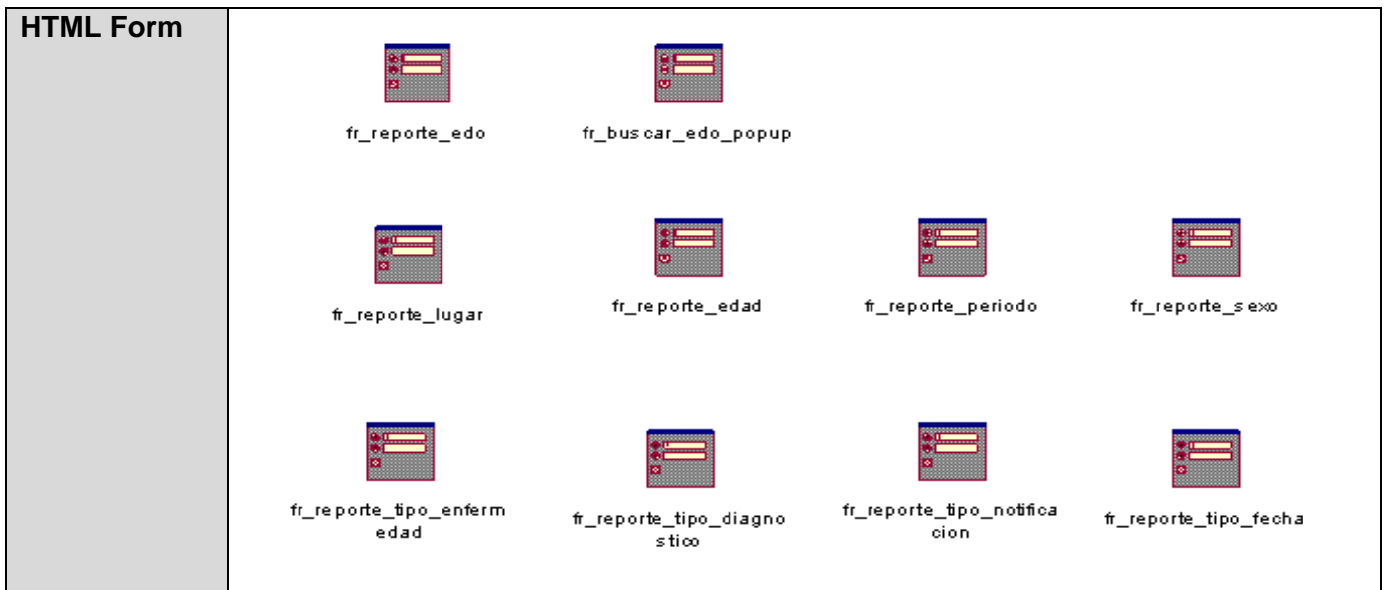


Figura 3.6 Capa de Presentación *HTML Form*



Figura 3.7 Servicios de Negocio *Capa de Negocio*

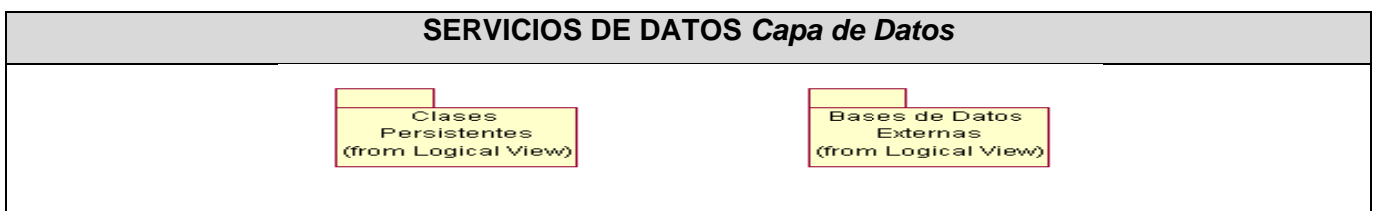


Figura 3.8 Servicios de Datos *Capa de Datos*

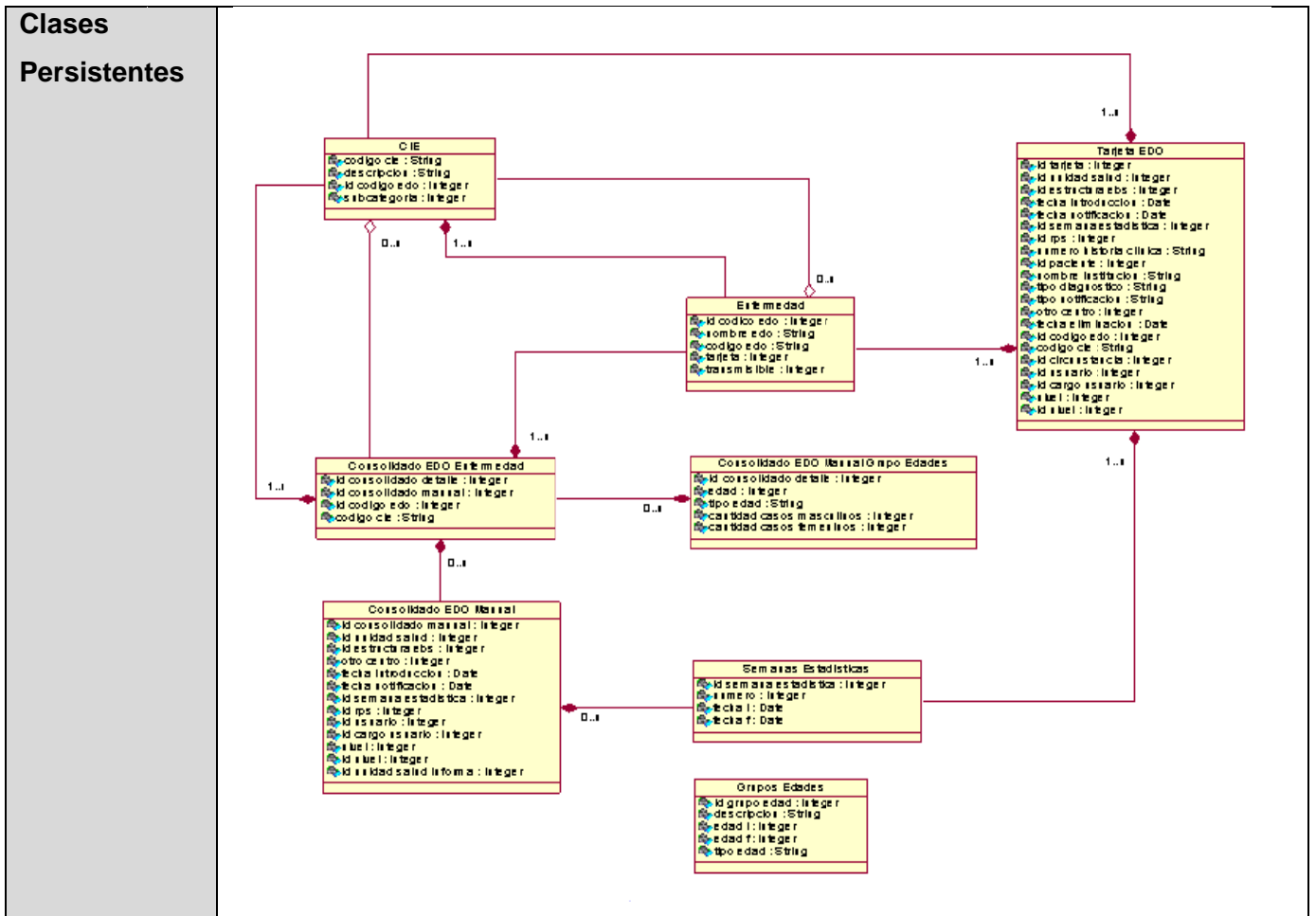


Figura 3.9 Capa de Datos Clases Persistentes

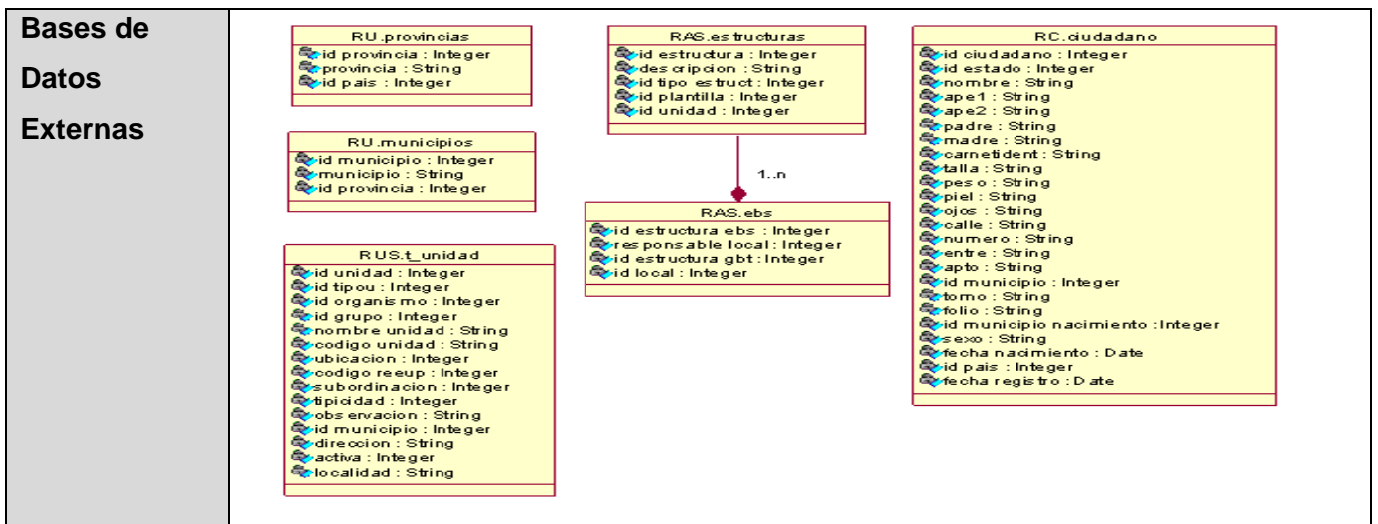


Figura 3.10 Capa de Datos Bases de Datos Externas

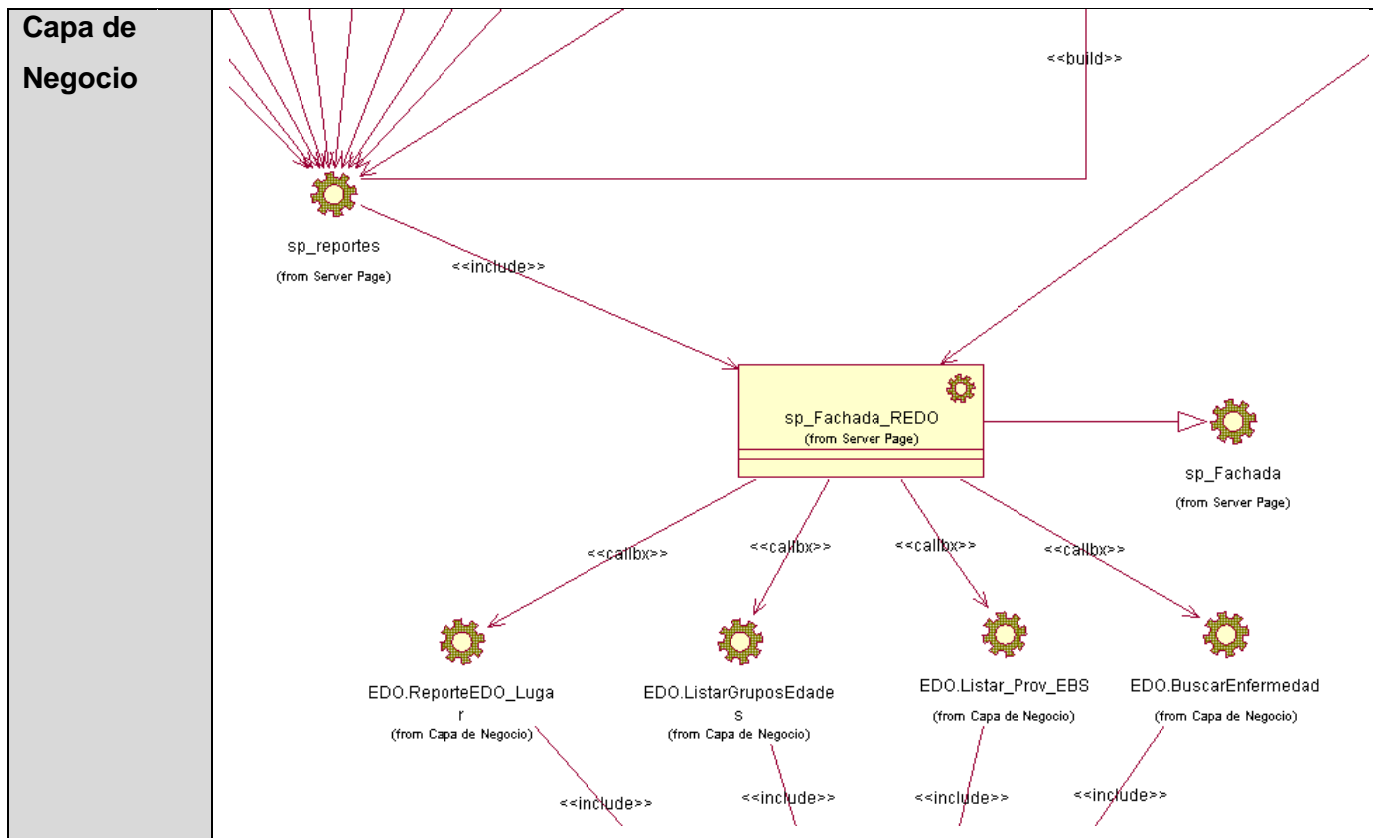


Figura 3.11 (c) Diagrama de Clases del Análisis DCD Reporte EDO Lugar

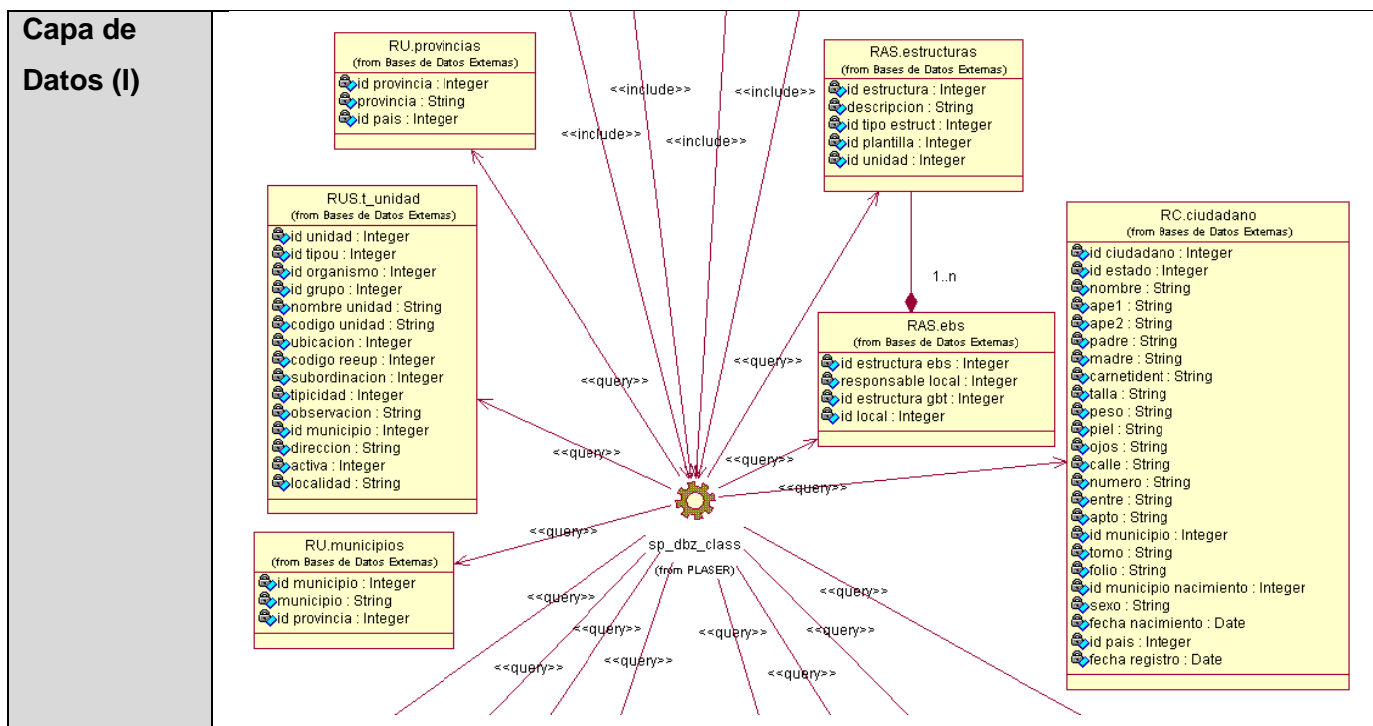


Figura 3.11 (d) Diagrama de Clases del Análisis DCD Reporte EDO Lugar

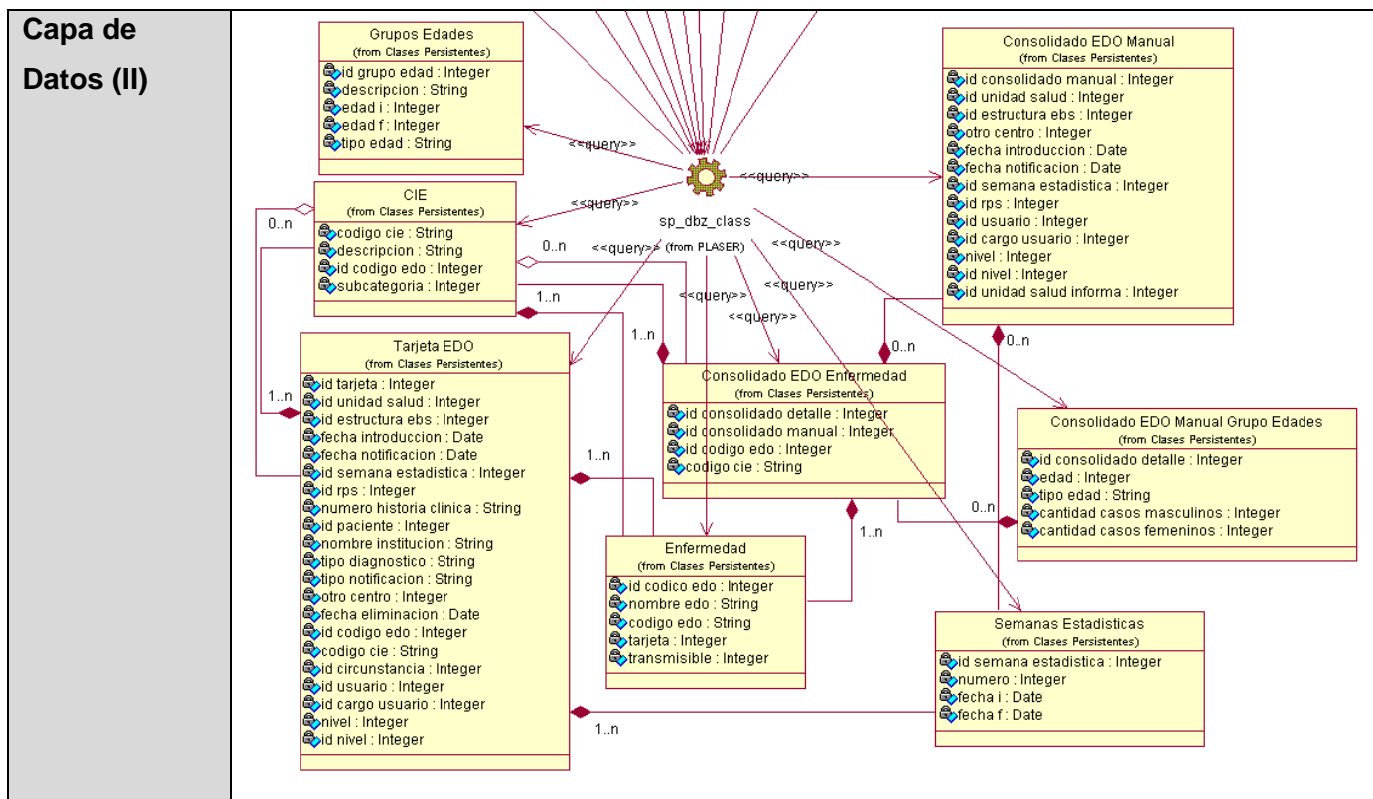


Figura 3.11 (e) Diagrama de Clases del Análisis DCD Reporte EDO Lugar

Los Diagramas de Clases del Diseño de cada uno de los 5 Casos de Uso del Sistema varían únicamente en la llamada (<<callBx>>) que realiza la Server Page sp Fachada REDO a las Server Pages del Negocio correspondientes a cada uno de los métodos del redo_core invocados.

El Diagrama de Clases del Diseño del Caso de Uso del Sistema Obtener Reporte de EDO por Periodo es el único caso en que la Server Page sp Fachada REDO realiza una llamada a 6 métodos del redo_core (Anexo I), 3 de ellos relacionados directamente con el CUS en contraposición con los restantes CUS que se relacionan directamente con un método del redo_core per cápita (BuscarEnfermedad.php, ListarProv_EBS.php y ListarGruposEdades.php son comunes para todos los CUS).

MÉTODOS DEL REDO CORE	FUNCIONALIDADES DEL SISTEMA
ReporteEDO_Periodo.php	EDO por Semanas Estadísticas
ReporteEDO_Meses.php	EDO por Meses
ReporteEDO_Semestres.php	EDO por Semestres

Tabla 3.2 Métodos del Negocio y Funcionalidades del Sistema

CUS Obtener Reporte de EDO por Periodo

Descripción de las Clases UML

La estructura general de la Capa de Presentación para cada uno de los casos de uso identificados para el desarrollo de los Reportes Estadísticos es similar, contando con las mismas páginas clientes y servidoras. Del mismo modo ocurre con las páginas servidoras que responden a la Lógica de Negocio. Con la descripción de estas clases se tendrá una comprensión mayor del funcionamiento del sistema a desarrollar.

Descripción de las Páginas Clientes

NOMBRE:	cp reportes principal
TIPO DE CLASE:	Client Page
DESCRIPCION:	<p>Constituye el menú desplegable de la opción de Reportes Totales del REDO. Las 7 opciones disponibles en el mismo son: EDO por Lugar, EDO por Grupos de Edades, EDO por Semanas Estadísticas, EDO por Meses, EDO por Semestres, EDO por Lugar y Grupos de Edades y EDO por Lugar y Semanas Estadísticas.</p> <p>Tiene un vínculo (<<link>>) a la Server Page sp reportes y está involucrada en los siguientes Casos de Usos del Sistema: Obtener Reporte EDO por Lugar, Obtener Reporte EDO por Grupos de Edades, Obtener Reporte EDO por Periodo, Obtener Reporte EDO por Lugar y Grupos de Edades y Obtener Reporte EDO por Lugar y Semanas Estadísticas.</p>

Tabla 3.3 Descripción de la Página Cliente *cp reportes principal*

NOMBRE:	cp reportes
TIPO DE CLASE:	Client Page
DESCRIPCION:	<p>Constituye la página de selección de criterios de búsqueda para filtrar la información relativa al reporte en cuestión y en la misma se muestra el resultado de la búsqueda de los datos. Dispone de formas representativas para dichos criterios: Generales (nombre propiamente de los criterios) y Cualitativos (valores que pueden tomar los mismos). Los 9 criterios de selección disponibles son los siguientes: Enfermedad, Lugar, Periodo, Edad, Sexo, Tipo de Enfermedad, Tipo de Diagnóstico, Tipo de Notificación y Tipo de Fecha.</p>

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

	<p>Tiene (<<aggregation>>) un formulario (<<HTML Form>>) por cada criterio de búsqueda y un vínculo a la Server Page sp buscar edo popup. Está involucrada en los siguientes Casos de Uso del Sistema:</p> <p>Obtener Reporte EDO por Lugar, Obtener Reporte EDO por Grupos de Edades, Obtener Reporte EDO por Periodo, Obtener Reporte EDO por Lugar y Grupos de Edades y Obtener Reporte EDO por Lugar y Semanas Estadísticas.</p>
--	--

Tabla 3.4 Descripción de la Página Cliente *cp reportes*

NOMBRE:	cp buscar edo popup
TIPO DE CLASE:	Client Page
DESCRIPCION:	<p>Constituye la ventana de filtrado de Enfermedades y sus respectivas CIE. La búsqueda de las EDO se puede realizar por 2 criterios: Código EDO y/o Descripción EDO. Tiene un formulario para la búsqueda y selección de las EDO y CIE asociadas y realiza un re-direccionamiento (<<redirect>>) a la Client Page cp reportes principal. Está involucrada en los siguientes Casos de Uso del Sistema:</p> <p>Obtener Reporte EDO por Lugar, Obtener Reporte EDO por Grupos de Edades, Obtener Reporte EDO por Periodo, Obtener Reporte EDO por Lugar y Grupos de Edades y Obtener Reporte EDO por Lugar y Semanas Estadísticas.</p>

Tabla 3.5 Descripción de la Página Cliente *cp buscar edo popup*

Descripción de las Páginas Servidoras de Presentación

NOMBRE:	sp reportes
TIPO DE CLASE:	Server Page
DESCRIPCION:	<p>Construye (<<build>>) la Client Page cp reportes para la selección de criterios de filtrado y la representación del resultado de la búsqueda. Invoca a través de la Server Page sp Fachada REDO (<<include>>) a los siguientes métodos del negocio:</p> <p>Listar_Prov_EBS.php, ListarGruposEdades.php, ReporteEDO_Lugar.php, ReporteEDO_GrupoEdades.php, ReporteEDO_Periodo.php, ReporteEDO_Meses.php, ReporteEDO_Semestres.php, ReporteEDO_Lugar_Grupo.php y ReporteEDO_Periodo.php</p> <p>Aplica un fichero XSLT al documento XML retornado para transformarlo y</p>

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

	<p>mostrarlo al usuario en formato XHTML. Está involucrada en los siguientes Casos de Uso del Sistema:</p> <p>Obtener Reporte EDO por Lugar, Obtener Reporte EDO por Grupos de Edades, Obtener Reporte EDO por Periodo, Obtener Reporte EDO por Lugar y Grupos de Edades y Obtener Reporte EDO por Lugar y Semanas Estadísticas.</p>
--	--

Tabla 3.6 Descripción de la Página Servidora *sp reportes*

NOMBRE:	sp buscar edo popup
TIPO DE CLASE:	Server Page
DESCRIPCION:	<p>Construye la Client Page cp buscar edo popup. Invoca a través de la Server Page sp Fachada REDO al método del negocio BuscarEnfermedad.php. Aplica un fichero XSLT al documento XML retornado para transformarlo y mostrarlo al usuario en formato XHTML. Está involucrada en los siguientes Casos de Uso del Sistema:</p> <p>Obtener Reporte EDO por Lugar, Obtener Reporte EDO por Grupos de Edades, Obtener Reporte EDO por Periodo, Obtener Reporte EDO por Lugar y Grupos de Edades y Obtener Reporte EDO por Lugar y Semanas Estadísticas.</p>

Tabla 3.7 Descripción de la Página Servidora *sp buscar edo popup*

NOMBRE:	sp Fachada REDO
TIPO DE CLASE:	Server Page
DESCRIPCION:	<p>Hereda las funcionalidad de la Server Page sp Fachada y modela la comunicación entre la Capa de Presentación y la Capa de Negocio. Realiza las llamadas (<<callBx>>) a los métodos del negocio (cada uno representado por una Server Page) invocados desde las Server Pages sp reportes y sp buscar edo popup.</p>

Tabla 3.8 Descripción de la Página Servidora *sp Fachada REDO*

Descripción de las Páginas Servidoras de Negocio

NOMBRE:	EDO.Listar Prov EBS	
TIPO DE CLASE:	Server Page	
ATRIBUTOS:	TIPO:	
profundidad		int

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

DESCRIPCION:	Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta (<<query>>) a las tablas de bases de datos externas RU.provincias, RU.municipios, RUS.t_unidad, RAS.estructuras y RAS.ebs. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado de las provincias, municipios, unidades de salud, GBT y EBS según el nivel de acceso y la ubicación geográfica del actor.
---------------------	--

Tabla 3.9 Descripción de la Página Servidora *EDO.Listar Prov EBS*

NOMBRE:	EDO.ListarGruposEdades		
TIPO DE CLASE:	Server Page		
ATRIBUTOS:	TIPO:		
-	-		
DESCRIPCION:	Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a la tabla Grupos Edades. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado de todos los grupos de edades definidos.		

Tabla 3.10 Descripción de la Página Servidora *EDO.ListarGruposEdades*

NOMBRE:	EDO.BuscarEnfermedad		
TIPO DE CLASE:	Server Page		
ATRIBUTOS:	TIPO:	ATRIBUTOS:	TIPO:
codigo_edo	string	profundidad	int
nombre_edo	string	ordenar	string
offset	int	ordenar por	string
cantidad	int		
DESCRIPCION:	Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a la tabla Enfermedad. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado de todas las EDO que cumplen las restricciones impuestas para su búsqueda, así como las CIE asociadas a cada una de ellas.		

Tabla 3.11 Descripción de la Página Servidora *EDO.BuscarEnfermedad*

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

NOMBRE:	EDO.ReporteEDO_Lugar		
TIPO DE CLASE:	Server Page		
ATRIBUTOS:	TIPO:	ATRIBUTOS:	TIPO:
edo	array(int)	fecha_i	date
provincia	array(int)	fecha_f	date
municipio	array(int)	grupo_E	array(int)
unidad	array(int)	t_enfermedad	string
gbt	array(int)	t_diganostico	string
ebs	array(int)	T_notificacion	string
edad	array(int)	tipo_fecha	string
sexo	char	offset	int
semestre	array(int)	cantidad	int
semana	array(int)	offset_lateral	int
meses	array(int)	cantidad_lateral	int
DESCRIPCION:	<p>Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a las tablas de bases de datos externas RU.provincias, RU.municipio, RUS.t_unidad, RAS.estructuras y RAS.ebs. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado con la cantidad de casos notificados de determinadas EDO en los lugares seleccionados.</p>		

Tabla 3.12 Descripción de la Página Servidora *EDO.ReporteEDO_Lugar*

NOMBRE:	EDO.ReporteEDO_GrupoEdades		
TIPO DE CLASE:	Server Page		
ATRIBUTOS:	TIPO:	ATRIBUTOS:	TIPO:
edo	array(int)	fecha_i	date
provincia	array(int)	fecha_f	date
municipio	array(int)	grupo_E	array(int)
unidad	array(int)	t_enfermedad	string
gbt	array(int)	t_diganostico	string
ebs	array(int)	T_notificacion	string
sexo	char	offset	int

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

semestre	array(int)	cantidad	int
semana	array(int)	offset_lateral	int
meses	array(int)	cantidad_lateral	int
tipo_fecha	string		
DESCRIPCION:	Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a la tabla de base de datos externa RC.ciudadano y a la tabla Consolidado EDO Manual Grupo Edades. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado con la cantidad de casos notificados de determinadas EDO correspondientes a los grupos de edades seleccionados.		

Tabla 3.13 Descripción de la Página Servidora *EDO.ReporteEDO_GrupoEdades*

NOMBRE:	EDO.ReporteEDO_Periodo		
TIPO DE CLASE:	Server Page		
ATRIBUTOS:	TIPO:	ATRIBUTOS:	TIPO:
edo	array(int)	grupo_E	array(int)
provincia	array(int)	t_enfermedad	string
municipio	array(int)	t_diganostico	string
unidad	array(int)	T_notificacion	string
gbt	array(int)	tipo_fecha	string
ebs	array(int)	offset	int
edad	array(int)	cantidad	int
sexo	char	offset_lateral	int
semana	array(int)	cantidad_lateral	int
DESCRIPCION:	Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a las tablas Tarjeta EDO, Consolidado EDO Manual y Semanas Estadísticas. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado con la cantidad de casos notificados de determinadas EDO en las semanas estadísticas seleccionadas pertenecientes al año en curso.		

Tabla 3.14 Descripción de la Página Servidora *EDO.ReporteEDO_Periodo*

NOMBRE:	EDO.ReporteEDO_Meses
TIPO DE CLASE:	Server Page

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

ATRIBUTOS:		TIPO:		ATRIBUTOS:		TIPO:	
edo		array(int)		grupo_E		array(int)	
provincia		array(int)		t_enfermedad		string	
municipio		array(int)		t_diganostico		string	
unidad		array(int)		T_notificacion		string	
gbt		array(int)		tipo_fecha		string	
ebs		array(int)		offset		int	
sexo		char		cantidad		int	
edad		array(int)		offset_lateral		int	
meses		array(int)		cantidad_lateral		int	
DESCRIPCION:	Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a las tablas Tarjeta EDO, Consolidado EDO Manual y Semanas Estadísticas. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado con la cantidad de casos notificados de determinadas EDO en los meses seleccionados del año en curso.						

Tabla 3.15 Descripción de la Página Servidora *EDO.ReporteEDO_Meses*

NOMBRE:	EDO.ReporteEDO_Semestres						
TIPO DE CLASE:	Server Page						
ATRIBUTOS:		TIPO:		ATRIBUTOS:		TIPO:	
edo		array(int)		grupo_E		array(int)	
provincia		array(int)		t_enfermedad		string	
municipio		array(int)		t_diganostico		string	
unidad		array(int)		T_notificacion		string	
gbt		array(int)		tipo_fecha		string	
ebs		array(int)		offset		int	
sexo		char		cantidad		int	
semestre		array(int)		offset_lateral		int	
edad		array(int)		cantidad_lateral		int	
DESCRIPCION:	Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a las tablas Tarjeta EDO, Consolidado EDO Manual y Semanas Estadísticas. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado con la cantidad de casos notificados de determinadas						

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

	EDO en uno o ambos semestres del año en curso.
--	--

Tabla 3.16 Descripción de la Página Servidora *EDO.ReporteEDO_Semestres*

NOMBRE:	EDO.ReporteEDO_Lugar_Grupo		
TIPO DE CLASE:	Server Page		
ATRIBUTOS:	TIPO:	ATRIBUTOS:	TIPO:
edo	array(int)	fecha_i	date
provincia	array(int)	fecha_f	date
municipio	array(int)	t_enfermedad	string
unidad	array(int)	t_diganostico	string
gbt	array(int)	T_notificacion	string
ebs	array(int)	tipo_fecha	string
sexo	char	offset	int
semestre	array(int)	cantidad	int
semana	array(int)	offset_lateral	int
meses	array(int)	cantidad_lateral	int
grupo_E	array(int)		
DESCRIPCION:	<p>Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a las tablas de las bases de datos externas RU.provincias, RU.municipios, RUS.t_unidad, RAS.estructuras, RAS.ebs y RC.ciudadano y la tabla Consolidado EDO Manual Grupo Edades. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado con la cantidad de casos de una EDO específica correspondiente a los lugares y grupos de edades seleccionados.</p>		

Tabla 3.17 Descripción de la Página Servidora *EDO.ReporteEDO_Lugar_Grupo*

NOMBRE:	EDO.ReporteEDO_Lugar_Periodo		
TIPO DE CLASE:	Server Page		
ATRIBUTOS:	TIPO:	ATRIBUTOS:	TIPO:
edo	array(int)	grupo_E	array(int)
provincia	array(int)	t_enfermedad	string
municipio	array(int)	t_diganostico	string
unidad	array(int)	T_notificacion	string

gbt	array(int)	tipo_fecha	string
ebs	array(int)	offset	int
sexo	char	cantidad	int
semana	array(int)	offset_lateral	int
edad	array(int)	cantidad_lateral	int
DESCRIPCION:	Tiene una relación de inclusión con la sp dbz class a través de la cual realiza una consulta a las tablas Tarjeta EDO, Consolidado EDO Manual y Semanas Estadísticas. Retorna un XML Response a la Server Page sp Fachada REDO. El resultado es un listado con la cantidad de casos de una EDO específica correspondiente a los lugares y semanas estadísticas seleccionados del año en curso.		

Tabla 3.18 Descripción de la Página Servidora *EDO.ReporteEDO_Lugar_Periodo*

Diagramas de Interacción del Diseño

Los Diagramas de Interacción son los artefactos que UML brinda para expresar las interacciones entre objetos para cumplir con los requerimientos del sistema. Entre los aspectos más importantes a considerar para la realización de estos diagramas se encuentra:

- ❖ Se clasifican en 2 tipos: Diagramas de Secuencia y Diagramas de Colaboración.
- ❖ Modelan aspectos dinámicos del sistema.
- ❖ Se utilizan para realizar un traza de la ejecución de un escenario.
- ❖ A cada escenario le corresponde un diagrama de interacción.
- ❖ Una interacción es un conjunto de objetos y sus relaciones, incluyendo los mensajes mediante los cuales pueden establecer comunicación.
- ❖ Un diagrama de secuencia destaca la ordenación temporal de los mensajes.
- ❖ Un diagrama de colaboración destaca la organización estructural de los objetos que envían y reciben mensajes.
- ❖ Un diagrama de interacción contiene:
 - Objetos.
 - Enlaces.
 - Mensajes.
- ❖ Un mensaje de interacción puede contener:
 - Notas.
 - Restricciones.

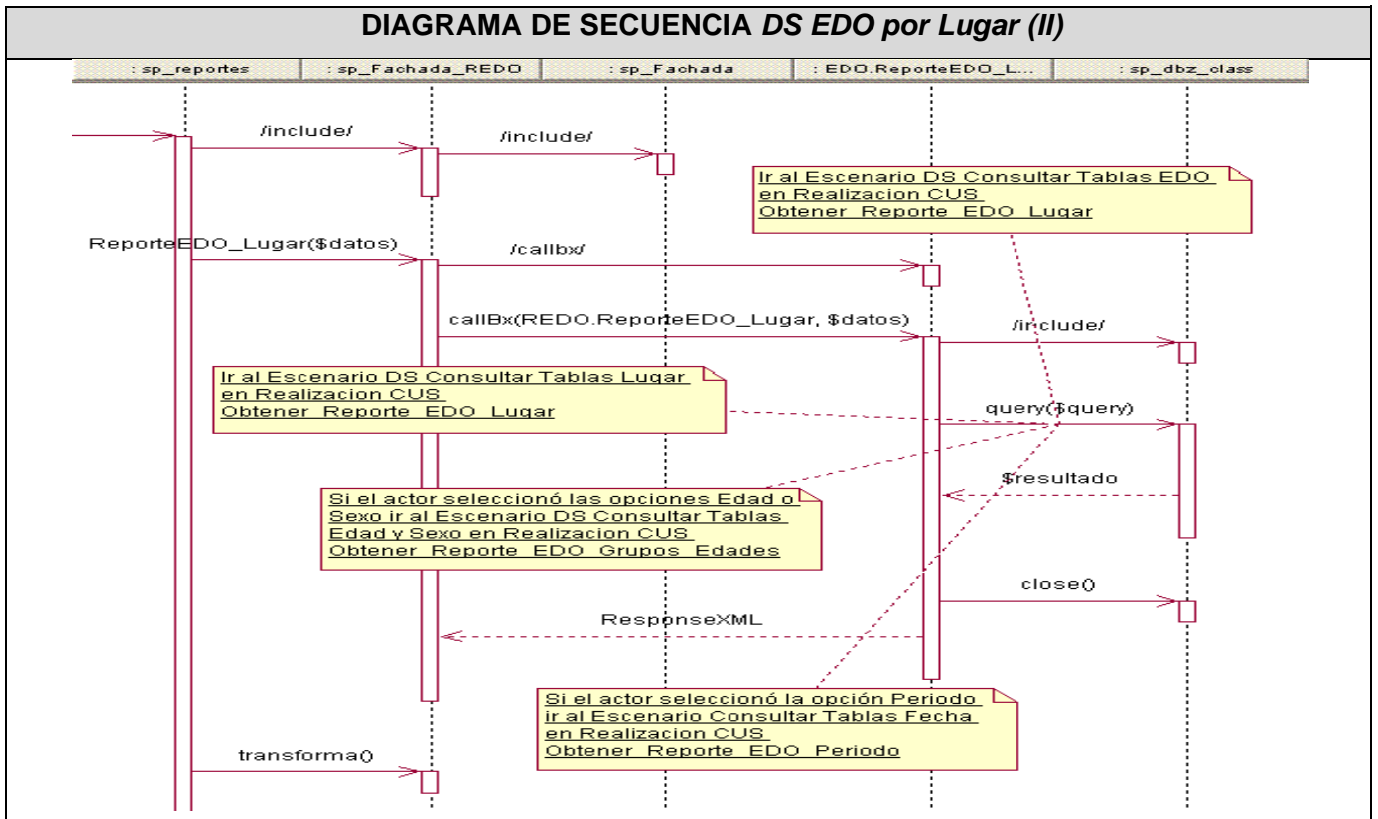


Figura 3.12 (b) Diagrama de Secuencia *EDO por Lugar*

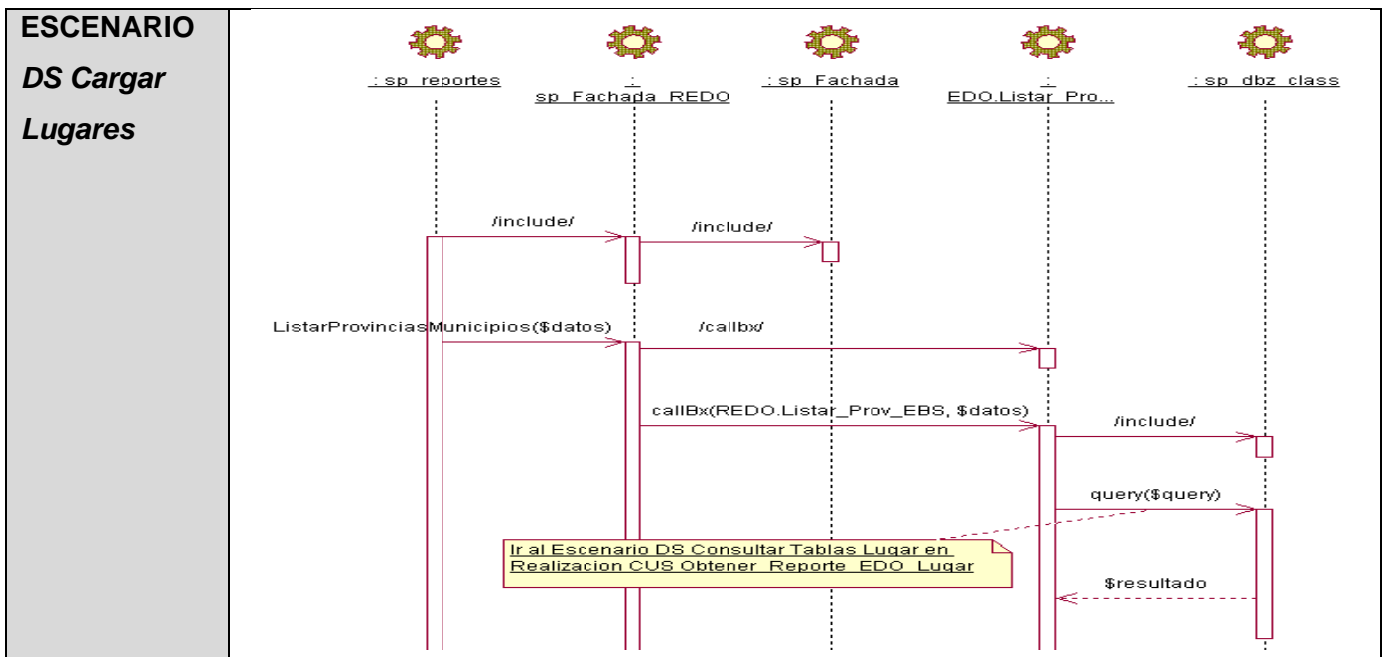


Figura 3.12 (c) Diagrama de Secuencia *EDO por Lugar*

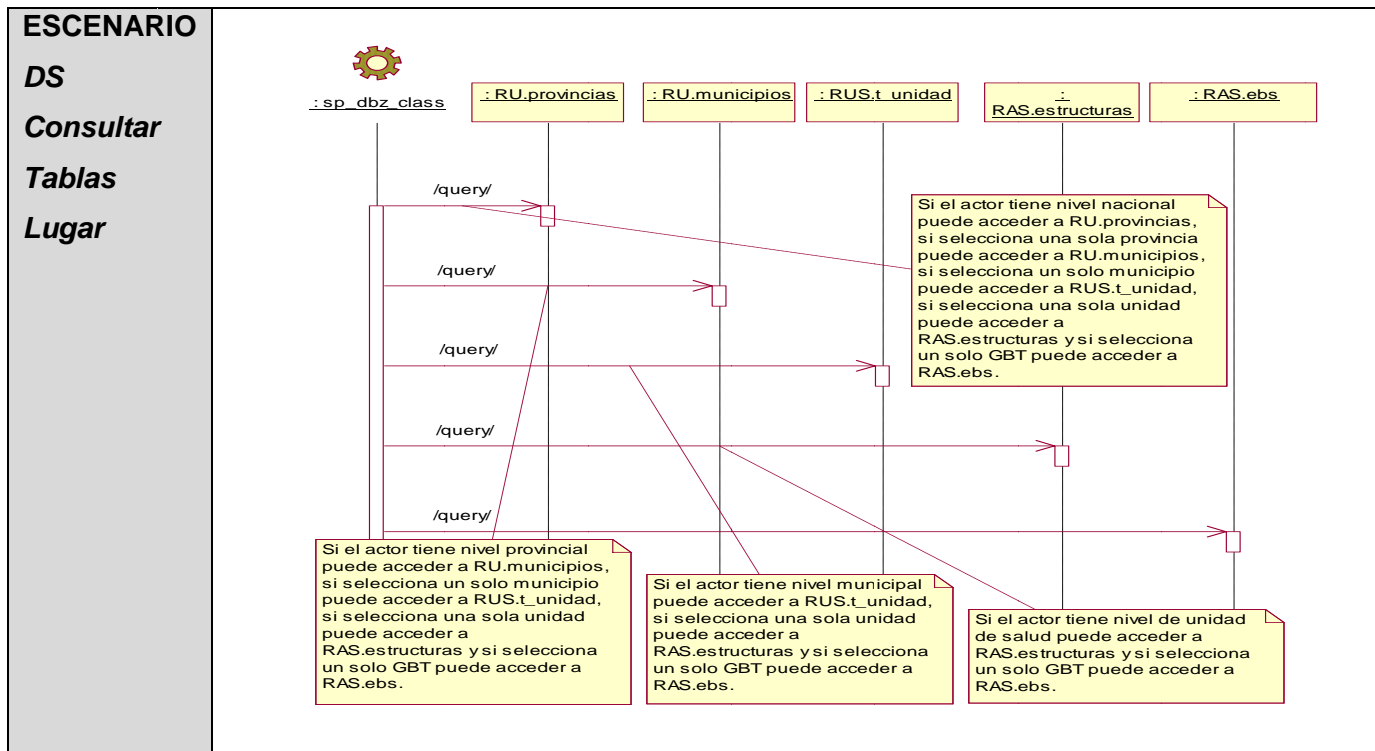


Figura 3.12 (d) Diagrama de Secuencia *EDO por Lugar*

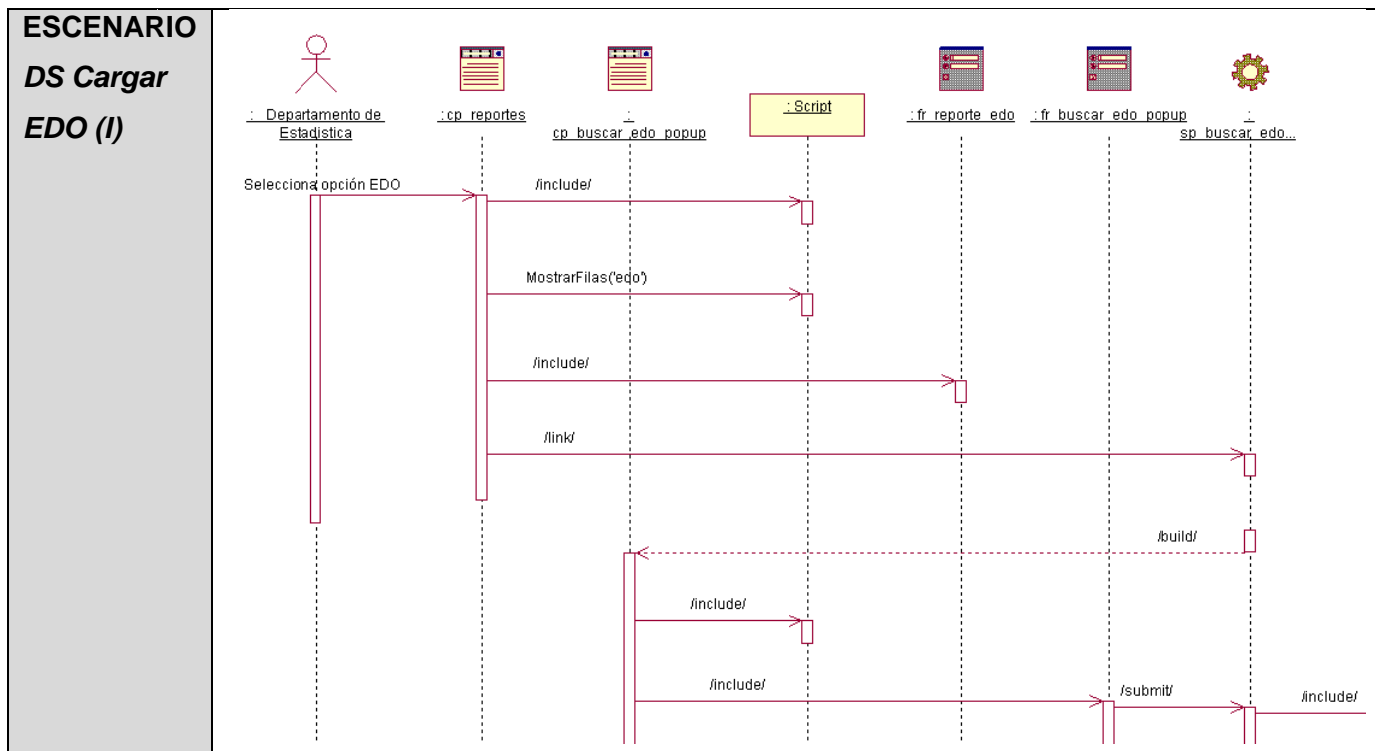


Figura 3.12 (e) Diagrama de Secuencia *EDO por Lugar*

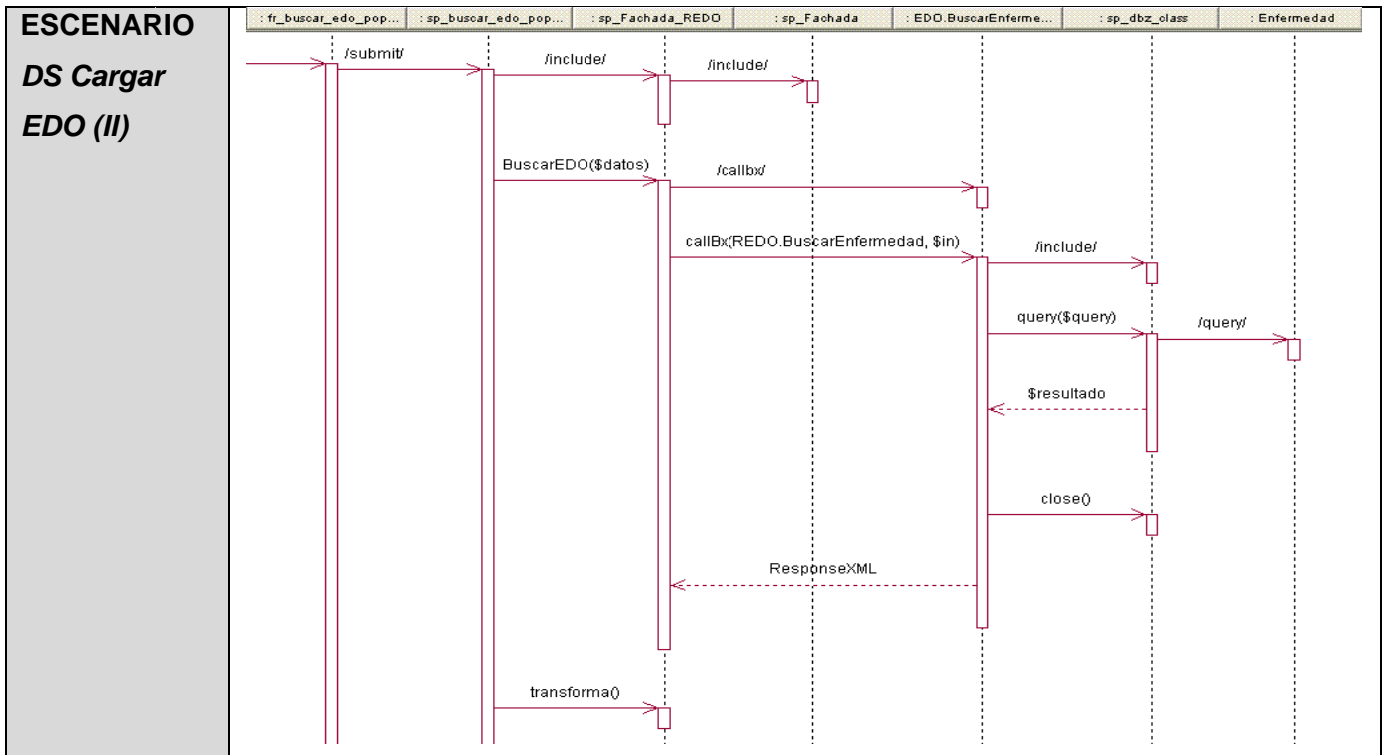


Figura 3.12 (f) Diagrama de Secuencia EDO por Lugar

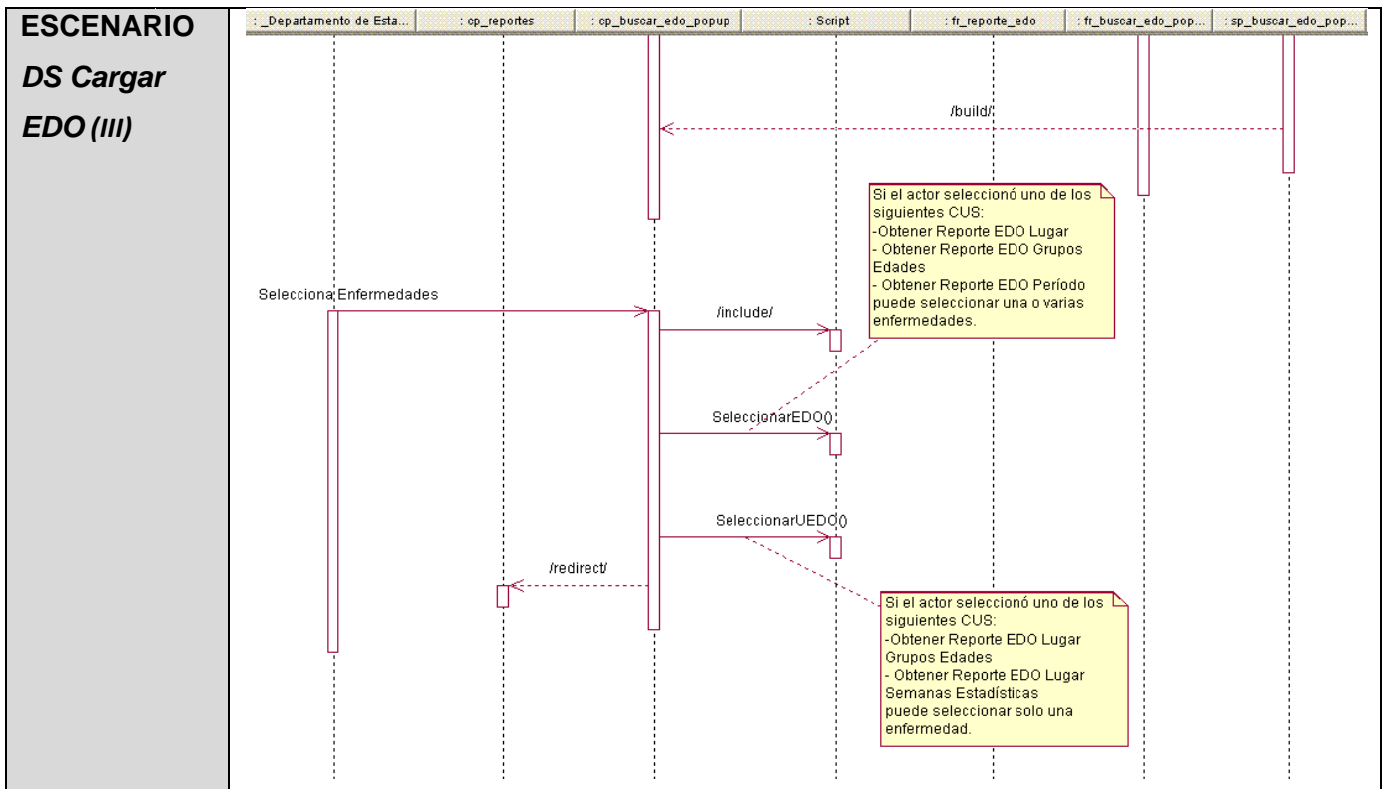


Figura 3.12 (g) Diagrama de Secuencia EDO por Lugar

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

El Escenario DS Seleccionar Opciones Periodo fue concebido específicamente para representar en el Diagrama de Secuencia EDO por Periodo la posibilidad que tiene el usuario del sistema de seleccionar entre las opciones de periodo siguientes: Semanas Estadísticas, Meses y Semestres. (Anexos II y III)

Los Diagramas de Colaboración expresan la interacción entre los objetos del diseño en tiempo de ejecución y se emplean solo si desea profundizar el nivel de detalle en los diagramas de secuencia y tener una visión del flujo de control en el contexto de la organización estructural de los objetos que colaboran entre sí.

En este capítulo, para la obtención del Modelo de Análisis del sistema propuesto se realizaron diferentes artefactos mediante los cuales es posible adquirir una visión detallada de las responsabilidades de cada Clase del Análisis para el correcto funcionamiento del sistema a desarrollar; los artefactos a saber son los Diagramas de Clases del Análisis y la Descripción de las Clases.

Así mismo, en el capítulo se obtuvo el Modelo de Diseño, donde se describió la realización física de los Casos de Uso del Sistema, centrándose en cómo los requerimientos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el producto software a desarrollar. Además, los artefactos generados durante la modelación del diseño brindan la entrada fundamental a las actividades que se realizan en el Flujo de Trabajo de Implementación, entre los artefactos realizados se encuentran los Diagramas de Clases del Diseño, la Descripción de las Clases y los Diagramas de Secuencia del Diseño.

CAPÍTULO 4: IMPLEMENTACIÓN

El presente capítulo constituye la secuencia lógica del Diseño del Sistema, a través del mismo se implementarán las clases y subsistemas encontrados durante el Flujo de Trabajo anterior en términos de componentes. Se justifica la integración del REDO con otros sistemas o componentes externos, específicamente para posibilitar la obtención de los Reportes Estadísticos. Como parte del Modelo de Implementación, se obtiene además el Diagrama de Despliegue, que indica cómo la solución implementada estará distribuida físicamente.

4.1 Justificación de la Integración con otros Sistemas Externos

La integración del sistema con otros componentes es un proceso imprescindible para el correcto funcionamiento del mismo sin necesidad de implementar servicios reutilizables o duplicar información almacenada en las bases de datos correspondientes a dichos sistemas externos. En el caso puntual de las Estadísticas Descriptivas del REDO, estas necesitan consumir servicios del componente SAAA (Single Authentication Authorization and Account) y solicitar información de los Registros de Áreas de Salud y Unidades de Salud e incluso a registros que no pertenecen al dominio de Salud Pública como los Registros de Ubicación y Ciudadano.

Single Authentication Authorization and Account (SAAA)

Este componente es el responsable de garantizar la seguridad de la aplicación verificando el acceso a la misma mediante un nombre de usuario único y una contraseña correspondiente, es por ello que cobra vital importancia para la plataforma SISalud en general. Además, para la obtención de las Estadísticas Descriptivas en particular, puesto que SAAA brinda información acerca del usuario logueado como tipo de usuario (administrador, editor o visualizador) y nivel de acceso (nacional, provincial, municipal y de área de salud). El nivel de acceso específicamente es imprescindible a la hora de mostrar la información solicitada en un Reporte Estadístico para retornar solamente los datos correspondientes a la ubicación geográfica del usuario:

- ❖ **Nivel de Acceso Nacional:** puede visualizar información correspondiente a todo el territorio nacional.
- ❖ **Nivel de Acceso Provincial:** puede visualizar solo la información correspondiente a la provincia a la que pertenece el usuario.

- ❖ **Nivel de Acceso Municipal:** puede visualizar solo la información correspondiente al municipio al que pertenece el usuario.
- ❖ **Nivel de Acceso de Unidad de Salud:** puede visualizar solo la información correspondiente a la unidad de salud a la que pertenece el usuario.

Registro de Ubicación (RU)

Este componente, uno de los que se encuentra fuera de las fronteras de Salud Pública, posee la información correspondiente a las provincias y municipios necesaria fundamentalmente para la obtención de los Reportes de EDO por Lugar, EDO por Lugar y Grupos de Edades y EDO por Lugar y Semanas Estadísticas a nivel nacional y/o provincial, aunque a estos mismos niveles de acceso puede dicha información ser necesaria para la obtención de los restantes reportes si el usuario logueado desea realizar un filtrado por determinadas provincias o municipios correspondientes a una provincia específica.

Registro de Unidades de Salud (RUS)

Este módulo, de forma similar a RU, posee la información correspondiente a las unidades de salud necesaria fundamentalmente para la obtención de los Reportes de EDO por Lugar, EDO por Lugar y Grupos de Edades y EDO por Lugar y Semanas Estadísticas a nivel municipal, aunque a este mismo nivel de acceso puede dicha información ser necesaria para la obtención de los restantes reportes si el usuario logueado desea realizar un filtrado por determinadas unidades de salud correspondientes a un municipio específico.

Registro de Áreas de Salud (RAS)

Al igual que RU y RUS el presente subsistema posee la información correspondiente a los GBT y EBS necesaria fundamentalmente para la obtención de los Reportes de EDO por Lugar, EDO por Lugar y Grupos de Edades y EDO por Lugar y Semanas Estadísticas a nivel de unidad de salud, aunque a este mismo nivel de acceso puede dicha información ser necesaria para la obtención de los restantes reportes si el usuario logueado desea realizar un filtrado por determinados GBT y/o EBS correspondientes a una unidad de salud específica.

Registro de Ciudadano (RC)

Este es otro componente no incluido dentro del dominio de Salud Pública, que posee información vital para la obtención de las Estadísticas Descriptivas del REDO, especialmente de los

Reportes de EDO por Grupos de Edades y EDO por Lugar y Grupos de Edades para casos de EDO notificadas mediante Tarjeta, aunque puede resultar necesaria además para la obtención de los restantes reportes si el usuario logueado desea realizar un filtrado por sexo o determinadas edades y/o grupos de edades siempre que se trate de EDO notificadas mediante Tarjeta.

4.2 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se encuentran datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos. Para representar los diagramas del Modelo de Implementación se puede emplear el diagrama de UML de Componentes.

Diagrama de Componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos (archivos, cabeceras, módulos, paquetes, etc.) y muestra las dependencias entre estos componentes.

Estos son utilizados para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. En ellos se sitúan librerías, tablas archivos, ejecutables y documentos que forman parte del sistema. Uno de los usos principales es que puede servir para ver que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

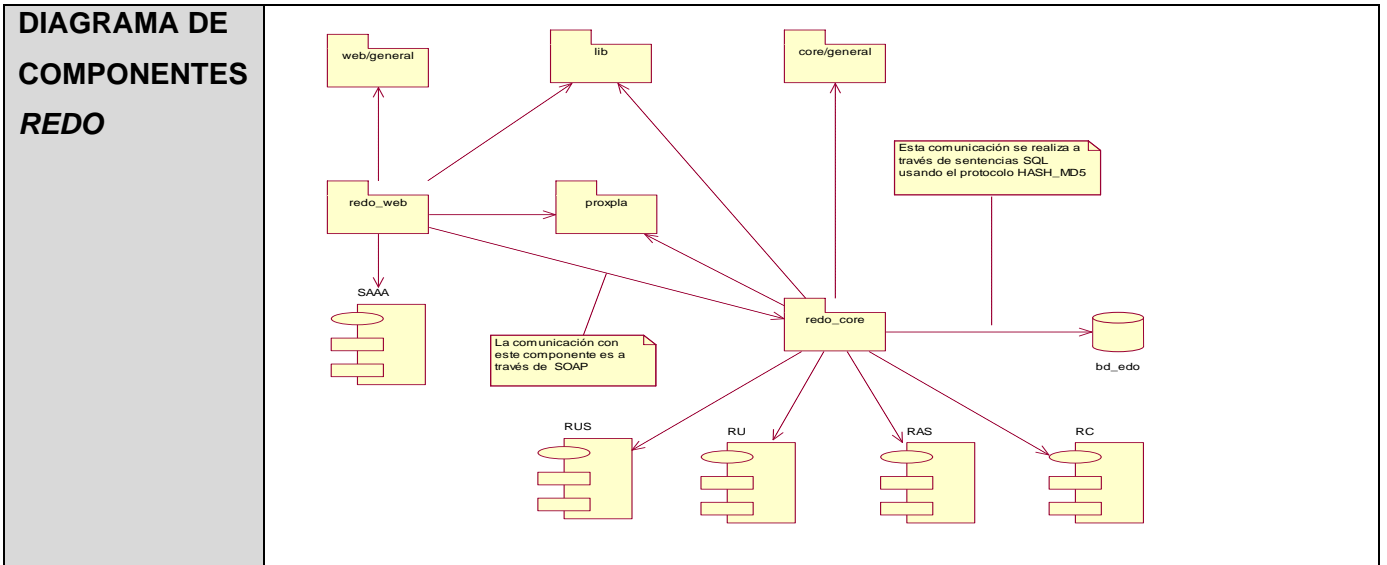


Figura 4.1 Diagrama de Componentes REDO

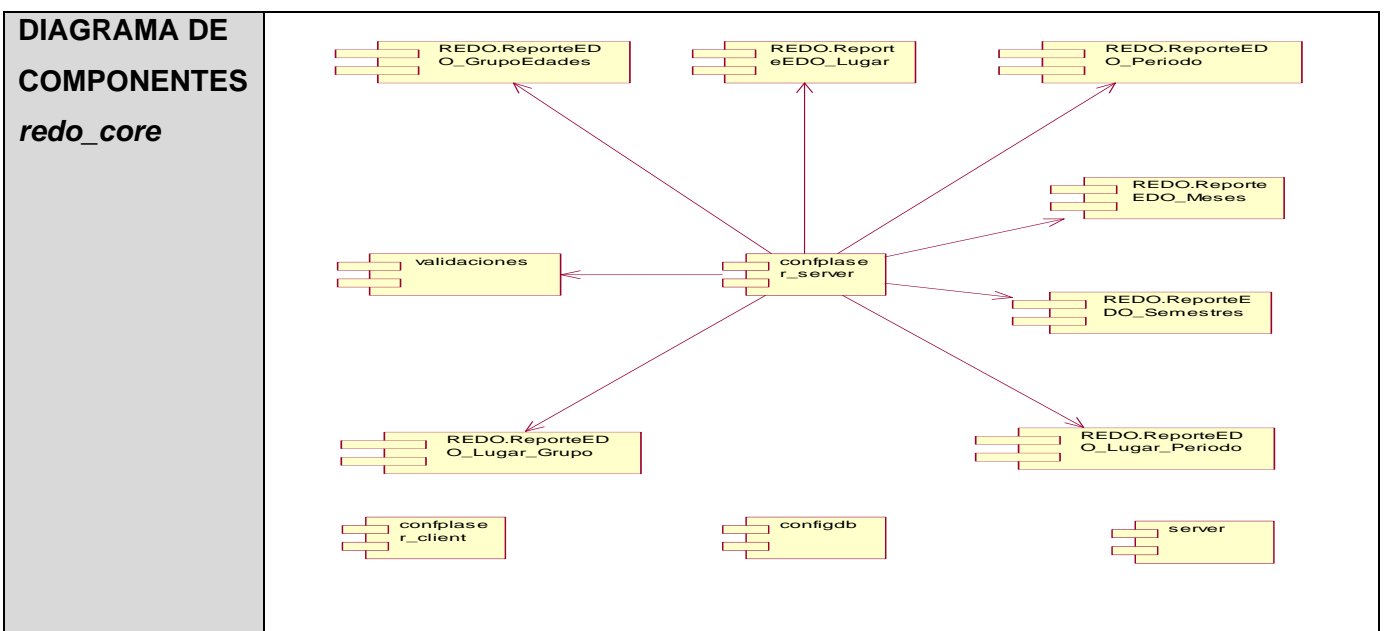


Figura 4.2 Diagrama de Componentes redo_core

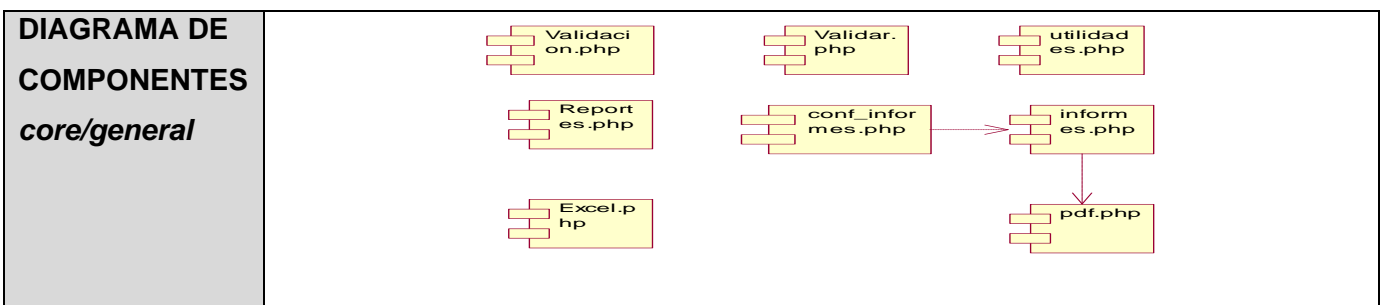


Figura 4.3 Diagrama de Componentes core/general

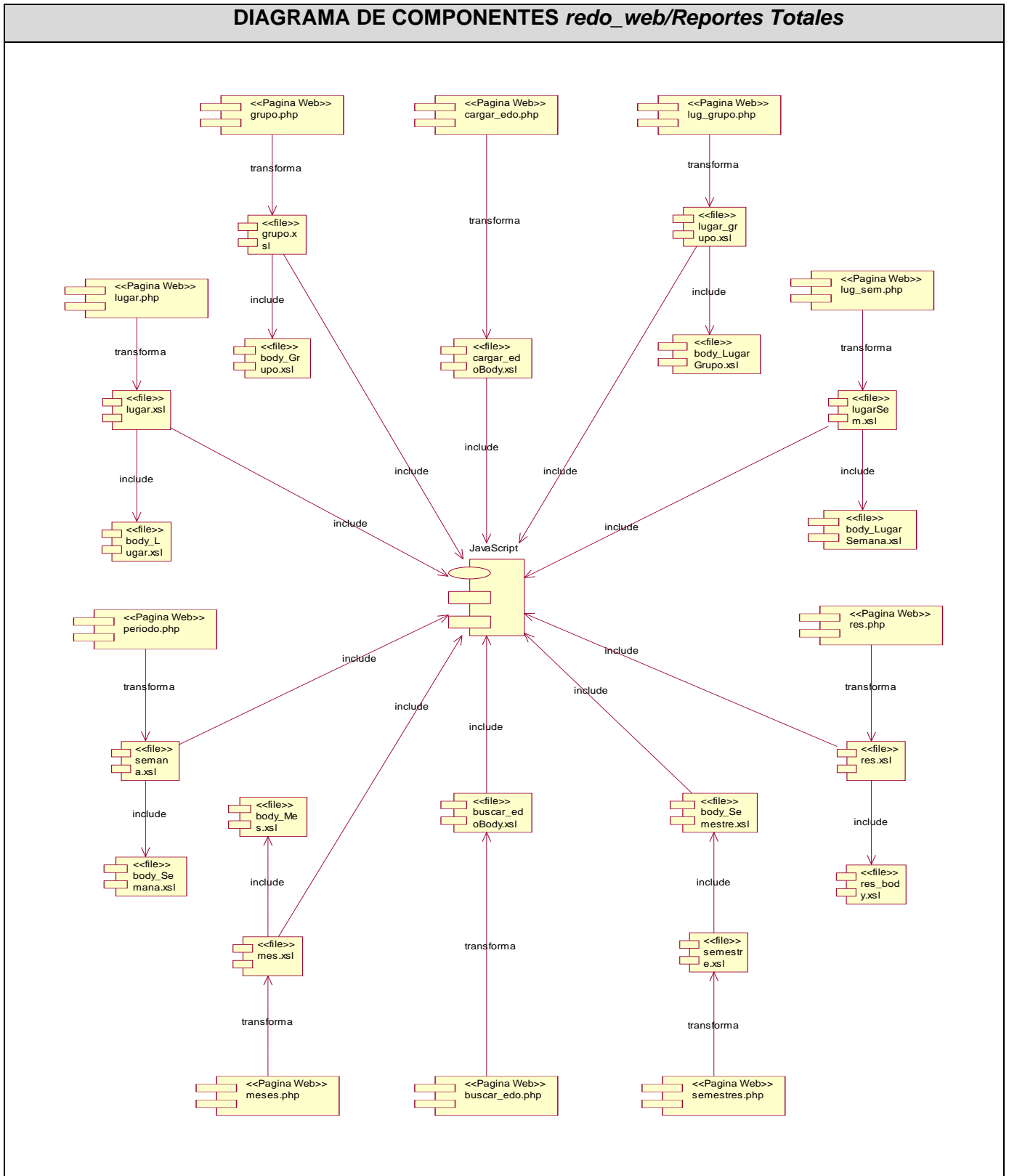


Figura 4.4 Diagrama de Componentes *redo_web/Reportes Totales*

Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes *software* (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes *software* que representan manifestaciones del código en tiempo de ejecución (los componentes utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes). Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos, procesos, etc. Un nodo será una unidad de computación de algún tipo. Las instancias de componentes *software* pueden estar unidas por relaciones de dependencia.

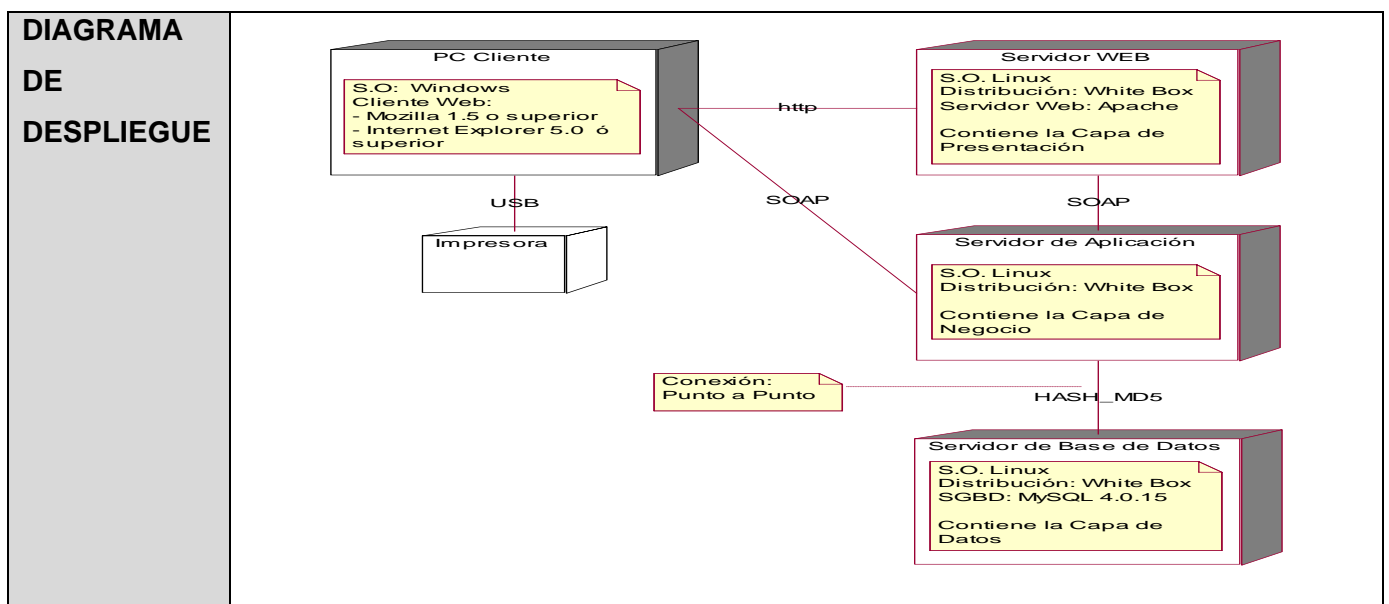


Figura 4.5 Diagrama de Despliegue

En el capítulo se ha desarrollado el proceso de implementación de los Reportes Estadísticos del REDO, obteniéndose un producto con las funcionalidades requeridas por el cliente. Este sistema tiene un correcto funcionamiento. Se ha descrito además la solución dada para la obtención de las Estadísticas Descriptivas de las EDO permitiendo la búsqueda dinámica de información, se brinda de esta manera a los usuarios del sistema una solución funcional de fácil interacción con nuevas opciones para la visualización de la información. Se han desglosado también los componentes que conforman el producto y las relaciones que establecen unos con otros para posibilitar el mantenimiento e incluso la incorporación de nuevas opciones para la obtención de datos estadísticos en versiones posteriores.

CONCLUSIONES

Con la automatización del proceso de obtención de los Reportes Estadísticos de REDO se obtuvo un producto que cumple con los requerimientos solicitados por el MINSAP, dando cumplimiento a los objetivos y tareas planteadas previamente, lo que permite arribar a las siguientes conclusiones:

- ❖ Se han asimilado la arquitectura definida por el MINSAP para sus aplicaciones, SOA-CBA, y la metodología de desarrollo RUP que la empresa SOFTEL utiliza para la construcción de productos software, lo que permitió la implementación de las Estadísticas Descriptivas del REDO según las políticas de desarrollo concebidas por el cliente.
- ❖ Han sido generados todos los artefactos que la metodología de desarrollo RUP establece deben ser obtenidos durante la construcción de un producto software permitiendo que el mismo una vez finalizado se encuentre correctamente documentado, lo que posibilita un fácil mantenimiento del producto.
- ❖ Las Estadísticas Descriptivas constituye una funcionalidad incorporada al REDO que facilita el proceso de obtención de los Reportes Estadísticos de las EDO, cuyo desarrollo ha sido concebido de modo que el usuario final solo necesite una capacitación mínima para la interacción con la aplicación.
- ❖ Los Reportes Estadísticos brindan la posibilidad de filtrar la información requerida por el usuario por diversos criterios de búsqueda de forma dinámica, siendo necesario solamente que el usuario seleccione los criterios deseados y sus atributos cualitativos.
- ❖ Se espera que la implementación de las Estadísticas Descriptivas del REDO minimice los errores propios del trabajo manual, permitiendo la obtención de una información más confiable. Al brindar la posibilidad de obtener la información estadística, eliminando la transmisión de datos de los niveles inferiores a los superiores, se fortalecerá el Sistema de Vigilancia y Control Epidemiológico. Contribuyendo así, a eliminar el desfase entre la notificación de las EDO y la toma de decisiones por parte del SNS.

RECOMENDACIONES

- ❖ Realizar un intensivo mantenimiento a las funcionalidades del REDO que posibilitan la captación de datos lo cual permitiría realizar mayor número de pruebas a la obtención de las Estadísticas Descriptivas y eliminaría el trabajo directo con la base de datos.
- ❖ Concebir los cambios necesarios en la clase PDF que permitan imprimir correctamente los Reportes Estadísticos que contienen paginado horizontal (Anexo IV), el cual funcionalmente posee una filosofía nueva que la impresión de los Reportes Estadísticos pero requiere de una clase PDF más robusta.
- ❖ Incorporar nuevas facilidades para la obtención de las Estadísticas Descriptivas que permitan la búsqueda de información no solo correspondiente a determinadas Enfermedades sino además a Grupos de Enfermedades, así como el filtrado por CIE específicas y por otros centros que notifican EDO como centros escolares, laborales, etc.
- ❖ Brindar la posibilidad al usuario de obtener Reportes Estadísticos cuyas salidas pueda ser configurables, dándole un carácter más dinámico puesto que el usuario no solo podría seleccionar los criterios de búsqueda sino el tipo de reporte que desea obtener.

REFERENCIAS BIBLIOGRÁFICAS

- [1]. Martín Andrés, Antonio; Luna del Castillo, Juan de Dios. Bioestadística para las ciencias de la salud. Madrid, Ediciones Norma-Capitel, 2004. 672
- [2]. López Serrano, Dra. Elena. Desarrollo histórico de las estadísticas sanitarias en Cuba, 1975. [Disponible en: http://bvs.sld.cu/revistas/his/cua_91/his1491.htm]
- [3]. Coutin Marie, Dra. Gisele. Utilidad de la Informática para la Vigilancia de enfermedades en el tiempo, 2006 [Disponible en: http://www.cecarn.sld.cu/rcim/revista_11/articulos_pdf/utilidadinformat.pdf]
- [4]. Rico Cordeiro, Dr. Osvaldo. Tecnología y Gestión de la Información. Bases del Desarrollo del Sistema de Vigilancia de la Salud en la Argentina, 2006. [Disponible en: <http://www.informatica2007.sld.cu/Members/orico/tecnologia-y-gestion-de-la-informacion-bases-del-desarrollo-del-sistema-de-vigilancia-de-la-salud-en-la-argentina>]
- [5]. Ídem a Referencia 3.
- [6]. Suárez Medina, Dr. Ramón. Software para el sistema integrado de vigilancia de dengue, 2005. [Disponible en: <http://www.ipk.sld.cu/memorias/dengue2005/conf/alvarez.pdf>]
- [7]. Colectivo de Autores. El modelo web vs cliente / servidor en los Sistemas Corporativos, 2002. [Disponible en: http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=152]
- [8]. Maldonado Segura, José Alberto. Tecnologías de la información al servicio de la historia clínica electrónica. España, Sociedad Española de Informática para la Salud. 2002. p 154.
- [9]. Parra, José David. Hacia una arquitectura empresarial basada en componentes, 2005. [Disponible en: <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art143.asp>]
- [10]. Ídem a Referencia 9.
- [11]. Larman, Craig. Patrones Grasp, 2005. [Disponible en: <http://jorgesaavedra.wordpress.com.>]

[12]. Ídem a Referencia 11.

[13]. Ídem a Referencia 11.

[14]. Pérez Valdés, Damián. Los Frameworks de PHP agilizan tu trabajo, 2007. [Disponible en: <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>]

[15]. Ayala Montero, Ramón. XML Iniciación y Referencia. Madrid, McGRAW-HILL/Interamericana de España, 2001. p 9,10.

[16]. Minnick, Chris; Valentine, Chelsea. XHTML Serie Práctica, 2000.

[17]. Gallego Vázquez, José Antonio. Desarrollo Web con PHP y MySQL. Madrid, Ediciones Anaya Multimedia, 2003. p 20.

[18]. Pulido Morera, Ing. Maydelys; Badaló González, Ing. Roberto. Análisis del Registro de Población de la Atención Primaria del Sistema de Información para la Salud. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2007. 103

[19]. Visual Paradigm. Visual Paradigm for UML Enterprise Edition Model-Code-Deploy Platform, 2008 [Disponible en: <http://www.visual-paradigm.com/product/vpuml/enterpriseedition.jsp>]

[20]. IBM. Rational Rose Enterprise, 2008 [Disponible en: <http://www-306.ibm.com/software/awdtools/developer/rose/enterprise/index.html>]

[21]. Zend. Zend Studio V5.5 Update and Support Information, 2008. [Disponible en: <http://www.zend.com/products/studio/studio55>]

[22]. Softonic. El mejor editor HTML visual para PC, 2007. [Disponible en: <http://adobe-dreamweaver.softonic.com/>]

[23]. EMS. EMS SQL Management Studio for MySQL, 2008. [Disponible en: <http://www.sqlmanager.net/en/products/studio/mysql>]

[24]. Bedford Mass. CALIFICAN A STYLUS STUDIO ENTRE LAS HERRAMIENTAS Y SERVICIOS DE WEB Y XML “EXCEPCIONALES” DEL MERCADO, MIENTRAS QUE XSLT MAPPER Y SQL/XML CONNECTIVITY RECIBEN UNA ALTA CALIFICACIÓN DE ACUERDO A UN REPORTE LÍDER DE LA INDUSTRIA, 2004 [Disponible en: <http://www.progresssoftware.com.mx/newsletter/nl0404.htm>]

[25]. ALTOVA. XMLSpy - XML editor for modeling, editing, transforming, & debugging XML technologies, 2008 [Disponible en: http://www.altova.com/products/xmlspy/xml_editor.html]

BIBLIOGRAFÍA

ÁLVAREZ, M. A. Lenguaje de Marcas: HTML, 2006a. Disponible en:

<http://www.webestilo.com/html/cap1a.phtml>

ÁLVAREZ, M. A. Lenguajes del lado del cliente, 2006b. Disponible en:

<http://www.desarrolloweb.com>

ALTOVA. XMLSpy - XML editor for modeling, editing, transforming, & debugging XML technologies, 2008 Disponible en: http://www.altova.com/products/xmlspy/xml_editor.html

ARMAS, Y. D. El desarrollo de la Atención Primaria de Salud en Cuba, 2006.

AUTORES, COLECTIVO DE. Desarrollo Orientado a Objetos con UML, 2004.

AUTORES, COLECTIVO DE. El modelo web vs cliente / servidor en los Sistemas Corporativos, 2002.

Disponible en: http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=152

AUTORES, COLECTIVO DE. Fortalecimiento de la Informática en el Policlínico, 2006a. Disponible en:

http://aps.sld.cu/bvs/materiales/carpeta/fortalecimiento_info.pdf

AUTORES, COLECTIVO DE. Informatización en el Sistema Nacional de Salud De Cuba, 2005.

Disponible en:

www.informatica2007.sld.cu/Member/anag/informatizacion-en-el-sistema-nacional-de-salud-de-cuba.doc

AYALA MONTERO, RAMÓN. XML Iniciación y Referencia. Madrid, McGRAW-HILL/Interamericana de España, 2001. p 9,10.

BARCO, A. SOA y los Servicios Web (I), 2006. Disponible en:

<http://arquitecturaorientadaaservicios.blogspot.com>

BEDFORD MASS. CALIFICAN A STYLUS STUDIO ENTRE LAS HERRAMIENTAS Y SERVICIOS DE WEB Y XML “EXCEPCIONALES” DEL MERCADO, MIENTRAS QUE XSLT MAPPER Y SQL/XML CONNECTIVITY RECIBEN UNA ALTA CALIFICACIÓN DE ACUERDO A UN REPORTE LÍDER DE LA INDUSTRIA, 2004 Disponible en: <http://www.progresssoftware.com.mx/newsletter/nl0404.htm>

CABRERA HERNÁNDEZ, M. et. at Propuesta de Esquema del Sistema de Información para la Salud (SISalud), 2006.

COUTIN MARIE, DRA. GISELE. Utilidad de la Informática para la Vigilancia de enfermedades en el tiempo, 2006 Disponible en: http://www.cecam.sld.cu/rcim/revista_11/articulos_pdf/utilidadinformat.pdf

CUENCA, C. L. Servidor Web Apache 2005. Disponible en:
<http://www.desarrolloweb.com>

DELGADO RAMOS, A. C., MIRNA; JUNCAL, VIRGINIA Registro Informatizado de Salud (RIS) Revista de Temas Estadísticos de Salud 2005; 1(1), 2005a.

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

DELGADO RAMOS, A. Informática en la salud pública cubana, 2005b. Disponible en:
http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm#cargo

EMS. EMS SQL Management Studio for MySQL, 2008. Disponible en:
<http://www.sqlmanager.net/en/products/studio/mysql>

FERNÁNDEZ GUTIÉRREZ, F. Las nuevas tecnologías de la información y las comunicaciones en salud Revista Cubana Educación Médica Superior v.16 n.2 Ciudad de La Habana abr.-un. 2002, 2002.

FRANCO NAVARRO, J. A. UML en acción. Modelando Aplicaciones Web, 2005.

GALLEGO VÁZQUEZ, JOSÉ ANTONIO. Desarrollo Web con PHP y MySQL. Madrid, Ediciones Anaya Multimedia, 2003. p 20.

HURTADO JARA, O. Sistemas Distribuidos, 2006. Disponible en:

<http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>

IBM. Rational Rose Enterprise, 2008 Disponible en: [http://www-](http://www-306.ibm.com/software/awdtools/developer/rose/enterprise/index.html)

[306.ibm.com/software/awdtools/developer/rose/enterprise/index.html](http://www-306.ibm.com/software/awdtools/developer/rose/enterprise/index.html)

JACOBSON, I. BOOCH., G; RUMBAUGH, J. El Proceso Unificado de Desarrollo de Software. 2000. p. Addison-Wesley.

LARGMAN, C. UML y patrones, 2004.

LARMAN, CRAIG. Patrones Grasp, 2005. Disponible en: <http://jorgesaavedra.wordpress.com>.

LÓPEZ SERRANO, DRA. ELENA. Desarrollo histórico de las estadísticas sanitarias en Cuba, 1975.

Disponible en: http://bvs.sld.cu/revistas/his/cua_91/his1491.htm

LUKE WELLING, L. T. Desarrollo web con PHP y MySQL. s.l. ANAYA Multimedia, 2003.

MALDONADO SEGURA, JOSÉ ALBERTO. Tecnologías de la información al servicio de la historia clínica electrónica. España, Sociedad Española de Informática para la Salud. 2002. p 154.

MARÍN DÍAZ, M. E. Capacitación en el área de la Informática en Salud 2006a. Disponible en:

http://www.cecarn.sld.cu/pages/rcim/revista_10/articulos_pdf/capacitacion.pdf

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

http://www.ucmh.sld.cu/rhab/editorial_rev10.htm

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

MARTÍN ANDRÉS, ANTONIO; LUNA DEL CASTILLO, JUAN DE DIOS. Bioestadística para las ciencias de la salud. Madrid, Ediciones Norma-Capitel, 2004. 672

MINNICK, CHRIS; VALENTINE, CHELSEA. XHTML Serie Práctica, 2000.

MINSAP Políticas, Estrategias y Programas, 2004. Disponible en:

http://www.sld.cu/sistema_de_salud/estrategias.html

MINSAP Vigilancia de Salud 2005. Disponible en:

<http://www.infomed.sld.cu/vigilancia/documento.html#Automatización>

MINSAP Acciones por niveles de atención, 2003. Disponible en:

<http://aps.sld.cu/bvs/materiales/programa/ecv/orga-func.pdf>

PARRA, JOSÉ DAVID. Hacia una arquitectura empresarial basada en componentes, 2005. Disponible en: <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art143.asp>

PÉREZ VALDÉS, DAMIÁN. Los Frameworks de PHP agilizan tu trabajo, 2007. Disponible en:

<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>

PRESSMAN, R. Ingeniería de Software, un enfoque practico, parte 1. La Habana, Editorial Félix Varela, 2004a. p.

PRESSMAN, R. Ingeniería de Software, un enfoque practico, parte 2. La Habana, Editorial Félix Varela, 2004b. p.

PULIDO MORERA, ING. MAYDELYS; BADALÓ GONZÁLEZ, ING. ROBERTO. Análisis del Registro de Población de la Atención Primaria del Sistema de Información para la Salud. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2007. 103

RICO CORDEIRO, DR. OSVALDO. Tecnología y Gestión de la Información. Bases del Desarrollo del Sistema de Vigilancia de la Salud en la Argentina, 2006. Disponible en:

<http://www.informatica2007.sld.cu/Members/orico/tecnologia-y-gestion-de-la-informacion-bases-del-desarrollo-del-sistema-de-vigilancia-de-la-salud-en-la-argentina>

SOFTEL Documento sobre una Arquitectura de Software para los componentes a emplear por el Sistema de Información para la Salud, 2006.

SOFTONIC. El mejor editor HTML visual para PC, 2007. Disponible en: <http://adobe-dreamweaver.softonic.com/>

STUSSER BELTRANENA, R. La informatización de la atención primaria de salud, 2006b. Disponible en: http://bvs.sld.cu/revistas/mgi/vol22_4_06/mgi12406.htm

SUÁREZ MEDINA, DR. RAMÓN. Software para el sistema integrado de vigilancia de dengue, 2005. Disponible en: <http://www.ipk.sld.cu/memorias/dengue2005/conf/alvarez.pdf>

VIDAL LEDO, M., DE ARMAS, Y Estrategias de informatización del Sector de la Salud (I) Revista Informatic@Médica. Año 3, No. 11. Diciembre 2002, 2002.

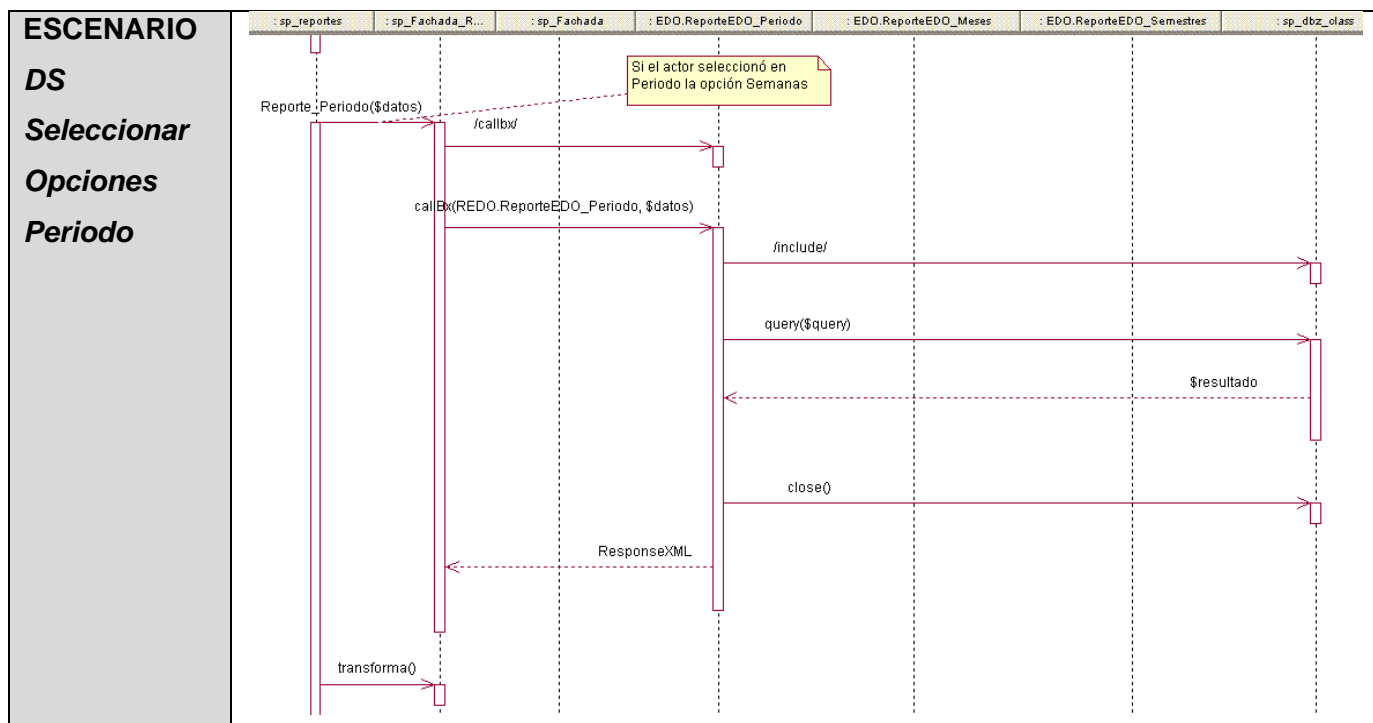
VIDAL LEDO, M., DE ARMAS, Y Estrategias de informatización del Sector de la Salud (II) Revista Informatic@Médica. Año 4, No. 12. Marzo 2003, 2003.

VIDAL LEDO, M. Información, informática y estadísticas de salud: un perfil de la tecnología de la salud, 2004. Disponible en: http://www.bvs.sld.cu/revistas/aci/vol12_4_04/aci08404.htm

VISUAL PARADIGM. Visual Paradigm for UML Enterprise Edition Model-Code-Deploy Platform, 2008 Disponible en: <http://www.visual-paradigm.com/product/vpuml/enterpriseedition.jsp>

ZEND. Zend Studio V5.5 Update and Support Information, 2008. Disponible en: <http://www.zend.com/products/studio/studio55>

Anexo III. Diagrama de Secuencia del Diseño Seleccionar Opciones Periodo



Anexo IV. Propuesta de Encabezado (Header) para salida de Reportes con Paginados Vertical y Horizontal y Listado Plegable de Criterios de Búsqueda

INTERFAZ DE SALIDA <i>Reporte EDO por Lugar</i>						
<input checked="" type="checkbox"/> Reportes - EDO por Lugar						
(-) Criterios de Filtrado						
Por Fecha: 2008-01-01, 2008-07-02 Tipo de Enfermedad: Por Consolidado Tipo de Notificación: Por Unidad a la que Perteneces Tipo de Fecha: Por Notificación						
Exportar a ▼ Página Vertical # 1 ▼ <input type="button" value="Siguiente"/> Página Horizontal # 1 ▼ <input type="button" value="Siguiente"/>						
Enfermedades	Provincias					Total
	Pinar del Río	La Habana	Ciudad de La Habana	Isla de la Juventud	Matanzas	
TB Pulmunar por Directo	0	0	0	0	0	0
TB Pulmunar por cultivo	0	0	0	0	0	0
TB Pulmunar no confirmada bacteriológicamente	0	0	0	0	0	0
TB Pulmunar por Necro o Certif. de Defunción	0	0	0	0	0	0
Otras TB aparato respiratorio	0	0	16	0	0	16
Total	0	0	16	0	0	16
Exportar a ▼ Página Vertical # 1 ▼ <input type="button" value="Siguiente"/> Página Horizontal # 1 ▼ <input type="button" value="Siguiente"/>						
<input type="button" value="Aceptar"/>						

GLOSARIO DE TÉRMINOS

- ❖ **Casos de Uso arquitectónicamente significativos:** Casos de uso que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para los usuarios del sistema y aquellos que ayudan a cubrir todas las funcionalidades significativas. Caso de uso que, dada las funcionalidades que agrupa, constituye pieza clave para la arquitectura del sistema. Esto puede estar dado por la dependencia que exista de otros casos de uso con él, que por su complejidad pueda constituir un riesgo para el sistema o por la cantidad de clases que puedan generarse del mismo.
- ❖ **CIE:** Clasificación Estadística Internacional de Enfermedades y Problemas Relacionados con la Salud.
- ❖ **EBS:** Equipo básico 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.
- ❖ **Estadísticas Descriptivas:** La estadística es una ciencia matemática que se refiere a la recolección, estudio e interpretación de los datos obtenidos en un estudio. Es aplicable a una amplia variedad de disciplinas, desde la física hasta las ciencias sociales, ciencias de la salud como la Psicología y la Medicina, y usada en la toma de decisiones en áreas de negocios e instituciones gubernamentales. La Estadística se divide en dos ramas:
 - **La estadística descriptiva**, que se dedica a los métodos de recolección, descripción, visualización y resumen de datos originados a partir de los fenómenos en estudio. Los datos pueden ser resumidos numéricamente o gráficamente. Ejemplos básicos de descriptores numéricos son la media y la desviación estándar. Resúmenes gráficos incluyen varios tipos de figuras y gráficos.
 - **La inferencia estadística**, que se dedica a la generación de los modelos, inferencias y predicciones asociadas a los fenómenos en cuestión teniendo en cuenta lo aleatorio e incertidumbre en las observaciones. Se usa para modelar patrones en los datos y extraer inferencias acerca de la población de estudio. Estas inferencias pueden tomar la forma de respuestas a preguntas sí/no (prueba de hipótesis), estimaciones de características numéricas (estimación), pronósticos de futuras observaciones, descripciones de asociación (correlación) o modelamiento de relaciones entre variables (análisis de regresión). Otras técnicas de modelamiento incluyen ANOVA, series de tiempo y minería de datos.

Ambas ramas (descriptiva e inferencial) comprenden la estadística aplicada. Hay también una disciplina llamada estadística matemática, la cual se refiere a las bases teóricas de la materia. La palabra estadísticas también se refiere al resultado de aplicar un algoritmo estadístico a un conjunto de datos, como en estadísticas económicas, estadísticas criminales, etc.

- ❖ **GBT:** Grupo Básico de Trabajo. Equipo de trabajo multidisciplinario integrado por un grupo de EBS (entre 15 y 20), por especialistas de Medicina Interna, Gineco-obstetricia, Pediatría, licenciado en psicología y de MGI, todos en función de interconsulta y de profesores, por una enfermera supervisora, un técnico de higiene y epidemiología y un técnico en Trabajo Social. El grupo cumple funciones asistenciales, docentes y gerenciales dirigidas a incrementar la calidad de la atención a la salud de la población.
- ❖ **Grupos etéreos:** Es una agrupación que se realiza por edades, estas son generalmente de 5 años, aunque esta agrupación depende del investigador o de los indicadores que se buscan.
- ❖ **Infomed:** Red telemática del MINSAP.
- ❖ **MINSAP:** Ministerio de Salud Pública, órgano rector del Sistema Nacional de Salud.
- ❖ **Navegador web (web browser):** Un navegador web (web browser en inglés), es una aplicación software que permite al usuario recuperar y visualizar desde servidores web remotos documentos de hipertexto, más conocidos por páginas web, escritas comúnmente en lenguaje HTML. Los navegadores actuales permiten mostrar y/o ejecutar: gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces. Ejemplos de navegadores son el Mozilla, el Internet Explorer, etc.
- ❖ **OMS:** Organización Mundial de la Salud.
- ❖ **Prototipo:** Maqueta visual funcional o no de la futura aplicación. Este puede ser una imagen o una aplicación software que simule funcionalidades del software.
- ❖ **Reporte:** Informe que se emite o presenta con base en la realización de una actividad o tarea.
- ❖ **Usuario Autenticado:** Es aquel usuario que ha proporcionado información mediante la cual el mecanismo de seguridad garantiza su identificación al intentar acceder a los componentes del sistema, los mecanismos de autenticación pueden ser tan simples como una contraseña o tan complejos como un dispositivo analizador de patrones retíales.