

Universidad de las Ciencias Informáticas

Facultad 7



**Servicios para la administración remota del sistema
alas PACSServer**

**Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas**

Autores: Ariel Udaeta Hernández

Marisol Guerra Aguilera

Tutores: Ing. Yasmay Gómez Suarez

Ing. Rafael Fernández Peñalver

La Habana, junio de 2011

“Año 53 de la Revolución”

Datos de Contacto

Tutores:

Tutor principal:

Ing. Yasmay Gómez Suárez: Graduado de Ingeniero en Ciencias Informáticas, egresado de la UCI en el 2008. Con categoría de Instructor. Ha impartido la asignatura Práctica Profesional I, Estándares Internacionales en la Gestión Informatizada de Salud y alas PACS-RIS. Pertenece al Departamento de Software Médico Imagenológico del Centro de Salud donde se desempeña como analista de software.

Correo: ygomez@uci.cu.

Co tutor

Ing. Rafael Fernández Peñalver: Graduado de Ingeniero en Ciencias Informáticas, egresado de la UCI en el 2009. Ha impartido clases de Introducción a la Programación, Programación 1, Aplicaciones informáticas para el Sector de la Salud y Estándar DICOM. Es profesor de la Facultad 7 y se desempeña actualmente como Administrador de la configuración en el Departamento de Producción de Software Médico Imagenológico, CESIM.

Correo: rpenalver@uci.cu.

AGRADECIMIENTOS

A la Revolución, por habernos dado la oportunidad de desarrollarnos profesionalmente.

Al compañero Fidel por darnos la oportunidad de estudiar en una escuela como esta.

A los tutores Rafael y Yasmay por su ayuda, muchas gracias.

A los profesores que nos han formado como profesionales.

A nuestros amigos.

A nuestros compañeros.

A todos lo que de una forma u otra nos han ayudado.

Gracias...

RESUMEN

Hoy en día los sistemas de procesamiento de imágenes médicas digitales han alcanzado un desarrollo incalculable en todo el mundo. La existencia de aplicaciones informáticas, que trabajan con imágenes digitales, ayuda al personal de salud a realizar una mejor atención y calidad de los pacientes. Para facilitar el trabajo de los galenos en los hospitales se utilizan las estaciones remotas que es el mantenimiento de sistemas informáticos a distancia. Una de estas aplicaciones informáticas, es el sistema alas PACSServer, el cual no posee una herramienta para la administración remota de su configuración y de los servicios que ofrece.

El desarrollo del servicio de administración remota para el sistema alas PACSServer esta guiado por el Proceso Unificado de Desarrollo que utiliza UML 2.1 como lenguaje de modelado. Utilizando como IDE de desarrollo Visual Studio 2008 con una tecnología de desarrollo ASP.NET y C# como lenguaje de programación. Como cliente del Subversion TortoiseSVN en su versión 1.6.6, la herramienta de modelado usada fue Enterprise Architect 7.5 y como Sistema de Gestión de Base de Datos se utiliza PostgreSQL en su versión 8.2.

Se desarrolló una herramienta que permite gestionar el sistema alas PACSServer, la cual tiene como beneficios la posibilidad de permitir administrar su configuración desde otro equipo, muestra el control del tráfico de las conexiones de forma gráfica y la información de estas en tiempo real, directamente desde la Conexión de Área Local.

Palabras claves:

Sistema alas PACSServer, administración remota, configuración, estaciones remotas, aplicaciones informáticas.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACION TEÓRICA	6
1.1. TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	6
1.2. METODOLOGÍA DE DESARROLLO.....	12
1.3. HERRAMIENTAS.....	17
1.4. RESULTADOS ESPERADOS.....	21
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	22
2.1. BREVE DESCRIPCIÓN DEL SISTEMA.....	22
2.2. MODELO DEL DOMINIO.....	22
2.3. ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE.....	25
2.4. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA.....	26
CAPÍTULO 3. ARQUITECTURA Y DISEÑO DEL SISTEMA	38
3.1. DESCRIPCIÓN DE LA ARQUITECTURA.....	38
3.2. MODELO DE DISEÑO.....	40
3.3. DESCRIPCIÓN DE LA CLASE DE DISEÑO.....	40
3.4. DISEÑO DE LA BASE DE DATOS.....	40
CAPÍTULO 4. IMPLEMENTACIÓN	42
4.1. DIAGRAMA DE COMPONENTES.....	42
4.2. DIAGRAMA DE DESPLIEGUE.....	43
4.3. COMPATIBILIDAD CON LA PLATAFORMA LIBRE.....	44
CONCLUSIONES	47
RECOMENDACIONES	48
REFERENCIAS BIBLIOGRÁFICAS	49
BIBLIOGRAFÍA	51
ANEXOS	54
ANEXO 1: REALIZACIÓN DE CASOS DE USO DEL DISEÑO.....	54

ANEXO 2: DESCRIPCIÓN DE LAS CLASES.....	59
ANEXO 3: DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS	69
GLOSARIO DE TÉRMINOS	72

INTRODUCCIÓN

Desde sus inicios, la informática ha sufrido disimiles cambios. Desde los grandes ordenadores que permitían realizar limitadas tareas y el uso exclusivo de organizaciones muy selectas, hasta los ordenadores de la actualidad ya sean tanto personales o portátiles, los cuales tienen las mismas capacidades e incluso mayores que los primeros y los cuales están cada vez más relacionados con el quehacer cotidiano de las personas.

En la década de los setenta se atribuyeron dos grandes cambios en el mundo de la informática:

Primero se tuvo el desarrollo de los microprocesadores, los cuales permitieron reducir en costo y tamaño los ordenadores y aumentar en gran medida la capacidad de estos y su acceso a más personas.

Como segundo se obtuvo la oportunidad de conectar los ordenadores con posibilidad de transferencia de datos de alta velocidad y esto se logró mediante el desarrollo de las redes de área local y de las comunicaciones. (1)

Precisamente en esta década surgió el término **radiología digital**. Este concepto evolucionó la tecnología de los equipos de adquisición de imágenes médicas. Modalidades como la Tomografía axial Computarizada (CT) comenzaron a desarrollarse con muy buenos resultados. En los ochenta surgieron otras de gran impacto como la Resonancia Magnética de Imágenes (MRI por sus siglas en inglés). Por el carácter no invasivo de las radiaciones que emite y las ventajas que desde el punto de vista clínico que ofrece, se convirtió rápidamente en una de las más fiables y flexibles. (2)

Partiendo de estas y otras tecnologías como el ultrasonido (US), la medicina nuclear (NM), las redes de comunicación, las computadoras personales y los medios de almacenamiento, se hizo factible la transmisión de imágenes a distancia; por otra parte el almacenamiento de estudios Imagenológicos en formato electrónico, las técnicas y algoritmos para el procesamiento de imágenes digitales dan comienzo a la carrera de los Sistemas de Transmisión y Almacenamiento de Imágenes (PACS, por sus siglas en inglés). Estos sistemas están formados por las modalidades o equipos de adquisición de imágenes, la infraestructura de comunicaciones, servidores de bases de datos y de archivos, y las estaciones de visualización y procesamiento de la información. Todos esos elementos se integran de forma coherente a través de aplicaciones de software que son la interfaz entre los usuarios y la infraestructura tecnológica.

Con el objetivo de favorecer el desarrollo y expansión de los PACS, surge el estándar DICOM (Digital Imaging and Communication in Medicine) el cual se encuentra en su versión 3.0. Este no es más que el estándar industrial para la transferencia y visualización de imágenes médicas digitales y la información asociada entre equipos y computadoras, a su vez incluye la definición de un formato de fichero y de un protocolo de comunicación de red. Además permite y facilita la interoperabilidad en un entorno de múltiples estaciones y modalidades. (2)

La estandarización de las imágenes médicas hace posible que, mediante una transmisión segura, los datos de los pacientes puedan viajar de departamento en departamento y de hospital en hospital, garantizando que toda la información pueda ser vista fuera del equipo de adquisición que le dio origen sin costo adicional al paciente y sin tener en cuenta la marca de los equipos que las visualizan o las almacenan. Estas ventajas que brinda la estandarización de las imágenes médicas, se han convertido en un factor muy importante para países del tercer mundo que no pueden adquirir todos los equipos del mismo fabricante por los altos costos de estos, donde podemos incluir a nuestro país, que además de poseer la condición de país del tercer mundo tiene como agravante de su situación económica el férreo bloqueo impuesto por los Estados Unidos.

Teniendo en cuenta los beneficios que traen consigo estas revolucionarias tecnologías, hoy en día los sistemas de procesamiento de imágenes médicas digitales han alcanzado un desarrollo incalculable en todo el mundo. Poco a poco los hospitales y clínicas se han dado cuenta que el futuro está en digitalizar sus radiografías en sistemas PACS ya que esto garantiza diagnósticos rápidos y confiables, de los cuales la mayoría genera la información en forma de imágenes médicas, así como la transmisión de estos de un lugar a otro en tiempo real y su almacenamiento de forma persistente, dándoles una alta disponibilidad y seguridad. Todos estos beneficios brindan a los médicos la posibilidad de acceder a dichos diagnósticos en cualquier momento y desde cualquier lugar, y así de esta forma estudiarlos, clasificarlos o compartirlos con otros médicos.

La producción de PACS y de todas las tecnologías necesarias para su despliegue en una instalación hospitalaria, es un tema dominado mundialmente por grandes compañías como la Philips, la Siemens, la General Electric, IQ, Kodak entre otras. Para cualquier país, la instalación de dichos productos en una clínica radiológica o en un hospital conllevaría a gastar millones de dólares, teniendo en cuenta que se debería comprar desde el software hasta todo un arsenal de tecnologías, que cubren desde las

conexiones de red y los servidores de imágenes hasta las más simples estaciones de trabajo. El alto costo de estos productos hace que en ocasiones sean inaccesibles por muchos de los países existentes.

Debido a la situación económica de Cuba, sería imposible acceder a esta tecnología y desplegarla en todas las instalaciones hospitalarias, por lo tanto la dirección del país, siguiendo las ideas del Comandante en Jefe Fidel Castro Ruz, se ha centrado en lograr la producción de este tipo de software en sus propias instalaciones destinadas al desarrollo de los mismos. Estos podrían ser utilizados en el ámbito nacional ahorrando al país millones de dólares y además se tendría la posibilidad de comercializarlos con los demás países del mundo, viendo en esto una forma de ayudar a los pueblos y una manera de obtener ingresos que ayuden a la economía, si se tiene en cuenta el alto precio de estos productos en el mercado mundial.

De esta forma, en La Universidad de las Ciencias Informáticas se ha logrado desarrollar un sistema de software llamado alas PACS, el cual integra varias aplicaciones con el objetivo de facilitar la realización de múltiples funciones enfocadas a mejorar los servicios de diagnóstico radiológico. Algunos de los componentes de alas PACS son:

- alas PACSViewer: estación de diagnóstico general, es el encargado de la visualización y el procesamiento de las imágenes médicas.
- alas PACSServer: sistema para la gestión de la transmisión y el almacenamiento de las imágenes.

Este sistema ha sido instalado en varias instituciones hospitalarias del país, sin embargo el proceso de despliegue se ve obstaculizado debido a que el sistema alas PACSServer no posee una herramienta para la administración remota de su configuración y de los servicios que ofrece. En el proceso de despliegue se instala el visor alas PACSViewer en varias estaciones clientes, en ocasiones dispersas dentro de la institución, al servidor de imágenes alas PACSServer hay que configurarle los datos para la comunicación con cada una de estas estaciones clientes, y la gestión de esto se hace de manera local, lo que hace mucho más lento este proceso.

Por otra parte, la visualización de la información relacionada con las conexiones de las estaciones clientes y/o equipos médicos se almacena en un fichero de trazas solamente, no se ofrece de forma gráfica, por lo que su interpretación es bien compleja. Este servidor tampoco tiene la posibilidad de visualizar en tiempo real el control de tráfico de las conexiones al mismo.

Se considera pues, que las instituciones hospitalarias se beneficiarían si el alas PACSServer poseyera los servicios web para la administración remota, ya que ganarían en tiempo y calidad en la utilización de los servidores de imágenes instalados en ellas y se haría óptima la gestión de las conexiones de las estaciones clientes a dichos servidores. Esto otorgaría un mayor dinamismo en el proceso de despliegue del PACS y daría un mayor control sobre la utilización de la red.

Por lo planteado anteriormente se definió como **Problema a Resolver**: ¿Cómo viabilizar el proceso de gestión de la información relacionada con la administración remota del sistema alas PACSServer? Como **Objetivo de Estudio** se tiene el proceso de gestión de la información de un servicio web que posibilite la administración remota, enfocado el **Campo de Acción** el proceso de gestión de la información relacionado con el sistema alas PACSServer.

Como **Objetivo General** del Trabajo de Diploma se tiene la implementación de servicios web que posibiliten la administración remota del servidor de imágenes médicas.

Para dar cumplimiento a este objetivo se definieron las siguientes **Tareas de la Investigación**:

1. Realizar un análisis crítico y valorativo de los sistemas informáticos de administración remota de las imágenes médicas existentes a nivel nacional e internacional, estableciendo similitudes con la investigación en curso.
2. Realizar un análisis crítico y valorativo de las tendencias, técnicas, tecnologías, plataformas, librerías, metodologías y herramientas usadas en la actualidad.
3. Generar los artefactos correspondientes a los Flujos de Trabajo propuestos por la Metodología seleccionada, sirviendo de base a los desarrolladores.
4. Analizar los procesos de negocio asociados a la gestión de la información relacionada con la administración remota, logrando un modelo único como guía para la implementación del sistema.
5. Implementar y realizar pruebas al sistema informático aplicando las pautas de diseño y siguiendo lo establecido en la Especificación de Requisitos de Software.

En el desarrollo del trabajo de diploma, el contenido está estructurado de la siguiente manera:

El **Capítulo 1** aborda en detalle lo relacionado con la fundamentación teórica que sustenta el presente trabajo, así como el estudio del estado del arte de las aplicaciones remotas y los servidores de imágenes médicas aplicadas a esta. También se analizan las principales herramientas, técnicas y tecnologías aplicadas al desarrollo del presente trabajo de diploma.

En el **Capítulo 2** se presenta la propuesta de solución del sistema para la situación problemática, se muestran los procesos del negocio mediante el modelo de dominio y las características y funcionalidades que tendrá el sistema a partir de los requerimientos funcionales y no funcionales del componente de software a desarrollar. Se identifican los actores y casos de uso del sistema, para una correcta presentación del diagrama de casos de uso y se priorizan los casos de uso arquitectónicamente significativos.

En el **Capítulo 3** se describe cómo implementar el componente, a través del diseño, enfocado a cómo el sistema cumple sus objetivos teniendo en cuenta los requisitos funcionales y no funcionales. Aquí se modela en detalle el sistema mediante las clases del diseño, describiendo también las interacciones entre éstas y se especifica el flujo de eventos entre las mismas a través de los diagramas de secuencia.

En el **Capítulo 4** se representan los componentes finales resultantes del sistema y la interacción entre estos; así como el modelo de despliegue, el cual define la forma en que debe realizarse la implantación del sistema.

En la sección de **Anexos** se encuentran los modelos auxiliares, tablas descriptivas de elementos del sistema y descripciones textuales de los casos de uso, entre otros artefactos.

CAPÍTULO 1. FUNDAMENTACION TEÓRICA

En este capítulo se aborda el tema servicios para la administración remota del sistema alas PACSServer. Así como el estudio del estado del arte de las aplicaciones remotas y los servidores de imágenes médicas aplicadas a esta. También se analizan las principales herramientas, técnicas y tecnologías aplicadas al desarrollo del presente trabajo de diploma.

1.1. Tendencias y tecnologías actuales.

En la actualidad, es innegable que la red se ha convertido en un elemento indispensable. Multitud de entidades, empresas, organizaciones, tienen su representación en este medio e incluso pueden ofrecer servicios específicos. No enumeraremos las ventajas y posibilidades que ofrece la red pero es importante resaltar la trascendencia que tiene en la actualidad, y por consiguiente, la necesidad de desarrollar aplicaciones web de calidad. Una aplicación web debe adecuarse siempre a las necesidades para la que es desarrollada. Se suele tender, y más en este campo, a usar las últimas novedades cuando lo que se requiere son aplicaciones estables, eficientes y seguras siempre con una presencia atractiva y moderna.

1.1.1. *Netretina*

En la actualidad existe gran variedad de aplicaciones remotas, tendríamos como ejemplos Netretina, que no es más que un software que permite dar un servicio de soporte remoto profesional a sus clientes y controlar así remotamente sus equipos de forma segura, rápida y sin previa instalación. Permite trabajar con múltiples canales de comunicación de forma simultánea e independiente gracias a la funcionalidad llamada Multi-Channel. Sus operadores de soporte podrán dar soporte remoto desde cualquier lugar y sin tener que instalar ninguna aplicación dado que Netretina es portable. Unifica todas estas funcionalidades en una misma aplicación, simplificando el servicio de soporte e integrando otras muchas funcionalidades. Tiene un soporte técnico vía Web, mantenimiento remoto por Internet vía Web. (3)

1.1.2. **LogMeIn**

Es un software de administración remota, que permite controlar un ordenador remotamente desde el navegador (Comprobado en Mozilla Firefox e Internet Explorer). Además de permitir transferir archivos y otras funcionalidades. Usarlo es muy sencillo, basta con registrarse en el sitio web oficial para poder descargar el software e instalarlo en la máquina que se desea controlar. (4)



Fig.1 Imagen del software LogMeIn

1.1.3. **Estrategia-Telemedicina (5)**

La Telemedicina existe desde fines de la década de 1950 y ha ido evolucionando hasta la actualidad. Esta es la utilización de tecnologías de comunicación e información para proveer asistencia y educación médica a distancia. Esto magnifica la importancia de la tecnología y sus adelantos para ponerla a disposición de la gente, para lograr un equilibrio equitativo y eficaz en los servicios que le competen al área de la salud.

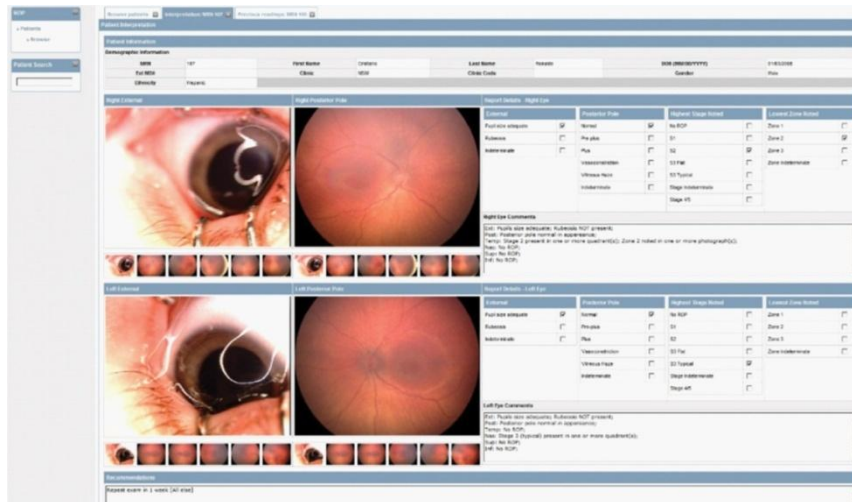


Fig.2 Telemedicina. Asistencia médica a distancia.

Se tendrían los siguientes usos principales:

- Prestación de servicios de medicina a distancia, diagnósticos inmediatos de especialistas Tele-diagnósticos.
- Servicios complementarios e instantáneos a la atención de un especialista (obtención de una segunda opinión).
- Educación remota para estudiantes de medicina.
- Servicios de archivos digitales de exámenes médicos.
- Acceso a congresos y seminarios médicos en cualquier parte del mundo.

Servicios:

Herramienta de informática en salud para administración, evaluación y diagnóstico de enfermedades en forma remota, – Medical + Imaging Systems Application.

Esta aplicación se ha desarrollado por fases:

1. Administración de historias clínicas de pacientes, conteniendo:

- a. Información básica de cada paciente (datos personales, entidad de salud, etc.).
- b. Información médica de paciente (alergias, enfermedades, historial médico).

- c. Información de consulta (motivos de consulta, diagnóstico por consulta, etc.)
- d. Anexos a consulta (exámenes de laboratorio, rayos x, prescripción de medicamentos).
 - i. Integración con plataforma de administración de medicamentos/tratamientos disponibles por entidad de salud.

2. Integración con módulos de especialidades basadas en imágenes médicas: oftalmología, radiología, etc.).

- a. Módulos permiten captura de imágenes digitales, asociadas a los pacientes en la base de datos para crear un banco de imágenes por paciente y por especialidad.
- b. Imágenes pueden ser usadas para consulta médica o académica.

3. Plataforma de diagnósticos y recomendaciones clínicas:

- a. Generación automática de diagnósticos y tratamientos relacionados mediante algoritmos propietarios.

Esta aplicación permite la transmisión de la información del paciente y de las imágenes relacionadas en tiempo real; esta información podrá ser visualizada desde cualquier lugar del mundo para darle atención especializada inmediata.

Confidencialidad y Seguridad:

- De manera adicional, el uso de tecnología DICOM para complementar e interactuar con sistemas internacionales de manejo de imágenes médicas añade al producto un mayor nivel de seguridad de información confidencial de pacientes. DICOM es un estándar tecnológico de producción, despliegue, almacenaje y transferencia de imágenes médicas de alta resolución entre diferentes sistemas.
- El acceso al programa vía Internet se encuentra protegido por certificados de seguridad de la más alta calidad, el sistema cuenta con protección tanto en sus servidores de Internet como en sus bases de datos.

1.1.3.1. HL7

HL7 International (Health Level Seven) es una “Organización de Desarrollo de Estándares” (SDOs iniciales en inglés), para el ámbito de la salud. Fue fundada en 1987, Health Level Seven Internacional (HL7) es una organización sin fines de lucro, las normas ANSI-acreditada en desarrollo organización dedicada a proporcionar un marco general y las normas relacionadas para el intercambio, la integración, el intercambio y recuperación de información de salud electrónica que soporta la práctica clínica y la gestión, ejecución y evaluación de los servicios de salud. HL7 de 2300 + miembros incluyen a aproximadamente 500 miembros corporativos que representan más del 90% de los sistemas de información de proveedores, sirviendo de salud. (6)

1.1.4. **SAIMED**

Es la abreviación de Sistema de Almacenamiento de Imágenes Médicas, este servicio permite subir a los servidores, Documentos Digitales de Resultados de Laboratorio, Radiografías, Rayos X, Ecografías, Tomografías, Informes Médicos, etc. Mantenerlos por el tiempo que desee el cliente y de inmediato poder Visualizarlos por internet, desde cualquier parte tanto por el Médico, Paciente, Aseguradora e Institución Emisora, manteniendo así la información digitalizada y organizada.

El sistema ha sido desarrollado con los más avanzados componentes Web existentes en el mercado tales como PHP, Ajax, Jquery entre otros componentes que le proporcionaran un servicio intuitivo, fácil de usar, y con la velocidad, seguridad y confidencialidad que requiere.(7)

1.1.4.1. PHP

Es un lenguaje de programación con variables, sentencias, condicionales, bucles, funciones, entre otras. Se parece mucho a Java Script o a C, pero PHP se ejecuta en el servidor no en el navegador como Java Script, por eso permite acceder a los recursos que tenga el servidor por ejemplo la base datos. El resultado que es enviado al navegador después de ser ejecutado el programa PHP en el servidor es normalmente una página HTML o una página WML.

Para que las páginas PHP funcionen el servidor donde está alojado debe soportar PHP, pero no es necesario que su navegador lo soporte por ser un lenguaje que se ejecuta en el servidor. (8)

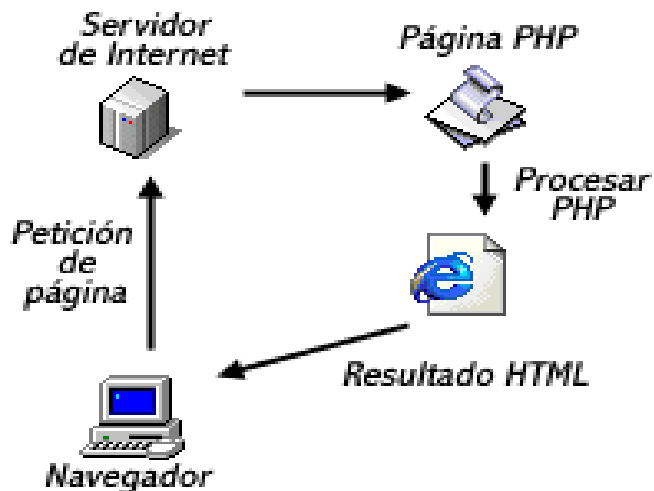


Fig.3 Funcionamiento de las páginas PHP.

1.1.4.2. Ajax

El termino Ajax se presentó por primera vez en el artículo publicado por James Garrett el 18 de febrero de 2005 titulado ``Ajax: A New Approach to Web Applications`` que vendría siendo Ajax: un nuevo acercamiento a aplicaciones web. El termino Ajax es un acrónimo de Asynchronous Java Script + XML, que se puede traducir como ``Java Script asíncrono + XML ``.

Ajax no es una tecnología sino la que se trata de varias tecnologías independientes que se unen en una sola de forma sorprendente.

Las tecnologías por las que está formada Ajax son:

- ✚ XHTML y CSS, para crear una representación 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.
- ✚ XML Http Request, para el intercambio asíncrono de información.
- ✚ Java Script, para unir todas las demás tecnologías.(9)

1.1.5. **SIENET**

La división Medical Solutions la cual es reconocida por sus innovadores productos, servicios y soluciones que abarcan desde equipamiento de diagnóstico y tratamiento, hasta soluciones en tecnología informática la cual optimiza el flujo de trabajo e incrementan la eficiencia de hospitales, clínicas y consultorios.

El grupo Siemens Medical Solutions no sólo es uno de los mayores y más diversificados fabricantes, proveedores de equipamiento médico y de soluciones, sino también uno de los más familiarizado con los procesos del cuidado de la salud. Esta línea de productos lleva el nombre SIENET. Todos estos equipos trabajan con un protocolo específico para imágenes médicas llamadas DICOM 3.0. Entre las actividades más importantes del conjunto de productos se encuentra La Distribución de Reportes e imágenes vía Web la cual lo realiza a través de un servidor de imágenes para su acceso vía Internet o red local. El usuario puede acceder desde cualquier lugar a la base de imágenes y reportes guardados, con una PC convencional y de acuerdo a los permisos adquiridos, pudiendo también realizar diagnósticos a la distancia con imágenes de menor resolución. etc. (10)

1.2. **Metodología de desarrollo.**

La metodología de desarrollo del software no es más que un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. La cual puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo.

Por tanto, las metodologías de desarrollo representan el camino para desarrollar aplicaciones informáticas de una manera sistemática. Hoy en día, existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Entre las más conocidas se tienen: Rational Unified Process (RUP), Extreme Programming (XP), Microsoft Solution Framework (MSF).

1.2.1. **Rational Unified Process (RUP)**

El proceso unificado conocido como RUP, es un modelo de software que permite el desarrollo de software a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Aunque con el inconveniente de generar mayor

complejidad en los controles de administración del mismo. Sin embargo, los beneficios obtenidos recompensan el esfuerzo invertido en este aspecto.

El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas.

Estructura del ciclo de vida del proceso de desarrollo unificado.

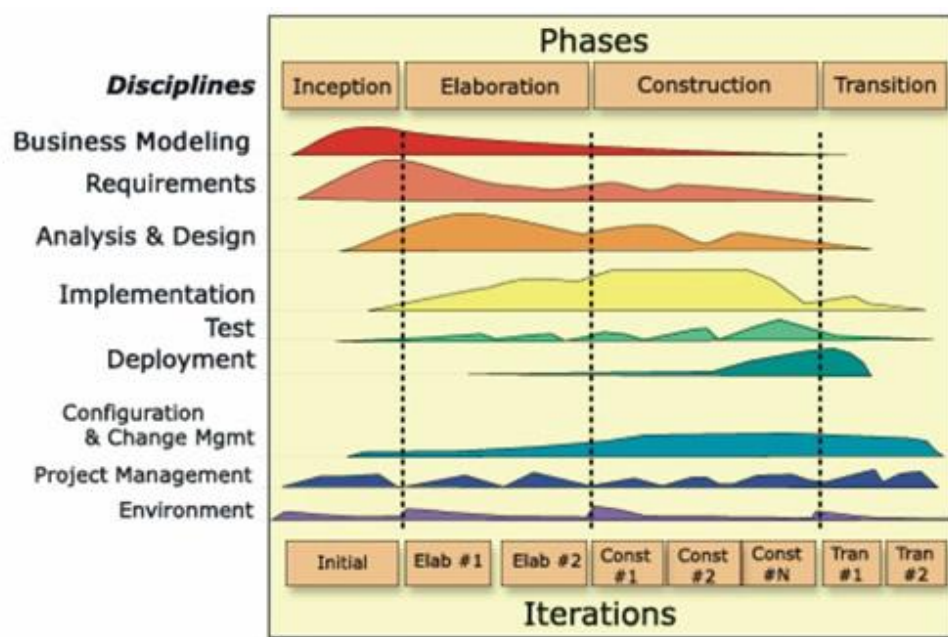


Fig. 4 Proceso de desarrollo unificado

Fase de concepción: Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos potenciales asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

Fase de elaboración: En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

Fase de construcción: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

Fase de transición: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. (11)

1.2.2. **Extreme Programming (XP)**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

La metodología se basa en las siguientes características de XP:

- Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- Re-fabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.(12)

1.2.3. **Microsoft Solution Framework (MSF)**

Es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características:

- Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.

- Escalable: puede organizar equipos tan pequeños entre 3 ó 4 personas, así como también, proyectos que requieren 50 personas a más.
- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.

Tecnología Agnóstica: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

Microsoft Solution Framework se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

- Modelo de Arquitectura del Proyecto: Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
- Modelo de Equipo: Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.
- Modelo de Proceso: Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.
- Modelo de Gestión del Riesgo: Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- Modelo de Diseño del Proceso: Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.

- **Modelo de Aplicación:** Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

Podemos concluir además, que lo más importante antes de elegir la metodología que usarás para la implementación de tu software, es determinar el alcance que tendrá y luego de ahí ver cuál es la que más se acomoda en tu aplicación.(12)

1.3. Herramientas.

Las herramientas de desarrollo son aquellos programas o aplicaciones que tengan cierta importancia en el desarrollo de un programa (programación). Pueden ser de importancia vital (como un ensamblador, un compilador o un editor) o de importancia secundaria, como una IDE (Entorno de Desarrollo Integrado). A continuación se describen las características principales de las herramientas Microsoft Project, Asp.net, Enterprise Architect y PostgreSQL utilizadas para la realización del trabajo de diploma.

1.3.1. **Enterprise Architect (EA)**

Es una herramienta flexible, completa y potente de modelado en UML bajo plataforma Windows. Provee lo más nuevo en desarrollo de sistemas, administración de proyectos y análisis de negocio. (13)

EA es una herramienta multi-usuario, con bases construidas sobre la especificación UML 2.0. Diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. Provee una generación poderosa de documentos y herramientas de reporte con un editor de plantilla completo WYSIWYG¹. Genera reportes detallados y complejos de EA con la información que se necesita en el formato que su compañía o cliente demanda. (14)

Una de las características más importantes es que EA soporta generación e ingeniería inversa de código fuente para muchos lenguajes conocidos, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP.

1- **WYSIWYG:** **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et ("lo que ves es lo que obtienes"). Se aplica a los editores de texto que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

También es posible vincularla con Visual Studio.NET, permitiéndole modelar en EA y saltar directamente al código fuente en su editor de .NET. Se decide utilizar Enterprise Architect como herramienta CASE profesional, ya que soporta el ciclo de vida completo, combina la potencia de la última especificación UML 2.1 con un alto rendimiento y una interfaz intuitiva. Además brinda características como: la generación de código en visual C# 2.0; integración con Visual Studio; soporte para pruebas; usabilidad; velocidad; y otras.

1.3.2. **Cliente de Subversión TortoiseSVN1.6.6**

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. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, como una novela, o el código fuente de un programa.

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. Está desarrollado bajo la Licencia Pública General GNU (GPL). TortoiseSVN se ejecuta en Windows 2000 SP2, Windows XP o superiores. (15)

1.3.3. **PostgreSQL**

El RSGBD Orientado a Objetos PostgreSQL, comenzó su desarrollo sobre los años 80. Es un sistema de gestión de bases de datos objeto relacional, basado en el proyecto POSTGRES. El gestor se encuentra bajo una licencia BSD y es dirigido por una comunidad mundial de desarrolladores conocida como Grupo Mundial de Desarrollo de PostgreSQL, y está basado en contribuciones de proveedores comerciales, empresas a portantes y programadores de código abierto mayormente. (PGDG, por sus siglas en inglés). Entre sus principales características se encuentra su destreza para manejar altos niveles de concurrencia mediante un sistema llamado Acceso Concurrente Multiversión (MVCC, por sus siglas en inglés). Posee una amplia variedad de tipos nativos y permite al usuario crear sus propios tipos de datos.

Es multiplataforma y además soporta procedimientos almacenados y triggers, los cuales pueden ser escritos usando lenguajes como PL/PgSQL, C++, Java PL/Java web, PL/Python y plPHP. (16)

1.3.4. **Microsoft Project**

Microsoft Project (o MSP) es un Software de administración de proyectos desarrollado y vendido por Microsoft el cual está creado para asistir a los administradores de proyectos. La primera versión de Microsoft Project fue lanzada para DOS en 1984 por una compañía que trabajaba para Microsoft. Microsoft adquirió todos los derechos del software en 1985 y liberó la versión 2. La versión 3 para DOS fue liberada en 1986. La versión 4 para DOS fue la última versión para este sistema operativo, liberada en 1987. La primera versión para Windows fue liberada en 1990, y fue llamada versión 1 para Windows. Un dato interesante es que la primera versión para DOS introdujo el concepto de Líneas de dependencia (link lines) entre tareas en la gráfica de Gantt. Aunque este software ha sido etiquetado como miembro de la familia Microsoft Office hasta el momento no ha sido incluido en ninguna de las ediciones de Office. Está disponible en dos versiones, Standard y Professional.

Una versión para Macintosh fue liberada en julio de 1991 y su desarrollo continuó hasta Project 4.0 para Mac en 1993. En 1994, Microsoft detuvo el desarrollo para la mayoría de las aplicaciones Mac, y no ofreció nuevas versiones de Office hasta 1998, después de la creación del nuevo Microsoft Macintosh Business Unit el año anterior. El MacBU nunca lanzó una versión actualizada para Project, y la versión anterior de 1993 no es ejecutada nativamente en Mac OS X.

Las versiones fueron lanzadas en 1992 (v3), 1993 (v4), 1995, 1998, 2000, 2002, 2003 y 2007.1

La aplicación crea calendarización de rutas críticas, además de cadenas críticas y metodología de eventos en cadena disponibles como add-ons de terceros. Los calendarios pueden ser resource leveled, y las gráficas visualizadas en una Gráfica de Gantt. Adicionalmente, Project puede reconocer diferentes clases de usuarios, los cuales pueden contar con distintos niveles de acceso a proyectos, vistas y otros datos. Los objetos personalizables como calendarios, vistas, tablas, filtros y campos, son almacenados en un servidor que comparte la información a todos los usuarios.

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

El software Microsoft Office Project en todas sus versiones (la versión 2010 es la más reciente) es útil para la gestión de proyectos, aplicando procedimientos descritos en el PMBoK (Management Body of Knowledge) del PMI (Project Management Institute). (17)

1.3.5. *Asp.Net*

El conjunto de clases del .NET Framework es la piedra angular de cualquier desarrollador de .NET, es un rico conjunto de clases, interfaces, tipos que simplifican y optimizan el desarrollo de aplicaciones .NET además de proporcionar acceso a la funcionalidad del sistema. Como desarrolladores el dominio de este conjunto de clases es vital para un buen desarrollo en .NET.

Luego de varios años, cambios y avances en las tecnologías. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). Este está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

Esta se ha creado bajo varios principios como son facilidad de desarrollo el cual da la posibilidad de elegir el lenguaje de programación con el que se desee trabajar, por defecto lleva integrado C#, VB.Net y J#, pero se pueden usar otros lenguajes. Tiene gran independencia de la herramienta de desarrollo y cuenta con una biblioteca de clases que lleva incorporada.

El alto rendimiento y escalabilidad, una mejorada fiabilidad con la cual ASP.NET es capaz de detectar pérdidas de memoria, problemas con bloques y protección ante caídas. Entre otras cosas, es capaz de detectar aplicaciones web que pierden memoria, arrancando otro proceso limpio con una nueva instancia de la aplicación para cerrar la que pierde memoria liberando así la memoria perdida.

También tendrá como principio la fácil distribución e instalación y esto permite que la aplicación ASP.NET se instale tan fácilmente como copiando los ficheros que la componen. No es necesario registrar ningún componente, tan solo copiar los ficheros al web. Se puede recopilar la aplicación o enviar nuevos ficheros sin necesidad de reiniciar la aplicación ni el servidor web. (18)

Cualquier persona que está familiarizada con el desarrollo de aplicaciones web sabrá que el desarrollo web no es una tarea simple. Ya que mientras que un modelo de programación para aplicaciones de uso

común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación web es una mezcla de varios lenguajes de etiquetas, un gran uso de lenguajes de script y plataformas de servidor.

1.4. Resultados esperados

Se espera la implementación de los servicios web que posibiliten la administración remota de alas PACSServer.

El control de tráfico de conexiones en el servidor de imágenes, se podrá visualizar la cantidad de estaciones y/o equipos conectados en cada instante de tiempo. Se tendrá también la posibilidad de ver los reportes de tráfico de una fecha o rango de fecha previamente seleccionada. Dichas funcionalidades serán de gran utilidad para diagnosticar y anticiparse a posibles problemas en la transmisión de las imágenes, identificar los picos en el tráfico y planificar posibles acciones para bajar la carga en la red en estos horarios y configurar de manera rápida y distribuida.

La utilización de los servicios implementados agilizará los procesos de despliegue y soporte del sistema alas PACS y facilitará la administración del servidor de imágenes y el control de tráfico de las imágenes.

En este capítulo se realizó un análisis del estado del arte y de las tendencias actuales en el desarrollo del los servicios para la administración remota de los sistemas de procesamiento de imágenes médicas. Se expusieron las herramientas y tecnologías a utilizar para el desarrollo del sistema. Además, la información recopilada sobre los antecedentes existentes en el mundo constituye la base para la toma de decisiones respecto a la manera de estructurar la solución.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Para un correcto comienzo del diseño de un sistema es necesario aclarar todos los conceptos y sus relaciones. Se define la visión general del proyecto y su alcance en iteraciones. Se muestra los procesos del negocio mediante el modelo de dominio, las características y funcionalidades que tendrá el sistema a partir de los requerimientos funcionales y no funcionales del componente de software a desarrollar. Se presenta el diagrama de casos de uso del sistema con las correspondientes especificaciones de los casos de uso asociados al componente.

2.1. Breve descripción del sistema.

Observando las funcionalidades existentes en el sistema creado en la Universidad de Ciencias Informáticas, Cuba, alas PACSServer para dar solución al problema existente en el país siendo este un sistema menos costoso y fácil de administrar y mantener. El cual brinda servicios a diversos clientes, gestiona la transmisión y el almacenamiento de imágenes médicas y garantizar la disponibilidad de las imágenes entre otras actividades.

El servicio a implementar es una herramienta para la administración remota de la configuración del sistema alas PACSServer y de los servicios que ofrece, el cual debe: permitir la configuración de manera remota de los datos para la comunicación del servidor de imágenes con cada una de las estaciones clientes asociadas al mismo. La visualización de forma gráfica de la información relacionada con la conexión de las estaciones clientes y/o equipos médicos al servidor y visualizar en tiempo real el control de tráfico de la conexiones del mismo.

2.2. Modelo del dominio.

La metodología de desarrollo RUP define en su primera fase la realización de un modelo de negocio, con el fin de comprender el entorno del cliente y detectar las mejoras existentes en los procesos de organización. Cuando el proceso de negocio no es posible identificarlo claramente, RUP propone realizar un modelo de dominio, el cual es un subconjunto del modelo de negocio. El modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los conceptos que existen o eventos que ocurren en el entorno en que trabaja el sistema. RUP es el resultado

de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del Lenguaje Unificado de Modelado (UML, siglas en inglés).

Teniendo en cuenta que los procesos de negocio no están claramente definidos, se describe el negocio a través del modelo de dominio.

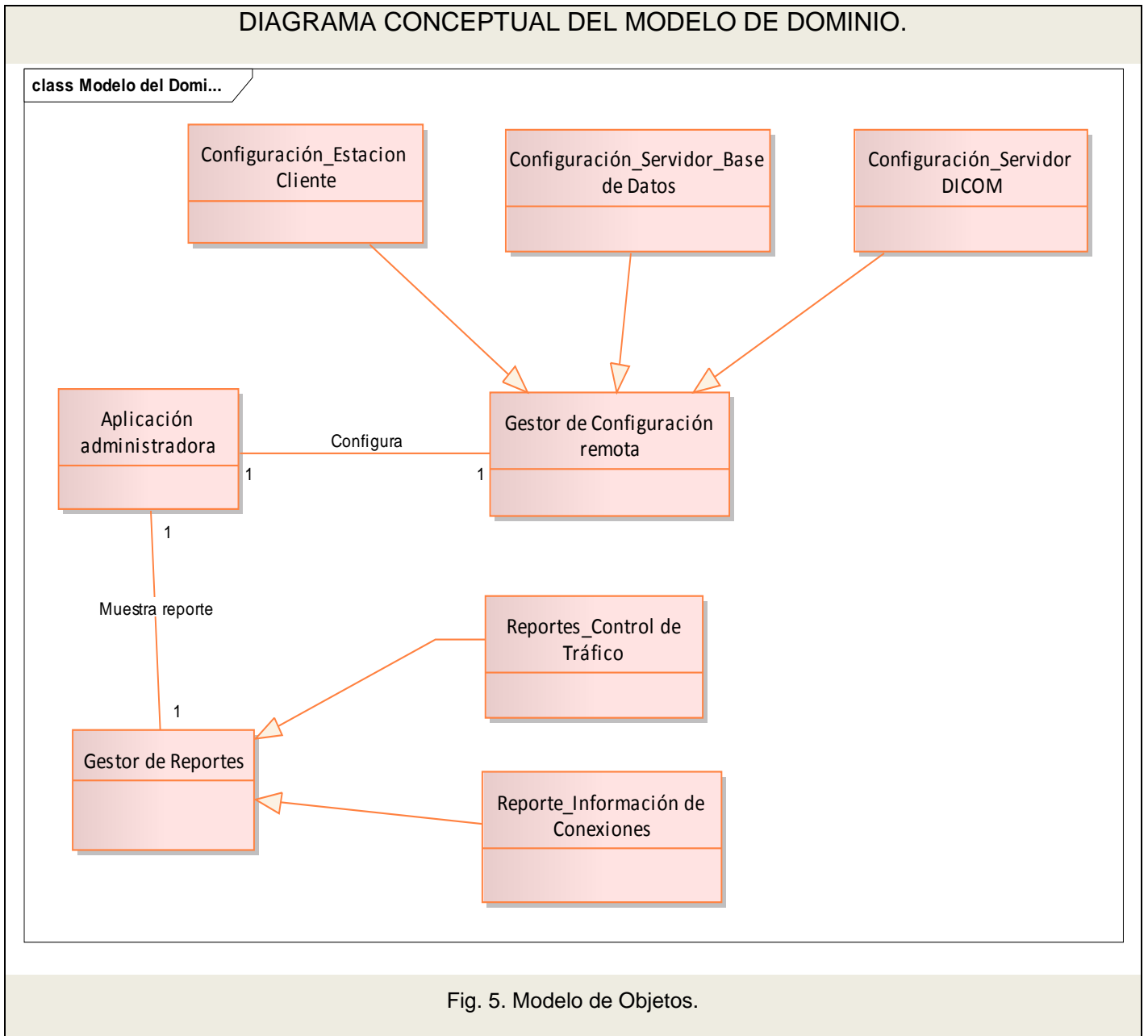


Fig. 5. Modelo de Objetos.

2.2.1. **Conceptos fundamentales**

Reportes_Control de Tráfico: Visualización en tiempo real del control de tráfico de las conexiones al servidor.

Reportes_Información de Conexiones: Información relacionada con las conexiones de las estaciones clientes y/o equipos médicos al servidor, para ser mostrada de forma gráfica, con el fin de facilitar su interpretación.

Gestor de Reportes: Aplicación que gestiona la información de los reportes emitidos por el servidor.

Gestor de Configuración remota: Aplicación que gestiona la configuración del sistema alas PACSServer de manera remota, usando servicios web, brindando la posibilidad al administrador del servidor a acceder a la configuración del mismo desde cualquier estación cliente conectada a la red para de esta manera hacer los cambios pertinentes sin tener que ir hasta la posición física del servidor.

Configuración_Estación Cliente: Se gestiona la configuración de las estaciones clientes que se conectan al servidor para obtener información. Cuando se necesite agregar un nuevo cliente, se deberá especificar la *Entidad de Aplicación del cliente (EA Cliente)*, la *Dirección IP* del mismo, y el *Puerto* por donde dicho cliente recibirá los mensajes que le envíe el servidor. De esta forma se podrán *agregar, modificar, o eliminar* clientes a este listado.

Configuración_Servidor_Base de Datos: Se gestiona la configuración de la conexión del sistema con la base de datos, para esto se deberá especificar la *Dirección IP* donde se encuentra el servidor que contiene la Base de Datos con la información referente a los ficheros DICOM, el nombre de la *Base de Datos* anteriormente mencionada, así como el *Usuario* y la *Contraseña* requeridos para la autenticación en la Base de Datos.

Configuración_Servidor DICOM: Se gestiona la configuración del servidor DICOM para esto se especifica el nombre de la *Entidad de Aplicación* que identificará al Servidor en la red imagenológica local, así como para comunicarse con los otros componentes de alas PACS. El Puerto por donde el servidor estará a la escucha de las peticiones de los clientes conforme al estándar DICOM 3.0, puede estar en el rango 104 -109 (se recomienda seleccionar el puerto 104). Por último, la *Bandeja de entrada*, que será el directorio donde se almacenarán todas las imágenes recibidas, tanto de los equipos médicos como de los clientes (alas PACSViewer).

2.3. Especificación de los requisitos del software.

2.3.1. *Requisitos Funcionales*

RF1 Configurar Servidor DICOM.

RF2 Configurar Estaciones Clientes.

RF2.1 Agregar un cliente

RF2.2 Eliminar un cliente.

RF2.3 Modificar información de un cliente.

RF3 Configurar Servidor de Base de Datos.

RF4. Salvar la configuración del sistema.

RF5 Visualizar información de conexiones de forma gráfica.

RF5.1. Permitir la visualización.

RF6 Visualizar el control del tráfico de conexiones.

RF6.1. Permitir visualizar en tiempo real.

2.3.2. *Requisitos No Funcionales*

- **Usabilidad**

RNU 1. Permitir acceder y hacer cambios a la configuración del sistema desde cualquier lugar, vía web.

- **Seguridad**

RNS 1. La conexión al servidor se realizará solo desde el IP donde se encuentra la aplicación administradora.

- **Funcionamiento**

RNFO 1. La configuración debe poderse preservar de una instalación a otra.

RNFO 2. Debe existir un registro y control de errores detallados.

RNFO 3. El puerto de escucha debe ser el 104 por defecto.

RNFO 4. NET Framework 3.5, o superior. (Software)

RNFO 5. Utilizar un motor de base de datos PostgreSQL 8.2. (Software)

RNFO 6. Debe existir una aplicación web que consuma los servicios web creados.

RNFO 7. Debe ejecutarse el software sobre la plataforma .NET.

- **Estándares aplicables**

RNEA 1. Los datos de configuración entrados mediante la aplicación web deben ser compatibles con los de la aplicación desktop del sistema.

- **Soporte**

RNSO 1. Siempre debe haber una forma de restaurar la configuración en caso de existir algún problema con esta, con el objetivo de mitigar la crisis.

- **Disponibilidad**

RND 1. Los reportes de control de tráfico de las conexiones al sistema y de las conexiones de las estaciones clientes y/o equipos médicos al servidor, deben estar disponibles en cualquier momento.

- **Diseño e Implementación**

RNDI 1. El análisis y diseño de la aplicación será basado en la metodología RUP con el uso del lenguaje de modelado UML.

RNDI 2. Se usará como herramienta CASE, Enterprise Architect para el modelado de los artefactos que se generan en cada uno de los flujos de trabajo.

RNDI 3. Se usará como metodología de programación ASP.NET.

RNDI 4. Se utilizará como IDE de desarrollo Visual Studio 2008.

2.4. Definición de los casos de uso del sistema.

2.4.1. *Actores del sistema*

Aplicación administradora	Es la persona que interactúa con el sistema. Vela por el apropiado funcionamiento del servidor y lo configura.
---------------------------	--

2.4.2. **Casos de Uso**

CU1	Configurar Servidor DICOM
Actor	Aplicación administradora
Descripción:	La persona autorizada puede acceder a la configuración del Servidor DICOM, modificarlo y salvar los cambios. Se guarda el evento sucedido.
Referencia	RF1; RF4.

CU2	Gestionar Estaciones Clientes
Actor	Aplicación administradora
Descripción:	La persona autorizada puede acceder a la configuración de la Estación Cliente, modificarla y salvar los cambios. Se guarda el evento sucedido.
Referencia	RF2; RF2.1; RF2.2; RF23; RF4.

CU3	Configurar el Servidor de base de datos
-----	---

Actor	Aplicación administradora
Descripción:	La persona autorizada puede acceder a la configuración del Servidor de bases de dato, modificarlo y salvar los cambios. Se guarda el evento sucedido.
Referencia	RF3; RF4.

CU4	Visualizar información de conexiones de forma gráfica.
Actor	Aplicación administradora
Descripción:	La persona autorizada puede observar la información relacionada con las conexiones de las estaciones clientes y/o equipos médicos en una forma gráfica.
Referencia	RF5; RF5.1.

CU5	Visualizar el control del tráfico de conexiones.
Actor	Aplicación administradora
Descripción:	La persona autorizada puede visualizar en tiempo real el control de tráfico de las conexiones al mismo.
Referencia	RF6; RF6.1.

2.4.3. Diagrama de Casos de Uso

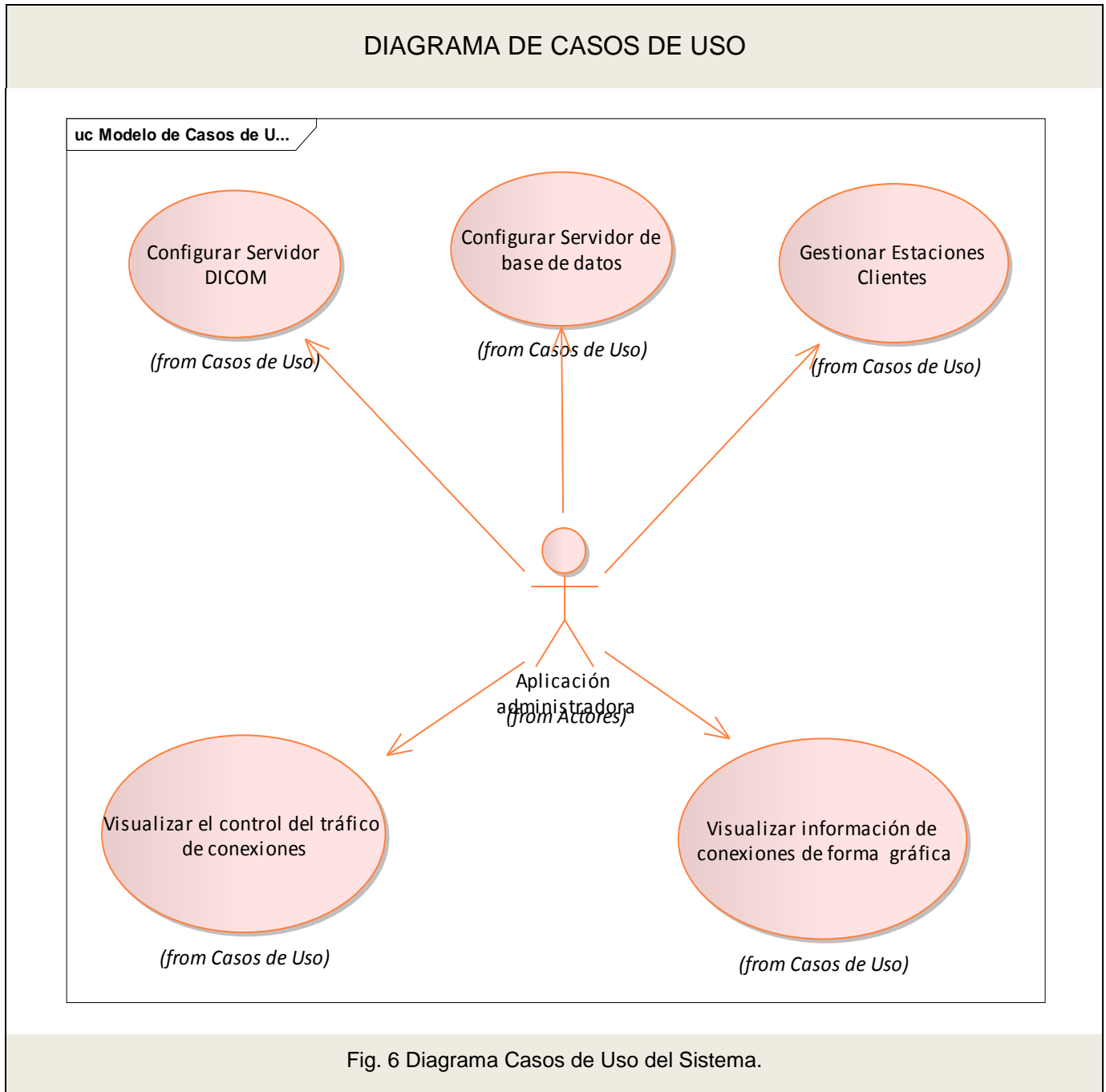


Fig. 6 Diagrama Casos de Uso del Sistema.

2.4.4. *Descripción extendida de los Casos de Uso*

CU # 1. Configurar Servidor DICOM	
Objetivo	La persona autorizada desea configurar el Servidor de DICOM
Actores	Aplicación administradora: (Inicia) Adiciona, modifica, elimina, ve y valida los datos del Servidor DICOM.
Resumen	La persona autorizada puede acceder a la configuración del Servidor DICOM, adicionarlo o modificarlo y salvar los cambios. Se guarda el evento sucedido.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF1; RF4.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. La Aplicación administradora introduce la entidad de aplicación del servidor. 2. El Aplicación administradora introduce el puerto de escucha del servidor. 3. El Aplicación administradora introduce la dirección del repositorio de imágenes. 	

CU # 2. Gestionar Estaciones Clientes	
Objetivo	La persona autorizada adiciona, elimine o modifique una Estación Cliente
Actores	Aplicación administradora: (Inicia) Adiciona, modifica, elimina, ve y valida los datos de la Estación Cliente
Resumen	La persona autorizada puede acceder a la configuración de la Estación Cliente, adicionar, eliminar o modificarla y salvar los cambios. Se guarda el evento sucedido.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF2; RF2.1; RF2.2; RF2.3; RF4.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El Caso de Uso inicia cuando la Aplicación administradora procede a modificar la configuración de la Estación Cliente. 2. El Sistema muestra los datos de la configuración de la Estación Cliente. 3. La Aplicación administradora procede a: <ul style="list-style-type: none"> Adicionar. Ver Evento 1. Eliminar. Ver Evento 2. 	

Modificar. Ver evento 3.

4. La **Aplicación administradora** procede a salvar la configuración.
5. El Sistema salva los datos de la configuración.
6. El Sistema actualiza todo los datos del sistema con la nueva configuración.
7. El Sistema reporta el suceso.

Evento 1. Adicionar

1. La **Aplicación administradora** selecciona adicionar una nueva estación.
2. El Sistema solicita los datos de la nueva estación.
3. La **Aplicación administradora** introduce la entidad de aplicación.
4. La **Aplicación administradora** introduce la dirección IP del cliente.
5. La **Aplicación administradora** introduce el puerto de escucha del cliente.
6. La **Aplicación administradora** adiciona la nueva estación.
7. El Sistema agrega la nueva estación a la lista de estaciones autorizadas.

Evento 2. Eliminar

1. La **Aplicación administradora** selecciona eliminar una estación.
2. El Sistema solicita los datos de la estación.
3. La **Aplicación administradora** introduce la entidad de aplicación.
4. La **Aplicación administradora** introduce la dirección IP del cliente.
5. La **Aplicación administradora** introduce el puerto de escucha del cliente.
6. La **Aplicación administradora** elimina la estación.
7. El Sistema elimina la estación de la lista de estaciones autorizadas.

Evento 3. Modificar

1. La **Aplicación administradora** selecciona modificar una estación.
2. El Sistema solicita los datos de la estación.
3. La **Aplicación administradora** introduce la entidad de aplicación.
4. La **Aplicación administradora** introduce la dirección IP del cliente.
5. La **Aplicación administradora** introduce el puerto de escucha del cliente.
6. La **Aplicación administradora** modifica la estación.
7. El Sistema modifica la estación a la lista de estaciones autorizadas.

Flujo Alternativo de los eventos

Línea 1: a) La persona autorizada selecciona modificar la configuración de una estación determinada: Ejecuta los pasos del 3 al 6 y el sistema actualiza la estación a la lista de estaciones autorizadas.

b) La persona autorizada selecciona eliminar una estación determinada y el sistema retira la estación de la lista de estaciones autorizadas.

CU # 3. Configurar Servidor de base de datos

Objetivo	La persona autorizada desea configurar el Servidor de base de datos.
Actores	Aplicación administradora: (Inicia) Adiciona, modifica ve y valida los datos del Servidor de la base de datos
Resumen	La persona autorizada puede acceder a la configuración del Servidor de bases de dato, modificarlo y salvar los cambios. Se guarda el evento sucedido.

Complejidad	Alta
Prioridad	Crítico
Referencias	RF3; RF4.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. La Aplicación administradora introduce la dirección IP del servidor de base de datos. 2. La Aplicación administradora introduce el nombre de la base de datos. 3. La Aplicación administradora introduce en nombre del usuario de la base de datos. 4. La Aplicación administradora introduce la contraseña del usuario de la base de datos. 5. La Aplicación administradora comprueba la conexión de la base de datos. 6. El Sistema verifica la conexión de la base de datos. 7. El sistema notifica el resultado de la verificación. 	
Flujo Alternativo de los eventos	
<p>Línea 2: Si la comprobación no es satisfactoria entonces se repiten los pasos del 1 al 5.</p>	

CU # 4. Visualizar información de conexiones de forma gráfica

Objetivo	Permitir a la persona autorizada visualizar la información de una forma gráfica.
Actores	Aplicación administradora: (Inicia) Va a la opción visualizar y ve de forma gráfica la información de conexiones.
Resumen	La información relacionada con las conexiones de las estaciones clientes y/o equipos médicos se visualizaran de forma gráfica logrando que su interpretación sea menos compleja.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF5; RF5.1.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. Este Caso de Uso se inicia cuando la persona autorizada selecciona la opción de visualizar la información, obtenida hasta el momento, de forma gráfica, de las conexiones de los equipos médicos y/o estaciones clientes. 2. El sistema muestra la gráfica de la información relacionada con las conexiones de las estaciones clientes y/o equipos médicos. 	

CU # 5. Visualizar el control del tráfico de conexiones	
Objetivo	Permitir a la persona autorizada visualizar el tráfico de las conexiones al servidor.
Actores	Aplicación administradora: (Inicia) Va a la opción visualizar y ve en tiempo real el tráfico de conexiones.
Resumen	La persona autorizada tiene la posibilidad de visualizar en tiempo real el control de tráfico de las conexiones al servidor.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF6; RF6.1.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. Este Caso de Uso se inicia cuando la Aplicación administradora selecciona la opción de visualizar el control del tráfico de las conexiones. 2. El Sistema muestra el tráfico existente hacia las conexiones del servidor en ese momento. 	

El desarrollo de este capítulo ha permitido alcanzar un mejor entendimiento del sistema y las restricciones que deben existir para satisfacer las necesidades del cliente. Se especificaron todos los requisitos funcionales y no funcionales del sistema, se identificaron los actores que intervienen. Así como todos los casos de uso, que fueron descritos de forma detallada y reflejan las funcionalidades recogidas previamente en los requisitos. El desarrollo de este flujo de trabajo y los artefactos obtenidos a partir de él permiten comenzar con el Flujo de Trabajo de Diseño, que requiere de mayor esfuerzo por parte del equipo de desarrollo para la construcción de la solución de software propuesta.

CAPÍTULO 3. **ARQUITECTURA Y DISEÑO DEL SISTEMA**

En este capítulo se definen las clases de diseño del software, así como sus atributos y responsabilidades. Se expone la arquitectura que tiene el sistema para que se cumplan sus requerimientos funcionales y no funcionales. Se podrán observar los distintos diagramas de clases de diseño y secuencia desarrollados así como los modelos de datos.

3.1. **Descripción de la Arquitectura.**

Un sistema software es una única entidad, pero al arquitecto y a los desarrolladores les resulta útil presentar el sistema desde diferentes perspectivas para comprender mejor el diseño. Mediante estas perspectivas se obtienen las vistas del modelo del sistema. Todas las vistas juntas representan la arquitectura.

El estudio del modelo arquitectónico implica una debida investigación, unido a una guía de diversos puntos que ayudarán a darle forma a la idea que se quiera expresar o bien a la conclusión que se quiera llegar. Se debe analizar desde lo más general hasta lo particular para así lograr con un criterio propio el objetivo determinado.

Una forma de expresar los estilos arquitectónicos es mediante los componentes o elementos que forman el sistema y sus interfaces; las conexiones que existen entre estos elementos ya sean dispositivos de redes, protocolos de comunicación u otros; las restricciones tanto de comportamiento o de características, y finalmente la configuración que rige la unión de los elementos, las cuales son llamadas las cuatro C, (Components, Connections, Communications and Constraints por sus siglas en inglés).

Sin embargo la arquitectura de software está afectada no solo por la estructura y el comportamiento, sino también por el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones y compromisos económicos, tecnológicos y la estética.

3.1.1. ***Patrón arquitectónico propuesto: Arquitectura en capas***

Este patrón simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las

superiores. Brinda modularidad a las aplicaciones, ayuda a identificar qué puede reutilizarse y proporciona una estructura que posibilita tomar decisiones sobre qué partes comprar y qué partes construir.

Se propone una arquitectura en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido. La primera capa se denomina capa de presentación y consiste en las interfaces gráficas de usuario. La capa intermedia, o capa de negocio, consiste en la aplicación o lógica de negocio que encapsula las principales funcionalidades, y la tercera capa, la capa de acceso a datos, contiene las operaciones necesarias para el acceso y la manipulación de la información que se almacena. Este modelo brinda la ventaja de que el desarrollo se pueda llevar a cabo en varios niveles de manera simultánea y, en caso de que se detecte algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Internamente el sistema está distribuido por las siguientes capas, una capa de datos que contiene todos los elementos persistentes, como es la Base de Datos. También existe una capa de servicios donde están agrupadas otras diferentes subcapas como son las de Acceso a Datos (CAD), de Configuración y Control y finalmente la subcapa del registro de errores de los eventos.

Servicio	Escritor/Lector de Logs.	Control de Configuración.	Control de Tráfico. Información de conexiones.
Datos	Logs.	Base de Datos.	Conexión de área local.

Fig. 7 Distribución de capas del sistema.

3.2. Modelo de Diseño

El propósito del diseño es especificar una solución y construir una arquitectura simple y fácilmente extensible. Este flujo de trabajo tendrá como objetivo traducir los requisitos a una especificación que describe cómo implementar el sistema. Debe ser suficientemente específico para que el sistema pueda ser implementado sin ambigüedades.

Sus objetivos fundamentales son:

1. Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones.
2. Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes.
3. Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
4. Ser capaces de visualizar y reflexionar sobre el diseño utilizando una notación común.
5. Crear una abstracción sin costuras de la implementación del sistema, en el sentido de que la implementación es un refinamiento directo del diseño que rellena lo existente sin cambiar la estructura. Esto permite la utilización de tecnologías como la generación de código y la ingeniería de ida y vuelta entre el diseño y la implementación. (Ver Anexo 1).

3.3. Descripción de la Clase de Diseño

Para entender mejor los detalles de las clases de diseño se pueden ver las siguientes descripciones de las principales clases del sistema (Ver Anexo 2).

3.4. Diseño de la Base de Datos

Diagrama Entidad Relación.

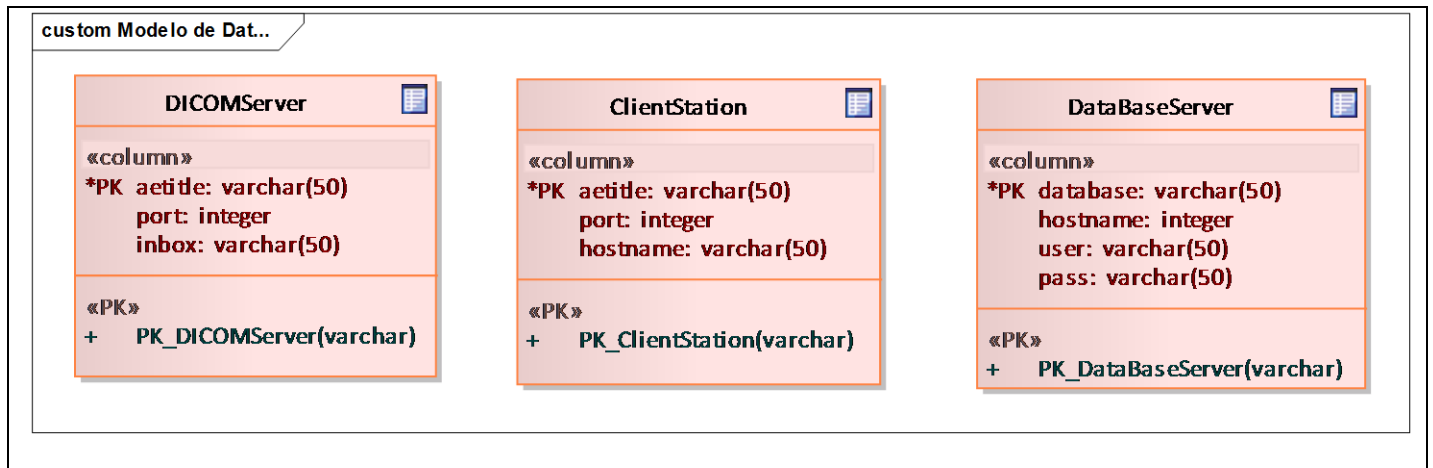


Fig. 8 Diagrama de Entidad relación.

Descripción de las tablas de la Base de Datos (Ver Anexo 3).

En este capítulo se presentaron los diagramas de clases del diseño y diagramas de secuencia de los casos, especificando los flujos de eventos e interacciones entre las clases y se explicaron brevemente las clases empleadas. Se definió la arquitectura de la aplicación y se expusieron las características de la misma.

CAPÍTULO 4. IMPLEMENTACIÓN

En este capítulo quedan expuestos los componentes del sistema, así como la forma en que estos interactúan entre sí. Se presenta el diagrama de componentes que muestra la distribución física de los componentes para la implementación y el diagrama de despliegue. Se realizan pruebas internas.

4.1. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Un componente es una parte modular del sistema, desplegable y reemplazable, contiene clases y puede ser implementado por uno o más artefactos (librerías, ficheros ejecutables, etc.). Es un elemento de implementación que representa algo físico, ya sea ficheros o archivos, y son creados para poner el código, ya sea código fuente, código binario o código ejecutable. En un componente se pueden implementar varios elementos, ejemplo las clases.

La Figura 11 muestra el diagrama de componentes del Servicio para la administración remota del sistema alas PACSServer.

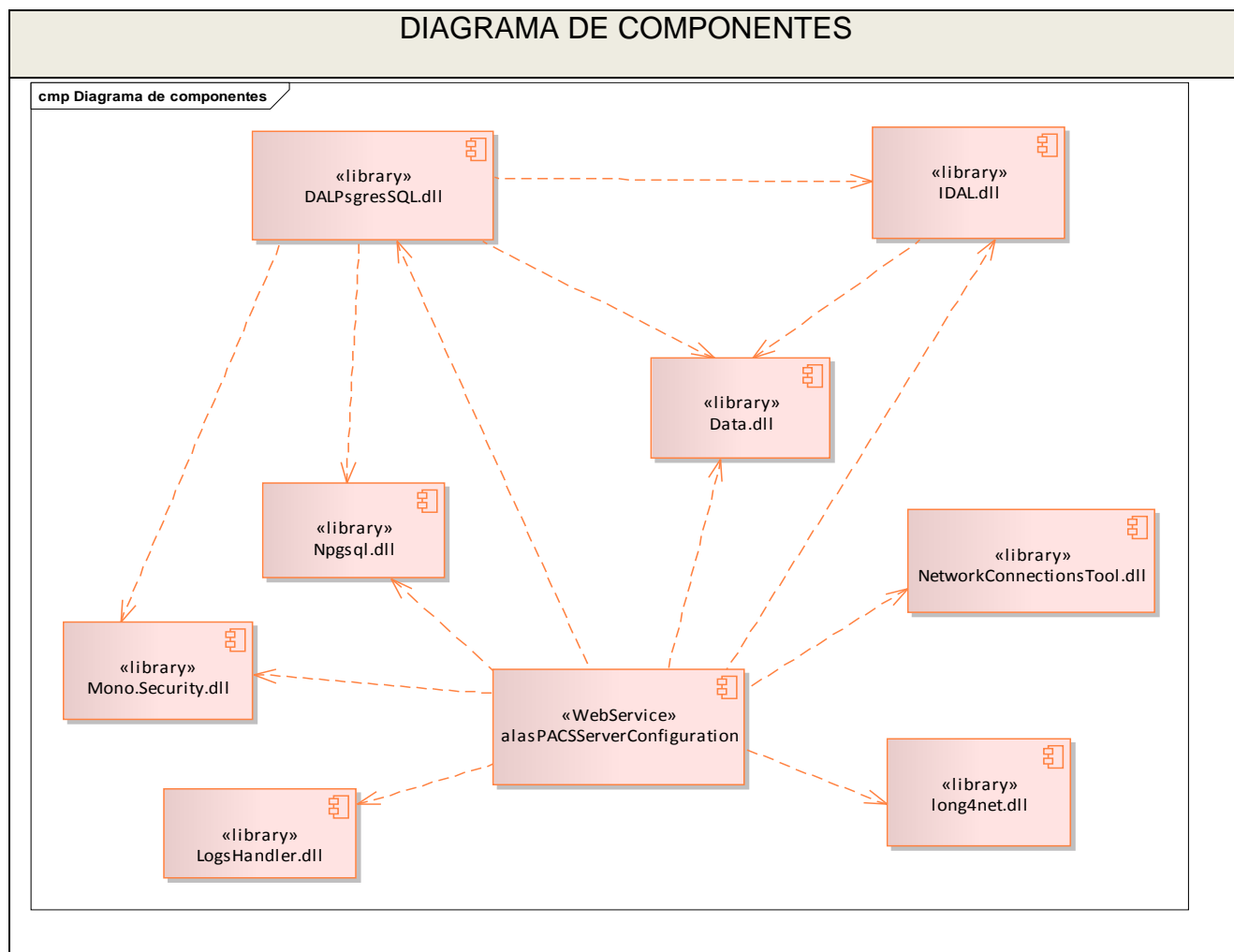


Fig. 9 Diagrama de Componentes.

4.2. Diagrama de despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación, que muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. En general un nodo será una unidad de computación de algún tipo. La arquitectura física del sistema estará compuesta por estaciones clientes, un servidor web y un servidor de base de datos.

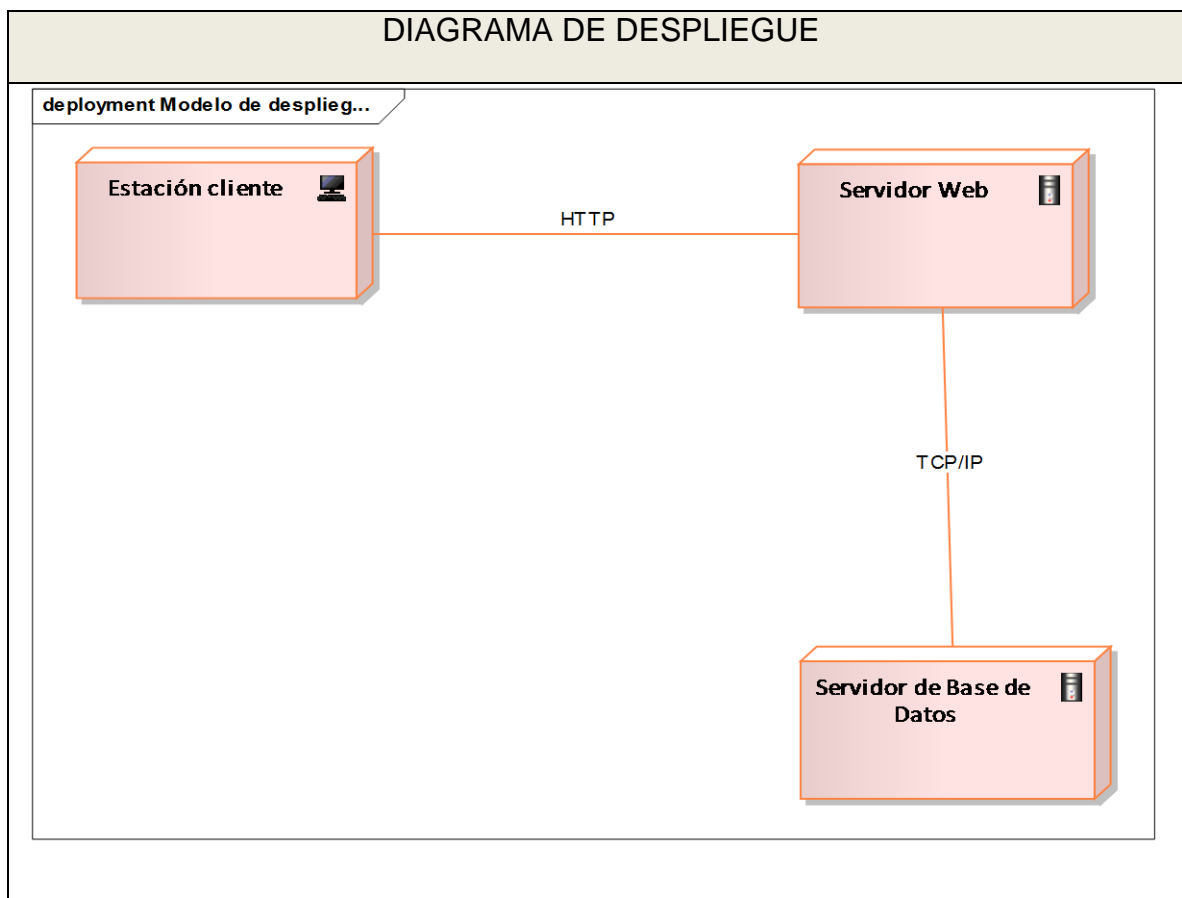


Fig. 10 Diagrama de Despliegue.

4.3. Compatibilidad con la plataforma libre

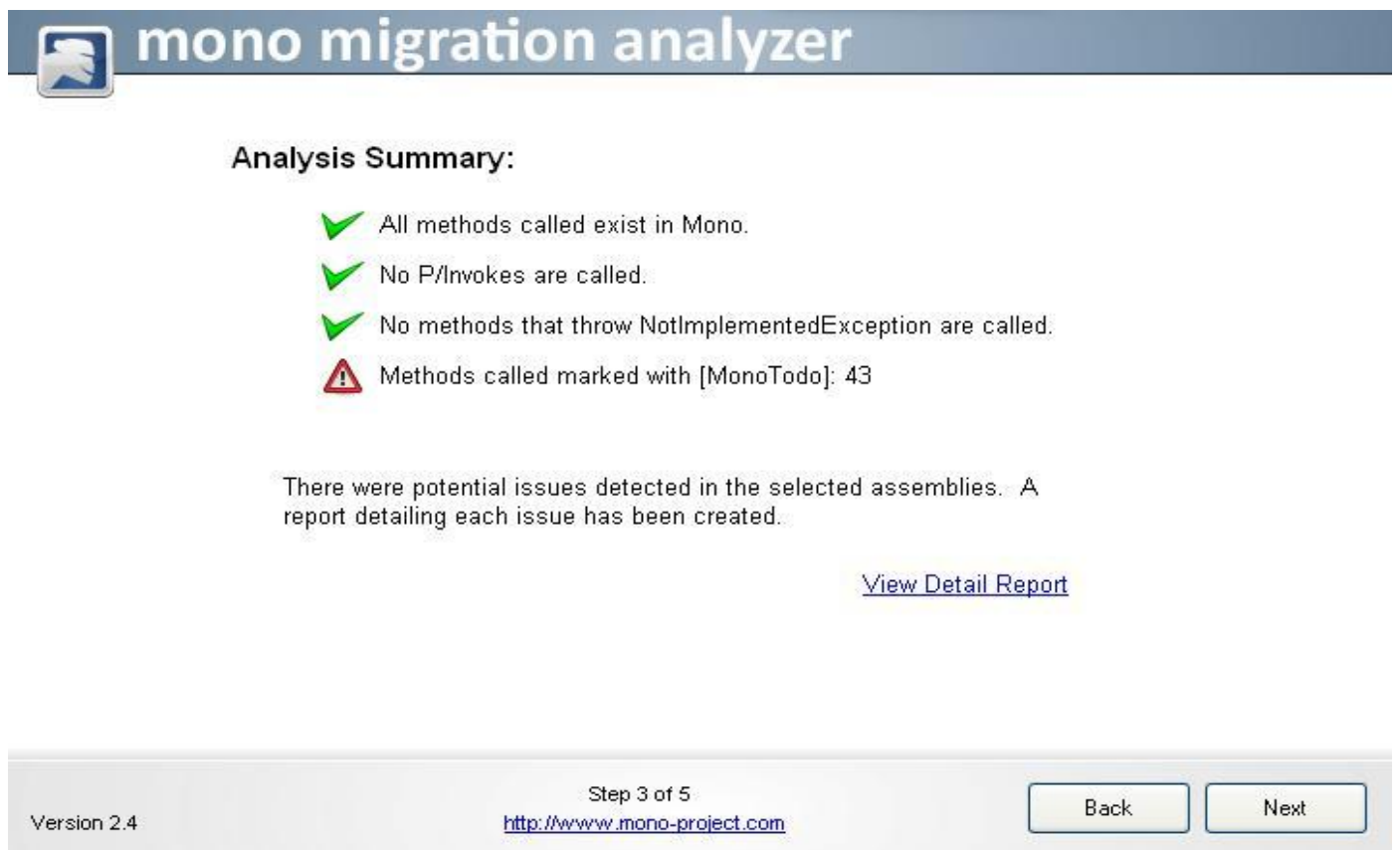
MoMA (Mono Migration Analyzer por su acrónimo en inglés) es una herramienta que ayuda a identificar los problemas que se les puede presentar a los desarrolladores al intentar migrar una aplicación desarrollada en .Net a Mono. Ayuda a identificar los métodos o funciones dependientes de la plataforma .Net que aun no son apoyadas por el proyecto Mono.

Mientras que el MoMA puede ayudar a mostrar los posibles problemas, hay muchos factores complejos que no pueden ser objeto de una sencilla herramienta por lo que puede fallar al señalar las esferas en que se presentarán dificultades, y puede señalar las áreas que en realidad no será un problema.

Mono puede ejecutar aplicaciones desarrolladas con Microsoft .NET Framework. Pero se debe tener lo siguiente presente: Mono no se ha completado todavía, así que unas cuantas llamadas a la API podrían no estar implementadas todavía, y en algunos casos el comportamiento de Mono podría ser incorrecto.

Mono hoy en día presenta soporte para .NET 2.0 en los espacios de nombres compatibles, pero el soporte a las versiones de .NET 3.0, 3.5 y 4.0 no está completo.

La Fig. 13, muestra los resultados arrojados por MoMA al analizar el código fuente del Servicio para la administración remota del sistema alas PACSServer, los componentes desarrollados y las DLL externas empleadas.



The screenshot displays the Mono Migration Analyzer (MoMA) interface. At the top, there is a header with the MoMA logo and the text "mono migration analyzer". Below this, the "Analysis Summary:" section lists four items:

- ✓ All methods called exist in Mono.
- ✓ No P/Invokes are called.
- ✓ No methods that throw NotImplementedException are called.
- ⚠ Methods called marked with [MonoTodo]: 43

Below the list, a message states: "There were potential issues detected in the selected assemblies. A report detailing each issue has been created." A link labeled "View Detail Report" is provided below the message.

At the bottom of the interface, there is a footer area containing "Version 2.4" on the left, "Step 3 of 5" and the URL "http://www.mono-project.com" in the center, and "Back" and "Next" buttons on the right.

Fig. 11. Resultados del MoMA.

Los métodos marcados con [MonoTodo] pueden o no causar problemas para la aplicación que se prueba. A veces, un método puede ser marcado con este para recordar a un desarrollador que alguna parte pequeña no está implementada. Otras veces, el método sencillamente no está implementado todavía en

Mono, y simplemente no se realiza ninguna función. Esto se hace para que una aplicación compile y se ejecute, aunque falten algunas funcionalidades.

El informe detallado puede mostrar una razón específica por la cual el método está marcado como [MonoTodo]. Estos métodos probablemente pueden ser ignorados inicialmente. La aplicación debería funcionar sin que se caiga. La funcionalidad que falte puede ser implementada por uno mismo, o esperar hasta que el método sea implementado completamente en Mono.

En el capítulo se mostraron los elementos relacionados con la implementación del sistema. Se dieron a conocer los elementos generales relacionados a la implementación de los componentes desarrollados. Se mostró la forma en que estos serán desplegados físicamente entre los servidores que se emplearán para realizar la instalación del sistema, así como la compatibilidad existente con la plataforma Mono

CONCLUSIONES

Con el desarrollo del Servicio para la administración remota del sistema alas PACSServer se arribaron a las siguientes conclusiones:

La información recopilada sobre los antecedentes existentes en el mundo constituye las bases que definieron la estructura de la solución desarrollada. Se ha alcanzado un mejor entendimiento del sistema así como de las condiciones, capacidades, habilidades y restricciones que deben existir para satisfacer las necesidades del cliente.

Los artefactos obtenidos a partir del desarrollo del Flujo de Trabajo de Diseño, permiten modelar en detalle el sistema mediante la realización, y especificación de los flujos de eventos e interacciones entre las clases, se implementó el software propuesto como solución. Y al concluir el desarrollo de esta investigación, se comprobó la compatibilidad del sistema con la plataforma libre mono.

RECOMENDACIONES

Para darle continuidad a este trabajo en aras de mejorar la calidad de los servicios de salud en el país se recomienda:

- ✓ Buscar variantes en la programación y en la utilización de algunas librerías, con el objetivo de lograr una entera compatibilidad con la plataforma de software Mono y de esta manera eliminar la posibilidad de que la aplicación pierda funcionalidades si es utilizada por distintas plataformas.
- ✓ Desarrollar nuevas herramientas para facilitar la administración del sistema.
- ✓ Mantenerse al tanto del estado del arte del estándar DICOM 3.0 para conocer los cambios que se realicen y conservar actualizado el sistema.

REFERENCIAS BIBLIOGRÁFICAS

- 1- Introducción [Citado el 4 de noviembre del 2010] lo puede encontrar en: www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml.
- 2- Introducción [Citado el 5 de noviembre del 2010] lo puede encontrar en: http://newton.azc.uam.mx/mcc/01_esp/11_tesis/tesis/terminada/060701_jimenez_herrera_armando.pdf.
- 3- Tendencias y tecnología actual – Netretina. [Citado el 12 de noviembre del 2010] lo puede encontrar en: <http://www.netretina.com/>.
- 4- Tendencias y tecnología actual – LogMeln. [Citado el 12 de noviembre del 2010] lo puede encontrar en: <http://www.wihe.net/mejores-programas-para-controlar-remotamente-computadora/>.
- 5- Tendencias y tecnología actual – Telemedicina [Citado el 12 de noviembre del 2010] lo puede encontrar en: <http://www.estrategias.com/page/telemedicina>.
- 6- Tendencias y tecnología actual – HL7. [Citado el 1 de noviembre del 2010] lo puede ver en: HL7 (Health Level 7).ppt.
- 7- Tendencias y tecnología actual – SAIMED. [Citado el 5 de enero del 2010] lo puede encontrar en: <http://www.coseraba.net/index.php/servicios/saimed.html>.
- 8- Tendencias y tecnología actual – PHP. [Citado el 10 de diciembre del 2011] lo puede encontrar en: <http://www.webestilo.com/php/php00.phtml>.
- 9- Tendencias y tecnologías actuales – Ajax. [Citado el 21/1/2011] lo puede encontrar en: <http://www.librosweb.es/ajax/capitulo1.html>.
- 10- Tendencias y tecnología actual – SIENET. [Citado el 11 de octubre del 2010] lo puede encontrar en: http://www.siemens.cl/med/gestion_imagenes_medicas.htm.
- 11- Metodología de desarrollo – Rational Unified Process (RUP). [Citado el 10/1/2011] lo puede encontrar en: <http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm>.
- 12- Metodología de desarrollo –Extreme Programming (XP) –Microsoft Solution Framework (MSF). [Citado el 10/1/2011] lo puede encontrar en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
- 13- Herramienta diseño UML – Enterprise Architect. [Citado el 8 de enero del 2011] lo puede encontrar en: <http://www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0> .

- 14- Herramienta diseño UML – Enterprise Architect. [Citado el 8 de enero del 2011] lo puede encontrar en: <http://www.sparxsystems.com.ar/products/ea.html>.
- 15- Herramientas – Cliente del Subversion *TortoiseSVN 1.6.6*. [Citado el 24 de mayo del 2011] lo puede encontrar en:
http://portal2005.santafe.gov.ar/index.php/web/content/download/89560/425711/version/2/file/DPGP_Instructivo_InstalacionAdmin.pdf.
- 16- Herramientas – PostgreSQL. [Citado el 8 de enero del 2011] lo puede encontrar en: Tesis.uci.cu.
- 17- Herramientas – Microsoft Project. [Citado el 8 de enero del 2011] lo puede encontrar en: http://es.wikipedia.org/wiki/Microsoft_Project.
- 18- Herramientas – ASP.NET. [Citado el 10 de enero del 2011] lo puede encontrar en: <http://www.webestilo.com/aspnet/aspnet01.phtml>.
- 19- Glosario – HL7. [Citado el 16 de marzo del 2011] lo puede encontrar en: <http://www.hl7spain.org/VerPagina.asp?IDPage=0>.
- 20- Glosario – AJAX. [Citado el 16 de marzo del 2011] lo puede encontrar en: <http://es.wikipedia.org/wiki/AJAX>.
- 21- Glosario – JQuery. [Citado el 16 de marzo del 2011] lo puede encontrar en: <http://es.wikipedia.org/wiki/JQuery>.
- 22- Glosario – ANSI. [Citado el 16 de marzo del 2011] lo puede encontrar en: <http://www.monografias.com/trabajos11/utp/utp.shtml#ANSI>.

BIBLIOGRAFÍA

- Ajax. [Citado el 21/1/2011] lo puede encontrar en: <http://www.librosweb.es/ajax/capitulo1.html>.
- ANSI. [Citado el 16 de marzo del 2011] lo puede encontrar en: <http://www.monografias.com/trabajos11/utp/utp.shtml#ANSI>.
- ASP.NET. [Citado el 10 de enero del 2011] lo puede encontrar en: <http://www.webestilo.com/aspnet/aspnet01.phtml>.
- Booch y Rumbaugh, Jame Ivan. El proceso unificado del desarrollo de software. 2004.
- Cliente del Subversion *TortoiseSVN 1.6.6*. [Citado el 24 de mayo del 2011] lo puede encontrar en: http://portal2005.santafe.gov.ar/index.php/web/content/download/89560/425711/version/2/file/DPG_P_Instructivo_InstalacionAdmin.pdf.
- Enterprise Architect. [Citado el 8 de enero del 2011] lo puede encontrar en: <http://www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0>.
- Enterprise Architect. [Citado el 8 de enero del 2011] lo puede encontrar en: <http://www.sparxsystems.com.ar/products/ea.html>.
- Extreme Programming (XP), Microsoft Solution Framework (MSF). [Citado el 10/1/2011] lo puede encontrar en:
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
- HL7 (Health Level 7). Organización de Desarrollo de Estándares + HL7. [En línea] 04 de Septiembre de 2009. [Citado el: 5 de Diciembre de 2011.] <http://www.google.com/#hl=es&source=hp&q=Organizaci...>
- HL7. [Citado el 1 de noviembre del 2010] lo puede ver en: HL7 (Health Level 7).ppt.
- HL7. [Citado el 16 de marzo del 2011] lo puede encontrar en: <http://www.hl7spain.org/VerPagina.asp?IDPage=0>.
- JQuery. [Citado el 16 de marzo del 2011] lo puede encontrar en: <http://es.wikipedia.org/wiki/JQuery>.
- Librería GTK +. [Citado el 14 de noviembre del 2010] lo puede encontrar en: <http://www.linperial.com/communities/forums/developers/?q=node/19>.

- Librería QT. [Citado el 14 de noviembre del 2010] lo puede encontrar en: [http://es.wikipedia.org/wiki/Qt_\(biblioteca\)](http://es.wikipedia.org/wiki/Qt_(biblioteca)).
- LogMeln. [Citado el 12 de noviembre del 2010] lo puede encontrar en: <http://www.wihe.net/mejores-programas-para-controlar-remotamente-computadora/>.
- Microsoft Project. [Citado el 8 de enero del 2011] lo puede encontrar en: http://es.wikipedia.org/wiki/Microsoft_Project.
- Ministerio de Gobierno y Reforma del Estado. [En línea] 11 de marzo de 2008. [Citado el: 24 de Mayo de 2011.] portal2005.santafe.gov.ar/.../DPGP_Instructivo_InstalacionAdmin.pdf. ISBN/ISSN.
- MoMA, Mono Migration Analyzer. [Citado el: 5 de Mayo del 2011] lo puede encontrar en: www.mono-project.com/DistroPackages/Ubuntu
- Navarro., Franco Jose Ángel. UML en acción. Modelando Aplicaciones Web. [Documento] Ciudad de La Habana: s.n., 2005.
- Netretina. Netretina. [En línea] [Citado el: 12 de Noviembre de 2010.] <http://www.netretina.com/>.
- PHP. [Citado el 10 de diciembre del 2011] lo puede encontrar en: <http://www.webestilo.com/php/php00.phtml>.
- PostgreSQL. [Citado el 8 de enero del 2011] lo puede encontrar en: Tesis.uci.cu.
- Programadores web. [aut. libro] Pérez Eguíluz Javier. Introducción a AJAX.
- Rational Unified Process (RUP). [Citado el 10/1/2011] lo puede encontrar en: <http://www.utvm.edu.mx/Organoinformativo/orgJul07/RUP.htm>.
- Pressman Roger S. Ingeniería de Software. Quinta Edición. McGraw-Hill Interamericana. Madrid. 2002. Coulouris George. Sistemas Distribuidos. Tercera Edición. Wesley Addison. Madrid. 2001. Guía Rápida de Internet, Saiz Zurdo David, Burgoa Sicilia Alejandro, Quero Acevedo Fernando, Paraninfo. Madrid. 1997.
- SAIMED. [Citado el 5 de enero del 2010] lo puede encontrar en: <http://www.coseraba.net/index.php/servicios/saimed.html>.
- SIENET. [Citado el 11 de octubre del 2010] lo puede encontrar en: http://www.siemens.cl/med/gestion_imagenes_medicas.htm.

- Sistema PACS mínimo basado en el estándar DICOM. Lo puede encontrar en: http://newton.azc.uam.mx/mcc/01_esp/11_tesis/tesis/terminada/060701_jimenez_herrera_armando.pdf.
- Telemedicina [Citado el 12 de noviembre del 2010] lo puede encontrar en: <http://www.estrategias.com/page/telemedicina>.

ANEXOS

Anexo 1: Realización de Casos de Uso del Diseño

A 1.1 Caso de uso Gestionar Estaciones Clientes.

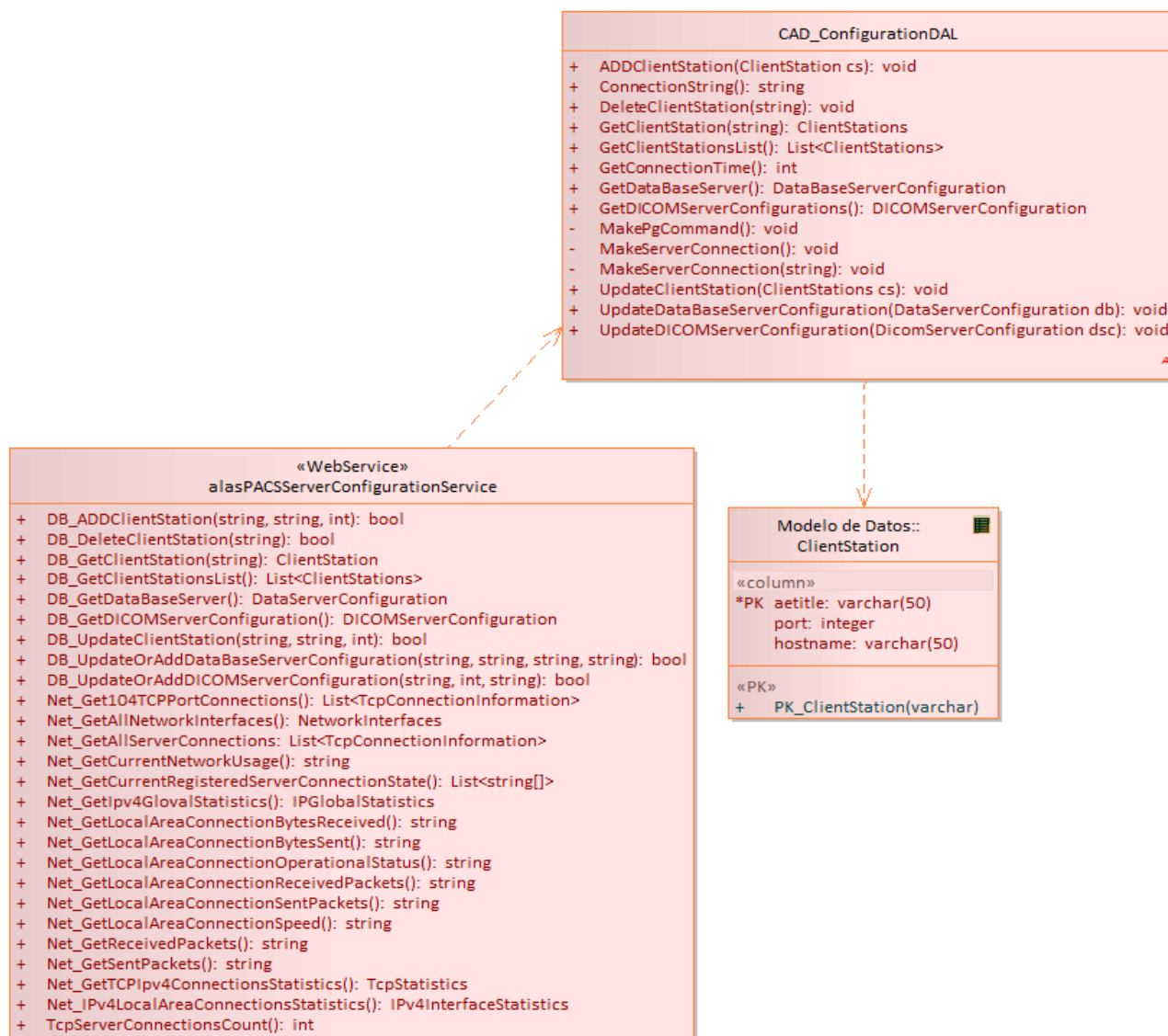


Fig. 14 Diagrama de clases del diseño "Gestionar estaciones clientes".

A.1.2 Caso de Uso Configurar Servidor DICOM

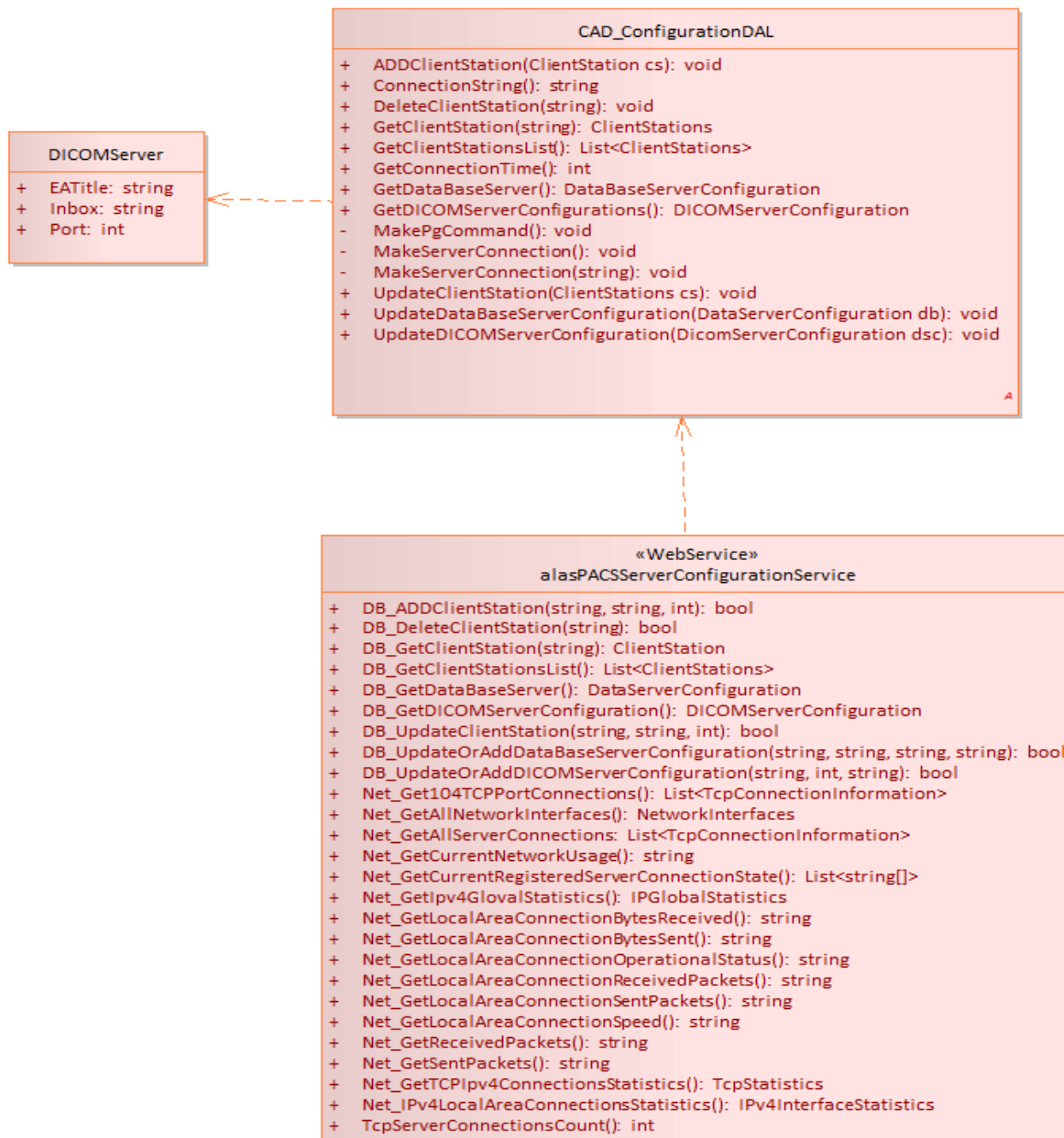


Fig. 18 Diagrama de clases del diseño "Servidor DICOM"

A.1.3 Caso de Uso Configurar Servidor de base de datos



Fig. 20 Diagrama de clase del diseño "Configurar servidor de base de datos"

A.1.4 Caso de Uso Visualizar la información de conexiones de forma gráfica

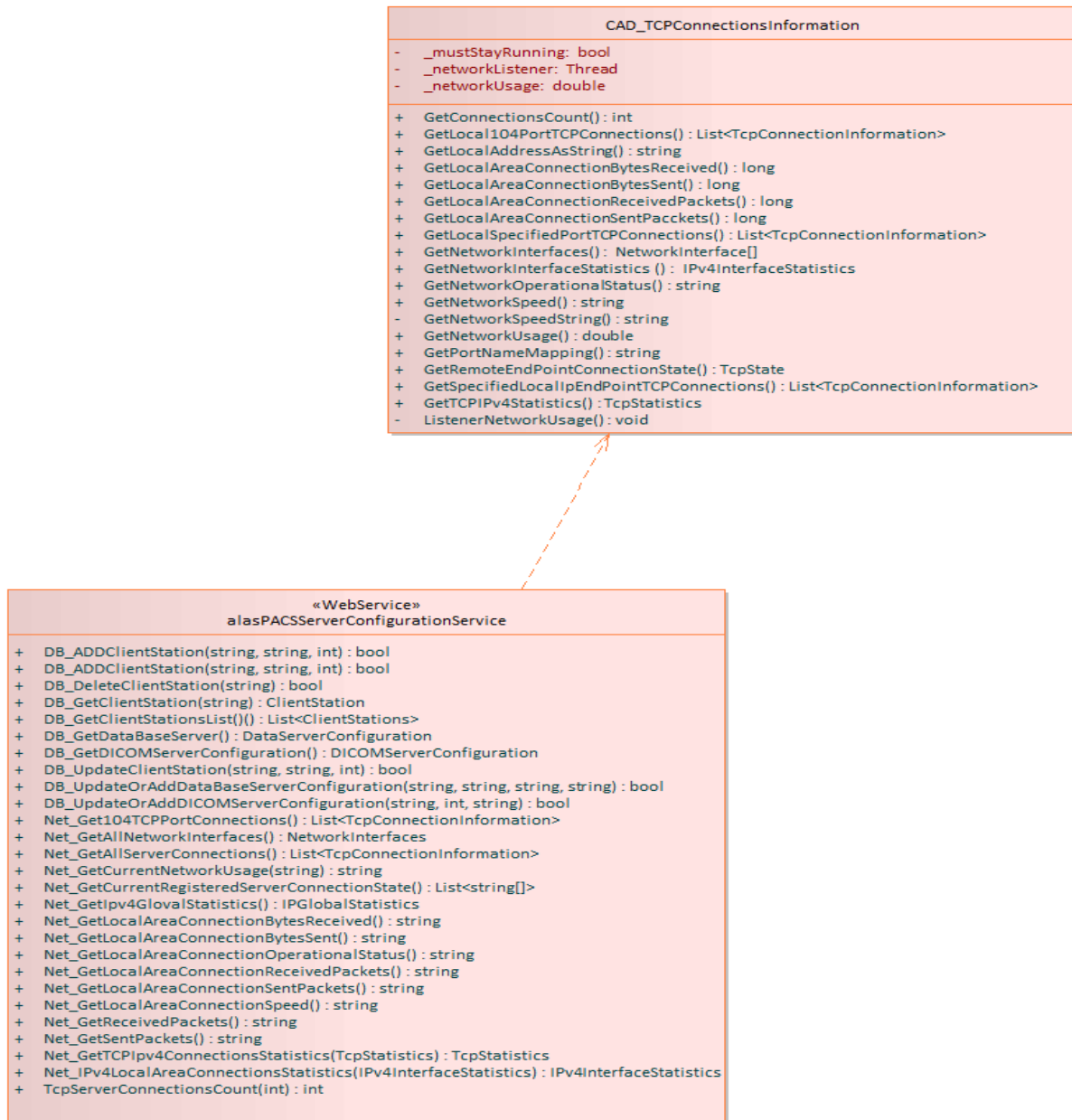


Fig. 22 Diagrama de clase del diseño "Visualizar Conexiones "

A 1.5 Caso de Uso Visualizar control del tráfico de conexiones.



Fig. 24 Diagrama de clase del diseño "Control del Tráfico de Conexiones "

Anexo 2: Descripción de las Clases

A.2.1 Descripción de la clase alasPACSServerConfigurationService (Web Service)

Nombre:	alasPACSServerConfigurationService (Web Service)	
Tipo de Clase:	Controladora	
Atributo:		Tipo:
Para cada responsabilidad:		
Nombre:	DB_ADDClientStation(string aetitle, string hostname, string port)	
Descripción:	Para insertar los datos de una nueva estación cliente en la base de datos.	
Nombre:	DB_UpdateClientStation(string aetitle, string hostname, string port)	
Descripción:	Para actualizar los datos de una estación cliente existente en la base de datos.	
Nombre:	DB_DeleteClientStation(string aetitle)	
Descripción:	Para eliminar los datos de una estación cliente de la base de datos.	
Nombre:	DB_GetClientStation(string aetitle)	
Descripción:	Devuelve los datos de una estación cliente a partir de su entidad de	

Servicios para la administración remota del sistema | **Anexos** alas PACSServer.

	aplicación.
Nombre:	DB_GetClientStationsList()
Descripción:	Devuelve la lista de las estaciones clientes guardadas en la base de datos.
Nombre:	DB_UpdateOrAddDICOMServerConfiguration(string aetitle, int port, string inbox)
Descripción:	Para actualizar los datos de configuración del servidor DICOM o en caso de que aún no estén en la base de datos los inserta en esta.
Nombre:	DB_GetDICOMServerConfiguration()
Descripción:	Devuelve los datos del servidor DICOM guardados en la base de datos.
Nombre:	DB_GetDataBaseServer()
Descripción:	Devuelve los datos del servidor de base de datos del sistema.
Nombre:	DB_UpdateOrAddDataBaseServerConfiguration(string serverHost, string dataB, string user, string pass)
Descripción:	Para actualizar los datos del servidor de base de datos del sistema o sino se encuentran registrados en la base de datos los inserta en esta.
Nombre:	IsValidIP(string _ipAddress)
Descripción:	Para comprobar si una dirección IP es válida o no.

Servicios para la administración remota del sistema | **Anexos** alas PACSServer.

Nombre:	Net_GetTcpServerConnectionsCount()
Descripción:	Devuelve el número de conexiones del protocolo TCP que se realizan al servidor a través del puerto que tiene registrado.
Nombre:	Net_Get104TCPPortConnections()
Descripción:	Devuelve todas las conexiones del protocolo TCP que se realizan al servidor a través del puerto 104.
Nombre:	Net_GetCurrentRegisteredServerConnectionsState()
Descripción:	Devuelve el estado de la conexión entre cada una de las estaciones clientes registradas en la base de datos y el servidor.
Nombre:	Net_GetAllServerConnections()
Descripción:	Devuelve todas las conexiones del protocolo TCP que se realizan al servidor a través del puerto que tiene registrado.
Nombre:	Net_GetCurrentNetworkUsage()
Descripción:	Devuelve el porcentaje de uso de la conexión de área local en un instante de tiempo.
Nombre:	Net_GetAllNetworkInterfaces()
Descripción:	Devuelve todas las interfaces de red que posee el servidor.

Servicios para la administración remota del sistema | **Anexos** alas PACSServer.

Nombre:	Net_IPv4LocalAreaConnectionStatistics()
Descripción:	Devuelve las estadísticas del protocolo Ipv4 de la conexión de área local en el servidor.
Nombre:	Net_GetReceivedPackets()
Descripción:	Devuelve el número de paquetes recibidos a través del protocolo Ipv4 en el servidor.
Nombre:	Net_GetSentPackets()
Descripción:	Devuelve el número de paquetes enviados a través del protocolo Ipv4 en el servidor.
Nombre:	Net_GetTCPIPv4ConnectiosStatistics()
Descripción:	Devuelve las estadísticas de conexión a través del protocolo TCP-Ipv4 en el servidor.
Nombre:	Net_GetLocalAreaConnectionSpeed()
Descripción:	Devuelve la velocidad de la interfaz de conexión de área local en el servidor.
Nombre:	Net_GetLocalAreaConnectionOperationalStatus()
Descripción:	Devuelve el estado operacional de la interfaz de conexión de área local en el servidor.

Nombre:	Net_GetLocalAreaConnectionReceivedPackets()
Descripción:	Devuelve el número de paquetes recibidos por la interfaz de conexión de área local en el servidor.
Nombre:	Net_GetLocalAreaConnectionSentPackets()
Descripción:	Devuelve el número de paquetes enviados por la interfaz de conexión de área local en el servidor.
Nombre:	Net_GetLocalAreaConnectionBytesReceived()
Descripción:	Devuelve el número de bytes recibidos por la interfaz de conexión de área local en el servidor.
Nombre:	Net_GetLocalAreaConnectionBytesSent()
Descripción:	Devuelve el número de bytes enviados por la interfaz de conexión de área local en el servidor.

A.2.2 Descripción de la clase ConfigurationDAL

Nombre:	ConfigurationDAL	
Tipo de Clase:	Acceso a datos	
Atributo:		Tipo:

pgConnection	NpgsqlConnection
pgCommand	NpgsqlCommand
Para cada responsabilidad:	
Nombre:	ConnectionString
Descripción:	Propiedad que devuelve la cadena de conexión a la base de datos guardada en la web.config del servicio.
Nombre:	TestDefaultConnectionString()
Descripción:	Prueba la conexión con la base de datos usando la cadena de conexión existente en la web.config del servicio.
Nombre:	TestConnectionString(string entireConnectionString)
Descripción:	Prueba la conexión con la base de datos usando una cadena de conexión dada.
Nombre:	MakeServerConnection()
Descripción:	Crea la conexión con la base de datos con la cadena de conexión por defecto.
Nombre:	MakeServerConnection(string pad)
Descripción:	Crea la conexión a la base de datos pasándole una cadena de conexión determinada.

Nombre:	MakePgCommand()
Descripción:	Crea un nuevo comando para postgres.
Nombre:	AddClientStation(ClientStations cs)
Descripción:	Para insertar los datos de conexión de una nueva estación cliente en la base de datos.
Nombre:	DeleteClientStation(string aetitle)
Descripción:	Para eliminar los datos de una estación cliente determinada.
Nombre:	UpdateClientStation(ClientStations cs)
Descripción:	Para actualizar los datos de una estación cliente determinada en la base de datos.
Nombre:	GetClietStation(string aetitle)
Descripción:	Devuelve los datos de conexión de una estación cliente determinada.
Nombre:	GetClientStationsList()
Descripción:	Devuelve la lista de estaciones clientes que se encuentran en la base de datos.
Nombre:	GetDICOMServerConfiguration()
Descripción:	Devuelve los datos de configuración del servidor DICOM guardados en la base de datos.

Nombre:	UpdateOrADD_DICOMServerConfiguration(DicomServer dsc)
Descripción:	Actualiza los datos de configuración del servidor DICOM y en caso de no estar aún en la base de datos, insertar los nuevos datos.
Nombre:	GetDataBaseServer()
Descripción:	Para conectarse a la base de datos y devolver los datos del servidor de base de datos.
Nombre:	UpdateOrADD_DataBaseServerConfiguration(DataBaseServer db)
Descripción:	Actualiza los datos de configuración del servidor de base de datos del sistema, en caso de que aún no se encuentren en la base de datos lo inserta.

A.2.3 Descripción de la clase TCPConnectionsInformation

Nombre:	TCPConnectionsInformation	
Tipo de Clase:	Acceso a datos	
Atributo:	Tipo:	
_networkUsage	double	
_mustStayRunning	bool	
_networkListener	Thread	

Para cada responsabilidad:	
Nombre:	GetLocal104PortTCPConnections()
Descripción:	Devuelve las conexiones del protocolo TCP que se realizan a través del puerto 104.
Nombre:	GetLocalSpecifiedPortTCPConnections(uint port)
Descripción:	Devuelve las conexiones del protocolo TCP que se realizan a través de un puerto dado.
Nombre:	GetSpecifiedLocalIpEndPointTCPConnections(IPEndPoint localIp)
Descripción:	Devuelve las conexiones del protocolo TCP que se realizan a través de una dirección de Ip y un puerto especificados.
Nombre:	GetRemoteEndPointConnectionState(IPEndPoint remoteEndPoint,uint localPort)
Descripción:	Devuelve el estado de la conexión de un punto de red dado con el servidor, a través de un puerto local dado.
Nombre:	GetConnectionsCount(int port)
Descripción:	Devuelve el número de conexiones del protocolo TCP que se realizan al servidor mediante un puerto dado.
Nombre:	GetNetworkInterfaces
Descripción:	Propiedad que devuelve todas las interfaces de red que posee el servidor.

Nombre:	GetNetworkInterfaceStatistics(NetworkInterface netInterface)
Descripción:	Devuelve las estadísticas del protocolo Ipv4 de una interfaz de red dada.
Nombre:	GetTCPIPv4Statistics()
Descripción:	Devuelve las estadísticas del protocolo TCP/Ipv4 del servidor.
Nombre:	ListenerNetworkUsage()
Descripción:	Método que esta calcula cada un segundo el porciento de uso de la interfaz de conexión de área local del servidor.
Nombre:	GetNetworkUsage()
Descripción:	Devuelve el porciento de uso de la interfaz de conexión de área local del servidor.
Nombre:	GetNetworkSpeed()
Descripción:	Devuelve la velocidad de la interfaz de conexión de área local del servidor.
Nombre:	GetNetworkOperationalStatus()
Descripción:	Devuelve el estado operacional de la interfaz de conexión de área local.
Nombre:	GetLocalAreaConnectionReceivedPackets()
Descripción:	Devuelve el número de paquetes recibidos por la interfaz de conexión de área local.

Nombre:	GetLocalAreaConnectionSentPacckets()
Descripción:	Devuelve el número de paquetes enviados por la interfaz de conexión de área local.
Nombre:	GetLocalAreaConnectionBytesReceived()
Descripción:	Devuelve el número de bytes recibidos por la interfaz de conexión de área local.
Nombre:	GetLocalAreaConnectionBytesSent()
Descripción:	Devuelve el número de bytes enviados por la interfaz de conexión de área local.

Anexo 3: Descripción de las tablas de la Base de Datos

A.3.1. Descripción de la tabla DICOMServer

Nombre: DICOMServer		
Descripción: Almacena los datos de la configuración del servidor DICOM.		
Atributo	Tipo	Descripción
aetitle	varchar(50)	Entidad de Aplicación del servidor DICOM. PK_DICOMServer (vachar).
port	integer	Puerto, el cual generalmente por defecto es el 104.

inbox	varchar(50)	Bandeja de entrada.
-------	-------------	---------------------

A.3.2. Descripción de la tabla ClientStation

Nombre: ClientStation		
Descripción: Es donde se guardan los datos con respecto a las Estaciones Clientes ya sea adicionados, eliminados o modificados.		
Atributo	Tipo	Descripción
aetitle	varchar(50)	Entidad aplicación del cliente. PK_ClientStation (varchar).
hostname	inbox	Puerto de la estación cliente.
port	varchar(50)	IP de la estación cliente que se va a configurar.

A.3.3. Descripción de la tabla DataBaseServer

Nombre: DataBaseServer.		
Descripción: Es donde se guardan los datos con respecto al Servidor de la Base de Datos.		
Atributo	Tipo	Descripción
database	varchar(50)	Base de datos que se usará.

		PK_DataBaseServer(varchar)
hostname	integer	Dirección IP de donde se va a configurar.
user	varchar(50)	Usuario de quien va a configurar el servidor de la base de datos.
pass	varchar(50)	La contraseña del usuario que va a configurar el servidor de la base de datos.

GLOSARIO DE TÉRMINOS

Ajax

Acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas se ejecutan en el cliente, es decir, en el navegador de los usuarios, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma (20).

ANSI

Instituto Nacional Americano de Normalización. Organización voluntaria compuesta por corporativas, organismos del gobierno y otros miembros que coordinan las actividades relacionadas con estándares, aprueban los estándares nacionales de los EE.UU. y desarrollan posiciones en nombre de los Estados Unidos ante organizaciones internacionales de estándares(22).

CASE

Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación. Es la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación.

CSS

Cascading Style Sheets traducido al español es Hojas de Estilo en Cascada, el cual es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

DOM

Modelo de Objetos del Documento es una interfaz de programación de aplicaciones (API) para documentos validos HTML y bien construidos XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula.

Framework

Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

HL7

Es una organización internacional, iniciada en los Estados Unidos en 1987, que pretende promover el desarrollo y evolución del estándar HL7 (Health Level Seven) para el formato de datos e intercambio de información entre diferentes Sistemas de Información de Salud.(19)

HTML

HyperText Markup Language, es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

JavaScript

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.

JQuery

Es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web(21).

JSON

JavaScript Object Notation - Notación de Objetos de JavaScript es un formato ligero de intercambio de datos. Es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Es un lenguaje ideal para el intercambio de datos.

PHP

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

Web

World Wide Web o Red Global Mundial es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet.

WML

El Wireless Markup Language o Lenguaje de Marcado Inalámbrico es un lenguaje cuyo origen es el XML (EXtensible Markup Language). Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDA).

XHTML

EXtensible Hypertext Markup Language o lenguaje extensible de marcado de hipertexto, es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. Tiene, básicamente, las mismas funcionalidades que HTML, pero cumple las especificaciones, más estrictas, de XML.

XML

Extensible Markup Language o lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium*.

XSLT

Extensible Stylesheet Language Transformations, o lenguaje de hojas extensibles de transformación, que permite convertir documentos XML de una sintaxis a otra (por ejemplo, de un XML a otro o a un documento HTML).