The background features a decorative graphic consisting of three overlapping circles in shades of blue, arranged in a descending sequence from top-right to bottom-right. Two thin, light blue lines intersect at the top-left corner, forming a large 'V' shape that frames the central text.

Implementación de mejoras a la plataforma para la administración, el procesamiento y la transmisión de la información en el Sistema Integral de Salud.

Sandra Gutiérrez Secades
03/03/2011



Universidad de las Ciencias Informáticas
DIRECCIÓN DE FORMACIÓN POSTGRADUADA

**Implementación de mejoras a la plataforma para la administración, el
procesamiento y la transmisión de la información en el Sistema
Integral de Salud.**

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

Autor: Ing. Sandra Gutiérrez Secades.

Tutor: Dr C José Alejandro Gutiérrez Fernández.



Ciudad de La Habana, Cuba. Marzo de 2011.

DECLARACION JURADA DE AUTORIA

Yo Sandra Gutiérrez Secades, con carné de identidad 82061204415, declaro que soy el autor del resultado que expongo en la presente memoria titulada Implementación de mejoras a la plataforma para la administración, el procesamiento y la transmisión de la información en el Sistema Integral de Salud. Este trabajo fue desarrollado durante el período Septiembre, 2008 – Febrero, 2011.

Declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los 7 días del mes de marzo del año 2011.

DEDICATORIA

A mí madre:

Tú que eres mi luz primera...

RESUMEN

La empresa Softel, perteneciente al Ministerio de la Informática y las Comunicaciones (MIC) y vinculada a la Universidad de las Ciencias Informáticas (UCI) fue designada como gestor técnico y tecnológico para la informatización de la salud cubana.

Uno de los proyectos liderados por la misma fue el desarrollo de El Registro Informatizado de Salud (RIS) como la solución informática integral para la Salud Pública; el cual sentó las bases para la existencia de un sistema sobre una arquitectura basada en componentes y orientada a servicios, desarrollado con tecnologías basadas en Internet y software libre.

El Proyecto SISalud continuó desarrollando un conjunto de componentes distantes geográficamente que se encuentran en constante interacción a través de la Red Telemática de Salud de Cuba, INFOMED y es además la herramienta que permite a los usuarios autorizados combinar la información para apoyar la toma de decisiones en los diferentes niveles de dirección, docencia, investigación y gestión en salud.

Como parte de la necesidad de mejorar los procesos de producción y servicios de software en la entidad se analizan un conjunto de aspectos importantes para un mejor funcionamiento de la aplicación, así como la implementación de soluciones a algunas de las problemáticas planteadas. Lo anterior constituye el tema de la presente investigación.

Para ello se realizó un análisis de las herramientas y técnicas actuales utilizadas en el proceso de desarrollo de aplicaciones web y se escogieron algunas de las adecuadas para aplicar a la plataforma SiSalud, de manera que se disminuyeran los tiempos de desarrollo del proyecto, los tiempos de respuesta de la aplicación y se facilitara la integración de aplicaciones externas. Las mejoras implementadas al proceso de desarrollo del proyecto, se constatan por los resultados satisfactorios arrojados por el mismo.

El trabajo realizado resuelve problemas internos presentados en el trabajo del equipo de desarrollo, así como aspectos que inciden directamente en los clientes de la aplicación, lo cual afecta la satisfacción del mismo en el servicio brindado.

Palabras claves: Componentes, Tiempos de respuesta, Tiempos de desarrollo, Integración, Mejora.

Índice

INTRODUCCION	8
CAPITULO I. Análisis de las herramientas y técnicas utilizadas en el desarrollo de aplicaciones web.....	12
1. Arquitectura del Sistema	17
2. Herramientas a integrar en el desarrollo del Sistema.	21
2.1 Sistema de control de versiones.....	22
2.2 Virtualización.....	25
2.3 Aplicaciones de Gestión. Tecnologías a aplicar.....	28
2.3.1 Ajax.....	29
2.3.2 Reglas del lenguaje. Coincidencias fonéticas	31
2.3.3 Comunicación entre componentes	34
2.3.4 Reportes en las aplicaciones informáticas	35
2.3.5 Integración de aplicaciones de terceros. Servicios Web.	35
CAPITULO II. Implementación de mejoras a SiSalud.....	45
1. Proceso de desarrollo y liberaciones de SiSalud.	45
1.1. Control de versiones y Máquinas Virtuales ACE.....	45
2. Tiempos de respuesta y optimización de procesos.	48
2.1 Integración de Ajax	48
2.2 Utilización de Soundex en la inserción de ciudadanos	52
2.3 Reportes con paginado horizontal.....	53
2.4 Componente lista de trabajo.....	55
2.5 Estandarización de Servicios Web.....	62
CONCLUSIONES.....	64

RECOMENDACIONES	65
REFERENCIAS BIBLIOGRAFICAS.....	66
BIBLIOGRAFIA.....	68

INTRODUCCION

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

El Sistema Nacional de Salud (SNS) se estructura en tres niveles que se corresponden con la estructura político-administrativa del país. El nivel nacional, representado por el Ministerio de Salud Pública que es el órgano rector con funciones metodológicas, normativas y de coordinación y control, al cual se le subordinan directamente los centros universitarios, institutos de investigaciones, centros hospitalarios de asistencia médica altamente especializados, centros de distribución y comercializadoras de suministros y tecnologías médicas, así como otros centros y entidades nacionales destinados a actividades técnicas y de apoyo.

Los otros dos niveles están representados por las direcciones provinciales y municipales de salud que agrupan a las instituciones de salud en su respectivo nivel y que, al igual que en el nivel central, se subordinan desde el punto de vista administrativo a las estructuras de Gobierno en los distintos niveles organizativos, representando sus intereses ante ellos y dando respuesta a las demandas y necesidades de la población.

Las áreas de salud y las unidades que prestan servicios en la atención primaria se subordinan a los Consejos Populares, quienes se encargan de resolver los problemas más específicos de la población de su radio de acción. Además la labor de los Equipos Básicos de Salud (EBS) se subordina a los Delegados de Circunscripción pertenecientes a los Consejos Populares.

El Programa del Médico y la Enfermera de la Familia, se ratifica como el eje del actual desarrollo estratégico, orientándose el resto de las estrategias en función del mismo. Este modelo de atención es la mayor fortaleza y potencialidad que tiene el Sistema Nacional de Salud (SNS). (Luna, 2008)

Con más de 20 años de experiencia en este programa se comienzan a experimentar cambios para la atención primaria, de esta forma, servicios que eran exclusivos de hospitales son

abiertos en instituciones de la atención primaria; surgiendo así el modelo de policlínico con nuevas funciones, acercando los servicios a la población.

La Informatización del Sistema Nacional de Salud tiene como objetivo acercar eficientemente y con calidad la prestación de los servicios de salud a la población. En las líneas generales del Desarrollo Informático en la Salud se encuentran: la Atención Primaria, Secundaria y Terciaria, el Sistema Integrado de Urgencia Médica, Vigilancia de Salud, Telemedicina, Medicamentos y Fármacos, Epidemiología, Biblioteca y Universidad Virtual, Docencia Médica, entre otros.

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

La Informatización de la Atención Primaria de Salud debe permitir la gestión médica, interacción con los consultorios del Médico de la Familia, obtención de estadísticas, apoyo en la logística de los nuevos servicios y el flujo de información hacia los diferentes niveles de toma de decisiones.

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

Solución integral propuesta para la Informatización del SNS

La Informatización de la Salud Cubana no ha ofrecido un mecanismo único de integración de los sistemas de información desarrollados. Generalmente se presentaban como componentes aislados, lo cual traía consigo la duplicación de información y la falta de integridad de la misma.

El Registro Informatizado de Salud (RIS), sentó las bases para la existencia de un sistema formado por componentes, desarrollados con un nivel de cohesión y acoplamiento que le permitieron ser capaces de interactuar entre ellos y reutilizar la información gestionada por cada componente. El RIS, sobre una arquitectura basada en componentes y orientada a servicios, fue desarrollado con tecnologías basadas en Internet (XML, Web Services), e implementado con software libre (PHP, MySql, Linux) por un grupo de especialistas del sector y

de varias instituciones del Ministerio de la Informática y las Comunicaciones (MIC), tarea encomendada a la Empresa SOFTEL.

Por definición el RIS surge como la solución informática integral para la Salud Pública, acorde con los objetivos de la informatización de la sociedad cubana. Constituido por un conjunto de aplicaciones independientes (módulos del sistema) que se interconectan según las necesidades del flujo de información. Es además la herramienta que permite a los usuarios autorizados combinar la información de los diferentes módulos que lo componen, para obtener una información integral en tiempo real para la toma de decisiones en los diferentes niveles de dirección, la docencia, investigación y la gestión en salud. (Delgado Ramos, 2005)

El Proyecto SISalud heredó las características del RIS y continuó desarrollando un conjunto de módulos o componentes distantes geográficamente que se encuentran en constante interacción a través de la Red Telemática de Salud de Cuba, INFOMED. (Infomed, 2010)

Para lograr la Informatización en este sector se pretende que todos los componentes que forman parte del Sistema Integral de Salud (SiSalud), se agrupen según necesidades reales del MINSAP.

Con ello se hizo preciso analizar un conjunto de aspectos claves en el correcto funcionamiento de la aplicación. Tal es el caso de la agrupación de los diferentes componentes según su función, los tiempos empleados para búsquedas extensas, la consolidación de información no redundante, así como la correcta integración de aplicaciones externas, desarrolladas en muchos casos en diferentes plataformas.

Se deriva por ende el siguiente **problema científico**: ¿Qué mejoras implementar en el Sistema Integral de Salud (SiSalud) para brindar flexibilidad en el desarrollo, posibilitar la reutilización del código y disminuir los tiempos de respuesta al usuario final?

Esta investigación tiene como **objeto de estudio** la plataforma para la administración, el procesamiento y la transmisión de la información en el Sistema Integral de Salud, centrando su **campo de acción** en la aplicación óptima de las técnicas y herramientas actuales utilizadas para el desarrollo de aplicaciones web.

De esta manera el **objetivo general** de este trabajo es lograr una aplicación confiable, con componentes reutilizables; que favorezca la toma de decisiones. Además de implementar mejoras en la aplicación que disminuyan los tiempos de desarrollo del proyecto y los tiempos de respuestas de la aplicación.

Se tienen como **objetivos específicos**:

1. Elegir entre las técnicas y herramientas actuales utilizadas en el desarrollo de aplicaciones web, cuales son idóneas para utilizar en el proyecto.
2. Disminuir los tiempos de respuestas de la aplicación.
3. Flexibilizar el entorno de desarrollo para disminuir los tiempos de desarrollo del proyecto.
4. Permitir la integración de aplicaciones externas.

Esta investigación tiene una estructura de capítulos. Incluye además Introducción, Conclusiones, Recomendaciones, Referencias Bibliográficas y Bibliografía Consultada.

El capítulo I Análisis de las herramientas y técnicas utilizadas en el desarrollo de aplicaciones web, contiene una síntesis del estudio realizado, con el objetivo de definir cuáles pueden aplicarse a la plataforma para la administración, el procesamiento y la transmisión de la información en el Sistema Integral de Salud, de manera que se disminuyan los tiempos de desarrollo del proyecto, los tiempos de respuesta de la aplicación y se facilite la integración de aplicaciones externas. El capítulo II Implementación de mejoras a SiSalud se refiere al proceso de implantación y pruebas de las soluciones, los procedimientos utilizados para ponerlas en práctica, así como los resultados obtenidos de la misma.

CAPITULO I. Análisis de las herramientas y técnicas utilizadas en el desarrollo de aplicaciones web.

El Sistema Nacional de Salud (SNS) se estructura en tres niveles que se corresponden con la estructura político-administrativa del país. El nivel nacional está representado por el Ministerio de Salud Pública que es el órgano rector con funciones metodológicas, normativas, de coordinación y control, al cual se le subordinan directamente los centros universitarios, institutos de investigaciones, centros hospitalarios de asistencia médica altamente especializados, centros de distribución y comercializadoras de suministros y tecnologías médicas, así como otros centros y entidades nacionales destinados a actividades técnicas y de apoyo.

Los otros dos niveles están representados por las Direcciones Provinciales y Municipales de Salud que agrupan a las instituciones de salud en su respectivo nivel y que, al igual que en el nivel central, se subordinan al Consejo de la administración Provincial y Municipal respectivamente.

Las áreas de salud y las unidades de salud que prestan servicios en la atención primaria se subordinan a los Consejos Populares, quienes se encargan de resolver los problemas más específicos de la población de su radio de acción. Además la labor de los Equipos Básicos de Salud (EBS) se subordina a los Delegados de Circunscripción pertenecientes a los Consejos Populares. (Luna, 2008)

El MINSAP, en el año 2003, decidió comenzar el desarrollo de un sistema que permitiera integrar la información de diferentes áreas con el principal objetivo de elevar la calidad en la prestación de los servicios de salud. Se definió que para lograr la informatización en el sector, todos los sistemas informáticos para la salud debían formar parte del Programa General de Informatización del SNS, surgiendo la necesidad de desarrollar una plataforma que se convirtió en el Sistema de Información para la Salud (SISalud).

SISalud como solución integral significa: la articulación de un nuevo paradigma en la prestación de servicios de salud, que contemple: la Atención Primaria, Secundaria y Terciaria, el Sistema

Integrado de Urgencia Médica, la Vigilancia en Salud, Docencia Médica, Telemedicina, Medicamentos y Fármacos, Epidemiología, entre otros.

Esta nueva estrategia del MINSAP estaba alineada al proceso de Informatización de la Sociedad Cubana, poniendo como centro de la misma al paciente y utilizando para su construcción las Tecnologías de la Información y las Comunicaciones (TIC) para el desarrollo de aplicaciones y componentes con un alto nivel de integración e interoperabilidad que permitieran obtener la información oportuna, consistente y confiable para los procesos de administración de salud y la toma de decisiones en los diferentes niveles de dirección y atención médica.

Durante los últimos 25 años un grupo de instituciones cubanas han desarrollado sistemas encaminados a lograr determinados niveles de informatización de la salud. Estas soluciones carecían de integración y de una definición generalizable, además de no existir los recursos tecnológicos necesarios para su implementación en el SNS. A partir de 1997 se implementa una primera estrategia de informatización con la finalidad de coordinar esfuerzos para el desarrollo de este proceso. (Vidal, 2002).

Una solución armónica a los problemas claves identificados por el sector y temas como la infraestructura tecnológica necesaria para el desarrollo y la generalización, así como los servicios de mantenimiento, actualización y en general la sostenibilidad, no había estado garantizada.

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

En el año 2003 el Registro Informatizado de Salud (RIS), sentó las bases para la existencia de un sistema formado por componentes, desarrollados con un nivel de cohesión y acoplamiento que permitiera interactuar entre ellos y de esta forma reutilizar la información gestionada por cada componente. El RIS, sobre una arquitectura basada en componentes y orientada a servicios, fue desarrollado con tecnologías basadas en Internet (XML, Web Services), e implementado con software libre (PHP, MySQL, Linux) por un grupo de especialistas del sector y de

varias instituciones del Ministerio de la Informática y las Comunicaciones (MIC), tarea encomendada a la Empresa SOFTEL.

Por definición el RIS surge como la solución informática integral para la Salud Pública, acorde con los objetivos de la informatización de la sociedad cubana. Constituido por un conjunto de aplicaciones independientes (módulos del sistema) que se interconectan según las necesidades del flujo de información. Es además la herramienta que permite a los usuarios autorizados combinar la información de los diferentes módulos que lo componen, para obtener una información integral en tiempo real para la toma de decisiones en los diferentes niveles de dirección, la docencia, investigación y la gestión en salud. (Delgado Ramos, 2005)

El Proyecto SISalud heredó las características del RIS y continuó desarrollando un conjunto de módulos o componentes distantes geográficamente que se encuentran en constante interacción a través de la Red Telemática de Salud de Cuba, INFOMED. (Infomed, 2010)

Los módulos desarrollados se integraron para garantizar la consistencia, no duplicidad, oportunidad y precisión de la información. Por otro lado se garantizaba que cada sistema creado no se convirtiera en una isla de información con utilidad y beneficios muy limitados.

Se llevó a cabo una revisión de las aplicaciones informáticas existentes en el país. Se estudiaron los principios y estrategias del MINSAP relacionadas con el proceso de informatización en el sector, se utilizó la entrevista grupal con los principales funcionarios del SNS según las especialidades y niveles de atención como uno de los métodos fundamentales en el proceso de levantamiento de requisitos, análisis y diseño de cada aplicación, contando con un grupo de Expertos Funcionales del Ministerio de Salud Pública que permiten la definición de las prioridades en cada etapa.

La propuesta del Portal SISalud contaba con la integración de los datos generados en los distintos niveles de salud donde puede ser atendido un paciente, optimizando de esta manera la calidad asistencial ofrecida a la sociedad, facilitando las funciones de los trabajadores de la salud, y colaborando con la actividad administrativa, asistencial, docente y de investigación. Es esto precisamente lo que permite hablar de informatización de la Salud Pública; no de proyectos aislados. (Delgado, 2006)

Teniendo en cuenta que los proyectos que se acometen están relacionados con la actividad de la atención primaria, hospitalaria, de las redes de especialidades y con sistemas que permitan tener centralmente la información útil para la planificación y toma de decisiones generada por cada uno de los anteriores, se propuso que el Sistema de Información para la Salud (SISalud) estuviese formado por los módulos que formaban parte del RIS, los módulos que responden a las funcionalidades de los diferentes niveles de atención médica y por los módulos que respondieran al resto de las funcionalidades administrativas del SNS.(7) La estructura del SISalud consta de Módulos que pueden clasificarse por áreas como se aprecia en la Figura 1.



Figura 1: Estructura del Sistema de Información para la Salud

Por definición SISalud se considera un Portal de Aplicaciones para el SNS que conforma su estructura a partir de los derechos del Usuario que se autentica, dándole solamente la posibilidad de acceso a los Módulos o Aplicaciones que este requiera y mostrándole además los Avisos que le corresponden.

Se define una estructura que contemple:

- Administración del Sistema.
- Registros Básicos y Codificadores.
- Administración de Salud.
- Atención Médica (Atención Primaria, Hospitalaria y Especializada).
- Ayuda a la Decisión.

La seguridad en el portal comienza por la identificación de los usuarios que pueden acceder al sistema, credenciales con las que el administrador puede, en primera instancia, controlar la entrada y, una vez franqueado el paso, fiscalizar su acceso y privilegios a los módulos disponibles, garantizando la trazabilidad de todos los procesos.

Administración del Sistema: Garantiza la seguridad en el acceso a determinado nivel de profundidad y derechos de edición por medio de la Autenticación, Autorización y Auditoría y ofrece prestaciones según tipo de usuario y nivel de gestión por medio de contraseñas y certificados digitales.

Registros Básicos y Codificadores: Son módulos que registran datos que son de necesidad común a otros módulos y favorecen la estandarización de la información. La información es administrada y gestionada a nivel nacional o central.

Administración de Salud: Son los módulos que responden a otros sistemas de gestión e información de salud que no forman parte directamente de los niveles de atención médica. Por ejemplo: Medicamentos y Fármacos, Balance Material, Economía, entre otros.

Módulos de Atención Médica: Gestionan la información y generan reportes referentes a la actividad asistencial en los diferentes niveles de atención médica.

Se puede acceder a un conjunto de módulos desde cualquier nivel de atención médica o institución de salud, permitiendo la continuidad en el seguimiento del paciente, o sea son módulos que no son del nivel de atención primaria propiamente, pero procesan y generan información que se obtiene de este nivel comunitario y a la vez lo retroalimenta.

De igual forma se accederá a los módulos que pertenecen al nivel de atención secundario y terciario como el Registro de Autopsias, los módulos del Sistema Gestión Hospitalaria, Registro Centralizado de Donantes, entre otros.

Ayuda a la Decisión: Integra la información de los módulos o componentes para garantizar el conocimiento oportuno, consistente y confiable para la toma de decisiones en cada nivel de dirección del SNS y el mejoramiento continuo de la calidad de la atención médica y la administración de salud.

Los usuarios de SISalud pueden ser de tres tipos: los Administradores que son usuarios con posibilidad de Gestionar (crear, modificar o eliminar) en los diferentes niveles (Nacional, Provincial, Municipal, Unidad de Salud) a otros usuarios en el sistema y chequear las trazas de los usuarios. Existirá un Administrador por cada nivel. Los Editores están encargados de

agregar o actualizar información en los módulos y los Visualizadores tienen sólo la posibilidad de ver la información de los módulos en dependencia de su nivel.

En la página de inicio de SISalud, se dispone de un sistema de avisos que permitirá establecer mecanismos de control entre los usuarios profesionales y técnicos de la salud y para las diferentes instancias de dirección.

El módulo de Avisos se presenta como una propuesta para los directivos que deben velar por la calidad de los servicios en salud. Es la comunicación que realizan los componentes entre sí y que debe llegar al usuario de cada módulo, para acometer las acciones que se deriven de cada aviso. Cuando se logre la integración paulatina de todas las aplicaciones que tributan a la informatización del SNS, estos avisos los pueden ejecutar los módulos de forma automatizada.

Se muestran al usuario desde que se autentifica en el sistema; es decir, los avisos son personales y no modulares; esto evita que el usuario tenga que entrar a diferentes módulos para conocer sus avisos. Se le muestran ordenados descendientemente por fecha de envío.

La validación de los módulos y componentes de las primeras versiones de SISalud se realiza en instituciones seleccionadas del Sistema Nacional de Salud, garantizando de esta manera la consistencia, robustez y calidad de los productos en el momento de su generalización, proceso que tiene lugar de manera progresiva y controlada garantizando desde el inicio el desarrollo y capacitación de los recursos humanos asociados a los mismos y la asistencia técnica de manera rápida y oportuna, aspectos estos que garantizan la sostenibilidad y el éxito en el proceso de implantación a escala nacional.

1. Arquitectura del Sistema

Este sistema está concebido completamente sobre arquitectura basada en componentes y orientada a servicios, empleando servicios web XML, específicamente SOAP. Utiliza PLASER (PLAtaforma de SERvicios) para facilitar la programación y homogeneidad de los componentes. PLASER está integrado fundamentalmente por varias clases en PHP, es una librería que puede o no ser usada para que un componente se integre al RIS pero de no ser usada la seguridad corre por parte del programador. PLASER puede verse como una especie de *framework* sobre el cual se instalan los módulos o componentes del sistema. (Pompa, 2006)

En esta versión PLASER sólo soporta como llamada RPC el protocolo SOAP.

PLASER en su estructura encapsula varios componentes:

- XSL: Extensión para transformaciones de ficheros XML.
- Pear SOAP: Biblioteca para la comunicación SOAP.
- DBX: Extensión para la abstracción del acceso a datos.

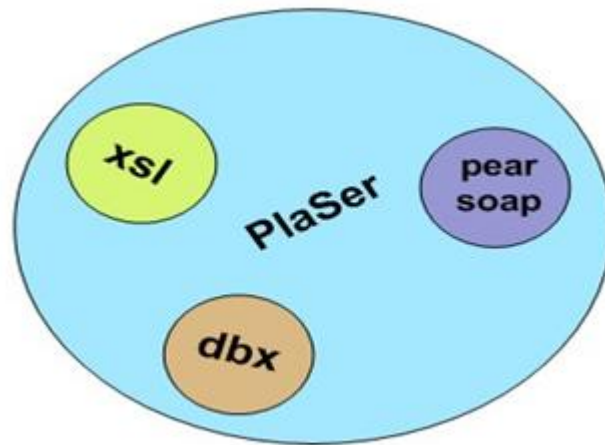


Figura 2: Encapsulación de componentes de PLASER.

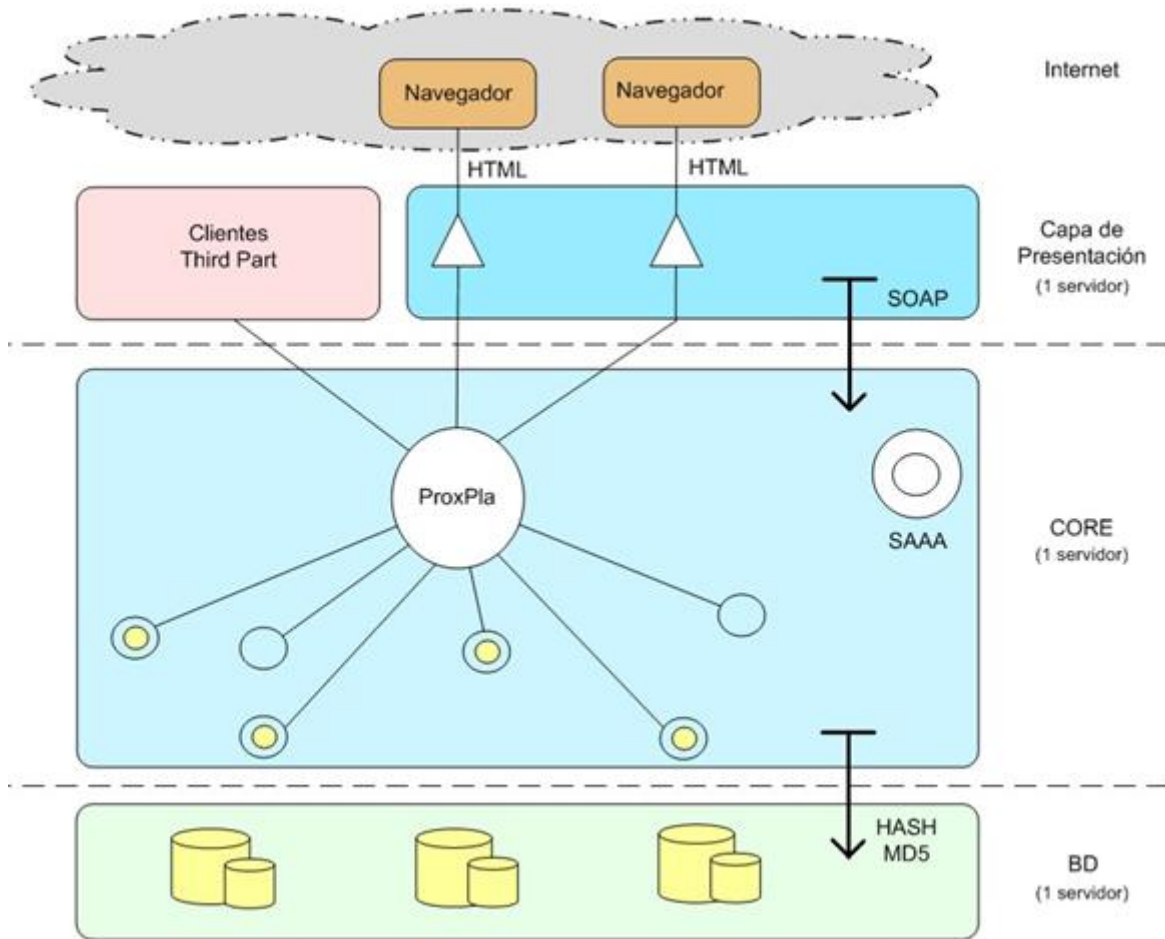


Figura 3. Topología de PLASER.

La figura muestra como cada componente del **Core** es un sitio virtual.

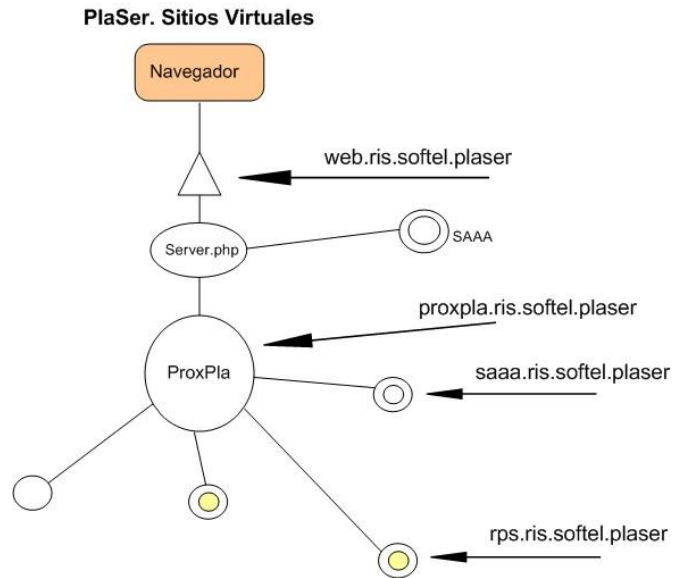


Figura 4. Sitios virtuales por componentes.

En el desarrollo de la capa de presentación se utilizó el patrón de diseño Fachada (Facade) con el objetivo de lograr una interfaz sencilla, unificada y simple; disminuir el intercambio de mensajes entre componentes, además de minimizar la comunicación y las dependencias entre subsistemas, así como proveer un alto grado de desacoplamiento. También en esta capa se utilizan rutinas en lenguaje JavaScript que permiten modificar la página actual obteniendo mayor dinamismo y mejorando la presentación y la interactividad.

En fachada se invocan los métodos de la capa de negocio desde la capa de presentación, porque realmente no se toma ninguna decisión a este nivel y por tanto no hay negocio, simplemente se necesita invocar varios métodos para, por ejemplo, obtener el xml que se necesita para crear un formulario; pero la invocación puede ser en cualquier orden y los parámetros de salida de uno no son entrada de ningún otro método de negocio, es decir, no estamos en presencia de un agente.

Funcionamiento:

Cuando un cliente web (navegador o web browser) o una aplicación específica (Cliente ThirdPart) realiza una petición de acceso a alguno de los módulos instalados sobre PLASER en el RIS, dicha petición es atendida por plaser.php; una vez que este haya verificado todo lo

concerniente a la seguridad con el componente SAAA, entonces se ejecuta el código correspondiente a la aplicación solicitada.

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

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

Es una arquitectura de componentes independientes que puede ser interpretada por capas, de la forma siguiente:

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

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

2. Herramientas a integrar en el desarrollo del Sistema.

SiSalud, desde sus inicios, posee deficiencias claves en su desarrollo. Como eje central de la informatización del Sistema Nacional de Salud, es necesario que posea características deseadas en todo momento, para cualquier aplicación informática, pero necesarias en gran medida para una de esta envergadura.

Un elemento clave que no puede faltar en el desarrollo del proyecto, es una efectiva gestión del proyecto. En ocasiones no es necesario implementar en un desarrollo, todos los aspectos que propone una metodología en específico, pero sí que los que puedan aplicarse reporten los mayores beneficios para el producto.

Es importante una adecuada gestión de cambios y configuración, así como la existencia de un entorno de trabajo cómodo, flexible y eficiente para los desarrolladores.

La optimización de la aplicación requiere el acceso rápido a la información así como que la gestión de la misma sea confiable.

Las aplicaciones que se integran a SiSalud, deben hacerlo siguiendo reglas definidas por el Minsap, por lo que además es imprescindible brindarle la información a las mismas de modo que puedan utilizarla sin importar la plataforma en la que se desarrolla.

2.1 Sistema de control de versiones

Dentro de las mejoras implementadas al desarrollo de SiSalud, se destaca la utilización de un sistema de control de versiones para ser utilizado por los desarrolladores, con el fin de mantener el código fuente de la aplicación disponible en un servidor, actualizado, con posibilidades de restaurar la aplicación a un estado anterior, así como mejorar el proceso de liberación del producto final, a la Dirección de Servicios.

Un sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados.

Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local. Esto permite compartir los cambios sobre un mismo conjunto de ficheros. El repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de ser necesario.

Para tratar el tema de que más de un usuario, intenten acceder al mismo fichero para ser modificado, se realizan dos actividades fundamentales:

Bloqueos: el usuario bloquea el fichero durante su edición, evitando el acceso concurrente de otros usuarios.

Fusión de cambios (*Merge*): los ficheros se acceden concurrentemente. Los cambios realizados sobre un mismo fichero son fusionados inteligentemente por el sistema. El intento de fusión de cambios incompatibles, debe solucionarse manualmente.

Subversion sobre CVS

Entre los sistemas de control de versiones puede mencionarse el CVS (Concurrent Versions System). CVS tuvo el mérito de ser el primer sistema usado por el movimiento de código abierto para que los programadores colaboraran remotamente mediante el envío de parches. Es de utilización gratuita, código abierto, y emplea fusión de cambios.

Subversion se creó para igualar y mejorar la funcionalidad de CVS, preservando su filosofía de desarrollo, que comenzó en el año 2000. La versión 1.0 fue publicada en febrero del 2004, empleando licencia Apache/BSD.

Subversion es relativamente un nuevo sistema de control de versiones, diseñado para ser el sucesor de CVS. El mismo es potente, muy usado y flexible. (Collins, 2010)

Dentro de las ventajas pueden listarse las siguientes:

- Actualiza ficheros modificados. El cliente recorre el código y sincroniza la copia local con el repositorio.
- Copias de seguridad centralizadas. Solo el administrador debe preocuparse de realizar copias de seguridad en el repositorio. Esto se automatiza fácilmente con una tarea cron o similares.
- Historial de cambios. El repositorio guarda registro de todos los cambios realizados. Es posible recuperar cualquiera de las versiones anteriores de cualquier fichero. Si alguien borra todos los ficheros, puede volverse atrás y recuperar el contenido.
- Acceso remoto. Es posible acceder remotamente al repositorio. No es necesario que el equipo esté dentro de la misma LAN.
- Seguridad. Es posible otorgar diferentes permisos sobre diferentes ramas del proyecto. Por ejemplo, estableciendo permiso universal de lectura, y permiso de escritura solo a ciertos usuarios.

Como limitantes del CVS pueden mencionarse las siguientes:

No registra cambios en la estructura de directorios: no es posible mover, renombrar, ni copiar. Estas operaciones se consiguen eliminando y añadiendo, pero de esta manera se pierde el historial de cambios.

Es necesario interrumpir el acceso al repositorio para crear copias de seguridad.

No permite "conjuntos de cambios". Cuando un desarrollador sube un conjunto de cambios, se van subiendo uno a uno, quizás al mismo tiempo que otro desarrollador hace lo mismo. Al no ser una operación atómica, no se puede asegurar que el estado del repositorio tras un *commit*, sea el mismo que el estado que se probó en la copia de trabajo, y por tanto, el proyecto puede

estar en un estado que nadie ha probado. Además, deshacer un conjunto de cambios requiere recorrer el repositorio completo comparando las fechas.

Almacena ficheros binarios (no sus diferencias entre versiones). Esto consume espacio en disco y ancho de banda.

No usa la red eficientemente. Las diferencias entre versiones solo se envían desde el servidor al cliente, cuando el cliente sube sus cambios envía ficheros en su totalidad.

El código fuente es difícil de mantener. CVS comenzó como un conjunto de scripts shell que usaban RCS e implementaban algoritmos desarrollados entre los años 60-80. El resultado actual es producto de sucesiones de parches, y no tiene un diseño fácil de entender o mejorar. Esto dificulta su evolución. La idea de crear un nuevo CVS desde cero, surgió en la propia compañía que ofrecía soporte comercial para el CVS.

Ventajas de Subversion

Registra cambios en la estructura de directorios (permite mover y renombrar sin perder el historial). Subversion utiliza un sistema virtual de ficheros versionado sobre una base de datos.

El uso de la base de datos Berkeley permite aislamiento, atomicidad, recuperación de datos, integridad, salvos y concurrencia sin necesidad de usar ficheros de bloqueo.

Las trazas de la base de datos pueden ocupar un espacio considerable, pero existen herramientas para eliminarlos (svnadmin).

Las subidas de código al servidor son atómicas, se realizan todas o ninguno. Las transacciones atómicas permiten identificar conjuntos de cambios. Cuando un desarrollador sube un conjunto de ficheros lo hace en una transacción atómica, de modo que todos los ficheros se etiquetan con un número de revisión en el repositorio. La atomicidad también impide que el repositorio quede inestable porque la red cae durante la subida de cambios.

El servidor y cliente intercambian diferencias entre versiones. Al enviar una nueva versión nunca es necesario transmitir ficheros completos.

Pueden añadirse propiedades arbitrarias (pares de clave y valor) a ficheros y directorios.

Interoperabilidad con WebDAV. Es posible acceder al repositorio con cualquier software que soporte dicho protocolo.

Apache + SSL puede usarse con firewalls y proxys.

MIME types y detección automática de ficheros binarios.

Permite operar directamente sobre el repositorio, sin copia local.

Permite salvas.

Se basa en APIs C (interfaz de programación de aplicaciones) bien definidas y documentadas.

Usa la biblioteca Apache Portable Runtime, que permite portar la capa de red a varios sistemas operativos.

El cliente es una pequeña aplicación que usa una biblioteca de alto nivel.

Versiona todos los ficheros guardando comprimidas sus diferencias.

No es necesario duplicar el código en el repositorio para crear ramas. Subversion usa copia perezosa, solo se crea un nuevo fichero cuando es modificado. Mientras tanto, el fichero de la nueva rama, está implementado como un enlace al fichero original.

No es necesario conexión a red para ciertas operaciones, como por ejemplo: status, diff, revert. Esto se debe a que la copia local contiene una copia del fichero original presente en el repositorio. Este comportamiento ahorra ancho de banda con el costo de mayor utilización de espacio en disco.

2.2 Virtualización

El proceso de producción en la empresa, en gran medida, se realiza utilizando un servidor de máquinas virtuales ya sea el despliegue o a veces incluso para el desarrollo. Estas máquinas poseen características muy similares al entorno donde se despliega la aplicación para su explotación, entornos a veces imposible de mantener en una computadora local de un programador, pero sí deseables para la consolidación de no conformidades, o la detección de errores al cambiar de plataforma, entre otros.

Una máquina virtual es un contenedor de software aislado que puede ejecutar sus propios sistemas operativos y aplicaciones como si fuera una computadora física. Una máquina virtual se comporta exactamente como una computadora física ya que contiene su propia CPU, memoria RAM, disco duro y tarjetas de interfaz de red virtuales.

Está compuesta totalmente por software y no contiene ningún tipo de componente de hardware. Por lo tanto, las máquinas virtuales ofrecen una cantidad de ventajas únicas sobre el hardware físico.

Las máquinas virtuales se pueden clasificar en dos grandes categorías según su funcionalidad y su grado de equivalencia a una verdadera máquina.

Las máquinas virtuales de sistema, también llamadas máquinas virtuales de hardware, permiten a la máquina física subyacente multiplexarse entre varias máquinas virtuales, cada una ejecutando su propio sistema operativo. A la capa de software que permite la virtualización se la llama monitor o hypervisor. Un monitor de máquina virtual puede ejecutarse o bien directamente sobre el hardware o bien sobre un sistema operativo.

Aplicaciones de las máquinas virtuales de sistema

- Varios sistemas operativos distintos pueden coexistir sobre la misma computadora, en sólido aislamiento el uno del otro, por ejemplo para probar un sistema operativo nuevo sin necesidad de instalarlo directamente.
- La máquina virtual puede proporcionar una arquitectura de instrucciones (ISA) que sea algo distinta de la verdadera máquina. Es decir, se puede simular hardware.
- Varias máquinas virtuales (cada una con su propio sistema operativo), pueden ser utilizadas para consolidar servidores. Esto permite que servicios que normalmente se tengan que ejecutar en computadoras distintas para evitar interferencias, se puedan ejecutar en la misma máquina de manera completamente aislada y compartiendo los recursos de una única computadora. La consolidación de servidores a menudo contribuye a reducir el coste total de las instalaciones necesarias para mantener los servicios, dado que permiten ahorrar en hardware.

- La virtualización es una opción hoy día, para las máquinas que están siendo "sub-utilizadas". Al virtualizar, la necesidad de nuevas máquinas en una ya existente permite un ahorro considerable de los costos asociados.[18]

Una máquina virtual de proceso, se ejecuta como un proceso normal dentro de un sistema operativo y soporta un solo proceso. La máquina se inicia automáticamente cuando se lanza el proceso que se desea ejecutar y se detiene para cuando éste finaliza. Su objetivo es el de proporcionar un entorno de ejecución independiente de la plataforma de hardware y del sistema operativo, que oculte los detalles de la plataforma subyacente y permita que un programa se ejecute siempre de la misma forma sobre cualquier plataforma.

Muchísimas organizaciones utilizan soluciones de VMware para crear servidores e infraestructuras de escritorio para mejorar la disponibilidad, seguridad y capacidad de gestión de las aplicaciones de misión crítica.

El producto de software VMware es capaz de virtualizar el hardware, de modo que varios sistemas operativos puedan compartirlo.

En general, las máquinas virtuales de VMware cuentan con cuatro características claves que benefician al usuario:

- **Compatibilidad:** las máquinas virtuales son compatibles con todos los ordenadores x86 estándar.
- **Aislamiento:** las máquinas virtuales están aisladas unas de otras, como si estuvieran separadas físicamente.
- **Encapsulamiento:** las máquinas virtuales encapsulan un entorno informático completo.
- **Independencia del hardware:** las máquinas virtuales se ejecutan de forma independiente del hardware subyacente.

El VMware Workstation 6.0 posee capacidades de creación de ACE, lo que permite desarrollar paquetes e implementar máquinas virtuales seguras (llamadas ACE), con lo cual puede realizarse encriptación, acceder a redes y controlar contraseñas en dispositivos de medios portátiles.

2.3 Aplicaciones de Gestión. Tecnologías a aplicar.

Una aplicación de gestión es aquella que se diseña para sustituir uno o varios procedimientos, tanto comerciales como administrativos, que habitualmente realiza una persona en una empresa o institución de forma presencial, por una serie de pantallas en un ordenador, que permitan realizar al cliente los mismos procedimientos de forma no presencial.

En particular, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportados por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de plugins tales como JavaScript, Java, Flash, entre otros, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa y un motor capaz de usar alguna tecnología web dinámica (ejemplo: PHP, Java Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python (programming language) o Ruby onRails) constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa.

El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.

Las interfaces web tienen ciertas limitaciones en las funcionalidades que se ofrecen al usuario. Hay funcionalidades comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar y soltar, que no están soportadas por las tecnologías web estándar. Los desarrolladores web generalmente utilizan lenguajes interpretados (scripts) en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios). Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con las

tecnologías en el lado del servidor. Un ejemplo de esto, AJAX, que es una técnica de desarrollo web que utiliza una combinación de varias tecnologías.

2.3.1 Ajax

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005.

Donde se define como:

“Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.” (Eguíluz, 2005)

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las anteriores.

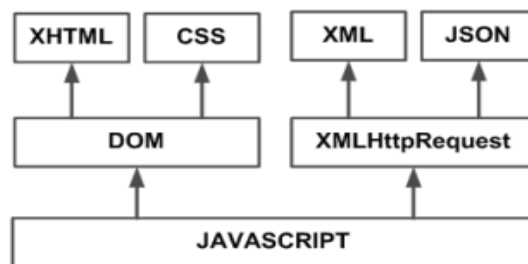


Figura 5. Tecnologías agrupadas bajo el concepto de AJAX

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas las tecnologías anteriores.

En las aplicaciones web tradicionales, las acciones del usuario en la página desencadenan continuas solicitudes al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

En el siguiente esquema, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX:

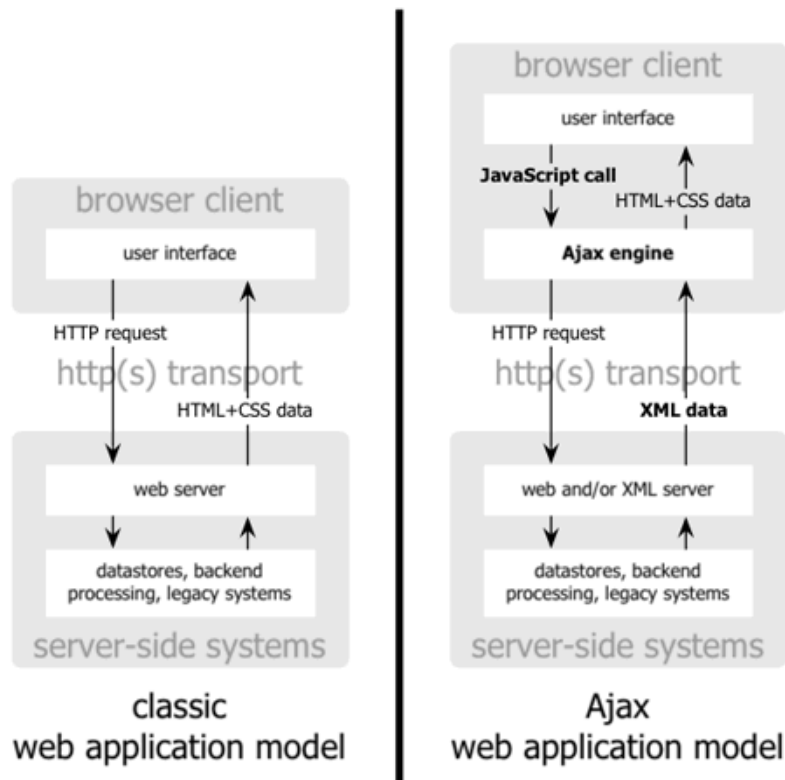


Figura 6. Modelos de aplicaciones web.

Debido a las peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitado.

AJAX permite mejorar considerablemente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano (comunicación asíncrona).

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de

AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

La historia de AJAX está íntimamente relacionada con un objeto de programación llamado XMLHttpRequest. El origen de este objeto se remonta al año 2000 con Alex Hopmann, cuando se desarrollaba la entonces futura versión de Exchange 2000. El punto débil del servidor de correo electrónico era su cliente vía web, llamado OWA (Outlook Web Access).

Motivado por las posibilidades futuras de OWA, se crea la primera versión de lo que denominó XMLHTT, solo restaba incluirlo de forma nativa en el navegador.

Todas las aplicaciones realizadas con técnicas de AJAX deben instanciar en primer lugar el objeto XMLHttpRequest, que es quien permite realizar comunicaciones con el servidor en segundo plano, sin necesidad de recargar las páginas. La implementación del objeto XMLHttpRequest depende de cada navegador.

2.3.2 Reglas del lenguaje. Coincidencias fonéticas

El análisis de la información puede definirse como la aplicación de técnicas de procesamiento automático del lenguaje natural, de clasificación automática y de representación gráfica del conocimiento.

Esta definición corresponde al análisis asistido por computadora. En general, por análisis de la información se entiende la fase de interpretación que el usuario realiza de una manera directa y manual.

Los límites de este tipo de análisis son evidentes desde el momento que se trabaja sobre una cantidad importante de datos, y se trata además de incorporar el análisis en un sistema de producción de información elaborada o especializada.

La identificación unívoca de las personas en los sistemas de registro electrónico es aspecto central difícil de resolver en cualquier escenario. Es sabido que el problema no es simplemente encontrar un identificador único y sus atributos, ya que ninguno ha demostrado ser lo suficientemente eficiente.

En las aplicaciones para la salud, es de vital importancia lograr un registro de pacientes centralizado, para eliminar múltiples ingresos de un mismo registro y para permitir la recuperación y análisis de datos efectiva.

En redes integradas de salud, donde diferentes sistemas informáticos interactúan y comparten datos, este punto es uno de los más críticos a considerar. Entre los motivos más frecuentes por los cuales se produce duplicación de registros de personas al momento del ingreso de datos de un paciente son:

- Errores de cargado con transposición de letras y errores de tipo.
- Registro descentralizado de pacientes en áreas con diferentes políticas y procedimientos.
- Falta de entrenamiento, compromiso y auditoría de los usuarios en el uso del sistema.
- Factores culturales que pueden llevar a información inconsistente tales como intercambiar primeros nombres con los segundos.
- Presión en los puntos de contacto para disminuir los tiempos de registro de pacientes.
- Información brindada por terceros en puntos de atención tales como centrales de emergencia

Cualquiera de las situaciones previas atenta contra un postulado principal de los sistemas dedicados al cuidado de la salud: la posibilidad de error en la asignación de la identidad puede llevar a la creación de registros duplicados.

La verdadera dimensión de este problema se jerarquiza al comprender que el aspecto administrativo siente un impacto menor debido a que sus procesos pueden cumplimentarse independientemente de quien sea el sujeto de la atención, brindando los registros duplicados solamente trastornos en el análisis secundario de las bases de datos administrativas.

Pero el ámbito de la atención de la salud no puede permitirse tal grado de error: la asignación incorrecta de la identidad de un paciente durante el proceso de atención puede tener

consecuencias en los resultados del proceso y aún en la salud del individuo, como por ejemplo la asignación errónea de datos acerca de la positividad de serologías para el virus del VIH o alergias medicamentosas que pueden contraindicar equivocadamente un tratamiento.

En este contexto, se hace necesario paulatinamente obtener a nivel institucional una fuente de identificación única para la persona.

Dentro de la elección de los atributos inalterables resulta útil el registro sistemático del primer apellido materno, que permite discriminar posibles duplicaciones en casos de coincidencia de nombres y apellidos comunes, apellidos que cambian a lo largo de la vida del individuo o casos de registro erróneo de apellidos dobles.

Es necesario verificar previamente a la inserción de un nuevo individuo con datos idénticos o muy similares a los que ya existen, a fines de evitar la incorporación de un mismo sujeto en diferentes ocasiones, generando duplicaciones.

Para ello, es necesaria la normalización fonética y la asignación de importancia diferencial a los atributos.

La fonética es el estudio de los sonidos físicos del discurso humano. Es la rama de la lingüística que estudia la producción y percepción de los sonidos de una lengua en específico, con respecto a sus manifestaciones físicas.

Se basa en la pronunciación de acuerdo con los alfabetos fonéticos, y no con la etimología.

Las búsquedas fonéticas transforman el texto en una representación diferente según la cual dos palabras que suenen igual tendrán la misma representación.

Soundex es un algoritmo fonético, para indexar un nombre por su sonido. El objetivo básico es codificar de la misma forma los nombres con la misma pronunciación o bastante similar. Fue desarrollado por Robert Russell y Margaret Odell y patentado en 1918 y 1922. Una variación llamada American Soundex fue desarrollada en 1930 para realizar un análisis retrospectivo del censo de los Estados Unidos de América entre los años 1890 y 1920. El código de Soundex tomó notoriedad en los años 60 cuando protagonizó varios artículos en *Communications of the ACM* y *Journal of the ACM* de la Association for Computing Machinery, y especialmente al ser descrito en la obra maestra de Donald Knuth, *The Art of Computer Programming*.

Knuth busca la esencia de los problemas informáticos y los aborda desde un punto de vista matemático. Se abordan aspectos novedosos que, o bien no existían en el momento de la creación de los textos originales, o eran demasiado avanzados (o se encontraban en un estado aún embrionario), como para proceder a una formalización.

La obra fue incluida por la revista "American Scientist", en la lista de los 100 libros que más influencia científica han tenido a lo largo de la historia.

El código Soundex para un nombre consiste en una letra seguida de tres números: la letra es la primera letra del nombre, y el número codifica el resto de consonantes. Las consonantes que suenan de forma similar comparten el mismo número así, por ejemplo, la B, F, P y V son codificadas como 1. Las vocales pueden afectar la codificación, pero nunca se codifican directamente a menos que aparezcan al principio de los nombres.

2.3.3 Comunicación entre componentes

La comunicación es el proceso mediante el cual se transmite información de una entidad a otra. Los procesos de comunicación son interacciones mediadas por signos entre al menos dos agentes que comparten un mismo repertorio de signos y tienen unas reglas semióticas comunes.

Tradicionalmente, la comunicación se ha definido como "el intercambio de sentimientos, opiniones, o cualquier otro tipo de información mediante habla, escritura u otro tipo de señales". Todas las formas de comunicación requieren un emisor, un mensaje y un receptor destinado, pero el receptor no necesita estar presente ni consciente del intento comunicativo por parte del emisor para que el acto de comunicación se realice. En el proceso comunicativo, la información es incluida por el emisor en un paquete y canalizada hacia el receptor a través del medio. Una vez recibido, el receptor decodifica el mensaje y proporciona una respuesta.

En el mundo de las aplicaciones informáticas no deja de ser importante este tema. En ocasiones es preciso que componentes de software relacionados, interactúen entre sí, e intercambien mensajes de importancia, en dependencia del negocio que estén modelando. Puede ser con aplicaciones encuestando el tema o servicios web que se consumen para mantener actualizados nuestros componentes.

En SiSalud, se proponen los avisos como forma de comunicación entre componentes con el fin de velar por la calidad de los servicios en salud. En función de las reglas del negocio, los mismos se generarán automáticamente, también teniendo en cuenta el tipo de usuario autenticado.

2.3.4 Reportes en las aplicaciones informáticas

Un reporte es un documento, generado por el Sistema, que presenta de manera estructurada y/o resumida, datos relevantes guardados o generados por la misma aplicación de tal manera que se vuelvan útiles para la Institución.

Los reportes generalmente agrupan los datos de acuerdo a un interés específico. A diferencia de un formulario, los datos dentro de un reporte no pueden ser manipulados o modificados directamente, sino que tienen que ser afectados en alguna otra parte del Sistema para que se reflejen los cambios una vez que el reporte sea generado nuevamente.

Un reporte es generado dinámicamente, es decir, cada vez que se invoca desde el Sistema, el reporte actualiza la información a los datos más recientes disponibles.

En el caso particular de SiSalud, los reportes constituyen un elemento clave para el análisis y la toma de decisiones, por parte de los directivos. Actualmente, la consolidación de la información que se gestiona, carece de relevancia en ocasiones, por errores propios del negocio, y en gran medida, porque la forma en que se presentan no es factible para los que lo utilizan.

Se estudiaron tendencias utilizadas en aplicaciones de gestión, que se comercializan exitosamente debido a la aceptación del usuario final, con el fin de implementarlas en SiSalud en la medida de lo posible.

2.3.5 Integración de aplicaciones de terceros. Servicios Web.

A mediados de la década de los 90 y con la aparición de Internet y su posterior, ha existido siempre la necesidad e inquietud por parte de las empresas desarrolladoras de software de buscar o contar con la manera de lograr la integración entre sistemas heterogéneos. Para tal efecto muchas compañías fueron creando de forma individual la mejor manera de lograr esta integración, cada vez más difícil. La manera de hacer negocios y la comunicación entre las personas y las empresas ha cambiado significativamente. Bajo este contexto se hizo mayor la

necesidad de integrar y compartir información entre distintas plataformas de software y hardware.

En lugar de crear una plataforma integrada se decidió algo mejor, buscar un lenguaje común de intercambio de información aprovechando los estándares existentes en el mercado.

Bajo este contexto nacen los Servicios Web basados en XML.

Los web services son componentes software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, vía Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos web estándar, como XML, SOAP, UDDI o WSDL. El objetivo final es la creación de un directorio en línea de servicios web, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad.[15]

Aunque la idea de la programación modular no es nueva, el éxito de esta tecnología reside en que se basa en estándares conocidos y la utilización de los servicios web aporta ventajas significativas a las empresas. El principal objetivo que se logra, es la interoperabilidad y la integración. Mediante los mismos, las empresas pueden compartir servicios software con sus clientes y sus socios de negocio. La integración de aplicaciones hace posible obtener la información demandada en tiempo real, acelerando el proceso de toma de decisiones.

Requisitos de un Servicio Web:

- Interoperabilidad: Un servicio remoto debe permitir su utilización por clientes de otras plataformas.
- Amigabilidad con Internet: La solución debe poder funcionar para soportar clientes que accedan a los servicios remotos desde internet.
- Interfaces fuertemente tipadas: No debería haber ambigüedad acerca del tipo de dato enviado y recibido desde un servicio remoto. Más aún, los tipos de datos definidos en el servicio remoto deben poderse corresponder razonablemente bien con los tipos de datos de la mayoría de los lenguajes de programación.
- Posibilidad de aprovechar los estándares de Internet existentes: La implementación del servicio remoto debería aprovechar estándares de Internet existentes tanto como sea posible. Una solución construida sobre un estándar de Internet ampliamente adoptado

puede aprovechar conjuntos de herramientas y productos existentes creados para dicha tecnología.

- Soporte para cualquier lenguaje: La solución no debería ligarse a un lenguaje de programación particular. Un cliente debería ser capaz de implementar un nuevo servicio Web existente independientemente del lenguaje de programación en el que se haya escrito el cliente.
- Soporte para cualquier infraestructura de componente distribuida: La solución no debe estar fuertemente ligada a una infraestructura de componentes en particular. De hecho, no se debería requerir el comprar, instalar o mantener una infraestructura de objetos distribuidos, solo construir un nuevo servicio remoto o utilizar un servicio existente. Los protocolos subyacentes deberían proporcionar un nivel base de comunicación entre infraestructura de objeto distribuidos existentes tales como DCOM y CORBA.

El lenguaje de descripción de servicios Web (WSDL, Web Service Description Language) es un dialecto basado en XML sobre el esquema que describe un servicio Web. Un documento WSDL proporciona la información necesaria al cliente para interactuar con el servicio Web. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red, incluyendo SOAP sobre HTTP e incluso protocolos que no se basan en XML como DCOM sobre UDP. [15]

Dado que los protocolos de comunicaciones y los formatos de mensajes están estandarizados para la web, cada día aumenta la posibilidad e importancia de describir las comunicaciones de forma estructurada. WSDL afronta esta necesidad definiendo una gramática XML que describe los servicios de red como colecciones de puntos finales de comunicación capaces de intercambiar mensajes. Las definiciones de servicio de WSDL proporcionan documentación para sistemas distribuidos y sirven como fórmula para automatizar los detalles que toman parte en la comunicación entre aplicaciones.

Los documentos WSDL definen los servicios como colecciones de puntos finales de red o puertos. En WSDL, la definición abstracta de puntos finales y de mensajes se separa de la instalación concreta de red o de los enlaces del formato de datos. Esto permite la reutilización de definiciones abstractas: mensajes, que son descripciones abstractas de los datos que se están intercambiando y tipos de puertos, que son colecciones abstractas de operaciones. Las

especificaciones concretas del protocolo y del formato de datos para un tipo de puerto determinado constituyen un enlace reutilizable. Un puerto se define por la asociación de una dirección de red y un enlace reutilizable; una colección de puertos define un servicio. Por esta razón, un documento WSDL utiliza los siguientes elementos en la definición de servicios de red:

- Types: contenedor de definiciones del tipo de datos que utiliza algún sistema de tipos (por ejemplo XSD).
- Message: definición abstracta y escrita de los datos que se están comunicando.
- Operation: descripción abstracta de una acción admitida por el servicio.
- Port Type: conjunto abstracto de operaciones admitidas por uno o más puntos finales.
- Binding: especificación del protocolo y del formato de datos para un tipo de puerto determinado.
- Port: punto final único que se define como la combinación de un enlace y una dirección de red.
- Service: colección de puntos finales relacionados.

A continuación se detalla un poco más en profundidad cada uno de estos elementos

El elemento Types contiene información de esquema referenciado en el documento WSDL. El sistema de tipos predeterminado que admite WSDL es de esquema de XML. Si se usa esquema de XML para definir los tipos que contiene el elemento Types el elemento schema aparecerá inmediatamente como elemento hijo.

Se pueden utilizar otros sistemas de tipo tipos por extensión. Si desea, utilizar otro sistema de tipo puede aparecer un elemento de extensibilidad bajo el elemento Types. El nombre de este elemento debería identificar el sistema de tipos utilizados. El esquema de XML es el sistema de tipos dominante en los documentos WSDL.

El elemento Message proporciona una abstracción común para el paso de mensajes entre el cliente y el servidor. Como puede utilizar múltiples formatos de de definición de esquema en documento WSDL es necesario de disponer de un mecanismo común de identificar los

mensajes. El elemento Message proporciona este nivel común de abstracción al que se hará referencia en otras partes del documento WSDL.

Pueden aparecer múltiples elementos Message en un documento WSDL, uno para cada mensaje que se comunica entre el cliente y el servidor. Cada mensaje contiene uno o más elementos "Part" que describen las piezas del contenido del mensaje. Un ejemplo de una parte es el cuerpo de un mensaje de SOAP o un parámetro que forma parte de una cadena de petición, un parámetro codificado en el cuerpo del mensaje de SOAP o todo el cuerpo de un mensaje de SOAP.

El elemento portType contiene un conjunto de operaciones abstractas que representan los tipos de correspondencia que pueden producirse entre el cliente y el servidor. Para los Servicios Web de estilo RPC se puede pensar en un portType como una definición de internas en donde cada método se puede definir como una operación.

Un tipo puerto se compone de un conjunto de elementos operation que define una determinada acción. Los elementos operation se componen de mensajes definidos en el documento WSDL. WSDL define cuatro tipos de operaciones denominadas tipo operaciones:

- Request-response (petición-respuesta) comunicación del tipo RPC en la que el cliente realiza una petición y el servidor envía la correspondiente respuesta.
- One-way (un-sentido) Comunicación del estilo documento en la que el cliente envía un mensaje pero no recibe una respuesta del servidor indicando el resultado del mensaje procesado.
- Solicit-response (solicitud-respuesta) La contraria a la operación petición-respuesta. El servidor envía una petición y el cliente le envía de vuelta una respuesta.
- Notification (Notificación) La contraria a la operación un-sentido el servidor envía una comunicación del estilo documento al cliente.

El elemento binding contiene las definiciones de la asociación de un protocolo como SOAP a un determinado bindingType. Las definiciones binding especifican detalles de formatos del

mensaje y el protocolo. Por ejemplo, la información de asociación especifica si se puede acceder a una instancia de un portType de forma RPC.

Las definiciones binding también indican el número de comunicaciones de red que se requieren para realizar una determinada acción. Por ejemplo, una llamada RPC de SOAP sobre HTTP podría involucrar un intercambio de comunicación HTTP, pero esa misma llamada sobre SMTP podría involucrar dos intercambios de comunicaciones de SMTP discretas.

La asociación se logra utilizando elementos de extensión. Cada protocolo tiene su propio conjunto de elementos de extensión para especificar los detalles del protocolo y el formato de los mensajes. Para un determinado protocolo los elementos de extensión se suelen utilizar para decorar las acciones individuales de una operación y la propia operación con la información de asociación del protocolo. A veces los elementos de extensión se utilizan en el propio nivel portType.

Un servicio es un grupo de puertos relacionados y se definen en el elemento service. Un puerto es un extremo concreto de un Servicio Web al que se hace referencia por una dirección única. Los puertos que se definen en determinado servicio son independientes. Por ejemplo, la salida de un puerto que no puede utilizarse como una entrada de otro.

Los elementos de extensibilidad se utilizan para representar determinadas tecnologías. Por ejemplo, se puede utilizar los elementos de extensibilidad para especificar el idioma que se utiliza en el esquema de los elementos types.

El esquema para un determinado conjunto de elementos de extensibilidad se debe definir dentro de distintos espacios de nombres que WSDL. La definición de los propios elementos puede contener un atributo wsdl:required que indique un valor boolean si el atributo required se establece a true en una definición de elementos una asociación que haga referencia a ese conjunto concreto de electos de extensibilidad tiene que incluir dicho elemento.

Lo más habitual es que los elementos de extensibilidad se utilicen para especificar especificación de asociación. La especificación WSDL define un conjunto de elementos de extensibilidad para la asociación SOAP, HTTP GET, HTTP POST, MIME. Sin embargo, la especificación sólo define las asociaciones para dos de los cuatro tipos de operaciones. Un sentido y petición repuesta.

Herramientas para programar los servicios web. IDE

Un entorno de desarrollo informático (en inglés Integrated Development Environment) es un programa informático compuesto por un conjunto de herramientas de programación.

Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación.

Algunos IDE's soportan múltiples lenguajes, tales como Eclipse o NetBeans, ambas basadas en Java o MonoDevelop, basado en C#. El soporte para lenguajes alternativos es a menudo proporcionada por plugins, que les permite ser instalado en el mismo IDE para soportarlos. Por ejemplo, Eclipse y NetBeans tiene plugins para C / C + + , Ada , Perl , Python , Ruby y PHP , entre otros lenguajes.

Una decisión tanto o más importante a la hora de desarrollar en PHP que el framework a utilizar, es el IDE, ya que el entorno de desarrollo que se elige puede suponer una verdadera diferencia en el tiempo de trabajo invertido

NupherePhped Profesional

NupherePHPEd es un IDE para PHP, que tiene compatibilidad con otros lenguajes como: Javascript, SQL, HTML, Perl, C++. Es un editor de código avanzado, contiene un inspector de código en tiempo de ejecución, cliente para conexión a bases de datos y posibilidades de despliegues rápidos y seguros.

Aunque requiere ciertos conocimientos para su utilización, y posee características importantes como: sistema de ayuda, autocompletado de fórmulas, remarcación de errores, diferentes colores para diferenciar los lenguajes, soporte de lenguaje Java, entre otros.

Posee buen rendimiento, aunque carece de características tales como refactorización o integración de pruebas unitarias y control de versiones. Tiene herramientas para la generación

de código, para trabajar con SOAP y para operaciones CRUD (Create, Retrieve, Update, Delete). Su obtención no es gratuita y es solo para sólo para sistema operativo Windows.

NeatBeans

Es una opción ideal para los grupos de trabajo que utilizan software libre. Es importante destacar que consume bastante recursos del ordenador y presenta problemas en ocasiones con la codificación de los ficheros, aspecto clave en la aplicación en cuestión, debido al procesamiento xsl de los xml de respuesta. Es multiplataforma (Windows, Linux, Mac OS X y Solaris), gratuito, de código abierto y se puede utilizar para programar en otros lenguajes además de PHP.

Además de las funciones básicas con las que cuenta cualquier IDE, posee otras funcionalidades como la integración con PHPUnit para las pruebas unitarias y con CVS, Subversion y Mercurial para el control de versiones.

Zend Studio

Desarrollado por una de las empresas que más han contribuido al desarrollo de PHP, y creadora de Zend Framework y Zend Server, Zend Studio se basa en Eclipse PDT, al que añade funcionalidades como generación de código, formateo de código configurable, pruebas unitarias y desarrollo remoto. No es gratuito, aunque sí multiplataforma (Linux, Windows y Mac OS).

Dentro de las características más importantes puede mencionarse la posibilidad de edición, generación de wsdl y soporte para el trabajo con servicios web.

Alternativas implementadas

Softel, como empresa que controla y dirige el desarrollo de las aplicaciones de salud del país que formarán parte del Sistema Integral de Salud (SiSalud) según necesidades del MINSAP, necesita que las mismas, exporten los resultados de muchos de los negocios que gestionan. De esta manera, se exponen los servicios para ser utilizados por terceros, tal es el caso de los nomencladores, que tienen que ser utilizados de manera única por todas las aplicaciones. Así mismo, se exponen servicios de nuestro negocio, que resuelven búsquedas e incluso en casos muy aislados, inserciones.

Han existido durante algún tiempo, varios intentos por parte de los desarrolladores, de generar los wsdl de cada servicio, de manera automática. De igual manera, los mismos debían generarse con un formato estándar, capaz de entender por aplicaciones desarrolladas en cualquier otra plataforma, dígase .net, GeneXus, o Java, que son las que se utilizan dentro de la propia Empresa.

Con el entorno existente para el desarrollo, sin un IDE eficiente y falta de herramientas que facilitaran las pruebas a los servicios, este trabajo era muy costoso en tiempo, y en recursos. Humanamente se vuelve bien complicado, cuando los servicios envuelven un número considerable de procesos de negocio, hacer el wsdl de forma manual, cuando con la lógica de programación debía ser suficiente.

En los intentos de estandarización de los servicios, estuvo la idea e implementación en la medida de lo posible, de realizar los servicios web de la aplicación en Java, utilizando AndroMDA.

Andromda es un framework (marco de trabajo) de generación extensible de código que se adhiere al paradigma de la arquitectura dirigida por modelos. Los modelos de las herramientas de UML son transformados en componentes desplegados para algunas plataformas. Incluye un conjunto de bibliotecas enfocadas a herramientas de desarrollo actuales como son Axis, jBPM, Struts, JSF, Spring e Hibernate. Plugins para Eclipse y el trabajo con UML 2.0

Las versiones más actuales lo desacoplan de Maven, lo que permite utilizarlos con mayor facilidad en diferentes herramientas como son maven1, maven2, Eclipse, Ant, entre otros.

Algunos, de los más sencillos, fueron terminados y probados satisfactoriamente. Pero esta alternativa tuvo como limitantes, el que los desarrolladores tuvieran que aprender otro lenguaje de programación, herramientas de modelado, configuraciones específicas para el entorno de trabajo, que más que ser de ayuda, se volvieron un estorbo en el proceso. Además, que el proceso de desarrollo, no se iba a centrar únicamente en el desarrollo de los servicios a exponer a terceros, si no, que el proceso de mantenimiento tanto correctivo como evolutivo debía continuarse y a la par, poder de alguna manera, estandarizar los mismos.

Era preciso dar respuestas acertadas, en un tiempo prudencial, porque las aplicaciones de terceros esperaban por la estandarización de los servicios para su integración al Sistema

Nacional de Salud, incluso las propias aplicaciones de Softel, desarrolladas en otras plataformas requerían de las mismas necesidades.

Surge la idea entonces, con un previo estudio de la herramienta y analizando las potencialidades de la misma para este tema, el utilizar el Zend Studio for Eclipse con este fin.

Zend Studio es un editor de texto para páginas PHP que proporciona facilidades desde la creación y gestión de proyectos hasta la depuración del código.

En las últimas versiones, el IDE es basado en Eclipse, la cual es una plataforma de software de código abierto independiente de especificidades para desarrollar, lo que el proyecto llama “Aplicaciones de Cliente Enriquecido”. Es importante mencionar que es más ligero que los anteriores y posee características significativas para agilizar el trabajo con los servicios web.

Conclusiones parciales

Se estudiaron algunas de las técnicas y tecnologías actuales aplicadas en el desarrollo de aplicaciones de gestión para seleccionar las más adecuadas para implementar en una primera fase de mejoras para SiSalud.

1. La creación de la máquina virtual ACE, implementación de VMware Workstation 6.0 integrada a la utilización del Subversion para el control de versiones del código, permite flexibilizar el entorno de trabajo y agilizar los tiempos de desarrollo.
2. La utilización de Ajax mejora la interacción del usuario con la aplicación, evitando las recargas constantes de la página (comunicación asíncrona).
3. La utilización del algoritmo de análisis fonético permite la identificación unívoca de las personas, elemento de vital importancia en las aplicaciones para la salud, permitiendo eliminar múltiples ingresos de un mismo registro.
4. La comunicación entre componentes y la creación de reportes que apoyen la toma de decisiones, mejorarían considerablemente la calidad de los servicios ofrecidos al cliente final.
5. La creación de servicios web estandarizados, con una herramienta de generación automática de documentos de descripción permite a las aplicaciones externas integrarse eficientemente y desde cualquier plataforma.

CAPITULO II. Implementación de mejoras a SiSalud.

La propuesta de mejoras al sistema en su conjunto y su implantación, aumenta en gran medida la eficiencia en el proceso de desarrollo del producto y gestión del proyecto, así como la satisfacción del usuario final en el momento de explotación del mismo.

Con este fin se utilizan herramientas para el control de versiones, así como entornos apropiados para el desarrollo, en dependencia de las necesidades del equipo. Se integran tecnologías actuales del desarrollo web para mejorar los tiempos de respuesta de la aplicación y se utilizan algoritmos para evitar la duplicidad de información, aspecto crucial para las aplicaciones de salud. Se optimiza la programación de reportes utilizando técnicas tradicionales de herramientas ofimáticas que apoyen de manera significativa la toma de decisiones.

Igualmente, se realizan esfuerzos para lograr la programación estandarizada de los servicios web que se exponen a aplicaciones de terceros, con un costo mínimo debido a la herramienta estudiada y utilizada para ello.

1. Proceso de desarrollo y liberaciones de SiSalud.

1.1. Control de versiones y Máquinas Virtuales ACE

El proceso de desarrollo de SiSalud dentro de la Factoría de Software, tiene como objetivo analizar, agendar, estimar el tiempo, implementar, probar y liberar las solicitudes de nuevos desarrollos o cambios que se reciben de la Unidad de Servicios Informáticos.

Ante la presentación de una solicitud de cambio para algunos de los componentes del RIS en la herramienta Rational ClearQuest, se realizan las siguientes acciones:

1. Los analistas de sistema analizan la solicitud que se encuentra en estado de *Submitted*, si la misma contiene la información suficiente se estima el tiempo que demorará en ejecutarse, asignándosela al trabajador que la ejecutará. Si la solicitud no posee toda la información necesaria para iniciar su trabajo, el analista la pone en estado de *Más Info*, para que la unidad de servicio le brinde más información sobre la misma, en caso de que la solicitud haya sido presentada anteriormente, en el tab de *Resolution* se pone el atributo de *resolution* como *duplicate* y la solicitud se pasa al estado de *Duplicate*.
2. El analista de sistema y/o programador trabaja en las solicitudes de cambio según el cronograma definido previamente.

3. El analista de sistema y/o programador ejecuta las pruebas necesarias para chequear que los cambios realizados funcionan correctamente.

Posterior a las pruebas realizadas a todos los componentes por parte del integrador, se preparaba una liberación que es ubicada en un servidor para que el Gestor del Producto, los despliegue en sus máquinas de prueba y si todo es satisfactorio, posteriormente en Infomed. Este proceso, en ocasiones sufría interrupciones en la copia, demora para el despliegue, y falta de ficheros en las liberaciones por falta de disciplina en el proceso.

Incluso dentro de la factoría, el proceso se tornaba ineficiente. Para que no existiera duplicidad o que se trabajara sobre copias de código no actualizadas, los programadores debían trabajar sobre la misma copia, desplegada en una máquina a la que todos accedieran a la vez, por supuesto, siempre con la premisa de no modificar el mismo fichero.

Esto era una máquina virtual VMWare, con Linux, a la que se accedía al código con la herramienta WINSCP, la cual tiene un editor bastante pobre.

Se implementaron un nuevo entorno para desarrollo y procedimiento para las liberaciones a Servicio.

Solución para entorno de desarrollo. VMware ACE

VMware ACE es una solución de virtualización de escritorios altamente controlada y fácil de gestionar para usuarios de empresas.

Al admitir una única plataforma VMware, los usuarios de VMware Workstation 6 con ACE Option Pack pueden crear y empaquetar de forma segura máquinas virtuales VMware ACE para implementarlas entre los usuarios finales.

El código de la aplicación situado en la partición D:/ de cualquier computadora, en la carpeta RISWC (copia de trabajo del RIS), se mantiene actualizado diariamente, con la versión estable de la misma. Excepto cuando los cambios son muy significativos o difieren en gran medida de la versión actual, y necesita pruebas e integración con otros componentes, que se trabaja en una rama del repositorio.

Cada programador, que trabaja con un componente diferente, sube sus modificaciones y actualiza su código para tenerlo en cuenta para próximos cambios.

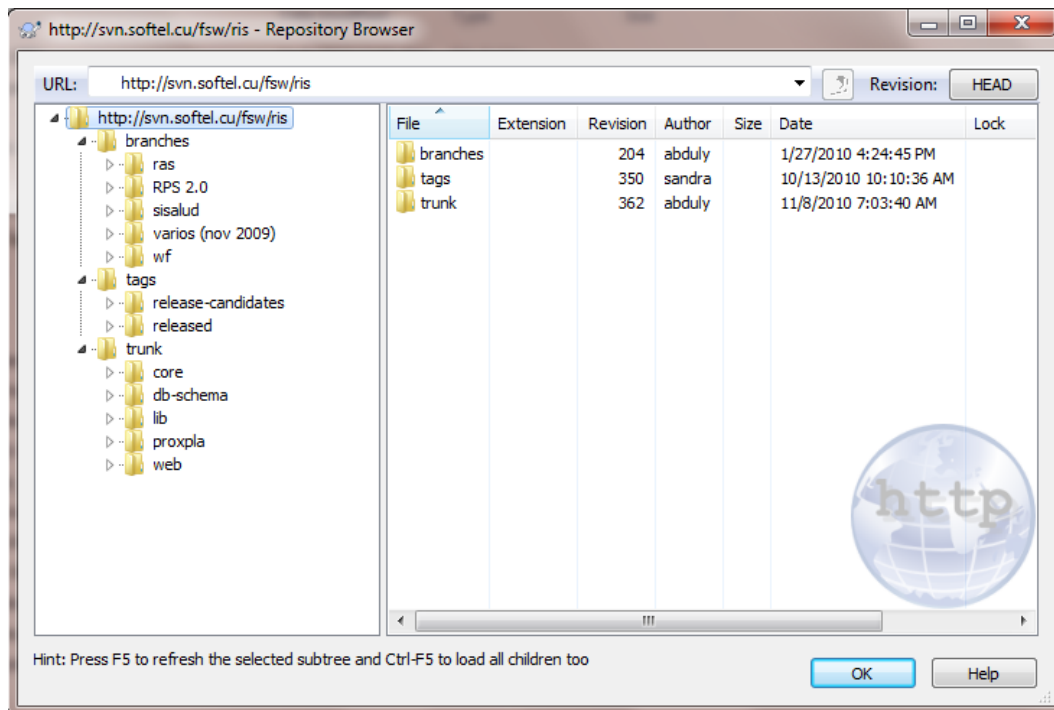


Figura 7. Estructura del subversion para la aplicación SiSalud.

Por tanto, el proceso actual incluye:

1. Cada estación de trabajo adiciona y sube al servidor de subversión los ficheros nuevos o modificados. Puede ser para el **Trunk** o para el **Branches**.
 2. El integrador prueba todos los componentes en el entorno preparado en el servidor de máquinas virtuales y prepara una liberación en el subversion para Servicios, en el tag **Release-candidate**.
 3. Se pone el atributo de *resolution* con cualquiera de los siguientes datos:
 - a. Fixed: Cuando la solicitud fue arreglada:
 - b. Fixed indirectly: Cuando la solicitud fue arreglada de forma indirecta cuando se estaba trabajando en otra solicitud.
 - c. Function as Designed: Cuando la solicitud funciona según se diseñó.
 - d. Duplicate: Cuando la solicitud está duplicada.
- y se le agrega una nota con algunos detalles.

4. Se prepara un acta con comentarios para enviar por correo a la gestora de productos, la cual de cada una de las solicitudes, especifica los señalamientos necesarios y se avisa de la nueva liberación.
5. El analista verifica en el Rational ClearQuest si existen solicitudes de cambio en estado de *Reject-Validate*, de existir alguna, esto se considera como un rechazo de la solicitud de cambio por la Unidad de Servicio y se comienza nuevamente.
6. El analista debe verificar que las solicitudes de cambio que se encuentran en estado de *Validate*, pasan al estado de *Closed*. Implantada la nueva liberación del producto en Infomed. Cada liberación desplegada debe ser actualizada en el tag **Release**, para tener control del despliegue de las mismas, para el trabajo del cliente final.

La gestora de productos, tiene permisos de lectura a las liberaciones. Solo puede copiar el código a desplegar en sus servidores de prueba, y escribe entonces en el subversion para generar las liberaciones estables para explotar en Infomed. Esto significa hacer un vínculo para el tag Release, a partir de uno de los especificados en el Release-Candidate.

Esta forma de trabajo, permite a los desarrolladores, tener una máquina virtual, que con la instalación de la misma, se despliega de manera rápida una aplicación funcional, a la cual es necesario únicamente actualizar la carpeta en la máquina local, del repositorio del Subversion, para estar trabajando en una versión determinada del producto, con el código fuente requerido.

Por tanto, permite utilizar editores eficientes para el trabajo con código php, con completamiento del código y en un entorno de sistema operativo Windows, donde en la máquina ACE se encuentra empaquetada y actualizada, el montaje complejo que necesita la aplicación para ser desplegada, en cuanto a la configuración de los **virtual host**, las restricciones de seguridad que debe tener la base de datos para el acceso a los componentes, entre otras considerables ventajas.

2. Tiempos de respuesta y optimización de procesos.

2.1 Integración de Ajax

Al iniciar una página que utilice tecnología AJAX el navegador cargará el motor AJAX (intermediario entre el cliente y el servidor) en vez de la página Web. Una respuesta a una acción del usuario que no necesite información del servidor es manejada exclusivamente por el

motor AJAX y si se necesitaran datos de este, las peticiones se realizarán asíncronamente usando XML sin que el usuario se percate de estas.

En el RIS la utilización de esta tecnología es de gran utilidad ya que mejora considerablemente los tiempos de respuesta de cada petición al servidor.

Hoy, en la aplicación de manera general, cuando se realiza una solicitud al servidor, se envía la página completa teniendo que cargarla nuevamente en su totalidad para mostrarle la información al usuario. Esto ocasiona que lo que debería ser una búsqueda simple puede tornarse más compleja, al tener que buscar no solamente los datos deseados sino también todos los necesarios para el correcto funcionamiento de la página (dígase privilegios del usuario, datos almacenados en sesión para el llenado interactivo de los componentes). Una vez introducida esta tecnología solamente se buscará en cada petición lo que se necesita sin tener que actualizar la página completa.

JQuery es una biblioteca o framework libre y de código abierto, pensada para acceder a los elementos del DOM de un modo más simplificado.

Al incluir esta biblioteca en la web se cuenta con un conjunto de funciones para los efectos visuales, compatibilidad con todos los navegadores e integración con Ajax; que simplifican el desarrollo de aplicaciones complejas del lado del cliente.

Las búsquedas y los reportes eran los más afectados con la sobrecarga, se mantuvo el xml, enviado por el servidor y parseado por el xslt en el cliente. Después de cargada, se asocian los tags HTML a variables JQuery y mediante Ajax se hace la solicitud al servidor para buscar los datos de los elementos que van a ser actualizados en la página. De esta forma la respuesta es más rápida y se evita el error por tiempo excedido de ejecución.

El primer componente de SiSalud al que se le aplicó Ajax fue el Registro de Localidades.

Este componente, perteneciente al Registro Informatizado de Salud (RIS), se refiere a la clasificación administrativa de los espacios o territorios de residencia tanto rurales como urbanos, por debajo del nivel administrativo Municipal donde habita la población. Las localidades se encuentran registradas en el Registro de Ubicaciones.

Tiene como objetivo principal, gestionar en tiempo real la información detallada de la estructura político-administrativa que conforman los municipios: Consejos Populares, Circunscripciones,

Zonas y CDR con que cuenta el país; en aras de la posterior configuración de las Áreas de Salud.

Debido a la necesidad de la consolidación de la información a este nivel, fue decisión de la dirección del proyecto comenzar en este, con la aplicación de la tecnología.

En el ejemplo en cuestión se muestra como en el evento onchange del combobox donde se muestran los Municipios y se escoge uno, se envía una petición al servidor y este devuelve las localidades y los consejos populares del mismo, para mostrar en pantalla la información recibida.

El primer paso es incluir en el fichero de transformación xslt correspondiente, la librería jquery.js y los ficheros de JavaScript (.js) en los que se va a programar dicha funcionalidad.

Se implementan las funcionalidades teniendo siempre como premisa la reutilización de código cuando sea posible, y lo que no, se programa a la medida.

Los parámetros para estas funciones definen los valores enviados al servidor, así como la dirección de la página que atenderá la petición en cuestión, y otra función es la encargada de hacer la petición AJAX. Se tratan los errores del modo que conciba el programador.

Resultado de las pruebas

A continuación se listan los resultados de las pruebas realizadas a la aplicación con las mejoras propuestas, según los entornos en los que se despliega la misma.

Para ello es preciso mencionar las características de la computadora de cualquier desarrollador donde se instala la máquina virtual de trabajo, el ACE.

Dual Core 2.00 GHz, 2.00 Gb de RAM, 80 Gb de Disco duro. Sistema operativo Windows XP o Seven.

Y el servidor de máquinas virtuales, tiene un disco duro SCSI, con 26 Gb de memoria RAM y dos procesadores Quad Core a 2.00 GHz de velocidad.

Entornos	I. Máquina ACE (VMWare) (I)	II. Servidor de máquinas virtuales(aplicación distribuida por capas en máquinas virtuales diferentes)	III. Máquinas de Infomed (aplicación distribuida por capas en máquinas físicas diferentes)
----------	-----------------------------	---	--

Sistema operativo	Linux (Debian etch)	Linux (Debian etch)	Linux (Debian etch)
Memoria RAM	256 Mb	256 Mb	3 Gb
Capacidad de disco duro	10 GB	3 Gb	80 Gb (web y core) 240 Gb (data) SCSI
Procesador	2.00 GHz	2.00 GHz	3.06 GHz

Se presentan algunas funcionalidades claves de los módulos para especificar los tiempos de respuestas.

Componente: Registro de Localidades

Entorno	I	II	III
Funcionalidad	Mostrar todos los consejos populares	Mostrar todos los consejos populares	Mostrar todos los consejos populares
Tiempo anterior	4 seg	5 seg	5 seg
Tiempo posterior	2 seg	3seg	3seg
Cantidad de datos	1 080 tuplas	1 080 tuplas	1 080 tuplas

Componente: Registro de Trabajadores de la Salud

Entorno	I	II	III
Funcionalidad	Mostrar todos los profesionales de la salud	Mostrar todos los profesionales de la salud	Mostrar todos los profesionales de la salud
Tiempo anterior	6 seg	7 seg	7 seg
Tiempo posterior	3 seg	3 seg	3 seg
Cantidad de datos	121 000 tuplas	121 000 tuplas	121 000 tuplas

2.2 Utilización de Soundex en la inserción de ciudadanos

Debido a la gran diversidad de variantes léxicas en las cadenas de texto de los nombres y apellidos, origen de posibles errores que pueden llegar a generar duplicados (cambios entre c y s, b y v, entre otros) se recurre a la adaptación al español de un algoritmo de normalización por medio de decodificación fonética (Soundex). Esto permite obtener cadenas “normalizadas” donde las variantes léxicas han disminuido al punto de permitir sospechar como posibles duplicados aquellas combinaciones de caracteres que aún sin ser exactamente iguales, tienen una pronunciación muy similar y pueden encubrir duplicaciones.

Se tomaron como coincidencias absolutas, el caso en que el carnet de identidad y los códigos resultado de los apellidos son iguales, debido a su poder discriminatorio y coincidencias semejantes cuando coinciden los códigos de los apellidos y fecha extraída del CI es igual a la fecha de nacimiento.

Para optimizar el funcionamiento de las búsquedas de ciudadanos en el RIS, se utilizó este algoritmo fonético, que permite a través de criterios definidos para el tema en cuestión, cuándo una respuesta podría ser considerada una coincidencia semejante o igual.

Para esto se desarrolló una clase que está compuesta por un método principal que su nombre es “soundexExt”, este es el encargado de codificar un apellido a una forma fonética con reglas aplicadas según el idioma español.

La clase tiene otras 2 funciones auxiliares, “quitarEspacios” y “eliminarAcentos” las cuales son funciones auxiliares que su nombre bien explica cuál es su función.

Para la utilización en el RIS, fue necesario agregar dos atributos en la Base de Datos de ciudadanos, para los códigos del primer y segundo apellido (codAp1) y (codAp2) respectivamente.

Fue necesario aplicar un script para codificar los apellidos de cada uno de los ciudadanos que se encontraban ya almacenados para que se tomaran en cuenta para las búsquedas, esto disminuye la posibilidad de inserción de información redundante. Luego, hubo que modificar los servicios de inserción y modificación para que se tuvieran en cuenta estos apellidos codificados a la hora de realizarse algunas de estas acciones.

Esta mejora, incide considerablemente en dos de los componentes más utilizados del RIS, que son Registro de Ciudadanos (RC) y el módulo de Autenticación y Autorización (SAAA). La

búsqueda fonética permite, la búsqueda de ciudadanos, previa a la inserción, para eliminar redundancias, a veces dadas simplemente, por la ausencia de búsquedas que tengan en cuenta, coincidencias significativas con el que vaya a insertarse

La utilización de este algoritmo dentro de SiSalud, de manera eficiente, depende del trabajo a realizar por los editores del sistema a los diferentes niveles de atención. Las organizaciones deberán abocar recursos humanos para sortear exitosamente el problema de la multiplicidad de registros y los errores devenidos con ellos.

Lograr cambios en la cultura institucional e incorporar los procesos necesarios sin entorpecer el funcionamiento habitual de todas las áreas es uno de los mayores desafíos.

En los entornos de prueba se ha verificado que el 98 % de los ciudadanos a insertar en el sistema que poseen coincidencias semejantes o absolutas, son los mismos que ya existen en la base de datos, por lo que la aplicación del algoritmo evita redundancias y la presencia de información duplicada.

Se considera como una de las fortalezas del modelo, la presencia de una auditoría permanente sobre los datos. De esta forma, cuando a pesar de las medidas preventivas algo indica que se han ingresado registros sospechosos de ser idénticos, la resolución de la situación es de resorte humano, aportando las correcciones necesarias.

2.3 Reportes con paginado horizontal

Principalmente en el despliegue del componente Registro de Personal de la Salud (RPS), los reportes cuyo resultado debía mostrar más de un cierto número de columnas, eran desglosados utilizando una paginación horizontal; lo cual dificultaba el análisis del reporte completo y en ocasiones era necesario hacer varias peticiones al servidor para ver un registro específico en el mismo.

Esto también provocaba que en una sola página no pudiese disponerse de un reporte en toda su extensión, de manera que se tuviera un consolidado de la información con los detalles en las filas, de todos los elementos de las columnas, pues estas últimas también había que ubicarlas en otra página, si excedía la cantidad determinada para el reporte.

Se estudia entonces para la nueva versión de RPS, que se denomina Registro de Trabajadores de la Salud (RTS), un modo de resolver esta limitante. En este caso, con el objetivo fundamental del componente que es la gestión en línea y en tiempo real de los datos del

personal profesional y no profesional vinculado con todas las Unidades del Sistema Nacional de Salud. Estos son: la información general, ubicación laboral y lo referente a cargos administrativos, especialidades, categorías docentes e investigativas, grados científicos y otros aspectos, además ofrece la posibilidad de realizar búsquedas de todo el Personal de la Salud por diferentes criterios de selección para la elaboración de informes u otras necesidades de información cuyo resultado podrá imprimir posteriormente.

Desde la versión anterior, para mejorar la forma de visualización de estas tablas tan extensas sin afectar el diseño de la página, se toma una idea implementada en las hojas de cálculo de Excel, “Inmovilizar paneles”. Esto significa moverse por el reporte utilizando un scroll (barras de desplazamiento) vertical y horizontal, manteniendo siempre visible el encabezado de las filas y las columnas; logrando que el usuario pueda observar cualquier porción del reporte sin perder de vista los nombres de la fila y columna de la celdas que está analizando.

PROFESIONALES DE SALUD POR PROVINCIAS						Exportar a Excel	Filtro
Provincias/Perfil	Estomatólogo	Lic. Enfermería	Lic. en Tecnología de la Salud	Otros Profesionales y Especialidades	Técnico F	Subtotales	
Pinar del Río	303	1149	11	578		4319	
La Habana	528	976	16	371		4006	
Ciudad de La Habana	1606	2607	47	1391		18560	
Isla de la Juventud	93	229	2	0		632	
Matanzas	525	888	2	483		4973	
Cienfuegos	239	882	0	519		3111	
Villa Clara	586	1873	3	1000		7919	
Sancti Spiritus	319	996	36	410		4160	
Ciego de Avila	248	747	2	241		2900	
Camagüey	441	1641	4	461		5636	
Las Tunas	353	1032	0	462		4088	
Holguín	622	1862	0	662		7221	
Granma	601	1592	38	539		6458	
Santiago de Cuba	620	2231	20	690		8054	
Guantánamo	253	793	27	311		3481	
Totales	7337	19498	208	8118		85518	

Figura 8. Ejemplo de reporte con barras de desplazamiento vertical y horizontal.

La implementación de este tipo de reporte utiliza especificaciones de la biblioteca JQuery, para el acceso a los elementos html y sus eventos.

La respuesta de los servicios web de los reportes en el módulo RPS consiste en tres arreglos; el primero con las etiquetas de las filas, el segundo con las etiquetas de las columnas y un tercer arreglo bidimensional con los valores de las celdas interiores del reporte.

Para que estos datos lleguen a la presentación en el formato deseado, los arreglos son insertados en tablas (formato HTML), y se generan otras dos con el cálculo de los totales y los subtotales en la página php de la capa de presentación. Esta estructura es recogida mediante Ajax en formato JSON (acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos) para ser insertadas en la página html, que ya contiene una tabla con la estructura del reporte, con varios espacios con identificadores donde mediante las funciones JQuery estos espacios son cubiertos por las tablas obtenidas en la respuesta enviada por la página servidora.

Mediante dos funciones implementadas en JQuery, se sincroniza el evento “onscroll”, de los subtotales, las celdas interiores y las etiquetas de las filas. De manera similar, los de los totales, las celdas interiores (este espacio es común en los dos desplazamientos) y las etiquetas de las columnas; dando como resultado que el movimiento de las barras de desplazamiento sea parejo en todos los espacios y no afecte la correspondencia entre los elementos del reporte.

Para exportar estos reportes a fichero externo, solo se da la posibilidad para Excel, pues por tamaño de las tablas y la cantidad de información que muestran, no es posible ubicarlo en el tamaño de hoja estándar de un documento con extensión .pdf.

La clase “Excel”, utiliza del paquete PEAR la clase "Spreadsheet Excel Writer". Esta clase estaba concebida para exportar los resultados de las búsquedas, que a diferencia de los reportes no presentan etiquetas laterales. Debido a la similitud del resultado final, se decide agregar algunos métodos a esta clase para incrementar la utilidad de la misma en cuanto a generación de reportes se refiere.

La generación de este tipo de reportes, le provee operatividad al personal que trabaja con los mismos, ya que al ser la cantidad de información grande se volvía engorroso el análisis de la misma, así como su impresión en pantalla.

2.4 Componente lista de trabajo

Los Avisos se presentan como una propuesta para los directivos que deben velar por la calidad de los servicios en salud. Es la comunicación que realizan los componentes entre sí y que debe llegar al usuario para acometer las acciones que se deriven de cada uno. Los mismos se generan automáticamente, en función de las reglas del negocio para cada componente, por lo

que tendrá especificado también el tipo de usuario para los que será visible cada uno en específico.

Se muestran al usuario desde que se autentifica en el sistema; es decir, los avisos son personales y no modulares; esto evita que el usuario tenga que entrar a diferentes módulos para recibir los que le conciernen.

Ejemplos de avisos pueden ser:

- Parto de una embarazada
- Paciente es alta en otra Área de Salud
- Un paciente tiene un problema de salud
- Se realizó una acción de salud a un paciente
- Falleció un paciente
- Se captó a un nacido vivo
- Un paciente tiene un problema de salud (EDO)

La Suite de Colaboración Zimbra (Zimbra Collaboration Suite o ZCS) es un programa informático colaborativo o Groupware creado por Zimbra Inc., y adquirida más tarde por Yahoo! Inc. Con el acuerdo de mantener sus estándares de Software Abierto. Posteriormente fue vendida por Yahoo a VMWare. Posee tanto el componente de servidor como su respectivo cliente. Existen varias versiones de Zimbra disponibles: una versión soportada por la comunidad de código abierto, y otras con parte del código cerrado y soportada comercialmente que contiene algunas mejoras.

El Servidor ZCS hace uso de proyectos Open Source existentes como son: Postfix, MySQL, OpenLDAP y Lucene. Expone una interfaz de programación de aplicaciones (API) SOAP para toda su funcionalidad y actúa tanto como un servidor IMAP y POP3.

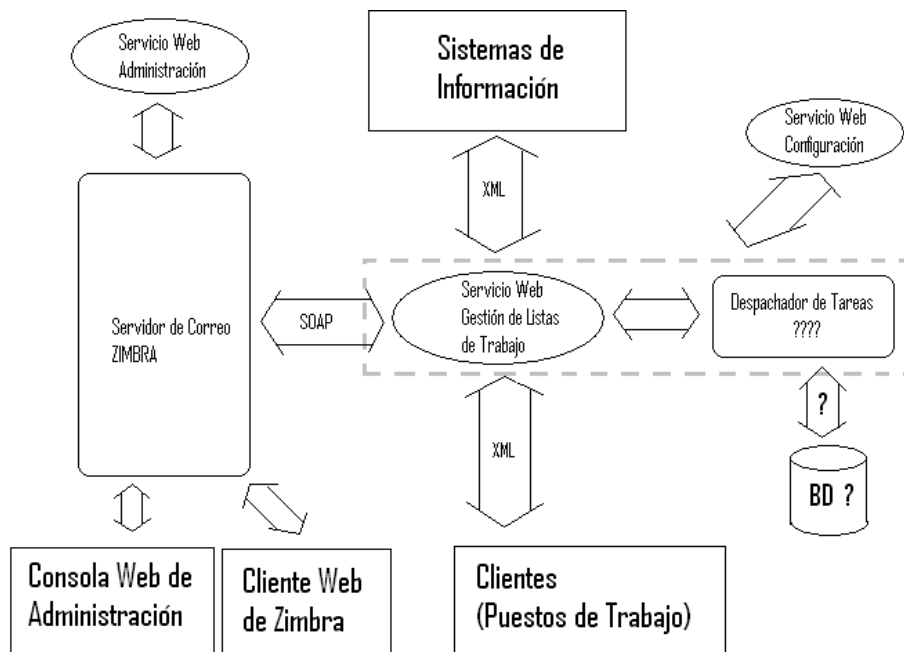
El Cliente Web ZCS es una interfaz de colaboración y administración completa creada empleando el Toolkit Zimbra Ajax. Soporta correos electrónicos y calendarios a través de una impresionante interfaz web Ajax. También incluye capacidades de búsqueda avanzada y permite relacionar fechas. El calendario compartido en grupo también está incluido.

ZCS es compatible con clientes propietarios tales como Microsoft Outlook, Novell Evolution y Apple Mail. También provee soporte de sincronización nativo de dos vías para muchos dispositivos móviles.

Zimbra te permite montar todo un sistema de correo electrónico basado en paquetes libres. Se trata de una recopilación de paquetes ya existentes y probados junto con una programación de Webmail. El mérito de Zimbra está en su interfaz de Webmail (con soporte a Ajax) y en el empaquetado de todas las aplicaciones de terceros, como son:

- Apache + Tomcat (servidor Web),
- Clamav (antivirus),
- Spamassassin + DSpam (filtro de spam),
- Amavis-new (conector entre los antivirus y el servidor de correo),
- Jdk (Máquina Virtual de Java),
- MySQL (servidor de base datos),
- OpenLdap (servidor de directorio),
- Postfix (servidor de correo SMTP) y
- Cyrus (servidor de correo POP/IMAP).

Gestión de avisos



Dentro de la categoría de Sistemas de Información se encuentran todos aquellos sistemas o módulos que utilizarán el componente de Gestión de Listas de Trabajo, ya sea para el envío de dichas listas a los Clientes o por ejemplo, el envío de avisos.

Estos sistemas interactuarían con el Servicio Web Gestión de Listas de Trabajo mediante el envío de documentos XML que contengan los ítems de listas de trabajo.

Los Clientes son los que solicitan los ítems de Listas de Trabajo. Existen dos tipos de clientes:

1. Los que interactúan con el Servicio Web Gestión de Listas de Trabajo para obtener dichos ítems.
2. Los que obtienen la Lista de Trabajo directamente del Servidor de correos (clientes de correo).

Los primeros son aplicaciones clientes de servicio web, las cuales se encargarán de recibir los ítems en forma de documentos XML, tales son los casos de las Listas de Trabajo y los módulos que reciben avisos.

Los otros, clientes de correo, por ejemplo, el cliente web de Zimbra, desarrollado con tecnología Ajax y que brinda una gran variedad de oportunidades, como la mensajería instantánea (chat) y la incorporación de los llamados Zimlets, que permiten interactuar con otros componentes (servicios web por ejemplo) ubicados en cualquier parte sin necesidad de abrir ninguna otra interfaz.

El Servicio Web Gestión de Listas de Trabajo sería el componente que se encargaría de realizar operaciones propias de un cliente de correos (enviar, recibir, borrar mensajes, entre otros.). Para ello utilizaría el protocolo SOAP mediante una API desarrollada en el lenguaje de programación JAVA.

Las tareas de administración (crear usuarios, eliminar usuarios) se pueden realizar a través de la Consola Web de Administración de Zimbra. También puede analizarse la opción de realizar las tareas administrativas mediante un Servicio Web u otra aplicación, utilizando el protocolo SOAP sobre http seguro.

Los Sistemas de Información llaman al método encargado de enviar mensajes en el Servicio Web Gestión de Listas de Trabajo, enviando así las tareas o avisos representados por un conjunto de parámetros que forman parte de un documento XML, uno de los cuales contiene la

dirección de correo del destinatario, que pudiera ser uno de los clientes descritos en el segundo punto del apartado anterior, por lo tanto los mismos tendrían su propio buzón en el Servidor Zimbra.

El Servicio Web Gestión de Listas de Trabajo actúa como un cliente de correo, enviando los datos en el cuerpo de un mensaje dirigido a la dirección del destinatario. Cuando un destinatario necesita acceder a su buzón en busca de la Lista de Tareas o Avisos, llama al método encargado de obtener los mensajes en el Servicio Web Gestión de Listas de Trabajo, al cual se le pasa como parámetro un documento XML de solicitud (con la identificación del cliente) y devuelve otro documento XML con los ítems de Listas de Trabajo o Avisos. Debido a que cada destinatario tiene un buzón, entonces para poder acceder a los mensajes dentro de dicho buzón es necesario enviar el nombre de usuario y la contraseña respectivos dentro del XML de solicitud, en dependencia de la estructura definida.

El servicio web de gestión de Listas de Trabajo y Avisos cuenta con los métodos: enviarLista, enviarMensaje, obtenerLista y borrarLista.

Para el componente de avisos se utilizarán los tres últimos, cuyo funcionamiento se describe a continuación:

enviarMensaje: se utiliza para enviar un aviso a una dirección de correo determinada. Para ello se le pasa como parámetro un documento xml con una estructura determinada.

Un ejemplo de xml que puede ser utilizado como entrada sería:

```
<wlDocumentType>
<Email>
<To>
    sisSalud@sld.zimbra.cu
</To>
<To>
    xxx@infomed.sld.cu
</To>
<User>
    RPS
</User>
<Password>
    rpspassword
</Password>
</Email>
<Comment>
<Text>Este es un aviso de nacimiento en el Hospital.....</Text>
```

```
</Comment>  
</wIDocumentType>
```

Este método devuelve otro xml:

```
<wIDocumentType>  
<End>  
<EndCode>1</EndCode>  
</End>  
</wIDocumentType>
```

donde EndCode devuelve 1 si el mensaje fue enviado y 0 si no se envió.

obtenerLista: se utiliza para obtener todos los mensajes de un buzón determinado. Un ejemplo de xml de entrada es:

```
<wIDocumentType>  
<Email>  
<User>  
sisSalud  
</User>  
<Password>  
sisSaludpassword  
</Password>  
</Email>  
</wIDocumentType>
```

Este método devuelve otro xml con todos los mensajes en dicho buzón. Un ejemplo:

```
<MultiDocumentType>  
<wIDocumentType>  
<Email>  
<MsgId>2245</MsgId>  
</Email>  
<Comment>  
<Text>Este es un aviso de nacimiento en el Hospital.....</Text>  
</Comment>  
</wIDocumentType>  
<wIDocumentType>  
<Email>  
<MsgId>2248</MsgId>  
</Email>  
<Comment>  
<Text>Falleció Pedro Perez con carné de identidad 8312013445.....</Text>  
</Comment>  
</wIDocumentType>  
</MultiDocumentType>
```

El campo MsgId tiene el id de cada mensaje, el cuál será uno de los campos del xml de entrada del método borrarLista.

borrarLista: se utiliza para borrar uno o más mensajes determinados de acuerdo a sus id. Ejemplo de xml de entrada:

```
<wIDocumentType>
<Email>
<User>
sisSalud
</User>
<Password>
sisSaludpassword
</Password>
<MsgId>2245</MsgId>
<MsgId>2248</MsgId>
</Email>
</wIDocumentType>
```

Este método devuelve un xml con una lista de id correspondientes a los mensajes borrados. Ejemplo:

```
<wIDocumentType>
<Email>
<MsgId>2245</MsgId>
<MsgId>2248</MsgId>
</Email>
</wIDocumentType>
```

Un error en cualquiera de estos métodos descritos anteriormente devolverá el siguiente xml:

```
<wIDocumentType>
<End>
<EndCode>0</EndCode>
<RecordType>Aquí va el texto que describe el error</RecordType>
</End>
</wIDocumentType>
```

El fichero WLDocumentSchema es un XML Schema que define los parámetros de entrada y salida de todos los métodos. Todos los campos (tags) son opcionales por lo que el cliente del servicio web puede utilizar los que desee, excepto <Email> que es de utilización obligatoria, puesto que en él se envían los datos de usuario necesarios para la autenticación en zimbra.

Los avisos permiten interactividad entre los componentes que se relacionan dentro de SiSalud, así como que cada usuario autenticado sea consciente de la información relacionada con él, que se maneja dentro del sistema; de modo que pueda desencadenar acciones en dependencia de cada negocio y sus especificidades.

2.5 Estandarización de Servicios Web.

Se opta por la utilización del nuevo IDE Zend Studio for Eclipse para el desarrollo de los servicios web a exponer a terceros, por las facilidades que posee en cuanto a la generación automática de los wsdl y las facilidades de la programación orientada a objetos.

Esta realidad representó un problema crucial durante mucho tiempo, se tornaba muy costoso que otras aplicaciones desarrolladas en disímiles plataformas se integraran a la nuestra, teniendo en cuenta el dinamismo con que se pedía exponer servicios de los registros centralizados, en dependencia de los negocios que trataran las mismas y la detección de sus necesidades.

Estos servicios web se desarrollan utilizando la librería Soap de php, no la que encapsula PLASER. Para ello se implementa la clase servicioWeb, que se encarga de resolver los aspectos relacionados con la seguridad, el nivel de acceso de los usuarios, y además contiene funciones generales para validaciones. Además resuelve las peticiones realizadas desde otras plataformas, como por ejemplo GeneXus, que enviaba el mensaje Soap con una estructura que fue necesario inspeccionar, para obtener la solicitud correctamente.

Es la encargada de crear un servicio web utilizando la librería antes mencionada donde se especifica la dirección del servicio y el nombre; por la que todos los servicios que se implementen, deben heredar de la misma.

La funcionalidad a la que se trata de acceder, se verifica a través de un fichero de configuración, si está disponible o no, y si necesita o no autenticación.

Utilizando las potencialidades de la programación orientada a objetos, se declaran las clases a utilizar, y de los atributos se comentan los tipos de datos para ser utilizados en la creación automática del wsdl. Si se conocen los parámetros de entrada y las salidas de las funciones, las mismas se declaran y se genera el wsdl sin necesidad de haber escrito una línea de código.

El cliente hace la petición, y a través del wsdl crea la instancia de la clase del servicio web donde se implementaron las diferentes funcionalidades.

El Zend Studio permite generar el wsdl a partir de las clases definidas. El mismo queda resuelto en un porcentaje alto, solo le resta agregar la estructura para el token e incluirla en el fichero, para asegurar la autenticación adecuada para el acceso al servicio.

Anteriormente la programación de un servicio web y la escritura de su wsdl podía tomarle dos días de trabajo al programador, en dependencia de la complejidad del código. Con la utilización adecuada de la herramienta, el wsdl es generado de manera automática al término de la programación del servicio, de modo que permite la comunicación entre diferentes plataformas sin transformaciones.

Conclusiones parciales

1. Se define una estructura para llevar el control de versiones de la aplicación, la cual se integra con un paquete de virtualización local que permite a los desarrolladores tener un entorno flexible y actualizado para la edición de código.
2. Con la utilización de la tecnología Ajax se disminuyen los tiempos de respuesta de cada petición al servidor, se simplifica la complejidad de las aplicaciones del lado del cliente y se mejoran los aspectos de efectos visuales y compatibilidad con los navegadores.
3. Con la utilización del algoritmo fonético Soundex en el proceso de búsquedas de ciudadanos en el RIS, disminuye la propabilidad de inserción de personas duplicadas al sistema.
4. Se implementa la inmovilización de paneles para mejorar el modo de visualización de grandes volúmenes información sin afectar el diseño de la página. Esto permite moverse por el reporte utilizando barras de desplazamiento vertical y horizontal, manteniendo siempre visible el encabezado de las filas y las columnas; logrando que el usuario pueda observar cualquier porción del reporte sin perder de vista los encabezados.
5. La gestión de avisos permite la comunicación de los componentes, de manera que el usuario puede acometer las acciones que se deriven de los mismos.
6. La estandarización de los servicios web permite la integración de las aplicaciones externas, así como la generación automática de los documentos wsdl.

CONCLUSIONES

1. SISalud, más que un sistema, es una plataforma de aplicaciones, abierta y con una interfaz de programación que permite incorporar nuevos módulos compatibles entre sí, permitiendo la integración de componentes. Es una plataforma en constante desarrollo.
2. Facilita a todos los niveles de salud la información estandarizada y homogénea. Sirve de guía, consulta y aprendizaje para la actividad pre-profesional y post-profesional.
3. La aplicación de técnicas y herramientas actuales utilizadas en el desarrollo de aplicaciones web propicia la utilización de un entorno de desarrollo flexible y la disminución considerable de los tiempos de desarrollo del proyecto así como los tiempos de respuestas de la aplicación.
4. La estandarización de los servicios web permite la adecuada integración de aplicaciones externas.

RECOMENDACIONES

1. Realizar estudio para utilizar una herramienta de integración continua de código abierto que se encargue del monitoreo de algunas funciones.

REFERENCIAS BIBLIOGRAFICAS

Cabrera M., SOFTEL. *Propuesta de Esquema del Sistema de Información para la Salud (SISalud)*. 2005.

Comité Ejecutivo del Consejo de Ministros (CECM), *Lineamientos estratégicos para la informatización de la Sociedad Cubana*. La Habana, Cuba. Junio, 1997.

Collins-Sussman B., Fitzpatrick B., Pilato C. *Control with Subversion*. Sitio consultado: 10/08/10. Disponible en <http://chestofbooks.com>

Delgado A., Vidal M., *Informática en la Salud Pública Cubana*. Rev. Cubana Salud Pública 2006; 32(3).

Delgado Ramos, A., Cabrera Hernández, M., Juncal V. *Registro Informatizado de Salud (RIS)*. Revista Estadística. Junio 2005.

Eguíluz J. *Ajax: A New Approach to Web Applications*. 2005. Sitio consultado: 14/09/10. Disponible en: <http://www.adaptivepath.com/ideas/essays/archives/000385.php>

Garfi L., Navajas P., Gómez A., Luna D., González B. *Implementación de un sistema centralizado para la identificación de pacientes en un hospital de alta complejidad*. Sitio consultado: 09/11/10 Disponible en:

http://www.hospitalitaliano.com.ar/archivos/servicios_attachs/1125.pdf

INFOMED. Acerca de Infomed. Sitio consultado 01/09/10. Disponible en: <http://www.sld.cu/acerca/acercade.html#que>

Knuth D. *The Art of Computer Programming*. Sitio consultado: 20/10/10. Disponible en: <http://media.pragprog.com/articles/other-published-articles/ArtInProgramming.pdf>

Luna C. *OMS/La revolución de la atención primaria en Cuba cumple 30 años*. Mayo 2008. Sitio consultado 01/09/10. Disponible en: <http://www.who.int/entity/bulletin/volumes/86/5/08-030508/es/index.html>

NuSphere Corp. *NuSpherePhpED - The World Famous PHP IDE*. Sitio consultado: 27/09/10. Disponible en: <http://www.nusphere.com/products/phped.htm>

Paz V., De Paz J. *Servicios Web*. Sitio consultado: 16/08/10. Disponible en:
<http://zarza.usal.es/~fgarcia/doctorado/iweb/05-07/Trabajos/ServiciosWeb.pdf>

Pan American Health Organization (1998b). *Sistemas de Información y Tecnología de Información en Salud: Desafíos y Soluciones para América Latina y el Caribe. Programa de Sistemas de Información sobre Servicios de Salud / División de Desarrollo de Sistemas y Servicios de Salud*. Washington, DC., Junio 1998.

Pompa F., SOFTEL. *Guía para el Desarrollo de los Módulos del Registro Informatizado de Salud (RIS)*. 2006.

Servicios Web en plataforma .NET. Sitio consultado: 16/08/10. Disponible en:
<http://www.desarrolloweb.com/manuales/54>

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

Vidal M., De Armas, Y., *Estrategias de informatización del Sector de la Salud (I)*. Revista Informatic@Médica. Año 3. No. 11. Diciembre 2002.4:24-27.

VMware, inc. 2007. *VMware Workstation 6 El software de virtualización de escritorios líder*. Sitio consultado: 17/08/10. Disponible en: <http://www.vmware.com>

VMware, inc. 2007. *VMware ACE 2 Tome las riendas de los escritorios virtuales en la empresa*. Sitio consultado: 17/08/10. Disponible en:
http://i.dell.com/sites/content/business/solutions/brochures/es/Documents/vsl-vmware-ace_es.pdfso

BIBLIOGRAFIA

A Business-Oriented Foundation for Service Orientation. Disponible en:

http://msdn.microsoft.com/architecture/solutions_architecture/service_orientation/default.aspx?pull=/library/en-us/dnbda/html/ServOrient.asp

Diseño y desarrollo de aplicaciones web. Disponible en: http://www.babsoft.com/es/disenio_desarrollo_aplicaciones_web.php

Huibert, Aalbers. *¿Cómo lograr una implementación exitosa de SOA?* Disponible en:

http://www.huibert-aalbers.com/IT_Insight/Spanish/PDF/ITI007Sp-SuccessfulSOAImplementation.pdf

Patterns & Practices: Web Service Software Factory. Disponible en:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/ServiceFactory.asp>

Productos y tecnologías de virtualización. Disponible en:

<http://www.microsoft.com/latam/virtualizacion/products/server/default.msp>

Salm Junior, José Francisco. *Arquitectura SOA: Uso eficiente de Web Services, WSDL e UDDI.*

Disponible en: <http://bvs4.icml9.org/gt/ti/activity.php?lang=es&id=3>

Ventajas de la virtualización. Disponible en: <http://www.virtualizacion.com/>

Service Orientation and Its Role in Your Connected Systems Strategy. Disponible en:

http://msdn.microsoft.com/architecture/solutions_architecture/service_orientation/default.aspx?pull=/library/en-us/dnbda/html/srorientwp.asp

Software as a Service (SaaS): An Enterprise Perspective. Disponible en:

<http://msdn2.microsoft.com/en-us/architecture/aa905332.aspx>

Tecnología aplicaciones web. Disponible en: <http://www.slideshare.net/zamanthag/tecnologia-aplicaciones-web>

Anexos