INSTITUTO SUPERIOR POLITÉCTICO “JOSÉ ANTONIO ECHEVERRÍA”

UNIVERSIDAD DE HOLGUÍN “OSCAR LUCERO MOYA”

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

REGISTRO DE ACTIVIDADES DIARIAS DEL EQUIPO BÁSICO DE SALUD PARA EL SISTEMA INFORMATIZADO DE ATENCIÓN PRIMARIA

TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA

AUTOR: WILLIAM SÓÑORA CRUZ

TUTOR: Ing. MIRNA CABRERA HERNÁNDEZ

LA HABANA, JUNIO 2005

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos los únicos autores de la presente investigación y autorizamos a la Empresa SOFTEL, perteneciente al Ministerio de Informática y las Comunicaciones (MIC) a que hagan el uso que estimen pertinente con la misma.

Para que así conste, firmamos la presente a los 22 días del mes de Junio del 2005.

Firma del Autor
Diplomante
Williams Sóñora Cruz

Firma del Tutor
Ingeniera
Mirna Cabrera Hernández

Dedicatoria y Agradecimientos

"A nadie y en especial a todos, pero sobre todo a mí, que fuí, el que lo hizo todo"...

"La flexibilidad invita a la exploración,
y la exploración se recompensa con
el descubrimiento".

Bill Gates

Resumen

El Registro de Actividades Diarias del Equipo Básico de Salud(EBS) es un módulo o componente del Sistema Informatizado de Atención Primaria que viene desarrollando la empresa SOFTEL del Ministerio de la Informática y las Comunicaciones (MIC).

Este componente informatizará los procesos primarios de los servicios médicos que tienen lugar en el nivel más básico y fundamental del Sistema Nacional de Salud, específicamente informatizará los procesos de captura, almacenamiento, transporte y transformación de la información que se genera en las consultas del EBS; labor que hoy se realiza de forma manual y de forma semiautomática en escasos momentos del proceso, por lo que ocurren demoras en la entrega de información y errores en el procesamiento e interpretación de datos.

La realización de este proyecto se integra a las acciones que se realizan en el programa de Informatización de la sociedad cubana y en especial en el Programa General de Informatización del Sistema Nacional de Salud (SNS). Tiene como objetivo diseñar un componente para el Sistema Informatizado de Atención Primaria que gestione toda la información relativa al Registro de Actividades Diarias del EBS, y que sea construido por la empresa SOFTEL en la infraestructura productiva creada para tal efecto en la Universidad de Ciencias Informáticas (UCI) y ha permitido el uso de tecnologías modernas y estandarizadas mundialmente para el desarrollo y producción de software en nuestro país.

Índice

Introducción----1

Capítulo 1 Fundamentación Teórica----5

Introducción----5

1.1 Sistema Nacional de Salud en Cuba----5

1.1.1 Informatización del Sistema Nacional de Salud en Cuba en los inicios del siglo XXI----6

1.1.2 Informatización de la Atención Primaria de Salud en Cuba en la segunda mitad de la primera década del siglo XXI.----8

1.1.3 Registro Informatizado de Salud (RIS)----9

1.1.4 Solución integral propuesta para la Informatización del SNS en la etapa actual.----9

1.2 Objeto de Estudio----10

1.3 Objetivos General y Específicos----11

1.4 Conclusiones del Capítulo...----11

Capítulo 2 Tendencias y Tecnologías Actuales----12

Introducción----12

2.1 El Software Libre en la Industria Informática----13

2.2 Internet y las Aplicaciones Web----17

2.2.1 La información a través de Internet. La Web----18

2.2.2 Aplicaciones Web----18

2.2.2.1 Tipos de Aplicación Web----19

2.3 Tendencias y Tecnologías Actuales en la Ingeniería Web----19

2.3.1 Arquitecturas de Desarrollo----20

2.3.1.1 Arquitecturas de capas----20

2.3.1.1(a) Middleware----22

2.3.1.1(b) Sistemas Clientes-Servidor Distribuidos----22

2.3.1.2 Arquitecturas basadas en Componentes----24

2.3.1.3 Arquitecturas orientadas a Servicios----25

2.3.1.3(a) XML en los Negocios----26

2.3.1.4 Arquitecturas basadas en Eventos----31

2.3.2 Lenguajes de Programación----32

2.3.2.1 Perl----32

2.3.2.2 ASP----33

2.3.2.3 PHP----33

2.3.2.4 SMARTY----34

2.3.2.5 JSP----34
2.3.2.6 Ruby----34
2.3.3 Gestores de Base de Datos----35
2.3.3.1 MySQL----35
2.3.3.2 PostgreSQL----35
2.4 Metodologías de desarrollo para Aplicaciones Web----36
2.4.1 Proceso Unificado+UWE----37
2.4.2 Metodologías Ágiles (Programación Extrema)----38
2.4.3 Metodología hipermedial de diseño orientada a objetos----40
2.5 Tecnologías a utilizar para la construcción del sistema----40
2.6 Conclusiones del Capítulo...----42
Capítulo 3 Modelado del Negocio----43
Introducción----43
3.1 Descripción del proceso de negocio actual----43
3.2 Modelación del Negocio----46
3.2.1 Proceso de Gestión del Registro de Actividades Diarias del Equipo Básico de Salud ----49
3.2.2 Proceso de Elaboración de Información Estadística a partir de los datos del RAD----52
3.2.3 Proceso de Transportación de los RAD al Departamento de Estadística del Área de Salud----53
3.2.4 Modelo de Objetos del Negocio----55
3.3 Descripción de los Procesos de Negocio Propuestos----55
3.4 Reglas del Negocio----57
3.5 Requerimientos del Sistema----58
3.5.1 Requisitos Funcionales----59
3.5.2 Requisitos No Funcionales----61
3.5.2.1 Requisitos de Apariencia o Interfaz externa----61
3.5.2.2 Requisitos de Usabilidad----61
3.5.2.3 Requisitos de Portabilidad----61
3.5.2.4 Requisitos de Seguridad----62
3.5.2.5 Requisitos de Confiabilidad----62
3.5.2.6 Escalabilidad----62
3.5.2.7 Requisitos de Interfaz interna----63
3.5.2.8 Requisitos de Software----63
3.5.2.9 Requisitos de Hardware----63
3.5.2.10 Requisitos de Diseño e Implementación ----63
3.6 Conclusiones del Capítulo...----64

Capítulo 4 Análisis y Diseño del Sistema----65
4.1 Análisis del Sistema----65
4.1.1 Actores del Sistema----65
4.1.2 Casos de Uso del Sistema. Descripción. Especificación. Prototipo de Interfaz de Usuario----67
4.2 Mapa de Navegación----80
4.3 Diseño del Sistema----81
4.3.1 Paquetes de organización. Relación entre paquetes----81
4.3.1.1 Paquete # 1 “Registro de Actividades Diarias”----82
4.3.1.2 Paquete # 2 “Reportes”----83
4.3.2 Diseño de la Base de Datos----84
4.3.3 Principios de Diseño----86
4.3.3.1 Estándares en la Interfaz de la aplicación----86
4.3.3.2 Concepción General de la Ayuda----90
4.3.3.3 Tratamiento de Excepciones----91
4.3.4 Estándares de Codificación----93
4.3.5 Diagragama de componentes del Módulo de Registro de Actividades Diarias del Equipo Básico de Salud----95
4.3.5.1 Plataforma de Servicio ”PlaSer”----98
4.3.8 Modelo de Despliegue----99
4.4 Conclusiones del Capítulo... ----101
Capítulo 5 Estudio de Factibilidad----102
Introducción----102
5.1 Resultados del cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo----104
5.2 Beneficios tangibles e intangibles----105
5.3 Análisis de costos y beneficios----105
5.4 Conclusiones del Capítulo...----106
Conclusiones Generales----107
Recomendaciones----108
Bibliografía----109

Introducción

Desde finales del siglo XX la Informática y la producción de software ocupan un lugar predominante a nivel mundial. La sociedad ha experimentado un desarrollo tecnológico acelerado en las últimas décadas en todas las esferas, con el surgimiento de las microcomputadoras y en especial de la Internet.

Cuba enfrenta el reto de informatizar la sociedad con vistas a integrarse plenamente a la infraestructura global de la información, así como de hacer un uso óptimo de las nuevas tecnologías. Como parte de la Batalla de Ideas y del programa de informatización de la sociedad, el Ministerio de Salud Pública desarrolla múltiples tareas orientadas por la dirección del país y uno de los programas que más expectativas y posibilidades abre en el campo del conocimiento, la información científica y la asistencia médica está relacionado con la Informatización del Sistema Nacional de Salud (SNS). [1]

El Ministerio de Salud Pública de Cuba (MINSAP), ha definido la informatización del sector de la salud como el proceso, cuyos procedimientos se enmarcan en el concepto de la informatización social, en busca de la optimización de los servicios de salud que se brindan a la población; mayor productividad y competencia en el desempeño de sus profesionales y técnicos, control en la administración de sus recursos, eficacia y eficiencia en su gerencia.

Con este proyecto se persigue implementar un Programa General de I nformatización del Sistema Nacional de Salud teniendo como centro del mismo al Policlínico, que apoye las estrategias y políticas trazadas por la dirección del país y el MINSAP; ;de manera que se logre la incorporación progresiva y sistemática de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) en función de la adquisición y gestión del conocimiento y los servicios de salud, logrando que las instituciones del país alcancen un elevado nivel de informatización de las actividades que brindan, partiendo del Sistema de Atención Primaria y tomando como eje al policlínico, permitiendo un incremento de la calidad, efectividad y eficiencia de los servicios que se presten a la población, contribuyendo al logro de la satisfacción de los usuarios del Sistema Nacional de Salud.

El Sistema Informatizado de Atención Primaria (SIAP), surge orientado a la informatización de la sociedad, como parte del Sistema Integral de Salud (SIS) y como una herramienta de ayuda en la toma de decisiones a partir de toda la información que se recopila en sus registros y se encuentra disponible en la red pública de transmisión de datos.

La creación y desarrollo de la Red Telemática de Salud en Cuba, INFOMED, aporta la infraestructura necesaria que comunicará a todas las instituciones y centros de salud del país a través de una densa red de computadoras posibilitando las consultas e intercambio científico, así como aportando la comunicación necesaria para la interacción de las aplicaciones que forman parte de la solución que se espera.

El Sistema Informatizado de Atención Primaria de Salud (SIAP), es un sistema

que gestiona información de salud que debe ser compartida con otros sistemas, se trata de la base que debe sustentar la estandarización de la información más utilizada por sistemas específicos, garantizando la gestión centralizada de dicha información en los distintos niveles: nacional, provincial, municipal y unidades de salud.

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

Como parte de los esfuerzos por acelerar el diseño y puesta en funcionamiento de una solución que permita dar cumplimiento a los objetivos trazados por el SNS, el MINSAP ha establecido estrategias de trabajo en conjunto con el Ministerio de Informática y las Comunicaciones (MIC), siendo la Empresa SOFTEL, con la participación de estudiantes de 5to año de la carrera de Ingeniería Informática provenientes de diversas universidades del país y de estudiantes de 3er año de la facultad 7 de la Universidad de las Ciencias Informáticas (UCI), la ejecutora directa de dicha solución, empresa que toma la misión de producir todo el software necesario que permita el uso eficiente de los medios y la tecnología que se está instalando en las instituciones de salud.

La intensa, sistemática y profesional labor que en los últimos meses se ha desarrollado por el grupo de trabajo del Proyecto APS, ha permitido contar con el diseño de nuevos módulos que se incorporan al Programa General de Informatización del Sistema Nacional de Salud en la medida en que sean implementados. Entre ellos podemos mencionar: Registro de Áreas de Salud, Registro de Medios de Diagnóstico, Registro del Clasificador Internacional de Enfermedades y Problemas de Salud (CIE), Registro de Enfermedades de Declaración Obligatoria, Registro de Población, Registro de Partos y Nacimientos, Registro de Fallecidos, Registro de Autopsias y el que ocupa la atención del presente trabajo, el Registro de Actividades Diarias del Equipo Básico de Salud.

Actualmente todo este proceso de registro y elaboración de informes estadísticos se realiza de forma manual o semiautomática en algunos casos, enrareciendo algunos datos, demorando su ejecución y por consiguiente la toma de medidas para el control, la prevención o la mejora de la calidad de vida de los pacientes. La forma de recolección de datos actual aleja al personal de salud de labores investigativas, así como impide que el personal administrativo pueda auditar la labor de sus trabajadores eficazmente debido al gran número de papeles que deben manipularse constantemente. Además intervienen varias personas para completar cada Registro de Actividades Diarias lo que hace mucho más engorroso el proceso y demora mucho más tiempo la culminación de la tarea y la elaboración de información.

☞ se tiene entonces un proceso de recopilación de datos de las consultas y de elaboración de informes basados en la Hoja de Cargo, poco productivo, de baja calidad y de eficiencia bastante cuestionable. ☜

☞ Por lo que cómo mejorar el proceso de recopilación y emisión de información referente al Registro de Actividades Diarias (Hoja de Cargo) del Equipo Básico de Salud es el problema que se tratará de resolver con la realización de este proyecto. ☜

☞se define como objeto de estudio en el desarrollo de esta investigación el Proceso de gestión y control de las Actividades Diarias del Equipo Básico de Salud en el Sistema Nacional de Salud.☜

☞El campo de acción se enmarca en la "informatización del proceso de recopilación, transformación y emisión de información basada en el Registro de Actividades Diarias (Hoja de Cargo) del Equipo Básico de Salud".☜

☞El objetivo general es modelar un sistema informático para la gestión automatizada del Registro de Actividades Diarias (RAD) del Equipo Básico de Salud, que se inserte como un módulo más del Sistema Informatizado de Atención Primaria.☜

El diseño de un sistema informático para la gestión automatizada del Registro de Actividades Diarias (RAD) del Equipo Básico de Salud, modelado bajo estándares de la industria (UML), que refleje y contemple todos los artefactos y modelos propios de un proceso ingenieril para la producción de un software, que represente un producto con características cliente-servidor, distribuido, con arquitectura de tres capas o multicapas, orientado a la conexión y basado en componentes, desarrollado para plataforma web; favorecerá a la construcción del módulo que se integrará al Sistema Informatizado de Atención Primaria, a la creación de equipos de trabajo orientados a la producción de software en la Universidad de las Ciencias Informáticas y a la formación de conductas y hábitos de trabajo profesionales acorde con las prácticas de los líderes de la industria.

Para cumplir el objetivo general se realizaron las siguientes tareas:

- Estudio del proceso de registro de la consulta a un paciente en el modelo oficial para el Registro de Actividades de Medicina Familiar.

- Estudio de la visión del sistema general y ubicación del módulo diseñado dentro del mismo.

- Selección de las tecnologías, acorde a las tendencias actuales, a emplear para dar solución al problema.

- Análisis riguroso de los requerimientos para el software.

- Estudio profundo de las metodologías de desarrollo de software basadas en UML y con estereotipos para el desarrollo de aplicaciones web dinámicas que incluyan el uso de las tecnologías de los Servicios Web XML.

- Se realizó un estudio de los estilos de arquitectura de desarrollo para escoger el idóneo para la aplicación.

El presente documento está compuesto por cinco capítulos, que incluyen todo lo relacionado con el trabajo investigativo realizado, la modelación del negocio, así como el análisis y el diseño de la aplicación que se propone.

En el Capítulo 1: Fundamentación Teórica. Se explican los principales conceptos del Sistema Nacional de Salud y del nivel de atención primario de este. Explica la necesidad del Registro de las Actividades Diarias del Equipo Básico de Salud y los

procesos que serán objeto de automatización y los objetivos del trabajo.

En el Capítulo 2: Tendencias y Tecnologías Actuales. Se describen las tendencias y tecnologías actuales a tener en cuenta para implementar la aplicación. Una breve explicación relacionada con los motivos que permitieron definir la variante seleccionada. Se hace referencia a las herramientas utilizadas para realizar el análisis, diseño y las que serán utilizadas para la implementación.

En el Capítulo 3: Modelación del Negocio: Se aborda lo referente al funcionamiento del negocio, abordando sus reglas, los casos de uso y las descripciones de casos de uso del negocio y las mejoras que se propone al mismo. Se describe además la solución propuesta, utilizando los requerimientos funcionales y no funcionales.

En el Capítulo 4: Análisis y Diseño del Sistema. Se hace referencia a todo los procesos de análisis y diseño del sistema y los artefactos propios de estas etapas en el desarrollo de un producto informático, se describe la forma en que se realizará la implementación a través del diagrama de clases del diseño, el diagrama de clases persistentes y el modelo de datos. Se hace referencia a los principios de diseño y al modelo de implementación mediante el diagrama de despliegue y de componentes.

En el Capítulo 5: Estudio de Factibilidad. Se describe el estudio de factibilidad realizado a la aplicación, así como un análisis de los costos-beneficios que esta trae consigo.

Esta tesis brinda el análisis y diseño del módulo de gestión del Registro de Actividades Diarias del Equipo Básico de Salud y se complementa con toda la documentación del mismo, lo cual permitirá en etapas posteriores la construcción del mismo.

CAPÍTULO 1
Fundamentación Teórica

INTRODUCCIÓN

En este capítulo se brindará una panorámica sobre el Sistema Nacional de Salud (SNS) y la Atención Primaria (APS) , conoceremos un poco sobre cuales son sus precedentes y su comportamiento en la sociedad cubana actual.

Se comenzará a sentar las bases teóricas para en los siguientes capítulos comenzar a modelar todos los artefactos necesarios para desarrollar un buen proceso de ingeniería, tal y como se pretende realizar con el desarrollo de este folleto, muestra documental del trabajo realizado como Analistas de Sistemas en este proyecto de desarrollo y producción elaborado en conjunto con la empresa cubana de software SOFTEL.

1.1 SISTEMA NACIONAL DE SALUD EN CUBA

La garantía de atención médica gratuita a toda la población cubana se convirtió desde los primeros momentos del triunfo de la Revolución en uno de los paradigmas sociales fundamentales, en correspondencia con la esencia humanista y de justicia social que caracteriza a nuestro proceso revolucionario. [2]

Desde el propio triunfo revolucionario se adoptaron medidas para transformar la Salud Pública en Cuba, una de las principales y más novedosa fue la creación del Sistema Nacional de Salud (SNS), designándose al Ministerio de Salud Pública como su organismo rector.

Esta estructura organizativa comenzó a realizar importantes reformas a partir de los años 60, como parte fundamental de las transformaciones del período revolucionario y en respuesta al respeto más absoluto de uno de los derechos humanos fundamentales de todo ciudadano. Surge el servicio de hospitales rurales llevando la atención médica a zonas apartadas de la geografía nacional, se dan los primeros pasos para el fortalecimiento de la atención primaria; surgen los policlínicos integrales como una unidad asistencial creada para brindar servicios y resolver los principales problemas existentes en los primeros años de la revolución.

En la década del 70, por los cambios en el cuadro de morbilidad - mortalidad, los servicios prestados en los policlínicos integrales cobran nuevas funciones, cambiando la estructura de los mismos, pasando a una atención médica general, surgiendo así el policlínico comunitario donde prestaban atención los médicos generales.

En la década del 80 surge el Programa del Médico y la Enfermera de la Familia, sentando precedentes en la salud pública internacional por su carácter novedoso y futurista, especialmente con la implantación y desarrollo del modelo de Atención de

Medicina Familiar a partir de 1984.[3]

En 1996, el SNS adoptó desde el punto de vista organizativo, estrategias fundamentales y priorizó cuatro programas básicos para continuar perfeccionándose: el Programa de Atención Materno Infantil (PAMI), de Control de Enfermedades Transmisibles y Crónicas no Transmisibles, y el de Atención al Adulto Mayor, todos los que han sido monitorizados, controlados y evaluados de acuerdo a la metodología establecida.

El Programa del Médico y la Enfermera de la Familia, se ratifica como el eje del actual desarrollo estratégico, orientándose el resto de las estrategias en función del mismo. Este modelo de atención es la mayor fortaleza y potencialidad que tiene el SNS. Por su existencia, filosofía, bases teóricas y lo que ha podido proporcionarle al sistema de salud se ha logrado mantener los indicadores de salud y satisfacer las necesidades de la población, constituyendo un pilar básico de la Salud Pública cubana. [4]

Con más de 20 años de experiencia en este programa se comienzan a experimentar cambios para la atención primaria, de esta forma, servicios que antes eran exclusivos de hospitales son abiertos en instituciones de la atención primaria; surgiendo así hace aproximadamente 2 años el novedoso modelo de policlínico con nuevas funciones, acercando los servicios a la población, para hacer realidad las palabras de nuestro Comandante en Jefe: “... una profunda revolución en los servicios de salud tendrá lugar en nuestra Patria...” [5]

1.1.1 Informatización del Sistema Nacional de Salud en Cuba en los inicios del siglo XXI

La informatización del SNS tiene como objetivo acercar eficientemente y con calidad la prestación de los servicios de salud a la población, por lo que se pretende implementar un Programa General de Informatización del SNS, que apoye las estrategias y políticas trazadas por la dirección del país y del MINSAP; de manera que se logre la incorporación progresiva y sistemática de las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC) en función de la adquisición y gestión del conocimiento y los servicios de salud.

Se quiere que las instituciones del país alcancen un elevado nivel de informatización de las actividades que brindan, partiendo del Sistema de Atención Primaria y tomando como eje al policlínico, de manera que se logre un incremento de la calidad, efectividad y eficiencia de los servicios que se presten a la población, contribuyendo al logro de la satisfacción de los usuarios del Sistema Nacional de Salud.

Como solución integral significa la articulación de un nuevo paradigma en la prestación de servicios de salud, regido por el principio básico de lograr acercar cada vez más los servicios de salud a la población. Entre los principales impactos esperados con la Informatización del SNS podemos mencionar:

Para la población:

- Equidad distribuida de acceso a servicios, tecnologías e información de salud independientemente de áreas geográficas, ni niveles de atención.

- Disfrutará la sensación de ser atendida por un personal médico mejor preparado y actualizado, elevando su confianza hacia el sistema de atención.

- Reducción del número de desplazamientos innecesarios entre instituciones de salud con el consecuente impacto en su vida social.

- Reducción de tiempos de esperas para el acceso a servicios especializados con la posibilidad de recibirlos en su propio escenario social.

Para el SNS:

- Gestión oportuna de una información confiable y actualizada que propiciará una optimización considerable de recursos.

- Elevación de la capacidad y calidad de la toma de decisiones asistenciales y gerenciales por la disposición oportuna de información actualizada para todos los niveles del SNS, que permitirá una rápida transferencia de la información sanitaria de un paciente.

- Disponer de un soporte y herramientas poderosos para la formación y actualización constante de sus miembros desde sus propios escenarios de desempeño, potenciando la investigación científica multicéntrica, nacional e internacional.

- Elevará el papel del Médico y Enfermera de la Familia, incrementando su nivel científico y profesional.

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

SOFTEL, Empresa del Ministerio de la Informática y las Comunicaciones (MIC), tiene la misión de generar las soluciones informáticas e implementar un sistema de excelencia para el desarrollo y mantenimiento de productos de software especializados en salud y además organizar un esquema para la prestación de los servicios informáticos a dicho sector.

En la actualidad, utiliza una estrategia nunca antes concebida en el país en un proceso de desarrollo de software, con una organización del proceso productivo a través de una eficiente gestión de requerimientos, donde participan desde un inicio, médicos y trabajadores de la salud, vinculados directamente a la Atención Primaria en calidad de expertos funcionales y en estrecho vínculo con los especialistas de Informática.

A través del proceso de desarrollo unificado (RUP) y haciendo uso del Lenguaje Unificado de Modelado (UML) se describen los procesos que se proponen para automatizar.

SOFTEL ejecuta estos objetivos en colaboración con la Universidad de las Ciencias Informáticas (UCI), para lograr la vinculación a la producción desde los primeros años de estudio de los estudiantes y los profesores líderes de proyectos y la formación en un segundo perfil en temas relacionados con la salud.

La experiencia de este trabajo en SOFTEL debe constituir el inicio de buenas prácticas en la producción de software con alta calidad y un ejemplo de normativa para los proyectos que deben irse abriendo de ahora en adelante en la Informatización del Sector de la Salud. [6]

El Sistema de Salud Cubano, posee en el nivel de Atención Primaria una plataforma ideal para articular los avances de las nuevas tecnologías de la información en función de hacer más eficiente todo el aparato estratégico y administrativo que rodea al propio sistema.

1.1.2 Informatización de la Atención Primaria de Salud en Cuba en la segunda mitad de la primera década del siglo XXI.

El Proceso de Informatización de la Atención Primaria de Salud (APS), es un proyecto priorizado para el SNS, cuyo objetivo fundamental consiste en la creación del Sistema Informatizado de Atención Primaria que permita la gestión médica, interacción con los consultorios del Médico de la Familia, obtención de estadísticas y apoyo en la logística de los nuevos servicios.[6]

En el marco del Programa de Informatización de la Sociedad Cubana, el Proyecto APS en su concepción general se propone abordar el análisis, diseño y desarrollo de un producto de software, siguiendo las buenas prácticas internacionales y las normativas del MINSAP, logrando que facilite la gestión de la información en la Atención Primaria, acorde a los cambios y necesidades de este sector, permitiendo el flujo de información hacia los diferentes niveles de toma de decisiones.

En esta nueva etapa de fortalecimiento del Sistema Nacional de Salud, la Atención Primaria de Salud (APS) es el eje fundamental de estas transformaciones, teniendo como objetivo fundamental convertir a los Policlínicos en centros de atención primaria de salud de la más alta calidad, cada vez más accesibles a la población, consolidando el Sistema Municipal de Salud, para dar cumplimiento al principio de la descentralización de las soluciones según los problemas de salud de la comunidad.

La automatización de la gestión de la APS debe comenzar por utilizar las tecnologías que permitan modelar la gestión de la información en este nivel para almacenar, procesar, recuperar y comunicar información clínica y administrativa, relativa a todas las actividades de los policlínicos y unidades de la atención primaria. Debe tener la capacidad de comunicación y de integración de toda la información,

independientemente de donde se haya generado y que sirva para el aprendizaje basado en experiencias compartidas entre los profesionales en el país y fuera de nuestras fronteras, así como para lograr la integración con los procesos de los otros niveles de atención.

1.1.3 Registro Informatizado de Salud (RIS)

La Informatización de la Salud Cubana no ofrece un mecanismo único de integración de los sistemas de información desarrollados, estos en la actualidad se presentan como componentes aislados, lo cual trae consigo la duplicación de información y la consiguiente falta de integridad de la misma.

El Registro Informatizado de Salud (RIS), sentó las bases para la existencia de un sistema formado por componentes, desarrollados con un nivel de cohesión y acoplamiento que le permiten ser capaces de interactuar entre ellos y de esta forma reutilizar la información gestionada por cada componente.[7]

El Proyecto APS vinculado a la informatización de la salud en Cuba encamina su tarea a analizar, diseñar y desarrollar un producto de software, único en su tipo, que hereda las características del RIS, pero que se caracteriza por ser un sistema distribuido de componentes distantes geográficamente, en constante interacción a través de la Red Telemática de Salud de Cuba, INFOMED, para dar respuesta a los procedimientos establecidos por el SNS para este nivel de atención.

Por tanto, es necesario desarrollar una arquitectura que garantice la máxima disponibilidad de cada uno de sus componentes, que permita la recuperación del sistema ante posibles fallos de conectividad o resolver problemas tales como la recuperación de la información, independientemente de su ubicación.

El RIS se basa en una arquitectura orientada a servicios, desarrollado con tecnología XML Web Services e implementado con PHP y MySQL. Desde el año 2003 forman parte del RIS los siguientes componentes: Registro de Unidades de Salud, Registro de Profesionales de la Salud, Registro de Ubicación, Registro del Ciudadano y Registro de Equipos de Salud.[7]

1.1.4 Solución integral propuesta para la Informatización del SNS en la etapa actual

Para lograr la Informatización en este sector se pretende que todos los módulos estén incluidos en un conjunto de aplicaciones que formarán parte del Sistema Integral de Salud (SISalud), compuesto a su vez por el Registro Informatizado de Salud (RIS), el Sistema Informatizado de Atención Primaria (SIAP) y el Sistema Informatizado de Gestión Hospitalaria (SIGH).

Registro Informatizado de Salud (RIS): Está formado por los registros que son administrados o gestionados a nivel nacional o central y que integran el Registro No Médico Informatizado de Salud (RNMIS) y por los registros que pueden ser accesados

desde cualquier nivel de atención o institución de salud para lograr la continuidad en el seguimiento del paciente, agrupándose éstos en el Registro Médico Informatizado de Salud (RMIS).

Registro No Médico Informatizado de Salud (RNMIS): En esta nueva etapa de análisis, diseño y desarrollo se incorporarán al RNMIS: el Registro de Áreas de Salud, Registro de Medios de Diagnóstico, Registro del Clasificador Internacional de Enfermedades y Problemas de Salud (CIE) y los codificadores propios de APS que se gestionan también a nivel central y definen diferentes aspectos que son utilizados localmente: Registro de Conductas e Indicadores y Registro de Dispensarización. De igual forma se ubicarán en el RNMIS todos los registros que en la actualidad pertenecen al RIS, mencionados en el epígrafe anterior.

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

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

En una segunda etapa continuarán incorporándose al SIAP los próximos módulos que se definan, según las prioridades del usuario.

Sistema Informatizado de Gestión Hospitalaria (SIGH): Se agruparán aquí los módulos que pertenecen al nivel de atención secundario u hospitalario y que serán definidos para próximos desarrollos. En esta etapa comenzará a formar parte del mismo el Registro de Autopsias, diseñado en la etapa actual bajo el Proyecto APS por la integración que tiene con el resto de los módulos que se desarrollan.

1.2 OBJETO DE ESTUDIO

El objeto de estudio en que se centra este documento es el Proceso de gestión y control de las Actividades Diarias del Equipo Básico de Salud en el Sistema Nacional de Salud.

El Registro de Actividades Diarias (RAD), Hoja de Cargo del Equipo Básico de salud (EBS) como también se le conoce o Modelo 18-145 de Actividades de Medicina Familiar es el registro diario que lleva el médico y la enfermera de la comunidad sobre las actividades de medicina familiar que realizan en su consultorio o en el área de la comunidad a la que pertenecen; es el registro primario que se le hace al paciente

que viene a recibir atención médica por cualquier motivo y que es utilizado para el control del trabajo de los médicos, así como para la elaboración de estadísticas acerca las poblaciones, grupos de salud y de riesgo, prevención de enfermedades y demás informes que deben y pueden generarse para la administración a distintos niveles y para la investigación general de las poblaciones o grupos poblacionales.

Actualmente todo este proceso de registro y elaboración de informes estadísticos se realiza de forma manual o semiautomática en algunos casos, enrareciendo algunos datos, demorando su ejecución y por consiguiente la toma de medidas para el control, la prevención o la mejora de la calidad de vida de los pacientes. La forma de recolección de datos actual aleja al personal de salud de labores investigativas, así como impide que el personal administrativo pueda auditar la labor de sus trabajadores eficazmente debido al gran número de papeles que deben manipularse constantemente. Además intervienen varias personas para completar cada RAD lo que hace mucho más engorroso el proceso y demora mucho más tiempo la culminación de la tarea y la elaboración de información.

1.3 OBJETIVOS GENERALES Y ESPECÍFICOS

El objetivo general es modelar un sistema informático para la gestión automatizada del Registro de Actividades Diarias(RAD) del Equipo Básico de Salud, que se inserte como un módulo más del Sistema de Integral de Atención Primaria.

A partir de un análisis del objetivo general se derivan los siguientes objetivos específicos:

- Realizar un estudio detallado del proceso de gestión de la información de los pacientes que son atendidos por los Equipos Básicos de Salud.

- Realizar un estudio de las tendencias y tecnologías, para escoger la más factible opción para dar solución al problema.

- Diseñar un sistema que se integre a todos los artefactos desarrollados para el Sistema Integral de Salud y en particular al Sistema Integral de la Atención Primaria de Salud.

- Diseñar un sistema que permita registrar la información de los resultados de la consulta de los pacientes por los EBS y que a través de los reportes brinde información elaborada a cualquier nivel del Sistema Nacional de Salud.

1.4 Conclusiones del capítulo...

Este capítulo ha servido de marco teórico para sentar las bases de la investigación que se desarrolló y como preámbulo del proceso ingenieril. Además se han optenido conceptos y definiciones útiles para enfrentar la investigación y comenzar el estudio preliminar que se realizó en la fase de Concepción de este proyecto.

CAPÍTULO 2
Tendencias y Tecnologías Actuales

INTRODUCCIÓN

Hablar del mundo de la Informática de hoy y de las tendencias actuales del mismo, puede ser algo complicado, quizás, demasiado arriesgado.

Es seguro que lo que se escriba en este documento hoy, como un intento de recopilar la información más actual, mañana, sea actualizado y queden, estas palabras, como simple historia, de lo que un día fue. Se intentará entonces hacer un bosquejo acerca de las tendencias más novedosas, más utilizadas y a la vez, económicamente más convenientes, que en el área de la Ingeniería Web son temas de discusión, de desarrollo, de investigación e implantación.

Claro, no se abarcará todo el espectro de herramientas, formas, procesos, metodologías, lenguajes de programación y demás cosas que sobre la Ingeniería Web existen, nos limitaremos en este capítulo, a realizar un barrido de la información existente en los siguientes temas:

- El Software Libre en la Industria Informática.
- Internet y las Aplicaciones Web.
- Tendencias y Tecnologías Actuales en la Ingeniería Web.
  - Arquitecturas de desarrollo.
  - Lenguajes de programación.
  - Gestores de Base de Datos.
- Metodologías para el desarrollo de Aplicaciones Web.
- Tecnologías a utilizar para la construcción de nuestro sistema.

Vale aclarar que no se detallará al máximo cada tema; reflejar aquí todas las aristas que sobre los temas antes mencionados existen, ocuparían todos los esfuerzos de este trabajo y se perderá la esencia del proceso ingenieril desarrollado para encontrar una solución a una situación problémica en un área determinada; es por ello que solo se presentan los elementos más importantes que guardan una relación estrecha con los objetivos presentados en el capítulo 1 para este trabajo, de esta forma se podría cumplir con las tareas proyectadas referentes a esta etapa de la investigación, convencer al lector o al menos, hacerlo entender, que la solución que se propone será una variante correcta, útil, mutuamente beneficiosa para todos los involucrados en este proyecto y sobre todo, acorde con las Tendencias y Tecnologías Actuales.

2.1 EL SOFTWARE LIBRE EN LA INDUSTRIA INFORMÁTICA.

El software libre conlleva toda una serie de ventajas sobre el software propietario por los derechos que otorga a sus usuarios. Algunas de estas ventajas pueden ser más apreciadas por los usuarios particulares, otras por las empresas, y otras por las administraciones públicas. Desgraciadamente el software libre ha sido objeto de desinformaciones y mitos, algunos provocados deliberadamente, que han intentado distorsionar su credibilidad. Se mencionan las principales ventajas con el objetivo de mostrar que el uso de estas tecnologías son una gran oportunidad para los desarrolladores de la industria del software y en especial para la implementación de este proyecto. [12]

Para la mayoría de usuarios individuales el software libre es una opción atractiva por las libertades que garantiza sin necesidad de verse lastrados por el precio. Sin embargo, en el caso de empresas y la Administración Pública, el coste del software es un factor importante y a veces determinante en la elección de nuevos sistemas informáticos.

Cuando se analiza el precio de una solución tecnológica se suele hablar del TCO (Total Cost of Ownership), es decir, del coste total de la propiedad que tiene una determinada solución de software. Este concepto fue inventado por el Gartner Group en 1987 como herramienta de análisis exhaustiva de los costes de una solución de mercado y desde entonces se ha convertido en un estándar. En este análisis se reflejan el coste del programa, la ayuda, y el mantenimiento tecnológico de la solución.[13]

Se parte de la base que el software libre no tiene prácticamente coste de licencia y por lo tanto que esta parte del presupuesto se puede invertir para mejores fines como mejorar la adaptación de los programas y la formación en esta tecnología.

Según un estudio de la consultora “Robert Frances Group” publicado en el año 2002 el coste total de propiedad del sistema operativo libre Linux era menos de la mitad que el de Windows. En el estudio se analiza el coste de diferentes servidores durante un período de tres años y se constata que gran parte del ahorro proviene de no tener que pagar licencia y de sus menores costes de administración. En el mismo sentido se expresa un estudio realizado por la consultora “Consulting Times”, en este caso, sobre el coste de propiedad de sistemas de correo; también concluye que las soluciones basadas en software libre son mucho más económicas en todos los casos planteados.[12]

Sin embargo, a parte de los menores costes, también deben considerarse otros aspectos positivos del software libre que no quedan reflejados en los análisis TCO, como la independencia del proveedor y la posibilidad de una adaptación completa.

Por último, destacar que existen bastantes análisis de TCO que se decantan claramente hacia el fabricante que patrocinó la ejecución del análisis. Es imprescindible ser cauto con este tipo de análisis y buscar segundas fuentes que nos puedan ayudar a contrastar la información facilitada.

Por otra parte, el modelo del software libre, donde prima el hecho de compartir la información y el trabajo cooperativo, es bastante similar al que tradicionalmente se ha usado en el mundo académico y científico. En estos ámbitos, los resultados de las

investigaciones se publican y se divulgan en publicaciones científicas, y sirven de base para nuevas investigaciones. Éste es principalmente el modelo sobre el que la humanidad ha innovado y avanzado.

En el mundo del software propietario, las licencias de software, las patentes de software y otras herramientas legales y técnicas se utilizan para impedir que terceros participen en ese conocimiento y para que éste continúe siendo patrimonio exclusivo de la empresa que lo creó. La innovación pertenece a una empresa, mientras que en el mundo del software libre, de forma muy similar al dominio público, el conocimiento pertenece a la humanidad.

En el software libre los usuarios tienen un destacado papel dado que influyen decisivamente en la dirección hacía donde evolucionan los programas: dando a conocer los errores que quieren que sean corregidos, proponiendo nueva funcionalidad al programa, o contribuyendo ellos mismos en el desarrollo.

A finales del año 2004 se publicó una lista de las innovaciones más importantes en software de es año. Se consideró como innovación número uno el navegador libre FireFox y entre los diez programas mencionados también se encontraba OpenOffice.

Además, aunque resulta imposible generalizar, existen casos bien documentados donde las soluciones de software libre tienen unos requisitos de hardware menor, y por lo tanto son más baratas de implementar y de utilizar o trabajar con ellas. Por ejemplo, los sistemas Linux que actúan de servidores pueden ser utilizados sin la interfaz gráfica con la consecuente reducción de requisitos de hardware necesarios.

También es importante destacar que en el software propietario el autor puede decidir en un momento dado no continuar el proyecto para una cierta plataforma, para un hardware que considera antiguo, o descontinuar el soporte para una versión de su software. En las aplicaciones d e software libre, estas decisiones no pueden ser tomadas por una empresa o individuo sino por toda una comunidad con diferentes intereses. Esto se traduce en un mejor soporte en general para las versiones antiguas de software y de plataformas de hardware o software más minoritarias.[14]

El modelo de desarrollo de software libre sigue un método por el que el software se escribe de forma cooperativa por programadores, en gran parte voluntarios, que trabajan coordinadamente en Internet. Lógicamente, el código fuente del programa está a la vista de todo el mundo, y son frecuentes los casos en que se reportan errores que alguien ha descubierto leyendo o trabajando con ese código.

El proceso de revisión pública al que está sometido el desarrollo del software libre imprime un gran dinamismo al proceso de corrección de errores. Los usuarios del programa de todo el mundo, gracias aque disponen del código fuente de dicho programa, pueden detectar sus posibles errores, corregirlos y contribuir a su desarrollo con sus mejoras. Son comunes los casos en que un error de seguridad en Linux se hace público y con él la solución al mismo.[15]

Con el software propietario la solución de los errores no llega hasta que el fabricante del programa puede asignar los recursos necesarios para solventar el problema y publicar la solución.

Además, uno de los grandes problemas en la industria de software es la dependencia que se crea entre el fabricante y el cliente. Este hecho se acentúa con especial gravedad cuando el fabricante no entrega el código fuente, ya que inevitablemente el cliente queda atado a él para nuevas versiones y, en general, para cualquier mejora que necesite.

El software libre garantiza una independencia con respecto al proveedor gracias a la disponibilidad del código fuente. Cualquier empresa o profesional, con los conocimientos adecuados, puede seguir ofreciendo desarrollo o servicios para nuestra aplicación. En el mundo del software propietario, sólo el desarrollador de la aplicación puede ofrecer todos los servicios.

A menudo los proveedores de software propietario se ven obligados a dejar de fabricar un producto por un cambio drástico de las condiciones del mercado, o simplemente porque consideran que ya no podrán rentabilizar la inversión. Disponiendo del código fuente, cualquier programador puede continuar su desarrollo y sus actualizaciones hasta que el cliente decida que es el momento adecuado para migrar a un nuevo sistema informático. De esta forma no se estaría obligado a las constantes actualizaciones a las que hoy se ven arrastrados todos los usuarios de sistemas basados en software propietario.

Otro aspecto a considerar es la repercusión en las industrias nacionales, regionales y demás que puede generar el uso de las tecnologías de software libres; según Sedisi (Asociación Española de Empresas de Tecnologías de la Información), en el año 2001 la industria del software en España movió 1.139,84 millones de euros, de los que 315 millones se destinaron a las ventas de sistemas operativos, que en su totalidad están desarrollados en Estados Unidos. Del resto, las herramientas de desarrollo y software de bases de datos representan 126,68 y 156,03 millones, respectivamente, de los cuales un gran número son desarrollados en los Estados Unidos. [16]

Es fácil deducir que los países importadores invierten mucho dinero en distribuir, dar apoyo y formación a productos realizados fuera de sus fronteras, balanza que podría cambiar si se dedicara parte de ese capital al desarrollo de los productos informáticos, que es realmente la parte de la industria que requiere ingenieros más calificados y que genera más valor y conocimiento en los países que hoy dominan el mercado, utilizando para ello las bondades del software no propietario para incentivar el surgimiento y desarrollo de esta industria en nuestras naciones.

En el software libre no hay coste de licencia debido al derecho a copia y, al disponer del código fuente de la aplicación, es posible desarrollar internamente las mejoras o las modificaciones necesarias, en vez de encargarlas a empresas de otros países que trabajan con sistemas propietarios. De este modo, se contribuye a la formación de profesionales en nuevas tecnologías y al desarrollo local.

Por otro lado, todas las mejoras que se realicen no tienen restricciones y se pueden compartir con cualquier otra administración, empresa, institución u organismo que las necesite. En el software propietario, estas mejoras no se pueden llevar a cabo o quedan en manos de la empresa creadora, que normalmente se reserva los derechos de uso y propiedad intelectual y establece en qué condiciones las comercializará.

El software libre, al disponer del código fuente, mejora diversos aspectos relacionados con la perennidad de los datos y su seguridad. Para empezar, los sistemas de almacenamiento y recuperación de la información del software son públicos y cualquier programador puede ver y entender cómo se almacenan los datos en un determinado formato o sistema, lo que garantiza la durabilidad de la información y su posterior migración. Nos sirve de ejemplo el caso de una base de datos de un censo electoral. El software propietario trabaja habitualmente con formatos propios, cuyos mecanismos de almacenamiento no siempre se han hecho públicos, por lo que quizás no sería posible, en caso de que se precisara migrar el sistema, recuperar el contenido de este censo.[16]

El software libre, por su carácter abierto, dificulta la introducción de código malicioso, espía o de control remoto, debido a que el código lo revisan muchos usuarios y desarrolladores que pueden detectar posibles puertas traseras. En el software propietario nunca se podrá saber si los programadores originales introdujeron a título personal, o por encargo de la empresa, puertas traseras que ponen en peligro la seguridad del sistema o la privacidad de los datos.

Algunos fabricantes de software propietario han colaborado con agencias gubernamentales para incluir accesos secretos al software para así poder visualizar datos confidenciales; de este modo, se comprometen aspectos de la seguridad nacional cuando estos sistemas se utilizan para almacenar datos críticos de gobiernos. En el mundo del software libre, cualquier programador puede realizar una auditoría para comprobar que no se ha introducido ningún código malicioso y, a su vez, cualquier entidad puede añadir libremente encriptación adicional a la aplicación que utilice para proteger sus datos.

La empresa Mitre ha elaborado un estudio por encargo del Departamento de Defensa de los Estados Unidos donde se analiza el uso de software libre y de código abierto en sistemas que se encuentran en producción en este departamento. Las conclusiones son claramente favorables a seguir incrementando el uso del software libre, y se destaca la posibilidad que ofrece de solucionar errores de seguridad de forma inmediata sin depender de un proveedor externo.[16]

Según este informe, hay más de 115 aplicaciones de software libre en uso en el Departamento de Defensa con más de 250 ejemplos de su empleo en diferentes entornos.

Otra posibilidad del software libre es la adaptación del software; el software propietario habitualmente se vende en forma de paquete estándar, que muchas veces no se adapta a las necesidades específicas de empresas y administraciones. Una gran parte de la industria del software se basa en desarrollar proyectos donde se requiere software personalizado. El software libre permite personalizar, gracias al hecho de que disponemos del código fuente, los programas tanto como sea necesario, hasta que cubran exactamente nuestra necesidad. La personalización es un área muy importante en que el software libre puede responder mucho mejor que el software de propiedad a unos costes mucho más razonables. [15]

Aproximadamente el 75% del software que se escribe en el mundo es software de uso interno para empresas, que requiere un alto grado de personalización y donde el software puede proporcionar desarrollos más económicos.

La mayoría de la infraestructura de Internet está basada en software libre y protocolos abiertos. Actualmente más del 60% de los servidores web utilizan Apache, un gran número de servidores de correo usan Sendmail para gestionar el envío de correo electrónico y prácticamente la totalidad de los servidores de nombres (DNS) esenciales en el funcionamiento de la Red, utilizan el programa BIND o derivados de su código fuente. Es indiscutible la importancia que ha tenido el software libre en la extensión y desarrollo de Internet desde sus inicios, y la influencia mutua de estos dos ámbitos tecnológicos es un hecho contrastado. Por lo tanto, el éxito del software libre va mucho más allá de la disponibilidad de una enorme cantidad de programas con licencias libres (entre los cuales el sistema operativo GNU/Linux, el navegador Mozilla o el paquete ofimático OpenOffice son ejemplos notables).

A manera de conclusión de este epígrafe es importante destacar que hoy en día hay bastantes decisiones gubernamentales que apoyan la migración de las tecnologías no libres hacia las tecnología libres. Ejemplo de ello son las manifestaciones de intención expresadas por gobiernos de países como China, Reino Unido, España, Brasil, Venezuela (a partir del 1er Foro Mundial de Tecnologías Libres celebrado en noviembre del 2004 en Venezuela), Cuba (aunque existía una estrategia aprobada desde noviembre del 2002 para la extensión de la tecnologías libres, no es hasta mayo del 2005 en la XI Convención Internacional de Informática, Habana 2005 en que se anunció la migración progresiva por los organismos centrales del Estado, hacia las tecnologías libres, lo que supone una extensión posterior a todos los niveles, instituciones y empresas del país) y otros países, principalmente de Europa y Asía. Debido a ello, a las ventajas argumentadas y luego de realizar un estudio en el mercado de las tecnología libres existentes, asociadas con las intenciones de solución con el problema planteado, junto con los desarrolladores e interesados en la ejecución de este proyecto,se tomó la decisión de desarrollar este producto informático haciendo uso de las bondades que brindan estos tipos de tecnologías y guiarnos por muchos de los principios del Software Libre.

2.2 INTERNET Y LAS APLICACIONES WEB.

Internet es el fenómeno social, cultural, sociológico y comercial más importante del pasado siglo, tiene su origen en la década del 60 y se relaciona con un proyecto de defensa financiado por el gobierno de Estados Unidos. Gracias a esta iniciativa, hoy es posible buscar, crear y transferir información en tiempo real para 6 mil millones de personas.

Desde sus inicios el crecimiento de Internet ha sido fenomenal, especialmente en la década del 90, época en que la red se convirtió en una herramienta fundamental de comunicación, información e integración, que permite a los usuarios ahorrar tiempo y dinero, además de poner a su alcance todos los productos y servicios que estos requieran, sin fronteras de espacio o de tiempo.

Para que esta maravilla pueda funcionar y dos computadoras se comuniquen vía Internet debe existir un camino físico que los una (líneas telefónicas, conmutadas, redes digitales, enlaces satelitales, microondas, fibra óptica, cable coaxial, etc.) y un mismo protocolo de comunicación entre ellos (TCP/IP).[24]

El desarrollo de Internet no sólo se ha traducido en beneficios para los usuarios, sino también para las empresas, organismos e instituciones. Dentro de este ámbito el

comercio electrónico ha tenido un crecimiento constante. Los principales productos que se transan a través de la red son: información, música, viajes, libros, hardware y software.

2.2.1 La información a través de Internet. La Web.

World Wide Web (WWW), o simplemente Web, es el universo de información accesible a través de Internet, una fuente inagotable del conocimiento humano. Es un sistema de información global, interactivo, dinámico, distribuido, gráfico, basado en Hipertexto, con plataforma de enlaces cruzados que se ejecuta en Internet.

El componente más usado en Internet es definitivamente la Web. Su característica sobresaliente es el texto remarcado, un método para referencias cruzadas instantáneas. Usando la Web, se tiene acceso a millones de páginas de información. La exploración se realiza por medio de un software especial denominado “Browser” o Explorador. La apariencia de un Sitio Web puede variar ligeramente dependiendo del explorador que use. Así mismo, las versiones más recientes disponen de una funcionalidad mucho mayor tal como animación, realidad virtual, sonido y música. El protocolo que se utiliza para la comunicación en la Web es el http (Hypertext Transfer Protocol) y el formato que se utiliza para la transferencia es el HTML (Hypertext Transfer Protocol).

2.2.2 Aplicaciones Web.

Las aplicaciones Web se desarrollan como una extensión de los Sistemas Web para agregar funcionalidad de negocio al proceso. En términos más simples, una Aplicación Web es un Sistema Web que permite a los usuarios ejecutar lógica de negocio a través de un Navegador (Browser), o lo que es lo mismo, modificar el estado del negocio o dicho de otra forma es un sistema informático donde una gran cantidad de datos volátiles, altamente estructurados, van a ser consultados, procesados y analizados mediante navegadores.

Una de las principales características debe ser su alto grado de interacción con el usuario, y el diseño de su interfaz debe ser claro, simple y debe estar estructurado de tal manera que sea orientativo para cada tipo de usuarios.

Las Aplicaciones Web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. Si no existe lógica de negocios en el servidor, el sistema no puede ser considerado una aplicación Web, en ese caso se considera como un sitio Web.

La arquitectura de un Sitio Web es simple. Contiene como componentes principales: el Servidor Web, una Red y un Navegador o cliente. La arquitectura de una aplicación Web además incluye la aplicación en el Servidor, que es la que permite al sistema manejar lógica de negocio y tener un estado. [17]

Las libertades que brindan este tipo de aplicaciones, la usabilidad, portabilidad, seguridad y su fácil mantenimiento y adaptabilidad por parte de los usuarios, así como el auge y la posición que están tomando en el mercado, hicieron que nos decidiéramos a elegir este tipo de solución para la construcción de nuestro producto informático

A la hora de establecer una clasificación la podemos realizar atendiendo a criterios

como pueden ser la complejidad de los datos, de la propia aplicación, la volatilidad, la estructuración de los datos o la intencionalidad de la aplicación. Dependiendo de la intencionalidad de la Aplicación Web se pueden clasificar en las denominaciones que a continuación se presentarán.

2.2.2.1 Tipos de Aplicación Web.

- Informacionales: Orientadas a la difusión de información personalizada o no, y con acceso a la BD o sin él.
- Orientados a la descarga de datos: Servidores de Material didáctico, servidores de canciones, …
- Interactivas: Orientadas ala interacción con el usuario.
- Orientas al Servicio: Sistemas de ayuda financiera, simuladores, …
- Transaccionales: Compra electrónica, banca electrónica,…
- De Flujo de Datos: Sistemas de planificación en línea, manejo de inventario, …
- Entornos de Trabajo Colaborativo: Herramientas de diseño colaborativo, sistemas de autoría distribuidos,…
- Comunidades on-line (Sistemas C2C): Foros de debate, servicios de Subastas,…
- Portales Web: Centros comerciales de compra electrónica, intermediarios en línea,…
- Orientados al análisis de datos: Dataworkhousing, aplicaciones OLAP,…

2.3 TENDENCIAS Y TECNOLOGÍAS ACTUALES EN LA INGENIERÍA WEB.

La implantación en la sociedad de las denominadas "nuevas tecnologías" de la información y las comunicaciones, está produciendo cambios insospechados respecto a los originados en su momento por otras tecnologías, como la imprenta, y la electrónica. Sus efectos y alcance, no sólo se sitúan en el terreno de la información y la comunicación, sino que lo sobrepasan para llegar a provocar y proponer cambios en la estructura social, económica, laboral, jurídica y política. Y ello es debido a que no sólo se centran en la captación de la información, sino también, y es lo verdaderamente significativo, a las posibilidades que tienen para manipularla, almacenarla y distribuirla.

Las tecnologías actuales para el desarrollo de aplicaciones sobre plataforma web son muy variadas y dinámicas, tanto como el mismo Internet es por ello que nos dedicaremos en este epígrafe a hacer un estudio “detallado” de las mismas.

Los sistemas y aplicaciones basados en la Web hacen posible que una población extensa de usuarios finales dispongan de una gran variedad de contenido y funcionalidad. La ingeniería Web no es un clónico perfecto de la Ingeniería de software, pero toma prestado muchos de los conceptos y principios básicos de esta, dando importancia a las mismas actividades y técnicas de gestión. Existen diferencias sutiles en la forma en que se llevan a cabo estas actividades, pero la filosofía primordial es idéntica dado que dicta un enfoque disciplinado para el desarrollo de un sistema basado en computadora. [20]

La Ingeniería Web es importante porque a medida que las aplicaciones Web se integran cada vez más en grandes y pequeñas compañías, es imprescindible la necesidad de construir sistemas fiables, utilizables y adaptables. Esta es la razón por la que es necesario un enfoque disciplinado para el desarrollo de este tipo de

aplicaciones informáticas.

La Ingeniería Web aplica un enfoque genérico que se suaviza con estrategias, tácticas y métodos especializados. El proceso de ingeniería comienza con una formulación del problema que quiere resolverse con la aplicación Web. Se planifica el proyecto y se analizan los requisitos que debe cumplir el sistema, entonces se lleva a cabo los diseños de interfaces, arquitectónico y de la navegación. El sistema se implementa utilizando lenguajes y herramientas especializados asociados con la Web y entonces comienzan las pruebas. Dado que las aplicaciones Web están en constante evolución, deben establecerse los mecanismos para el control de configuraciones, garantía de calidad y soporte continuado.

Luego de todas estas tareas obtenemos productos de ingeniería tales como el modelo de análisis, de diseño, procedimientos de prueba, entre otros y el producto final que es, la aplicación Web operativa.

Esta explicación no ha dicho nada en realidad y puede hacer parecer que la Ingeniería Web y por tanto el desarrollo de aplicaciones Web es un procedimiento rutinario y sencillo, lo cual está muy lejos de la realidad. En los siguientes epígrafes se tratará de mostrar y abundar sobre algunas cuestiones importantes sobre estos temas en los cuales serán los principios en los que se basará la aplicación que se explica en este documento.

2.3.1 Arquitecturas de desarrollo.

La arquitectura en el desarrollo de software se define como una estructura de los componentes de un programa o sistema, sus interrelaciones, y los principios y reglas que gobiernan su diseño y evolución en el tiempo. (Garlan y Perry, 1995); también es reconocida como la estructura o estructuras de un sistema, lo que incluye sus componentes de software, las propiedades observables de dichos componentes y las relaciones entre ellos. (Bass, Clements y Kazman, 1998).

En los siguientes subepígrafes se expondrán algunos ejemplos significativos de arquitecturas de desarrollo utilizados hoy en la construcción y concepción de aplicaciones Web.

2.3.1.1 Arquitecturas de capas.

Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia en el desarrollo de aplicaciones hoy en día. El estilo en capas se define como una organización jerárquica de forma tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. En algunos ejemplares, las capas internas están ocultas a todas las demás, menos para las capas externas adyacentes, y excepto para funciones puntuales de exportación; en estos sistemas, los componentes implementan máquinas virtuales en alguna de las capas de la jerarquía. En otros sistemas, las capas pueden ser sólo parcialmente opacas. En la práctica, las capas suelen ser entidades complejas, compuestas de varios paquetes o subsistemas. El uso de arquitecturas en capas, explícitas o implícitas, es muy frecuente; solamente hay cerca de cien patrones que son variantes del patrón básico de capas. [17]

Muchas más veces se sacrifica la pureza de la arquitectura en capas precisamente para mejorarla: colocando, por ejemplo, reglas de negocios en los procedimientos almacenados de las bases de datos, o articulando instrucciones de consulta en la capa de la interface del usuario.

El número mínimo de capas es obviamente dos, y en ese umbral la literatura arquitectónica sitúa a veces al sub-estilo cliente-servidor como el modelo arquetípico del estilo de capas y el que se encuentra con mayor frecuencia en las aplicaciones en red. Este modelo particular dudosamente necesite descripción, pero de todos modos aquí va: Un componente servidor, que ofrece ciertos servicios, escucha que algún otro componente requiera uno; un componente cliente solicita ese servicio al servidor a través de un conector. El servidor ejecuta el requerimiento (o lo rechaza) y devuelve una respuesta, luego abundaremos un poco más en este modelo pero en el caso especial de los sistemas cliente-servidor distribuidos. [19]

Las ventajas del estilo en capas son obvias. Primeramente, el estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. En segundo lugar, el estilo admite de forma natural optimizaciones y refinamientos y en tercer lugar, proporciona amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas.

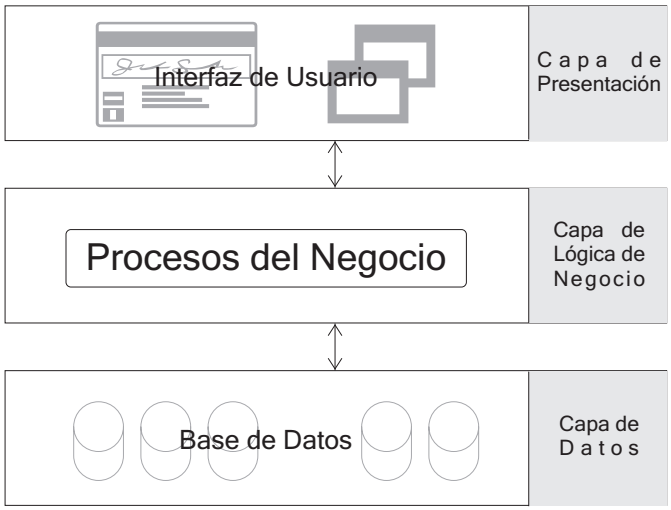


Fig.2.1 Arquitectura de 3 Capas

También se han señalado algunas desventajas de este estilo, muchos problemas no admiten un buen mapeo en una estructura jerárquica, incluso cuando un sistema se puede establecer lógicamente en capas, consideraciones de performance pueden requerir acoplamientos específicos entre capas de alto y bajo nivel. A veces es también extremadamente difícil encontrar el nivel de abstracción correcto. También se admite que la arquitectura en capas ayuda a controlar y encapsular aplicaciones complejas, pero complica no siempre razonablemente las aplicaciones simples.[21]

Resumiendo, la separación y especialización de interfaces de usuario, reglas de negocio y estructuras de datos (o sus equivalentes arquitectónicos en otros dominios), conservan todavía plena relevancia. La documentación correspondiente ha elaborado numerosos patrones de arquitectura y diseño derivados de este estilo como las arquitecturas multicapas o N-niveles como también se le conoce.

Las arquitecturas de N-niveles resultan de la conexión de varios sistemas de 3 niveles entre sí y/o añadiendo una capa adicional que permite el acceso al sistema a través de un servidor Web.

La capa Web era, inicialmente, externa al sistema (una capa real adicional); hoy en día, esta capa se va incorporando lentamente a la capa de presentación que reside en el lado del servidor (parte de la infraestructura middleware en un sistema de 3 niveles, o parte del servidor directamente en un sistema de 2-niveles).

La incorporación de la capa Web nos lleva a la noción de “servidores de aplicación” que se utiliza para referirse habitualmente a plataformas middleware que soportan acceso a través de Web.

2.3.1.1(a) Middleware.

El Middleware es una capa de software intermedio entre el cliente y el servidor. Es la capa de software que nos permiten gestionar los mecanismos de comunicaciones. Ejemplo si se hace la petición de una página web desde un browser en el cliente, el middleware determina la ubicación y envía una petición para dicha página. El servidor Web, interpreta la petición y envía la página al software intermedio, quien la dirige al navegador de la máquina cliente que la solicitó. Existen dos tipos: software intermedio general, son servicios generales que requieren todos los clientes y servidores, por ejemplo: software para las comunicaciones usando el TCP/IP, software parte del sistema operativo que, por ejemplo, almacena los archivos distribuidos, software de autenticación, el software intermedio de mensajes de clientes a servidores y viceversa; Software intermedio de servicios, software asociado a un servicio en particular, por ejemplo: software que permite a dos BD conectarse a una red cliente/servidor (ODBC: Conectividad abierta de BD), software de objetos distribuidos, por ejemplo la tecnología CORBA permite que objetos distribuidos creados en distintos lenguajes coexistan en una misma red (intercambien mensajes), software intermedio para software de grupo, software intermedio asociado a productos de seguridad específicas (Conexiones Seguras: Sockets), etc.

Las principales caraterísticas les permite independizar el servicio de su implantación, del sistema operativo y de los protocolos de comunicaciones; permiten la convivencia de distintos servicios en un mismo sistema y permite la transparencia en el sistema.

2.3.1.1(b) Sistemas Cliente-Servidor Distribuidos.

El concepto de "Sistemas Distribuidos" que se ha popularizado tanto en la actualidad y teniendo como ámbito de estudio las redes, como por ejemplo: Internet, redes de teléfonos móviles, redes corporativas, redes de empresas, etc.[31]

Los Sistemas Distribuidos son aquellos cuyos componentes hardware y software, que están en ordenadores conectados en red, se comunican y coordinan sus acciones

mediante el paso de mensajes, para el logro de un objetivo. Se establece la comunicación mediante un protocolo prefijado por un esquema “cliente-servidor" y constituyen un caso especializado de las arquitecturas en capas, en especial de los modelos multicapas.

Se identifican por su concurrencia.- Esta característica de los sistemas distribuidos permite que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios y/o agentes que interactúan en la red, la carencia de reloj global.- Las coordinaciones para la transferencia de mensajes entre los diferentes componentes para la realización de una tarea, no tienen una temporización general, esta más bien distribuida a los componentes, y los fallos independientes de los componentes.- Cada componente del sistema puede fallar independientemente, con lo cual los demás pueden continuar ejecutando sus acciones. Esto permite el logro de las tareas con mayor efectividad, pues el sistema en su conjunto continua trabajando.

Otra característica de este tipo de sistemas es que los objetos del mismo también pueden encontrarse de forma distribuida. En los sistemas Cliente/Servidor, un objeto distribuido es aquel que esta gestionado por un servidor y sus clientes invocan sus métodos utilizando un "método de invocación remota". El cliente invoca el método mediante un mensaje al servidor que gestiona el objeto, se ejecuta el método del objeto en el servidor y el resultado se devuelve al cliente en otro mensaje. También suelen mantener sus bases de datos distribuidas, construida sobre una red y que pertenecen, lógicamente, a un solo sistema distribuido, las cuales cumplen que la información de la base de datos esta almacenada físicamente en diferentes sitios de la red, aunque en cada sitio de la red, la parte de la información, se constituye como una base de datos en sí misma, además las bases de datos locales tienen sus propios usuarios locales, sus propios gestores de base de datos y programas para la administración de transacciones, y su propio administrador local de comunicación de datos. Estas base de datos locales deben tener una extensión, que gestione las funciones de sociedad necesarias; la combinación de estos componentes con los sistemas de administración de base de datos locales es lo que se conoce como Sistema Administrador de Base de Datos Distribuidas. Este gestor global permite que los usuarios puedan acceder a los datos desde cualquier punto de la red, como si lo hicieran con los datos de su base de datos local, es decir, para el usuario, no debe existir diferencia en trabajar con datos locales o datos de otros sitios de la red. En consecuencia, la base de datos distribuida, es como una unidad virtual, cuyas partes se almacenan físicamente en varias bases de datos "reales", ubicadas en diferentes sitios.[30]

Las aplicaciones Web son un caso particular de los sistemas Cliente-Servidor con representación remota. En donde se dispone de un protocolo estándar: HTTP y un Middleware denominado WebServer . En la actualidad la aplicación de sistemas informáticos basados en Internet es una herramienta fundamental para las organizaciones que desean tener cierta presencia competitiva.

Las ventajas de los sistemas distribuidos con respecto a los centralizados radica en su economía, pues es mucho más barato, añadir servidores y clientes cuando se requiere aumentar la potencia de procesamiento; tienen una mayor confiabilidad, al estar distribuida la carga de trabajo en muchas máquinas la falla de una de ellas no afecta a las demás, el sistema sobrevive como un todo; y en la capacidad de crecimiento incremental, pues se puede añadir procesadores al sistema incrementando su potencia en forma gradual según sus necesidades. La principal ventaja que nos

brindan los sistemas distribuidos es que pueden integrar aplicaciones empresariales que equidistan unas de otras o que se distribuyen en una WAN, es por ello de la elección de integrar este tipo de arquitectura en la construcción y el diseño del producto del cual habla este documento. Las principales desventajas de los sistemas distribuidos radica en el software, en el diseño, la implantación y el uso del software distribuido, la pérdida de mensajes, la saturación en el tráfico y en un problema que puede surgir al compartir datos es la seguridad de los mismos. En general se considera que las ventajas superan a las desventajas, si estas últimas se administran seriamente, por tanto el trabajo de los desarrolladores requiere un esfuerzo muy disciplinado a la hora de desarrollar aplicaciones bajo estos conceptos haciendo mucho hincapié en los llamados detalles o puntos fricción de este tipo de sistemas.

2.3.1.2 Arquitecturas basadas en Componentes.

Los sistemas de software basados en componentes se basan en principios definidos por una ingeniería de software específica. En un principio, hacia 1994, se planteaba como una modalidad que extendía o superaba la tecnología de objetos. Con el paso de los años el antagonismo se fue aplacando y las herramientas (orientadas a objeto o no) fueron adaptadas para producir componentes. En la mayoría de los casos, los componentes terminan siendo formas especiales de DLLs que admiten la tecnología binding, que necesitan registración y que no requieren que sea expuesto el código fuente de la clase.[17]

Hay un buen número de efiniciones de componentes, un componente de software, es una unidad de composición con interfaces especificadas contractualmente y dependencias del contexto explícitas. Que sea una unidad de composición y no de construcción quiere decir que no es preciso confeccionarla: se puede comprar hecha, o se puede producir en casa para que otras aplicaciones de la empresa la utilicen en sus propias composiciones.

Algunas características presentes en el estilo o arquitectura de este tipo son las siguientes:

- Los componentes en el sentido estilístico son componentes y a la vez son las unidades de modelado, diseño e implementación.

- Las interfaces están separadas de las implementaciones, y las interfaces y sus interacciones con el centro de incumbencias en el diseño arquitectónico. Los componentes soportan algún régimen de introspección, de modo que su funcionalidad y propiedades puedan ser descubiertas y utilizadas en tiempo de ejecución. En cuanto a las restricciones, puede admitirse que una interfaz sea implementada por múltiples componentes. Usualmente, los estados de un componente no son accesibles desde el exterior. Que los componentes sean locales o distribuidos es transparente en la tecnología actual.

Este tipo de arquitecturas permite producir componentes de software reutilizables en otras aplicaciones, cualquier componente que sea desarrollado como una parte de una aplicación distribuida es un candidato para ser reutilizado. Organizando los procesos de desarrollo alrededor del paradigma de los componentes permite continuar aumentando el nivel de funcionalidad en las nuevas aplicaciones y reducir el tiempo de desarrollo por lo que ayuda a disminuir también el tiempo de producción de

algunas aplicaciones informáticas. Su uso está muy generalizado en e l desarrollo actual de sistemas por las numerosas ventajas que presenta.

2.3.1.3 Arquitecturas orientas a Servicios.

Sólo recientemente estas arquitecturas que los conocedores llaman SOA han recibido tratamiento intensivo en el campo de exploración de los estilos. Al mismo tiempo se percibe una tendencia a promoverlas de un sub-estilo propio de las configuraciones distribuidas que antes eran a un estilo en plenitud. Muchos la califican como la tendencia que habrá de ser dominante en la primera década del nuevo milenio. Ahora bien, este predominio no se funda en la idea de servicios en general, comunicados de cualquier manera, sino que más específicamente va de la mano de la expansión de los Web services basados en XML, en los cuales los formatos de intercambio se basan en XML 1.0 Namespaces y el protocolo de elección es SOAP, el cual es un formato de mensajes que de XML, comunicado sobre un transporte que por defecto es HTTP, pero que puede ser también HTTPs, SMTP, FTP, IIOP, MQ o casi cualquier otro o puede incluir prestaciones sofisticadas de última generación como WS-Routing, WS-Attachment, WS-Referral, etcétera.[26]

Alrededor de los Web Services, que dominan el campo de un estilo SOA más amplio que podría incluir otras opciones, se han generado las controversias que usualmente acompañan a toda idea exitosa. Al principio hasta resultaba difícil encontrar una definición aceptable y consensuada que no fuera una fórmula optimista de mercadotecnia. El grupo de tareas de W3C, por ejemplo, demoró un año y medio en ofrecer la primera definición canónica apta para consumo arquitectónico, que no viene mal reproducir aquí:

Un Web Service es un sistema de software diseñado para soportar interacción máquina-a-máquina sobre una red, es una entidad programable que proporciona alguna funcionalidad determinada, y es accesible a cualquier número de sistemas que usen las normas de Internet XML y HTTP. Un servicio Web puede ser usado internamente por una aplicación o ser publicado hacia Internet. Estos servicios permiten la ejecución de sus funcionalidades sin importar la plataforma, sistema operativo, o lenguaje en el cual estén implementados.. Posee una interfaz descrita en un formato procesable por máquina (específicamente WSDL). Otros sistemas interactúan con el Web Service de una manera prescripta por su descripción utilizando mensajes SOAP, típicamente transportados usando HTTP con una serialización en XML en conjunción con otros estándares relacionados a la Web.[27]

El mundo de los servicios Web o Web Service está evolucionando a gran velocidad. En los últimos tres años, las especificaciones de los Servicios Web han sido definidas, refinadas y a veces, criticadas; se han lanzado kits de herramientas de servicios Web y los desarrolladores los han utilizado para construir sistemas, y los fabricantes han trabajado para garantizar la interoperabilidad. Al igual que con la infraestructura Web clásica, se está desarrollando e implantando tecnologías de servicios Web en paralelo, progresando a través de un ciclo de realimentación positivo. Aunque los servicios Web han progresado mucho en los últimos años, el trabajo sobre la plataforma continúa. Si vamos a desarrollar sistemas basados en servicios Web, es importante comprender dónde está la plataforma hoy y cual va a ser el futuro. Actualmente, la plataforma de servicios Web ha sido desarrollada lo suficiente para crear aplicaciones distribuidas que se comunican enviando mensajes SOAP sobre http.

En la literatura clásica referida a estilos, las arquitecturas basadas en servicios podrían engranar con el estilo de procesos distribuidos. Otros autores hablan de arquitecturas de componentes independientes que se comunican a través de mensajes. Según esta perspectiva, no del todo congruente con la masa crítica que han ganado las arquitecturas orientadas a servicios en los últimos años, habría dos variantes del estilo: el de participantes especificados (named) que es un estilo basado en el proceso de comunicación donde el ejemplar más conocido sería el modelo cliente-servidor (si el servidor trabaja sincrónicamente, retorna control al cliente junto con los datos; si lo hace asincrónicamente, sólo retorna los datos al cliente, el cual mantiene su propio hilo de control); y el de participantes no especificados (unnamed), paradigma tipo publish/subscribe, o también estilo de eventos. Desde el punto de vista arquitectónico, se puede hacer ahora una caracterización escueta del estilo de participantes especificados.

Un servicio es una entidad de software que encapsula funcionalidad de negocios y proporciona dicha funcionalidad a otras entidades a través de interfaces públicas bien definidas. Los componentes del estilo (o sea los servicios) están débilmente acoplados acoplados. El servicio puede recibir requerimientos de cualquier origen. La funcionalidad del servicio se puede ampliar o modificar sin rendir cuentas a quienes lo requieran. Los servicios son las unidades de implementación, diseño e implementación. Los componentes que requieran un servicio pueden descubrirlo y utilizarlo dinámicamente mediante UDDI y sus estándares sucesores. En general (aunque hay alternativas) no se mantiene persistencia de estado y tampoco se pretende que un servicio recuerde nada entre un requerimiento y el siguiente.

Las especificaciones de arquitecturas basadas en servicios son lo suficientemente amplias para servir de marco de referencia estándar tanto a objetos como a componentes y servicios, pero las herramientas usuales de diseño (por ejemplo UML) no poseen una notación primaria óptima que permita modelar servicios, a despecho de docenas de propuestas en todos los congresos de modelado. Un servicio puede incluir de hecho muchas interfaces y poseer propiedades tales como descripción de protocolos, puntos de entrada y características del servicio mismo. Algunas de estas notaciones son provistas por lenguajes declarativos basados en XML, como WSDL (Web Service Description Language). Como todos los otros estilos, las SOA poseen ventajas y desventajas. Como se trata de una tecnología que está en su pico de expansión, virtudes y defectos están variando mientras esto se escribe.

2.3.1.3(a) XML en los Negocios.

Desde hace algún tiempo se ha comenzado a convivir con unas nuevas siglas en el mundo tecnológico: XML (Extensible Markup Language) o Lenguaje de marcas extensible. Pero en qué consiste este nombre tan poco significativo a nuestros oídos. A continuación se mostrará las ventajas competitivas de utilizar esta tecnología en las aplicaciones empresariales.

XML es un lenguaje de Tags o etiquetas que permite definir de un modo muy sencillo la estructura jerárquica a la que pertenece un dato, así como HTML permite definir la forma en que se muestra un dato en el navegador.

Hasta ahora el déficit más importante a la hora de integrar aplicaciones desarrolladas en distintos lenguajes o sobre distintas plataformas, era que cada forma de

trasmitir los datos era propietaria de la aplicación que la generaba, y en muchos casos la forma de transmisión no permitía que la comunicación fuera fluida. Este problema sucede incluso al intentar integrar aplicaciones sobre una misma plataforma, desarrolladas en un mismo lenguaje. En el mejor de los casos, una vez establecida la comunicación de datos, nos encontrábamos con un verdadero problema a la hora de estructurarlos jerárquicamente de acuerdo al modelo de origen.

XML permite de un modo sencillo estructurar la información de modo que el receptor sepa la relación entre los datos, ya que el mismo documento XML describe el modelo relacional de los mismos. También XML puede describir qué tipo de dato es el que está recibiendo (XML Schema), puede establecer cómo mostrarlo (XSL) e incluso cómo tiene que devolverlo (SOAP).[25]

Así XML permite la comunicación de una aplicación a otra, o recibir y enviar datos estructurados mediante Internet sin tener que idear mecanismos complejos o estructurados excesivamente pesados para rearmar la información como en su origen.

XML tiene múltiples utilizaciones. La transmisión de datos es su origen, pero integrada integrada con XML Schema se puede definir el tipo de dato que está viajando, o si se permiten valores nulos, repetidos, decimales o si se trata de un dato que mantiene una integridad referencial con otra información en el mismo documento transmitido.

XML es la fuente de SOAP, un protocolo basado en el estándar que permite el envío de paquetes de información bidireccional para la integración de aplicaciones remotas. Pudiendo de este modo transmitir datos por referencia e incluso en una transacción.

XML con XSL permite modelar la información visualmente para su presentación de modo que permite generar presentaciones dinámicas principalmente orientadas a B2C (E-Commerce/ Comercio Electrónico).

XML es un modo de parametrizar aplicaciones de forma sencilla, legible y comprensible tanto por aplicaciones como por personas y fácilmente accesible desde cualquier tipo de aplicación.

El 99% de las aplicaciones de escritorio actuales soporta lectura, escritura, importaciones y exportaciones a este formato para persistir la información de manera consistente, y con cada nueva versión XML se integra más en el Back Office de los sistemas de escritorio, gestión, Web, etc.

Todo esto con un modelo descriptivo en formato de texto, y basado en estándares de la industria definidos por el W3C (World Wide Web Consortium), que garantiza que la información podrá ser transmitida por Internet sin ningún tipo de traba (Firewalls) y que la interpretación de la misma es universal más allá de plataformas o lenguajes de desarrollo.

Una solución hoy día, no debiera cerrar la posibilidad de integración o comunicación con nuevas aplicaciones, módulos, funcionalidad o dispositivos. Tener en cuenta cuenta la transmisión e integración de información utilizando XML es un requisito necesario a la hora de establecer los alcances de una solución. Por todo esto una

solución tecnológica no es tal si no se ha analizado convenientemente la utilización de la infraestructura XML en la misma.

Como cualquier otra herramienta, arquitectura, metodología, etc. Relacionada con proyectos, no solo informático, sino de cualquier índole, el elemento inicial e indispensable es la detección de una necesidad en un determinado momento y el posterior análisis de la misma.

En este análisis, donde se definen las necesidades que dan origen a la solución, se deben medir las posibilidades inmediatas y futuras de utilizar XML. Pero su utilización puede variar drásticamente según el escenario en el cual se quiera implementar la solución.

A continuación se plantean algunos ejemplos prácticos que nos encontramos en los proyectos que desarrollamos y que muestran claramente las posibilidades reales de XML en la integración de aplicaciones de gestión en la empresa.

Un ejemplo clásico y muy importante es el uso de XML en la integración de sistemas de información heterogéneos.

En la actualidad el mercado está inundado de aplicaciones específicas y/o verticales que junto a la existencia de aplicaciones generales y/o horizontales, lleva a que muchas veces se deban integrar aplicaciones desarrolladas sobre plataformas, modelos de datos y lenguajes distintos.

Así, habitualmente nos encontramos tres opciones cuando se plantea este tipo de problemas:

- Mantener las aplicaciones que funcionan correctamente e integrar con XML.
- Cambiar todos los sistemas para conseguir una integración “de fábrica”.
- No integrar manteniendo las aplicaciones independientes.

Es lógico que la decisión tomada deba sustentarse tecnológicamente y en relación Coste - Beneficio.

La tercera opción, la existencia de aplicaciones no integradas es muy problemática debido a las ineficiencias que se genera en los procesos de la empresa por lo que esa opción se debe descartar aunque se encuentra en la realidad más veces de lo que sería aconsejable.

La segunda opción, cambiar todos los sistemas tiene un impacto muy importante en cuanto a costes y en cuanto a cambios en las empresas por lo que muchas veces también es desestimada.

Frente a las dos opciones anteriores, la utilización de XML permite desarrollar una solución integradora (MiddleWare) para que puedan comunicarse entre sí los sistemas que están probados y funcionando correctamente, otorgando la ventaja de lograr la mejor integración, con un coste contenido y con las ventajas de lograr resultados en corto plazo.

XML también ofrece importantes oportunidades para la gestión de catálogos electrónicos a través de Internet. Frente a otros lenguajes, la utilización de XML permite que la gestión del contenido se limite a su carga en base de datos y no hay que hacer artesanalmente cada página del catálogo.

Los catálogos se deben gestionar con la utilización de XML como medio de transporte de los datos de artículos, familias, categorías, descripciones, etc., y los formatos de su visualización estarán dados por XSL y su lenguaje XPath, que permite dinámicamente armar los contenidos de un catálogo. El catálogo electrónico es el caso por excelencia donde todos los productos con sus descripciones y características siempre se muestran con el mismo formato.

En la actualidad la movilidad del personal de una empresa es en muchos casos vital para su funcionamiento. La principal complicación en estos escenarios consiste normalmente en dotar al usuario del dispositivo móvil de información inmediata, oportuna y actualizada proveniente del centro de datos.

Además, el usuario debe tener la posibilidad de modificar dicha información y actualizarla en el centro de datos sin tener que trasladarse físicamente, conectarse a la red y actualizar.

La tecnología móvil nos permite actualmente utilizar PDAs, Portátiles, teléfonos móviles, etc. que se puede comunicar con un servidor intercambiando información en línea con XML, WML y Servicios Web, y así optimizar la dinámica de la empresa contando con información fiable y actualizada en todo momento.

Quizás la tecnología que más dará que hablar en el futuro inmediato es la relacionada con los Servicios Web. Esta nueva forma de transmisión de datos permite la comunicación bidireccional, con lo cual se pueden establecer comunicaciones entre aplicaciones utilizando protocolos estándares basados en XML como SOAP (Simple Object Access Protocol), y especificaciones (aunque no estándares aún) como UDDI (Universal Description, Discovery and Integration) y WSDL (Web Service Definition Language).

Estas tecnologías encapsulan el XML en paquetes de transmisión o mensajes (SOAP), permiten la ubicación en internet de los Servicios Web existentes cual si se tratase de unas Páginas Amarillas de Web Services (UDDI), y dan la posibilidad de que la aplicación que hace uso del Servicio Web comprenda las interfaces de comunicación de este último (WSDL).

Este conjunto de definiciones permiten que las aplicaciones distribuidas se basen en tecnología abierta basada en estándares a diferencia de protocolos propietarios como DCOM o CORBA.

El resultado es la posibilidad de que aplicaciones de escritorio o Web se comuniquen con otras aplicaciones remotas para obtener datos o gestionarlos, como si se tratase de una aplicación local, sin importar la plataforma en que se encuentra cada una en tanto se respeten los estándares citados.

Así, si nuestra aplicación requiriese de la cotización actual de determinada empresa, se puede buscar mediante UDDI un Servicio Web que brinde dicha información,

utilizar las interfaces programáticas del mismo con WSDL, y finalmente comunicar nuestra aplicación con el Web Service mediante SOAP, sin necesidad de tener que preocuparse por cortafuegos (firewalls) que pudieran interrumpir la comunicación por tratarse de un estándar basado en texto plano y que se comunica por un protocolo montado sobre HTTP.

Quizás el hecho de escuchar tanto acerca de XML, XSL, SOAP, Servicios Web, nos puede llevar a la conclusión errónea de que no es más que una simple moda producto de la escasez en los resultados tecnológicos de los últimos tiempos. La verdad es que XML es una tecnología que promete quedarse entre nosotros por mucho tiempo. Las principales empresas de software a nivel mundial han apostado todas sus fichas a la integración de sistemas y dispositivos. El objetivo a corto y medio plazo es que las aplicaciones sirvan datos que puedan ser visualizados indistintamente en páginas Web, teléfonos móviles, PDAs, portátiles, televisión, electrodomésticos, etc. y la única forma posible de integración hasta el momento es la transmisión de datos por XML y la comunicación por SOAP (Servicios Web).

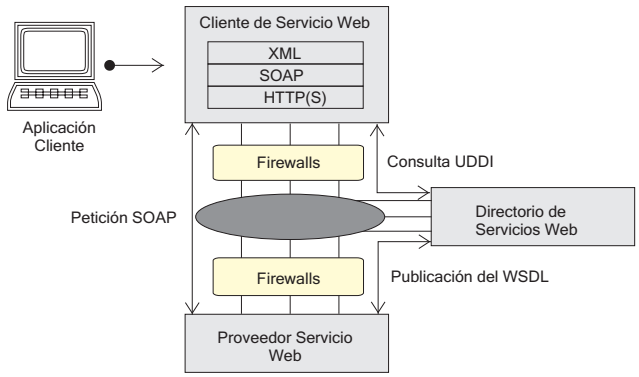


Fig.2.2 Conexión de sistemas clientes con aplicaciones remotas

Un estudio realizado por Giga Group muestra que durante el año 2002 se ha utilizado XML en un 45% de las aplicaciones denominadas críticas, lo cual da clara idea del peso de esta tecnología en el desarrollo de aplicaciones actual.

XML es una herramienta que se debe tener en cuenta a la hora de establecer soluciones tecnológicas. Los escenarios presentados en este artículo no son más que algunas de la infinidad de posibilidades que presenta este lenguaje.

XML tampoco es la solución ideal en todos los casos debido a que tiene su cuello de botella en la performance de ejecución y que los costes de ancho de banda no son los ideales para el uso de esta tecnología, aunque día a día se hacen más accesibles.

Asimismo las empresas líderes de tecnología como Sun, Microsoft, IBM, etc. soportan XML en todas las líneas de productos software y tienen sus propios marcos de trabajo (frameworks) para el desarrollo de Servicios Web XML.

La tendencia en el futuro está claramente marcada.

2.3.1.4 Arquitecturas basadas en Eventos.

Las arquitecturas basadas en eventos se han llamado también de invocación implícita. Otros nombres propuestos para el mismo estilo han sido integración reactiva o difusión (broadcast) selectiva. Por supuesto que existen estrategias de programación basadas en eventos, sobre todo referidas a interfaces de usuario, y hay además eventos en los modelos de objetos y componentes, pero no es a eso a lo que se refiere primariamente el estilo, aunque esa variedad no está del todo excluida. En términos de patrones de diseño, el patrón que corresponde más estrechamente a este estilo es el que se conoce como Observer, un término que se hizo popular en Smalltalk a principios de los ochenta; en el mundo de Java se le conoce como modelo de delegación de eventos.

Las arquitecturas basadas en eventos se vinculan históricamente con sistemas basados en actores, daemons y redes de conmutación de paquetes (publicación-suscripción). Los conectores de estos sistemas incluyen procedimientos de llamada tradicionales y vínculos entre anuncios de eventos e invocación de procedimientos. La idea dominante en la invocación implícita es que, en lugar de invocar un procedimiento en forma directa (como se haría en un estilo orientado a objetos) un componente puede anunciar mediante difusión uno o más eventos. Un componentes de un sistema puede anunciar su interés en un evento determinado asociando un procedimiento con la manifestación de dicho evento. Desde el punto de vista arquitectónico, los componentes de un estilo de invocación implícita son módulos cuyas interfaces proporcionan tanto una colección de procedimientos (igual que en el estilo de tipos de datos abstractos) como un conjunto de eventos. Los procedimientos se pueden invocar a la manera usual en modelos orientados a objeto, o mediante el sistema de suscripción que se ha descripto anteriormente en estas [17]

Los ejemplos de sistemas que utilizan esta arquitectura son numerosos. El estilo se utiliza en ambientes de integración de herramientas, en sistemas de gestión de base de datos para asegurar las restricciones de consistencia (bajo la forma de disparadores, por ejemplo), en interfaces de usuario para separar la presentación de los datos de los procedimientos que gestionan datos, y en editores sintácticamente orientados para proporcionar verificación semántica incremental.

Entre las ventajas enumeradas en relación con el modelo señalan que se optimiza el mantenimiento haciendo que procesos de negocios que no están relacionados sean independientes, se alienta el desarrollo en paralelo, lo que puede resultar en mejoras de performance, es fácil de empaquetar en una transacción atómica, es agnóstica en lo que respecta a si las implementaciones corren sincrónica o asincrónicamente porque no se espera una respuesta, se puede agregar un componente registrándolo para los eventos del sistema; se pueden reemplazar componentes. Entre las desventajas se pueden mencionar que el estilo no permite construir respuestas complejas a funciones de negocios, un componente no puede utilizar los datos o el estado de otro componente para efectuar su tarea, y que cuando un componente anuncia un evento, no tiene idea sobre qué otros componentes están interesados en él, ni el orden en que serán invocados, ni el momento en que finalizan lo que tienen que hacer. Además pueden surgir problemas de performance global y de manejo de recursos cuando se comparte un repositorio común para coordinar la interacción. En esta estrategia juega un rol importante el Servicio de Eventos, el cual a su vez proporciona un buen

punto de partida para la implementación del estilo o patrón, según se esté concibiendo la arquitectura o implementándola.

2.3.2 Lenguajes de Programación.

Uno de los ejes fundamentales que diferencian a Internet de otros medios de comunicación es la interacción y personalización de la información con el usuario. Esto se logra por medio de alguno de los diferentes lenguajes de programación para Web que existen hoy en día. Dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura Cliente/Servidor de esta plataforma de desarrollo:los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor podemos encontrar entre los más sobresalientes por el auge que estos han tenido, algunos como PERL, ASP, PHP, Java, JSP, los módulos CGIs e ISAPIs etc., etc. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la Información etc. Del lado del cliente se encuentran principalmente el JavaScript y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores.

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

Existe una gran variedad de lenguajes de programación en los que se pueden desarrollar aplicaciones o funcionalidades para la Web, en esta sección explicaremos algunas cosas sobre los más utilizados no sin antes mencionarles gran parte de los que hoy pueden encontrarse; podrán darse cuenta que hacer una selección puede convertirse en un proceso bastante engorroso por la amplia lista de opciones presentes.

Lenguajes para el desarrollo de Aplicaciones Web:

Lenguajes de Programación Orientada a Objetos: C++, Java, J2SE, J2EE, J2ME, C#, Smalltalk, Eiffel, Lexico, Oberon, Objective-C, Simula, Sather, J#, ..

Lenguajes de Shell y Script: Phyton, Perl, AWK, Ruby, tcl, Bash, sh, Javascript, VBScript, ActionScript, MaxScript, ...

Lenguajes de Programación Web: PHP, ASP, ASP.NET, ColdFusion, ...

Lenguajes de Etiquetado (marcado): HTML, CSS, XML, WAP, XHTML, XLS, XLS-FO, XLST, PostScript, RTF, SMIL, SGML, SVG, Docbook, Latex, MathML, Natural Docs, VRML, X3D, Xlink, XPath, XPointer, ...

2.3.2.1 Perl.

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el Web. Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer

información de archivos de texto y generar informes a partir del contendido de los ficheros. Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP.[33]

2.3.2.2 ASP

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

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

2.3.2.3 PHP

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

PHP, en el caso de estar montado sobre un servidor Linux o Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM (Common Objet Model, un componente desarrollado por Microsoft para el trabajo con Aplicaciones Web) que se realizan entre todas las tecnologías implicadas en una página ASP.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales. [16]

PHP es la gran tendencia en el mundo de Internet. Últimamente se puede observar un ascenso imparable, ya que cada día son muchísimas más las páginas Web que lo utilizan para su funcionamiento, según las estadísticas, PHP se utiliza en más de 10 millones de páginas, y cada mes realiza un aumento del 15%. [16]

Resumiendo, el PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos y contiene unas 40 extensiones estables sin contar las que se están experimentando, además de que: es software libre, lo que implica menos costes y servidores más baratos que otras alternativas,es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado, su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de

modo que si se esta familiarizado con esta sintaxis, le r esultara muy fácil aprender PHP. wSu librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP, además PHP tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos.[29][35]

2.3.2.4 SMARTY

SMARTY es un motor de plantillas para PHP. La finalidad de trabajar con plantillas es la de separar el código PHP del código HTML, con la ventaja de que un diseñador pueda trabajar en su ámbito sin tener que saber PHP. Por consiguiente, el programador puede hacer los cambios a la lógica de la aplicación sin la necesidad de reestructurar el diseño, y el diseñador puede hacer los cambios a las plantillas sin romper la lógica de la aplicación. Algunos de los principales aspectos de SMARTY son: es sumamente rápido, ninguna plantilla se analiza dos veces, sólo compila una vez, tiene inteligencia para recompilar sólo los archivos de las plantillas que han cambiado, se pueden hacer funciones personalizadas y personalizar las variables, por lo que el idioma de la plantilla es sumamente extensible, se puede configurar los delimitadores que etiquetan la sintaxis, se puede usar {}, {{}}, <!--{}-->, <% %>,etc, se pueden anidar ilimitadas secciones de if/else, for, foreach, etc y permite el uso arbitrario de las fuentes de la plantilla, o sea que una plantilla puede ser usada por varias páginas PHP, siempre que muestren el mismo contenido.

2.3.2.5 JSP

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es pues, una tecnología orientada a crear páginas Web con programación en Java.

Con JSP podemos crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual. [36]

2.3.2.6 Ruby

Ruby es un " lenguaje de guiones (scripts) para una programación orientada a objetos rápida y sencilla" creado en Japon en el año 1993 por Yukihiro Matsumoto. Es un lenguaje de guiones interpretado que da la posibilidad de realizar directamente llamadas al sistema operativo, tiene potentes operaciones sobre cadenas de caracteres y expresiones regulares y una gran retroalimentación inmediata durante el proceso de desarrollo. Además es rápido y sencillo porque son innecesarias las declaraciones de variables , las variables son de tipo dinámico, la sintaxis es simple y consistente y la gestión de la memoria es automática.

Ruby es un lenguaje de programación interpretado, de muy alto nivel y orientado a objetos. es diferente, casi todos los comentarios sobre Ruby son puro elogio, y es que combina las mejores características de Smalltalk, Perl e incluso alguna cosa de programación funcional.

2.3.3 Gestores de Base de Datos.

Un Sistema de Gestión de Bases de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Los SGBD permiten al programador convencional ahorrarse horas de trabajo dedicadas a la seguridad, gestión de los datos, chequeo de errores, etc.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Mocrosoft SQL Server, PostgreSQL, Internase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

2.3.3.1 MySQL

MySQL AB desarrolla una familia de soluciones de bases de datos de alto rendimiento. Su producto principal es el servidor MySQL que junto con Linux, Apache, PHP y Perl, se ha convertido en una de las herramientas más populares para crear sitios web en Internet y es la base de datos libre más popular.

La empresa fue fundada en el año 2001 por dos suizos y un finlandés, y ha sido rentable desde sus inicios. En octubre de 2001 obtuvo financiación de un fondo de capital de riesgo que ha utilizado para crecer ordenadamente.

MySQL AB es el poseedor único de los derechos de autor de sus productos. Utiliza el sistema de licencia dual y ofrece su producto bajo una licencia GPL y otra propietaria. Los usuarios pueden descargar el software, usarlo libremente y modificarlo, integrar y distribuir estas mejoras. Sin embargo, los usuarios de la versión libre deben seguir las reglas de la licencia GPL que estipula: Si se redistribuyen una aplicación basada en MySQL el código de fuente completo de la solución también debe esta abierto y disponibles para la redistribución. Los clientes que utilicen MySQL como parte de una solución de software o hardware propietario y no quieran proveer el código fuente pueden comprar una licencia propietaria del producto a partir de 500 dólares.

Los beneficios de MySQL AB proceden principalmente de la venta de servicios, soporte técnico, y licencias comerciales de su producto. Estos ingresos se utilizan para continuar desarrollando el producto libre MySQL.

2.3.3.2 PostgreSQL

Posee una estabilidad y confiabilidad legendaria nunca ha presentado caídas en varios años de operación de alta actividad. Está disponible 34 plataformas Unix en la última versión estable, existe una versión para Windows usando la plataforma Cygwin. Fue diseñado para ambientes de alto volumen intentando estar a la altura de Oracle, Sybase o Interbase. Escala muy bien al aumentar el número de CPUs y la cantidad de RAM. Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial. Tiene mejor soporte para subselects, triggers, vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos. Sin embargo consume muchos recursos y no escala bien en la plataforma Windows.

2.4 METODOLOGÍAS DE DESARROLLO PARA APLICACIONES WEB.

La aparición de aplicaciones y sitios Web proporciona la explotación de otros mercados y servicios antes impensables como el comercio electrónico, la enseñanza virtual, etc., y esto conlleva un importante crecimiento en el desarrollo del software sobre dicha tecnología. Ahora bien, desde el punto de vista de la Ingeniería del Software es importante dotar de los mecanismos adecuados, para que la realización de este tipo de aplicaciones satisfaga las necesidades tanto de los usuarios como de los clientes que contratan el desarrollo de este tipo de aplicaciones. Pero actualmente no existe una metodología universalmente aceptada, que guíe en el proceso de desarrollo de aplicaciones Web.

En cualquier caso, existen criterios universalmente aceptados acerca del desarrollo software. Por ejemplo, y según afirma Jacobson (2000), el modelo de proceso más adecuado para el desarrollo de software es un proceso iterativo e incremental, puesto que a diferencia de otros modelos de proceso, como por ejemplo el modelo en cascada, permite la obtención de diversas versiones del producto software antes de la entrega final del mismo y la depuración y validación progresiva del mismo, lo que sin duda redundará en un software más satisfactorio para usuarios y clientes. Además y según indica Conallen (2000), con este tipo de proceso es posible añadir o modificar requisitos que no han sido detectados con anterioridad. Aún no existe ninguna propuesta universalmente aceptada para el desarrollo Web, pero Fraternali (2000) indica que una posible solución al desarrollo adecuado de aplicaciones Web, sería combinar los ciclos de vida tradicionales con las propuestas de diseño para el desarrollo de aplicaciones hipermedia. De hecho, algunos de los trabajos existentes, relacionados con la tecnología hipermedia y Web, combinan el tratamiento de esas características especiales, con el uso de un modelo de proceso iterativo e incremental (Atzeni 1998; Fraternali, 1998; Isakowitz, 1995; Schawbe and Rossi, 1995; Lowe and Hall, 1999). En cualquier caso los métodos clásicos no son adecuados para el desarrollo de aplicaciones Web, puesto que no contemplan determinadas características específicas de este tipo de aplicaciones, Lowe and Hall (1999). Por otra parte, las metodologías tradicionales generalmente imponen un proceso de desarrollo demasiado pesado y burocrático según afirma Fowler (2001), lo que impide un desarrollo ágil y rápido para este tipo de aplicaciones.

Como reacción a estas metodologías clásicas, recientemente han aparecido un nuevos paradigmas de ciclo de vida del software. Son las metodologías y procesos de desarrollo para aplicaciones web o Ingeniería Web, las cuales hoy garantizan un proceso de desarrollo suficiente pero no excesivo.

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un software más eficiente y predecible.

Para ello, se hace un especial hincapié en la planificación total de todo el trabajo a realizar y una vez que esta todo detallado, comienza el ciclo de desarrollo del producto software. Este planteamiento está basado en el resto de disciplinas de ingeniería, a pesar de que el software no pueda considerarse como la construcción de una obra clásica de ingeniería.

Con estas metodologías se lleva trabajando desde hace tiempo y no ha habido en

ningún caso ninguna experiencia traumática acerca de su uso. Pero aún así, han recibido diversas críticas, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Una metodología para el desarrollo de un proceso de software es un conjunto de filosóficas, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas Informáticos. Por ello escoger la metodología que va a guiar el proceso de desarrollo del sistema es un paso tan importante.

Las distintas metodologías se pueden dividir en tres generaciones en base a su sofisticación, estas son:

- Primera Generación:(Principios de los 90) Se sientan las bases de la ingeniería Web, en los que se incluyen conceptos como construcción de navegación, separación entre estructuras y el contenido durante el ciclo de desarrollo.

- Segunda Generación: (Segunda mitad de los 90) Se refinan los primeros modelos y se añaden los soportes de funcionalidad básica y se llevan a cabo los primeros esbozos de proceso donde se delimitan los modelos conceptual, lógico y físico.

- Tercera generación: (A partir del 2000): Se lleva a cabo la profundización en el soporte para la funcionalidad, enfatizacion de la figura del usuario en los métodos, y se avanza hacia la estandarización de notaciones, procesos y lenguajes de especificación.

Las metodologías que hoy compiten en el mercado para el desarrollo de aplicaciones Web son: HDM (Método de Diseño Hipermedia, RMM (Metodología de Administración de Relaciones), EORM (Metodología de Relaciones de Objetos Mejorada], OOHDM, SOHDM (Metodología de Diseño Hipermedia orientada a objetos y basada en escenarios), WSDM (Método de Diseño de Sitios Web) , WAE-Proceso Conallen (Extensión de Aplicación Web para UML), RUP+UWE (Proceso Unificado de Desarrollo de Aplicaciones Web), XP (Programación Extrema). En este epígrafe abundaremos en algunos datos sobre algunas de ellas.

2.4.1 Proceso Unificado+UWE

El Proceso Unificado es una propuesta de proceso para el desarrollo de software orientado a objeto que utiliza Unified Model Language (UML) para describir todo el proceso. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.[19]

Sus características principales son: 1. Guiado/Manejado por casos de uso. 2.Centrado en arquitectura. 3.Iterativo e Incremental. 4.Desarrollo basado en componentes. 5.Utilización de un único lenguaje de modelación. 6.Proceso Integrado.

Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases, las cuales se explican detalladamente en la bibliografía correspondiente. Las etapas y los subproductos de cada fase(“artefactos”).

La propuesta de Ingeniería Web basada en RUP-UML (UWE (Koch, 2000)) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

2.4.2 Metodologías Ágiles (Programación Extrema)

Las metodologías de dasrrollo ágiles aportan como novedad, nuevos métodos de trabajo que apuestan por una cantidad apropiada de proceso. Es decir, ni se pierden en una excesiva cantidad de cuestiones burocráticas ni defienden tampoco la falta total del proceso. Buscan el equilibrio en la relación proceso/esfuerzo.

Las diferencias existentes entre ambos grupos de metodologías surgen por un enfoque y objetivos diferentes. Y como principales diferencias, Fowler (2001) identifica las siguientes:

- Las metodologías ágiles son adaptativas mas que predictivas. Las metodologías tradicionales potencian la planificación detallada de prácticamente todo el desarrollo software a largo plazo. Pero cuando se produce un cambio, toda esta planificación puede venirse abajo. Sin embargo, las metodologías ágiles proponen procesos que se adaptan y progresan con el cambio, llegando incluso hasta el punto de cambiar ellos mismos.

- Las metodologías ágiles están orientadas al personal mas que orientadas al proceso. Intentan trabajar con la naturaleza del personal asignado al desarrollo, más que contra ellos, de tal forma que permiten que la actividad de desarrollo software se convierta en una actividad grata e interesante.

Existen diversas metodologías que coinciden en llamarse metodologías ágiles. Y aunque entre ellas comparten muchas características tienen también diferencias significativas. A continuación se presentan algunas de las metodologías ágiles más representativas.

① Extreme Programming (XP) - La programación extrema (Beck, 1999; Mc Breen, 2000) concede una gran importancia a las pruebas del software (testing). Aunque la mayoría de los procesos las tienen en cuenta, generalmente lo contemplan de una forma demasiado ligera y superficial. Sin embargo, XP lo toma como base para el desarrollo y cada programador que escribe código también escribe los casos de prueba.

Estos forman parte del proceso continuo de generación de código y se integra continuamente con ello, lo que garantiza una plataforma estable para el futuro desarrollo. Sobre dicha plataforma se genera un proceso de diseño evolutivo, que es la base del sistema y que se enriquece con cada iteración. Nunca se generan diseños futuros. Según afirma Fowler (2001), el resultado es un proceso de diseño que combina adecuadamente la disciplina con la adaptabilidad.

② Open source - Open source apuesta por la distribución de trabajo entre diferentes equipos, al igual que ocurre con la mayoría de los procesos adaptativos. La mayoría de los proyectos open source cuentan con supervisores de código. Estos supervisores de código, son las únicas personas autorizadas para realizar un cambio en el repositorio del código fuente. Por otra parte, el resto del personal puede realizar cualquier cambio en el código base. Sin embargo, el supervisor del código es la persona responsable de coordinar y de mantener la consistencia del diseño del software. Una de las principales ventajas de los desarrollos open source es que la depuración es altamente paralelizable, aunque un gran número de personas puedan verse involucradas. Cuando se soluciona un error, se envía la solución al supervisor de código, lo que garantiza que alguien realiza la modificación de forma fiable mientras otra parte del personal se dedica a las tareas de depuración.

Los procesos ágiles son una buena elección cuando trabajamos con requisitos desconocidos o variables. Si no existen requisitos estables, no existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo ni en dinero. En estas situaciones, un proceso adaptativo será mucho más efectivo que un proceso predictivo. Por otra parte, los procesos de desarrollo adaptativos también facilitan la generación rápida de prototipos y de versiones previos a la entrega final, lo cual agradará al cliente. Pero la mayor barrera que habrá que salvar será convencer al cliente de que no existen una planificación y una forma fija de hacer las cosas. En cualquier caso, lo que se garantiza es un menor riesgo ante la posibilidad de cambios en los requisitos, porque los cambios existen, y los procesos adaptativos permitirán estos cambios lo que en definitiva, garantizará que el producto final sea el deseado por el cliente. Según afirma Booch (2001), todas estas razones son las que hacen que los procesos indicados por las comunidades de Extreme Programming y de Open Source hayan suscitado tanto interés.

En general, las aplicaciones Web cumplen la mayor parte de las características mencionadas en el párrafo anterior, por lo que la utilización de procesos ágiles podría ser beneficiooso para este tipo de desarrollos. La necesidad del cliente que contrata un desarrollo Web es que su producto esté disponible en la red lo más pronto posible. Si no se contempla esta necesidad, la aplicación no resultará un producto satisfactorio para el cliente. Puesto que los procesos ágiles permiten obtener versiones de producto previas a la versión final, si se aplican adecuadamente estos procesos el cliente podrá disponer de forma rápida de alguna versión intermedia. Además el ciclo de desarrollo de la mayoría de los sitios y aplicaciones Web es extremadamente corto, Overmyer (2000). Esto implica que generalmente no se aplique ningún tipo de proceso, pero sin duda y como se mencionaba anteriormente más vale un proceso ágil que nada. Por otra parte, los desarrollos Web se perciben como desarrollos sencillos y los desarrolladores son sometidos a una gran presión de trabajo para terminar lo más pronto posible. Esta forma de trabajar va a implicar sin duda algunas modificaciones. Luego sería conveniente garantizar un proceso de desarrollo adaptable a los cambios. Otra cuestión fundamental a tener en cuenta es que las aplicaciones

Web se desarrollan sin conocer los perfiles de los usuarios finales de las mismas, o lo que es lo mismo sin conocer los requisitos de usuario del sistema. Sin lugar a dudas esto implicará cambios en los requisitos inicialmente detectados, lo que nos lleva de nuevo a la elección de un proceso adaptativo. Por lo tanto, podríamos concluir que este tipo de procesos son especialmente aplicables al desarrollo de aplicaciones para la Web.

2.4.3 Metodología hipermedial de diseño orientada a objetos (OOHDM).

OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación.

Se usa notación similar a UML (Lenguaje de Modelado Unificado) y tarjetas de clases y relaciones similares a las tarjetas CRC (Clase Responsabilidad Colaboración). El esquema de las clases consiste en un conjunto de clases conectadas por relaciones. Los objetos son instancias de las clases. Las clases son usadas durante el diseño navegacional para derivar nodos, y las relaciones que son usadas para construir enlaces.

Dicha metodología propone dedicar un tiempo importante en las fases previas a la implementación. Esta inversión de tiempo está ampliamente justificada no sólo porque simplifica el proceso de Desarrollo, facilitando el trabajo del equipo encargado de cada capa de la aplicación, sino también durante su mantenimiento y eventual extensión. Son quizás estas últimas tareas las más difíciles de lograr con tecnologías tradicionales, y aún imposibles en muchos casos donde no existe diseño detallado y la implementación concentra conceptos heterogéneos muy difíciles de modificar.

OOHDM propone un conjunto de tareas que en principio pueden involucrar mayores costos de diseño, pero que a mediano y largo plazo reducen notablemente los tiempos de desarrollo al tener como objetivo principal la reusabilidad de diseño, y así simplificar la evolución y el mantenimiento.

2.5 TECNOLOGÍAS A UTILIZAR PARA LA CONSTRUCCIÓN DEL SISTEMA.

Luego del estudio de las tendencias y tecnologías actuales, acorde con los intereses y las posibilidades del cliente, el equipo de desarrolladores se manifestó, para la ejecución de este proyecto, hacia las siguientes tecnologías.

Para dicha elección se debía cumplir con la idea de no hacer un sistema rígido que no permitiera la migración hacia otras tecnologías si fuese necesario, además que los costes de producción y mantenimiento fueran lo más bajos posibles sin que esto afectara la calidad de la solución.

Se decidió entonces utilizar no una arquitectura de desarrollo pura sino, obligados por las características del proyecto, el cual estará soportado sobre plataforma Web, usar una combinación de las explicadas en este capítulo, o sea, emplear una arquitectura multicapas, basada en componentes, orientada a servicios y con tecnología cliente-servidor distribuida.

Construir el sistema utilizando el PHP como lenguaje de programación y MySQL

como Sistema Gestor de la Base de Datos, además, emplear el XML y sus tecnologías asociadas para garantizar la portabilidad y la reusabilidad de componentes ya desarrollados, por ejemplo reutilizar la plataforma de servicios PLASER la cual ya tiene implementada una serie de servicios para la gestión y la comunicación con las bases de datos del sistema, así como la gestión de la seguridad, la comunicación entre componentes, entre otras funciones.

Además decidimos se utilizaría principalmente el Racional Rose Enterprise Edition 2003, Software Propietario para Windows, de la Rational Software Corporation, como herramienta para la modelación ingenieril basada en RUP-UML y dotadas de estereotipos para el desarrollo de Aplicaciones Web; el Corel Draw 12, de la Corel Corporation, Software Propietario para Windows para el diseño y tratamiento de imágenes, así como para generar la documentación de este proyecto; el DreamWeaver MX 2004, de Macromedia Inc., software propietario para Windows para el diseño y montaje de la aplicación Web; el Stylus Studio 5.1, de la Sonic Software Corporation, el cual 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. 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; el NuSphere PHPEd 3.3.3, el PHPEd es un editor para programadores con soporte para múltiples formatos, similar a otras aplicaciones como PHP Coder, incluye un cliente de FTP y un servidor Web integrados, es un software propietario multiplataforma; el WAMP5 v1.4.4 , como paquete que incluye el servidor Apache 1.3.x. con PHP 5.x.x , el servidor MySQL 4.x.x , el PHPmyadmin, el SQLitemanager y el cual es una herramienta para la múltiple gestión de Aplicaciones Web ; entre otras herramientas más según sea la necesidad durante el proceso de construcción del sistema.

La poca experiencia en entornos de desarrollo bajo plataforma Linux nos obligaron a decidirnos por algunas herramientas de desarrollo bajo licencias no libres, lo cual podría tener alguna repercusión económica y legal en el futuro si no garantizamos el pago de las mismas para la explotación y venta del producto, aunque las principales herramientas de soporte e implementación se mantienen bajo estándares libres. La elección de la metodología de desarrollo y muchas herramientas a utilizar tuvo un fuerte impulso institucional por el centro de estudios al cual pertenecemos y por la administración de la empresa en la cual estamos realizando este trabajo de ingeniería, aunque siempre evaluamos otras variantes con el fin de encontrar las mejores propuestas a nuestro alcance.

La decisión de tomar, evaluar o usar una determinada herramienta de desarrollo, o tecnología, cuando se va a realizar la tarea de enfrentar un proyecto informático, es una de las tareas más importantes y a la vez más difíciles. La causa es la gran variedad de aplicaciones, elementos teóricos, tendencias y tecnologías, con virtudes y defectos cada una, existentes en el mercado de la Información.

2.6 Conclusiones del Capítulo...

Se ha tratado en este capítulo las ventajas del uso de software desarrollado y entregado bajo lincencias libres, de las tendencias y tecnologías que sobre el mundo de la Web son más actuales e importantes para los desarrolladores de sistemas con estás características, acerca de las metodologías de trabajo para el desarrollo de productos informáticos con calidad, con capacidad de ser reutilizables y económicamente accesibles para todos los interesados en el mismo y finalmente se ha planteado las herramientas que pensamos utilizar en la confección de sistema.

Se ha intentado dar una panorámica lo más actual posible del tema en cuestión, se conoce que existe mucha más información de la que se ha podido reflejar aquí de forma muy resumida, es por ello que se recomienda, si está interesado en conocer más sobre los temas tratados que se dirija a la referencia bibliográfica de este trabajo documental.

Por otra parte hemos dejado en su conocimiento la forma de pensar de los desarrolladores de este proyecto, los objetivos que persiguen y como intentan encontrar la solución a la situación problémica que los trajo a realizar el presente trabajo.

También con este estudio quedan cumplidas parte de las tareas que teníamos para el enfrentamiento de este proyecto, explicadas en el capítulo anterior. Alcanzándose una posición superior y mucho más madura en el sentido profesional, que permite lanzarse al controvertido mundo de la Ingeniería de Sistemas Informáticos y comenzar a realizar análisis más profundos y el diseño detallado de la solución que se persigue desarrollar para la empresa contratante y los clientes de la misma.

CAPÍTULO 3
Modelado del Negocio

INTRODUCCIÓN

Ya habiendo hecho todo un estudio de las tendencias y tecnologías que más se utilizan en el desarrollo de las Aplicaciones Web, se dirigirá el curso de acciones entonces a realizar el proceso ingenieril de acuerdo a la metodología a emplear en este proyecto de desarrollo (RUP+UWE), se intentará describir el negocio de los clientes, tal y como funciona actualmente en el Sistema Nacional de Salud, luego se decribirá la solución que le proponemos a la Dirección Nacional de Atención Primaria de Salud y la de Registros Médicos y Estadísticas del MINSAP, los cuales son nuestros usuarios finales y clientes respectivamente, con el fin de comenzar la modelación de nuestro sistema, en cumplimiento con las buenas prácticas de la Ingeniería de Software y con el objetivo de presentar un producto a nuestro cliente y posteriormente a los usuarios finales de esta aplicación, que cumpla con todas las expectativas generadas con el comienzo de este proyecto de innovación tecnológica.

Este capítulo muestra la fase inicial de desarrollo de una aplicación informática real, los sumergirá en las venas de un proceso de ingeniería actual y en una industria que despunta en nuestro país y que promete, y es, un puntal del desarrollo mundial.

Siguiendo el ciclo de desarrollo en la Ingeniería de Software estaríamos ubicados en la fase inicial del proyecto, realizando el flujo de trabajo de Modelación del Negocio en las siguientes páginas quedarán recogidos los artefactos propios de esta etapa.

El propósito del Modelo de Negocio es ayudar a lograr una mejor comprensión del problema que su software tiene que resolver. De hecho, los requerimientos para la aplicación pueden ser derivados a partir de este.

De la calidad , especificidad, y profesionalidad con que sea vencida esta fase dependerán muchos de los resultados a obtener; el paso a etapas y fases posteriores estarán estrechamente relacionadas con la información que se registre en este capítulo, es por ello la importancia real del mismo, ya que mucho proyectos en nuestra industria que suelen fracasar es debido a errores cometidos por los desarrolladores del mismo, en especial por los analistas de sistemas, en la concepción propia del producto.

3.1 DESCRIPCIÓN DEL PROCESO DE NEGOCIO ACTUAL

Nuestro negocio está enmarcado en las actividades relacionadas con los servicios primarios de salud que se prestan en todo el país, en particular con los servicios que involucran a la Dirección Nacional de Estadística y Registros Médicos y a la Dirección de Atención Primaria de Salud .

Los procesos actuales de recopilación, transformación y emisión de información basada en el Registro de Actividades Diarias (Hoja de Cargo) del Equipo Básico de Salud (EBS) comienza en las consultas de los hasta hoy llamados Consultorios del Médico de Familia (CMF), ahora área de trabajo o sede de los Equipos Básicos de Salud,

de todo el Sistema Nacional de Salud, allí los miembros del Equipo Básico de Salud, entiéndase médico y enfermera(o), toman los datos referentes a los pacientes y los que se derivan de la consulta del mismo, dichos datos están normalizados en la Hoja de Cargo del Equipo Básico de Salud, que no es más que el registro de todas la actividades diarias que realiza dicho equipo en su área de trabajo; estos Registros de Actividades Diarias(RAD) son almacenados hasta que pueden ser recogidos por un “mensajero” de la sede del Área de Salud(AS) a la cual pertenece el Equipo Básico de Salud, estos registros son entregados en el Departamento de Estadística del Área de Salud al Técnico de Estadística del Grupo Básico de Trabajo (GBT) (en los casos que los Grupo Básico de Trabajo al cual pertenezca el Equipo Básico de Salud tengan a este técnico) o del Área de Salud, el cual termina de llenar la Hoja de Cargo con los códigos CIE-10(de la Clasificación Internacional de Enfermedades y Problemas relacionados con la salud, 10ma edición) de cada problema de salud presentado por los pacientes y registrados allí por el Equipo Básico de Salud, terminando de esta manera el proceso de gestión de los Registros de Actividades Diarias.

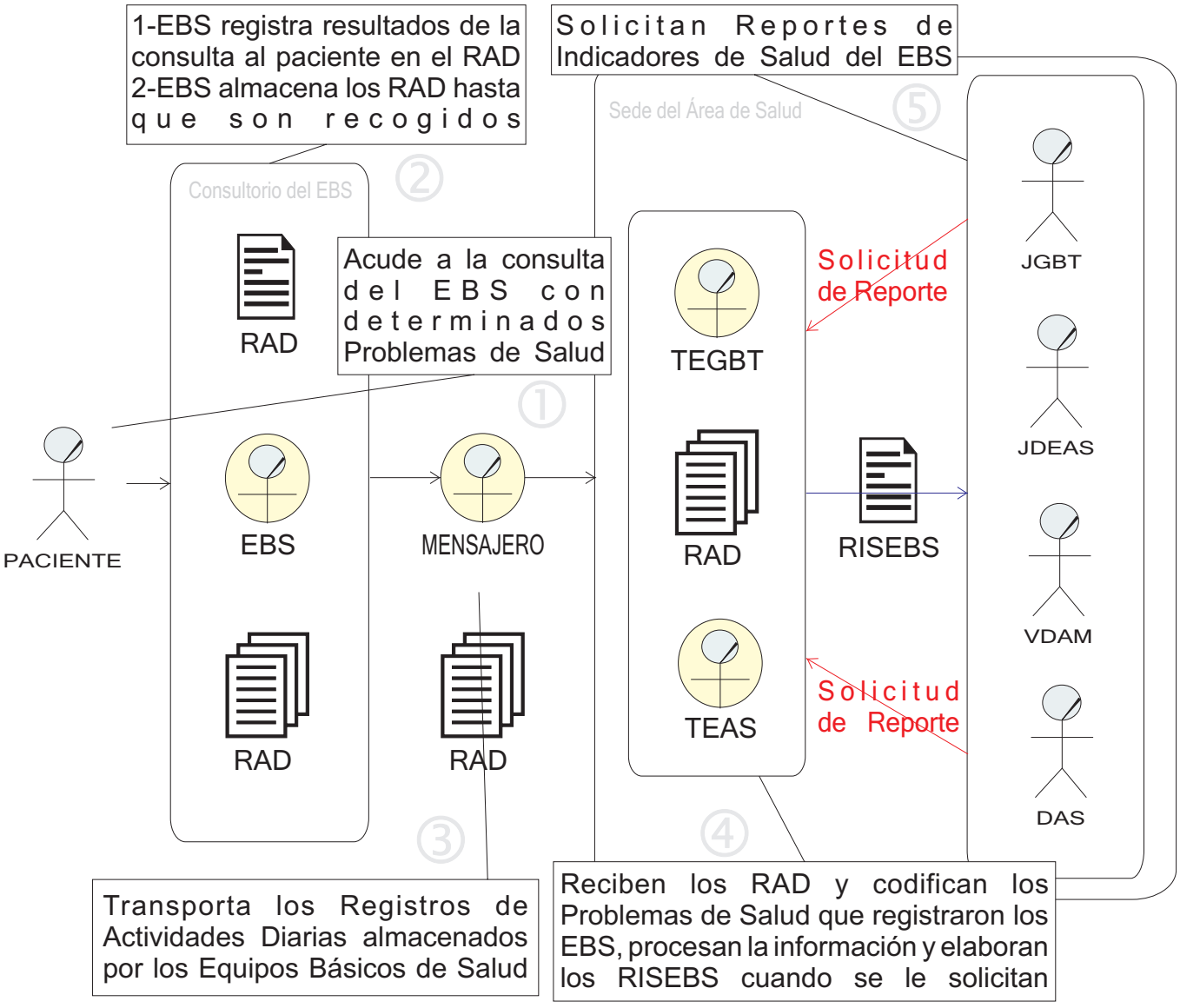
Por otro lado en el Departamento de Estadística del Área de Salud los técnicos de estadística procesan los datos registrados en los Registros de Actividades Diarias, según unos indicadores con interés desde el punto de vista estadístico e investigativo, de forma manual o semi automática, cuando le son solicitados por los directivos del Grupo Básico de Trabajo y el Área de Salud, elaboran los reportes correspondientes y entregan a sus superiores los informes de los llamados Reportes de Indicadores de Salud del Equipo Básico de Salud (RISEBS) para su aprobación y para su utilización en las reuniones del grupo de trabajo o para ser enviada a otros niveles con interés en los mismos.

Como es fácil observar, los procesos antes descritos son muy vulnerables a los errores del trabajo humano, pueden ser equivocados los datos iniciales del paciente, la denominación que cada médico le da a un padecimiento o a una conducta a seguir, haciendo que los técnicos deban deban tomar decisiones al respecto a la hora de codificar los problemas de salud que presentaron los pacientes y fueron registrados en la Hoja de Cargo por los Equipos Básico de Salud, estás decisiones pueden estar equivocadas o alejadas del diagnóstico real, haciendo que al final los informes que se emitan tengan una alta probabilidad de estar plagados de datos inconsistentes o erróneos, trayendo como consecuencia que el organismo rector del Sistema Nacional de Salud, el Ministerio de Salud Pública, pueda que tome medidas que no tengan nada que ver con la situación real de una población. Además, los procesos completados pueden tornarse demasiado lentos para la rapidez de acción que requieren algunas manifestaciones de salud de alguna determinada comunidad, por lo engorroso que puede convertirse este proceso de interpretación, clasificación y codificación de los problemas de salud de cada paciente registrado, así como las dificultades propias que puede ocasionar el almacenamiento y transportación de los Registros de Actividades Diarias hasta los lugares donde van a ser procesados finalmente.

Es por ello que la Dirección de Registros Médicos y Estadísticas y la dirección de Atención Primaria de Salud(APS) dieron la tarea de analizar a detalle el negocio que se dedica al Registro de las Actividades Diarias (RAD) de los Equipo Básico de Salud y a la posterior elaboración de reportes basados en la información registrada en los Registro de las Actividades Diarias, con el fin de entregarles una propuesta de cambio, en los procesos actuales, que garantice la mayor veracidad de los datos con los cuales se trabaja y un tiempo de respuesta mucho menor al actual a la hora

de elaborar los Reportes de Indicadores de Salud del Equipo Básico de Salud. Además que correspondan con los intereses de Atención Primaria de Salud de modernizar e integrar todos sus servicios en un sistema de información automatizado.

En la descripción anterior hemos definido dos procesos que ocurren con los Registros de Actividades Diarias, el primero dedicado a la creación de los mismos y el segundo enfocado a la generación de información estadística a partir de la información que se registra en la Hoja de Cargo del Equipo Básico de Salud(RAD).



Leyenda:
- EBS: Equipo Básico de Salud.
- JGBT: Jefe de Grupo Básico de Trabajo.
- JDEAS: Jefe de Departamento de Estadística del Área de Salud.
- VDAM: Vice-Director de Asistencia Médica del Área de Salud.
- DAS: Director del Área de Salud.
- TEGBT: Técnico de Estadística del Grupo Básico de Trabajo.
- TEAS: Técnico de Estadística del Área de Salud

Fig.3.1 Esquema del proceso actual del negocio.

3.2 MODELACIÓN DEL NEGOCIO

Basados en la información anterior se pretende modelar a detalle el negocio con el fin de asegurarnos de que los clientes, usuarios finales y desarrolladores tienen una idea común de la organización, para al final derivar en los requerimientos que el sistema va a tener.

Luego de varias entrevistas con especialistas de Atención Primaria de Salud del MINSAP y de la Dirección de Registros Médicos y Estadísticas, de la misma institución, y estudios de mesa efectuados por el equipo de desarrolladores, se obtuvo un listado de actores y trabajadores del negocio, con los cuales comenzaremos a modelar todo lo referente a estos procesos.

La definición de los roles de cada actor o trabajador es una acción primordial; delimitar a los interesados en que los procesos ocurran y a los que explotarán los recursos que pondremos a su disposición contribuye a un mejor entendimiento de la organización donde ocurre el negocio, a la posterior definición de configuraciones posibles del sistema a implementar, así como para la asignación de permisos y accesos a este.

A continuación, serán identificados y definidos cada uno de los actores que participan en el negocio. En la tabla 3.1 presentamos a los actores del negocio, explicamos brevemente la justificación y el rol que desempeñan en la organización, utilizamos además los diagramas UML que le identifican, el uso de siglas para nombrarlos se debe a la relativa extensión de sus identificadores.

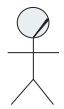
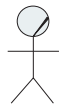
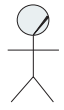
ACTOR DEL NEGOCIO	JUSTIFICACIÓN
1  PACIENTE	Es el principal beneficiado de los resultados finales del proceso, ya que cualquier medida que se tome al respecto lo afecta directamente a él, sin su existencia no existiría proceso alguno.
2  JGBT	Se beneficia de los resultados del proceso, mediante el mismo puede controlar las actividades de sus subordinados e informar, por norma o a solicitud, a los niveles superiores.
3  JDEAS	Se beneficia de los resultados del proceso, mediante el mismo puede controlar las actividades de sus subordinados e informar, por norma o a solicitud, a los niveles superiores y a los miembros del Consejo de Dirección de su Área de Salud.
4  VDAM	Se beneficia de los resultados del proceso, mediante el mismo puede controlar las actividades de sus subordinados e informar, por norma o a solicitud, a los niveles superiores.
5  DAS	Se beneficia de los resultados del proceso, mediante el mismo puede controlar las actividades de sus subordinados e informar, por norma o a solicitud, a los niveles superiores.

Tabla 3.1 Actores del negocio.

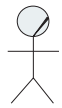
ACTOR DEL NEGOCIO	JUSTIFICACIÓN
6  DIA	DIA es el personificación del tiempo, el cual es un actor del negocio; el negocio utiliza este actor para verificar hasta cuando puede existir un RAD y a partir de cuando se puede crear una nueva entidad del negocio del tipo RAD.
7  DIA DE TRABAJO	Es la generalización que se forma con la unión de DIA y PACIENTE, un RAD se crea partir de que llega el primer paciente a la consulta, hasta que llega el último durante un mismo día de trabajo para un EBS (0000 horas hasta 2359 horas).
8  TIEMPO	Es un actor de negocio abstracto que cumple el rol de indicar cuando deben llevarse, recogerse, enviarse o transportarse los RAD, que han sido elaborados por los EBS, hacia el Departamento de Estadística del Área de Salud.

Tabla 3.1 Actores del negocio. (CONTINUACIÓN...)

Como se puede leer la mayoría de los actores del negocio tienen un patrón de comportamiento bastante parecido con respecto al negocio, todos excepto PACIENTE, que tiene un papel bien definido en el primer proceso, el de captura de datos de la consulta, DIA y DIA DE TRABAJO; todos los demás actores del negocio, lo solo necesitan los resultados elaborados de la información que se almacena en los Registros de las Actividades Diarias de los médicos de el Área de Salud donde estos actores pertenecen, resultados que se obtienen gracias a la ejecución del segundo proceso referente a este negocio.

Ahora serán identificados y definidos cada uno de los trabajadores que participan en el negocio. En la tabla 3.2 presentamos a los trabajadores del negocio, e igualmente como en el caso anterior explicamos brevemente la justificación y el rol que desempeñan en la organización, utilizamos también los diagramas UML que le identifican respectivamente, el uso de siglas para nombrarlos se debe a la relativa extensión de sus identificadores, aunque en la leyenda del cuadro se encuentran las aclaraciones pertinentes.

Estos trabajadores son “candidatos”muy significativos para convertirse en los actores del sistema, para que esto no suceda de esta manera la solución que se proponga debe justificar la razón o las razones que provocaron dicha exclusión, la realidad apunta a que en el proceso de automatización de los procesos de un negocio algunos roles de trabajadores del negocio serán sustituidos por el sistema que se desarrolle, lo que no significa que el trabajador u operario que realiza ese papel quedará excedente en su organización.

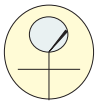
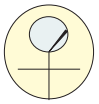
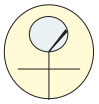
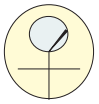
TRABAJADOR DEL NEGOCIO	JUSTIFICACIÓN
1  EBS	Es el responsable de registrar correctamente toda la información normalizada en el Registro de Actividades Diarias con los datos derivados de la consulta a cada paciente y de almacenar los RAD hasta que son recogidos; la actividad le sirve como prueba de sus labores y acciones en su área de trabajo.
2  TEGBT	Es el que recibe inicialmente los RAD de los EBS que pertenecen a su grupo, procesa la información registrada en ellos y elabora reportes e informes con interés estadístico, investigativo y de auditoría para los distintos niveles que se lo solicitan, siempre y cuando estos tengan la aprobación de los jefes de su nivel.
3  TEAS	Es el que recibe inicialmente los RAD de los EBS de los GBT que no cuentan todavía con TEGBT, procesa la información registrada en ellos y elabora reportes e informes con interés estadístico, investigativo y de auditoría para los distintos niveles que se lo solicitan, siempre y cuando estos tengan la aprobación de los jefes de su nivel, haciendo cierres con la información procesada por él y con la que le entregan los distintos TEGBT pertenecientes a su AS.
4  MENSAJERO	Es el encargado de recoger los RAD de los EBS y llevarlos a la sede del Área de Salud y entregarlos al trabajador correspondiente que va a procesarlos, TEGBT, o TEAS. En ocasiones algún miembro del EBS desempeña este rol.

Tabla 3.2 Trabajadores del negocio.

Correspondiendo con lo que se ha dicho hasta ahora, se ha identificado que el negocio que se ha estudiado hasta ahora se mueve entorno a tres procesos bien definidos, independientes entre sí, pero a la vez dependientes el uno del otro.

Estos procesos son:

- El primero es el que garantiza el registro de los datos referentes a la consultas y su actualización en caso de que se haya cometido algún error en la introducción de datos y transporta dichos registros al lugar indicado por la administración del Área de Salud.

- El segundo es el que se refiere a la actividad de procesar la información y presentar, en un formato bien definido, los datos que se toman de los Registros de Actividades Diarias .

- El tercero es que se encarga de aglutinar todos los RAD almacenados por los EBS y llevarlos hasta el Departamento de Estadística del Área de Salud y entregarlos al técnico correspondiente.

Para un mejor entendimiento formularemos los procesos a manera de casos de uso relacionados cada uno con los actores y trabajadores del negocio que intervienen en cada uno de ellos.

3.2.1 Proceso de Gestión del Registro de Actividades Diarias del Equipo Básico de Salud.

El proceso o caso de uso Gestionar Registro de Actividades Diarias se centra en las actividades que realizan los miembros del Equipo Básico de Salud en la consulta o en en su labor en el terreno de la comunidad a la que que estos le prestan atención o sea su radio de acción, estas actividades como ya habíamos mencionado en la descripción del proceso generan una serie de datos, cuya información tienen una gran relevancia desde el punto de vista estadístico e investigativo, por ello la necesidad de registrarla en un soporte normalizado, la Hoja de Cargo, y con persistencia en el tiempo, en este caso, un registro en papel.

El diagrama que modela este caso de uso del negocio se presenta a continuación, es importante observar y comprender la generalización que representa al actor que inicia este proceso.

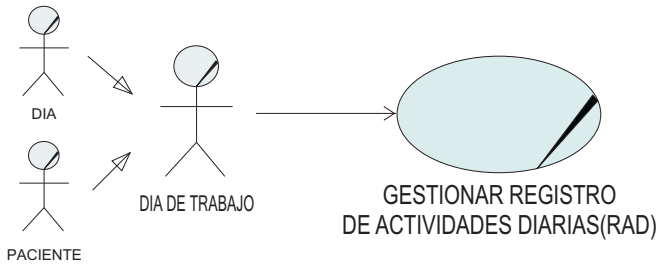


Fig.3.2 Diagrama de caso de uso del negocio # 1.

En el diagrama UML de caso de uso del negocio de la figura 3.2, el actor “Dia de Trabajo” inicia el proceso o caso de uso Gestionar Registro de Actividades Diarias (RAD); este actor es una abstracción conformada por la relación directa entre los actores del negocio PACIENTE y DIA, que intenta modelar el rol que activa el comienzo y el fin de la gestión del Registro de Actividades Diarias por día. El caso de uso inicia cada dia con la llegada del primer paciente a la consulta del EBS y termina coincidentemente con la culminación del día (23:59:59 horas).

CASO DE USO DEL NEGOCIO # 1	Gestionar Registro de Actividades Diarias(RAD)
ACTORES	Día de Trabajo, DIA, Paciente
PROPÓSITO	Registrar la información correspondiente a la Hoja de Cargo del EBS
RESUMEN El Caso de Uso se inicia cuando se abre diariamente la consulta del Equipo Básico de Salud y llega el primer paciente del día a la consulta, esta acción crea el Registro de Actividades Diarias de ese Equipo Básico de Salud para ese día , luego continua cuando el Equipo Básico de Salud desarrolla un grupo de actividades con el paciente, las cuales generan datos que le sirven para ir conformando y completando el Registro de Actividades Diarias de ese día, para ello se apoya en el Libro de Historia de Salud Familiar(HSF), el Registro de Actividades Diarias se va llenando mientras sigan llegando pacientes a la consulta durante ese día; el caso de uso culmina cuando todos los campos del Registro de Actividades Diarias, osea, todos datos referentes a las consultas de los pacientes quedan registrados en el RAD, y culmina el día de trabajo del EBS .	
ACCIÓN DE ACTOR	RESPUESTA DEL NEGOCIO
1-Llega 1er paciente a la consulta del EBS, ese día. 5-El paciente da sus datos personales(pasa al paso 4). 9-Se Marcha paciente de la consulta del EBS. 10-Llega otro paciente a la consulta.	2-Comienza la jornada laborar del EBS, se crea el RAD para ese día con los datos referentes a la ubicación del EBS y los propios del EBS. 3-El EBS comprueba si el paciente que ha llegado pertenece a la población a él asignada. a) Si pertenece a su población, busca los datos personales del paciente en la Historia de Salud Familiar(HSF)(pasa al paso 4). b)En caso negativo le pregunta al paciente directamente por sus datos personales(pasa al paso 5). 4-Se registran los datos personales del paciente en el RAD(pasa el paso 6). 6- Determina problema de salud del paciente. 7-Registran el problema de salud en el RAD. 8-Registra todos los demás datos referentes a la consulta de el paciente en el RAD relacionados con la consulta y el problema de salud que presentó el paciente. 11-Comprueba si no ha terminado el “Dia de Trabajo” a) Si el “Dia de Trabajo” ha concluido, culmina caso de uso (pasa al paso 1). b)Si el “Dia de Trabajo” no ha concluido (pasa al paso 3).
PRIORIDAD	Responde al principal objetivo de automatización en el RAD, al resolver gran parte de los problemas actuales.
MEJORAS	☑ Se automatizará el proceso de introducción de datos en el RAD. ☑ Se garantizará la confiabilidad de la informacion introducida en el RAD al disminuir los errores en la entrada de datos. ☑ Se garantizará un control más eficiente de las labores d el EBS.
OTRAS SECCIONES	

Tabla 3.3 Especificación del caso de uso del negocio # 1.

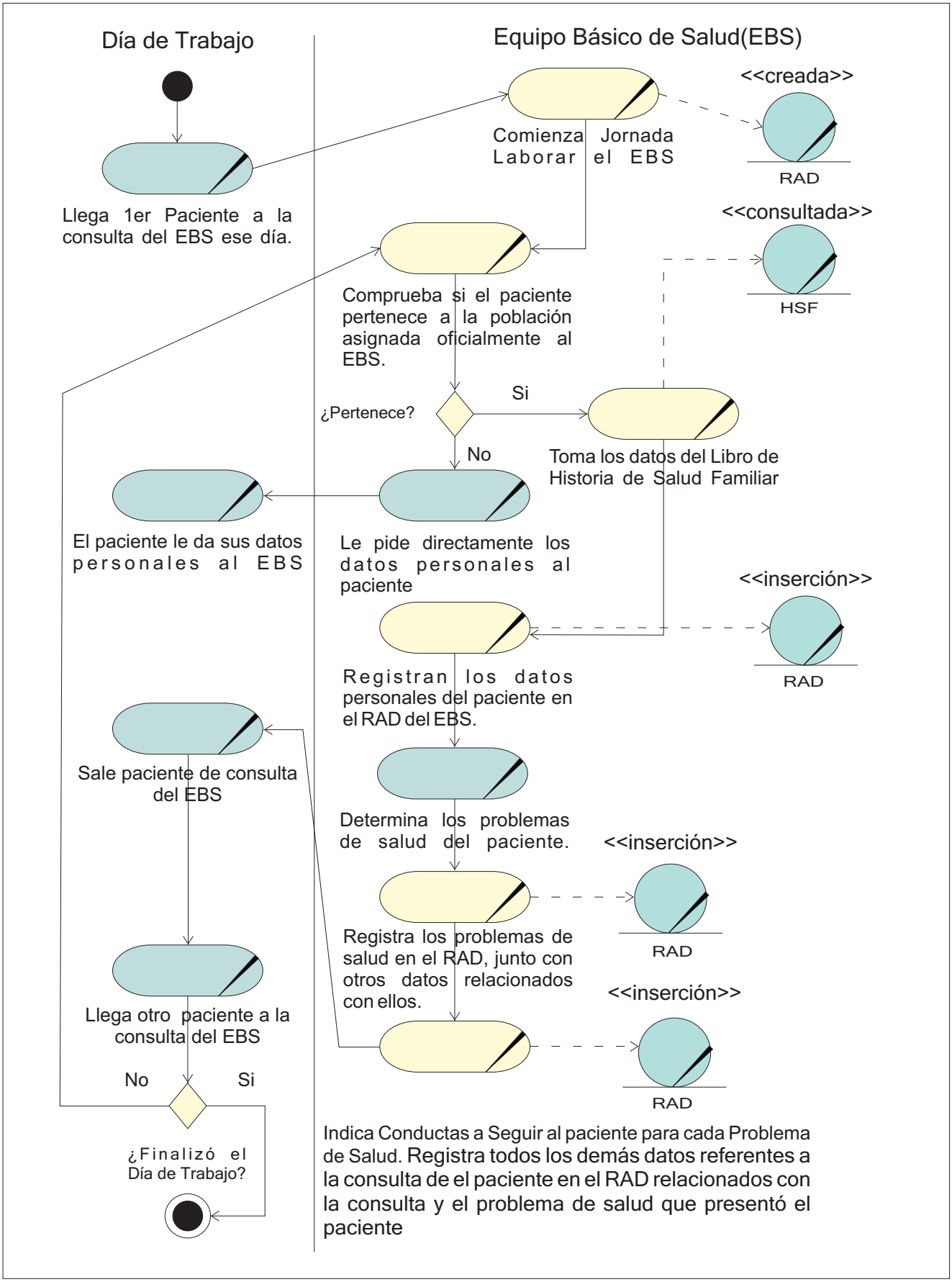


Fig.3.3 Diagrama de Actividades del caso de uso del negocio # 1.

3.2.2 Proceso de Elaboración de Información Estadística a partir de los Datos del RAD.

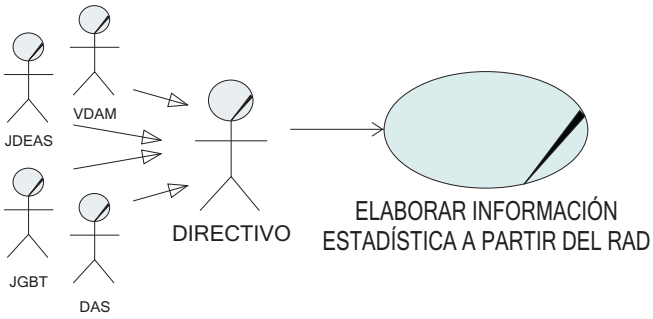


Fig.3.4 Diagrama de caso de uso del negocio # 2.

En este caso de uso (figura 3.3) presentamos las generalización de los actores de negocio JDEAS, JGBT, VDAM, DAS en el actor de negocio DIRECTIVO, este actor representa a todo el personal con autoridad en el Área de Salud para pedirle a los técnicos, que laboran en el Departamento de Estadística de dicha AS, la elaboración de información, estadísticamente procesada, en forma de reporte a partir de la información que se registra por los EBS en el Registro de Actividades Diarias de cada EBS.

CASO DE USO DEL NEGOCIO # 2	Elaborar Información Estadística a partir de RAD.
ACTORES	JGBT, JDEAS, VDAM, DAM
PROPÓSITO	Elaborar informes a partir de las Hojas de Cargo del EBS almacenadas.
RESUMEN El Caso de Uso se inicia cuando cualquiera de los actores solicita los reportes de los indicadores de salud del Equipo Básico de Salud(RISEBS) para saber el comportamiento de los mismos en las poblaciones del Equipo Básico de Salud(EBS), el Grupo Básico de Trabajo(GBT) o en el Área de Salud(AS), según sea el caso de quién lo solicite; culmina cuando el actor que lo solicita recibe los reportes con los indicadores que el solicitó.	
ACCIÓN DE ACTOR	RESPUESTA DEL NEGOCIO
1-Solicita informes de RISEBS. 6-Recibe y aprueba RISEBS.	2-El Técnico de Estadística del Área de Salud o del GBT (TEAS, TEGBT) busca los datos necesarios en los RAD. 3-TEAS ó TEGBT procesa la información obtenida de los RAD almacenados. 4- Elabora Reportes de Indicadores de Salud del EBS(RISEBS). 5-Entrega los RISEBS a sus superiores.
PRIORIDAD	Responde al principal objetivo de automatización en el RAD, al resolver gran parte de los problemas actuales.
MEJORAS	☑ Se automatizará el proceso de elaboración de reportes. ☑ Se garantizará la confiabilidad de la informacion obtenida y presentada en los reportes. ☑ Disminuirá el tiempo para la obtención de estos informes de salud
OTRAS SECCIONES	

Tabla 3.4 Especificación del caso de uso del negocio # 2.

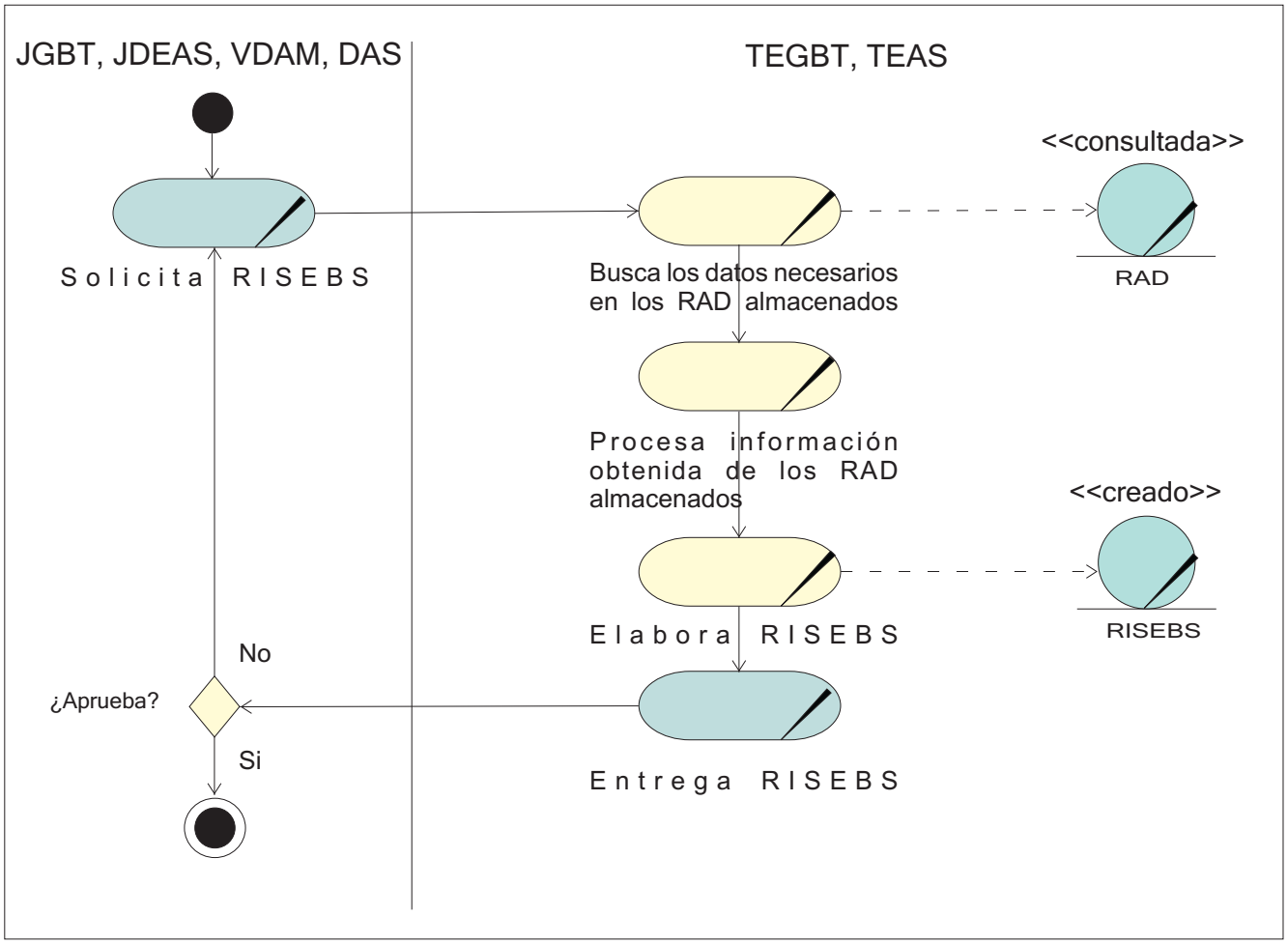


Fig.3.5 Diagrama de Actividades del caso de uso del negocio # 2.

3.2.3 Proceso de Transportación de los RAD al Departamento de Estadística del Área de Salud.

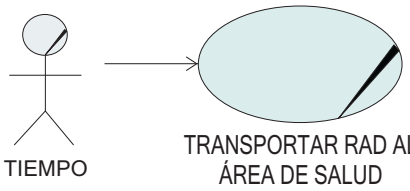


Fig.3.6 Diagrama de caso de uso del negocio # 3.

En el caso de uso del negocio de la figura 3.6 se ve como el actor abstracto “TIEMPO” inicializa el caso de uso señalado en el diagrama; el tiempo corresponde en este caso a una norma, pre-establecida o no, para el cumplimiento o el acometimiento de una determinada labor en el negocio de la organización, en este caso la labor es la búsqueda de los RAD que están almacenados por los EBS, para ser entregados en el lugar donde van ser procesados finalmente.

Este será el último caso de uso que describiremos, en relación con los tres procesos que identificamos en la organización y que se corresponden con la situación problémica planteada y que con este trabajo pretendemos encontrarle una solución viable y económica. Luego tendremos información suficiente para desarrollar un modelo de objetos primario que nos permita abrirnos camino para el anális, diseño y la posterior implementación de la solución objetivo de este trabajo.

CASO DE USO DEL NEGOCIO # 2	Proceso de Transportación de los RAD al Departamento de Estadística del Área de Salud.
ACTORES	MENSAJERO
PROPÓSITO	Entregar los RAD a los técnicos que van a procesar la información registrada en ellos.
RESUMEN El Caso de Uso se inicia cuando se cumple el lapso de tiempo que hay para recoger los RAD almacenados por los EBS del Área de Salud; culmina cuando los RAD son entregados en el Departamento de Estadística del Área de Salud	
ACCIÓN DE ACTOR	RESPUESTA DEL NEGOCIO
1-Llega día establecido para la recogida de los RAD del Área de Salud .	2-.Recoge los RAD almacenados por los EBS del Área de Salud. 3-Transporta los RAD recogidos en la sede de trabajo del EBS. 4-Entrega RAD a los técnicos de estadísticas del GBT o del AS en el Departamento de Estadística del Área de Salud. 5-Culmina proceso.
PRIORIDAD	Corresponde con las exigencias de operabilidad y disponibilidad del sistema.
MEJORAS	☑ Se automatizará el proceso de recogida, transportación y entrega de los RAD, ayudando a disminuir el tiempo para la obtención de resultados.
OTRAS SECCIONES	

Tabla 3.5 Especificación del caso de uso del negocio # 3.

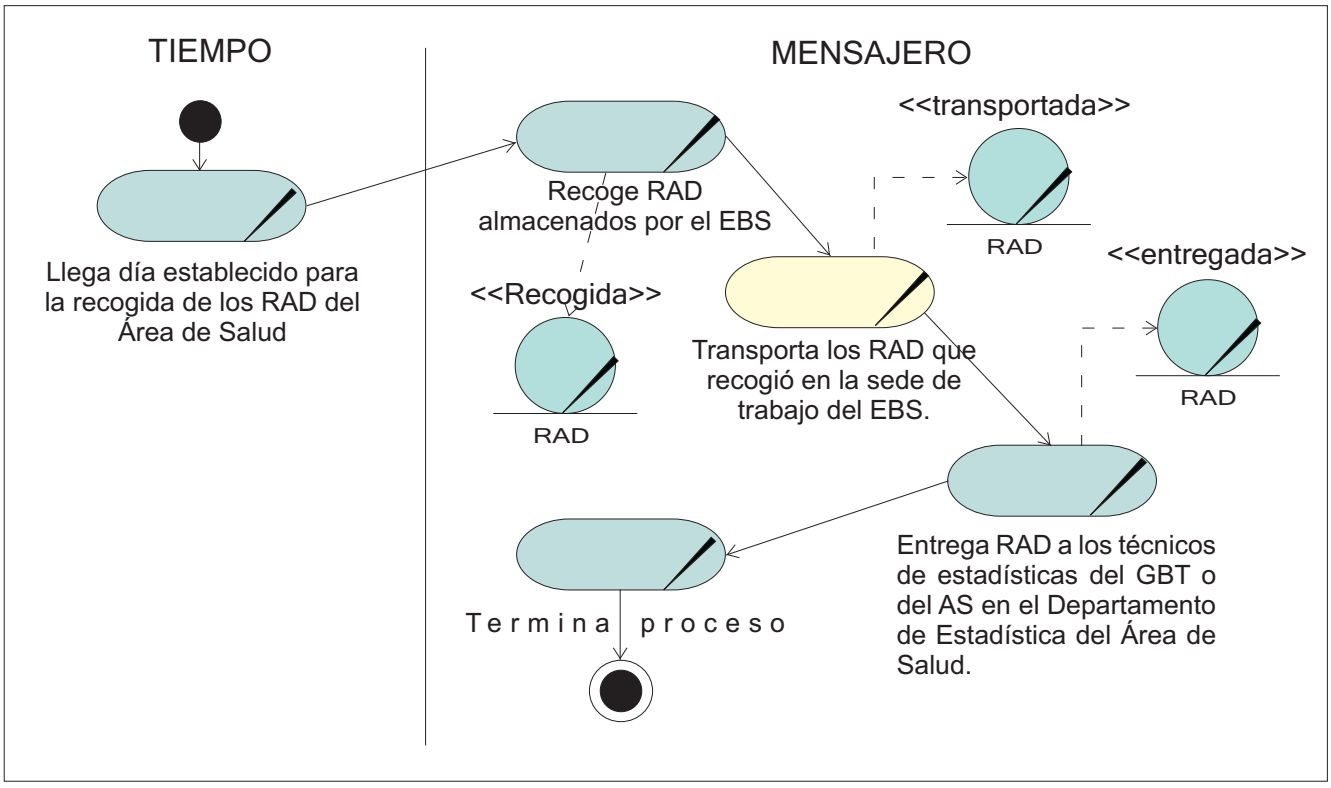


Fig.3.7 Diagrama de Actividades del caso de uso del negocio # 2.

3.2.4 Modelo de Objetos del Negocio.

De los diagramas anteriores podemos extraer tres entidades (Registro de Actividades Diarias(RAD), Reporte de Indicadores de Salud del Equipo Básico de Salud (RISEBS) e Historia de Salud Familiar (HSF)); las mismas junto con los trabajadores del negocio que las crean, modifican o consultan servirán de base para crear el modelo de objetos inicial, con el objetivo de entender, desde el punto de vista que necesitamos los desarrolladores, las áreas de interés que pudieran ser objeto de automatización, si así se determinara cuando se llegue a los flujos de trabajo de análisis y diseño correspondientes a la etapa(fase) de Elaboración de un producto informático, según la metodología de desarrollo de software RUP+UWE.

Cabe agregar que aunque presentaremos la Historia de Salud Familiar (HSF) como una clase entidad en nuestro modelo de objetos del negocio no será de nuestro interés abundar en ella, ya que pertenece la misma a otro módulo en desarrollo en este proyecto, el cual utilizaremos como un sistema externo al cual solicitaremos información importante para la funcionalidad del nuestro.

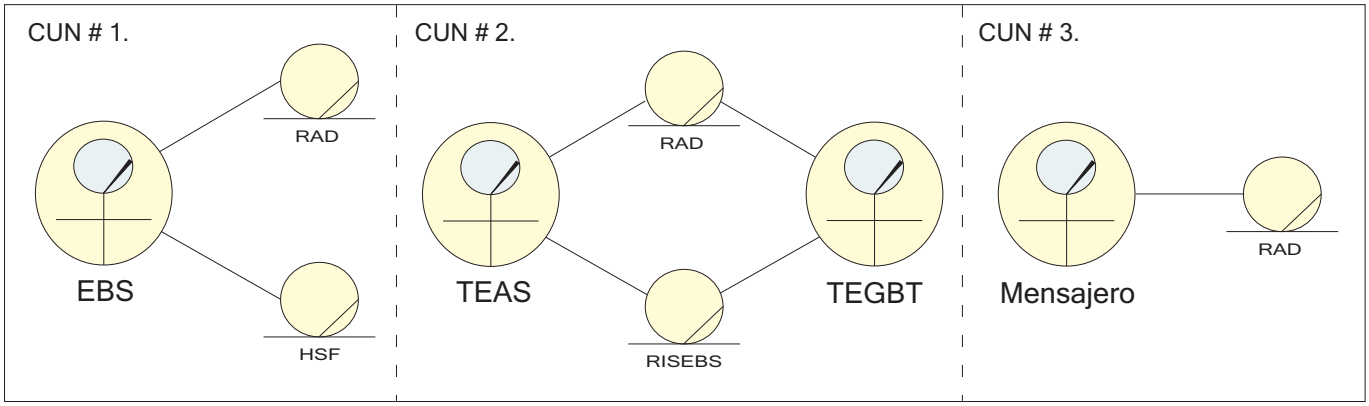


Fig.3.8 Modelo de Objetos derivado de los Casos de Uso del Negocio.

3.3 DESCRIPCIÓN DE LOS PROCESOS DE NEGOCIO PROPUESTOS.

Tras el análisis exhaustivo de los procesos en el negocio actual con los interesados en el mismo y su modelación con estándares de la industria del software (UML), el equipo de desarrollo pudo detectar las áreas de acción principales para este proyecto de innovación tecnológica y transición de tecnologías; la mayoría de los criterios versaron sobre los cambios que ocurrirían en los procesos actuales de recopilación, transformación y emisión de información basada en el Registro de Actividades Diarias (Hoja de Cargo) del Equipo Básico de Salud partir de la informatización del mismo.

En la propuesta del proceso del negocio los miembros del Equipo Básico de Salud (EBS) podrán registrar datos en el Registro de Actividades Diarias (RAD) “automatizado”, dependiendo solo de si cuentan o no con las tecnologías necesarias para que un sistema informático como el que pretendemos entregar funcione; el sistema sería el encargado de procesar la información y elaborar los Reportes de Indicadores de Salud del Equipo Básico de Salud, de forma personalizada [tarea que antes realizaba el Técnico de Estadística del Área de Salud o del Grupo Básico de Trabajo, según sea el caso, de forma manual o semi-automática] y por última, también el

Sistema sería el encargado de emitir los reportes correspondientes por las formas que finalmente se decida o exija para el cumplimiento de los requerimientos de los clientes; vale aclarar que todas esas tareas que realizaría el Sistema serían a partir de peticiones exteriores de un determinado usuario del mismo, otro cambio en el proceso es que también sería el Sistema el encargado de almacenar y transportar los Registros de Actividades Diarias hasta los lugares correspondientes o designados en la arquitectura de la Aplicación.

A continuación mostramos un diagrama que representa el proceso de negocio propuesto, aunque el Sistema Informático está representado por una computadora, el producto que vamos a entregar está lejos de ser una común aplicación de Escritorio.

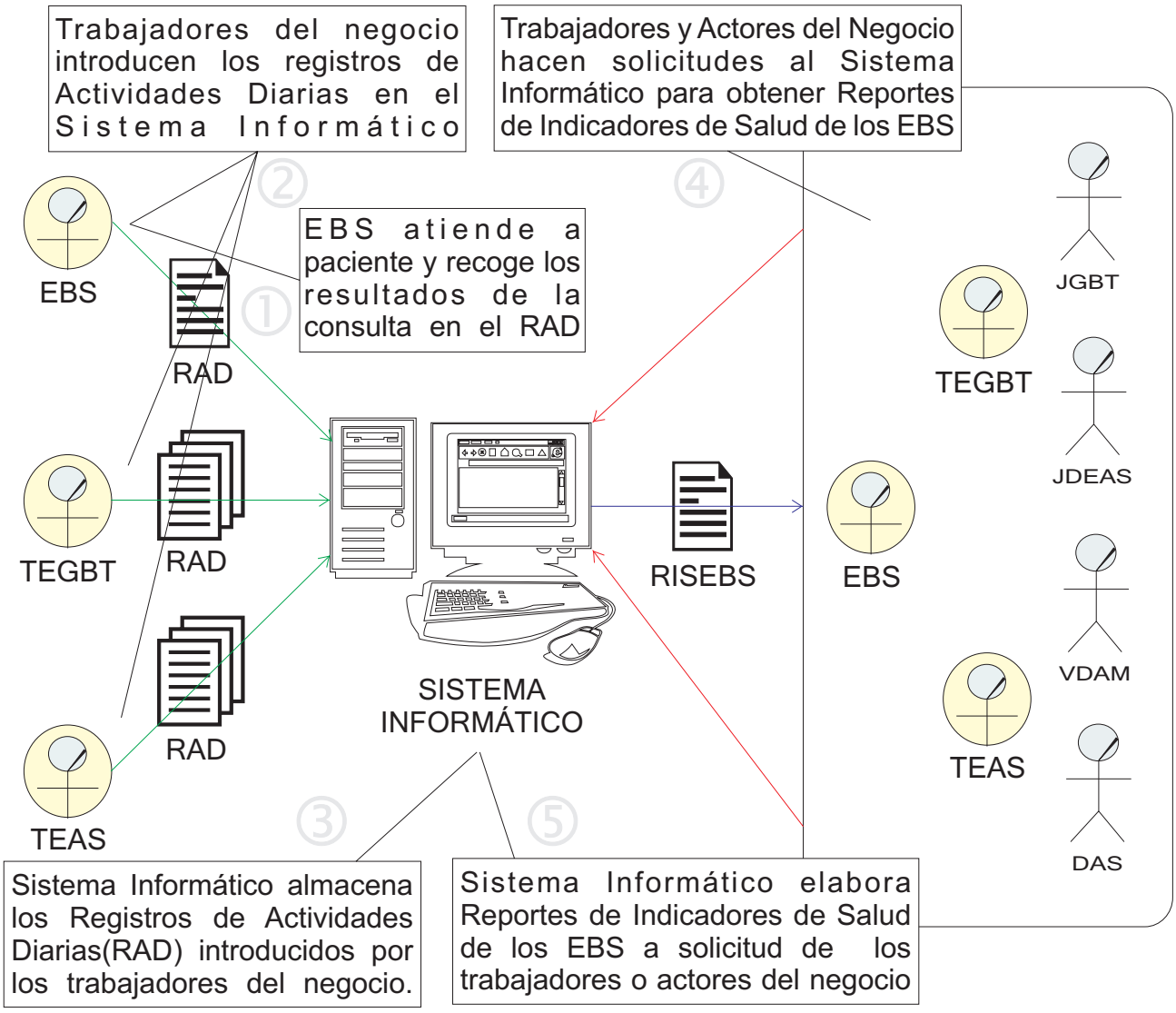


Fig.3.9 Esquema del proceso propuesto del negocio.

Teniendo en cuenta la propuesta del proceso de negocio que se presentó, se determinó con los clientes las reglas que debe cumplir el sistema que se desarrolle; la idea es capturar todas las inquietudes y expectativas que estos tengan y satisfacer todas sus necesidades y sobre todo conocer de nuestra parte las regulaciones o los estándares que debe cumplir la solución que se pretende; estas reglas luego

las podremos clasificar, derivándose de esta manera en los requerimientos generales y específicos por los cuales debe regirse nuestro trabajo y todas las actividades posteriores en el proceso de ingeniería.

El proceso de captura de las reglas de negocio comienza desde el inicio de la modelación del negocio y continúa durante cada entrevista de trabajo con los clientes y usuarios.

3.4 REGLAS DEL NEGOCIO.

En relación con los procesos de negocio actual y propuesto, siguiendo las normalizaciones establecidas por la Dirección de Estadística y Registros Médicos del MINSAP en la Aplicación deben contemplarse estos criterios y normas del cliente.

En los Registros de Actividades Diarias quedarán almacenados, por día, los datos de las consultas a los pacientes que son atendidos por el EBS; de cada consulta quedarán plasmados en el RAD, el número de historia clínica, los nombres y apellidos del paciente, su edad, el sexo, el lugar de la consulta (terreno, consulta), los problemas de salud que lo aqueja y el código de cada problema (ambos extraídos del CIE-10); para cada problema de salud presentado por el paciente se registrará también el tipo de actividad (1ra consulta, seguimiento médico, seguimiento de enfremería, alta), el tipo de diagnóstico (presuntivo, confirmado), si es una incidencia para el paciente ese problema de salud y las conductas a seguir o accionares; cada Registro de Actividades Diarias estará encabezado por los datos del Equipo Básico de Salud y de la ubicación del consultorio de dicho equipo, que es la cadena formada por el número de Área de Salud, el número del Grupo Básico de Trabajo y el número del Equipo Básico de Salud. Los registros solo podrán ser agregados o actualizados en el término de las 24 horas que dura el día pero nunca borrados y las trazas de su estado anterior deben ser almacenadas durante un tiempo determinado. Cabe agregar que un paciente puede ser consultado un mismo día en varias ocasiones y en cada consulta puede presentar más de un problema de salud, por lo que se le indicarán varias conductas a seguir por cada problema de salud presentadado; también es importante saber que, aunque oficialmente los EBS están formados por un médico y una enfermera en casos excepcionales esta estructura puede romperse y conformarse por más de una enfermera y más de un médico, este último caso ocurre solamente cuando al medico del EBS se le han asignado algunos estudiantes de medicina(internos), en estos casos en el Registros de Actividades Diarias se deberá reflejarse todo el personal de salud que estubo presente ese día en la consulta sin detallarse cual médico realizó la consulta o que enfermera atendió a determinado paciente. Además los Registros de Actividades Diarias pueden ser llenados por el médico o por la enfermera y también por los técnicos de estadística correspondientes, en el caso de la enfermera solo podrá hacerlo si el tipo de actividad para el problema de salud es un seguimiento de enfermería, osea un tratamiento ambulatorio que se le está efectuando en la consulta a un paciente.

Inicialmente los RAD podrán ser introducidos en el sistema instantáneamente por los miembros del EBS o de forma posterior por los técnicos de estadísticas del GBT o del AS, según sea la disponibilidad de Tecnologías para el Tratamiento de la Información en Atención Primaria de Salud. Decimos inicialmente porque en un futuro se espera que sean solo los miembros del EBS sean quienes introduzcan los datos en el RAD y solo en caso excepcionales los técnicos de estadísticas del GBT o del AS.

Cuando un paciente va por un problema de salud que resulta ser una enfermedad de declaración obligatoria(EDO) confirmada o presuntiva entonces el EBS debe llenar instantáneamente un Registro Oficial de EDO siempre que no sea una consulta de seguimiento médico, seguimiento de enfermería o alta médica.

Las conductas a seguir o accionar y los indicadores de salud para reportes son entidades fijas que son determinados por entidades superiores del MINSAP por lo que no están sujetos a cambios o adaptaciones propios que quieran hacer, a voluntad, los miembros del EBS o algún técnico de estadísticas del GBT o del AS.

Los reportes deberán ser configurables por los usuarios y brindar toda la información o la mayor posible a partir de los datos que se registran en el RAD.

Cada usuario que esté trabajando con la aplicación solo podrán trabajar con la información autorizada para su nivel de acceso, por lo que se deberá garantizar la seguridad y confiabilidad de los datos y del software como tal.

Con los datos anteriores, unido a los formatos y estándares de trabajo de la empresa que nos “contrató” para realizar la modelación de este producto, y guiados por los lineamientos dictados en la política de informatización de la sociedad cubana tenemos elementos suficientes para formular organizadamente los requerimientos que deberán cumplirse, para ello los clasificaremos en requisitos funcionales y no funcionales los cuales serán las líenas guías por las cuales deberá regirse nuestro proyecto de desarrollo de un producto informático y nuestra labor como analistas de sistemas. Esta actividad pertenece al flujo de trabajo de requisitos o requerimientos como también se le conoce y es la etapa de transición entre los flujos de trabajo de modelación del negocio y el de análisis, incluídos todos en las fases de Concepción y Elaboración de RUP.

3.5 REQUERIMIENTOS DEL SISTEMA.

Hasta ahora hemos hablado acerca la organización donde se implantará el producto fruto de nuestra labor, y ya tenemos datos suficientes para introducirnos a detallar cada uno de los elementos que nos servirán para implementar nuestro producto, en uso práctico de los elementos y artefactos propios del proceso ingenieril.

La aplicación que pretendemos desarrollar es un módulo, más bien un componente, del Sistema Distribuido de Gestión Integral de Atención Primaria de Salud que se está desarrollando hoy por varios especialistas y estudiantes, este componente deberá integrarse como los demás que están en construcción, con estructuras, componentes y sistemas ya desarrollados por la empresa para este y otros proyectos desarrollados anteriormente y que son fácilmente reutilizables a nuestro entender, luego de haberlos evaluados previamente. Entre los componentes que reutilizaremos en nuestra aplicación están los componentes que conforman la plataforma PLASER por lo que esto también pasará a formar parte de los requisitos que deberán registrarse para este sistema en desarrollo.

Ahora describiremos cada uno de los requisitos que tiene nuestro módulo de programación y deberá cumplir el software que se entregue para satisfacer las demandas del cliente.

3.5.1 Requisitos Funcionales.

■ R1-Registrar Información sobre las Actividades Diarias del Equipo Básico de Salud(EBS).

1.1-Obtener datos del encabezamiento de la Hoja de Cargo los cuales están relacionados con el usuario que se autenticó y entró al Sistema (Nro. del Área de Salud a la que pertenece, Nro. del Grupo Básico de Trabajo, Nro. del Equipo Básico de Salud, fecha del RAD que se va a confeccionar, datos de los miembros del EBS [Nro. Registro Profesional, Nombre y Apellidos, Especialidad]).

1.2-Obtener datos sobre los registros de la consultas que pertenecen al RAD (Nro. de Historia Clínica, Nombre y Apellidos del paciente, Sexo, Edad, Lugar de Consulta [consulta, terreno], Hora de Consulta, Problema de Salud del paciente y su código CIE-10, Tipo de Diagnóstico [presuntivo, confirmado], Incidencias,Tipo de Actividad [primera consulta, seguimiento médico, seguimiento de enfermería, alta médica], Conductas a Seguir).

■ R2-Actualizar Información sobre las Actividades Diarias del Equipo Básico de Salud(EBS).

2.1-Solicitar fecha del RAD que desea actualizar, comprobar que tipo de usuario (miembro del EBS, técnico de estadística, directivo) es que lo está solicitando y comprobar si está autorizado para hacer esta operación.

2.2- Solicitar registro que desea modificar.

2.3- Permitir modificación de los datos referentes al problema de salud del paciente, las conductas a seguir para el mismo y así como el tipo de actividad y de diagnóstico para este.

2.4- Almacenar datos del estado anterior.

■ R3-Ofrecer Información con interés estadístico e investigativo.

3.1- Permitir elaborar y configurar Reportes de Indicadores de Salud.

3.2- Permitir elaborar y configurar Reportes de incidencias en Consultas del EBS.

3.3- Permitir elaborar y configurar Reportes de Casos Muestrales de la Población.

■ R4-Gestionar información externa de apoyo al Sistema, que a manera de “Servicios Web”(WS) se le ofrece a la Aplicación.

4.1- Solicitar datos de ubicación y acceso del usuario que entró a la Aplicación a través del WS que brindado por el Registro de Áreas de Salud(en construcción).

4.2- Solicitar datos de los miembros del EBS a través del WS brindado por el Registro de Áreas de Salud(en construcción).

4.3- Solicitar datos del paciente a través del WS brindado por el Registro de Población (en construcción).

4.4-Solicitar datos del Problema de Salud a través del WS brindado por el codificador CIE (en construcción).

4.5-Solicitar datos de las Conductas a Seguir a través del WS brindado por el codificador CCA (en construcción).

4.6-Solicitar datos de los Indicadores de Salud a través del WS brindado por el codificador CIS (en construcción).

■ R5-Integrar el módulo de programación y servicios “Registros de Actividades Diarias”(RAD) a la plataforma “PLASER”.

5.1-Realizar las llamadas de los Servicios Web a través de los métodos implementados para ello en PLASER.

5.2-Realizar las llamadas a las funciones de acceso a la Base de Datos a través de los métodos implementados para ello en PLASER.

5.3-Realizar las llamadas de la funciones propias de mi lógica del negocio a través de los métodos implementados para ello en PLASER.

5.4-Realizar transformaciones XML y XSL para la capa de presentación de mi negocio a través de los métodos implementados para ello en PLASER.

5.5-Obtener los permisos y datos de acceso a la Aplicación a través de los métodos implementados para ello en PLASER.

■ R6-Integrar el módulo de programación y servicios “Registros de Actividades Diarias” (RAD) a los demás módulos del Sistema de Gestión Integral de Atención Primaria de Salud, así como a otros módulos alojados en el Registro Informatizado de Salud(RIS) relacionados con el nuestro.

6.1-Comprobar si el Problema de Salud presentado por el paciente, que es consultado, es una EDO que lleva tarjeta.

6.2-Llamar al Registro de EDO en caso de que R 6.1 sea positivo.

6.2-Enviar aviso para Registro de Fallecidos en caso de que se registre en el RAD un Problema de Salud relacionado con las causas de fallecimiento o muerte.

■ R7-Brindar información para otros sistemas con interés en los datos registrados en el Registro de Actividades Diarias (RAD) a través de la implementación de Servicios Web (WS).

7.1-Brindar, a partir de una lista de Problemas de Salud, los datos de los pacientes, la Fecha en que fueron consultados, y los números de los EBS que los atendió a cada uno, en forma de WS para el Registro EDO(en construcción).

7.2-Brindar datos registrados en el RAD, que todavía están por determinar por los desarrolladores de ese módulo,en forma de WS para el Registro de Población(en construcción).

■ R8-Almacenar modificaciones(actualizaciones) realizadas a los Registros de Actividades Diarias, el tiempo “necesario”, con fines de auditorías y servicios legales relacionados con la actividad.

8.1-Almacenar datos modificables necesarios del estado anterior del registro que ha sido modificado, y la fecha de dicha actualización.

3.5.2 Requisitos No Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable, entre otras más. Generalmente están vinculados a requerimientos funcionales y forman una parte significatica de las especificaciones del software; este tipo de requerimientos puede marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

En estos requerimientos se incluyen el conjunto de facililidades del Sistema, las capacidades de este y algo muy importante, la seguridad.

3.5.2.1 Requisitos de Apariencia o Interfaz externa.

El sistema debe ser diseñado con pantallas que permitan la navegación de forma fácil y rápida; que caracterícen tanto a la empresa que realizó el sistema como a la que explotará el mismo. Aunque el uso de colores puede ser libre, deben priorizarse los que tengan características más refrescantes a la vista teniendo en cuenta las gran cantidad de horas que estarán mirando el sistema los usuarios finales de la aplicación.

3.5.2.2 Requisitos de Usabilidad.

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

Aunque los clientes del producto cuentan con la red informática más extendidad del país deberán hacerse inversiones en el orden tecnológico para evitar la ocurrencia de errores, en la comunicación y trasmisión de datos, de forma repetitiva.

3.5.2.3 Requisitos de Portabilidad

El producto debe poder ejecutarse sobre los Sistemas Operativos Windows 98, Windows 98 SE,Windows NT, Windows 2000, Windows XP o las variantes existentes de Linux; en pocas palabras ser multiplataforma o estar implementado de forma tal que pueda migrarse a a la plataforma deseada en cada momento sin necesidad de rehacerlo tod**o.**

3.5.2.4 Requisitos de Seguridad.

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

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

- Integridad: La información manejada por e l sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes. Deberán existir mecanismos de chequeo de integridad. Se permitirá la creación de copias de respaldo que puedan restaurarr el sistema en caso de fallo crítico o pérdida total de la información.

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

El módulo de registro de las Actividades Diarias utilizará las configuraciones y accesos que proporciona el módulo SCA(Sistema o Centro de Control), que gestiona a través de SAAA del “marco de trabajo”(framework) PlaSer, para cada usuario autenticado.

Deberá generarse archivos de traceo de la información modificada en la apliacación para la futura implementación de los módulos de auditoria de los servicios médicos y de control de los mismos.

3.5.2.5 Requisitos de Confiabilidad.

En el producto software deben prevenirse posibles fallos y tener formas de recuperación ante ellos; la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes; deberán existir mecanismos de chequeo de integridad. Deberá hallarse una estrategia de replicación que permita, de manera transparente para el usuario final, balancear la carga de acceso entre múltiples servidores aumentando los tiempos de respuesta y facilitar la recuperación inmediata del sistema si falla uno de ellos. Se permitirá la creación de copias de seguridad o respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información. El sistema debe ser capaz de realizar copias de seguridad de las bases de datos semanalmente. Deberá garantizarse la gestión de la información externa al módulo para evitar la inconsistencia de datos, así como la fidelidad de la información con la cual se trabaja.

3.5.2.6 Escalabilidad

La arquitectura de la aplicación debe desarrollarse basándose siguiendo una arquitectura multicapas (cada capa se debe construir como componente independiente, facilitando el mantenimiento del software.), Interfaz de usuario (presentación), Lógica de negocio, Acceso a Datos y Base de datos, orientada a servicios y basada en componentes, la capa de negocio estará disponible a otras aplicaciones utilizando

las tecnologías de XML Web Services. Todos los componentes del sistema deben desarrollarse siguiendo el principio de máxima cohesión y mínimo acoplamiento.

3.5.2.7 Requisitos de Interfaz interna.

Todos los componentes del sistema informático a desarrollar deben seguir el principio de máxima cohesión y mínimo acoplamiento como ya mencionamos en el requisitos anterior; los componentes reutilizables que integran el Sistema de Gestión Integral de Atención Primaria de Salud deberán ser desarrollados como Servicios Web XML que interactúan a través de SOAP con otros componentes.

3.5.2.8 Requisitos de Software.

Los clientes tendrán acceso al módulo Registro de Actividades Diarias y sus demás componentes a través de cualquier navegador web, recomendamos Mozilla 1.5, Internet Explorer 5.0, Netscape 4.0, Opera 7.0, o cualquier versión superior de cualesquiera de la variantes antes mencionadas.

3.5.2.9 Requisitos de Hardware.

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

Para que el sistema sea explotado al máximo de sus posibilidades y realmente cumpla los objetivos para los cuales va a ser creado, deberán dotarse de computadoras y modems a todos las consultas de los Equipos Básicos de Salud, así como a todas las áreas y departamentos de las Área de Salud, además de vías de comunicación (alámbricas o inalámbricas) preferiblemente de última generación si se desean mayor rendimiento y eficiencia del sistema, aunque todo el sistema podrá ser soportado por hardware de generaciones anteriores.

Requerimientos mínimos del sistema:

Procesador 486DX / 66 MHz o superior. 16 MB de memoria o de más capacidad de almacenaje. Monitor VGA o superior. Ratón Microsoft o compatible. No existen restricciones específicas en cuanto al Servidor, Impresora local o de red para imprimir los reportes solicitados.

3.5.2.10 Requisitos de Diseño e Implementación

Se utilizarán herramientas de desarrollo que garanticen la calidad de todo el ciclo de producción del Software, libres de licencia, por tanto tecnologías que garanticen el bajo costo de implantación del sistema y de su posterior mantenimiento; para ello deberá utilizarse como servidor y gestor de bases de datos el MySQL; como soportes de la programación, de la comunicación, en fin de la implementación del sistema se utilizarán lenguajes y estándares de la industria como : PHP, JavaScript, HTML, XML, XSL y demás tecnologías asociadas a las mismas. [Para mayor información al respecto puede remitirse al Capítulo 2 de este trabajo]. También se utilizarán algunas herramientas para el diseño visual del software y como marco de trabajo para la programación y el traceo de código fuente que aunque no libres serán de utilidad.

3.6 Conclusiones del Capítulo...

En este capítulo se ha descrito todo lo referente al negocio y presentado a manera de modelo estandarizado por la metodología de desarrollo de software que se está empleando en la elaboración de este producto. También, debido a los resultados de la modelación anterior, haciendo un estudio minucioso de la información que esta aporta, hemos podido describir organizadamente “todos” los requerimientos que deberá cumplir la aplicación que es objetivo de este trabajo.

El desarrollo de este flujo de trabajo ha posibilitado un entendimiento a fondo de los procesos de negocio de la organización donde estamos trabajando y los requisitos que esta necesita que sean cumplidos en el proyecto.

En este momento se está más preparado para enfrentar las fases y flujos posteriores del proceso de Ingeniería de Software, y contamos con los elementos suficientes para comenzar a modelar y enfrentar el Análisis y el Diseño del Sistema, con la meta de que los clientes puedan tener un ejemplo palpable, o al menos, más fehaciente del producto del que dispondrán en un futuro no muy lejano.

CAPÍTULO 4
Análisis y Diseño del Sistema

INTRODUCCIÓN

En este capítulo se documentará toda la información generada en los flujos de trabajo de Ánalisis y Diseño correspondientes a la la fase de Elaboración de la metodología de desarrollo de software que se ha empleado.

Modelaremos, a partir de la información registrada en este trabajo en los capítulos precedentes, los artefactos que describen el sistema que será desarrollado.Por lo tanto se podrá disponer de artefactos tales como los diagramas de caso de uso del sistema, los diagramas de clases del modelo de análisis y diseño, presentados en sus respectivos paquetes de organización y desarrollo, el diagrama de clases persistentes y el modelo de datos; todos con sus respectivas explicaciones y aclaraciones necesarias para que puedan ser entendidos.

Aunque el proyecto en estos momentos se encuentra en la fase de Construcción y algunas bases documentales no han sido lo suficientemente refinadas y por tanto pueden estar sujetas a cambios, también en esta sección modelaremos algunos artefactos propios del flujo de trabajo de Implementación y Prueba, los cuales serán los diagramas de despliegue y el diagrama de componentes, obtenidos por la organización generada por el Arquitecto de Software, de la empresa, para este proyecto.

El desarrollo de este capítulo es el progreso a etapas superiores en el proceso de Ingeniería de Software; esta base documental es la primícia para el desarrollo de las versiones necesarias por las cuales transitará el producto que se entregará hasta que se arribe la versión final de este, constituye el material de enlace y comunicación entre los participantes e implicados en la producción de esta aplicación y la razón de ser de estos flujos de trabajo.

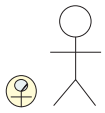
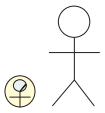
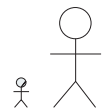
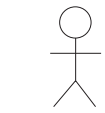
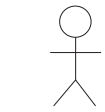
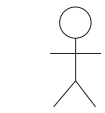
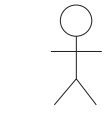
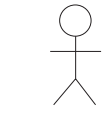
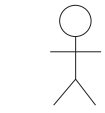
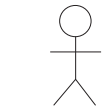
4.1 ANÁLISIS DEL SISTEMA

4.1.1 Actores del sistema. Justificación.

Los actores del sistema que serán presentados se han mantenido desde el modelado del negocio, por lo que se presentaron anteriormente como trabajadores y actores en este flujo de trabajo, aunque en comparación con este algunos no están reflejados porque para el nuevo proceso que se desea implementar sus roles desaparecen al ser sustituidos por acciones que puede realizar el software.

Ahora bien, muchos de estos actores son sistemas de información, más bien módulos o componentes de este mismo proyecto de desarrollo que, de la misma manera, se encuentran en construcción pero que su relación y por tanto interacción con el nuestro está bien definida por los analistas que están conciviendo el producto.

En la tabla que se presenta a continuación están reflejados estos actores y sus roles en relación al trabajo que realizan en la aplicación.

ACTOR DEL SISTEMA	JUSTIFICACIÓN
1  EBS	Son los encargados de interactuar con la aplicación, gestionan la información brindada por otros sistemas externos para completar la información suficiente que necesita el Registro de Actividades Diarias, pueden modificar la información ya registrada y realizar estudios e informes a través de los reportes que elabora la aplicación.
2  TEAS TEGBT	Mientras los EBS no cuenten con las tecnologías necesarias para realizar automaticamente la gestión del RAD, podrán realizar el rol de estos, cuando esta situación sea saldada podrán apoyar el trabajo de los EBS cuando fuese necesario y además realizar estudios apartir de la información elaboradade los reportes del sistema.
3  J G B T JDEAS VDAM D A S	Gracias a las facilidades de la solución informática, estos directivos del nivel de Área de Salud, podrán hacer solicitudes al software de los reportes que este realiza, así como podrán visualizar los Registros de Actividades Diarias que se estén efectuando y los ya fueron realizados.
4  RP	Sistema de Información externo que brinda al módulo Registro de Actividades Diarias los datos personales del paciente que están registrados como ciudadanos de este país. También este sistema se beneficia de la información almacenada en el RAD para actualizar sus bases de datos.
5  CIE	Sistema de Información externo que brinda al módulo Registro de Actividades Diarias los datos sobre el problema de salud del paciente, en otras palabras los nombres oficiales del problema de salud y su código CIE-10.
6  RAS	Sistema de Información externo que brinda al módulo Registro de Actividades Diarias los datos referentes al Equipo Básico de Salud que va a trabajar con el sistema, así como la ubicación y pertencia del RAD que se está realizando, a través de él, también se pueden hacer actualizaciones en la conformación oficial del EBS.
7  CCA	Sistema de Información externo que brinda al módulo Registro de Actividades Diarias los datos de las conductas a seguir que pueden ser indicadas por los Equipos Básicos de Salud ante los problemas de salud presentados por los pacientes en cada consulta.Estas conductas o accionares son determinados por APS en el MINSAP.
8  CIS	Sistema de Información externo que brinda los datos necesarios para poder realizar uno de los reportes que este facilita para la investigación estadística, en este caso los indicadores de salud que en todo el país se están teniendo en cuenta para el análisis y balance nacional de comportamiento de enfermedades en APS.
9  SAAA	Sistema de Información externo que brinda al módulo Registro de Actividades Diarias los accesos que tienen los usuarios para este, garantiza además la seguridad total del sistema, la confiabilidad de la información, así como el traceo de acciones realizadas por los usuarios que trabajan con la aplicación.
10  SCA	Sistema de Información externo que crea los usuarios de la aplicación, brinda a estos usuario al auteticarse en el sistema la configuración general de este y los permisos que tiene para trabajar y canaliza las peticiones que el sistema hace fuera del área de alcance que tiene.

Leyenda:
RP: Registro de Población, CIE: Clasificación Internacional de Enfermedades y de Problemas Relacionados con la Salud, RAS: Registro de Áreas de Salud, SAAA: Sistema de Seguridad y Control de Acceso, CCA: Codificador de Conductas a Seguir o Accionar, CIS: Codificador de Indicadores de Salud, SCA: Sistema de Control de Avisos. EBS, TEGBT, TEAS, JGBT, JDEAS, VDAM, DAS (VER Capítulo 3 ”MODELADO DEL NEGOCIO” Tablas 3.1 y 3.2)
Trabajador del negocio Actor del negocio

Tabla 4.1 Definición y justificación de los Actores del sistema.

4.1.2 Casos de Uso del Sistema. Descripción. Especificación. Prototipo de Interfaz de Usuario.

Los actores del sistema, presentados en el epígrafe anterior, participan en los procesos que se reflejarán en el siguiente diagrama de casos de uso del sistema:

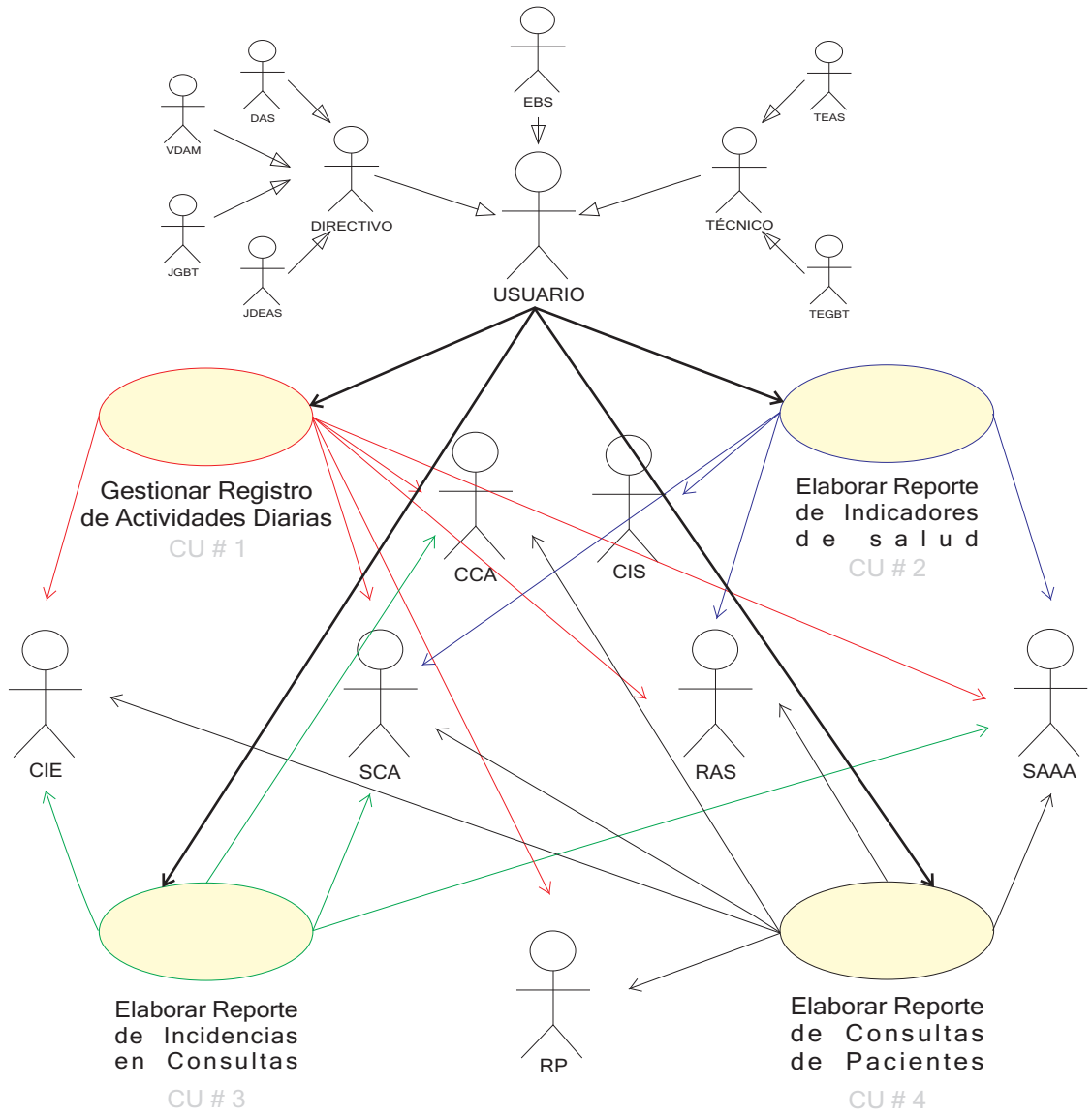


Fig.4.1 Diagrama de Casos de Uso del sistema.

Un análisis a priori del anterior diagrama muestra el nivel de distribución de la información a la cual tiene que acceder el Registro de Actividades Diarias para garantizar su negocio, así como la justificación de utilización de la arquitectura de desarrollo que se decidió emplear (ver capítulo 2).

Garantizar la seguridad y accesibilidad a la aplicación, así como la disponibilidad de todos sus servicios, son factores muy importantes a tener en cuenta en el análisis y posterior diseño de este sistema, hay que valorar alternativas de hardware, situación económica actual de la organización y posibilidad real de nuestro país de enfrentar el desarrollo e implantación de un sistema informático como el que se pretende producir, pero estos temas serán abordados en el capítulo siguiente.

Volviendo a los casos de uso del sistema, ahora se describirán y especificarán en relación con los prototipos de las interfaces de usuarios que para cada proceso se propondrá.

CASO DE USO DEL SISTEMA # 1	Gestionar Registro de Actividades Diarias(RAD)
ACTORES	EBS, TEGBT, TEAS
PROPÓSITO Registrar todas las actividades referentes a las consultas realizadas por el EBS a los pacientes de su población, normalizadas en la Hoja de Cargo de los EBS.	
RESUMEN El Caso de Uso se inicia cuando el EBS desea registrar los datos referentes a las consultas de los pacientes de su población en cada jornada laboral o actualizar un registro que ya ha introducido previamente y desea corregirlo por alguna equivocación; para ambos casos deberá gestionar, a través del sistema, información que le brindan otros sistemas externos; concluyendo el Caso de Uso con el almacenado o la modificación de los registros derivados de las consultas de los pacientes y la labor de los EBS, así como las trazas, por así llamarlo, de los cambios que efectuó el EBS en los registros del RAD por motivos legales y de seguridad. El Caso de Uso también lo pueden iniciar el TEGBT o el TEAS mientras los EBS no cuenten con las tecnologías necesarias para registrar los datos referentes a las consultas de los pacientes de su población en cada jornada laboral; igualmente deberá, a través del sistema, gestionar información que le brindan otros sistemas externos; concluyendo el Caso de Uso con el almacenado de los registros derivados de las consultas de los pacientes y la labor de los EBS.	
REFERENCIAS	RF1
PRECONDICIONES	☑Usuario autenticado. ☑Permisos de trabajo para el Usuario autenticado asignados. ☑Página Principal del RAD cargada.
POSTCONDICIONES	☑Almacenado o Modificación de los datos referentes a la actividad de los EBS en el RAD.. ☑Nuevos registros adicionados a la base de datos. ☑Registro de cambios ocurridos en el RAD en tabla de seguridad.
REQUERIMIENTOS ESPECIALES	
CURSO NORMAL DE EVENTOS >>> CUS # 1	
Acción del Actor	Respuesta del Sistema
1-Usuario elige una opción en el menú “Consultas” de la pantalla.	2-Si el usuario elige opción del menú: A)Registrar Nueva Consulta ⇒ ir a sección # 1 “Registrar Consulta” CU#1. b)Ver Consultas del día ir a sección # 2 “Ver Registro de Actividades Diarias” CU#1.
3-Usuario elige salir del menú Consultas e ir a cualquier otro menú de la pantalla.	4-culmina acciones del caso de uso.

La pantalla inicial (INICIO) del módulo no será mostrada en este trabajo, debido que cuando se confeccionó este documento todavía esta no había sido presentada a los analistas, la misma contará con menú de opciones como las demás que se muestran en este trabajo, entre otras funcionalidades e informaciones útiles

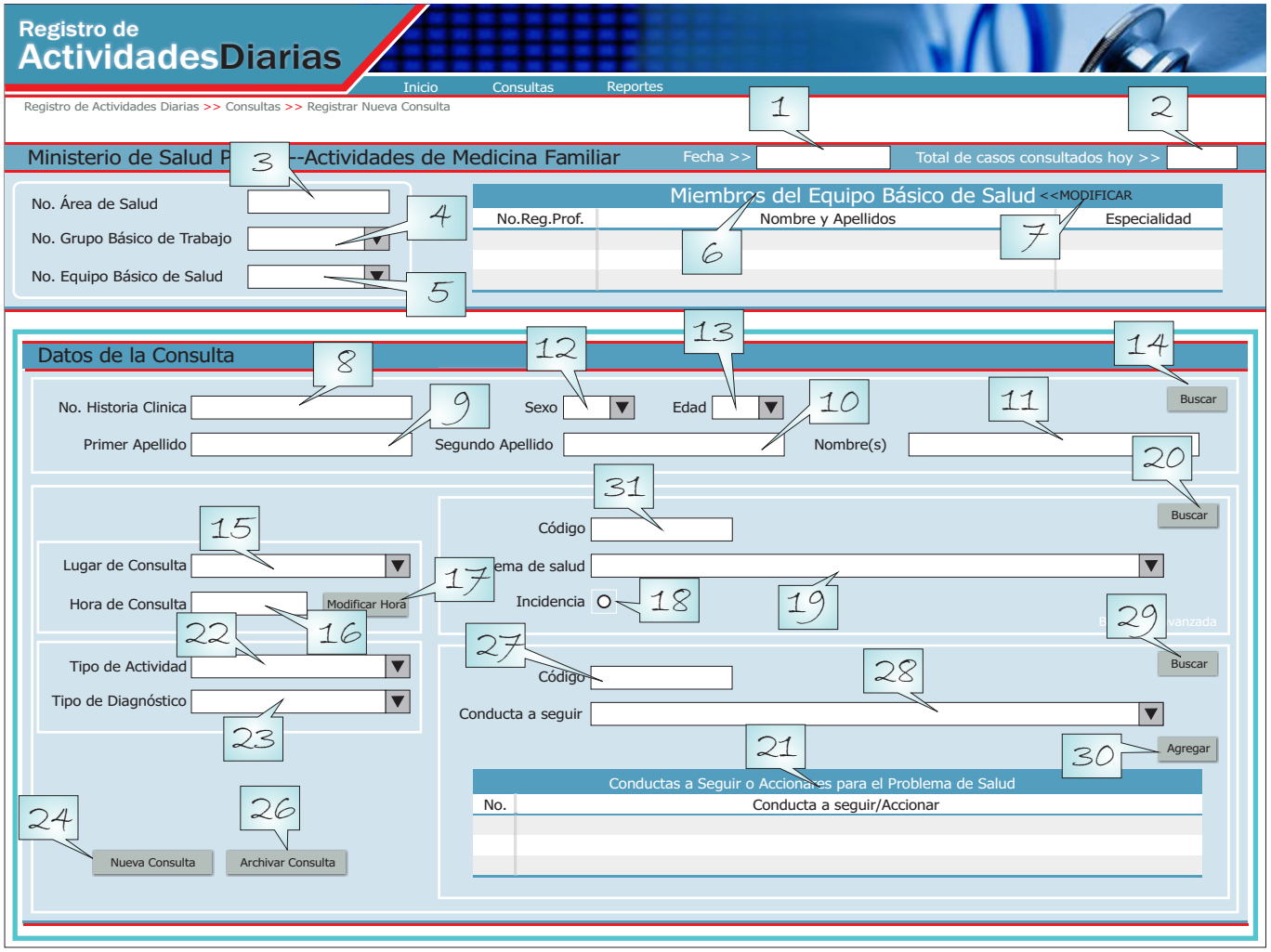


Fig. 4.2 Prototipo de Interfaz de Usuario (Pantalla # 1--- CUS#1).

SECCIÓN # 1 “Registra rNueva Consulta”” >>> CUS # 1	
Acción del Actor	Respuesta del Sistema
	1-Carga pantalla # 1 CU#1 obtiene y muestra Fecha y Hora del Servidor en [1] y [16] respectivamente; total de registros en [2] en el día que se muestra en [1]; los números del AS, GBT(s), EBS(s) en [3][4][5] respectivamente del usuario que hizo la petición y los datos de los miembros del EBS en [6] correspondiente co el EBS [5], GBT[4] y el AS[3] .
2-Usuario ejecuta cambiar fecha[17] de[16] .	3-Ir a sección # 1.1 “Modificar Fecha”.
4-Usuario ejecuta Modificar Miembros del EBS [7].	5-Ir a sección # 1.2 “Modificar Miembros del EBS”.
6-Usuario desea cambiar el GBT[4] y el EBS[5] con los que trabajará.	7--Ir a sección # 1.3 “Modificar GBT y EBS”.

SECCIÓN # 1 “Registrar Consulta”” >>> CUS # 1 continuación...	
Acción del Actor	Respuesta del Sistema
8-Usuario introduce datos “suficientes” en [8][9][10][11][12][13] para buscar los datos personales del paciente al cual va a registrarle la consulta y ejecuta acción Buscar[14].	9-Realiza la acción ejecutada, obtiene y muestra los resultados esperados completando en [8][9][10][11][12][13].
10-Usuario selecciona Lugar de Consulta[15].	11-Muestra selección en [15].
12-Usuario ejecuta Modificar Hora[17].	13-Ir a sección # 1.4 “Modificar Hora”.
14-Usuario escoje Problema de salud en[19].	15-Muestra algún problema de salud en[19] si ya se ha ejecutado al menos una vez acción[20].
16-Usuario escribe datos suficientes en [31] y [19] y ejecuta acción Buscar Problema de Salud [20].	17-Ir a sección # 1.5 “Buscar Problema de Salud”.
18-Usuario marca [18] si seleccionado en [19] resulta ser una Incidencia para el paciente.	19-Muestra resultado de selección [18].
20-Usuario selecciona Tipo de Actividad [22] y de Diagnóstico [23] para el problema de Salud seleccionado en [19].	21-Muestra tipo de actividad y de diagnostico seleccionados en[22] y [23] respectivamente.
22-Usuario escoje Conducta a Seguir en [28].	23-Muestra alguna Conductas a Seguir en [28] si ya se ha ejecutado al menos una vez acción[29].
24-Usuario escribe datos suficientes en [27] y [28] y ejecuta acción Buscar Conducta a Seguir [29].	25-Ir a sección 1.6 “Buscar Conducta a Seguir ”
26-Usuario ejecuta acción Registrar Consulta [26].	27-Envía los datos del formulario (si todos los campos necesarios están llenos) a la base de datos de la aplicación, manteniendo solamente los datos del paciente visualizados en el formulario, además de los de encabezamiento del RAD.
28-Usuario ejecuta acción Nueva Consulta [24]	29-Limpia todos los datos del formulario, excepto los de encabezamiento del RAD

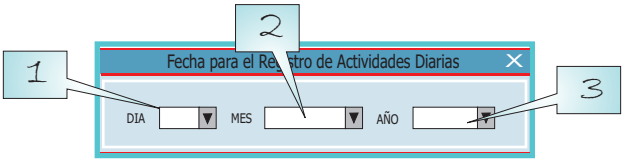


Fig. 4.3 Prototipo de Interfaz de Usuario (Pantalla # 1.1--- CUS#1).

SECCIÓN # 1.1 “Modificar Fecha” >>> CUS # 1	
Acción del Actor	Respuesta del Sistema
	1-Verifica si el Usuario que hace la petición tiene la autorización para ejecutar esta acción: a) Si lo tiene ⇒ Carga pantalla # 1.1 CU#1 y va al paso 2 . b)Si no los tiene ⇒ Muestra mensaje de error.
2-Usuario selecciona en [1] el día, en [2] el mes y en [3] el año.	3-Muestra resultados de la selección.
4-Usuario cierra pantalla emergente.	5-Muestra en [1] de la pantalla 1 CU#1 los resultados de esta sección

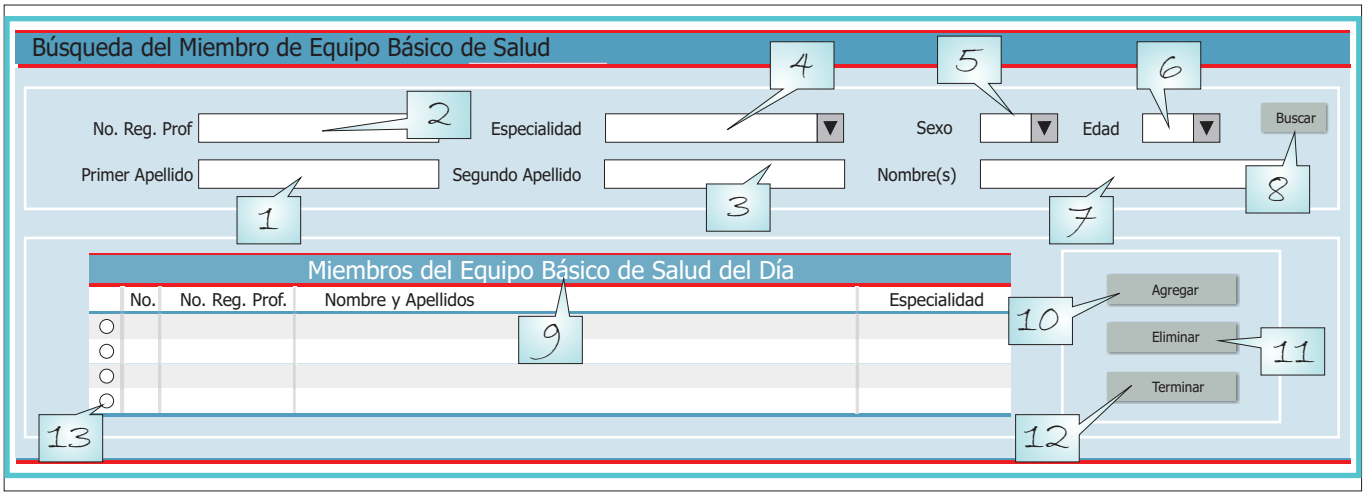


Fig. 4.4 Prototipo de Interfaz de Usuario (Pantalla # 2 --- CUS#1).

SECCIÓN # 1.2 “Modificar miembros del EBS” >>> CUS # 1	
Acción del Actor	Respuesta del Sistema
	1-Carga pantalla # 2 CU#1 con los datos que están en [6] de la pantalla # 1 CU#1 en [9] de esta pantalla.
2-Usuario selecciona en [13] el miembro del EBS que desea eliminar del equipo de ese día y ejecuta acción Eliminar[11].	3-Marca el registro seleccionado y elimina de [9] registro que fue seleccionado.
6-Usuario introduce datos “suficientes” en [1], [2], [3], [4], [5], [6], [7] para buscar los datos del miembro del EBS que va a estar presente en la consulta, y ejecuta acción Buscar[8]. 8-Usuario ejecuta acción Agregar[11].	7-Realiza la acción ejecutada y muestra los resultados completando en [1], [2], [3], [4], [5], [6], [7]. 9-Agrega el miembro previamente buscado y encontrado en el paso 6 a las lista [9].

SECCIÓN # 1.2 “Modificar miembros del EBS” >>> CUS # 1 continuación...	
Acción del Actor	Respuesta del Sistema
10-Usuario ejecuta acción Terminar	11-Cierra pantalla # 2 CU#1 y actualiza la tabla [6] de la pantalla # 1 con los datos que quedaron en la pantalla # 2 en la tabla [9].

SECCIÓN # 1.4 “Modificar Hora” >>> CUS # 1	
Acción del Actor	Respuesta del Sistema
	1-Verifica si el Usuario que hace la petición tiene la autorización para ejecutar esta acción: a) Si la tiene ⇒ habilita región editable [16] de la pantalla # 1 CU#1 y pasa al paso 2. b)Si no la tiene ⇒ Muestra mensaje de error.
2-Usuario escribe en [16] de la pantalla # 1 CU#1 hora nueva con el formato establecido.	3-Muestra la hora introducida.

SECCIÓN # 1.5 “Buscar Problema de Salud”” >>> CUS # 1	
Acción del Actor	Respuesta del Sistema
.	1-Realiza la acción ejecutada, obtiene y muestra los resultados esperados completando en [31] y [19] de la pantalla # 1 CU#1.

SECCIÓN # 1.6 “Buscar Conductas a Seguir”” >>> CUS # 1	
Acción del Actor	Respuesta del Sistema
	1-Realiza la acción ejecutada y muestra los resultados completando en [27] y [28].
3-Usuario ejecuta acción Agregar [30].	4-Agrega los que está en [27] y [28] en [21].

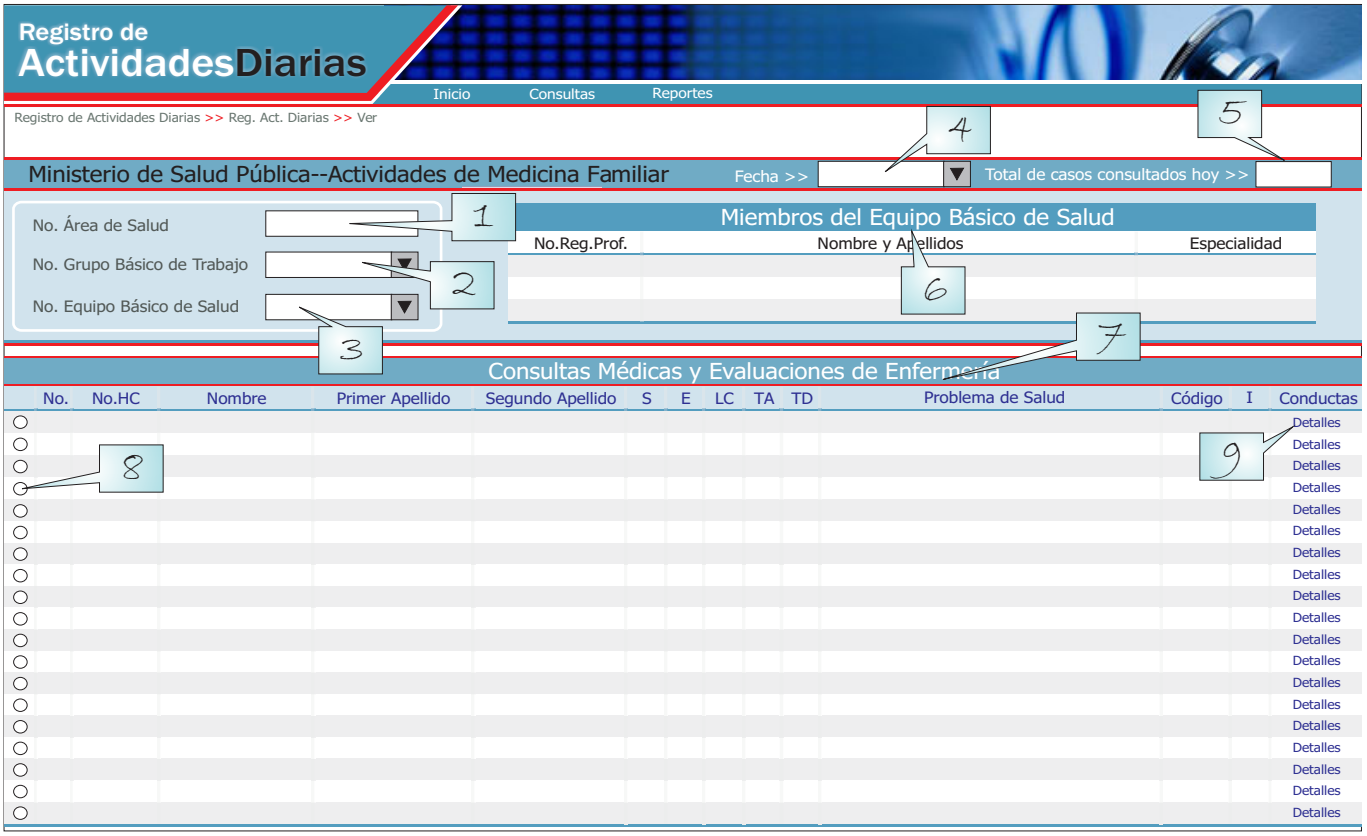


Fig. 4.5 Prototipo de Interfaz de Usuario (Pantalla # 5 --- CUS#1).

SECCIÓN # 2 “Ver Consultas del Registro de Actividades Diarias” >>> CUS # 1	
Acción del Actor	Respuesta del Sistema
	1-Carga pantalla # 5 CU#1 y muestra Fecha del Servidor en [4]; total de registros en [5] en el día [4]; AS, GBT(s), EBS(s) en [1][2][3] respectivamente del usuario que hizo la petición y datos de los miembros del EBS [6] del EBS [3], GBT[2] y el AS[1].
	2-Para la Fecha señalada en [4] aparecen en [7] los registros de consultas efectuadas ese día.
3-Usuario selecciona registro [8].	4-Carga pantalla # 1 CU#1 con los datos antes reflejados en la pantalla # 5. Según el usuario que esté autenticado en esta acción podrá modificar o no un registro siempre que se cumpla el requisito existente para la actualización de registros.
5-Usuario elige ver Detalles [9] de las conductas	6-Carga en pantalla emergente los datos referentes a las conductas a seguir almacenadas en la base de datos del RAD pertenecientes a ese registro.

SECCIÓN # 2 "Ver Consultas del Registro de Actividades Diarias" >>> CUS # 1 continuación...	
Acción del Actor	Respuesta del Sistema
7-Usuario decide cambiar Fecha [4]	8-Visualiza los registro que están almacenados en el RAD para la nueva fecha que se ha introducido, actualiza, total de registros para ese día en [5].

CASO DE USO DEL SISTEMA # 2	Elaborar reporte de Indicadores de Salud del Equipo Básico de Salud
ACTORES	EBS, TEGBT, TEAS, JGBT, JDEAS, VDAM, DAS
PROPÓSITO Realizar reportes/informes, por petición, a partir de la información almacenada por el EBS en el RAD de los indicadores de salud regulados por el MINSAP.	
RESUMEN El Caso de Uso se inicia cuando el EBS,TEGBT,TEAS, JGBT, JDEAS, VDAM o el DAS desea tener reportes de los indicadores de salud de la población de un EBS, un GBT o del AS, según sea el caso, en un período determinado, a partir de la información almacenada en el RAD; el Usuario podrá escoger los indicadores que desee que se reflejen en el informe, personalizando de este modo la vista final de este reporte, así como escoger el tamaño de la población que quiere reportar dependiendo solo este punto de su nivel de acceso y autorización; concluyendo el Caso de Uso con la muestra en pantalla de la información estadísticamente elaborada que antes se solicitó.	
REFERENCIAS	RF3
PRECONDICIONES	☑Usuario autenticado. ☑Permisos de trabajo para el Usuario autenticado asignados. ☑Página de los Reportes RISEBS cargada.
POSTCONDICIONES	☑Reporte de los Indicadores de Salud de un EBS, GBT o un AS elaborado según sea el usuario que hizo la petición.
REQUERIMIENTOS ESPECIALES	

CURSO NORMAL DE EVENTOS >>> CUS # 2	
Acción del Actor	Respuesta del Sistema
1-Usuario elige la opción Reporte de Indicadores de Salud del EBS en el menú "Reportes" de la pantalla.	2-Carga pantalla # 1 CU#2 y muestra AS, GBT(s), EBS(s) en [1],[2] y [3] respectivamente del usuario que hizo la petición, obtiene y muestra las agrupaciones de indicadores de salud en [5] y el total de indicadores que forman dicho grupo en [7].
2-Usuario escoje rango de tiempo que desea reportar en [4] .	3-Sistema muestra rango de tiempo que seleccionó el Usuario.

CURSO NORMAL DE EVENTOS >>> CUS # 2	
Acción del Actor	Respuesta del Sistema
4-Usuario selecciona las agrupaciones de indicadores que desea expandir en [6] y ejecuta acción Expandir Grupos [9].	5-Obtiene y muestra los indicadores de los grupos de indicadores que fueron seleccionados.
6-Usuario ejecuta acción Elaborar Reporte de la pantalla.	7-Obtiene y muestra el total de casos antendidos referentes a los indicadores de salud seleccionados en el paso 4 en [7] en el rango de tiempo [4] y almacenados en el RAD.

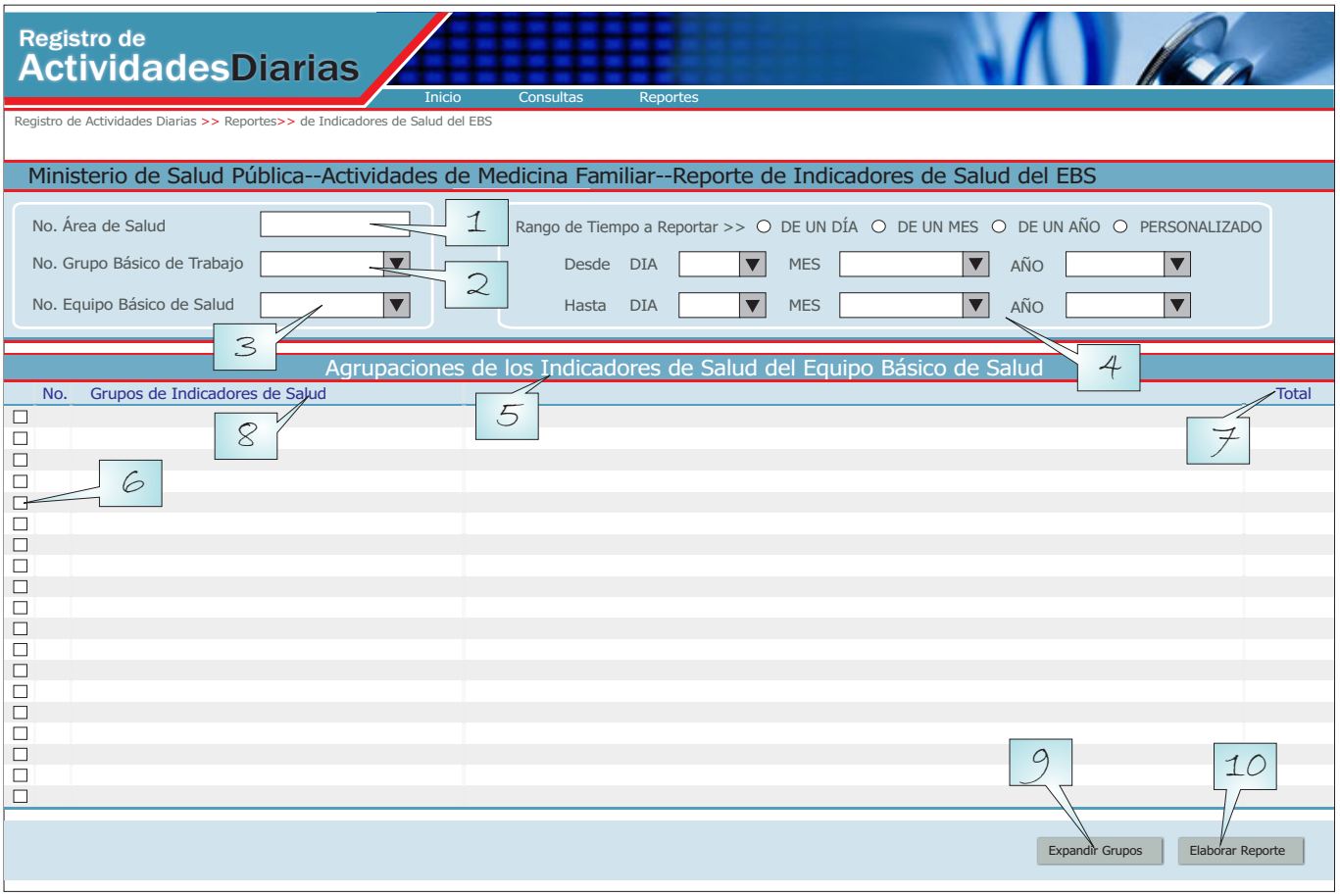


Fig. 4.6 Prototipo de Interfaz de Usuario (Pantalla # 1 --- CUS#2).

CASO DE USO DEL SISTEMA # 3	Elaborar reporte de Incidencias en Consultas del Equipo Básico de Salud
ACTORES	EBS, TEGBT, TEAS, JGBT, JDEAS, VDAM, DAS
PROPÓSITO Realizar reportes/informes, por petición, a partir de la información almacenada por el EBS en el RAD de las incidencias en consultas del EBS.	
RESUMEN El Caso de Uso se inicia cuando el EBS,TEGBT,TEAS, JGBT, JDEAS, VDAM o el DAS desea tener reportes de las incidencias en consultas del EBS, un GBT o del AS, según sea el caso, en un período determinado, a partir de la información almacenada en el RAD; el Usuario podrá configurar el reporte tanto como desee y el diseño del mismo se lo permita, personalizando de este modo la vista final de este reporte, así como escoger el tamaño de la población que quiere reportar dependiendo solo este punto de su nivel de acceso y autorización; concluyendo el Caso de Uso con la muestra en pantalla de la información estadísticamente elaborada que antes se solicitó.	
REFERENCIAS	RF3
PRECONDICIONES	☑Usuario autenticado. ☑Permisos de trabajo para el Usuario autenticado asignados. ☑Página de los Reportes RICEBS cargada.
POSTCONDICIONES	☑Reporte de las incidencias en consultas del EBS, GBT o un AS elaborado según sea el usuario que hizo la petición.
REQUERIMIENTOS ESPECIALES	

CURSO NORMAL DE EVENTOS >>> CUS # 3

Acción del Actor	Respuesta del Sistema
1-Usuario elige la opción Reporte de Incidencias en Consultas del EBS en el menú “Reportes” de la pantalla.	2-Carga pantalla # 1 CU#3 y muestra AS, GBT(s), EBS(s) en [1],[2] y [3] respectivamente del usuario que hizo la petición, obtiene y muestra los problemas de salud en [9] que han sido diagnósticados, al menos una vez en el AS ala que pertenece el usuario autenticado y las conductas a seguir en [10] que igualemente han sido indicadas alguna vez en el AS a la que pertenece el usuario autenticado.
2-Usuario escoje rango de tiempo que desea reportar en [4] .	3-Sistema muestra rango de tiempo que seleccionó el Usuario.
4-Usuario selecciona a voluntad Lugar de Consulta [6], Tipo de Diasgnóstico [7], y Tipo de Actividad [8] para configurar el reporte.	5-El sistema muestra selección realizada.
6-Usuario selecciona a voluntad Problema de Salud en [9] y Conducta a Seguir en [10] y Agrega a listas [11] y [12] respectivamente.	7-Muestra resultados de selección y de la agregación.
8-Usuario ejecuta acción Elaborar Reporte [14] de la pantalla.	9-Obtiene y muestra las incidencias en consultas, según la configuración en [13].

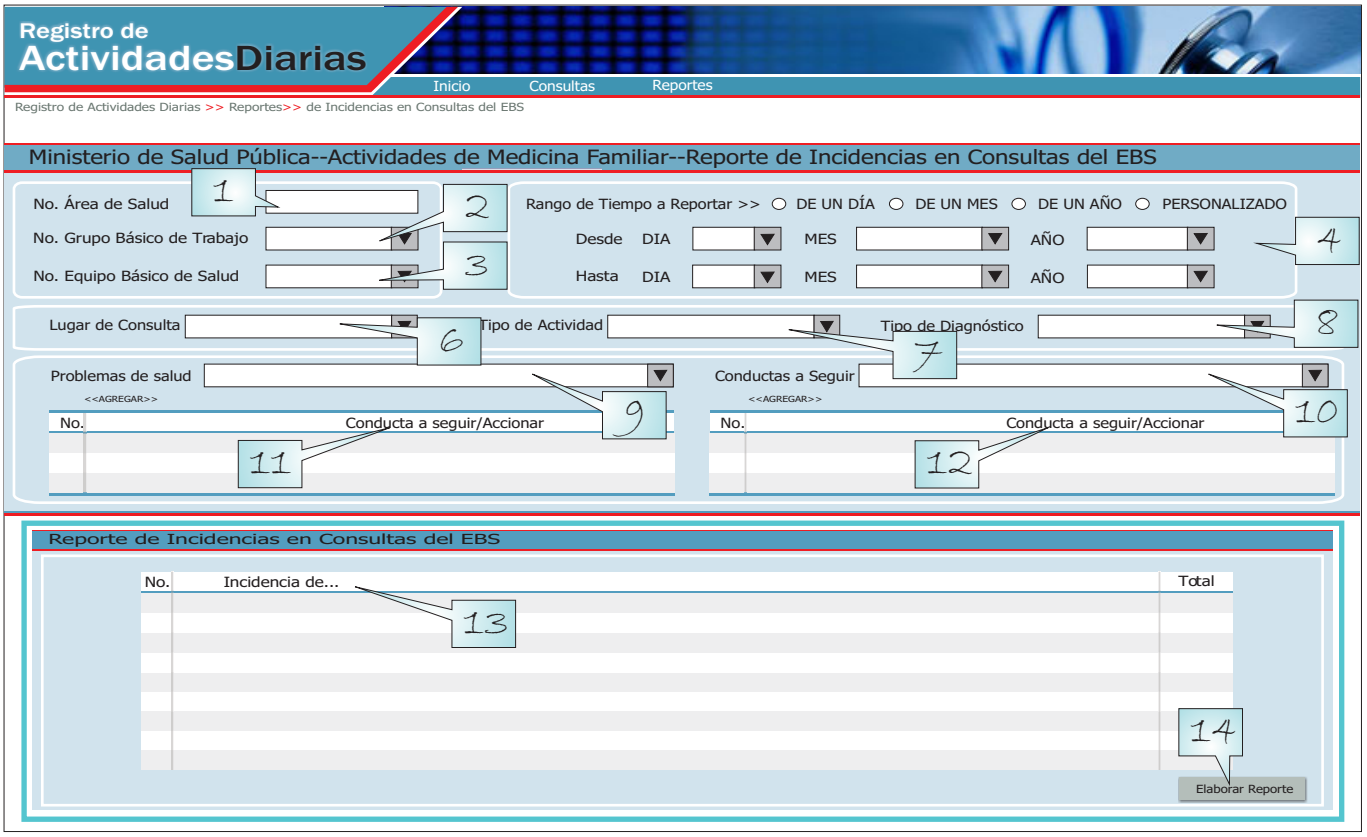


Fig. 4.7 Prototipo de Interfaz de Usuario (Pantalla # 1 --- CUS#3).

CASO DE USO DEL SISTEMA # 4	Elaborar reporte de Casos Muestrales de la Población del Equipo Básico de Salud
ACTORES	EBS, TEGBT, TEAS, JGBT, JDEAS, VDAM, DAS
PROPÓSITO Realizar reportes/informes, por petición, a partir de la información almacenada por el EBS en el RAD de las incidencias en consultas del EBS.	
RESUMEN El Caso de Uso se inicia cuando el EBS,TEGBT,TEAS, JGBT, JDEAS, VDAM o el DAS desea tener reportes de los Casos muestrales de la población del EBS, un GBT o del AS, según sea el caso, en un período determinado, a partir de la información almacenada en el RAD; el Usuario podrá configurar el reporte tanto como desee y el diseño del mismo se lo permita, personalizando de este modo la vista final de este reporte, así como escoger el tamaño de la población que quiere reportar dependiendo solo este punto de su nivel de acceso y autorización; concluyendo el Caso de Uso con la muestra en pantalla de la información estadísticamente elaborada que antes se solicitó.	
REFERENCIAS	RF3
PRECONDICIONES	☑Usuario autenticado. ☑Permisos de trabajo para el Usuario autenticado asignados. ☑Página de los Reportes RCMPEBS cargada.
POSTCONDICIONES	☑Reporte de los casos muestrales de la población del EBS, GBT o un AS elaborado según sea el usuario que hizo la petición.
REQUERIMIENTOS ESPECIALES	

CURSO NORMAL DE EVENTOS >>> CUS # 3	
Acción del Actor	Respuesta del Sistema
1-Usuario elige la opción Reporte de los casos Muestrales de la Población del EBS en el menú “Reportes” de la pantalla.	2-Carga pantalla # 1 CU#3 y muestra AS, GBT(s), EBS(s) en [1],[2] y [3] respectivamente del usuario que hizo la petición, obtiene y muestra los problemas de salud en [10] que han sido diagnósticados, al menos una vez en el AS a la que pertenece el usuario autenticado y las conductas a seguir en [11] que igualemente han sido indicadas alguna vez en el AS a la que pertenece el usuario autenticado.
2-Usuario escoje rango de tiempo que desea reportar en [4] .	3-Sistema muestra rango de tiempo que seleccionó el Usuario.
4-Usuario selecciona a voluntad Lugar de Consulta [5], Tipo de Diasgnóstico [6], y Tipo de Actividad [7] para configurar el reporte.	5-El sistema muestra selección realizada.
6-Usuario selecciona a voluntad Problema de Salud en [9] y Conducta a Seguir en [10] y Agrega a listas [11] y [12] respectivamente.	7-Muestra resultados de selección y de la agregación.
8-Usuario introduce datos “suficientes” en [14], [15], [15], [16], [18], [19], [20] para buscar los datos personales del paciente al cual va a registrarle la consulta y ejecuta acción Buscar[17].	9-Realiza la acción ejecutada, obtiene y muestra los resultados esperados completando en [14], [15], [16], [18], [19], [20].
10-Usuario ejecuta acción Agregar[22]	11-Agrega a [21] lo que se buscó está en [18], [19], [20] si ya se ha realizado el paso 8.
8-Usuario ejecuta acción Elaborar Reporte [23] de la pantalla.	9-Obtiene y muestra los casos muestrales de la población en consultas, según la configuración en [24].

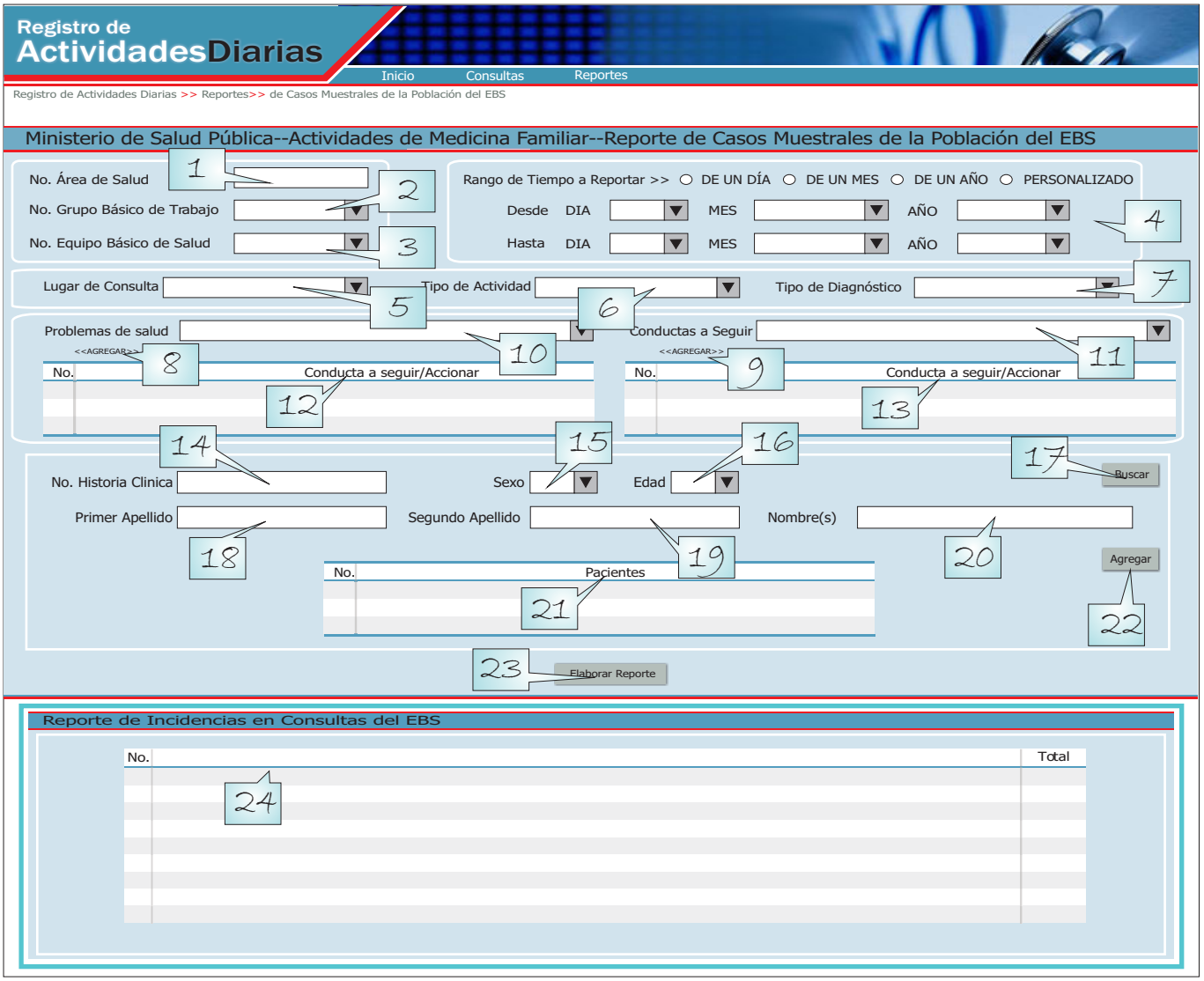


Fig. 4.8 Prototipo de Interfaz de Usuario (Pantalla # 1 --- CUS#4).

Siguiendo las actividades que recomienda la metodología de desarrollo de Proceso Unificado (RUP+UWE) se mostrará el mapa de navegación para la aplicación Web que se está implementando. Este servirá para que los programadores y diseñadores gráficos inmiscuidos en el proyecto puedan tener una idea exacta de la navegabilidad de la aplicación que están desarrollando y del producto como tal.

A partir de este mapa tendremos una idea de cómo se posicionan algunas de las pantallas que han sido mostradas anteriormente y asumir la funcionalidad y razón de ser de otras que no han sido explicadas pero que en este mapa se presentan.

La construcción de un mapa de navegación no solo ayuda a un mejor entendimiento de la solución para los desarrolladores sino que también ayuda a comprender y situarse en tiempo y espacio a los clientes y futuros usuarios del software.

El uso de estos mapas no se reducen solo a al desarrollo de aplicaciones Web sino para cualquier otro tipo de aplicaciones que muestren diversas vistas o interfaces al los usuarios a partir de acciones realizadas por este.

4.2 MAPA DE NAVEGACIÓN.

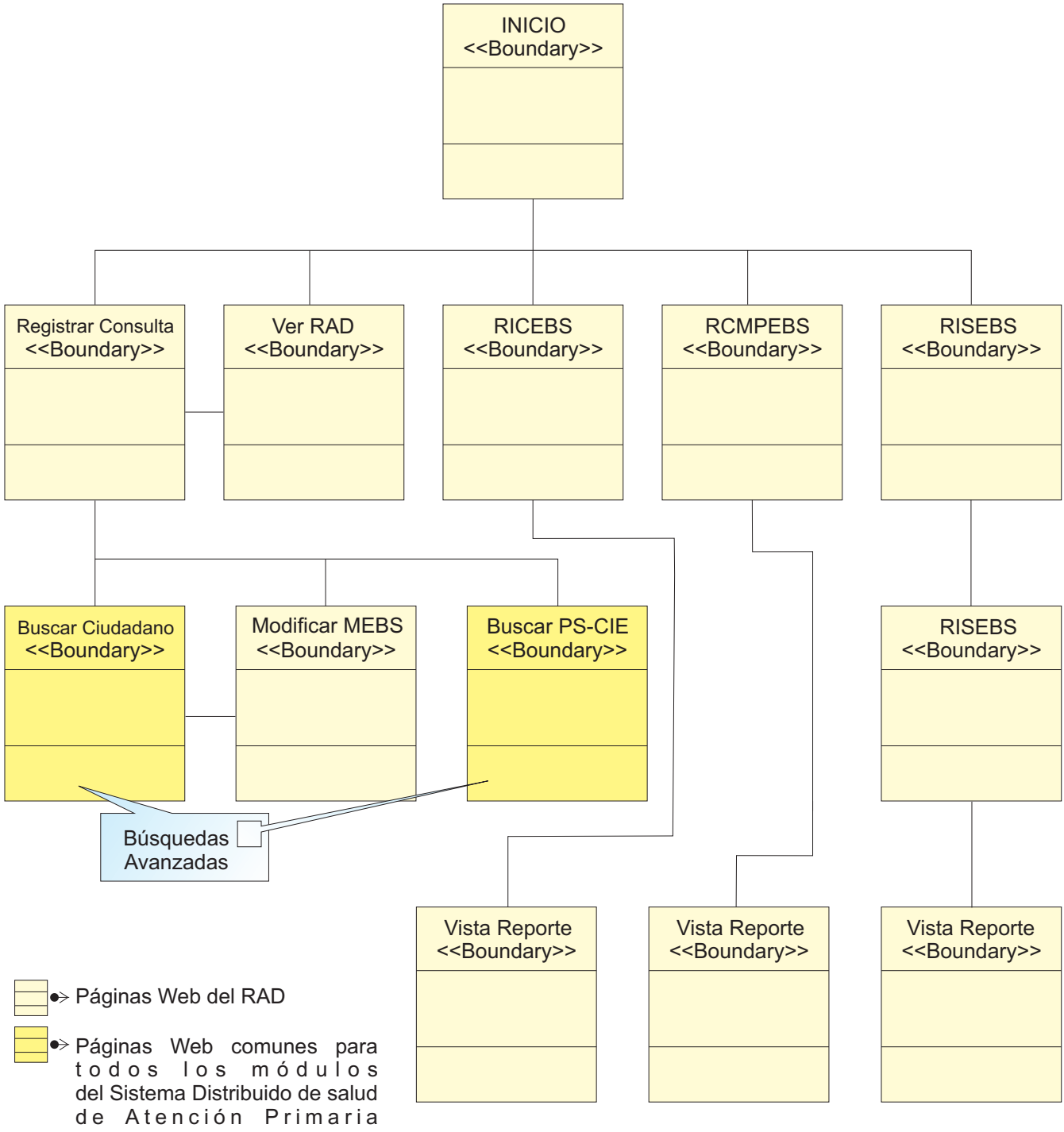


Fig. 4.9 Mapa Navegacional del Módulo de Registro de Actividades Diarias.

El mapa de navegación representa y estructura la visión global del sistema, generalmente se presentan definidos para cada tipo de usuario y especificando la navegación permitida, en el caso anterior solo se presenta el mapa general de la aplicación.

4.3 DISEÑO DEL SISTEMA

4.3.1 Paquetes de organización. Relación entre paquetes.

Los paquetes son formas de organización a la hora de diseñar cualquier sistema informático, en ellos se reúnen elementos comunes, que guardan estrecha relación en el o los procesos que se quieren modelar.

En los paquetes se organiza y presenta la modelación de los diagramas de clases del sistema, base estructural primaria para todo sistema desarrollado bajo principios orientados a objetos que servirá para la posterior implementación y por tanto, la construcción del producto software.

El módulo de programación del que se hablado en este documento, se dividirá en dos paquetes organizacionales; el primero concentrará todos los elementos que conformarán la gestión del Registro de Actividades Diarias y sus actividades de inserción y actualización; el segundo integrará las partes que gestionarán la información que se almacena en el Registro de Actividades Diarias, a manera de reportes.

Estos paquetes de organización o programación son presentados en el siguiente diagrama:

Fig. 4.10 Paquetes de Organización de la Modelación del Diseño.

En los diagramas de clases de los paquetes organizacionales de este módulo se presentarán clases ya desarrolladas para otras aplicaciones realizadas por la empresa y que serán ahora reutilizadas en este sistema. Estas clases han sido reformadas para adaptarlas a las condiciones de esta aplicación y su uso es generalizado para todos los demás módulos de programación de este proyecto.

En los diagramas de clases correspondientes a cada paquete, esas clases reutilizadas aparecerán resaltadas en comparación con las demás.

También existen clases de uso común para ambos paquetes, creadas para este producto, estás clases estarán distribuidas en las diferentes capas de la aplicación y funcionan como “agentes” o “fachadas” (en este trabajo así es como se le nombrarán).

Los agentes o fachadas son clases que sirven como intermediaras para la gestión de mensajes entre las clases, por tanto, de información útil para las diferentes capas de la arquitectura de desarrollo de la aplicación.

Los métodos, que se presenten seguidamente, pueden sufrir cambios efectuados durante el proceso de implementación del producto, aunque estos cambios se reflejarán en la documentación del producto no podrá ser así en el presente documento.

4.3.1.1 Paquete # 1 “Registro de Actividades Diarias”.

El diagrama de clases, para este paquete, es el que se muestra en la figura siguiente:

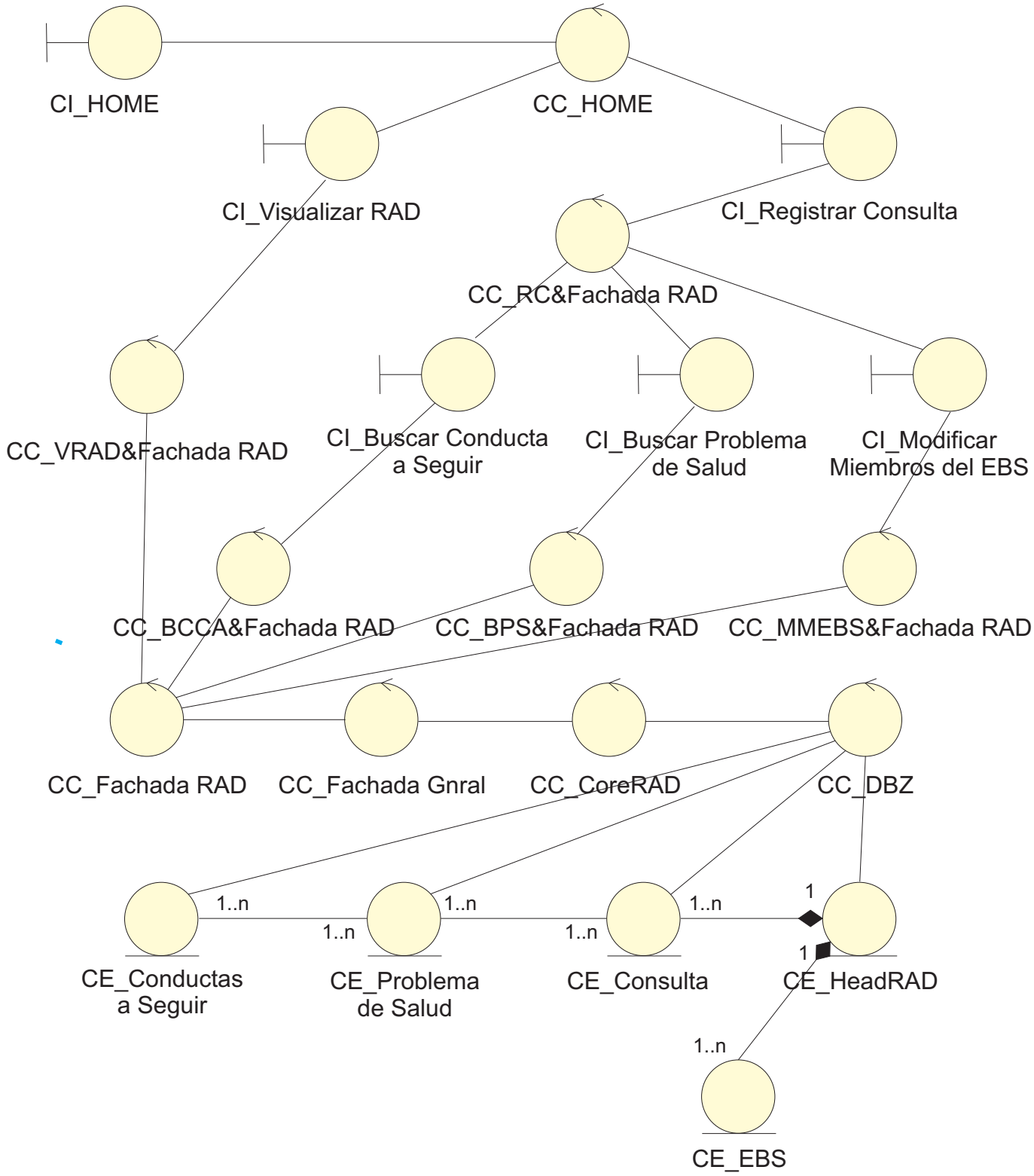


Fig. 4.11 Diagrama de clases Paquete # 1 “Registro de Actividades Diarias”.

4.3.1.2 Paquete # 2 “**Reportes**”.

El diagrama de clases, para este paquete, es el que se muestra en la figura:

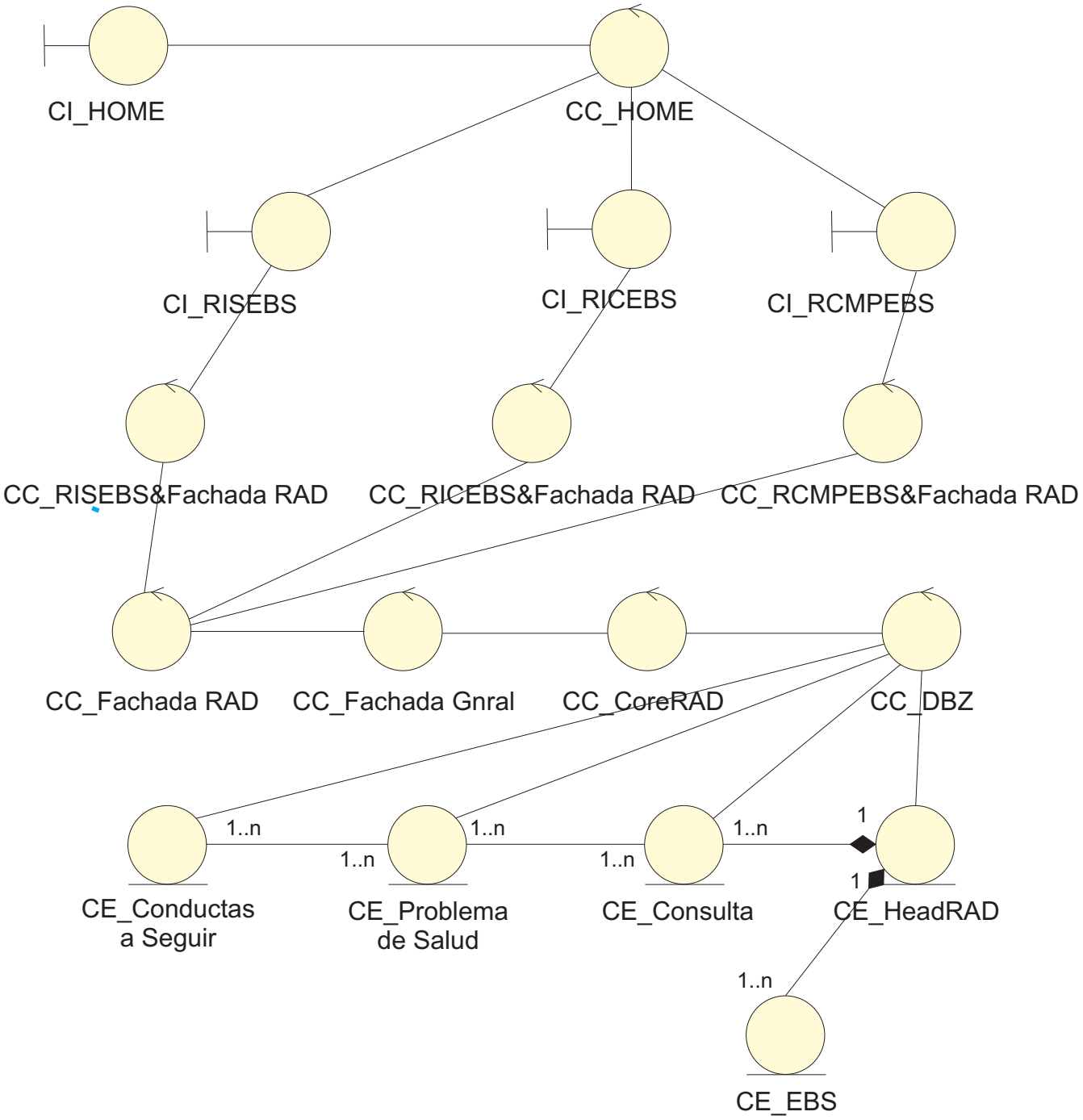


Fig. 4.12 Diagrama de clases Paquete # 1 “Registro de Actividades Diarias”.

En la herramienta utilizada para la modelación de estos artefactos o diagramas, se reflejaron los métodos y atributos principales de las clases, que se presentaron anteriormente. Estos diagramas completos están presentes en la documentación oficial del producto, perteneciente a la empresa que solicitó los servicios de diplomantes del ISPJAE para que estos realizaran el análisis ingenieril de este producto informático.

4.3.2 Diseño de la Base de Datos.

Para diseñar la base de datos del sistema, utilizamos el diagrama de clases persistentes y el modelo de datos, que están basados en la modelación de las clases del epígrafe anterior. Algunas de las clases representaban los datos que se obtienen y almacenan durante los procesos de la aplicación, estos son lo que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará el diseño de la lógica de negocio de la aplicación. Se presentarán entonces los diagramas de clases persistentes y el de modelo de datos.

La persistenacia es la capacidad de un objeto para existir fuera de un programa, proceso, función o hilo de control; de manera que se conserva su estado y su comportamiento.

A partir del desarrollo de estos modelos se obtendrá las base de datos con la cual trabajará nuestro sistema, por tanto se tendrán suficientes elementos para comenzar a realizar las versiones necesarias y suficientes del producto.

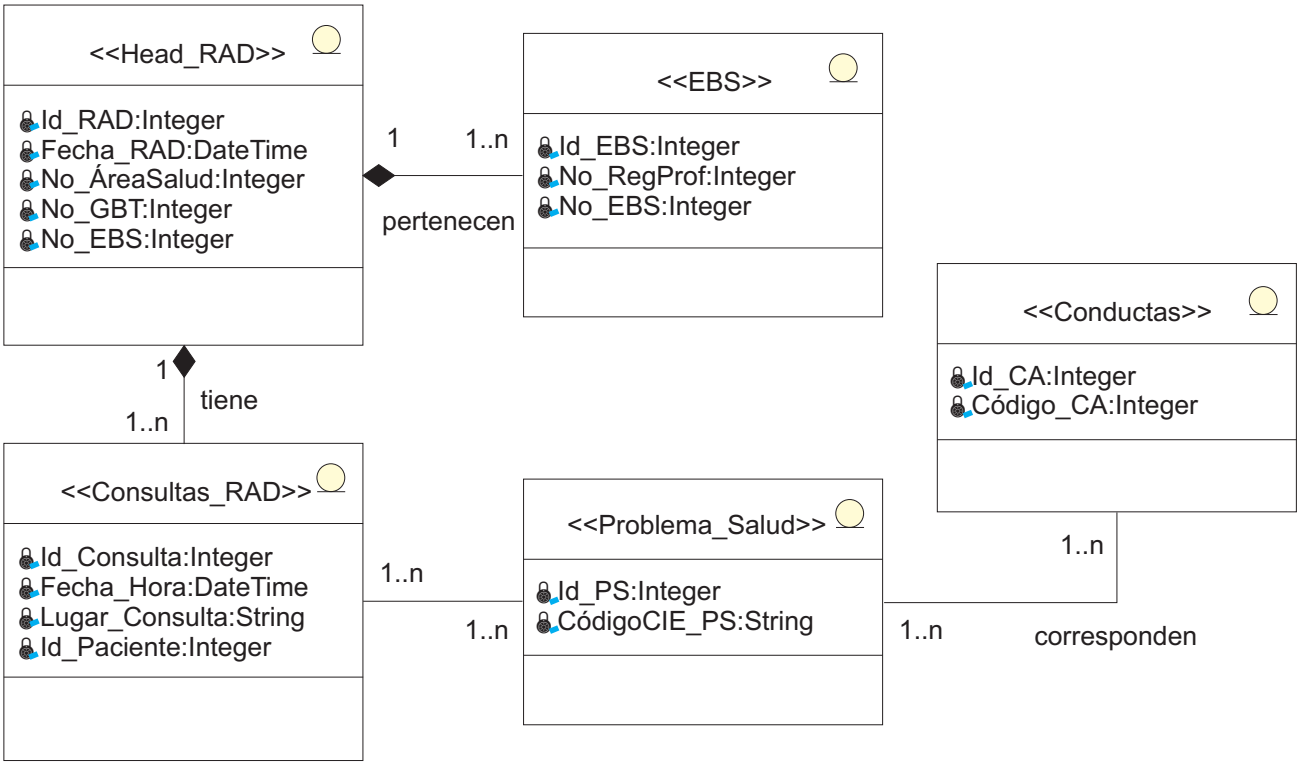


Fig. 4.13 Diagrama de clases persistentes.

Como se muestra en la figura anterior muchos atributos de las clases entidades del diagrama de clases persistentes son solo datos de indexación, respondiendo al principo de una base de datos distribuidas como la que se quiere diseñar para este módulo, en correspondencia con los requisitos arquitectónicos de la aplicación.

Los datos que complementan estas clases persistentes pertenecen o estarán alojados en otros módulos, en sus capas de datos para ser más específicos y nos brindarán estos datos a manera de servicios Web XML o XML Web Service .

Tomando como base el diagrama de clases persistentes de la figura 4.12 se generará el modelo de datos de nuestra aplicación, además de las tablas que se elaborarán a partir de las relaciones entre las clases de la figura se incorporarán otras en cumplimiento con los requerimientos de seguridad, auditori a y funcionalidad que registrarán, a manera de trazas de navegación, los datos que modifiquen los usuarios del software en los registros de la base de datos del sistema.

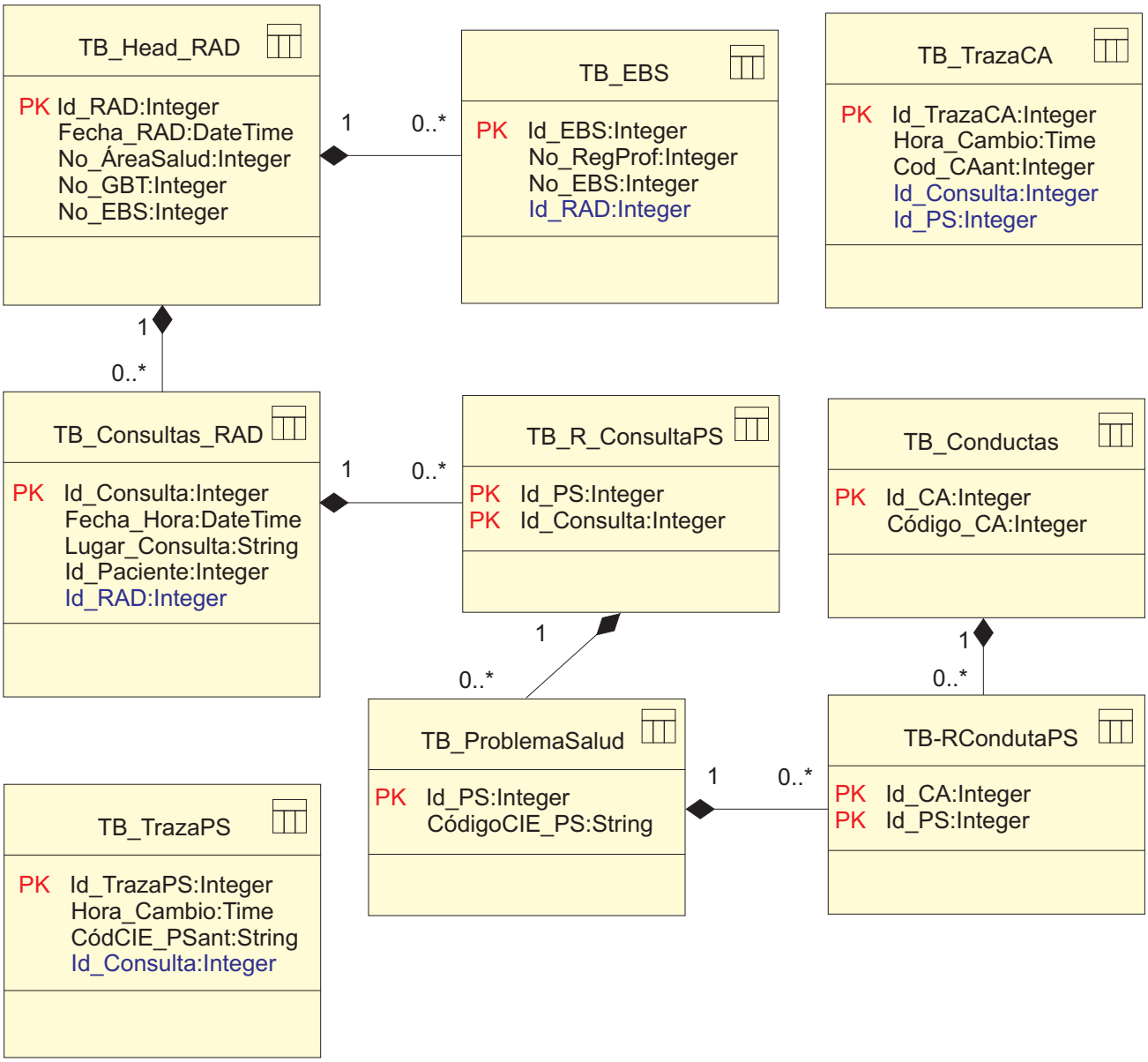


Fig. 4.14 Diagrama del Modelo de Datos del Sistema.

Las tablas TB_TrazaPS y TB_TrazaCA son las trazas del estado anterior, ante cada modificación de los problemas de salud y conductas a seguir respectivamente que se le diagnosticaron al paciente; como se puede observar estas no se relacionan con ninguna otra del Modelo de Datos, en ellas nuestra aplicación solo inserta nuevos registros, jamás los modifica, ni los utiliza para nada. La justificación de la existencia de estas es que uno de los posibles módulos que se le adicionen al Sistema Distribuido de Salud para la Gestión de la Atención Primaria es un módulo de auditoría de los servicios médicos y para la evaluación de la profesionalidad del personal de salud, donde los datos que en estas tablas se almacenan tendrían un valor y uso relevante.

4.3.3 Principios de Diseño.

El diseño, sea cual sea el objeto del mismo, tiene que basarse en el usuario y caracterizar a la empresa que ha desarrollado el producto, cualesquiera que este sea; en nuestro caso nuestros usuarios y clientes serán miembros de distintos niveles del personal de salud, o sea médicos, enfermeras , técnicos es distintas especialidades, personal administrativo, entre otros; muchos de ellos sin una preparación en las cuestiones de la informática. Para ello, este sistema utiliza ciertos principios generales que garantizan la usabilidad en los diseños para aplicaciones Web.

1. Principio de uso equiparable: donde las características de privacidad, garantía y seguridad estén igualmente disponibles para todos los usuarios, y que el diseño sea atractivo para todos los usuarios.

2. Principio de la flexibilidad: donde se ofrezcan posibilidades de elección en los métodos de uso, que facilite al usuario la exactitud y precisión, y se adapte al paso o ritmo del usuario.

3. Principio de la Información perceptible: donde se usen diferentes modos para presentar de manera redundante la información esencial (gráfica y verbal), se proporcione contraste suficiente entre la información esencial y sus alrededores, se amplíe la legibilidad de la información esencial, y que diferencie los elementos en formas que puedan ser descritas (por ejemplo, para las funciones de catalogación).

4. Principio de tolerancia al error: donde se dispongan los elementos para minimizar los riesgos y errores, por ejemplo utilizando elementos comunes; y los elementos peligrosos eliminados, aislados o tapados, que se proporcionen advertencias sobre peligros y errores. Hay que posibilitar el descubrimiento interactivo y el aprendizaje ensayo-error, y posibilitar la reversibilidad y la recuperabilidad de las acciones.

5. Principio de esfuerzo de acceso y uso: que minimicen las acciones repetitivas, y que proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie.

4.3.3.1 Estándares en la Interfaz de la aplicación.

Los estándares se refieren a elementos o funcionalidades comunes utilizados frecuentemente y que por la exposición, uso y finalmente la costumbre que generan en el público, los usuarios terminan identificándose y utilizándolos de modo espontáneo, asumiendo naturalmente sus características. Por el contrario, cuando un usuario espera que algo opere de un modo específico y esto no ocurre, se genera confusión y pérdida de confianza. También los estándares suelen ser normalizaciones internacionales que imponen los grupos o empresas dominantes en el mercado.

Estos estándares y convenciones constituyen patrones de diseño, que contribuyen a la generación de modelos mentales respecto a elementos específicos o incluso a tipos de sitios en general. La importancia de los patrones radica en la automatización del uso de las interfaces e interacciones, lo que implica la realización de tareas y permite que el usuario se concentre en los aspectos más importantes, más que en cómo resolver los problemas más básicos, como lograr completar correctamente una tarea específica.

El uso de estándares o patrones garantiza que los usuarios se sentirán en control, sabrán qué esperar de una tarea particular, se sentirán cómodos y utilizarán de un modo más apropiado una aplicación web o cualquier otro tipo de sistema.

La lógica detrás de un patrón de interacción está dada por la construcción de modelos mentales, que nos permiten identificar y reconocer fácilmente aquellos elementos que hemos aprendido y asociado a través de la experiencia. En la medida que este proceso se concreta, se eliminan una serie de factores innecesarios, asociados a procesos congnitivos que se traducen en que nos enfrentamos a esquemas similares y conocidos de resolver un problema. Este proceso nos permite pasar de un mecanismo que podría tomar algunos segundos, a un reconocimiento que ocurre en fracciones de segundos. Esto no es irrelevente si pensamos en que estos segundos de diferencia, y la calidad de la experiencia que ellos constituyen, tienen una relación directa con el éxito de una venta en línea o el envío correcto de un mensaje de contacto para un potencial negocio.

1- Estándares Web y usabilidad. Se tendrá en cuenta diversos factores para el óptimo funcionamiento de la aplicación:

- Tiempos de carga, donde se tiene en cuenta que la mayoría de los usuarios tienen conectividades dial-up de 56 k. Lo que otorga una velocidad de descarga real de entre 3 y 4 kbps.

- Correcto funcionamiento de links y secciones: para evitar el típico error "HTTP 404 - Not found" que surge cuando hay links mal asignados.

- Depuración de código del lado del cliente (javascript, flash) y del lado del servidor (scripts de proceso server-side: cgi, php, asp, jsp).

- Se contemplan las resoluciones de pantalla que utilizan los usuarios, para que el maquetado del sitio se vea de manera correcta no solo en 800 x 600 px., y 1024 x 800 px. sino en todas las resoluciones simultáneamente.

- Se busca una ámplia compatibilidad con la mayoría de los navegadores de versiones actualizadas.

- Se valida el código según los estándares propuestos por W3C (World Wide Web Consortium) en cuanto a la codificación html-xml y las hojas de estilo en cascada (CSS).

2- Contenidos textuales de una Aplicación Web. El formato de textos es un factor muy importante; se busca otorgar la mejor información y objetividad en un marco de coherencia. Lo que destaca muchas veces a las aplicaciones exitosas es un contenido original muy bien elaborado que no se encuentra en otras aplicaciones.

3- Navegabilidad. Toda la información y opciones de la aplicación deben estar bien organizadas y fácilmente accesible, para que los usuarios puedan encontrar lo que buscan rápidamente y hacer lo que “quieran” sin muchas complicaciones.

Esto se logra con un menú bien organizado, y también se implementan mapas de sitio, mostrar contenidos clasificados por temáticas y buscadores internos, para la búsqueda de términos clave.

4- Interactividad. Deben agregarse aplicaciones que brinden servicios útiles a los usuarios e información extra; cuanto más interactivo sea una aplicación Web, más directo será el contacto que tendremos con los clientes y usuarios.

5- Estética de la Aplicación y diseño gráfico. Se opta por diseños sencillos, funcionales y livianos.

Para lograr una mayor eficiencia en el proceso de trabajo, y sobre todo para lograr una coherencia formal entre todos los módulos del sistema, y que sean identificados así como parte de un todo, se han pautado una serie de elementos comunes que facilitarán su reconocimiento y el uso que se haga de ellos.

Se diseñará una Pantalla Inicial global del Sistema Integral de Salud, desde la cual se accederá a los diferentes módulos del RIS, del SIAP y del SIGH. Esta pantalla contará con accesos a los diferentes módulos, informaciones generales, guías de ayuda, sistema de avisos que genera cada registro y enlaces definidos.

Así mismo será diseñada una Pantalla Inicial para cada una de las aplicaciones, que contará con accesos a todas las utilidades, avisos, ayuda y un enlace para regresar regresar a la Pantalla Inicial del Sistema Integral de Salud.

La estructura base de las aplicaciones es la misma para todos los módulos: las pantallas más usadas, los modelos establecidos, las rutas de navegación, las utilidades básicas, la organización de los elementos en pantalla y el diseño de identificadores serán comunes para todos.

Para particularizar el diseño de cada módulo se ha definido entonces una pauta de dos colores básicos para cada uno, con sus degradaciones hacia blanco y negro, así como la diferenciación por logotipo e imagen principal del cabezal, que identificará a cada módulo.

Su diseño está determinado fundamentalmente por el principio de la usabilidad, teniendo en cuenta que no se trata de un sitio web, sino de una aplicación de trabajo donde el diseño tiene como principal propósito facilitar su uso, comprensión y navegación, por encima de ornamentos inútiles, aunque manteniendo pautas estéticas, orgánicas y agradables.

Formalmente, usabilidad se define como la medida en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción, en un contexto de uso especificado [ISO 9241-11].

La resolución óptima para la cual están diseñadas las aplicaciones es de 800 x 600 pixels (px). El fondo siempre será blanco y los elementos de pantalla de los colores definidos para cada módulo.

Se ha definido un cabezal pequeño de 65 px de altura, más pequeño que el utilizado en las páginas web, que recomiendan cabezales de hasta 80 px de altura.

El menú principal siempre estará situado en una barra superior horizontal de solo 15 px de altura. No existirá barra vertical de menú situada la izquierda de la página (como usualmente se hace) para ampliar el espacio de trabajo, pues estará reservado lo más amplio posible para la inserción de grandes tablas y formularios que

Constituyen la base fundamental de estas aplicaciones.

El logo siempre estará ubicado en el extremo superior izquierdo de la página, es una imagen que cuenta con un ancho de 270 px y se corresponde con el nombre de cada módulo. Estará constituido por un juego tipográfico en Frankling Gothic Medium, y en el caso de las aplicaciones propias del Proyecto APS, estando especificado dentro del logo como una especie de genérico.

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

La tipografía será siempre Tahoma, por su amplia legibilidad y por las facilidades conocidas que brinda para la lectura digital. El menú principal será a 7 puntos y los submenús a 6 ptos. Los demás puntajes se definirían en dependencia de las necesidades puntuales de cada pantalla.

El espacio de trabajo comienza 33 px por debajo del menú. El espacio intermedio que queda es también con fondo blanco y está reservado para el texto de ubicación dentro del sitio (justificado a la izquierda) y para ubicar los botones propios de la pantalla (justificados a la derecha). Estos se organizarán en una o dos filas, de hasta cuatro botones (13 x 72 px) cada una. Los botones se corresponden también con los colores pautados.

Entre los elementos comunes del menú principal se encuentran Inicio para regresar a la página inicial del módulo, Salir para desconectarse del sistema, y otros módulos para facilitar los enlaces a otros módulos necesarios. Son también comunes a casi todos los botones del menú principal Configurar para la configuración de codificadores, Cierre para la realización de cierre estadístico y Reportes para generar reportes de actividades u operaciones.

Es común para todos los módulos el diseño de una serie de ventanas, en las que solo cambiarían los colores, en dependencia de cada uno. Son estas las ventanas de precaución, error, validación de datos, etc.

En cuanto a los elementos de diseño del interior de las pantallas, es decir, de las tablas, formularios, etc., se definen los edit que se utilicen con una altura de 16 px y la separación entre estos y entre ellos y los bordes de tablas será de 8 px. Será de 8 px la separación entre el texto y el edit. Los textos de estos campos serán justificados siempre a la derecha, es decir, justificados a 8 ptos de cada edit.

En el caso de tablas generadas por búsquedas, que ordenan una serie de elementos, y necesiten selección, se harán a través de checkboxs justificados a la izquierda de la tabla. Siempre habrá un checkbox en la fila de título, también a la izquierda, que facilite seleccionar todos. Es necesario destacar que estas tablas pueden tener una cantidad grande de líneas generadas por la búsqueda, por lo que debe quedar pautado que hasta 25 resultados la tabla funcione con scroll, pero más de esta cantidad será entonces por paginado, al estilo de Google, con 25 resultados por página.

El interés general es mantener el diseño y la estructura del sitio lo más simple posible, la simplicidad es entendimiento del contenido, de la estructura, es facilidad para encontrar lo que se busca, es también velocidad de descarga.

A continuación se muestra una representación gráfica de las pautas generales de la organización de elementos en pantalla:

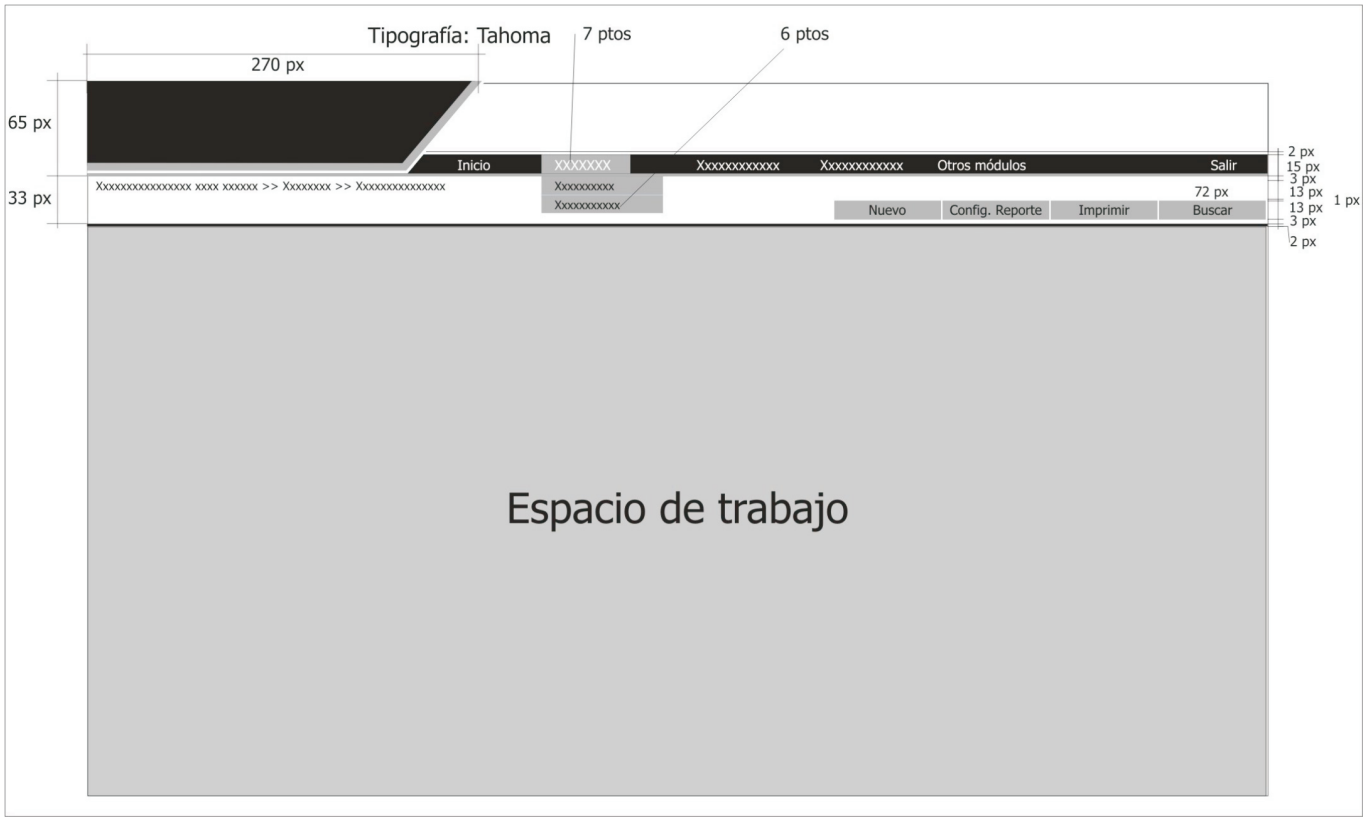


Fig. 4.15 Formato para el diseño de las pantallas o interfaces de usuario.

4.3.3.2 Concepción General de la Ayuda.

Las ayudas en las aplicaciones Web dinámicas y en todas las aplicaciones Web en general no suelen ser explicaciones detallistas del sistema informático al cual representan como sucede en la acostumbradas aplicaciones de Escritorio, generalmente son simples aclaraciones, informaciones generales de la aplicación o datos de la empresa que le da soporte o realizó el producto; este comportamiento es debido al propio dinamismo que encierran las aplicaciones Web, lo cual provoca la caducidad casi inmediata de cualquier información que de forma persistente que se sitúe en esta opción indispensable e inherente a un sistema informático.

En el módulo de Registro de las Actividades Diarias, que para el Sistema Distribuido de Salud de Atención Primaria se está desarrollando, igualmente se comportará como los estándares del mercado para este tipo de sistemas, lo cual es extensible para todos los demás modulos de este producto; es por ello que la ayuda para este sistema informático estará concebida bajo los principios del soporte técnico en línea, la confección de manuales de usuario y la capacitación técnica directa ante la implantación del producto.

El soporte técnico en línea es una práctica muy utilizada en las aplicaciones Web dinámicas, como la que se está desarrollando, generalmente se implementan con una explicación general y opciones o vínculos a sistemas de correo o a otros sitios Web fuera de la aplicación principal para que los usuarios puedan informar acerca de errores que suceden en la aplicación, dar sugerencias de la funcionalidad de este o recibir soluciones a las preguntas que de forma “directa” pueden realizar a los administradores y creadores del producto.

Esta forma de ayuda resulta de gran ventaja para los desarrolladores del sistema ya que contribuye a la resolución de problemas del software, la gestión de cambios y configuraciones de este y la actualización y el mantenimiento del producto, además les permite mantener un contacto directo y continuado con los usuarios y clientes; por otra parte esta forma de ayuda no resulta del agrado extendido para todo el universo de usuarios ya que generalmente deben esperar un tiempo determinado para recibir la respuesta a sus cuestionamientos, sin embargo no tiene detractores platónicos porque las respuestas suelen ser personalizadas, mucho más consistentes y útiles no solo pra el que la solicita sino también para otros usuarios ya que las respuestas y preguntas antes realizadas se suman a la información que se presenta.

La ayuda que le será entregada en primera instancia a los clientes y usuarios del módulo será un manual de usuario en formato digital o impreso que explicará deforma detallada las principales funcionalidades y opciones que brinda el software.

Además está concebida por parte de la empresa productora, bajo acuerdo con los clientes del producto la puesta en marcha de cursos de capacitación o entrenamiento en el uso de la aplicación, dirigida a todos los usuarios potenciales antes y durante de la implantación oficial de este producto en el Sistema Nacional de Salud.

4.3.3.3 Tratamiento de Excepciones.

Las excepciones son condiciones excepcionales que pueden ocurrir dentro del programa durante su ejecución (por ejemplo una división por cero, se agote la memoria disponible, se pierda la comunicación entre componentes, no se produzca el resultado esperado luego de alguna petición, etc.) y que requieren recursos especiales para su control.

El manejo correcto de la excepciones y su tratamiento desde la concepción del producto garantizan un aumento considerable de la calidad de las aplicaciones que se desarrollen.

El funcionamiento general del mecanismo de lanzamiento y tratamiento de excepciones es el siguiente: Existe un método que invoca la ejecución de otro luego este método más interno se encuentra en una situación que puede considerarse como excepcional por lo tanto lanza una excepción, en este momento termina la ejecución del método más interno y se retorna inmediatamente al método que lo invocó, el método llamador debe capturar la excepción y la trata (parte del tratamiento de la excepción puede ser volver a lanzarla al método que invocó a este).

La correcta programación de excepciones significa diseñar los algoritmos pensando únicamente en la forma habitual en la que deben ejecutarse, manejando las situaciones extraordinarias a parte. De esta manera se consigue un diseño mucho más estructurado, legible, robusto y fácil de mantener.

Para depurar los errores se hará utilizando JavaScript. Por medio de este lenguaje serán informados la mayoría de los errores de la página, como apoyo a las validaciones de entrada de datos, garantizando que los datos introducidos por los usuarios sean validos, o les sea posible corregirlos en caso contrario.

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

FaultCode:

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

Código del proyecto-código del modulo (:) número del método (.) número del error. Ejemplo: APS-RAD: 1.5 que indica error 5 en el método 1 del módulo Registro de Actividades Diarias perteneciente al Proyecto APS.

FaulStrign:

Una explicación del error asequible al humano (leíble). Debe tenerse en cuenta que este texto puede ser mostrado al operador final del sistema. Ejemplo: Formato de entrada no válido para la fecha del registro de actividades diarias que quiere mostrar.

FaultActor:

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

Detail:

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

También se utilizaron codificadores para evitar posibles errores por parte del usuario al registrar información de poca variabilidad, se pueden citar los casos de los datos de los pacientes y del Equipo Básico de salud, la descripción y código de los problemas de salud y las conductas a seguir, los indicadores de salud para los reportes, entre otros datos registrados en el Registro de Actividades Diario.

4.3.4 Estándares de Codificación.

Muchos webmasters, a lo largo de su carrera acostumbran a utilizar editores visuales html, como por ejemplo Dreamweaver o Frontpage. Lo cierto es que lo más importante es conocer a fondo las reglas de codificación html estándar y revisar el código manualmente para corregir los errores que estos editores visuales provocan. Cuando este paso es omitido (o ignorado) en la etapa de desarrollo, el resultado es que se obtienen sitios web con mucho código basura, etiquetas mal anidadas o con una asignación errónea, etc. Todo esto provoca incompatibilidades con muchos navegadores, carga exesiva de las páginas, errores de validación... En resumen: aplicaciones Web que no son eficientes tecnológicamente hablando.

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

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

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

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

Todos los campos id van a comenzar con el identificador (id) seguido del nombre del campo. Ejemplo id_enfermedad.

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

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

En el caso de las clases y sus métodos no se usarán abreviaciones y las palabras continuas deben comenzar con mayúsculas. Ejemplo ListarTotalPersonalSalud

Para comentar el código se utilizarán, en el caso de una línea al final de la misma el carácter “// “ y seguido el comentario, y en el caso de un método se harán debajo del mismo en bloques utilizando los caracteres “/* */”.

Se colocará el corchete de apertura, aunque {, en la línea siguiente a la instrucción y alinee con este y con el corchete de cierre }. El código dentro del bloque se debe indentar con el tabulador, para permitir mayor legibilidad se recomienda configurar el editor para que despliegue el tabulador con una longitud de cuatro espacios. Se

utilizarán corchetes en todas las sentencias que lo requieran aunque estas contegan una sola línea de código.

El idioma de las clases auxiliares como sesión, error, será el inglés para garantizar la homogeneidad con las programadas en este ámbito en el mundo, en el caso de los Servicios Web y la interfase de administración se usarán el español para esclarecer los objetivos de cada método o script a utilizar.

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

El almacenamiento de la información será en scripts SQL para construir la base de datos e interactuar con ella desde las aplicaciones.

Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

En las consultas de inserción se debe colocar siempre el nombre de los campos en los cuales desea escribir en la tabla.

Los nombres de las tablas debe ir en minúsculas y cada palabra separada por línea abajo(guión bajo) "_".

Las llaves primarias tendrán la palabra "id" unido con el nombre de la tabla, con una línea abajo (ej. id_nombre_tabla).

Las llaves foráneas tendrán la palabra "fk" unido con el nombre de la tabla a la que hace referencia, con una línea abajo (ej. fk_tabla_referenciada).

En el caso de los ficheros xsl deben tener por defecto el nombre del fichero php que lo llamará.

Control	Prefijo	Ejemplo
Botón	Btn	btnAceptar
Etiqueta	Lbl	lblNombre
Lista/Menú	Mn	mnPrincipal
Campo de Texto	Txt	TxtFecha
Botón de Opción	Opt	OptSexo
Casilla de Verificación	Chx	ChxBorrar

Tabla. 4.2 Normas de Codificación de formularios

Las páginas HTML se harán sin incluir código, las funciones JavaScript que se usarán en varias páginas se escribirán dentro de ficheros .js

4.3.5 Diagrama de Componentes del módulo “Registro de Actividades Diarias”.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

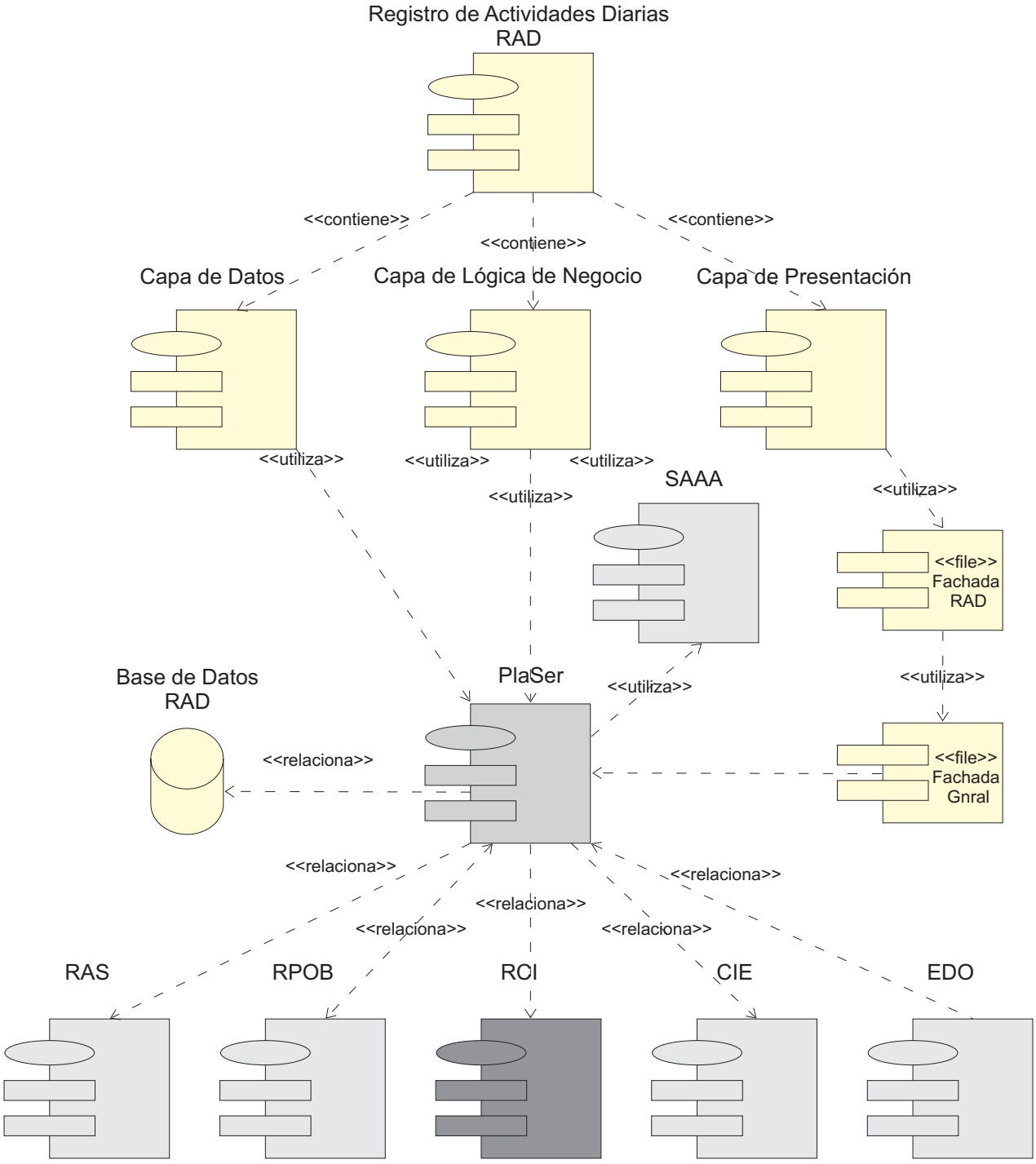


Fig. 4.16 Diagrama General de Componentes del Sistema.

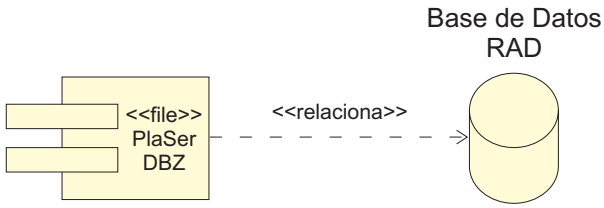


Fig. 4.17 Diagrama de Componentes de la Capa de Datos del Sistema.

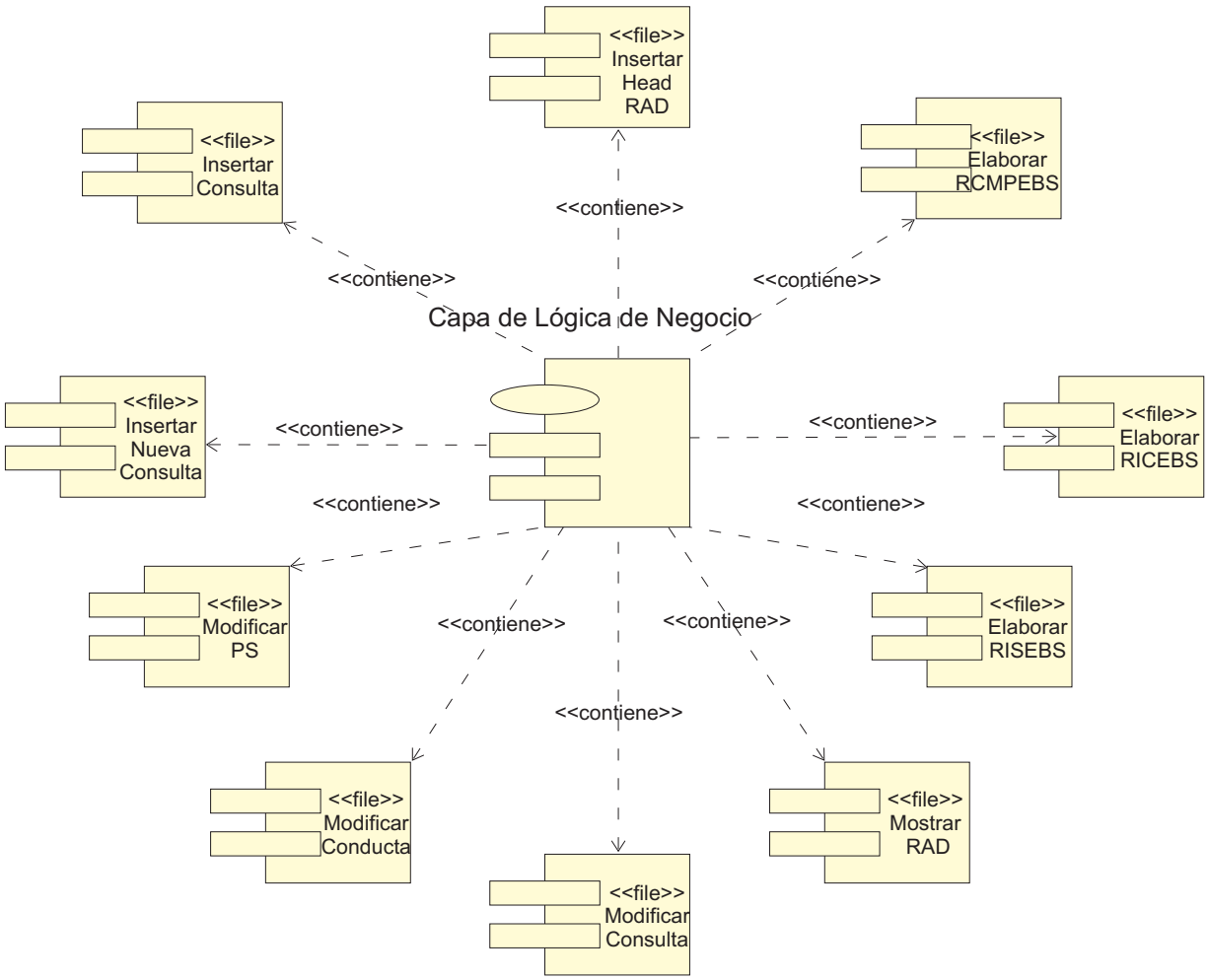


Fig. 4.18 Diagrama de Componentes de la Capa de Lógica de Negocio del Sistema.

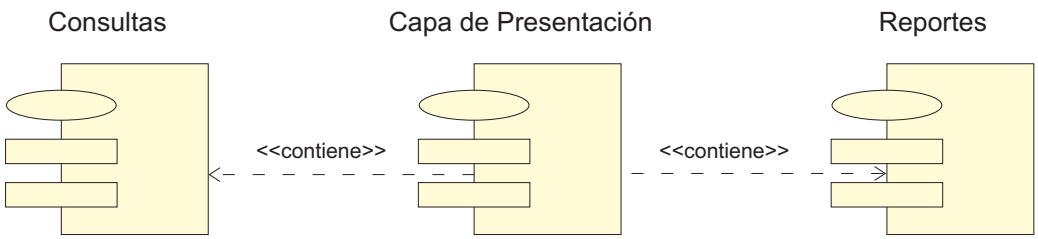


Fig. 4.19 Diagrama de Componentes de la Capa de Lógica de Negocio del Sistema (vista general).

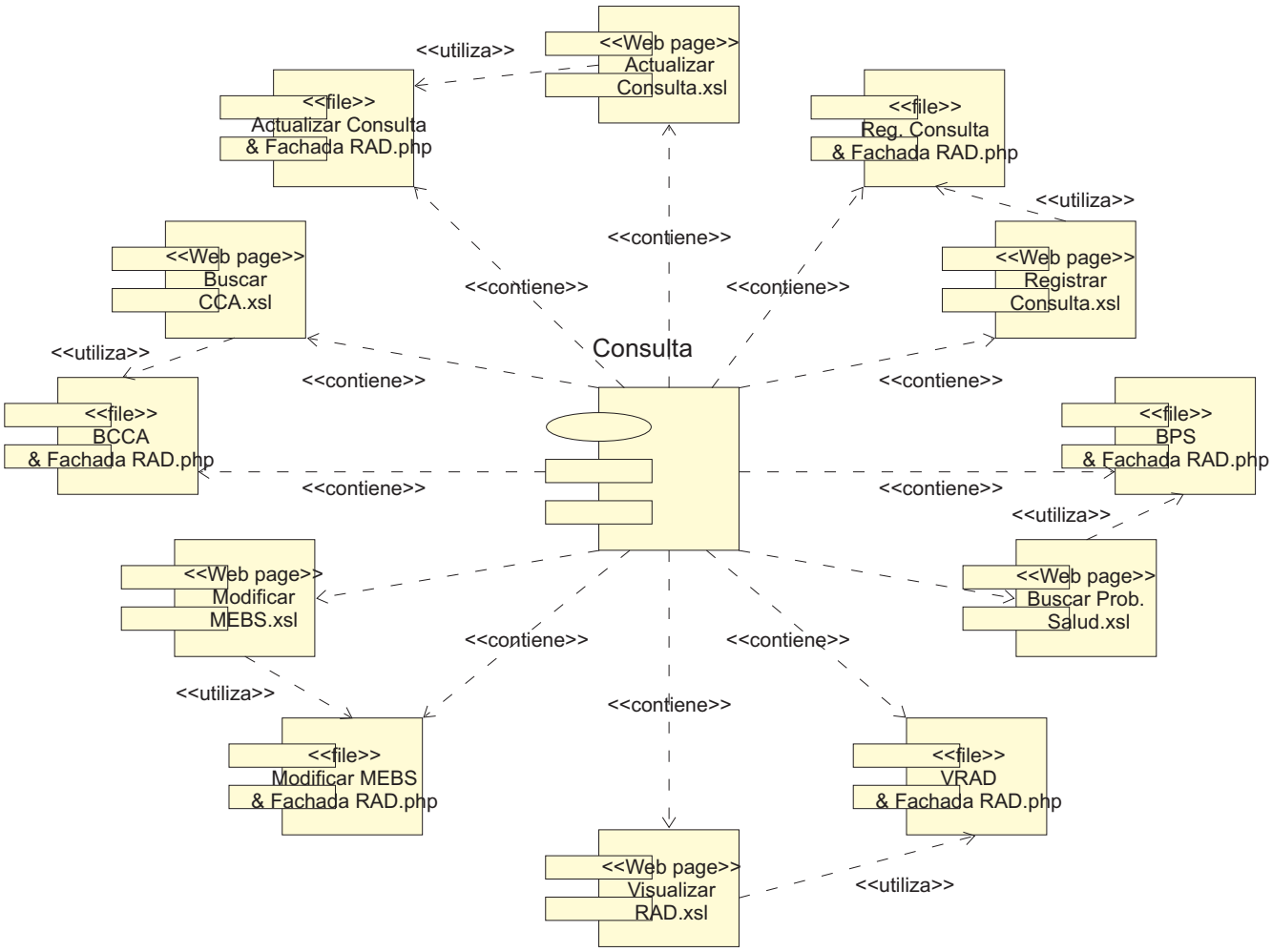


Fig. 4.20 Diagrama de Componentes de la Capa de Presentación del Sistema (extendido).

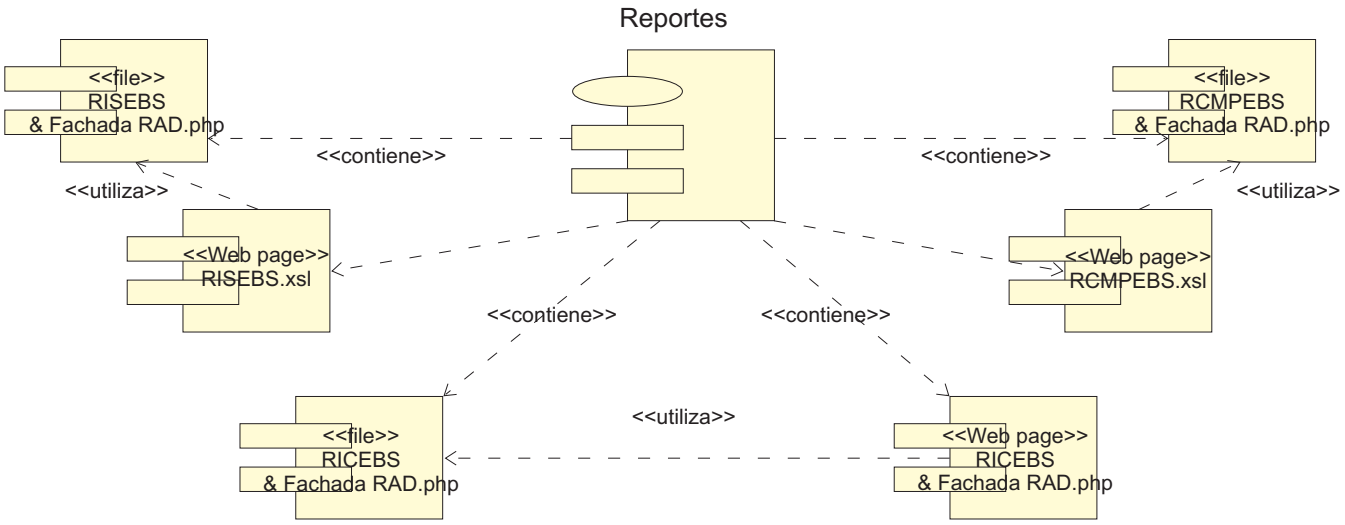


Fig. 4.21 Diagrama de Componentes de la Capa de Presentación del Sistema (extendido).

4.3.5.1 Plataforma de Servicio “PlaSer”.

PlaSer, acrónimo de Plataforma de Servicio, constituye una plataforma sobre la que se pueden desplegar aplicaciones XML – Web Services. Este sistema esta concebido completamente sobre Arquitectura Basada en Componentes y Orientada a Servicios, usando el paradigma de XML Web Services, específicamente SOAP. En su concepción se han utilizado estándares actuales y normas abiertas. Todo el código ha sido programado en PHP.

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

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

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

PlaSer, en lo fundamental, esta integrado por varias clases desarrolladas en PHP, una librería, que puede o no ser usada para que un componente se integre, pero que de no ser usada la seguridad corre por parte del programador.

En esta versión de PlaSer solo soporta como llamada RPC el protocolo SOAP, pero en futuras versiones se piensa en otros protocolos de transportes o incluso el acceso local a código a nivel de File System, de forma tal que para el programador sea totalmente transparente si la invocación del proceso es remoto, local, por SOAP, directamente a código, etc.

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 Autorization Autentication and Account) componente de seguridad, que autentifica y autoriza, además tiene registrados los nombres de los métodos. Y el tercer servidor contiene las bases de datos desarrolladas en MySQL.Para aumentar la seguridad se pudiera colocar a ProxPla solo en un cuarto servidor separado de lo demás componentes.

También PlaSer permite ser desplegado totalmente en un solo servidor, aunque esto no es recomendable por motivos de seguridad.

Cuando un cliente web (navegador) o una aplicación específica realiza una petición de acceso a alguno de los módulos instalados sobre PlaSer, dicha petición es atendida por plaser.php; una vez que este haya verificado todo lo concerniente a la seguridad con el componente SAAA, entonces se ejecuta el código correspondiente a la aplicación solicitada.

Si es necesario acceder a una base de datos esto se hace a través de la clase DBZ.

Como se puede apreciar el código de los módulos o aplicaciones instalados sobre PlaSer esta completamente “rodeado” por las clases de PlaSer, lo que contribuye a elevar la seguridad del Sistema.

Los Sistemas que utilicen a PlaSer deben trabajar con Arquitectura Basada en Componentes donde exista el concepto de módulo para agrupar un conjunto de componentes que dan una funcionalidad o servicio integrado. Cada modulo (como concepto global) tiene su contraparte hecha en la Capa de Presentación.

Es una arquitectura de componentes independientes que puede ser interpretada por capas, para esto se crea una tubería que fuese algo como esto:

browser->web->proxpla->core->...->core->db

Note que 'core->...->core->' lo que significa es que hay muchos procesos que trabajan en cascada, o sea, componentes internos hacen uso de otros componentes, todos se hace usando PLASER transportando los mensajes en SOAP.

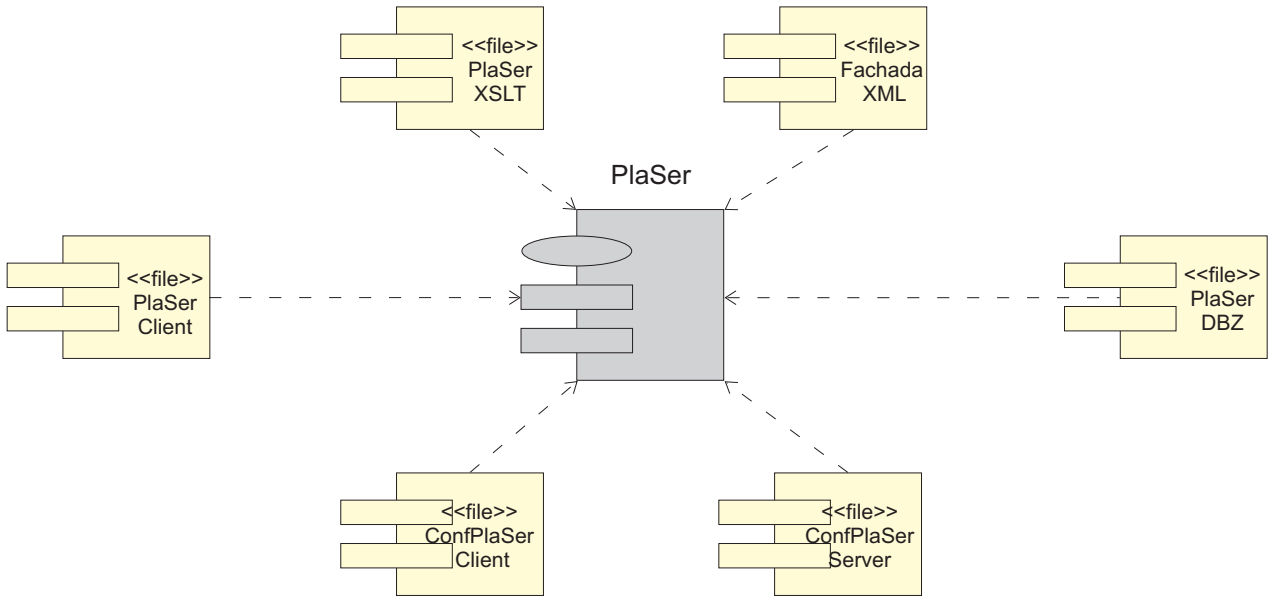


Fig. 4.22 Diagrama de Componentes de PlaSer.

4.3.6 Modelo de Despliegue.

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instacias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo (Dispositivos, Procesadores, ,Memoria).

Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez

Esta vista permite determinar las consecuencias de la distribución y la asignación de recursos. Las instancias de los nodos pueden contener instancias de ejecución, como instancias de componentes y objetos. El modelo puede mostrar dependencias entre las instancias y sus interfaces, y también modelar la migración de entidades entre nodos u otros contenedores.
Esta vista tiene una forma de descriptor y otra de instancia. La forma de instancia muestra la localización de las instancias de los componentes específicos en instancias específicas del nodo como parte de una configuración del sistema. La forma de descriptor muestra qué tipo de componentes pueden subsistir en qué tipos de nodos y qué tipo de nodos se pueden conectar, de forma similar a una diagrama de clases, esta forma es menos común que la primera.

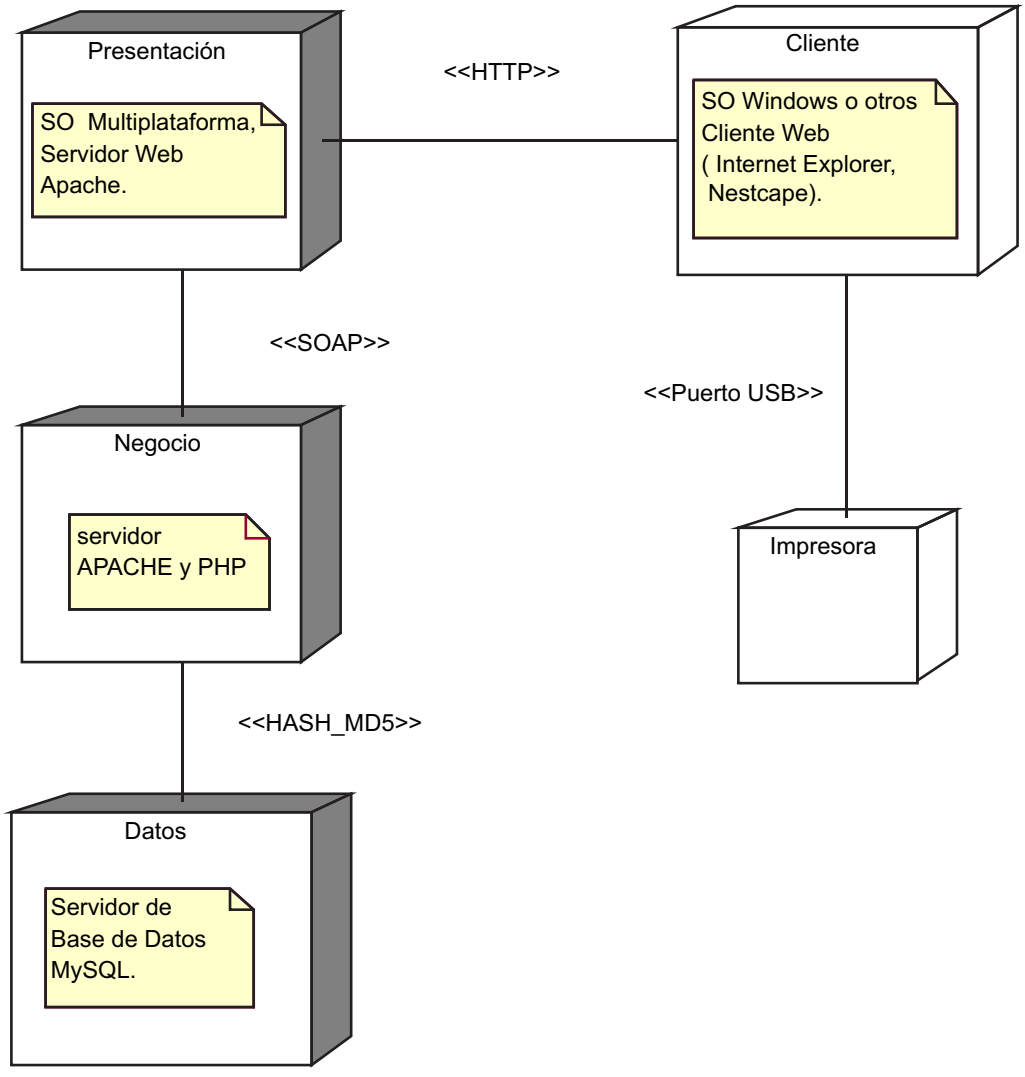


Fig. 4.23 Diagrama de Despliegue del Sistema.

4.4 Conclusiones del Capítulo...

Con todos los elementos mostrados hasta ahora, en este capítulo, el proyecto para la construcción del módulo de Registro de Actividades Diarias del Sistema Integral de Salud se encuentra listo para la confección de los contratos de trabajo o programación para que los programadores, diseñadores y demás participantes comiencen a desarrollar las primeras versiones del producto que se ha descrito hasta este momento.

Muchos de los artefactos que se han descrito pueden sufrir cambios generados durante el proceso de implementación, algo que no parecería extraño y está contemplado en el proceso de ingeniería como un caso común (gestión de cambios y configuraciones); aunque generalmente un diseño robusto es muy difícil que le sucedan cambios conceptuales o demasiados drásticos, la mayoría de los cambios podrían ocurrir a partir de la transformación de los requisitos funcionales y los no funcionales para el sistema.

Por otra parte el desarrollo de estos flujos de trabajo, en sus respectivas fases han generado la documentación necesaria , que la empresa desarrolladora necesita para la ejecución, construcción y posterior mantenimiento de este proyecto y en especial para el módulo diseñado y mostrado en este documento.

CAPÍTULO 5
Estudio de Factibilidad

INTRODUCCIÓN

La administración efectiva de un proyecto de software depende de planear completamente el progreso del proyecto. El administrador del proyecto debe anticiparse a los problemas que podrían surgir, así como preparar soluciones tentativas a esos problemas.

La planeación es un proceso iterativo que solamente se completa cuando el proyecto mismo se termina. El plan debe revisarse conforme la información se haga disponible. Cuando se modela un negocio, las metas globales del mismo son un factor importante que debe considerarse a medida que se formula el plan del proyecto. Según esto cambie, los cambios en el proyecto serán necesarios.

El proceso de planeación del proyecto inicia con una valoración de las restricciones que lo afectan (fecha de entrega requerida, personal disponible, presupuesto global, etc.). Esta se lleva a cabo con una estimación de los parámetros como su estructura, tamaño y distribución de funciones. Entonces se definen los hitos de progreso y productos a entregar. En ese momento, el proceso entra en un ciclo. Se prepara un calendario y las actividades definidas en el calendario inician o continúan. Después de algún tiempo (por lo general 2 o 3 semanas), se revisa el proyecto y señalan las discrepancias. Debido a que las estimaciones iniciales de los parámetros del proyecto son tentativas, el plan siempre deberá actualizarse.

Las estimaciones están asociadas con el esfuerzo y el tiempo con las actividades identificadas del proyecto. Los administradores deben estimar las respuestas a las siguientes preguntas:

¿Cuánto esfuerzo (personal necesario) se requiere para completar una actividad?

¿Cuánto tiempo se necesita para completar una actividad?

¿Cuál es el costo total de una actividad?

La estimación y la realización del cronograma de producción se llevan a cabo de forma conjunta. Sin embargo, en las primeras etapas del proceso productivo se requieren algunas estimaciones de costos, antes que se tenga el cronograma detallado. Estas estimaciones son necesarias para establecer un presupuesto para el proyecto o para asignar un precio para el software de un cliente.

Una vez que el proyecto se comienza a ejecutar, las estimaciones se actualizan de forma regular. Esto ayuda al proceso de planeación y permite la utilización efectiva de los recursos. Si el gasto real es significativamente más grande que las estimaciones, entonces el administrador del proyecto debe tomar algunas acciones. Estas pueden ser: solicitar recursos adicionales para el proyecto o modificar el trabajo a realizar.

Existen tres parámetros involucrados en el cálculo del costo total de un proyecto de

desarrollo de software: los costos de hardware y software, incluyendo el mantenimiento; los costos de viajes y capacitación; los costos de esfuerzo (los costos de pago a los ingenieros de software).

Para muchos proyectos, el costo dominante es el costo del esfuerzo. Las computadoras que son suficientemente poderosas como para desarrollar el software son relativamente baratas. Aunque los costos de viajes pueden ser importantes si un proyecto se desarrolla en sitios distintos, son relativamente bajos para muchos proyectos. Además, el uso de correo electrónico, los chat, fax y teleconferencias reduce los viajes requeridos.

Los costos del esfuerzo no son simplemente los relacionados a los salarios de los ingenieros de software involucrados en el proyecto. Las organizaciones calculan los costos de esfuerzo en función de los costos de sobrecarga donde se toma en cuenta el costo total de hacer funcionar la organización y dividen este entre el número de personas productivas. Por lo tanto, los siguientes gastos son parte de los costos de esfuerzo totales: los costos de proveer, climatizar e iluminar las oficinas; los costos del personal de apoyo como los contadores, secretarias, limpiadores y técnicos; los costos de las redes y comunicaciones; los costos de los recursos centralizados como las bibliotecas, los recursos recreativos, etc.; los costos de seguridad social y beneficio de empleados como las pensiones.

Debido a las consideraciones organizacionales involucradas, asignar precio del proyecto por lo general le concierne al administrador principal de la organización, así como a los administradores del proyecto de software.

El proceso de desarrollo de aplicaciones hipermedia al igual que el de cualquier producto de software, requiere la aplicación de métricas de estimación para garantizar resultados más precisos en su ciclo de vida.

El objetivo de este capítulo es hacer un estudio inicial de factibilidad para el proyecto que se desarrolla , donde se utilizará un método para la estimación del tamaño que tendrá el sistema o del esfuerzo que tomaría implementarlo.

Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de Función. Esta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas, denominadas Puntos de Función.

Por otro lado, el SEI (del inglés, Software Engineering Institute) propone desde hace algunos años un método para la estimación del esfuerzo llamado COCOMO II. Este método está basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code).

Este modelo, también conocido Modelo Constructivo de Costes, es utilizado para calcular la estimación del costo de un sistema, es uno de los métodos más difundidos por su flexibilidad y la posibilidad de utilización en cualquier tipo de proyecto, además es uno de los más documentados , así como fácil de utilizar.

5.1 RESULTADOS DEL CÁLCULO DEL ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO.

Uno de los factores importantes a tener en cuenta en el diseño o mejoramiento de un sistema de información o apoyo a la toma de decisiones, es si las ventajas del sistema propuesto justifican o no su costo. Como objetivo fundamental de estos sistemas está ofrecer la información adecuada en el momento que se solicite. Pero si los ahorros que se obtienen con la información no compensan su costo puede no ser rentable. Sin embargo la rentabilidad de un sistema de este tipo a veces resulta difícil de estimar, pues el valor de la información no es fácilmente cuantificable.

Haciendo uso del método constructivo de estimación de costes se determinaron algunos elementos necesarios que servirán de datos iniciales para hacer dicha estimación, entre estos elementos determinados fueron el cálculo de esfuerzo, el tiempo de desarrollo, la cantidad de hombres a utilizar y el costo del producto.

El cálculo del esfuerzo necesario por mes de trabajo arrojó que para la implementación de este módulo se necesitarían al menos un promedio de **27 hombres por mes**, lo cual haría que el tiempo de desarrollo estimado será de **9 meses,** utilizándose o siendo necesario al menos **3 hombres** para la construcción del módulo.

Luego de estos resultados reajustamos los mismos para adaptarlos a la situación real con la que contamos para la continuación del proyecto.

Como contamos con 4 trabajadores asignados para la construcción del sistema se obtuvo el tiempo que tardaría en terminarse la aplicación para este nuevo valor de la variable que se introdujo, determinándose que el tiempo de desarrollo sería ahora de aproximadamente **7 meses**.

Asumiendo que el salario promedio mensual de los trabajadores asignados en este momento es de **$ 93,75** se incurriría en un gasto mensual de salario de **$ 375,00** por lo que el costo total estimado del software, en función del gasto en salario por hombres vinculados a la construcción del mismo y del tiempo estimado de producción sería de **$ 10 125,00**.

Este costo sólo se refiere al costo del capital humano necesario o que se utiliza para la producción del software; para tener una idea más clara de la relación ganancia costo se deberían tener en cuenta muchos más factores y variables objetivas y subjetivas que intervienen en la construcción de un producto software, así como utilizar otros métodos y técnicas para calcular los costos de una aplicación informática que se enfoquen en esas otras variables.

Tras haber calculado el estimado de tiempo de implementación del módulo de Registro de Actividades Diarias para el Equipo Básico de Salud del Sistema Integral de Salud, así como el esfuerzo nominal (hombre-mes), la cantidad de hombres y el costo en función de la relación salario-tiempo de producción, analizaremos los beneficios que aportará el software cuando se construya el mismo, y por tanto se implante de forma gradual e incremental en todo el Sistema Nacional de Salud.

5.2 BENEFICIOS TANGIBLES E INTANGIBLES

Beneficios Intangibles

- Aumentará la calidad de la información que se registre en las Hojas de Cargo del Equipo Básico de Salud (RAD), permitiendo la actualización confiable del libro de Historia de Salud Familiar, así como la generación de estadísticas e informes de indicadores, incidencias y casos muestrales de la población, acordes con la situación real que se presente en cada momento la misma.

- Los miembros del Equipo Básico de Salud y otros miembros del Sistema Nacional de Salud podrán integrarse a labores investigativas a partir de los datos primarios que se almacenan en el Registro de Actividades Diarias, tarea que no pueden realizar hoy por las dificultades de acceso a la información y de comunicación entre las partes que manejan estos datos.

- Servirá para el control y la administración de las actividades de medicina familiar que ocurren en la estructura básica del SNS posibilitando las labores de auditoría del trabajo del personal de salud asociado a esta tarea.

- Quedarán todos los conocimientos (know how) y experiencias generadas en el desarrollo de este proyecto dentro del marco nacional y accesible para el desarrollo de otros proyectos nacionales con características similares.

Beneficios Tangibles

- Ahorro considerable de divisas por concepto de contratación de servicios a empresas extranjeras (importación de tecnologías) para que desarrollen este trabajo.

- El marco actual de creación de la Alternativa Bolivariana para las Américas (ALBA) abre un universo de posibilidades reales de comercialización del sistema, aunque para ello haya que realizar algunas adaptaciones propias a la realidad de cada cliente probable.

- Los beneficios intangibles traen aparejados un ahorro considerable de recursos que a la postre se convierten es disminución de gastos en los presupuestos asignados en el área donde se utiliza el Registro de Actividades Diarias del Equipo Básico de Salud o su destinación a labores e inversiones que hoy no pueden realizarse.

5.3 ANÁLISIS DE COSTOS Y BENEFICIOS

¿Es factible realizar este módulo que informatice el Registro de Actividades Diarias del Equipo Básico de Salud?

Valorando que el costo estimado asociado a la producción del módulo que se obtuvo, en función del gasto en salario por hombres vinculados a la construcción del mismo, en las actuales condiciones, fue de $ 10 125.00; que existen los medios materiales y la infraestructura necesaria; que está la experticia (know how) suficiente en el área de salud y que se desarrolla aceleradamente en el área de informática, para enfrentar y dar solución al problema; además, se cuenta con los recursos humanos para cada labor durante todo el proceso ; hay un contrato de trabajo y compromiso legal para el estudio y desarrollo del proyecto entre el Ministerio de Salud Pública(MINSAP)

y la empresa SOFTEL; otro punto a valorar es la existencia de la Universidad de las Ciencias Informáticas (UCI) y las características y logística de la misma y por último un punto a tener en cuenta igualmente es que existe un mercado seguro y extendido a todo lo largo y ancho del país.

Por tanto luego de estos argumentos se estima que sí es factible la construcción de este sistema, que los costos generales asociados a la producción no impedirán la rentabilidad y viabilidad del proyecto, ni superarán los beneficios a corto y largo plazo engendrados durante la construcción, la implantación, y partir de la comercialización de la aplicación. Aunque se considera que se debe continuar realizando estudios de factibilidad, por diversos métodos, para corroborar y actualizar en cada momento los datos económicos y contables del proyecto.

5.4 Conclusiones del capítulo...

Con los resultados del estudio de factibilidad del proyecto se puede determinar que el desarrollo de este componente es viable. Todo lo que cuesta hacerlo se transforma en beneficios muy importantes para la empresa, el MINSAP y sobre todo para todo el país.

En este capítulo se realizó el estudio de factibilidad correspondiente al sistema, obteniéndose como resultado un costo total del proyecto de $10 125.00 a desarrollar por 4 personas en un tiempo de 7 meses de trabajo. Se realizó además el análisis entre los costos y los beneficios que reporta la aplicación concluyendo que es factible el desarrollo del software propuesto.

Conclusiones Generales

Con la culminación de este trabajo se han alcanzado los siguientes objetivos:

- Se diseñó el módulo del Registro de Actividades Diarias (RAD) correspondiente al Sistema Informatizado de Atención Primaria (SIAP) para el Sistema Integral de Salud (SIS). En dicho módulo se recogen todas las actividades que comúnmente ocurren en el proceso que se pretende automatizar y se realizaron adaptaciones que guardan relación directa con los resultados de este diseño.

- Se tuvo en cuenta, en el diseño realizado, la existencia de otros sistemas que se están desarrollando igualmente bajo los mismos principios y que deberán integrarse debidamente e indistintamente unos con otros, para lo cual se emplearon las herramientas de modelado de software basadas en UML.

- Las metodologías de desarrollo para lograr la visión integral del sistema, si bien han aportado un marco de trabajo organizado y disciplinado han influido en una demanda mayor de tiempo de elaboración.

- La organización del trabajo estudiantil con alumnos del tercer año de la Facultad 7 de la Universidad de las Ciencias Informáticas alrededor de este proyecto, nos permitió dirigir el proceso de creación de artefactos necesarios para el diseño del módulo y con ello desarrollar técnicas efectivas de trabajo en equipo, necesarias para la Ingeniería de Software. Además ha permitido la experimentación con técnicas de trabajo individual y colectivo, y la participación en un proyecto de producción real con compromisos contables de entrega, evaluación y producción.

- En cuanto al Estudio de Factibilidad, hasta el momento sólo se ha considerado el factor capital humano, pero deberán tenerse en cuenta muchas más variables objetivas y subjetivas que intervienen en la construcción de un producto de software.

Recomendaciones

Luego de la culminación del presente trabajo de investigación y desarrollo se recomienda:

- La implementación de los artefactos de Ingeniería de Software presentados en este documento.

- La debida actualización de este documento con la información obtenida de los procesos productivos, haciendo uso de técnicas de reingeniería.

- Que se valore la necesidad y posibilidad de que la solución que se desarrolle sea construida en sistemas informáticos bajo licencia abierta.

- La valoración del uso de metodologías de desarrollo ágiles para el diseño y producción de otros sistemas con similitud al que se ha modelado actualmente.

- La construcción de un prototipo inicial (versión “Beta”) que sólo garantice la comunicación e intercambio de datos entre la capa de presentación y la capa de datos, obviando las solicitudes de información externa para el completamiento de datos del formulario, con el objetivo de evaluar el diseño de forma general con el cliente y los usuarios del sistema que se ha diseñado.

- Las evaluaciones y asignaturas de los estudiantes asignados, como trabajadores de estos proyectos, debería vincularse estrechamente al problema que estos están solucionando, para no disociar la atención que se requiere y exige para este tipo de proyectos, para ello los profesores de las distintas asignaturas deben también integrarse integrarse a las labores productivas de sus estudiantes.

- Estudiar otros métodos y técnicas para calcular los costos de una aplicación informática que se enfoquen hacia las otras variables que intervengan en la modelación del proceso de desarrollo del software acorde con sus características.

Bibliografía

Referencia Bibliográfica

[1] MINREX. Programa sobre la informatización de la sociedad cubana. Informe de Ministerio de Relaciones Exteriores de Cuba marzo 1ro, 2004. http://www.rebelion.org/cuba.

[2] Castro Ruz, Fidel. Discurso pronunciado en el acto conmemorativo del aniversario 40 del Instituto de Ciencias Básicas y Preclínicas Victoria de Girón, el 17 de octubre de 2002, Granma.

[3] Castro Ruz, Fidel. Discurso pronunciado en la Tercera Graduación del Contingente del Instituto Superior de Ciencias Médicas de la Habana. Teatro “Carlos Marx”. Ciudad de la Habana. 27 de agosto de 1990, Granma.

[4] Castro Ruz, Fidel. Discurso pronunciado en la Clausura del VI Seminario Internacional de Atención Primaria, Ciudad de la Habana. 28 de noviembre de 1997, Granma.

[5] Castro Ruz, Fidel.. Discurso en el acto de inauguración de obras del extraordinario programa de salud. Teatro Astral. 7 de abril del 2003, Granma.

[6] Consideraciones sobre el Proyecto de Informatización de la Atención Primaria de Salud. Revista Habanera de Ciencias Médicas. Volumen 3, No. 10, año 2004

[7] “Registro Informatizado de Salud”. http://www.ris.sld.cu (15/01/05).

[8] Softel.”Descripción por capas de la arquitectura para APS”. Sep, 2003.

[9] OPS. Perfil del Sistema de Servicios de Salud de Cuba. “2da edición. Programa de organización ,gestión de sistemas y servicios de salud.División de desarrollo de sistemas y servicios de salud.2001.

[10] Mirar a Cuba. www.cubaminrex.cu/sociedad. (Granma-Cubaminrex) 1 dic. 2004 (11/06/2005).

[11] Rojas Ochoa, Francisco. Orígenes del movimiento de atención primaria de salud en Cuba. Rev Cubana Med Gen Integr 2003;19(1)

[12] M. Stallman Richards. Software Libre para una sociedad libre. 1ra edición, Traficantes de sueños, Madrid , 2004.

[13] Implantación del software libre en la sociedad y en especial en la administración administración publica. www.libroblanco.com. (15/05/2005).

[14] Lessig, Lawrence. Cultura Libre, cómo los grandes medios usan la tecnología y las leyes para encerrar la cultura y controlar la creatividad. España, 2004.

[15] Matías Sánchez, Enrique. Breve introducción Software Libre. España, octubre 2004.

[16] Mas i Hernández, Jordi. Software libre: técnicamente viable, económicamente sostenible y socialmente justo. 1ra edición. Cargraphics, Barcelona 2005.

[17] Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. www.microsoft.com/spanish/msdn/arquitectura. (22/05/2005).

[18] El Futuro de los Servicios Web XML. www.microsoft.com/spanish/msdn/xml. (22/05/2005).

[19] Pelechado, Vicente. Desarrollo de Aplicaciones WEB basadas en Servicios WEB XML. Un caso práctico/ Marta Ruiz, Joan J. Fons, Pedro Valderas. Valencia, 2004.

[20] Cachero, Cristina. Navigation Análisis and Navigation Design in OO-H and UWE/ Nora Koch. Departamento de Lenguajes y Sistemas Informáticos, Unicersidad de Alicante. España, 2003

[21] Koch, Nora. Modeling Web Bussiness Process whit OO-H and UWE/ Andrea Claus, Cristina Cachero, Santiago Meliá. Ludwing-Maximilians-Universität Müchen. Germany, 2004.

[22] Deboni, José Eduardo. Modelando a Web con UML. Objetos Distribuidos. Sao Paulo, Brasil,1999. Diapositivas

[23] Baumeister H. Towards a UML extensión for Hypermedia Design/ N Koch. Fort Collins. E.U.A. 1999.

[24] Internet, http://es.wikipedia.org/wiki/Internet (20/02/2005)

[25]What is Service-Oriented Architecture?, http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html (10/3/2005)

[26]SOAP: The Simple Object Access Protocol, http://www.microsoft.com/mind/0100/soap/soap.asp (10/3/2005)

[27] Simple Object Access Protocol (SOAP), http://www.desarrolloweb.com/articulos/1557.php?manual=54 (10/3/2005)

[28] Lenguaje de descripción de servicios Web (WSDL), http://www.microsoft.com/spanish/msdn/articulos/archivo/090201/voices/wsdl.asp (12/3/2005)

[29]Apache. http://es.wikipedia.org/wiki/Apache (3/4/2005)

[30].Introducción a la Arquitectura de Software, http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.asp#17 (12/3/2005)

[31].Sistemas distribuidos, www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml(12/3/2005)

[32].Understanding Single Sign-on , http://www.intranetjournal.com/articles/200205/se_05_28_02a.html, (3/4/2005)

[33].Introduction to Perl, www.cclabs.missouri.edu/things/ instruction/perl/perlcourse.html , (3/4/2005)

[34]., http://www.tecnociencia.es/mediawiki/index.php/Active_Server_Pages (19/04/2005)

[35].Introducción a php. http://www.ciberteca.net/webmaster/php (19/04/2005)

[36]JavaServer Pages Technology, http://java.sun.com/products/jsp/ (19/04/2005)

[37]Manual de JavaSript www.redestb.es/soporte/aula/jScript (02/03/2005)

[38]XSLT, http://es.wikipedia.org/wiki/XSLT, (02/03/2005)

Bibliografía

✱Booch, G.: Rumbaugh, J. y Jacobson, I.; El Lenguaje Unificado de Modelado. 2000. Addison-Wesley.

✱Cocomo: Resource Estimation. http://www.cs.unc.edu/~stotts/COMP145/cocomo.html. 22/05/2004.

✱de la Fuente Moya, Antonio. Cocomo v2, Modelo de estimación de costes para proyectos Jacobson, I.; Booch, G. y Rumbaugh, J.. El Proceso Unificado de Desarrollo de software. 2000. Addison-Wesley.

✱Pressman, R. “Software Engineering. A Practitioner’s Approach”. Fourth Edition. McGraw – Hill. USA, 1999.

✱Rumbaugh, J.; Jacobson, I. y Booch, G.; El Lenguaje Unificado de Modelado. Manual de referencia. 2000. Addison-Wesley.

✱Larman, C. “Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design”. Prentice-Hall, Inc. 1998.

✱Larman, Craig.UML y patrones,Introducción al análisis y diseño. Editorial Felix Varela, La Habna,2004