

Universidad de las Ciencias Informáticas
Facultad 7



Título:

**“Plataforma para el desarrollo de
aplicaciones de escritorio en el entorno de
la Atención Primaria de Salud”**

Trabajo de Diploma para optar por el título de
Máster en Informática Aplicada

Autor:

Ing. Yoenny Pérez Romero

Ciudad de La Habana, Junio de 2011

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	9
1.1 Informatización del Sector de la Salud	9
1.2 Ciber-infraestructura para la salud	10
1.3 Informatización del SNS.....	10
1.4 Algunas soluciones de software	12
1.4.1 Sistema informático para la dispensarización en la atención primaria de salud (SIDAPS).....	12
1.4.2 Sistema Informático para la atención para la Atención primaria de Salud (APUS)	12
1.4.3 Sistema Automatizado de Dispensarización de Adultos (SADA)	13
1.4.4 Retrato de Salud de mi Familia (My Family Health Portrait)	13
1.4.5 Paracelsus.....	13
1.5 Tendencias y tecnologías a considerar.	14
1.5.1 Lenguaje de Programación.....	14
1.5.2 Marco de Trabajo	15
1.5.5 Motor de Bases de Datos.....	22
1.5.6 Software libre.....	23
1.5.7 Sistema Operativo Linux.....	23
1.6 Conclusiones del capítulo	24
CAPÍTULO II: SOLUCIÓN PROPUESTA	25
2.1 Solución propuesta.....	25
2.1.1 Modelo del dominio	26
2.1.2 Especificación de los requerimientos	28
2.2 Diseño Propuesto.....	29
2.3 Modelo de datos propuesto.	32
2.4 Implementación.	34
2.4.1 Plantillas CRUD.....	34
2.4.1.1 Problemática.....	35
2.4.1.2 Solución	35
2.4.2 Mejorando presentación de Informes.	37
2.4.2.1 Problemática.....	37
2.4.2.2 Solución	39

2.4.3 Seguridad y confiabilidad. Firmas digitales.	40
2.4.4 Gestión de seguridad. Roles y Usuarios.	42
2.5 Conclusiones del capítulo	42
CAPÍTULO III: ANÁLISIS DE RESULTADOS.....	44
3.1 Concepción del proyecto.....	44
3.2 Módulo Configuración.	44
3.2.1 Requerimientos Funcionales	45
3.2.2 Diseño e Implementación.....	46
3.3 Módulo Medicina Familiar	47
3.3.1 Requerimientos Funcionales	47
3.3.2 Diseño e Implementación.....	47
3.4 Tiempo de desarrollo	48
3.5 Rendimiento	49
3.6 Validación por Comité de Experto	50
3.6.1 Selección de los expertos	51
3.6.2 Número óptimo de expertos.....	52
3.6.3 Aplicación del método	53
3.6.4 Resultados de la evaluación de las encuestas	53
3.7 Conclusiones del capítulo	57
CONCLUSIONES	58
RECOMENDACIONES	59
REFERENCIAS BIBLIOGRÁFICAS	60
ANEXOS	64
GLOSARIO	68

INTRODUCCIÓN

La Organización Mundial de la Salud (OMS) define Sistema Nacional de Salud como un *“complejo de elementos interrelacionados que contribuyen a la salud en los hogares, los lugares de trabajo, los lugares públicos y las comunidades, (...) es el conjunto de unidades administrativas, de producción, investigación y servicios, responsabilizado con la atención integral de la salud de una población”* [1].

El Sistema Nacional de Salud (SNS) cubano, está considerado un sistema complejo, en cuyas relaciones internas y externas descansan los procesos que apoyan la aplicación de planes, programas y acciones. Así se garantizan la consecución de los objetivos encaminados a garantizar la salud de nuestro pueblo. De acuerdo con la complejidad de las acciones preventivas, curativas y de rehabilitación, así como a la mayor especialización de los servicios, los diferentes niveles de atención médica se han organizado en: Atención Primaria en Salud (APS), Atención Secundaria en Salud y Atención Terciaria en Salud.

Las actividades de la Atención Primaria en Salud se realizan en cualquier unidad del Sistema Nacional de Salud y están relacionados fundamentalmente con las que se realizan en el Policlínico, en los Consultorios del Médico de la Familia, los Hospitales Rurales, los Dispensarios y las Postas Médicas. La Atención Secundaria es la brindada por los Hospitales, principalmente en cada una de las regiones y la Atención Terciaria que es la que brindan los Hospitales Especializados.

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 [2]. Este modelo de atención es la mayor fortaleza y potencialidad que tiene el SNS, donde se comienzan a experimentar cambios; y servicios que eran exclusivos de hospitales son abiertos en instituciones y unidades de la atención primaria; surgiendo así el novedoso modelo de policlínico comunitario, que se convierte en la columna vertebral del Sistema de Salud de Cuba [3].

El uso de la Informática en la medicina es una de las aplicaciones más comunes e importantes desde hace varias décadas, y ha permitido al sector de la salud, no sólo contar con métodos novedosos, sencillos y eficaces de gestión administrativa en

Introducción

consultas, hospitales y centros de investigación biomédica, sino también disponer de complejas aplicaciones que reducen la posibilidad de error en el diagnóstico de las enfermedades, y que aceleran su formulación [4].

En el ámbito internacional el desarrollo del software muestra un auge tecnológico, con incremento del conocimiento y de su valor [5] en el sector sanitario. Cada vez más, países como el Reino Unido han logrado digitalizar las historias clínicas de los pacientes [6] y otros como el Sistema de Salud Español están muy cerca, reportando hasta el 96% de los centros de salud que ya disponen de una solución informática para facilitar la gestión clínico asistencial de los médicos de familia y los pediatras [7]. La región latinoamericana no se queda atrás y en países como Argentina, Colombia y Perú desarrollan soluciones de software para el sector de la salud. Es fácil imaginar el impacto sobre el diagnóstico y tratamiento, de un mejor acceso a la información que permita realizar tele consultas, indicaciones de terapia del médico, incluso recetas, sin desplazamiento del paciente. Las TIC tienen un alto potencial para llevar a toda la población servicios de salud con alta calidad aún en los lugares geográficos más apartados [8].

Cuba se encuentra inmersa en un proceso de informatización y el sector de la salud no está ajeno a ello. El Ministerio de Salud Pública (MINSAP), órgano rector del Sistema Nacional de Salud, unido a la dirección de la Revolución, ha trazado los primeros pasos para la reorientación del SNS. Un proyecto con el que se persigue implementar un sistema integral de informatización del SNS que abarque todos los niveles, haciendo énfasis en la APS como nivel conductor de la estrategia sanitaria cubana. Teniendo como eje fundamental el paciente quién será el principal beneficiado al garantizar las aplicaciones, la calidad, oportunidad y consistencia de la información y de este modo para llegar a tener servicios de salud de excelencia.

Existen ya en nuestro país varios sistemas informáticos destinados a la APS como por ejemplo el Sistema de Información para la Salud (SISalud), iniciado en el 2005 por la empresa SOFTEL, el mismo soporta alguno procesos que se encuentran en este nivel, sin embargo posee poco valor de explotación, pues no contempla procesos complejos basados en flujos de trabajo, y está diseñado para ser desplegado de forma centralizada, con un único acceso desde las unidades de salud. Esta cuestión dificulta

Introducción

la comunicación, con muchas unidades que prestan los servicios primarios como son: los consultorios médicos familiares, de tipo uno y dos respectivamente, policlínicos, hogares maternos y de abuelos, entre otros.

A pesar de las estrategias que trazó el MINSAP para la informatización del SNS y la necesidad de lograr la interoperabilidad y comunicación entre las diferentes tecnologías y aplicaciones de software, todavía existen en nuestro país las llamadas “zonas de silencio” estas están ubicadas fundamentalmente en las regiones montañosas más remotas del país. Hoy en día Cuba posee alrededor de unos 19 000 km² de zonas montañosas las cuales están pobladas aproximadamente con 820 mil habitantes, quienes por lo general viven en zonas de difícil acceso. En las cuales existen más de 1 200 consultorios de salud para garantizar la asistencia médica a los pobladores de estas áreas [9].

A esto se le suman factores tecnológicos, o sea; presentan una infraestructura limitada, con ancho de banda reducida y de conexión restringida, una vez en el día o no existe, además las estaciones de trabajo poseen las prestaciones mínimas para trabajar (256 MB de RAM, en algunos casos 128), limitando las soluciones locales que puedan existir. Como también existen establecimientos restringidos por el área geográfica donde están ubicados y/o por la necesidad de una cobertura mínima de recursos.

Situación problemática

La Universidad de las Ciencias Informáticas es una casa de altos estudios que además de formar profesionales altamente capacitados en el área de la informática, tiene como misión potenciar el desarrollo de software en el país. Como parte de la estrategia para dar cumplimiento a sus objetivos se crearon varios centros especializados para el desarrollo de software en diferentes líneas de trabajo. El Centro de Informática Médica (CESIM), tiene como función principal ejercer el desarrollo de soluciones de software para el sector de la salud, por lo que lleva a cabo la informatización de los diferentes niveles del SNS distribuidos en los distintos departamentos que existen en su estructura.

El Departamento de Sistemas de Atención Primaria de Salud desarrolla un sistema de información para el nivel de atención primaria (SIAPS), con el objetivo de integrar los procesos de negocio de APS en una plataforma única para la gestión, procesamiento y

Introducción

transmisión de la información médica. Tarea de primordial importancia para llegar a tener servicios de salud de excelencia y de este modo consolidar la infraestructura global de información asociada al paciente. SIAPS es un sistema de información, que al igual que la mayoría de los sistemas desarrollados en el CESIM, está implementado sobre tecnología web.

Como es bien conocido las aplicaciones web están de moda sobre todo por las ventajas que representan para algunos entornos, no dependen de ningún sistema operativo ni configuración de hardware específica. Para su ejecución simplemente basta con teclear su dirección URL en cualquier navegador web. De igual manera sus actualizaciones se hacen de manera muy sencilla, sin necesidad de hacer descargas, instalaciones o comprar físicamente el producto. Pero como todo sistema de este tipo presenta algunas desventajas, como:

- **Acceso limitado**, la necesidad de conexión permanente y rápida a Internet o intranet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- **La interactividad no se produce en tiempo real**, en las aplicaciones web cada acción del usuario conlleva un tiempo de espera muchas veces excesivo hasta que se obtiene la reacción del sistema.
- **HTTP, no mantiene estado**, esto quiere decir que, entre una página y otra, no hay “memoria” de las acciones efectuadas anteriormente por el usuario.

Debido a las carencias presentes en las Aplicaciones Web; en entornos como los mencionados con anterioridad, donde no hay o casi no hay conexión y/o las limitaciones tecnológicas impiden o limitan un despliegue de un servidor web o de aplicaciones es que se hace necesario instrumentar una solución que permita facilitar los servicios de salud que se brindan a la población en estos parajes y que sea una alternativa a los desarrollos web del CESIM.

Por aplicaciones de escritorio se entiende toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea Windows, Linux ó Mac. Las aplicaciones de escritorio mantienen un contacto permanente entre los procesos internos del programa y lo que sucede en la interfaz de usuario. Es por esto que no requieren del paradigma de páginas de la web y tienden a ofrecer una experiencia de

usuario más fluida entre una acción y otra. Entre las ventajas de las aplicaciones de escritorio, se podrían destacar:

- **Mayor capacidad gráfica visual**, permiten una mayor gama de efectos y mejor interactividad.
- **Menor tiempo de respuesta**, el tiempo entre una acción y otra es más rápido porque tiene acceso local a los recursos del ordenador.
- **Mayor personalización.**

A menudo sucede que un desarrollador o un grupo de desarrollo comienzan a crear una aplicación de escritorio para un determinado cliente. Con el paso del tiempo, y comúnmente con el acercamiento de las fechas de entrega de versiones finales, demasiado a menudo sucede que la idea inicial de hacer una aplicación ligera, ágil, sencilla, intuitiva y usable, empieza a convertirse en lo que se ha dado por llamar con el nombre de clientes pesados. Finalmente, también a menudo muchos proyectos serán abandonados, en opinión de los desarrolladores, a causa de los pesados de los clientes (las aplicaciones, no los usuarios finales). Entre algunas de las problemáticas más comunes a las que se enfrentan los desarrolladores de las aplicaciones de escritorio se pueden mencionar:

- **La implementación de componentes de interfaz de usuario**, los lenguajes e IDEs de desarrollo no siempre tienen lo que hace falta para el desarrollo de la aplicación.
- **Implementación de los CRUD**, en todas las aplicaciones de gestión hay que implementar CRUD, funcionalidades que conceptualmente son iguales y que en la práctica se diferencian muy poco en su implementación, pero aún así se hace complejo la reutilización de código.
- **Presentación de informes o reportes**, un tema recurrente en cualquier tipo de aplicación que trabaje con datos.
- **Recurrente acceso a datos**, los mismos datos deben ser pedidos una y otra vez.

En otro orden, cuando se desarrollan aplicaciones para el dominio de la salud, un tema importante es la fiabilidad de los documentos clínicos, por lo que se hace necesario

Introducción

buscar técnicas y métodos que garanticen que la validez de los escritos en un Historia Clínica o el medicamento emitido en una receta, por solo poner algunos ejemplos. Para ello generalmente se utilizan técnicas de firma electrónica y/o digital.

Teniendo en cuenta la situación anterior se plantea como **problema**:

¿Cómo facilitar la implementación de una variante desktop a las aplicaciones Web desarrolladas en el CESIM para un entorno de salud?

El **objeto de estudio** apunta al proceso de desarrollo de sistemas informáticos para el entorno de la salud.

El **campo de acción** se centra en el proceso de desarrollo de sistemas informáticos en el CESIM, específicamente en el área de APS.

Como **objetivo de la investigación** se define la construcción de una plataforma para el desarrollo de aplicaciones de escritorio, que facilite la implementación de una aplicación desktop para gestionar la información médica asociada al paciente en entornos donde la conectividad es casi nula y/o la infraestructura tecnológica limitada.

Para lograr ese objetivo general se definen las siguientes **tareas de la investigación**:

- Realizar el estudio del estado del arte de las principales tendencias tecnológicas relacionadas con el desarrollo de software para el área de la salud.
- Desarrollar una plataforma para aplicaciones de escritorio que facilite la implementación de un sistema par gestión de los datos asistenciales del paciente.
- Poner en práctica la solución propuesta con la implementación de los principales requisitos funcionales del módulo de Historia de Salud Familiar y el de configuración.
- Evaluar la factibilidad de la puesta en práctica de la solución propuesta.

Teniendo en cuenta lo anteriormente planteado se formuló la siguiente **hipótesis**:

La construcción de una plataforma para el desarrollo de aplicaciones de escritorio, permitirá el desarrollo de aplicaciones de software que faciliten la gestión de la

Introducción

información médica asociada al paciente en entornos donde la conectividad es casi nula y la infraestructura tecnológica limitada.

Variables:

Variable independiente: Desarrollo de una plataforma para las aplicaciones desktop.

Variable dependiente: Contar con una variante de desarrollo de software para entornos donde la conectividad es casi nula y la infraestructura tecnológica limitada.

El **aporte teórico práctico** de este trabajo está dado por el hecho de que el desarrollo de un núcleo para las aplicaciones ligeras, posibilitará que las aplicaciones desarrolladas para el sector de la salud en el CESIM a las cuales no se tiene acceso desde locaciones donde la conectividad es casi nula y la infraestructura tecnológica limitada, puedan contar con una variante de desarrollo que permita la informatización de los servicios de salud en estos lugares y que posteriormente se puedan integrar como un solo paquete de software. Propiciando así una mejor calidad y completitud de los productos de software desarrollados en el CESIM. Además de que propicia una mayor productividad en los desarrolladores.

La **novedad científica** de este trabajo está en que la solución planteada consiste en la integración de un grupo de tecnologías que por separado solo resuelven parte del problema y que en un único paquete de software dan solución a la problemática expuesta. El desarrollo del núcleo para las aplicaciones ligeras permitirá que partiendo de este como base, se puedan desarrollar aplicaciones de escritorio para cualquier sistema desarrollado en el CESIM, evitándole a los desarrolladores de estos sistemas preocupaciones por temas como la forma de presentar información de informes, la gestión de los datos, entre otros. Por último es importante señalar que el núcleo para las aplicaciones ligeras, no solo está limitado a aplicaciones de salud sino que es una solución genérica que permite ser reutilizado en aplicaciones con otros tipos de procesos de negocio.

Entre los **métodos de trabajo científico** utilizados durante la investigación se destacan los siguientes:

Analítico-Sintético: este método permite estudiar los sistemas de salud y buscar información para encontrar una posible solución a la situación problemática planteada,

así como identificar herramientas y métodos a utilizar relacionados con el objeto de estudio.

Inductivo-Deductivo: este método permite llegar a conclusiones lógicas a partir de los conocimientos adquiridos en la investigación, y así poder plantear la estrategia de desarrollo para lograr una correcta interoperabilidad entre aplicaciones de salud.

Análisis Histórico-Lógico: este método permite estudiar cómo ha evolucionado el intercambio de información entre aplicaciones de salud y la interpretación de los vocabularios clínicos en los diferentes sistemas de salud.

El presente documento se estructura como se describe a continuación; tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y Anexos. En el Capítulo I “Fundamentación Teórica”, se documentan los fundamentos teóricos que constituyen las bases de la investigación. Se realiza un análisis crítico del estado del arte de las principales tendencias tecnológicas relacionadas con el desarrollo de software para el área de la salud. Se hace una pequeña reseña sobre la evolución del desarrollo de software en el país, especialmente en el área de la medicina, su impacto en la sociedad cubana y su desarrollo actual en la UCI. En el Capítulo II “Solución Propuesta”, se documenta la solución desde el punto de vista conceptual, así como las herramientas y tecnologías utilizadas en el desarrollo del trabajo. En el Capítulo III “Análisis de Resultados” se describe la aplicación de la solución propuesta y ejemplos de aplicaciones prácticas de la misma.

Capítulo I. Fundamentación Teórica

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se brinda la base teórica y conceptual del sistema a desarrollar. Como aspectos esenciales se abordan los antecedentes existentes de sistemas o subsistemas similares que dieron al traste con el problema en cuestión. Así como, un análisis de las tecnologías a utilizar. Se hace una pequeña reseña sobre la evolución del desarrollo de software en el país, especialmente en el área de la medicina y su impacto en la sociedad.

1.1 Informatización del Sector de la Salud

En Cuba, el MINSAP en el marco de la Batalla de Ideas, define la Informatización del Sector de la Salud como “el PROCESO, cuyos procedimientos se enmarcan en el contexto de la Informatización Social, con las TIC 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; administración de sus recursos, eficacia y eficiencia en su Gerencia”. Así pretende crear una infraestructura informática para el sector, que integre todos los productos o servicios y permita que todas las unidades de salud del país alcancen un nivel de informatización elevado en las actividades que realicen, influyendo directamente en la calidad de los servicios que se brinden [10].

La informatización del Sistema Nacional de Salud Cubano comenzó por la Atención Primaria, su objetivo fundamental consiste en la creación de un 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.

El programa de informatización de la APS tiene claramente definidos tres propósitos fundamentales: [11]

- Fortalecer la conectividad de las instituciones de salud territorial que permita una interrelación más efectiva entre ellas, así como un flujo informativo eficiente y eficaz para la toma de decisiones a nivel local y niveles superiores.

Capítulo I. Fundamentación Teórica

- Fortalecer el Subsistema del Médico y la Enfermera de la Familia, para lo cual hay que centrar los esfuerzos en los policlínicos, consultorios y comunidad.
- Fortalecer el resto de los Subsistemas de salud territorial, interrelacionados con el primero que le brindan la interacción necesaria

1.2 Ciber-infraestructura para la salud

La visión de una Ciber-infraestructura para la Salud pretende articular en un sistema los esfuerzos e iniciativas de todos los recursos y actores que trabajan en el campo de las TIC para ofrecer una plataforma estable y robusta que revolucione la gestión del proyecto de salud en el país. Incluye hardware, software, métodos, servicios, equipos médicos, sensores, soporte técnico, educación, instituciones y personas dedicadas a su desarrollo y funcionamiento (fig. 1). Ofrecer un sistema integrado de recursos computacionales para una atención integral, apoyados en las TIC y la gestión del conocimiento es el beneficio que ofrece la Ciberinfraestructura para la Salud, punto de partida para replantearse el proceso de informatización de la salud en Cuba [12].



Fig. 1 Ciber-infraestructura para la Salud

1.3 Informatización del SNS

Los productos y servicios que se integrarán a la ciber-infraestructura del sector se realizarán en lo fundamental sobre sistemas abiertos, utilizando software libre y de

Capítulo I. Fundamentación Teórica

calidad. Algunas de las políticas para la informatización del SNS son las siguientes [13]:

1. El proceso de informatización responde a las Políticas y Principios Socialistas. Es una estrategia vital y prioritaria del Sistema Nacional de Salud, que debe alinearse con las tecnologías de punta y los estándares de calidad desarrollados en el mundo, adecuados a nuestras condiciones particulares.
2. Todas las inversiones y proyectos que se desarrollen para el SNS deben considerar el elemento informático desde su concepción inicial y responder a estrategias, planes de desarrollo y políticas de estandarización.
3. La seguridad informática y de contingencia son requisitos imprescindibles, responsabilidad ineludible de los productores, prestadores y usuarios, para confidencialidad, seguridad de los datos y autenticación.
4. La utilización y generalización deben estar avaladas de una investigación científica, así como la informatización, recursos y servicios estarán dirigidos al desarrollo y adquisición de conocimientos.

El centro de la gestión de la información está en el policlínico donde se gestiona la información que se genera en los Equipos Básicos de Trabajo (EBS) de los Consultorios del Médico de la Familia (CMF) por Grupos Básicos de Trabajo (GBT) y la generada en el propio policlínico, que será la base de la Historia Clínica para otros niveles de atención (secundario y especializado), según se muestra en el Figura 2. Por lo que es de vital importancia garantizar la gestión de la información en el nivel más bajo, los CMF.



Fig. 2. Policlínico como Centro de gestión de información

Capítulo I. Fundamentación Teórica

1.4 Algunas soluciones de software

Una aplicación de escritorio se refiere a una aplicación funcionando con un ambiente de escritorio como Windows o Macintosh. *“Cuando la interfaz gráfica de usuario fue desarrollada por Apple en los 80’s, esto permitió hacer cosas en una forma más fácil en un computador de escritorio. Los usuarios ahora podían realizar 95 % de sus tareas sin tener que recordar crípticos comandos”* [14]. A partir de aquí las aplicaciones se han continuado desarrollando, llegando a tener múltiples aplicaciones para diversos tipos de trabajos, aplicaciones de trabajo de oficina, asistentes especializados en diferentes áreas de la Ingeniería, entre otros.

“Los sistemas de información computarizados electromecánicamente en la APS se desarrollaron desde 1930-40's, y electrónicamente desde 1974. La informática e informatización de la APS aparecen a partir de 1985, como apoyo al médico en el diagnóstico y la terapéutica clínica, y al programa de salud en la comunidad” [15].

Los desarrollos de aplicaciones para entornos desconectados, especialmente para el dominio de la salud, fijan sus esfuerzos en soluciones de escritorio, que resuelvan los problemas asociados a un área determinada de la atención sanitaria. Por eso a continuación se ponen a consideración algunas soluciones existentes

1.4.1 Sistema informático para la dispensarización en la atención primaria de salud (SIDAPS)

SIDAPS permite a sus usuarios la gestión de la información de los individuos y familias, así como el funcionamiento familiar, factores socioeconómicos, entre otros aspectos, además de la obtención de reportes estadísticos que son de vital importancia para la evaluación y control de la salud de la población por el EBS [16].

Es una aplicación de escritorio, para la realización del mismo se utilizó como herramienta de programación Borland Delphi versión 7. Los tipos de tablas de la base de datos son DBF y usa el gestor de base de datos DataBase Desktop, el cual incorpora Delphi. Está diseñado para el sistema operativo Windows en cualquiera de sus versiones, pero no para Linux.

1.4.2 Sistema Informático para la atención para la Atención primaria de Salud (APUS)

Capítulo I. Fundamentación Teórica

APUS se creó con el objetivo de que fuera un sistema para la gestión médica en la APS, su principal objetivo era automatizar la información en un centro de Salud en la Atención Primaria. Además el mismo brinda información para la elaboración y actualización periódica del Análisis de la Situación de salud, que permita evaluar la eficiencia de los principales servicios y actividades médicas realizadas [17].

El producto fue elaborado en la plataforma Cliente/Servidor, utilizando un lenguaje visual (Borland Delphi), y SQL como gestor de base de datos, se programó también en ambiente Web, utilizando ASP (Active Server Page).

1.4.3 Sistema Automatizado de Dispensarización de Adultos (SADA)

SADA fue uno de los primeros sistemas de dispensarización automatizados en Cuba. El mismo se realizó en 1987 para programar turnos y seguimiento en los policlínicos y el terreno acerca de las enfermedades crónicas y factores de riesgos en los adultos [15]. Se desconoce bajo qué plataforma y tecnología fue desarrollado.

1.4.4 Retrato de Salud de mi Familia (My Family Health Portrait)

Este retrato es un componente de una iniciativa Nacional del Cirujano General de los Estados Unidos para mejorar la salud. Esta iniciativa anima a las personas a buscar el historial familiar de salud. Proporciona una tabla que los individuos pueden llenar con información relevante de la salud de sus familiares. La primera versión del Retrato de Salud de mi Familia se dio a conocer en noviembre del 2004, como un software descargable para computadoras con sistema operativo Windows.

Este reúne la información del individuo y su familia con el objetivo de crear un Familiograma. Es privado, no guarda su información. Le ofrece un historial de salud familiar que puede compartir con los miembros de su familia o enviarla a la atención de su médico. Se desconoce bajo qué plataforma y tecnología fue desarrollado [18].

1.4.5 Paracelsus

Paracelsus es una herramienta informática diseñada para el registro electrónico de la información de salud, es consecuente en su diseño lleva un orden semiológico, clínicamente técnico en relación con las necesidades médicas y epidemiológicas actuales y con los últimos progresos científicos en diagnóstico y terapéutica.

Capítulo I. Fundamentación Teórica

Paracelsus. reduce el tiempo que se toma para el registro de la información de la salud al mínimo y agiliza la dinámica de la entrevista para que el médico preste mayor atención al proceso diagnóstico, se acoge a los Estándares de Seguridad para la Protección de la Información de Salud Protegida Electrónicamente incluidos en los requerimientos de la Health Insurance Portability and Accountability Act de 1996 de Estados Unidos en cuanto a control de acceso, controles de auditoría, autenticación de datos, almacenamiento de datos, respaldo de datos, pruebas y revisiones. Se desconoce bajo qué plataforma y tecnología fue desarrollado [19].

La mayoría de los sistemas analizados anteriormente son aplicaciones de software que como bien se pudo constatar, fueron diseñados para resolver algún que otro problema particular o sencillamente facilitar el trabajo en el área de la salud. Son aplicaciones en su generalidad monolíticas desarrolladas bajo diferentes tecnologías, todas ellas dependientes del sistema operativo Windows. Algunas de estas desarrolladas con software privativo y otras no ofrecen información a cerca de la tecnología con las que se implementaron.

El acceso al código fuente permitiría hacer un análisis más profundo y en cualquier caso poder reutilizar este, pero no es accesible en ninguno de los casos, porque no están desarrolladas bajo estándares de software libre y/u Open Source o porque no están publicados para su descarga. No permiten en ningún caso la modificación de sus funciones para adaptarlo a la problemática planteada y no cumplen con la política de la soberanía tecnológica que se lleva a cabo en Cuba. En consecuencia se determina llevar a cabo un estudio de las tendencias y tecnologías actuales para desarrollar una solución al problema planteado.

1.5 Tendencias y tecnologías a considerar.

El desarrollo de software es un área de la ingeniería donde constantemente se están produciendo cambios en las técnicas y tecnologías, por lo que su vertiginosa evolución nos lleva a una constante búsqueda de las mejores y más adecuadas opciones para darle solución al problema que se quiera resolver.

1.5.1 Lenguaje de Programación

Capítulo I. Fundamentación Teórica

Un lenguaje de programación es cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora. Es complicado definir qué es y qué no es un lenguaje de programación. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Normalmente, es la computadora la que realiza la traducción [20].

Java es un lenguaje de programación sencillo, orientado a objetos, de propósito general e independiente de la plataforma de desarrollo, cuya sintaxis es muy parecida al lenguaje C o C++, fue creado por James Gosling, cumpliendo así con el slogan de la compañía "compilar una vez y ejecutar en cualquier parte".

Entre sus principales características se tiene que es un lenguaje independiente de la plataforma lo que posibilita que un programa en Java pueda funcionar en cualquier ordenador [21]. Se utiliza en un amplio abanico de posibilidades y casi cualquier implementación o desarrollo que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas.

La plataforma Java consta de las siguientes partes:

- El lenguaje de programación, mismo.
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.
- El API Java, una biblioteca estándar para el lenguaje.

Permite a los desarrolladores que la utilizan, entre otras:

- Desarrollar software en una plataforma y ejecutarlo en prácticamente cualquier otra plataforma.
- Crear programas para que funcionen en un navegador web y en servicios web.
- Combinar aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados.

1.5.2 Marco de Trabajo

Un marco de trabajo ("framework" en inglés) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática

Capítulo I. Fundamentación Teórica

particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. Provee una estructura y una metodología de trabajo así como una arquitectura de software definida.

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje, interpretado o compilado, entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Es un esqueleto, una armazón, sobre la que será construida la aplicación. Un framework tiene ciertas responsabilidades en el funcionamiento de una aplicación. Estas responsabilidades son:

- Gestión de Menú.
- Gestión de Formularios.
- Gestión de Seguridad.
- Gestión de Comunicación.
- Gestión de Acceso a Datos.

En general los framework son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

1.5.2.1 JSR 296 (*Swing Application Framework*)

Swing Application Framework (SAF, por sus siglas en inglés), que no es más que un conjunto de clases que ayudan a construir aplicaciones de escritorio, manejando ciertas tareas que pueden resultar complicadas para los desarrolladores. Puede dividirse en tres partes fundamentales:

- Swing application framework (SAF- jsr296)
- Beans Binding (jsr 295)
- Java persistence API (JPA)

Capítulo I. Fundamentación Teórica

SAF proporciona todas las funcionalidades básicas para el desarrollo de una aplicación con corte profesional. Este framework presenta las siguientes funcionalidades:

- Control del ciclo de vida de la aplicación.
- Manejo de recursos.
- Acciones (*Actions*).
- Tareas o trabajo en segundo plano (*Tasks*).
- Manejo de estado de sesión (*Session state*).

Beans Binding (jsr 295), se caracteriza por lo siguiente:

- Enlaza dos propiedades de objetos de forma sincrónica.
- Las propiedades fuentes se especifican usando Expression Language (EL)
- No se necesita de objetos especiales.

1.5.2.1.1 Java Persistence API (JPA)

Es la Interfaz de Programación de Aplicaciones (API) estándar para la persistencia y el mapeo objeto/relacional para la plataforma JEE 5, permite utilizar un modelo de dominio Java para administrar bases de datos relacionales [22]. Permite además unificar la manera en que funcionan las utilidades que provee un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.

1.5.2.2 Enterprise JavaBeans (EJB3)

Permite realizar la administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portables y persistencia de datos. Incorpora el estándar JPA como el principal API de persistencia para aplicaciones EJB3. Su objetivo es simplificar el desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java. Forma parte de la especificación JEE 5.

1.5.2.3 Hibernate

Capítulo I. Fundamentación Teórica

Hibernate es una herramienta que realiza el mapeo entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM (object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa. Hibernate no solo realiza esta transformación sino que proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que reducen el tiempo de desarrollo. A través de la implementación del estándar JPA que provee Hibernate 3.3, se puede realizar el acceso a datos.

Hibernate es la implementación de una capa de persistencia. Esto tiene como ventajas la independencia del sistema gestor de base de datos a utilizar. El lenguaje para realizar Querys es HQL (propio de Hibernate). Esto permite que no haya que cambiar nada de código para cambiar de Sistema Gestor de Base de Datos. Solo cambiar un parámetro en el fichero de configuración de Hibernate. Además Hibernate implementa una especie de caché, guardando datos en su sesión. Esto permite liberar la carga a la base de datos y permitir así multitud de consultas simultáneas sin que colapse la misma.

1.5.2.4 JasperReport

Es una librería de clases Java de código abierto desarrollada para facilitar adicionar capacidades de reporte a las aplicaciones Java. Permite realizar reportes de código abierto que tiene como función llevar documentos ricos en contenido a la pantalla, la impresora, o a archivos PDF, HTML, XLS, CSV y XML.

1.5.3 Arquitectura de software.

Según el estándar ANSI/IEEE Std 1471-2000 una arquitectura es “la organización fundamental de un sistema, compuesta por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán y los principios que orientan su diseño y evolución”. Consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software. Pauta los fundamentos para que analistas, diseñadores, programadores, etc., trabajen

Capítulo I. Fundamentación Teórica

en una línea común que permita alcanzar los objetivos trazados para el desarrollo de un sistema de software.

1.5.3.1 Arquitectura en Tres Capas

El uso de una arquitectura en capas, representa uno de los estilos que aparecen con mayor frecuencia. De forma general, se define el estilo encapas como una organización jerárquica, que separa la lógica del negocio de la lógica de diseño, de forma tal que cada capa, suministra servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. [23]

Las 3 capas fundamentales son:

- **Capa de Presentación:** es con la que interactúa el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario.
- **Capa de Negocio:** es donde se encuentran los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se nombra así porque es aquí donde se implantan las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.
- **Capa de Datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

La ventaja principal de este estilo es que se puede desarrollar en varios niveles y, en caso de que ocurra algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles

1.5.3.2 Arquitectura Basada en Componentes

Capítulo I. Fundamentación Teórica

La arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de orientación por objetos y no se enfoca en asuntos específicos de los objetos como los protocolos de comunicación y la forma como se comparte el estado.

Los siguientes son algunos de los principales beneficios del estilo de arquitectura basado en componentes:

- **Facilidad de instalación:** Cuando una nueva versión esté disponible, se podrá reemplazar la versión existente sin impacto en otros componentes o el sistema como un todo.
- **Costos reducidos:** El uso de componentes de terceros permite distribuir el costo del desarrollo y del mantenimiento.
- **Facilidad de desarrollo:** Los componentes implementan una interfaz bien definida para proveer la funcionalidad definida permitiendo el desarrollo sin impactar otras partes del sistema.
- **Reusable:** El uso de componentes reutilizables significa que ellos pueden ser usados para distribuir el desarrollo y el mantenimiento entre múltiples aplicaciones y sistemas.
- **Mitigación de complejidad técnica:** Los componentes mitigan la complejidad por medio del uso de contenedores de componentes y sus servicios.

1.5.4 Patrones arquitectónicos y de diseño

Un patrón es una unidad de información nombrada, instructiva e intuitiva que captura la esencia de una familia exitosa de soluciones probadas a un problema recurrente dentro de un cierto contexto. El objetivo de los patrones es crear un lenguaje común a una comunidad de desarrolladores para comunicar experiencia sobre los problemas y sus soluciones [24].

1.5.4.1 Patrones Arquitectónicos

Capítulo I. Fundamentación Teórica

Son los que definen la estructura de un sistema software, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema.

Modelo Vista Controlador (MVC)

El patrón Modelo Vista Controlador (MVC) separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

- **Modelo:** El modelo representa los datos y las reglas que gobiernan el acceso y actualización de los mismos.
- **Vista:** La vista muestra el contenido de un modelo. Especifica exactamente cómo los datos del modelo deben ser presentados. Si los datos del modelo cambian, la vista debe actualizar su representación según corresponda.
- **Controlador:** El controlador traduce las interacciones del usuario con la vista en acciones que el modelo puede realizar.

La lógica de una interfaz de usuario cambia con más frecuencia que los datos y la lógica de negocio. Si se realiza un diseño que mezcle los componentes de interfaz y de negocio, como consecuencia cuando se necesite cambiar la interfaz, se tendrán que modificar los componentes de negocio. De ahí que el uso de este patrón facilite la actualización de los diferentes componentes de software de una manera separada y en menor tiempo.

1.5.4.2 Patrones de diseño

Son el esqueleto de las soluciones a problemas específicos y comunes del diseño en el desarrollo de software. Puede considerarse como un documento que define una estructura de clases que aborda una situación particular.

Singleton

Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia, esto permite que si dos usuarios intentan crear la instancia al mismo tiempo y esta no existe todavía, sólo uno de ellos debe lograr crear el objeto, disminuyendo la sobrecarga de memoria por la creación de objetos. Además

Capítulo I. Fundamentación Teórica

proporciona un punto de acceso global a la instancia creada, lo que permite que no se tenga que repetir innecesariamente la instanciación de la clase en diferentes contextos de la aplicación.

Lazy

El Lazy Loading es un patrón de diseño útil y simple que se basa en ir cargando los distintos componentes de una clase a medida que se van usando. Es comúnmente utilizado para aplazar la inicialización de un objeto hasta el punto en que sea necesario. Puede contribuir a la eficiencia en la operación del programa, si se utilizan adecuadamente.

Abstract Factory

El patrón Abstract Factory ofrece una interfaz para la creación de familias de objetos relacionados o dependientes sin especificar las clases concretas a las que pertenecen. El problema que intenta solucionar este patrón es el de crear diferentes familias de objetos. Sus ventajas están dadas por permitir aislar las clases de implementación: ayuda a controlar los objetos que se creen y encapsula la responsabilidad de creación, además hace fácil el intercambio de familias de objetos sin mezclarse y fomenta la consistencia entre objetos [25].

1.5.5 Motor de Bases de Datos

HSQL es un motor de bases de datos SQL ligero (611 Kb aprox.), OpenSource e implementado completamente en Java. Ideal para realizar pruebas sin tener que conectarse a un gestor de bases de datos. HSQL permite la creación de índices, control de la integridad referencial, sentencias de definición de datos, etc. Ahorra bastante tiempo de desarrollo y al trabajar sobre bases de datos es flexible a la hora de realizar cambios en la información a tratar o cuando los requisitos de la información también cambien.

HSQLDB apoya el dialecto de SQL definido por las normas de SQL 92, 99 y 2003. Es una base de datos que según sus creadores puede llegar a manejar varios GB de DATA (8GB para tablas de disco), trabaja con el estándar SQL92 y soporta la creación de procedimientos almacenados (escritos en Java). Cuenta con un Driver para el trabajo con Java, que es capaz de ejecutarse dentro del espacio de memoria de la propia aplicación. Las tablas y datos se cargan desde un fichero de script SQL. Esto

Capítulo I. Fundamentación Teórica

permite una velocidad muy alta y la posibilidad de que sólo se permitan conexiones desde dentro de la propia máquina virtual de la aplicación, lo que aporta un extra de seguridad.

1.5.6 Software libre

Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software: [26]

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puede ayudar a otros (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3). De igual forma que la libertad 1 el acceso al código fuente es un requisito previo.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, deberías tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa, entre otras cosas que no tienes que pedir o pagar permisos. [27]

1.5.7 Sistema Operativo Linux

Linux, sistema operativo derivado de UNIX que manteniendo la generalidad de sus características, como el ser multitarea y basado en bibliotecas dinámicas, puede ser ejecutado en computadoras u ordenadores personales aunque su potencia sea limitada [28].

En sus orígenes fue desarrollado, en 1990, por el informático finlandés Linus Torvalds, que publicó su código como un denominado código abierto, esto es, accesible para toda la comunidad, sin restricciones para modificarlo y ampliarlo. Este planteamiento, favorecido por su estructura modular (basado en la instalación de diversos paquetes),

Capítulo I. Fundamentación Teórica

generó una nueva visión de desarrollo informático, ya que su expansión fue debida a la aportación, generalmente voluntaria y sin ánimo de lucro, de multitud de desarrolladores independientes.

En la actualidad este sistema operativo ha obtenido un cierto apoyo por parte de la industria, de forma que empresas como IBM o Hewlett-Packard lo integran en algunas de sus computadoras y prestan el soporte técnico correspondiente, normalmente, como parte de los sistemas servidores. Su implantación en sistemas para usuarios finales, aún no ha alcanzado la extensión que tiene en algunos de los ámbitos más profesionales, muy especialmente en servidores de Internet.

1.6 Conclusiones del capítulo

En este capítulo se valoraron los principales elementos relacionados con el objeto de estudio y campo de acción de la investigación, lo que permitió una mejor comprensión de los mismos. Se pudo constatar la necesidad de desarrollar una plataforma para el desarrollo de aplicaciones de escritorio que responda a las necesidades del CESIM, ya que ninguna de las analizadas constituye una variante de escritorio viable a las soluciones Web del Centro. Se escogieron las tecnologías adecuadas para el desarrollo de la plataforma evaluándose cada una de ellos por sus características y ventajas que aportan facilidad para el desarrollo de la misma.

Capítulo II. Solución Propuesta

CAPÍTULO II: SOLUCIÓN PROPUESTA

En este capítulo se expone la propuesta de solución. Se realiza una descripción del problema a resolver y se muestra el modelo de dominio junto con la descripción de cada uno de los componentes que lo forman. A continuación se encuentran una serie de epígrafes que describen cada uno de los componentes que conforman la plataforma. Algunas de las descripciones están acompañadas de ejemplos que ilustran su funcionamiento y uso. Además, se muestran los diagramas de clases del diseño de la solución propuesta.

2.1 Solución propuesta

La medicina como ciencia, la prestación de servicios de salud, la docencia, la formación de recursos humanos y la investigación están en un continuo cambio. Un factor clave de este cambio se sustenta en las nuevas tecnologías de la información y las comunicaciones (TIC) a través del uso de las herramientas de comunicación y de la informática médica, potenciando la mayor inmediatez posible, la seguridad, la calidad y el control de las acciones de Sistema Nacional de Salud [29].

El Centro de Informática Médica (CESIM) el cual se deriva de la facultad 7 de la UCI ha organizado el proceso productivo de forma uniforme y coherente. Para ello ha definido una estructura por departamentos que se especializan en determinadas líneas de trabajo con un objetivo común, Informatizar el Sistema Nacional de Salud y crear soluciones de software competitivas y comercializables.

Para lograr una mejor organización de los procesos productivos del CESIM, se creó un grupo de arquitectura con el objetivo de definir las políticas, tecnologías, procedimientos, herramientas, etc., que serán utilizadas durante el proceso de desarrollo de sistemas informáticos. En 2008 se elaboró un primer el documento de arquitectura que guiaría los desarrollos del centro.

Las definiciones pautadas en el documento de arquitectura están enfocadas principalmente a los desarrollos web bajo plataformas y tecnologías Java. De lo anterior se puede derivar que los proyectos productivos desarrollados en el CESIM son Web. Pero, ¿qué ocurre cuando por limitaciones tecnológicas o por limitaciones propias tecnología un desarrollo es este tipo no es factible? Es por ello que se hace

Capítulo II. Solución Propuesta

necesario definir nuevas pautas para aplicaciones de escritorio, y resolver una serie de problemas que vienen aparejados al desarrollo de aplicaciones de software de esta índole.

El solo hecho de pautar políticas, tecnologías, procedimientos, herramientas, etc. no garantiza la creación de una aplicación de escritorio sea ligera, ágil, sencilla, intuitiva y usable. Es necesario ir más cerca del problema, y lograr una mayor reutilización de componentes de software y resolver problemáticas comunes que se repiten una y otra vez, provocando que los desarrolladores tengan dedicarse a solucionarlas cuando podrían emplear ese tiempo en implementar las funcionalidades de los sistemas y lograr y así lograr mejores resultados y un mayor rendimiento.

En la actualidad está muy extendido el uso de marcos de trabajo (*framework*, en inglés). En su mayoría estos permiten trabajar eficientemente ahorrando tiempo ya que admiten una mayor concentración en la resolución del problema en cuestión. Son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel que hay que proveer en un sistema funcional.

En lo adelante se presenta un conjunto de soluciones a problemas de software que sobre el marco de trabajo *Swing Application Framework* conforman la plataforma de trabajo propuesta. Es importante señalar que esta solución está enfocada al desarrollo de aplicaciones de Salud.

2.1.1 Modelo del dominio

El modelo de dominio tiene como objetivo construir, catalogar y diseminar un conjunto de componentes de software que tienen aplicación en el software actual y futuro dentro de un dominio de aplicación en particular. El objetivo final es crear mecanismos que les permitan a los ingenieros reutilizar los componentes desarrollados [30]. Este tipo de modelo es una alternativa cuando no se tiene un negocio bien definido.

Capítulo II. Solución Propuesta

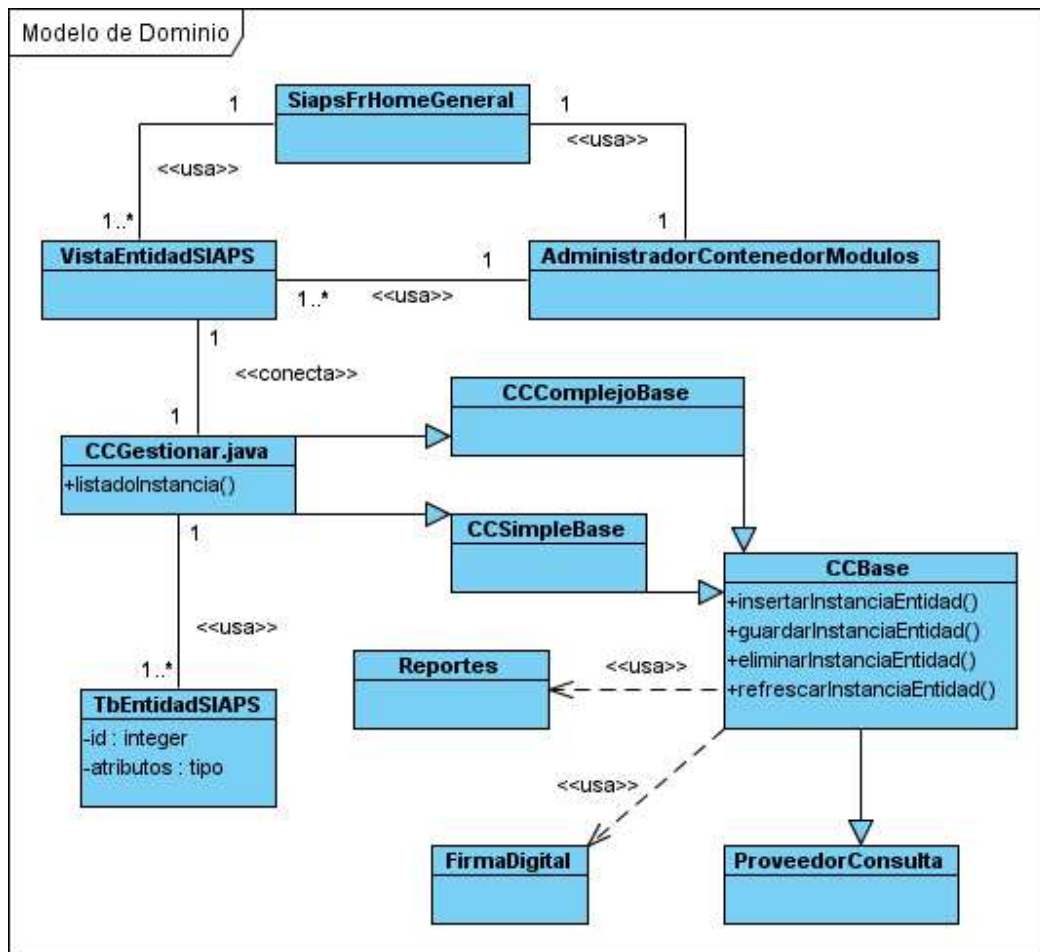


Fig. 3 Modelo de Dominio.

Se explican a continuación los conceptos que presentan en el diagrama:

SiapsFrHomeGeneral: Incluye toda la gestión general de la aplicación, manejo de módulos, opciones favoritas o otros elementos de configuración.

AdministradorContenedor módulos: Administra las configuración de los módulos de la Aplicación.

VistaEntidadSIAPS: es un formulario desktop que permite capturar los datos que serán persistidos en la base de datos, además de mostrar información útil al usuario.

CCGestionar: permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contiene.

Capítulo II. Solución Propuesta

TbEntidadSIAPS: representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes.

CCComplejoBase: permite encapsular las principales funcionalidades de la cuál heredan todas las clases controladoras para los CRUD complejos. (Cuando hay más de una entidad relacionada en el manejo de los datos o la funcionalidad).

CCSimpleBase: permite encapsular las principales funcionalidades de la cual heredan todas las clases controladoras para el CRUD simple. (Cuando hay solo una entidad relacionada en el manejo de los datos o la funcionalidad).

CCBase: define las principales funcionalidades de la cual heredan las clases *CCComplejoBase* y *CCSimpleBase*.

FirmaDigital: Permitirá la firma digital de documentos Clínicos en Formato XML.

ProveedorConsulta: define la forma en que se van a realizar las consultas a la BD.

2.1.2 Especificación de los requerimientos

La captura de los requisitos tiene como objetivo encontrar las necesidades de los clientes y representarlo de una forma adecuada para que puedan ser entendidos por los usuarios, clientes y desarrolladores. Es importante determinar cuáles son los verdaderos requisitos, pues de ello depende finalmente la satisfacción y aceptación del cliente [31].

En este caso, el objetivo es encontrar cuáles son las necesidades de los grupos de desarrollo a la hora de implementar las diferentes funcionalidades, o sea, determinar aquellos problemas horizontales que afectan en todos los desarrollos. A continuación se listan los requerimientos identificados.

R.1 Gestionar los datos de una funcionalidad definida para un proyecto determinado a través de un CRUD.

R.1.1 Desarrollar CRUD Simples.

R.1.2 Desarrollar CRUD Complejos.

R.1.3 Manejo de funcionalidades en ventanas hijas (*MDI child*).

Capítulo II. Solución Propuesta

R.2 Gestionar el paginado de los reportes emitidos.

R.2.1 Permitir definir cantidad de elementos por paginas.

R.2.2 Implementar un control visual.

R.3 Agilizar la presentación de informes con la plataforma JasperReport.

R.4 Permitir firmar documentos XML digitalmente.

R.5 Gestionar la seguridad.

R.5.1 Autenticar a los usuarios dentro de una aplicación desarrollada en la plataforma.

R.5.1 Configurar Roles, funcionalidades y permisos.

2.2 Diseño Propuesto

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

Descompone los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. Captura las interfaces entre los subsistemas en el ciclo de vida del software, lo cual es muy útil cuando se utilizan interfaces como elementos de sincronización entre diferentes equipos de desarrollo. [32].

Para un mejor entendimiento del funcionamiento se muestra en la figura 4 el diagrama general de diseño de clases que se debe tener en cuenta a la hora de usar la plataforma y se describen cada uno de los componentes del mismo. Se definieron, por la utilización del framework swing las clases, Swing component, Swing Contenedores y Swing app Framework, así como los paquetes Framework JPA, Framework Hibernate y Framework JasperReport. Algunos de estos elementos coinciden con los definidos en el modelo de dominio, por lo que solo se describirán los artefactos que no han sido descritos o se ampliarán su descripción con un mayor detalle.

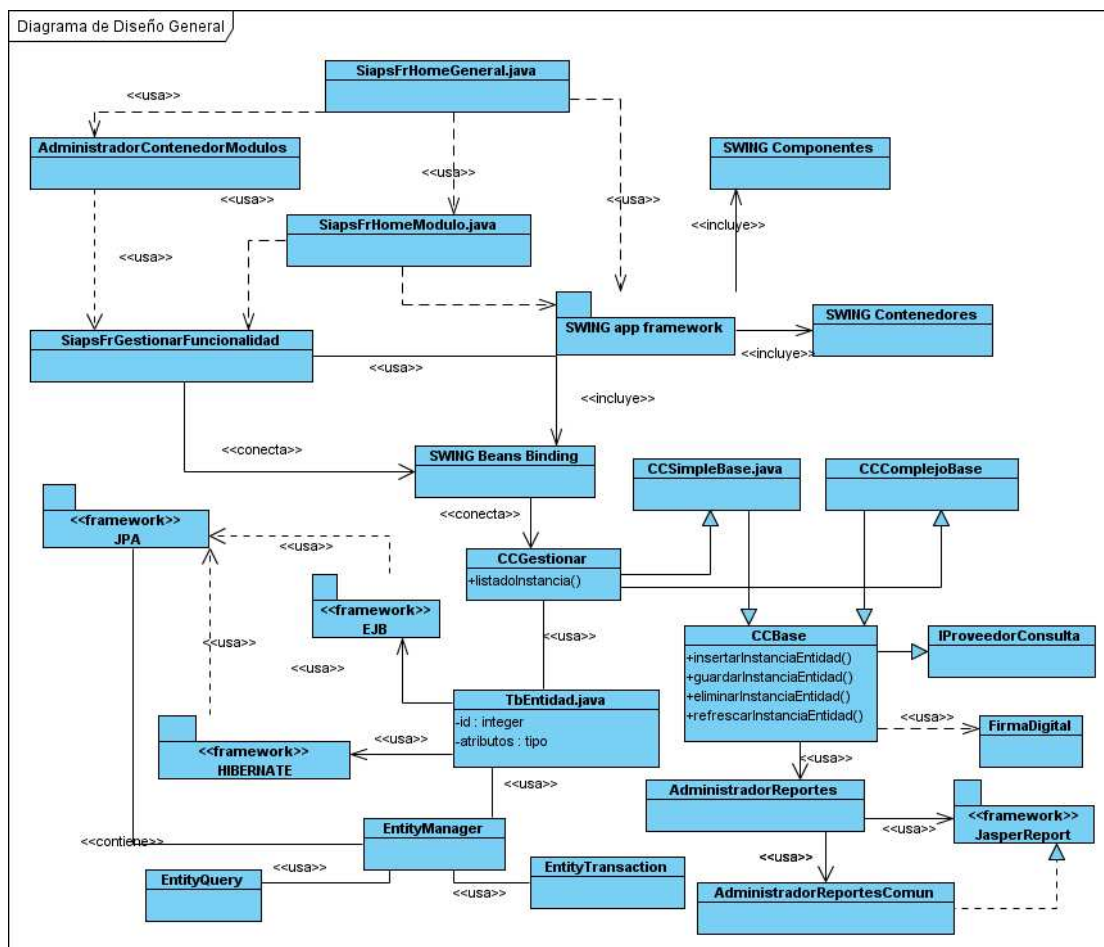


Fig. 4 Diagrama Diseño General.

SiapsFrGestionarFuncionalidad: es un formulario desktop que permite capturar los datos que serán persistidos en la base de datos, además de mostrar información útil al usuario.

CCGestionar: clase que permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contiene. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del marco de trabajo SWING.

TbEntidad: clase que representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las

Capítulo II. Solución Propuesta

clases servidoras para darle una respuesta a las páginas clientes. Hace uso del marco de trabajo Hibernate y JPA.

SwingApplication: marco de trabajo que provee la implementación prototipo de un conjunto de clases Java que permite tener un sistema de ventanas, componentes gráficos, independiente del sistema operativo y la librería de dibujo que estén disponibles en el ordenador donde se utilice. Contiene además:

- *Swing Componentes:* conjunto de componentes visuales (jtextbox, jlabel, jradiobutton, etc.) que permiten la construcción de la vista.
- *Swing Contenedores* conjunto de componentes visuales (jpanel, jframe, etc.) que permiten agrupar los *Swing Componentes* para la construcción de la vista.
- *SWING Beans Binding:* librería que permite el enlace dinámico entre los componentes visuales y los datos

CCComplejoBase: permite encapsular las principales funcionalidades de la cuál heredan todas las clases controladoras para los CRUD complejos. (Cuando hay más de una entidad relacionada en el manejo de los datos o la funcionalidad).

CCSimpleBase: permite encapsular las principales funcionalidades de la cual heredan todas las clases controladoras para el CRUD simple. (Cuando hay solo una entidad relacionada en el manejo de los datos o la funcionalidad).

CCBase: define las principales funcionalidades de la cual heredan las clases *CCComplejoBase* y *CCSimpleBase*. Implementa la interfaz *IProveedorConsulta*.

IProveedorConsulta: es una interfaz define la forma en que se van a realizar las consultas a la BD.

AdministradorReportes: es la clase que permite construir los reportes usando la biblioteca de clases JasperReport y usa la clase *AdministradorReportesComun*.

AdministradorReportesComun: es la clase que permite navegar por los datos a mostrar en el reporte y acceder a ellos a través de sus nombres. Hereda funcionalidades del framework JasperReport.

JasperReport: es una biblioteca de clases Java de código abierto desarrollada para facilitar el agregar capacidades de reporte a las aplicaciones Java. Permite realizar

Capítulo II. Solución Propuesta

reportes de código abierto que tiene como función el llevar documentos ricos en contenido a la pantalla, a la impresora, o a archivos PDF,HTML, XLS, CSV y XML.

EJB: es una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables utilizando el lenguaje de programación Java. Simplifica el desarrollo de aplicaciones Java y estandariza el API de persistencia para la plataforma Java.

EntityManager: permite realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) que se impliquen a las entidades. Se apoya para esto en:

- *EntityQuery*: clase que permite encontrar objetos persistentes manejando cierto criterio de búsqueda. Permite realizar peticiones a la base de datos y controla cómo se ejecuta dicha petición. Se utiliza para enlazar los parámetros de la petición, limitar el número de resultados devueltos por la petición y para ejecutar dicha petición.
- *EntityTransaction*: clase que permite realizar operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.

HIBERNATE: Conjunto de clases agrupadas en componentes que constituyen una herramienta de Mapeo objeto/relacional ó ORM de código abierto (Object Relational Mapping) y un generador de sentencias SQL.

JPA: conjunto de clases agrupadas en componentes que constituyen el API de persistencia incluida en el estándar EJB, aunque su uso no se limita a los componentes software EJB. Permite unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.

2.3 Modelo de datos propuesto.

Es el modelo que describe de manera abstracta cómo se representan los datos de una aplicación o sistema de información. Consiste en una descripción de algo conocido

Capítulo II. Solución Propuesta

como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores.

El Modelo de Datos tiene gran importancia en el ciclo de desarrollo de software, y de manera particular para la fase de implementación, pues define formalmente las estructuras permitidas y las restricciones que se aplican con el fin de representar los datos del dominio de la aplicación. Está compuesto por objetos: entidades que existen y se manipulan; y atributos: características básicas de dichos objetos y relaciones: forma en que se enlazan los objetos entre sí [33].

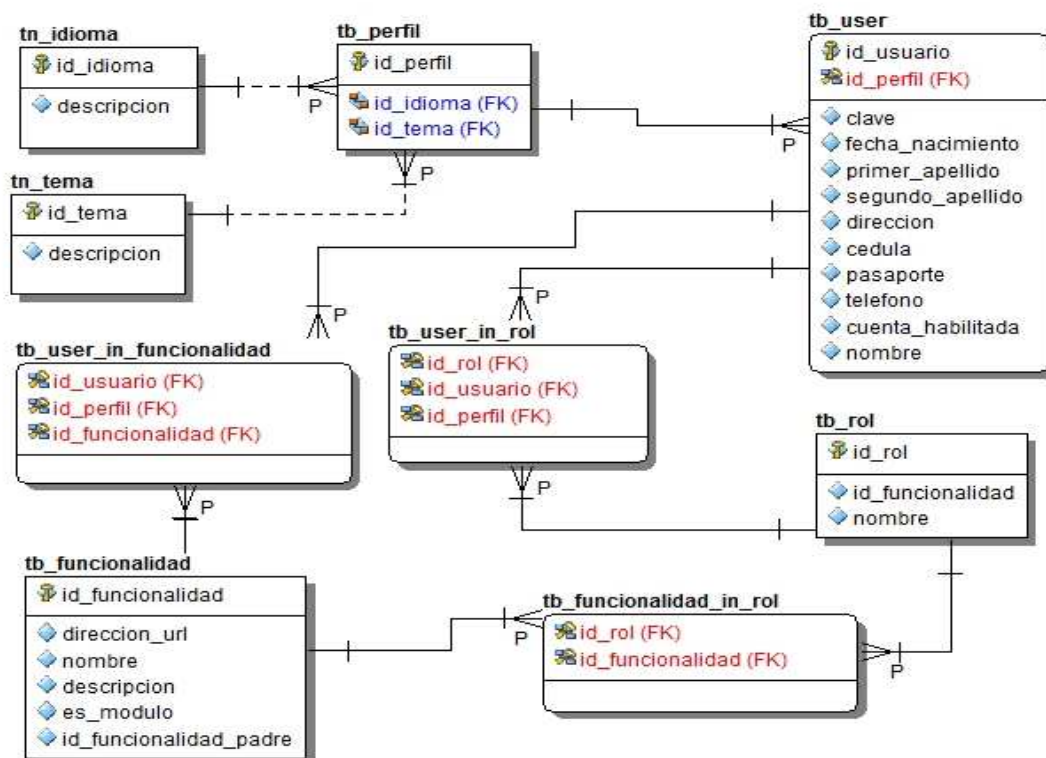


Fig. 4 Modelo de datos para la Seguridad.

Teniendo en cuenta que la presente investigación está orientada al desarrollo de una plataforma de trabajo y a que el modelo de datos va a depender de la aplicación en sí que se vaya a implementar solo se propone el modelo de datos que para el manejo de la seguridad. Teniendo en cuenta que forma parte importante de la solución propuesta.

Capítulo II. Solución Propuesta

Con el objetivo de un mayor entendimiento del Modelo de datos se describen a continuación de forma resumida las tablas representadas.

Nombre	Documentación
tb_user	Almacena los datos referentes al usuario
tb_perfil	Almacena los posibles perfiles de usuarios.
tb_rol	Almacena los posibles roles de usuarios.
tb_funcionalidad	Almacena las funcionalidades de los usuarios.
tb_user_in_funcionalidad	Almacena las relaciones entre el usuario, el perfil y las funcionalidades por las que ha transitado.
tb_funcionalidad_in_rol	Almacena las relaciones entre el rol del usuario y las funcionalidades por las que ha transitado.
tb_user_in_rol	Almacena las relaciones entre el rol del usuario, el usuario y el perfil de usuario por las que ha transitado.
tn_idioma	Almacena los tipos de idiomas.
tn_tema	Almacena clasificaciones de temas.

Tabla. 1 Descripción resumen de las tablas del Modelo de datos.

No se describen en mayor detalle los atributos de las tablas debido a cuestiones de seguridad.

2.4 Implementación.

En el flujo de trabajo de Implementación se comienza a partir del resultado obtenido en el diseño y se implementa el sistema en términos de componentes, es decir, ejecutables, ficheros, códigos de fuente y similares, necesarios para la implantación y despliegue del sistema. El objetivo fundamental es desarrollar la arquitectura y el sistema como un todo [34]. Se describen a continuación los principales elementos a tener en cuenta en la implementación de la solución propuestas y que son de vital importancia para el entendimiento del funcionamiento de la misma.

2.4.1 Plantillas CRUD

La necesidad de lograr soluciones que acorten los tiempos de desarrollo en los proyectos de software, conlleva a que se creen clases reutilizables tales como plataformas y/o bibliotecas de clases. Este epígrafe se centra en el diseño de plantillas

Capítulo II. Solución Propuesta

para la generación de CRUD, acrónimo de Crear, Obtener, Actualizar y Borrar (*Create, Read, Update y Delete en inglés*), así como el desarrollo de las clases asociadas a estos.

2.4.1.1 Problemática

Gestionar en su significado más general se define como la acción de administrar. Un Sistema de Gestión de Información vincula la generación y la aplicación de estrategias en un conjunto de etapas unidas en un proceso continuo enfocado en el uso racional, efectivo y eficiente de la información en función de los objetivos y metas de las compañías, en materia de desempeño y de calidad.

Cuando vinculamos lo anteriormente planteado a un sistema de software, se hace casi obligatorio referenciar lo que se conoce en computación como CRUD. Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un sistema de software.

Hoy en día existen tecnologías que permiten implementar este conjunto de acciones de forma automática, permitiendo hacer ingeniería inversa sobre la base de datos, llegando hasta la construcción de la interfaz visual, lo que acelera los tiempos de desarrollo y mejora la calidad del código generado ya que se rige por estándares previamente definidos que se mantienen en todo el código creado. Sin embargo la generación de código mediante ingeniería inversa para gestionar información desde la interfaz de usuario representada en bases de datos con relaciones de uno a mucho y mucho a mucho se hace en ocasiones imposible debido a la complejidad que pueden alcanzar las relaciones.

La plataforma Swing permite crear un CRUD [35] a partir de una base de datos pero con la limitante que solo permite hacerlo para una tabla no relacionada o para una tabla con relación de uno a mucho, faltando entonces la generación de los CRUD para relaciones mucho a mucho.

2.4.1.2 Solución

Para dar solución a la problemática planteada anteriormente se propone el uso de las plantillas genéricas usadas en el subsistema Consultorios para la generación de CRUD. Se puede decir que existen, teniendo en cuenta las bases de datos

Capítulo II. Solución Propuesta

relacionales, dos tipos de CRUD, los simples y los complejos, los primeros son aquellos que están relacionados con una sola tabla en la base de datos o con dos tablas relacionadas de uno a mucho, y los segundos pues los que están asociados a tablas relacionadas de mucho a mucho.

Antes de llegar a las plantillas y su forma de uso, que no son más que ficheros con una configuración preconcebida y que en este caso coinciden en cantidad con los tipos de CRUD señalados anteriormente, es preciso hablar de la jerarquía de clases creada y que hace posible que dichas plantillas funcionen correctamente.

BaseQueryProvider: Se encarga de proveer las consultas a través de Hibernate y de obtener las colecciones resultantes.

BaseCrudController: Contiene la definición de todos los métodos relacionados con las acciones de los CRUD, implementa los que funcionan de igual manera para los dos tipos referidos anteriormente.

BaseSingleCrudController y **BaseComplexCrudController:** implementan los métodos abstractos definidos en la clase **BaseCrudController**.

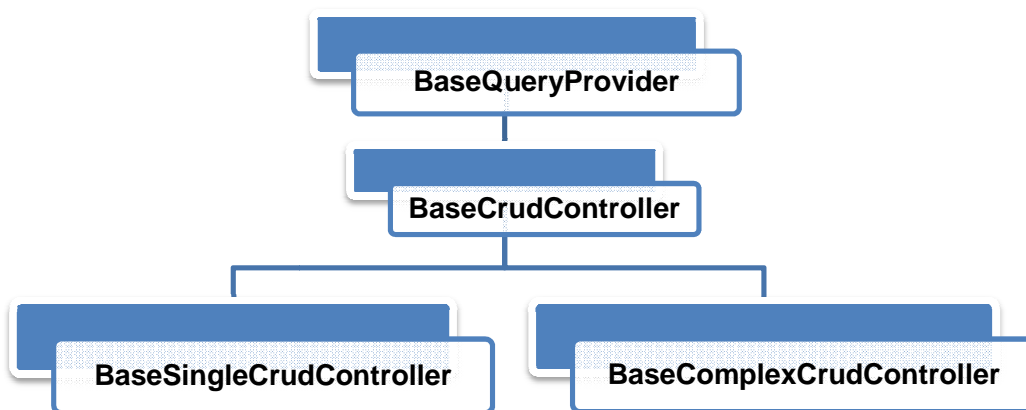


Fig. 5 Diagrama de clases CRUD.

Componentes Visuales.

Otro elemento fundamental es la presentación de contenidos al usuario, la plataforma Swing incorpora funcionalidades para el manejo de los componentes swing como el el API Beanbinding el cual facilita el enlace de las clases java con los distintos controles visuales. El volumen de datos de cualquier sistema de gestión, como es el caso del

Capítulo II. Solución Propuesta

subsistema Consultorios, presenta un problema a la hora de mostrar el resultado de las consultas sobre las bases de datos por lo que es muy usada la práctica de presentar solo parte del contenido y establecer controles para la paginación. Teniendo en cuenta que la plataforma usada no posee ningún componente visual que facilite este trabajo, los autores de este trabajo incluyen además en las un componente desarrollado para tales efectos.

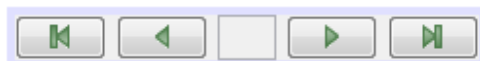


Fig. 6 Componente de paginación. JPager

Todo esto se encierra a modo de plantilla de manera que el desarrollador solo tiene que aplicarlas y personalizarlas según el proceso que se quiera implementar.

La utilización de estas plantillas en cualquier desarrollo que utilice la misma tecnología permitirá un incremento considerable en la productividad de los desarrolladores, así como una mejor calidad del código resultante.

2.4.2 Mejorando presentación de Informes.

El uso de plataformas o marcos de trabajo en la creación y generación de informes es usado con frecuencia en las aplicaciones informáticas empresariales, facilitando de este modo la toma de decisiones a varios niveles. Sobre como optimizar la creación de reportes que se tienen como resultado de la integración de las plataformas JasperReport e Hibernate apoyado en el uso de hilos, tiene objetivo el presente epígrafe.

2.4.2.1 Problemática

Hoy en día, el desarrollo de las organizaciones demanda una enorme cantidad de información, de ahí que las empresas están obligadas a tomar decisiones cada vez más precisas y con mayor rapidez. Por esta razón existen a nivel mundial una serie de aplicaciones que se encargan de la gestión de la información de manera muy eficiente.

Un requisito indispensable de los sistemas informáticos, es brindar la posibilidad de mostrar los resultados de la gestión de la información con diferentes grados de complejidad, exactitud y detalle, según sea requerido por los clientes y con un grado de personalización de las vistas, tal que satisfaga sus exigencias. Con este fin, existen en

Capítulo II. Solución Propuesta

el mundo varias herramientas, tanto libres como privativas, que permiten obtener y analizar información de bases de datos; precisamente son estas las herramientas para la generación de reportes. Las mismas posibilitan la materialización de la gestión de la información de una manera relativamente sencilla. Entre estas herramientas se pueden citar el JasperReports.

El CESIM es un centro de producción de software importante, reflejo de ello es la amplia gama de proyectos productivos que en ella se desarrollan, tanto para clientes nacionales como extranjeros. Los sistemas informáticos que se producen en él, en su mayoría brindan una abundante salida de información a modo de reportes, por lo que la funcionalidad que brinde soporte a la generación de las mismas es de vital importancia.

Integración básica JasperReport-Hibernate

Para integrar Hibernate y JasperReports, se debe definir un origen de datos JasperReports. Un enfoque sencillo e intuitivo es utilizar la fuente de datos `JRDataSource` o `JRBeanCollectionDataSource` que tiene como ancestro la clase `JRAbstractBeanDataSource`, ambas pertenecientes al marco de trabajo JasperReport [36]. Si se usa la primera (`JRDataSource`) entonces para las colecciones de objetos simples no habrá problemas:

```
List cats = em.createQuery("from Gatos g").
```

Mientras que cuando se consultan varios elementos como puede ser:

```
List cats = em.createQuery("Select, nombre, cumplenos, tipo from Gatos g").
```

Entonces esta clase no funciona y habría que crear alguna funcionalidad que permita referenciar los nombres a cada objeto en el arreglo devuelto.

Si se utiliza la segunda clase (`JRBeanCollectionDataSource`) el problema anterior relacionado con la asociación queda resuelto. Cuando la consulta de Hibernate devuelve una simple colección de objetos, la clase `JRBeanCollectionDataSource` perteneciente a la librería JasperReports funcionará bien, sin embargo cuando la consulta de Hibernate devuelve tuplas de los objetos (cada tupla como una matriz, cada matriz como un elemento en la lista que devuelta), el problema mencionado anteriormente no se resuelve tan fácilmente. JasperReports necesita una forma de

Capítulo II. Solución Propuesta

referencia de cada objeto en la matriz por un nombre de campo. Crear una clase que permita referenciar los nombres a cada objeto es una buena solución, pero es obligatorio a pasar una matriz de nombres de campo para comparar los resultados de la consulta.

Este enfoque funciona bien para las listas pequeñas. Sin embargo, para los informes de decenas o cientos de miles de líneas, es ineficiente ambas listas de elementos, pues consumen mucha memoria y es lento el procesamiento. La experiencia demuestra que, cuando se ejecuta en una configuración estándar, de una lista con tan sólo 10.000 elementos puede causar excepciones *OutOfMemory*. Asimismo, puede contaminar el período de sesiones de Hibernate y posiblemente de segundo nivel de caché con los objetos de carácter temporal [37].

Lo anteriormente planteado evidencia un problema de rendimiento trayendo como consecuencia lentitud en la generación del reporte cuando existan grandes cantidades de datos.

2.4.2.2 Solución

Integración Optimizada JasperReport - Hibernate

Se necesita una manera de leer y procesar de manera eficiente las consultas de Hibernate, sin crear demasiados objetos innecesarios en la memoria temporal. Una posible manera de hacerlo es la que se propone definiendo una capa optimizada para ejecutar consultas de Hibernate eficiente, otra para la abstracción de estas clases, siendo compatible con JasperReports y finalmente un envoltorio que puede ser conectado directamente a JasperReports, como se muestra en la figura 7.

Todas esta jerarquía de clases unida con el uso de hilos en el lenguaje de programación Java, para separar la creación y generación de los informes de los eventos de cualquier componente visual, mejora el rendimiento de cualquier aplicación que use estas tecnologías, ya que abstrae a los desarrolladores de todas estas funcionalidades que normalmente se hace manual, aumentando de esta forma el costo de implantación por ende demora en la culminación del proyecto.

Capítulo II. Solución Propuesta

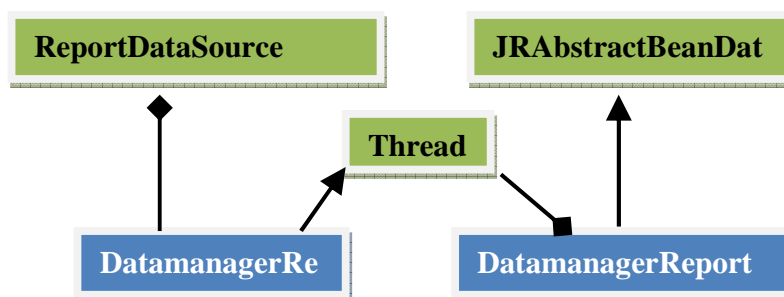


Fig. 7 Jerarquía de clases.

La optimización en la generación de informes en los sistemas operacionales suelen ser efectivos ya que reducen la carga en los servidores de bases de datos, aumenta la escalabilidad vertical de la aplicación y reduce la creación de errores en el sistema.

2.4.3 Seguridad y confiabilidad. Firmas digitales.

En el sector de la salud la seguridad y la confiabilidad son un tema crítico ya que todos los documentos clínicos referentes a un paciente en su mayoría deben ser confidenciales. Las decisiones de los galenos respecto que tratamiento adoptar respecto un paciente, así como las prescripciones de medicamentos para este último repercuten en su estado de salud, pudiendo en caso de error costarle, incluso, la vida, es por ello que se hace necesario garantizarla responsabilidad ante sucesos de este tipo.

Con el advenimiento de las Tic al sector de la salud, varios de los procedimientos médicos se realizan de forma electrónica por lo que es necesario buscar soluciones a los problemas anteriormente planteados. Muchos de los sistemas de información solo garantizan seguridad, integridad y confiabilidad a través de la usual práctica de la autenticación, supliendo una clave y usuario únicos para cada usuario del sistema, cifrando esta información de forma que nadie pueda interpretarla. ¿Que sucede si alguien se ha hecho con estos datos por otros medios? mediante un troyano o un *keylogger*, por ejemplo, en este caso podría suplantar la identidad sin problemas. La firma digital viene a suplir estas deficiencias.

La firma digital es un conjunto de datos generados mediante un algoritmo matemático y basado en técnicas criptográficas (de generación de claves) que se añade al documento que se quiere firmar y que permite no solo vincular ese documento a una

Capítulo II. Solución Propuesta

determinada persona o entidad, puede, además, asegurar la integridad del documento. En la figura 7 se muestra el proceso de firmado digital de un documento.

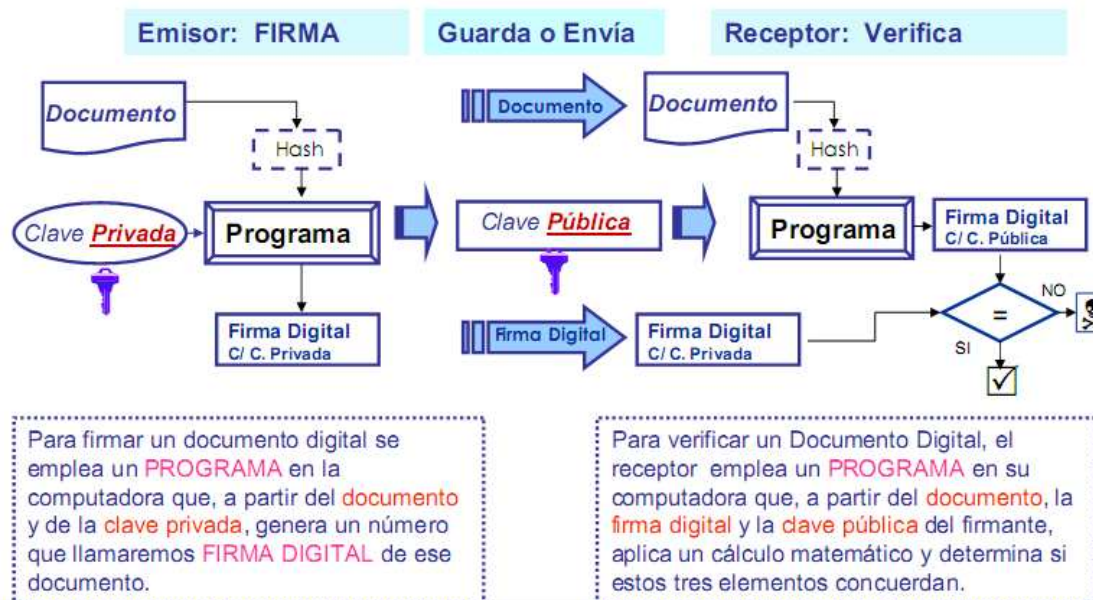


Fig. 8 Proceso de firmado digital.

La utilización de la firma digital asegura:

- La Identidad del signatario forma irrefutable,: la firma garantiza que el firmante o los intervinientes son quienes dicen ser.
- La integridad absoluta del mensaje: que el documento recibido sea exactamente el mismo que el emitido, sin que haya sufrido alteración alguna durante su transmisión.
- El origen, puesto que se garantiza que el firmante o las partes firmantes son quienes dicen ser, ninguno de ellos puede negar haber firmado, enviado o recibido el documento.
- La confidencialidad, el contenido al estar cifrado, sólo puede ser conocido por el firmante o por aquellos a quien el firmante autorice a acceder al documento firmado. El contenido del documento electrónico firmado sólo será conocido por quienes estén autorizados a ello.
- La temporalidad, por medio del Fechado (time stamping) podemos garantizar cuándo se generó esa información

Capítulo II. Solución Propuesta

Teniendo en cuenta todo lo anteriormente planteado, y que este trabajo esta orientado en alguna medida al desarrollo de aplicaciones para el sector de la salud, el autor del mismo consideró importante incluir dentro de la plataforma, una funcionalidad que permitiera la firma de documentos en formato XML, que es el formato que se utiliza para la generación de comentarios clínicos, debido a la especificación CDA del estándar HL7.

2.4.4 Gestión de seguridad. Roles y Usuarios.

Para el correcto funcionamiento de las aplicaciones que usen la plataforma para su desarrollo, se tuvieron en cuenta una serie de requisitos de seguridad con el objetivo que no se pueda afectar el funcionamiento del sistema finalmente implementado.

El sistema contará con un componente de seguridad encargado de la autenticación, auditoría y control de usuarios. Habrá una relación entre las tablas usuarios_roles y funcionalidad_roles permitiendo el acceso a la aplicación por tipo de usuario y dándole solo visibilidad a las áreas establecidas en dependencia del rol que posea.

Las bitácoras le permitirán al sistema llevar una traza de todas las operaciones llevadas a cabo por cada usuario mediante un registro de actividades con el objetivo de realizar reportes de todos los cambios realizados.

2.5 Conclusiones del capítulo

Luego de analizar los principales problemas a los que se enfrentó el autor de este trabajo, se llego a un modelo de dominio, un diagrama de clases general y un modelo de datos. Se desglosó la problemática a resolver en diferentes epígrafes cada uno de ellos tratando un problema distinto, proponiéndose además las soluciones a cada uno de estos.

El uso de las plantillas CRUD, de clases reutilizables y la optimización en la creación de informes ayudan a la finalización de todo proyecto de este tipo. Garantizan un mejor rendimiento y una mayor escalabilidad, al facilitar construir grandes cantidades de reportes en un menor tiempo con inmensos volúmenes de información. Propician además un menor costo de implementación, debido a que hay un aumento en la productividad del desarrollo por la abstracción que adquieren los desarrolladores y por el código completamente auditado que se genera. Por último y no menos importante se

Capítulo II. Solución Propuesta

facilitan los procesos de gestión de seguridad y se garantiza la veracidad e integridad de la información generada al brindarse la posibilidad de firmar digitalmente documentos en formato XML.

Capítulo III. Análisis de Resultados

CAPÍTULO III: ANÁLISIS DE RESULTADOS

El objetivo de este capítulo es validar la solución propuesta y de garantizar que cumple con los requerimientos necesarios para su utilización. Para ello se describen de forma resumida los procesos de diseño e implementación de dos módulos desarrollados para el subsistema de escritorio del sistema alasSIAPS usando la plataforma propuesta. Estos mismos dos módulos se han desarrollado en una versión web, de manera que se establecerá además una comparación en cuanto a tiempo de desarrollo de estos. Por otro lado se decidió evaluar el trabajo a través de un comité de expertos, los cuales fueron escogidos entre los desarrolladores con experiencia en la implementación de aplicaciones para el alasSIAPS y han tenido resultados comprobados en esta actividad.

3.1 Concepción del proyecto

En el marco de la informatización de la sociedad cubana, el sector de la salud se propone contar con una solución informática integral que dote al sistema de mayor grado de acceso a información unificada y confiable en tiempo real, que aporte la rapidez y fiabilidad necesaria para las modernas técnicas de administración y para la toma de decisiones en el nivel de Atención Primaria de Salud.

El Departamento de Atención Primaria de Salud del Centro de Informática Médica plantea como solución el sistema alasSIAPS. Este contará con dos Subsistemas, un subsistema web y uno de escritorio. El subsistema web será el más robusto y recomendado para instituciones que posean una infraestructura tecnología de alto nivel. El subsistema de escritorio es para instituciones con infraestructura limitada (subsistema más ligero). Este subsistema es una aplicación MDI (interfaz de múltiples documentos) para la gestión de la información vinculada al paciente en la atención primaria de salud y que se enmarca en la población asociada a un área de salud determinada.

3.2 Módulo Configuración.

El Sistema alasSIAPS incluye módulos como: Clínico Quirúrgico, Medicina Familiar, Medios de Diagnóstico, Enfermería entre otros, cada uno presenta características y objetivos específicos de acuerdo a la función que realice en los centros de salud. Por lo

Capítulo III. Análisis de Resultados

que es necesario lograr una interoperabilidad o interrelación entre los módulos, garantizando el buen funcionamiento en el intercambio de los datos.

El consenso sobre qué estándares utilizar y cómo utilizarlos juega un rol fundamental, ya que no existe un axioma sobre cómo llegar a la interoperabilidad. Actualmente existen deficiencias en cuanto al uso preciso de estándares y de clasificaciones internacionales de los problemas de salud, proceso de codificación en el cual se precisen y estandarizan cada una de las entidades a medir mediante un vocabulario controlado, todas estas deficiencias hacen engorrosa la labor que realizan los especialistas. Por lo que se debe permitir el manejo adecuado de nomencladores médicos, codificación de los problemas de salud y la utilización de estándares reconocidos internacionalmente tales como CIE, CIAP, EDO y CPAPS. Además es necesario manejar la seguridad del sistema, teniendo en cuenta que en cualquier sistema sanitario, una parte importante de la información manejada es sensible y debe ser protegida.

3.2.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, para el desarrollo del módulo en su primera versión se definieron 47 de estos. Divididos en los siguientes paquetes funcionales:

- Áreas de Salud.
- CIAP (Clasificación Internacional para la Atención Primaria)
- CIE (Clasificación Internacional de Enfermedades).
- CPAPS (Clasificación de Grupos Problemas de Atención Primaria de Salud)
- EDO (Enfermedades de Declaración Obligatoria)
- Estado del Paciente.
- Grupo Etario.
- Personal de Salud.
- Seguridad.
- Ubicación Geográfica.
- Unidades de Salud.

Capítulo III. Análisis de Resultados

3.2.2 Diseño e Implementación

Para el diseño e implementación de cada uno de estos paquetes funcionales se utilizó la plataforma propuesta como solución de esta investigación, lo cual se puede constatar en cada uno de los diagramas de diseño que aparecen en el Expediente de Proyecto [38]. Como muestra de lo planteado anteriormente a continuación se presenta el diagrama de clases del gestionar Grupo Etereo.

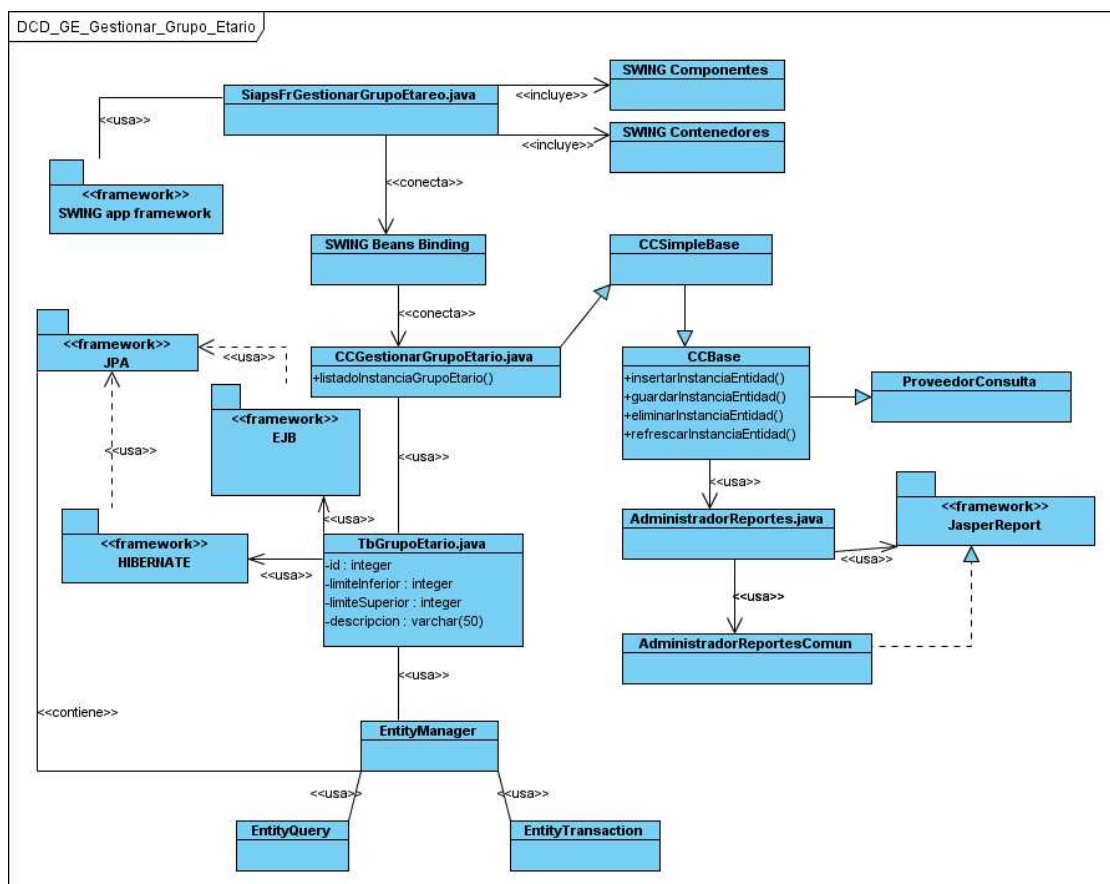


Fig. 9 Diagrama de Clases del Diseño Gestionar Grupo Etereo.

Cada una de las funcionalidades del módulo fueron diseñadas siguiendo este “patrón” de diseño, ya que como se mencionaba se implementaron a partir del uso de la plataforma y aplicando todas las soluciones propuestas en el Capítulo II del presente documento, demostrándose de esta forma el uso de la misma.

Capítulo III. Análisis de Resultados

3.3 Módulo Medicina Familiar

En el módulo de Medicina Familiar se gestionan los procesos básicos y fundamentales de la APS, es el núcleo del sistema, contiene los datos generales del paciente, dígase Historia Clínica Individual, Historia de Salud Familiar, la planificación-seguimiento y control del paciente, los ingresos en el hogar y seguimiento diario, las remisiones, la captación del recién nacido y la dispensarización.

3.3.1 Requerimientos Funcionales

Para el desarrollo del módulo en su primera versión se definieron 28 requerimientos funcionales. Divididos en los siguientes paquetes funcionales:

- Captación del recién nacido.
- Dispensarización.
- Codificadores.

3.3.2 Diseño e Implementación

Al igual que con el módulo de Configuración, para el diseño e implementación de cada uno de estos paquetes funcionales se utilizó la plataforma propuesta como solución de esta investigación, pudiéndose constatar en cada uno de los diagramas de diseño que aparecen en el Expediente de Proyecto [38]. Evidencia de ello es el diagrama de clases del Gestionar estilo de vida.

De manera similar al módulo de configuración cada una de las funcionalidades del módulo fue diseñada siguiendo este “patrón” de diseño, y se implementaron a partir del uso de la plataforma y aplicando todas las soluciones propuestas en el Capítulo II del presente documento, demostrándose de esta forma una vez más la puesta en práctica de dicha solución.

Capítulo III. Análisis de Resultados

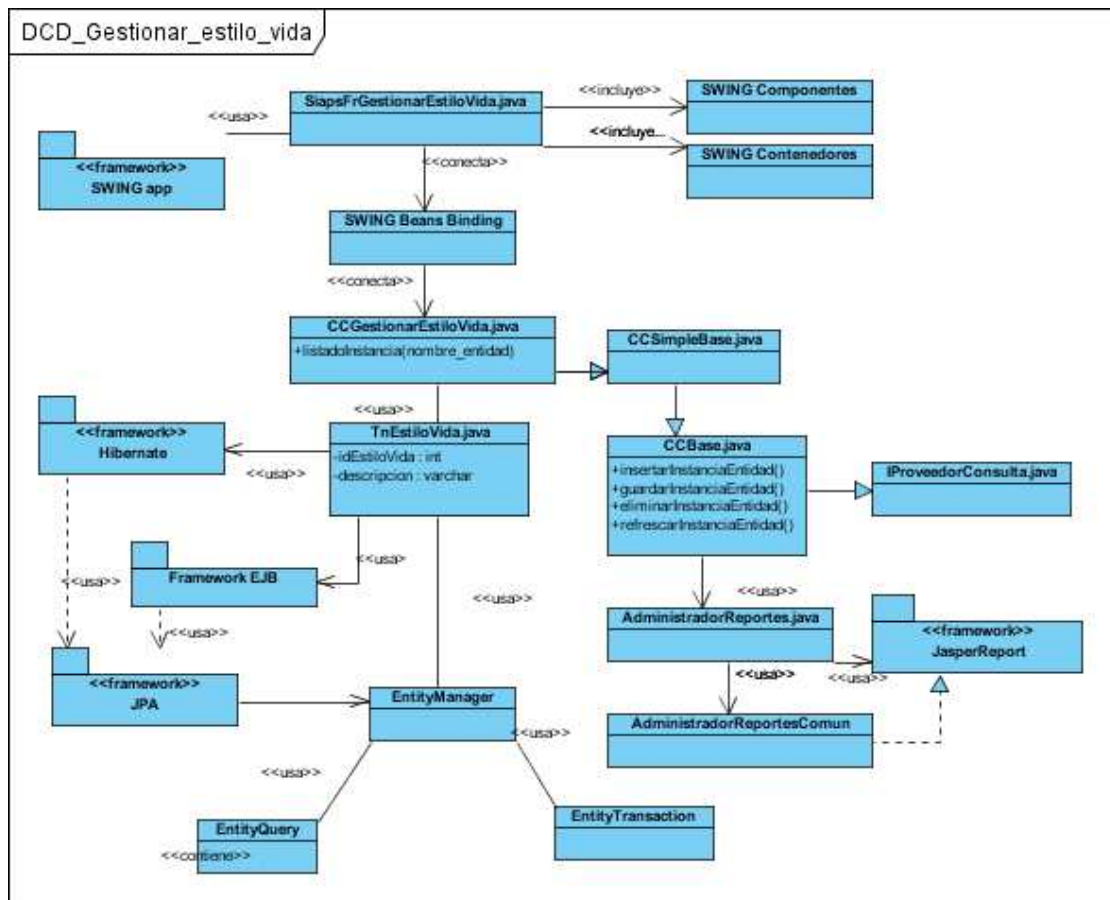


Fig. 10 Diagrama de Clases del Diseño Gestionar Estilo de vida.

3.4 Tiempo de desarrollo

Para el desarrollo de los módulos que se mencionaron con anterioridad se utilizaron como recursos humanos a un analista, dos programadores en dos equipos de trabajo diferentes, comenzando sus tareas el mes de noviembre de 2009 y concluyendo la implementación en mayo del 2010, como se puede ver en poco más de 6 meses se desarrollo este módulo. Al mismo tiempo se desarrollaba en otros dos equipos de trabajo con la misma cantidad y distribución de recursos humanos los mismos módulos pero en versión Web, sin embargo desarrollaron su versión en poco más de 8 meses ya que comenzaron el desarrollo en el mes de septiembre y finalizaron el mismo en el mes de junio. La puesta en manos de los desarrolladores de la plataforma les facilitó el trabajo propició que se disminuyeran los tiempos de producción, que estaban estimados en un inicio para desarrollarse en 8 meses.

Capítulo III. Análisis de Resultados

3.5 Rendimiento

El subsistema alasSIAPS para escritorio fue confeccionado como parte de una variante de solución ligera a la APS, el anterior planteamiento lo anterior se evidencia de manera cuantitativa de la siguiente manera.

Las base de datos embebida o incrustada son ficheros que se colocan dentro de los paquetes de una aplicación y que se ejecuta dentro de su espacio de memoria, siendo únicamente accedida por ésta. O sea, cuando se inicia el sistema se crea la BD en memoria, ejecutándose dentro de esta, guardando en los ficheros todas las acciones realizadas y cuando se cierra esta se finaliza además la BD de la memoria. Haciendo que el sistema sea mucho más rápido que otras. Esta es la posibilidad que brinda el haber usado HSQLDB.

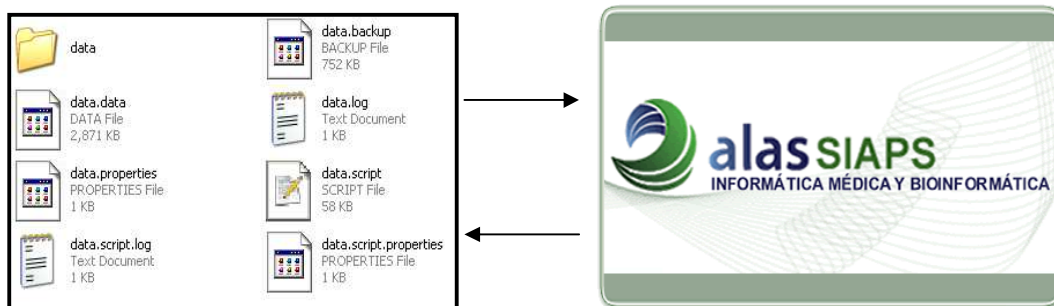


Fig. 11 HSQLDB en alasSIAPS.

Para optimizar el consumo de memoria de la aplicación, una de las opciones es actualizar el JVM (*Máquina Virtual de Java*) o JRE (*versión optimizada de la Máquina Virtual de Java*). El autor recomienda que preferentemente se tenga instalado un JRE optimizado, a continuación se describen algunas optimizaciones que se le pueden hacer al mismo y que fueron puestas en práctica por el autor.

Java Quick Starter (JQS) mejora el tiempo de inicio de la mayoría de las aplicaciones de Java. JQS se activa de forma predeterminada en los sistemas operativos Windows XP y Windows 2000. Las mejoras en el rendimiento de inicio se obtienen transfiriendo periódicamente algunos de los archivos Java más utilizados a la memoria (se ocupa menos de 20Mb de RAM). Una vez que Java se ha iniciado, se necesita una cantidad de discos de E/S mucho menor, por lo que el inicio es mucho más ligero y rápido.

Capítulo III. Análisis de Resultados

Una vez funcionando el sistema:

La aplicación solo necesita aproximadamente 60 Mg de Memoria RAM, hasta 80. Se espera que una vez optimizado el código y las librerías que el sistema usa, las cifras en cuanto al rendimiento disminuirán considerablemente.

Durante varias pruebas realizadas a la BD de 3 consultas con 1 único hilo, en una PC con memoria de 256 Mb y un microprocesador Intel Celeron a 1.3 GHz, arrojó como resultados los mostrados en la tabla 1. Teniendo en cuenta que la aplicación solo se ejecutará en un ordenador al mismo tiempo, es decir solo habrá un hilo corriendo, el autor considera que es suficiente para demostrar la factibilidad de su uso en este sentido.

Para 200 tuplas:				
Tiempo (ms)	Latencia* (ms)	Tamaño (Bytes)	Media (ms)	Transferencia (Kb/sec)
15 - 16	1	5050	15	2.7
Para 2 000 tuplas:				
772 - 805	1-2	49 061	487	15.5

Tabla 1. Resultados a prueba de rendimiento

Como puede evidenciarse en la tabla para gran cantidad de peticiones los resultados se comportan de manera aceptable, teniendo en cuenta que la aplicación se desplegará en una unidad de salud y la cantidad de pacientes a atender será solo la de un área de salud.

3.6 Validación por Comité de Experto

Para la validación por el comité de experto, se escogió el método DELPHY, que es uno de los más utilizados con este objetivo. El método se utiliza para la realización de pronósticos con base subjetiva donde se utiliza la intuición como una comprensión sinóptica de la realidad y basados en la experiencia y conocimientos de un grupo de personas considerados expertos en la materia a tratar [39].

Se entiende por experto, tanto al individuo en sí como a un grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia.

Capítulo III. Análisis de Resultados

Para la descripción de los expertos, desde el punto de vista de la valoración de la calidad de la solución del problema, se puede mencionar las siguientes características esenciales: competencia, creatividad, disposición a participar en la encuesta, conformismo, capacidad de análisis y de pensamiento, espíritu colectivista y autocrítico.

3.6.1 Selección de los expertos

La selección de los expertos partió de base de que el propósito era garantizar juicios autorizados en el tema y por consiguiente validez en la información. Por lo que se utilizó el muestreo intencional, se identificaron especialistas que llevaban como promedio tres años en el desarrollo de aplicaciones para el Centro de informática médica, y con experiencias en el trabajo relacionado con la plataforma java.

Luego de haber seleccionado a los posibles expertos se les explicó en qué consistía el método que se aplicaría, luego de haber dado su consentimiento para la evaluación, se les entregó un cuestionario impreso (anexo 1) que debían responder con el objetivo de valorar su nivel de competencia y conocimiento sobre el tema, a fin de determinar su experticia en el tema.

En esta metodología la competencia de los expertos se determina por el coeficiente k , el cual se calcula de acuerdo con la opinión del experto sobre su nivel de conocimiento acerca del problema que se está resolviendo y con las fuentes que le permiten argumentar sus criterios [40].

El coeficiente de competencia se calcula por la siguiente fórmula:

$$K = \frac{1}{2} (k_c + k_a),$$

Donde:

k_c : es el coeficiente de conocimiento o información que tiene el experto acerca del problema, calculado sobre la valoración del propio experto en una escala del 1 al 10, de esta forma, la evaluación "1" indica que el experto tiene muy poco conocimiento de la problemática correspondiente, mientras que la evaluación "10" significa todo lo contrario.

Capítulo III. Análisis de Resultados

ka: es el coeficiente de argumentación o fundamentación de los criterios del experto, obtenido como resultado de la suma de los puntos alcanzados a partir de una tabla patrón como la siguiente:

FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.2
Trabajos de autores nacionales	0.05	0.05	0.05
Trabajos de autores extranjeros	0.05	0.05	0.05
Su propio conocimiento del estado del problema en el extranjero	0.05	0.05	0.05
Su intuición	0.05	0.05	0.05

Después del cálculo de k , se determina el nivel de competencia según la siguiente tabla:

	Coeficiente de competencia
$0,7 < K < 1,0$	Alto
$0,4 < K < 0,7$	Medio
$K < 0,4$	Bajo

3.6.2 Número óptimo de expertos

Aunque no hay forma de determinar el número óptimo de expertos para participar en una encuesta Delphi, estudios realizados por investigadores de la Rand Corporation [41], señalan que si bien parece necesario un mínimo de siete expertos habida cuenta que el error disminuye notablemente por cada experto añadido hasta llegar a los siete expertos, no es aconsejable recurrir a más de 30 expertos, pues la mejora en la previsión es muy pequeña y normalmente el incremento en coste y trabajo de investigación no compensa la mejora.

En el procesamiento de esta ronda de información, se determinó que de los expertos encuestados 8 fueron seleccionados, pues presentaron un coeficiente de competencia entre 0,8 y 1,0.

Capítulo III. Análisis de Resultados

3.6.3 Aplicación del método

Con el objetivo de que reflejaran cuáles eran los principales problemas enmarcados en el campo de acción del presente trabajo se les aplicó a los expertos un segundo cuestionario (anexo 2). Esto además permitiría comprobar el nivel de coincidencia entre la opinión de los expertos y los elementos que se tuvieron en cuenta en la presente investigación respecto a los problemas a los que se debe enfrentar un desarrollador durante el proceso de implementación de sistemas informáticos de escritorio en el CESIM. Ofrecería además, una medida de la necesidad real de dar solución a dichos problemas.

En una tercera etapa, se le aplicó al comité de experto otra encuesta (anexo 3), para evaluar finalmente la propuesta que se hace en el presente trabajo. Para responder a dicha encuesta se les entregó además la documentación necesaria que explicaba las principales características a de la plataforma de desarrollo propuesta.

En esta encuesta se reflejan los criterios a evaluar en la propuesta, donde el experto debe responder si cada uno de esos criterios son: insuficientes, necesarios pero no suficientes o necesarios y suficientes. A parte de ello, el comité de experto debe responder otras preguntas que son realizadas con el propósito de valorar la utilidad e importancia de la propuesta [42].

Para el procesamiento de esta tercera ronda de información, se calculó la media aritmética en cada uno de los criterios a evaluar en cuanto al nivel de importancia de los mismos. Interpretando los resultados, “1” es de poco valor, y “10” es el máximo valor posible.

3.6.4 Resultados de la evaluación de las encuestas

Siguiendo los principios del método utilizado estas encuestas se aplicaron de manera individual y bajo total anonimato, con el objetivo de que ningún *“miembro del equipo fuera influenciado por la reputación de otro o por el peso que supone la opinión de la mayoría... El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos”* [43]. Esto posibilita obtener los verdaderos criterios de los participantes.

Capítulo III. Análisis de Resultados

Concluida la aplicación de la segunda encuesta y luego de procesada la información extraída de esta, se obtuvieron los principales los problemas a los que se debe enfrentar un desarrollador durante el proceso de implementación de sistemas informáticos de escritorio en el CESIM y que entorpecerían el desarrollo exitoso de los mismos. Se relación a continuación dichas limitantes junto con una breve explicación de cómo fueron resueltos en la propuesta de este trabajo:

1. **El tiempo necesario para el crear o re-implementar componentes de interfaz de usuario.** Siempre es necesario plantearse la necesidad de utilizar algún framework para el desarrollo de aplicaciones de escritorio, ya que la implementación de componentes de interfaz de usuario no es algo sencillo, es por ello que la propuesta incluye el uso del Swing Application Framework, re-implementa dos componentes del mismo y propone un componente para paginado (Jpager).proporcionando una mayor usabilidad al sistema que se desarrolle y aminorando los tiempos de entrega de los sistemas a los clientes.
2. **Demora en el inicio de la aplicación desarrollada.** La plataforma propone el uso de una primera pantalla de bienvenida (Splash Screens), con esto el usuario estará entretenido mientras se van cargando en segundo plano todos los datos que se necesiten y una vez cargada la pantalla principal de la aplicación, se muestran los iconos que dan acceso a los módulos según los privilegios del usuario.
3. **Trabajo con tareas en segundo plano.** En este sentido uno de los elementos que explota la aplicación es el uso de hilos de trabajo de manera extensiva. Se utilizan en las plantillad CRUD y en la presentación de informes.
4. **Aplicaciones “congeladas”.** Esto se escapa un poco de las manos del alcance de la solución propuesta, pero el esqueleto funcional o contenedor de funcionalidades que se propone incluye una barra de estado, en la cual se incluye una barra de progreso, que usada de forma inteligente permite avisar al usuario de lo que está pasando. El uso de hilos también viene solucionar este problema. La base de datos va a ejecutarse dentro del espacio de memoria de la propia aplicación y esto permite una velocidad alta.
5. **Cargar más información que la necesaria.** Se propone un diseño del tipo MDI (interfaz de múltiples documentos) por lo que cada una de las funcionalidades

Capítulo III. Análisis de Resultados

deberá ser implementada en una ventana diferente y solo con la información necesaria. Se implementa un componente visual para el paginado Jpager, que implementa el patrón *Page-by-page iterator*. El uso de Hibernate también ayuda en este apartado.

6. **Pre-cargar toda la información útil que se pueda.** En aplicaciones de salud sobre todo muchas veces esto es necesario. Se propone el uso de HSQLDB que es un motor de bases de datos que va a ejecutarse dentro del espacio de memoria de la propia aplicación, esto permite una velocidad muy alta, las conexiones sólo se harán desde dentro de la propia máquina virtual de la aplicación, por lo que la información siempre estará disponible.
7. **Presentación de reportes demasiado cargadas.** Se propone el uso del patrón *Page-by-page iterator*, a través del uso del componente *Jpager* propuesto.
8. **Aplicaciones multiplataforma.** Uno de los elementos por los que se desarrolla la plataforma en arquitectura java es precisamente este.
9. **Compleja reutilización de código.** La inclusión de las plantillas CRUD, las mejoras al JasperReport, mitiga este efecto.

Al analizar con detenimiento estos resultados se puede percibir que la mayoría de los elementos que se tuvieron en cuenta para hacer la propuesta de solución en el presente trabajo, coincidieron con la mayoría de los elementos identificados por los expertos, como problemas que entorpecerían el desarrollo exitoso de sistemas informáticos de escritorio en el CESIM.

Por otra parte la tercera encuesta arrojó los siguientes resultados:

El 87 % de los expertos consideró que, el uso de plantillas CRUD, son necesarios y suficientes para resolver los problemas planteados en este apartado, mientras que el resto lo consideró necesario pero no suficiente. El 100 % consideró que la gestión de la seguridad era necesaria y suficiente.

El 75 % de los expertos consideró que el uso de hilos y la optimización del framework JasperReport, el uso del motor de bases de datos HSQLDB y el ORM Hibernate y La firma digital de documentos XML, eran necesarios y suficientes. El 62,5 % consideró que el uso del componente visual propuesto para el paginado, Jpager, era necesario y suficiente.

Capítulo III. Análisis de Resultados

Estos resultados se reflejan en la siguiente tabla, donde: “I” es insuficiente, “NNS” es necesario pero no suficiente y “NS” es necesario y suficiente.

Criterio a evaluar sobre la propuesta	Cantidad de Expertos			Por Ciento (%)		
	I	NNS	NS	I	NNS	NS
Uso de las plantillas CRUD	0	1	7	0.00%	12.50%	87.50%
Uso de hilos y la optimización del framework JasperReport	0	2	6	0.00%	25.00%	75.00%
Uso del componente visual propuesto para el paginado, <i>Jpager</i> .	0	3	5	0.00%	37.50%	62.50%
Uso del motor de bases de datos HSQLDB y el ORM Hibernate	0	2	6	0.00%	25.00%	75.00%
La firma digital de documentos XML	0	2	6	0.00%	25.00%	75.00%
La gestión de la seguridad	0	0	8	0.00%	0.00%	100.00%

Otro de los resultados de esta tercera encuesta está relacionado con la media aritmética de cada uno de los criterios a evaluar. En la siguiente tabla se refleja el grado de importancia, como promedio, que le conceden los expertos a cada uno de estos elementos.

	Cantidad de Expertos que evaluó cada criterio										
	Evaluaciones										
Criterio	1	2	3	4	5	6	7	8	9	10	Media Aritmética
1								2	4	2	9
2									2	6	9.75
3						1		3	2	2	8.5
4								2	3	3	9.125
5								1	3	4	9.375
6								2	2	2	9

A partir de que los valores que están cercanos a “10” indican que esos criterios evaluados son de máxima importancia, mientras que los cercanos a “1” indican lo contrario y de los resultados obtenidos, se puede comprobar que cada uno de los elementos que evaluados, fueron considerados por los expertos de gran relevancia, pues los valores de la media aritmética oscila entre 8,5 y 9,75.

Capítulo III. Análisis de Resultados

3.7 Conclusiones del capítulo

La solución propuesta en esta investigación se validó en primera instancia a través de su puesta en práctica en el proyecto alasSIAPS, que se desarrolla en el CESIM. En el primer epígrafe de este capítulo se describen los elementos más importantes que lo demuestran su uso. Se desarrollaron en una primera versión los módulos de Configuración y de Historia de Salud Familiar. Además se dan algunos elementos de rendimiento a partir de pruebas hechas a la aplicación antes mencionada.

Para la validación del presente trabajo se aplicó también el método Delphi siguiendo la metodología que este plantea, lo que permitió la aplicación de las distintas encuestas y su procesamiento, con el cual se obtuvieron resultados satisfactorios. Se demostró de esta manera la importancia y necesidad de poner en marcha la aplicación de la propuesta presentada, en los sistemas que sean desarrollados en el CESIM para entornos desconectados o con bajas prestaciones en sus estaciones de trabajo.

La validación contó con el concurso de un grupo de expertos con bastante experiencia en el desarrollo de aplicaciones para el Salud y trabajo en la plataforma java, los cuales tenían entre 3 y 4 años de trabajo en esta actividad.

CONCLUSIONES

El desarrollo del presente trabajo permitió la construcción de una plataforma para el desarrollo de aplicaciones de escritorio, la que a su vez facilitará el desarrollo de aplicaciones de software que ayuden en la gestión de la información médica asociada al paciente en entornos donde la conectividad es casi nula y la infraestructura tecnológica limitada. Algo que se evidenció en la puesta en práctica de la misma con la implementación de los principales requisitos funcionales de los módulos de Historia de Salud Familiar y el de Configuración.

El desarrollo de la solución propuesta propicia una mejor calidad y completitud de los productos de software desarrollados con la misma, ya que:

- Código generado completamente auditado.
- Reducción de errores en el sistema.
- Mayor rendimiento en la generación de informes.
- Fiabilidad de los documentos clínicos.

RECOMENDACIONES

Luego de realizada la investigación y teniendo en cuenta las consultas realizadas a un conjunto de expertos en el tema, se identificaron algunos aspectos que deben ser tratados en próximas investigaciones, los cuales se listan a continuación:

- Extender las funcionalidades de las plantillas CRUD para que permitan generar componentes visuales en función de las funcionalidades que se implementarán con estas.
- Implementar el componente bitácora del componente de seguridad.
- Incluir en la plataforma un componente que permita la gestión de servicios web.
- Implementar un componente que permita la utilización de algún estándar de comunicación entre aplicaciones médicas como HL7.

Referencias Bibliográficas

REFERENCIAS BIBLIOGRÁFICAS

1. **MARÍN DÍAZ, M. E.** *Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática* 2006.
2. **LUNA, C.** OMS "La revolución de la Atención Primaria en Cuba cumple 30 años". 2008, [citado 13/08/08]. [Disponible en: <http://www.who.int/entity/bulletin/volumes/86/5/08-030508/es/index.html>].
3. **VIDAL, L., M. AND Y. DE ARMAS.** *Estrategias de Informatización del Sector de la Salud (I)*. In *Revista InforMéd.*, 2002.
4. **MARÍN DÍAZ, M. E.** *Pilares y debilidades del proceso de informatización de la Atención Primaria de Salud (APS)*. In. Ciudad de La Habana: Inédito. Taller de Informática en Salud. Universidad de las Ciencias Informáticas, 29 y 30 de abril 2004.
5. **OPS** *Renovación de la Atención Primaria de Salud en las Américas*. Documento de posición de la Organización Panamericana de la Salud, 2007.
6. **EUROPA, S. D. L. I.** *La información puede salvarle la vida. Portal Temático. Calidad de Vida Salud.*, 2007. [Disponible en: http://ec.europa.eu/information_society/tl/qualif/health/index_es.htm]
7. **SANITARIA, G.** *Sanidad sin papeles*, 2009. [Disponible en: <http://www.expansion.com/2009/02/03/funcion-publica/1233655976.html>]
8. **SANGUINETTI C., H. A., FÁBREGA R., SALAZAR A., GAVILANES P., BOLAÑOS J.** *Las TIC y el Sector Salud en Latinoamérica* [online]. 2008 [citado 2011]. [Disponible en: www.telefonica.com.pe/fundacion].
9. **FUENTES., P.** *Experiencias sobre la importancia de las comunicaciones en los consultorios médicos de la familia*, 2011. [Disponible en: <http://www.cubasolar.cu/biblioteca/Energia/Energia02/HTML/articulo06.htm>]
10. **HERNÁNDEZ, M. C.** *Gestión de la integración del proyecto Registro Informatizado de Salud*. Maestría en Gestion de Proyectos, 2009.
11. **AUTORES, C. D.** *Infomed Red de salud en Cuba. Fortalecimiento de la Informática en el Policlínico*. [Disponible en: http://aps.sld.cu/bvs/materiales/carpeta/fortalecimiento_info.pdf]
12. *Ídem a la anterior.*

Referencias Bibliográficas

13. **DELGADO RAMOS, A.** *Informatización del Sistema Nacional de Salud. Dirección Nacional de Registros Médicos y Estadísticas de Salud. Ministerio de Salud Pública.* . In. La Habana. Cuba., 2006.
14. **INC., S. L.** *Web based vs. Thick Clients in General Aviation.* 2005, [citado 31/03/2005]. [Disponible en: www.s2logic.com/WebGA2.pdf].
15. **BELTRANENA, R. J. S. AND A. R. DÍAZ.** *La informatización de la atención primaria de salud.* 2006, [citado 25-05-2011]. [Disponible en: http://bvs.sld.cu/revistas/mgi/vol22_4_06/mgi12406.htm].
16. **GARCÍA MORALES, J. C.** *Sistema Informático para la Dispensarización en la APS.* [citado 20-04-2011]. [Disponible en: <http://www.informatica2007.sld.cu/Members/jcarlosgm/sistema-informatico-para-ladispensarizacion-en-la-atencion-primaria-de-salud-sidaps/>].
17. **APUS.** *Sistema Informático para la Gestión Médica en la APS,* 2011. [Disponible en: <http://www.sld.cu/instituciones/cedisap/atepri1.htm>]
18. **InfoDiabetes.** *Retrato de Salud de mi Familia / My Family Health Portrait,* 2007. [Disponible en: <http://borealia.wordpress.com/2007/01/11/retrato-de-salud-de-mi-familia-my-family-health-portrait>]
19. **LTDA, I. C.** *Paracelsus Desktop 1.0.* [Disponible en: http://ecuador.acambiode.com/producto/paracelsus-desktop-1-0_114449]
20. **ECURED.** *Lenguaje de programación.* 2010, [citado 2 de junio de 2010]. [Disponible en: http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n].
21. **GARCÍA DE JALÓN, J., RODRÍGUEZ IÑIGO, MINGO JOSÉ IGNACIO, ALFONSO BRAZÁLEZ AITOR IMAZ, LARZABAL AND C. J. ALBERTO, GARCÍA JON.** *Aprenda Java como si estuviera en primero* [online]. [Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España]: 2000 [citado]. [Disponible en: <http://www.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/Java/Java2.pdf>].
22. **MICROSYSTEMS, S.** *Java Code Conventions.* In. California, Estados Unidos de América.

Referencias Bibliográficas

23. *Aplicación en capas*. 2008, [citado 2008 de Marzo de 2010]. [Disponible en: <http://oness.sourceforge.net/proyecto/html/ch03s02.html>].
24. **FIGUEROA, P.** 1997, [citado 5 de febrero de 2010]. [Disponible en: http://agamenon.uniandes.edu.co/~pfigueroa/soo/Magister_Patrones/intropatrones.html#tema2].
25. **HENAO CALDERÓN, D. A.** *SlideShare.Abstract Factory*. 2009, [citado 11 de febrero del 2010]. [Disponible en: <http://www.slideshare.net/dialheca/abstract-factory>].
26. *GNU Proyecto, Free Software Foundation*. [citado 25 de abril de 2011]. [Disponible en: <http://www.gnu.org/philosophy/free-sw.es.html>].
27. *Ídem a la referencia 26*.
28. **CABELLO ALBINO, C.** *El mundo Linux. Definición de Linux*, 2007. [Disponible en: <http://www.elmundolinux.com/definicionlinux.php>.]
29. **AUTORES, C. D.** *Documento de Arquitectura de Software. Facultad 7. CESIM*. In.: Universidad de las Ciencias Informáticas, 2009.
30. **PRESSMAN, R. S.** *Ingeniería de Software, un enfoque práctico*. Edtion ed. La Habana, Cuba.Editorial Félix Varela.
31. **JACOBSON, I., BOOCH, GRADY Y RUMBAUGH, JAMES.** *El proceso unificado de desarrollo de software*. Edtion ed.
32. **ECURED.** *Flujo de trabajo análisis y diseño*. 2010, [citado 2 de junio de 2010]. [Disponible en: http://www.ecured.cu/index.php/Flujo_de_Trabajo_An%C3%A1lisis_y_Dise%C3%B1o].
33. *Ídem a la referencia 30*.
34. *Ídem a la referencia 31*.
35. **WIELENGA, G.** "How to Create a Swing CRUD Application on NetBeans Platform 6.8", 2009. [Disponible en: <http://netbeans.dzone.com/crud-on-netbeans-platform>]
36. *Jasper Report*. [Disponible en: <http://www.todoexpertos.com/categorias/tecnologia-e-internet/respuestas/1182199/jasperreport>]
37. *Hibernate*. [Disponible en: <http://mundogeek.net/archivos/2007/01/27/hibernate/>]

Referencias Bibliográficas

38. **GÓMEZ, Y.** *Repositorio de documentos para el Departamento de Atención Primaria de Salud del Centro de Infomática Médica*, 2010. [Disponible en: https://repositorio.cesim.prod.uci.cu/svn/aps/configuracion/DESKTOP/EXPEDIENTE_DE_PROYECTO/]
39. **RIVERO, J. L. L.** *La utilización del método de evaluación de expertos en la valoración de los resultados de las investigaciones educativas*. In.: UNIVERSIDAD PEDAGÓGICA "BLAS ROCA CALDERÍO" GRANMA, 1998.
40. OÑATE MARTÍNEZ, N., RAMOS MORALES, L. Y DÍAZ ARMESTO, A. *Utilización del Método Delphi en la pronosticación: una experiencia inicial*. In. La Habana. Instituto de Investigaciones Económicas de la Junta Central de Planificación., 1990.
41. **NORMAN C. DALKEY, B. B. Y. S. C.** *The Delphi Method, III: Use of self rating to improve group estimates*. In., vol. vol 1, p. 283-291.
42. *Ídem a la referencia 39.*
43. **VARIOS.** *El método Delphi.* [Disponible en: <http://www.gtici.ssr.upm.es/encuestas/delphi.htm>]

ANEXOS

Anexo 1: Encuesta para determinar el nivel de competencia.

Encuesta No.1 a Expertos																																																													
<p>Nombre: _____</p> <p>Fecha: _____</p> <p>Cargo o Responsabilidad: _____</p> <p>Entidad: _____</p> <p>Nota: <i>El objetivo de esta encuesta es solamente investigativa. El responsable de esta encuesta se compromete a mantener total privacidad de la información recopilada.</i></p> <p>Pregunta 1 Marque con una X, en una escala del 1 al 10 el valor que se corresponde con el grado de conocimiento o información que usted considera que tiene respecto a las Mediciones de Software, y el empleo de métricas para el análisis estadístico. 1 indica que no tiene ningún conocimiento sobre el tema y 10 indica que tiene pleno conocimiento sobre él.</p> <table border="1" style="margin: 10px auto; width: 80%; text-align: center;"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td> </tr> <tr> <td style="height: 20px;"></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> <p>Pregunta 2 Señale con una X el nivel de influencia que ha tenido cada una de las fuentes indicadas en su conocimiento sobre la Mediciones de Software, y el empleo de métricas para el análisis estadístico.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th rowspan="2" style="text-align: center; padding: 5px;">FUENTES DE ARGUMENTACION</th> <th colspan="3" style="text-align: center; padding: 5px;">Grado de influencia de cada una de las fuentes en sus criterios.</th> </tr> <tr> <th style="text-align: center; padding: 5px;">A (alto)</th> <th style="text-align: center; padding: 5px;">M (medio)</th> <th style="text-align: center; padding: 5px;">B (bajo)</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">Análisis teóricos realizados por usted</td> <td style="height: 20px;"></td> <td></td> <td></td> </tr> <tr> <td style="padding: 5px;">Su experiencia obtenida</td> <td></td> <td></td> <td></td> </tr> <tr> <td style="padding: 5px;">Trabajos de autores nacionales</td> <td></td> <td></td> <td></td> </tr> <tr> <td style="padding: 5px;">Trabajos de autores extranjeros</td> <td></td> <td></td> <td></td> </tr> <tr> <td style="padding: 5px;">Su propio conocimiento del estado del problema en el extranjero</td> <td></td> <td></td> <td></td> </tr> <tr> <td style="padding: 5px;">Su intuición</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>											1	2	3	4	5	6	7	8	9	10											FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.			A (alto)	M (medio)	B (bajo)	Análisis teóricos realizados por usted				Su experiencia obtenida				Trabajos de autores nacionales				Trabajos de autores extranjeros				Su propio conocimiento del estado del problema en el extranjero				Su intuición			
1	2	3	4	5	6	7	8	9	10																																																				
FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.																																																												
	A (alto)	M (medio)	B (bajo)																																																										
Análisis teóricos realizados por usted																																																													
Su experiencia obtenida																																																													
Trabajos de autores nacionales																																																													
Trabajos de autores extranjeros																																																													
Su propio conocimiento del estado del problema en el extranjero																																																													
Su intuición																																																													

Anexo 2: Encuesta para determinar el nivel de coincidencia entre la opinión de los expertos.

Encuesta No.2 a Expertos																																
<p>Nombre: Fecha: Cargo o Responsabilidad: Entidad:</p> <p>Nota: El objetivo de esta encuesta es solamente investigativa. El responsable de esta encuesta se compromete a mantener total privacidad de la información recopilada.</p> <p>Pregunta 1 ¿Cuáles considera usted son los principales los problemas a los que se debe enfrentar un desarrollador durante el proceso de implementación de sistemas informáticos de escritorio en el CESIM y que entorpecerían el desarrollo exitoso de los mismos?</p> <p>1. _____ 2. _____ 3. _____ 4. _____ 5. _____ 6. _____ 7. _____</p> <p>Pregunta 2 Evalúe dichas actividades por el nivel de importancia que usted le concierne. (1-10). Donde 1 significa poca importancia y 10 Mucha importancia.</p> <table border="1"><tr><td>Actividades</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>Nivel de importancia</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>											Actividades	1	2	3	4	5	6	7	8	9	10	Nivel de importancia										
Actividades	1	2	3	4	5	6	7	8	9	10																						
Nivel de importancia																																

Anexo 3: Encuesta para evaluar la solución propuesta.

Encuesta No.3 a Expertos
<p>Nota: El objetivo de esta encuesta es solamente investigativa. El responsable de esta encuesta se compromete a mantener total privacidad de la información recopilada.</p> <p>Pregunta 1 Evalúe los elementos implementados o propuestos en la plataforma según los siguientes aspectos.</p> <p>a) El uso de las plantillas CRUD propuestas en el desarrollo de aplicaciones de escritorio de forma más rápida, usted considera que son: <input type="checkbox"/> Innecesarias. <input type="checkbox"/> Necesarias pero no suficientes para desarrollar una aplicación exitosa. <input type="checkbox"/> Necesarias y suficientes para desarrollar una aplicación exitosa. Explique brevemente su opinión si lo considera necesario:</p> <p>b) El uso de hilos y la optimización del framework JasperReport propuesto en el desarrollo de aplicaciones de escritorio ligeras, usted considera que son: <input type="checkbox"/> Innecesarias. <input type="checkbox"/> Necesarias pero no suficientes para desarrollar una aplicación exitosa. <input type="checkbox"/> Necesarias y suficientes para desarrollar una aplicación exitosa. Explique brevemente su opinión si lo considera necesario:</p> <p>c) El uso del componente visual propuesto para el paginado, <i>Jpager</i>, en el desarrollo de aplicaciones de escritorio, usted considera que son: <input type="checkbox"/> Innecesarias. <input type="checkbox"/> Necesarias pero no suficientes para desarrollar una aplicación exitosa. <input type="checkbox"/> Necesarias y suficientes para desarrollar una aplicación exitosa. Explique brevemente su opinión si lo considera necesario:</p> <p>d) El uso del motor de bases de datos HSQLDB y el ORM Hibernate para el desarrollo de aplicaciones de escritorio ligeras, usted considera que son: <input type="checkbox"/> Innecesarias. <input type="checkbox"/> Necesarias pero no suficientes para desarrollar una aplicación exitosa. <input type="checkbox"/> Necesarias y suficientes para desarrollar una aplicación exitosa. Explique brevemente su opinión si lo considera necesario:</p> <p>e) Los mecanismos implementados para la firma digital de documentos XML en el desarrollo de aplicaciones de escritorio ligeras orientadas al sector de la salud, usted considera que son: <input type="checkbox"/> Innecesarias. <input type="checkbox"/> Necesarias pero no suficientes para desarrollar una aplicación exitosa. <input type="checkbox"/> Necesarias y suficientes para desarrollar una aplicación exitosa. Explique brevemente su opinión si lo considera necesario:</p> <p>f) El uso de los mecanismos implementados para la gestión de la seguridad en el desarrollo de aplicaciones de escritorio ligeras, usted considera que son: <input type="checkbox"/> Innecesarias. <input type="checkbox"/> Necesarias pero no suficientes para desarrollar una aplicación exitosa. <input type="checkbox"/> Necesarias y suficientes para desarrollar una aplicación exitosa.</p>

Explique brevemente su opinión si lo considera necesario:

Pregunta 2

Evalúe los siguientes criterios según el nivel de importancia que usted le confiera donde: "1" significa poco valor y "10" significa el máximo valor posible:

Criterio	0	1	2	3	4	5	6	7	8	9	10
Uso de las plantillas CRUD											
Uso de hilos y la optimización del framework JasperReport											
Uso del componente visual propuesto para el paginado, <i>JPager</i> .											
Uso del motor de bases de datos HSQLDB y el ORM Hibernate											
La firma digital de documentos XML											
La gestión de la seguridad											

Pregunta 3

¿Considera usted que la puesta en práctica de la plataforma propuesta puede ser efectivo en el desarrollo de aplicaciones de escritorio ligeras orientadas al sector de la salud?

___ Si ___ No

¿Por qué?

Pregunta 4

Elabore un comentario general sobre el procedimiento que está siendo evaluado que aporte elementos a la mejora del mismo.

GLOSARIO

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

Aplicación o Sistema Informático: Programas con los cuales el usuario final interactúa a través de una interfaz y que realizan tareas útiles para éste.

Área de Salud: Área geográfica a la que presta sus servicios una Unidad de salud que contemple el Programa de Trabajo del Médico y la Enfermera de la Familia. Son las estructuras fundamentales del sistema sanitario, responsables de la gestión unitaria de los centros y establecimientos del servicio de salud de la Comunidad Autónoma en su demarcación territorial y de las prestaciones sanitarias y programas sanitarios por ellos desarrollados.

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

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

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

MINSAP: Ministerio de Salud Pública de Cuba.

Medicina Familiar: es una especialidad médico – quirúrgica, que tiene un cuerpo de conocimientos científicos propios con principios universales que se pueden aplicar de manera particular, siendo una oportunidad única para que el médico desarrolle habilidades y conocimientos en el manejo biopsicosocial del individuo y de su familia, inmersa en una comunidad.

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

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

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

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

Splash: Especie de pantalla de presentación que se muestra usualmente durante la carga de una aplicación con el objetivo de entretener e informar al usuario sobre temas como el copyright y el nombre del autor, para evitar que el usuario piense que el proceso de carga no está llevándose a cabo correctamente o que se haya paralizado por algún motivo.