

**PUESTA EN MARCHA DEL SISTEMA VIRTUAL DE ENSEÑANZA DE
ELECTROCARDIOGRAFÍA DE LA UNIVERSIDAD INDUSTRIAL DE
SANTANDER
-SIVEUIS-**

**BABINGTON LEONARDO ARENAS CALDERÓN
GRATEFUL DEAD MONTAÑO SIERRA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**PUESTA EN MARCHA DEL SISTEMA VIRTUAL DE ENSEÑANZA DE
ELECTROCARDIOGRAFÍA DE LA UNIVERSIDAD INDUSTRIAL DE
SANTANDER
-SIVEUIS-**

**BABINGTON LEONARDO ARENAS CALDERÓN
GRATEFUL DEAD MONTAÑO SIERRA**

**Trabajo de grado para optar al título de
Ingeniero de Sistemas**

Director

ALFONSO MENDOZA CASTELLANOS

BSD. DEA. Automática, Informática y Mecánica Aplicada a la Robótica

Codirector

OSCAR LEONEL RUEDA OCHOA

Médico internista

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

RESUMEN

TITULO: PUESTA EN MARCHA DEL SISTEMA VIRTUAL DE ENSEÑANZA DE ELECTROCARDIOGRAFÍA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER - SIVEUIS-*

AUTORES: ARENAS CALDERON, Babington Leonardo.
MONTAÑO SIERRA, Grateful Dead.**

PALABRAS CLAVE: Electrocardiografía, Internet, TIC, Ambiente virtual, PHP, Postgres.

DESCRIPCIÓN:

SiveUIS (Sistema virtual para la enseñanza de la electrocardiografía Universidad Industrial de Santander) es un entorno virtual de aprendizaje elaborado con el fin de reforzar los conocimientos del personal del área de la salud frente a los conceptos básicos de la electrocardiografía, la lectura del electrocardiograma y su interpretación clínica con ayuda de las grandes funcionalidades de las Tecnologías de Información y Comunicación (TIC) de manera práctica e interactiva empleando recursos Web como fuente de información e instrumento de procesamiento de datos.

Como metodología de desarrollo se utilizó Proceso Unificado que garantiza un orden y control total en cada una de las fases del proyecto. Utiliza el lenguaje unificado de modelado (UML) como herramienta de modelado y documentación.

-Entre las características que proporciona este entorno a sus integrantes se cuentan: abre canales de comunicación a través de los cuales se comparten ideas y materiales; se debaten temas y se consultan dudas; proporciona acceso tanto público como privado; establece perfiles de usuario; evalúa el rendimiento de los participantes, brinda herramientas que permiten la fácil administración y actualización de sus contenidos y que genera estadísticas de tráfico; vincula estrategias que permite agilizar el tiempo de ejecución de sus componentes y brindan seguridad a la plataforma en caso de ataques. Al usar software libre para el desarrollo de la aplicación, se obtiene un sistema robusto de gran desempeño, estabilidad, escalabilidad y a un costo menor del que se tendría al usar software propietario.

* Trabajo de grado

** Facultad de Ingeniería Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director de Proyecto: Alfonso Mendoza Castellanos. Codirector: Oscar Leonel Rueda Ochoa

ABSTRACT

TITLE: VIRTUAL SYSTEM OF EDUCATION OF ELECTROCARDIOGRAPHY SIVEUIS*

AUTHORS: ARENAS CALDERON, Babington Leonardo.
MONTAÑO SIERRA, Grateful Dead.**

KEY WORDS: Electrocardiography, Internet, TIC, Ambient virtual, PHP, Postgres.

DESCRIPTION:

SiveUIS (Virtual System for the teaching of the Electrocardiography) is a virtual environment of learning elaborated in order to improve the knowledge of the general Colombian doctor about the basics concepts of the Electrocardiography, the analysis of the electrocardiogram and its clinic interpretation with support of the great functionalities of the Information and Communication Technologies (ICT) making use of Web resources as information source, tool of data processing and relating strategies to permit quick disposition of its pedagogic tools.

Unified Process was used as a developing methodology that ensures an order and a total control in every phase at the project. Unified Modeling Language (UML) was used as a documentation and modeling tool.

This space opens channels of communication across which ideas, materials are shared, topics are debated and doubts consult; proportionate public and private access; it establishes user's profiles; it evaluates the performance of the participants; tools which permit the easy administration and update of contents and generate traffic statistics; strategies which permit to improve the time of execution of its components and they offer safety to the platform in case of assaults. On having used software free for the development of the application, there is obtained a robust system of great performance, stability, scalability and to a minor cost of the one that would be have on having used proprietary software.

* Work Of Degree

** Faculty of Physics – Mechanics Engineering. System Engineering And Informatics School. Director: Alfonso Mendoza Castellanos. Codirector: Oscar Leonel Rueda Ochoa.

CONTENIDO

1. PRESENTACIÓN DEL PROYECTO

1.1 TITULO DEL TRABAJO

1.2 OBJETIVOS

1.3 ENTIDADES INTERESADAS

1.4 JUSTIFICACIÓN

1.4.1 Antecedentes Y Situación Del Problema

1.4.2 Impacto y Viabilidad

2. MARCO TEÓRICO

2.1. ÁREA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

2.1.1. Aplicaciones Web

2.1.1.1. Arquitectura Web

2.1.2 Interfaz de usuario

2.1.3 Arquitectura Cliente – Servidor

2.1.3.1. Características de la Arquitectura Cliente Servidor

2.1.3.2. Componentes de la Arquitectura Cliente/Servidor

2.1.4 Proceso Unificado de Desarrollo

2.2 APRENDIZAJE COLABORATIVO EN ENTORNOS VIRTUALES

2.2.1. Aprendizaje Colaborativo vs. Aprendizaje Cooperativo

2.2.2. Grupos de Trabajo

2.2.3. Operaciones del pensamiento que apoyan el trabajo colaborativo

2.2.4. Cambio de la metodología tradicional al Aprendizaje Colaborativo

2.2.5. *Aprendizaje basado en la resolución de problemas*

2.2.5.1. ¿Qué es un problema?

2.2.5.2. Objetivos de la enseñanza problémica

2.2.5.3 Fases para la resolución de problemas de tipo numérico

2.2.6. Ambientes computacionales de aprendizaje colaborativo

3. DESARROLLO DE LA HERRAMIENTA

3.1 ANÁLISIS DE REQUISITOS

3.2 DISEÑO

3.2.1 Arquitectura del Sistema

3.2.2 Módulos del Sistema

3.2.2.1 Módulo de Registro

3.2.2.2 Módulo de Estudio

3.2.2.3 Módulo de Evaluación

3.2.2.4 Módulo de Logros

3.2.2.5 Módulo de Ayuda

3.2.2.6 Módulo de Comunicaciones

3.2.2.7 Módulo de Administrador

3.3 IMPLEMENTACIÓN

3.3.1 Diagramas de Actividad

Modelo de Actividad para el Módulo de Iniciar Sesión

Modelo de Actividad para el Módulo de Evaluación

3.3.2 Aspectos de Optimización del Sistema

3.3.2.2 Optimización del Tiempo de Retorno

3.3.2.2 Optimización de la Transferencia de Datos

3.3.2.3 Usuarios de Base de Datos Para el Correo

3.3.2.4 Monitoreo y Análisis del Sistema

3.3.2.5 Sistema de Alimentación Ininterrumpida SAI Administrable Vía Web

3.3.3 Funcionamiento de Siveuis

4. CONCLUSIONES

5. RECOMENDACIONES

6. GLOSARIO

7. BIBLIOGRAFÍA

ANEXOS

LISTA DE ILUSTRACIONES

- Figura 1 Diagrama de Actividad, Modulo de Iniciar Sesión
- Figura 2 Diagrama de Actividad, Modulo de Evaluación
- Figura 3 Modelo de cliente/servidor Web.
- Figura 4. Esquema páginas estáticas
- Figura 5. Esquema páginas dinámicas
- Figura 6. Arquitectura Cliente/servidor
- Figura 7 Los cinco flujos de trabajo requisitos, análisis, diseño, implementación y prueba – tienen lugar sobre las cuatro fases: inicio, elaboración, construcción y transición
- Figura 8. Arquitectura del Sistema
- Figura 9 Diagrama de Caso de Uso, Modulo de Registro, Vista de Usuario
- Figura 10. Diagrama de Caso de Uso, Modulo de Registro, Vista Software
- Figura 11. Diagrama de Caso de Uso, Modulo de Estudio, Vista Del Usuario
- Figura 12. Diagrama de Caso de Uso, Modulo de Evaluación, Vista Del Usuario
- Figura 13 Diagrama de Caso de Uso, Modulo de Evaluación, Vista Sistema
- Figura 14 Diagrama de Actividad, Modulo de Iniciar Sesión
- Figura 15 Diagrama de Actividad, Modulo de Evaluación

INTRODUCCIÓN

Hace algunos años era improbable imaginar una situación en el mundo de la educación como en la que ahora vivimos. La tecnología, entendida como el arte del saber hacer, ha estado siempre presente en la historia de la pedagogía, las últimas décadas ha tomado un papel predominante como instrumento, como máquina al servicio de la educación. Desde el uso de los magnetófonos, la televisión, el vídeo, pasando por la enseñanza asistida por ordenador, los distintos instrumentos tecnológicos que han entrado en nuestra cotidianeidad a través de los salones de nuestras casas, han tenido su repercusión en las metodologías de educación y aprendizaje. En la antigua Grecia, la erudición verdadera estaba definida como una de ocio dedicada al aprendizaje. La búsqueda de la sabiduría conllevaba una total inmersión sensorial e intelectual en la propia vida y los educadores se vivieron desafiados a nutrir el entusiasmo y proporcionar las herramientas adecuadas a las mentes jóvenes.

Hace tiempo que sabemos que lo que define la relación entre las personas no es tan solo el medio que usan para hacerlo si no que sobretodo es la capacidad que tenemos para comunicarnos. Y para hacerlo no usamos sólo la presencia física, el cara a cara. Hemos pasado por las señales de humos, por la comunicación epistolar, telefónica, hasta las posibilidades actuales del correo electrónico. Sea como sea las personas nos relacionamos. Forma parte de nuestro ser social. Y si hay relación hay también acto educativo. Sea cual sea el medio de comunicación.

Hoy el medio de comunicación e información que esta en auge es el que proporciona la red de redes: Internet. La red nació hace aproximadamente treinta años pero tan solo hace algunos años que incorporó color, imagines, sonido, y facilidad en el transporte de datos. Se ha vuelto amigable y atractiva, lo que ha supuesto una rápida penetración en nuestras vidas cambiando nuestras mentalidades, nuestras formas de acceder al saber y de conocer. Nos

encontramos, por tanto ante un cambio en los procesos de aprendizaje que los más jóvenes viven como parte de su formación inicial básica y los más adultos viven, a menudo, como un problema personal de adaptación a un mundo cambiante.

Sin lugar a dudas, la computación nos puede ayudar para lidiar con la complejidad del mundo moderno y la presión de la escasez de recursos, pero los problemas asociados con la misma no deben ser disfrazados. Con tal pretensión, que hace gala de avances significativos en las Telecomunicaciones y erigidos en resultados investigativos serios que dan a la imagen, al color y al sonido fortalezas magnas, se provee para la educación un modelo que no hace desplazar al adulto hasta su fuente emisora (como el estudiante lo ha venido haciendo tradicionalmente, hasta el aula), sino que al contrario, es tan audaz que hasta los rincones domésticos puede llegar con eficiencia, confianza y discreción.

1. PRESENTACIÓN DEL PROYECTO

1.1 TITULO DEL TRABAJO

Puesta en Marcha del Sistema Virtual de Enseñanza de Electrocardiografía de la Universidad Industrial De Santander -SIVEUIS-

1.2 OBJETIVOS

- GENERAL
 - Adecuar el entorno Web del programa de educación virtual SIVEUIS vinculando estrategias técnicas y metodológicas que lo fortalezcan para así dinamizar el proceso de aprendizaje cooperativo y continuado de la electrocardiografía en el contexto nacional.

- ESPECÍFICOS
 - Proporcionar al usuario contenidos teóricos que le permitan evaluar y reforzar sus conocimientos en el área de la electrocardiografía empleando recursos Web que brinden al participante las herramientas básicas para la adecuada interpretación del Electrocardiograma.

 - Implementar una estrategia que permita agilizar el tiempo de ejecución del código PHP.

 - Reducir el ancho de banda necesario para atender peticiones HTTP utilizando el algoritmo de compresión gzip¹ recomendado por la especificación del protocolo HTTP 1.0.

¹ GZIP: formato de compresión de datos.

- Instalar un “Sistema de Alimentación Ininterrumpida” administrable desde Web que evite caídas abruptas del servidor e informe del estado del mismo al administrador.
- Conectar el servidor SMTP y el servidor POP3 a un sistema de base de datos para evitar la creación de usuarios del sistema operativo.
- Configurar un monitor de carga para registrar el estado de la memoria, cpu, tarjeta de red, del servidor HTTP y del servidor SMTP que presente los datos en forma gráfica para facilitar su análisis.

1.3 ENTIDADES INTERESADAS

- Escuela de Ingeniería de Sistemas e Informática
Grupo de Investigación en Ingeniería Biomédica
- Facultad de Salud UIS
Grupo de Electrocardiografía
- Facultades de salud del país
- Personal del área de la salud

1.4 JUSTIFICACIÓN

1.4.1 Antecedentes Y Situación Del Problema

El grupo de EKG de la facultad de salud y el grupo GIIB de la escuela de sistemas, de la Universidad Industrial de Santander vienen desarrollando un trabajo cooperativo que tiene como objetivo reforzar los conocimientos del medico general colombiano frente a la lectura del electrocardiograma y su interpretación clínica. La idea de este trabajo surge como respuesta al bajo rendimiento mostrado por 501 médicos de las ciudades de Bucaramanga, Cúcuta, Bogota, Medellín, Cali,

Barranquilla, Tunja e Ibagué en un examen basado en el ECG-EXAM, preparado, desarrollado, aplicado y analizado por el grupo de EKG entre Octubre de 1998 y Octubre de 2001.

Como parte de este trabajo se elaboro un portal Web, que además de complementar los conceptos básicos de la electrocardiografía, aprovecha el Internet como fuente de información, canal de comunicación e instrumento de procesamiento de datos para brindar herramientas que faciliten la interacción entre los profesionales de la salud interesados en el tema.

Dicho proyecto se llevo a cabo hasta su etapa de diseño, dejando a través de sus etapas de requisitos y análisis una base con todos los elementos necesarios para poner en marcha la etapa de implementación y pruebas.

Debido a la naturaleza misma de Internet y de que este es el ambiente en el cual se encuentra el portal, se ve expuesto a una serie de inconvenientes que es necesario tener en cuenta para lograr una correcta puesta en marcha.

Para permitir un acceso masificado, el portal debe estar en funcionamiento 24 horas al día durante 7 días a la semana (7x24), este hecho acarrea la necesidad de disponer de un sistema que permita salvar problemas derivados de la falta de flujo eléctrico.

Los contenidos ofrecidos deben llegar al mayor numero de usuarios posibles interesados en el tema por consiguiente deben presentarse en forma ágil al usuario evitando que sea expuesto a largos tiempos de espera que ocasionarían un elevado numero de abandonos por cansancio. Es necesario por ende lograr una disminución los tiempos de descarga y de visualización de los contenidos sin degradar la calidad de los mismos.

Siendo una plataforma de uso público y que ofrece el habitual servicio de correo se encuentra expuesto al ataque de los spammers que deambulan en la red buscando servidores open relay para realizar su trabajo.

A pesar de estar detrás del firewall institucional de la UIS no esta exento a los ataques cracker de dentro y fuera de la Intranet.

Las características de la red y del funcionamiento continuo del servidor se prestan para que sucedan algunas situaciones indeseables pero comunes que provocan cuellos de botella.

Estos cuellos de botella se presentan comúnmente en dispositivos tales como:

- Memoria
- CPU
- Tarjetas de Red

Por razones como:

- Utilización ineficiente del recurso
- Aumento en la utilización del tiempo de CPU
- Procesos demasiado lentos
- Exceso de carga en los dispositivos

Cada vez que se presenta un cuello de botella se produce un aumento considerable en la demanda de los recursos del servidor. Este aumento desencadena una baja en el rendimiento y en ocasiones hasta la caída del mismo.

1.4.2 Impacto y Viabilidad

- IMPACTO

La creación del entorno virtual SIVEUIS (Sistema virtual para la enseñanza de la electrocardiografía Universidad Industrial de Santander) permitirá al grupo de EKG hacer presencia en Internet, favoreciendo el conocimiento del grupo tanto a nivel nacional como internacional, hecho de gran importancia hoy en día para la disminución de esfuerzos en todas las experiencias que se lleven a cabo en el entorno investigativo.

En una segunda instancia se realiza interacción bidireccional, es decir, se permite además de las consultas, la recepción de aportes de aquellos visitantes que buscan un espacio para expresar sus ideas e inquietudes alrededor del área de la electrocardiografía, y el cambio de datos guardados en el sistema por parte de algunos actores como profesores, integrantes y participantes del curso, hecho que facilita procesos de actualización inmediata de formación y comunicación permanente, entre otros.

El contar con las herramientas necesarias para llevar a cabo una experiencia educativa motiva a los actores a pensar de una manera creativa la utilización de estos elementos, convirtiéndose así el proceso de aprendizaje en una experiencia diferente a lo tradicionalmente acostumbrado.

Hay que resaltar el manejo de herramientas de libre distribución que permiten la implementación de un sistema robusto, seguro y económico, ideal para dar solución al problema planteado.

- VIABILIDAD

Existen ventajas en el aspecto humano y tecnológico, podemos ver a partir del análisis de la investigación que se realizó, las necesidades que tiene el médico general colombiano en cuanto al estudio de la EKG, pero también cabe resaltar la

disposición que se tiene para involucrarse en el desarrollo de proyectos que beneficien el aprendizaje individual y cooperativo.

El desarrollo estará basado en herramientas gratuitas que permanentemente están siendo actualizadas para la corrección de errores y cuentan con un gran número de usuarios a nivel mundial quienes a través de listas de discusión informan alrededor de los últimos cambios, hecho que permite sobrellevar fácilmente las dificultades presentadas en el camino

2. MARCO TEÓRICO

2.1. ÁREA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

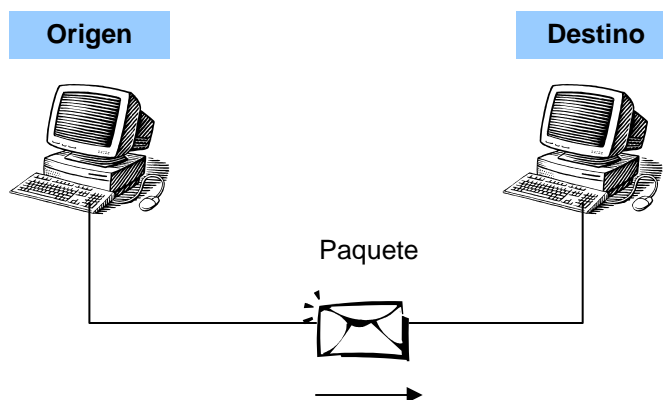
2.1.1. Aplicaciones Web

Con la introducción de Internet y el Web, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde cualquier lugar del mundo, de esta forma se crea un reto importante para los desarrolladores de aplicaciones ya que los avances en nuevas tecnologías demandan cada vez aplicaciones mas rápidas, ligeras y robustas que permitan utilizar la red.

2.1.1.1. Arquitectura Web

Para abrir una página Web en un navegador, cualquiera que sea, normalmente se teclea el correspondiente URL o se pica en el hiperenlace oportuno. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, el cual ubica la página Web en su sistema de ficheros y la envía de vuelta al navegador que la solicitó.

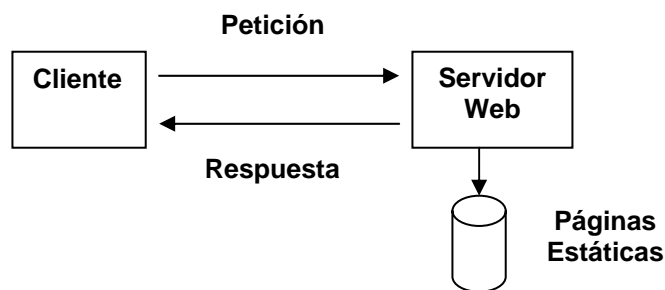
Figura 1 Modelo de cliente/servidor Web.



- **Páginas Estáticas**

En este grupo se encuentran las páginas que son actualizadas manualmente y son iguales para cualquier petición, la información solo cambia si la página es actualizada; solo contienen texto, imágenes y multimedia.

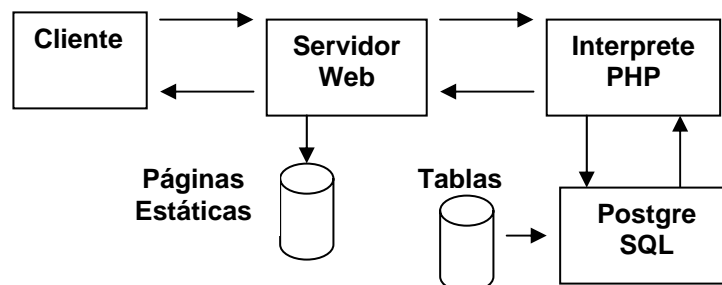
Figura 2. Esquema páginas estáticas



- **Páginas Dinámicas**

Las páginas dinámicas se crean en el momento justo de iniciar la petición, se generan diferentes resultados en las distintas peticiones; una aplicación Web está conformada por varias páginas dinámicas que se relacionan e interactúan para encontrar un objetivo.

Figura 3. Esquema páginas dinámicas



2.1.2 Interfaz de usuario

La interfaz de usuario (IU) es uno de los componentes más importantes de cualquier sistema computacional, pues funciona como el vínculo entre el humano y la máquina. La interfaz de usuario es un conjunto de protocolos y técnicas para el intercambio de información entre una aplicación computacional y el usuario. La IU es responsable de solicitar comandos al usuario, y de desplegar los resultados de la aplicación de una manera comprensible. La IU no es responsable de los cálculos de la aplicación, ni del almacenamiento, recuperación y transmisión de la información.

2.1.3 Arquitectura Cliente – Servidor

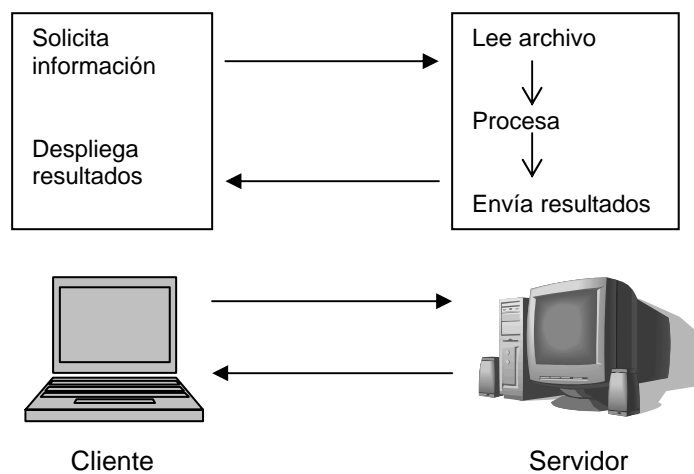
La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en que las transacciones se dividen en elementos independientes que cooperan entre si para intercambiar información, servicios o recursos.

En esta arquitectura la computadora de cada uno de los usuarios, denominada cliente, inicia un proceso de diálogo que produce una demanda de información o solicita recursos. La computadora que responde a la demanda del cliente, se conoce como servidor. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla según le convenga.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o una red mundial como lo es la internet. Cliente/Servidor es el modelo de interacción más común entre aplicaciones en una red.

La arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información; estableciendo así un enlace de comunicación transparente entre los elementos que conforman la estructura.

Figura 4. Arquitectura Cliente/servidor



2.1.3.1. Características de la Arquitectura Cliente Servidor

Entre las principales características de la arquitectura Cliente/Servidor, se encuentran las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, solo su interfaz externa.

- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Todos los sistemas desarrollados en arquitectura Cliente/Servidor poseen las siguientes características distintivas de otras formas de software distribuido:

Servicio. El servidor es un proveedor de servicios; el cliente es un consumidor de servicios.

Recursos compartidos. Un servidor puede atender a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.

Protocolos asimétricos. La relación entre cliente y servidor es de muchos a uno; los clientes solicitan servicios, mientras los servidores esperan las solicitudes pasivamente.

Transparencia de ubicación. El software Cliente/Servidor siempre oculta a los clientes la ubicación del servidor.

Mezcla e igualdad. El software es independiente del hardware o de las plataformas de software de sistema operativo; se puede tener las mismas o diferentes plataformas de cliente y servidor.

Intercambio basados en mensajes. Los sistemas interactúan a través de un mecanismo de transmisión de mensajes: la entrega de solicitudes y respuestas del servicio.

Encapsulamiento de servicios. Los servidores pueden ser sustituidos sin afectar a los clientes, siempre y cuando la interfaz para recibir peticiones y ofrecer servicios, no cambie.

Facilidad de escalabilidad. Los sistemas Cliente/Servidor pueden escalarse horizontal o verticalmente. Es decir, se pueden adicionar o eliminar clientes (con apenas un ligero impacto en el desempeño del sistema); o bien, se puede cambiar a un servidor mas grande o a servidores múltiples.

Integridad. El código y los datos del servidor se conservan centralmente; esto implica menor costo de mantenimiento y protección de la integridad de los datos compartidos. Además, los clientes mantienen su individualidad e independencia.

2.1.3.2. Componentes de la Arquitectura Cliente/Servidor

Cliente. Es la entidad por medio de la cual un usuario solicita un servicio, realiza una petición o demanda el uso de recursos. Este elemento se encarga, básicamente de la presentación de los datos y/o información al usuario en un ambiente gráfico.

Servidor. Es la entidad física que provee un servicio y devuelve resultados; ejecuta el procesamiento de datos, aplicaciones y manejo de la información o recursos. En el servidor se realiza el BACK END, que es la parte destinada a recibir las solicitudes del cliente y donde se ejecutan los procesos.

Por su parte los servidores realizan, entre otras cosas, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.

- Enlaces de comunicaciones con otras redes de área local o extensa.
- Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le responde proporcionándolo.

Frecuentemente el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en computadoras personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

2.1.4 Proceso Unificado de Desarrollo

Es un proceso de Ingeniería de Software, el cual sirve como marco de trabajo genérico que se puede especializar para el uso de diferentes sistemas de Software. Como todo proceso de desarrollo, se basa en los requisitos de usuario, y el producto final es un sistema.²

El proceso unificado se puede definir como forma disciplinada de asignar tareas y responsabilidades en un desarrollo de software, que puede asegurar la producción de software de calidad, dentro de plazos y presupuestos predecibles.

Las principales características del proceso unificado son:

- Dirigido por Casos de Uso: Debido a que la finalidad de un sistema es dar servicio a los usuarios (ya sean estas personas u otros sistemas), es imprescindible definir como se va a realizar la interacción entre el usuario y el sistema. Un caso de uso es una fracción de la funcionalidad del sistema, que

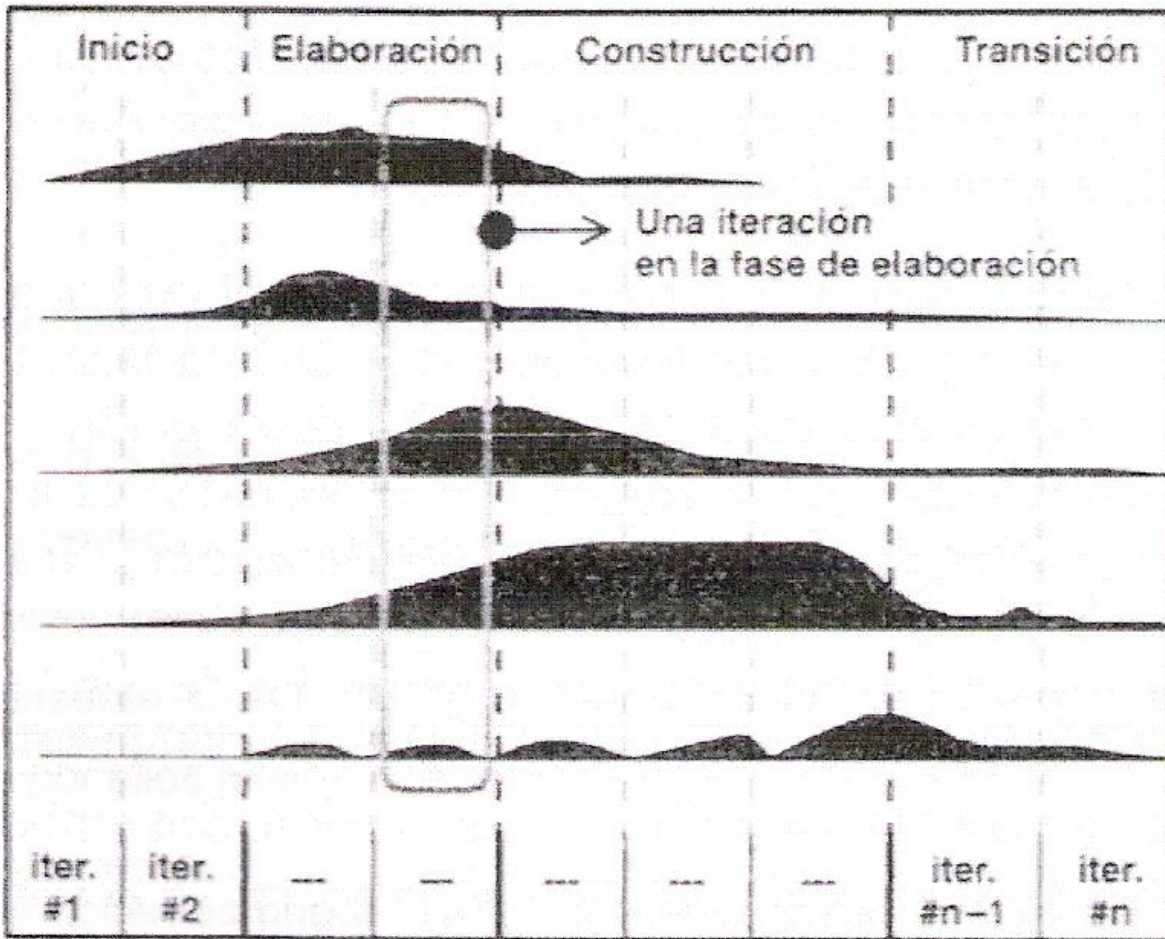
² Jacobson, Ivan. Booch, Grady. Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Primera edición. Adisson Wesley, España. 2000. Cáp. 1.

produce un resultado importante para el usuario. En cierta forma, se puede decir que los casos de uso representan los requisitos de funcionamiento del sistema, es decir, la respuesta a la pregunta: Que debe hacer el sistema para cada usuario?

- **Centrado en la arquitectura:** Al especificar la arquitectura del sistema, se obtiene una visión general del mismo, y se facilita su desarrollo. El concepto de arquitectura del sistema depende de muchos factores, entre ellos se pueden destacar las necesidades de la empresa, el sistema operativo sobre el que va a funcionar el software, el sistema de gestión de bases de datos y los protocolos de comunicación. El proceso unificado está centrado en la arquitectura, debido a que todo producto tiene tanto funcionalidad como forma. De tal manera, la funcionalidad está dada por los casos de uso y la forma por la arquitectura.
- **Iterativo e incremental:** El software moderno es complejo y novedoso, de modo que no es realista seguir un modelo lineal de desarrollo como el de cascada. Un proceso iterativo permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema. Teniendo esto en mente, es posible abordar las tareas más riesgosas primero, para así reducir los riesgos del proyecto y tener un subsistema ejecutable en corto tiempo.

El proceso unificado se divide en cuatro fases, en cada una de ellas los desarrolladores pueden descomponer adicionalmente el trabajo en iteraciones con sus incrementos resultantes.

Figura 5 Los cinco flujos de trabajo requisitos, análisis, diseño, implementación y prueba – tienen lugar sobre las cuatro fases: inicio, elaboración, construcción y transición



La Figura 5 muestra en la columna izquierda los flujos de trabajo. Las curvas son una aproximación de hasta donde se llevan a cabo los flujos de trabajo en cada fase; note que una iteración pasa por los cinco flujos de trabajo.

2.2 APRENDIZAJE COLABORATIVO EN ENTORNOS VIRTUALES

Las herramientas educativas que ofrecen los servicios de redes para apoyar procesos de enseñanza, posibilitan el desarrollo de actividades que favorecen el aprendizaje colaborativo, pues en este la adquisición de conocimientos se lleva a cabo con base en la interacción entre los miembros pertenecientes a una

comunidad de aprendizaje. Desde esta perspectiva, el aprender es un proceso de membresía por parte del individuo en una comunidad. Es un proceso de enculturación a través del cual el individuo se transforma en un miembro productivo de una comunidad específica. [Ayala, 2001.]

“comparando los resultados de esta forma de trabajo, con modelos de aprendizaje tradicionales, se ha encontrado que los estudiantes aprenden mas cuando utilizan el Aprendizaje Colaborativo, recuerdan por mas tiempo el contenido, desarrollan habilidades de razonamiento superior y de pensamiento crítico y se sienten mas confiados y aceptados por ellos mismos y por los demás”. [Millis, 1996.]

En todas las aulas, los docentes pueden estructurar sus actividades de manera tal que sus alumnos cumplan con alguno de estos tres criterios [Johnson]:

- Se involucren en una lucha de ganadores y perdedores para ver quien es el mejor(competencia.)
- Trabajen independientemente en sus propios objetivos de aprendizaje siguiendo su propio ritmo y en su propio espacio, para alcanzar un criterio preestablecido de excelencia (individualismo.)
- Trabajen de manera cooperativa en grupos pequeños, asegurando que todos los integrantes dominen los materiales asignados (cooperación.)

2.2.1. Aprendizaje Colaborativo vs. Aprendizaje Cooperativo

John Myers comenta: las definiciones del diccionario de “colaboración”, derivadas de su raíz latina, enfocan en el proceso de trabajar juntos; la palabra “cooperación” enfatiza el producto de tal trabajo.

Los dos tipos de aprendizaje tanto el Colaborativo como el Cooperativo contienen proyectos a realizar, grupo de trabajos, experiencias que compartir, pero la diferencia encontrada es que en el Aprendizaje Cooperativo el acercamiento sería más cuantitativo en donde se evalúa el producto de aprender; y en el aprendizaje Colaborativo se enfocaría cualitativamente ya que permite evaluar a los estudiantes en su papel activo en el aprendizaje, con cada uno de sus procesos, permitiendo además que el mismo grupo apruebe y revise el trabajo o producto realizado. Es responsabilidad del docente decidir cual estrategia de aprendizaje se va a utilizar dependiendo del enfoque y los objetivos del curso. Rockwood concluye: *“En mi experiencia del Aprendizaje Cooperativo representa un medio de acercamiento para la adquisición del conocimiento. Una vez los estudiantes se encuentren bastante versados, están listos para el Aprendizaje Colaborativo, para preparar, discutir y evaluar...”*

2.2.2. Grupos de Trabajo

Según Alexandre Dumas: *“el aprendizaje cooperativo toma muchas formas y definiciones, pero la mayoría de los acercamientos cooperativos involucran equipos pequeños, heterogéneos, normalmente de cuatro o cinco miembros, trabajando juntos hacia una tarea de grupo en que cada miembro es individualmente responsable para obtener un resultado que no puede completarse a menos que los miembros trabajen juntos; en otros términos, los miembros de un grupo son positivamente interdependientes.”*

El objetivo de los grupos de aprendizaje es lograr que cada integrante sea un individuo más fuerte. Los estudiantes aprenden juntos para poder desempeñarse mejor, luego, como individuos [Johnson y Johnson, 1994]; el aprendizaje colaborativo permite que el estudiante construya el conocimiento compartiendo sus opiniones, ya que la construcción de dos o más personas es más

enriquecedora y a su vez tiene la posibilidad de desarrollar valores como el compañerismo, la cooperación y el respeto.

El trabajo en grupo permite que los estudiantes analicen, expliquen entre sí y sintetizen el proyecto a realizar, de esta forma podrán desarrollar sus habilidades interpersonales las cuales contribuirán posteriormente a su quehacer profesional permitiéndole el trabajo en equipo, aprendiendo el uno del otro. De acuerdo con Johnson y Jonson, Sapon-Shevin, Ayres y Duncan, un grupo que trabaja bajo el enfoque del aprendizaje colaborativo ha de sustentarse en los siguientes principios: *(1) Los estudiantes se brindan ayuda y apoyo mutuo en el cumplimiento de las tareas y el trabajo hacia la obtención de metas comunes, (2) Las actividades colaborativas están basadas en habilidades interpersonales tales como: confianza mutua, comunicación clara y sin ambigüedades, apoyo mutuo y resolución constructiva de conflictos, (3) El grupo se somete a procesos de reflexión acerca de su proceso de trabajo y, a partir de ello, toma decisiones en cuanto a su funcionamiento, (4) El trabajo colaborativo es una expresión formalizada de los valores y las acciones éticas que imperan en una situación de enseñanza-aprendizaje, caracterizada por una comunidad de aprendizaje en la que se respeta la expresión de puntos de vista diferentes, (5) La formación de grupos es intencional y basada en la heterogeneidad. Los grupos se constituyen con base en las diferencias de habilidades, así como de características de personalidad y género de los estudiantes.*

Los grupos de trabajo deben crearse en ambientes abiertos y de confianza, de forma que los estudiantes se vean motivados a especular, innovar, preguntar y comparar ideas, conforme resuelven los problemas. El tamaño de los grupos de trabajo según Johnson y Jonson podrían tener entre dos y cuatro estudiantes, pero esa cantidad puede variar según las circunstancias y los objetivos específicos de cada lección.

2.2.3. Operaciones del pensamiento que apoyan el trabajo colaborativo

[Rthas] en su libro *¿Cómo enseñar a pensar?* Comenta las diferentes operaciones del pensamiento que pueden apoyar al estudiante en su participación en los grupos colaborativos. Entre estas podemos señalar:

- Comparar: examina dos o mas objetivos, ideas o procesos procurando observar cuales son sus interrelaciones. “! Si se les dio una tarea a un grupo de alumnos, es interesante comparar las comparaciones!; pues aprenden unos de los otros. Viendo como los demás reparan en semejanzas o diferencias que ellos pasaron por alto, su sensibilidad suele agudizar”.
- Resumir: es establecer de modo breve, condensado, la sustancia de lo presentado y replantear la esencia del asunto, de la idea o ideas centrales.
- Observar: vigilar, repara, notar, percibir. Observar es descubrir cosas, es parte de un proceso de reaccionar significativamente ante el mundo.
- Clasificar: es poner en orden en la existencia y contribuir a dar significado a la experiencia, encierra análisis y síntesis. Ordenar su mundo, pensar por si mismos, sacar sus propias conclusiones.
- Interpretar: cuando interpretamos una experiencia, explicamos el significado que ella tiene para nosotros. “Es un proceso por el cual damos y extraemos cierto significado de nuestras experiencias”.
- Formular críticas: la crítica permite abrir juicios, analizar y evaluar según ciertos principios y normas implícitas en nuestras observaciones.

- Búsqueda de suposiciones: una suposición puede ser cierta o probablemente cierta, no la sabemos con seguridad, de ahí la necesidad de “presuponerlo”.
- Imaginar: es percibir mentalmente algo no enteramente experimentado, se trata de una forma de creatividad.
- Resumir y organizar datos: plantea situaciones que obligan a pensar y que representan un verdadero reto para el estudiante.
- Formular hipótesis: una hipótesis es un enunciado que se propone como posible solución de un problema.

2.2.4. Cambio de la metodología tradicional al Aprendizaje Colaborativo

Cuando se decide apoyar el aprendizaje colaborativo, lo que antes era una clase ahora se convierte en un foro abierto al diálogo entre estudiantes y entre estudiantes y profesores, los estudiantes pasivos ahora participan activamente en situaciones interesantes. [Estrategias1], lo cual le permite una mayor motivación al estudiante para adquirir y construir grupalmente el conocimiento de un tema específico, compartiendo con sus compañeros experiencias, comentarios y críticas del proyecto a desarrollar.

Los elementos constituyentes del aprendizaje colaborativo según *Johnson y Jonson* son los siguientes:

- a) **Cooperación.** *Los estudiantes se apoyan mutuamente para cumplir con un doble objetivo: lograr ser expertos en el conocimiento del contenido, además de desarrollar habilidades de trabajo en equipo. Los estudiantes comparten metas, recursos, logros y entendimiento del rol de cada uno. Un*

estudiante no puede tener éxito a menos que todos en el equipo tengan éxito.

- b) **Responsabilidad.** *Los estudiantes son responsables de manera individual de la parte de tarea que les corresponde. Al mismo tiempo, todos en el equipo deben comprender todas las tareas que les corresponden a los compañeros.*

- c) **Comunicación.** *Los miembros del equipo intercambian información importante y materiales, se ayudan mutuamente de forma eficiente y efectiva, ofrecen retroalimentación para mejorar su desempeño en el futuro y analizan las conclusiones y reflexiones de cada uno para lograr pensamientos y resultados de mayor calidad.*

- d) **Trabajo en equipo.** *Los estudiantes aprenden a resolver juntos los problemas, desarrollando las habilidades de liderazgo, comunicación, confianza, toma de decisiones y solución de conflictos.*

- e) **Auto evaluación.** *Los equipos deben evaluar cuales acciones han sido útiles y cuáles no. Los miembros de los equipos establecen las metas, evalúan periódicamente sus actividades e identifican los cambios que deben realizarse para mejorar su trabajo en el futuro.*

2.2.5. Aprendizaje basado en la resolución de problemas

El aprendizaje basado en el método de resolución de problemas permite el análisis de situaciones reales las cuales presentan contextos, metas y obstáculos que conllevan al estudiante al desarrollo de nuevas estrategias para su solución, permitiendo de esta forma aumentar la creatividad y formar personas críticas, que no solamente analizan ejercicios o problemas típicos con un resultado correcto y

predeterminado. Los problemas complejos a utilizar deben conducir a la investigación y el desarrollo del pensamiento de los grupos de trabajo.

2.2.5.1. ¿Qué es un problema?

[García, 1998] define un problema como: “Una situación que presenta una oportunidad de poner en juego los esquemas de conocimiento, que exige una solución que aún no se tiene y en la cual se debe hallar la interrelación expresa y táctica entre un grupo de factores o variables, búsqueda que implica la reflexión cualitativa, el cuestionamiento de las propias ideas, la construcción de nuevas relaciones, esquemas y modelos mentales”.

2.2.5.2. Objetivos de la enseñanza problémica

(García, 1998) presenta los objetivos de la enseñanza problémica que son:

- El desarrollo de la creatividad.
- El desarrollo de la independencia cognoscitiva.
- La asimilación de sistemas teóricos y metodológicos de las ciencias.
- El desarrollo mismo de la capacidad para resolver problemas de manera creativa.

2.2.5.3 Fases para la resolución de problemas de tipo numérico

[García, 1998] define las siguientes fases para la resolución de problemas tipo numérico:

Fase	Descripción
Representación y replanteamiento del problema	Definición del problema, se establecen incógnitas y cantidades conocidas.
Presolución	Recolección de información, se establecen los principios físicos, planteando las posibles ecuaciones y una guía de cálculos a utilizar.

Resolución	Se definen las condiciones para las incógnitas, escritura de las ecuaciones en lenguaje algebraico. Posteriormente se sustituyen las cantidades numéricas y se efectúan los cálculos.
Revisión	Se verifica si la solución es razonable verificando las magnitudes y condiciones iniciales, identificar otros posibles caminos de solución.

2.2.6. Ambientes computacionales de aprendizaje colaborativo

Para la implementación de ambientes computacionales de aprendizaje colaborativo, los cuales tienen como propósito ayudar a la creación de una comunidad virtual según lo define Ayala: “El uso de la computadora como un dispositivo mediador que ayuda a que los aprendices se comuniquen, cooperen y colaboren a través de una red, asistiendo en su coordinación y en la construcción y aplicación del conocimiento, convirtiéndolos en miembros activos dentro de una comunidad virtual”.

Un ambiente computacional de aprendizaje colaborativo debe permitir que los participantes desarrollen las siguientes técnicas [Ayala, 2001]:

- Técnicas de comunicación y cooperación.
- Creación de nuevos conocimientos en conjunción con otros aprendices.
- Administración de los recursos de conocimiento compartidos.
- Influencia sobre otros participantes.
- Cuestionamiento, reflexión y discusión.

Ayala propone un ambiente “lifelong learning” definido de la siguiente manera: “Un ambiente basado en Internet que soporta el aprendizaje autodidacta, generativo e intencional, donde los elementos de software ayudan al aprendiz a planear sus

actividades de aprendizaje de acuerdo a las demandas de una sociedad cambiante, permitiéndole colaborar en la construcción social de conocimiento en una comunidad virtual". Este ambiente según Ayala debe incluir técnicas adicionales que permitan al estudiante:

- Asumir la responsabilidad de su propio aprendizaje, manteniendo un plan de aprendizaje actualizado.
- Aprender a su propia velocidad, a cualquier hora y en cualquier lugar.

3. DESARROLLO DE LA HERRAMIENTA

3.1 ANÁLISIS DE REQUISITOS

El estudio de la electrocardiografía en la UIS, actualmente, hace parte de la asignatura de electrofisiología, que incluye contenidos que pretenden enseñar a los estudiantes de medicina y paramédicas a leer electrocardiogramas. Complementario a esto, en la facultad de salud se dictan cursos extracurriculares para fortalecer el poco tiempo que se le dedica a esta área.

Para el estudio de los requisitos de esta plataforma educativa, se tuvo en cuenta una encuesta que se realizó a nivel nacional sobre “Situación del Médico General Colombiano Frente a la Lectura del Electrocardiograma”³, la experiencia del profesor de electrofisiología de la UIS y director del grupo EKG⁴, los integrantes del grupo EKG⁵.

Problemas Educativos

- Se dedica muy poco tiempo al estudio de la electrocardiografía.
- La realización de cursos extracurriculares, permite tener un mejor rendimiento en la lectura del electrocardiograma.
- La electrocardiografía exige estar en constante estudio, para obtener un mejor desempeño a la hora de realizar diagnósticos electrocardiográficos mas acertados y confiables

³ Estudio hecho por el Grupo EKG de la UIS, presentado en el XIX Congreso Colombiano de Cardiología del 27-30 de noviembre de 2001 en Cartagena y en el XVII Congreso Colombiano de Medicina Interna del 10-12 de octubre de 2002 en Cartagena.

⁴ Dr. Oscar Leonel Rueda Ochoa, Médico Internista de la UIS.

⁵ Grupo dedicado al estudio de la electrocardiografía, de la Facultad de Salud –UIS, dirigido por el Dr. Oscar Leonel Rueda

- La enseñanza de la electrocardiografía necesita de material didáctico para apoyar el aprendizaje de ciertos conceptos, difíciles de asimilar en el papel.
- Las mayores dificultades en la lectura del electrocardiograma se presentan al identificar en su orden las siguientes patologías: Bloqueos, Arritmias, Enfermedad Coronaria e Hipertrofias.
- Se observa un bajo nivel de conocimientos en electrocardiografía por parte del médico general Colombiano, evidenciado en el estudio hecho a nivel nacional, donde solo aprobó el 25%.
- Leer más de diez electrocardiogramas semanales mejora el nivel de autopercepción.
- En la práctica los médicos generales remiten al cardiólogo pacientes que fácilmente se puede identificar que no poseen enfermedades cardiacas.

Especificación de Requisitos

Elaborar una plataforma Web educativa para el aprendizaje de la electrocardiografía teniendo en cuenta que:

- Se necesita una clave de acceso que le permita al usuario hacer un seguimiento personalizado de sus logros en el estudio de la electrocardiografía.
- Debe tener un módulo de registro al iniciar la sesión si el usuario utiliza por primera vez la herramienta.

- Los diferentes temas que componen su estudio son: Historia, Anatomía, Electrofisiología, Electrocardiograma Normal, Arritmias, Bloqueos, Hipertrofias y Electrocardiograma Pediátrico.
- Cada lección debe tener su respectivo módulo de evaluación.
- Debe contener un módulo para observar el historial de las evaluaciones que el usuario ha desarrollado y sus respectivas calificaciones.
- Es necesario implementar un módulo de ayuda para guiar al usuario en la utilización de la plataforma.
- Debe tener una opción que permita al usuario realizar una evaluación combinada de las diferentes lecciones incluidas.
- Permitir al usuario determinar el número de preguntas que desea responder por evaluación.
- Las preguntas del cuestionario de evaluación deben presentarse en forma aleatoria.
- En el proceso de evaluación cuando un usuario selecciona una respuesta incorrecta debe recibir una observación con la respuesta correcta y una explicación de la misma.
- Integrar elementos multimedia para apoyar las estrategias educativas y así mantener motivados a los usuarios.
- El usuario pueda explorar la plataforma según su criterio y necesidad.

3.2 DISEÑO

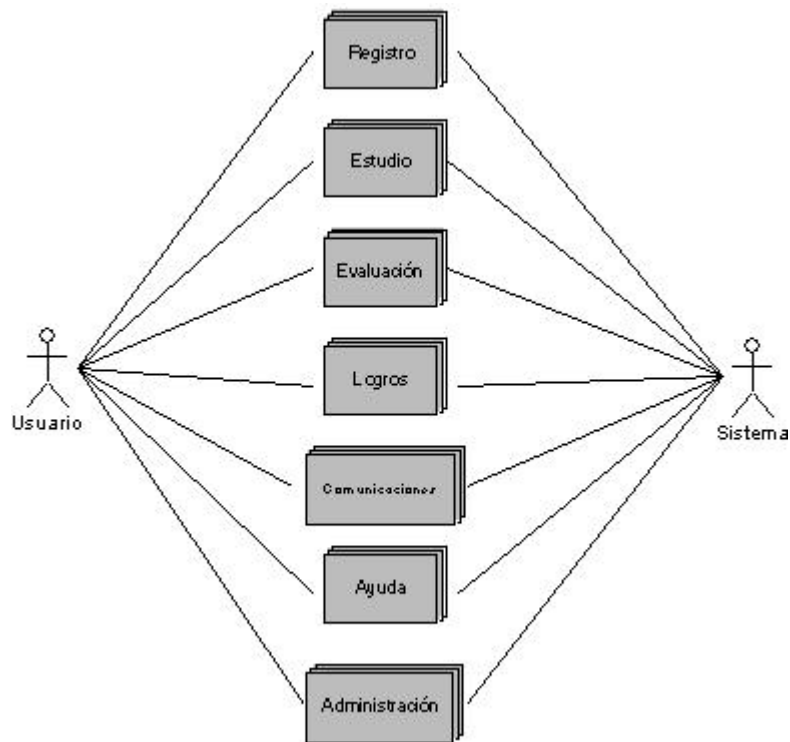
Como ya se dijo anteriormente, para el diseño de Siveuis se utilizó el Lenguaje Unificado de Modelado (UML). Inicialmente se hicieron los modelos desde el punto de vista del usuario y luego teniendo en cuenta la operatividad del sistema.

Se presentan los diagramas de Casos de Uso y de Actividad para visualizar y documentar el software.

3.2.1 Arquitectura del Sistema

La arquitectura de Siveuis plantea la necesidad de crear seis operaciones fundamentales que involucran al usuario del software, como el destinatario de todos los servicios que presta la herramienta; el software como tal, que se encarga de satisfacer los requerimientos del usuario.

Figura 6. Arquitectura del Sistema



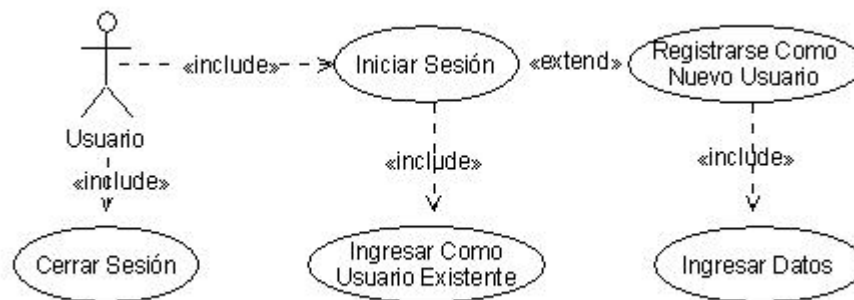
3.2.2 Módulos del Sistema

Siveuis se diseñó con base en siete módulos principales:

- Módulo de Registro
- Módulo de Estudio
- Módulo de Evaluación
- Módulo de Logros
- Módulo de Comunicaciones
- Módulo de Ayuda
- Módulo de Administración

3.2.2.1 Módulo de Registro

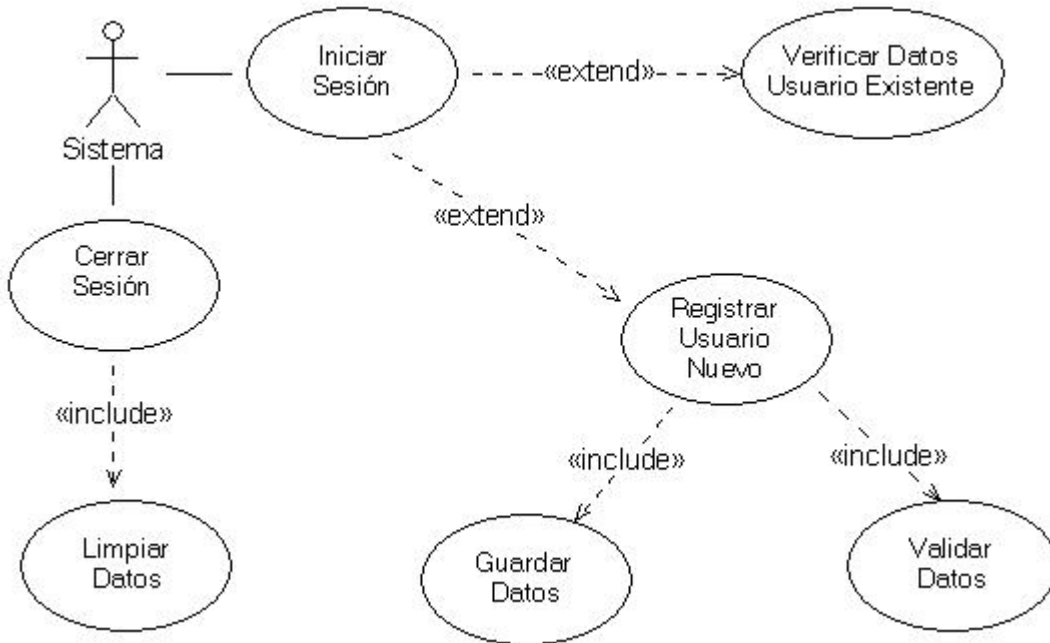
Figura 7 Diagrama de Caso de Uso, Modulo de Registro, Vista de Usuario



Cuando un usuario entre en el sistema debe pasar por el proceso de Registro, con el fin de habilitar su ingreso a la plataforma.

- **Iniciar Sesión:** permite ingresar a Siveuis, si es la primera vez que el usuario accede al sitio, debe **Registrar Como Nuevo Usuario** para determinar el Nombre de Usuario y la clave o password que utilizará en próximos inicios de sesión. En el caso de que el usuario ya se encuentre registrado en el sitio, solo debe ingresar su Nombre de Usuario y Clave para iniciar sesión de trabajo en Siveuis.
- **Cerrar Sesión:** este caso de uso termina la sesión del usuario que esta utilizando el sistema.

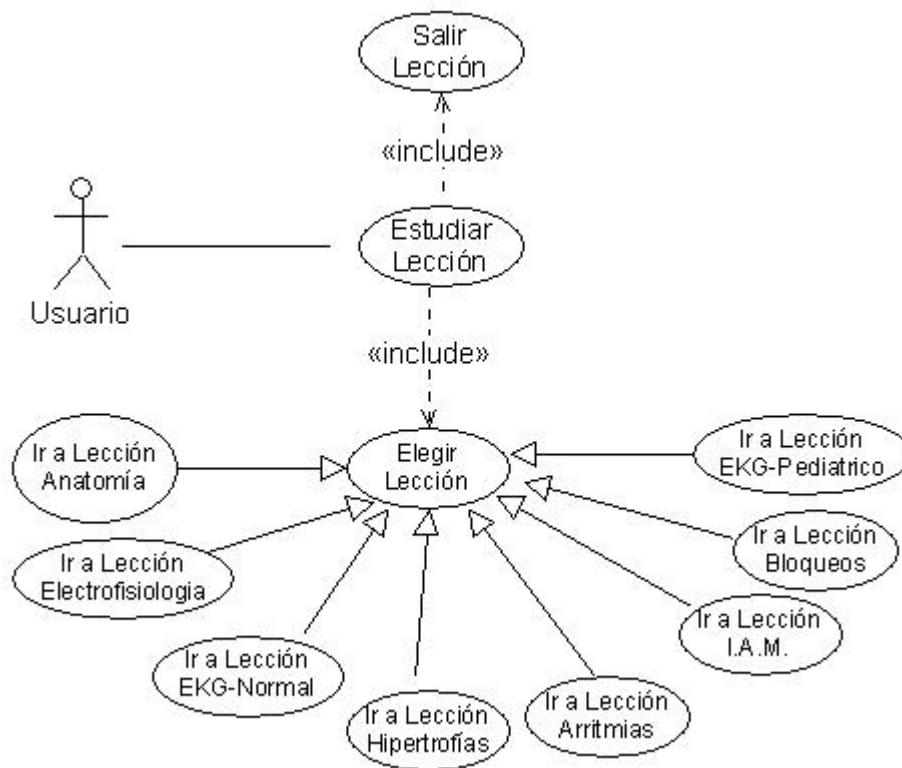
Figura 8. Diagrama de Caso de Uso, Modulo de Registro, Vista Software



En el caso del software cuando un usuario da **Inicio De Sesión** este debe o bien **Verificar** los **Datos Existentes** del usuario cuando este ya se ha registrado o iniciar el **Registro** de un **Usuario Nuevo**, para lo cual debe **Validar** los **Datos** del usuario cuando este los haya tecleado y proceder a **Guardar Datos** en el caso de que sean correctos. Por otra parte, cuando un se da orden par **Cerrar Sesión** se deben **Limpiar** todas las variables de usuario que se cargan al iniciar sesión.

3.2.2.2 Módulo de Estudio

Figura 9. Diagrama de Caso de Uso, Modulo de Estudio, Vista Del Usuario

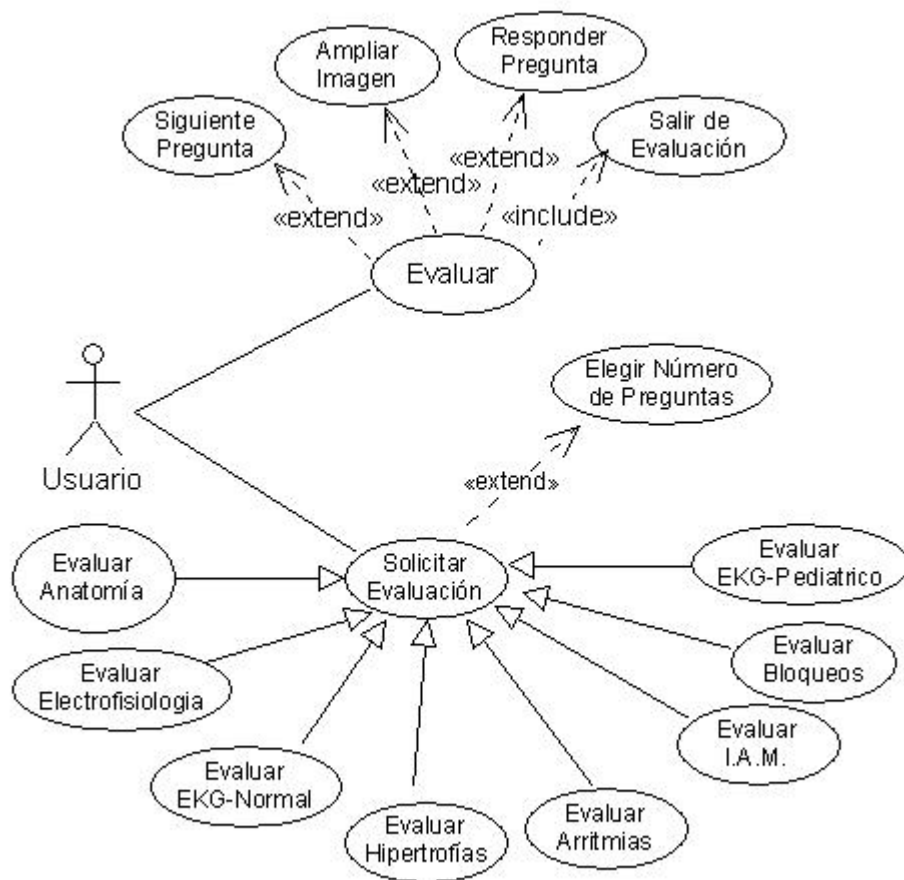


Cuando el usuario decide **Estudiar** una **Lección**, debe **Elegir** la **Lección** que desea estudiar, teniendo la posibilidad de escoger entre: Historia, Anatomía, Electrofisiología, EKG-Normal, Arritmias, I.A.M., Bloqueos, Hipertrofias y EKG-Pediátrico.

Las anteriores son las alternativas que encuentra el usuario a la hora de seleccionar una lección como material del estudio.

3.2.2.3 Módulo de Evaluación

Figura 10. Diagrama de Caso de Uso, Modulo de Evaluación, Vista Del Usuario



En la opción **Solicitar Evaluación**, el usuario puede elegir que tipo de evaluaciones desea realizar, las cuales pueden ser:

- **Evaluar EKG-Pediátrico:** este tipo de evaluación tiene como fin que el usuario examine los electrocardiogramas que se le muestran y determine la edad aproximada del paciente al que se le hizo el examen. Permitiendo que el usuario evalúe su nivel de aprendizaje en la lección de Electrocardiograma Pediátrico.

- **Teórica:** permite hacer una evaluación de *Historia, Anatomía y Electrofisiología*, que corresponden a las lecciones teóricas que se presentan en Siveuis.
- **Práctica:** esta opción da paso a una evaluación de los temas de valoración práctica del software: *Hipertrofias, I.A.M, Bloqueos y Arritmias*, es decir, las lecciones que enseñan a diagnosticar electrocardiogramas.
- **Elegir el Número de Preguntas:** permite al usuario escoger el número de interrogantes a resolver en su evaluación.

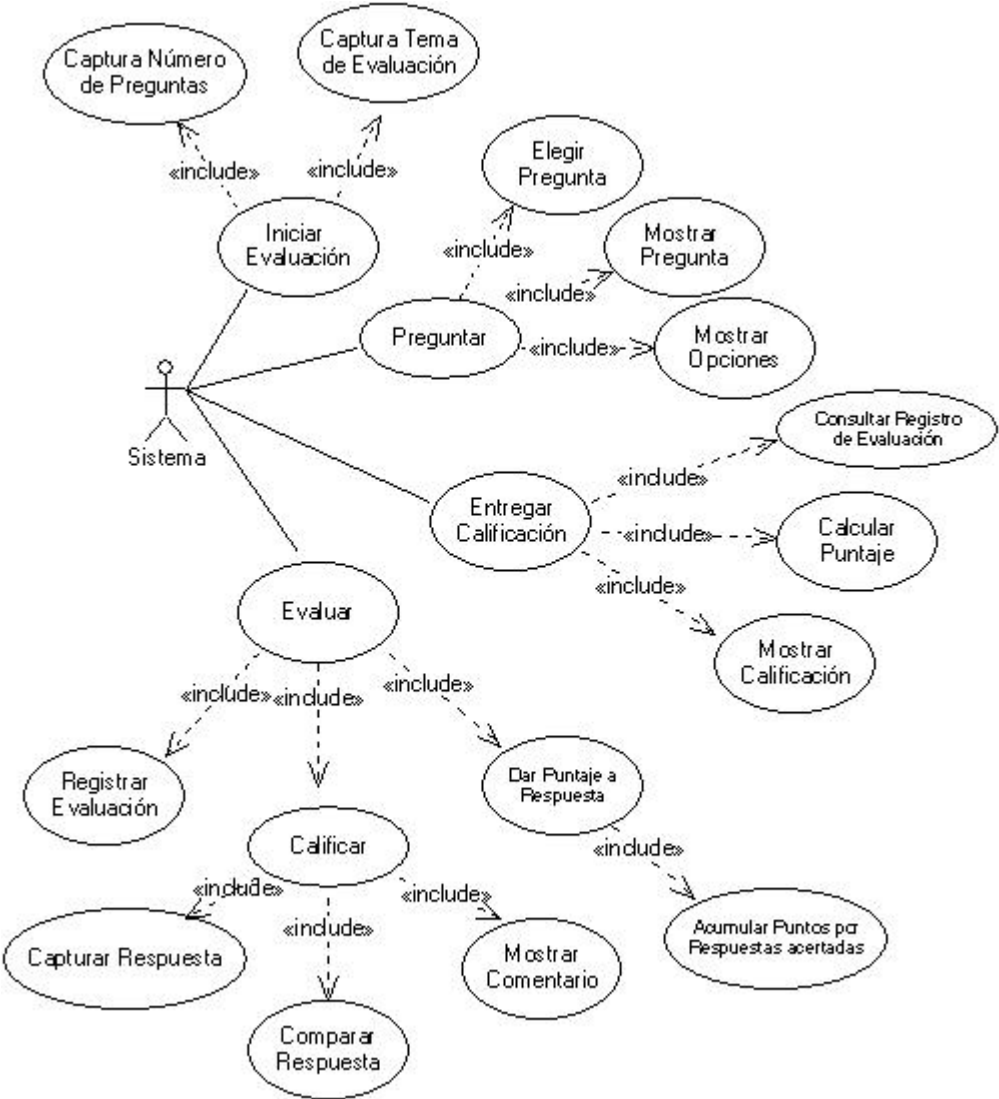
En el caso de uso **Evaluar**, el usuario puede:

- **Responder Pregunta:** como su nombre lo indica, le brinda la opción de responder a la pregunta que se le hace.
- **Ampliar Imagen:** permite ver en más detalle la imagen del electrocardiograma en cuestión, con el fin de que el usuario pueda hacer su respectivo diagnóstico.
- **Siguiente Pregunta:** esta opción es válida para el caso en que el usuario ya ha dado su diagnóstico y desea continuar con la siguiente pregunta del cuestionario.
- Por último tenemos el caso de uso **Ver Resultados** de la **Evaluación**, que será utilizado, cuando se ha finalizado una evaluación.

En el caso de uso de la figura 11 se muestra el comportamiento del sistema cuando se ingresa al módulo de evaluación.

Cuando el usuario inicia el proceso de evaluación, elige el tipo de evaluación que desea realizar y establece el número de preguntas a responder. Por tanto el software al **Iniciar una Evaluación** debe **Capturar el Tema de Evaluación** y el **Número de Preguntas**.

Figura 11 Diagrama de Caso de Uso, Modulo de Evaluación, Vista Sistema



El sistema es el encargado de **Preguntar**, para lo cual realiza los siguientes procesos:

- **Elegir una pregunta:** el sistema elige una pregunta en forma aleatoria de la Base de Datos, con el fin de que no siempre las evaluaciones sean las mismas.
- **Mostrar pregunta:** permite que el usuario visualice la pregunta que va a resolver.
- **Mostrar opciones:** muestra las posibles respuestas que el usuario puede dar a la pregunta que se le realizó y de las cuales debe escoger una para continuar con la evaluación.

Cuando el usuario da la orden de **Evaluar**, el sistema debe iniciar los siguientes procesos:

- **Calificar:** este caso de uso debe **Capturar** la **Respuesta** que dio el usuario, luego **Comparar** esta **Respuesta** con la correcta para mostrar al usuario un mensaje que le indica si la respuesta fue correcta o no.
- **Dar Puntaje a Respuesta:** para determinar el puntaje que obtiene un usuario en una pregunta, el sistema debe **Acumular Puntos Por Respuestas Acertadas**.
- **Registrar Evaluación:** luego de haber obtenido el puntaje alcanzado por el usuario en una pregunta, el software debe registrar en la Base de Datos el puntaje de la evaluación.

Finalizada una evaluación el usuario debe recibir un reporte de sus logros, por lo cual el software inicia el proceso de **Entregar la Calificación** y utiliza los siguientes casos de uso:

- **Consultar Registro de Evaluación:** el software consulta la Base de Datos para obtener los puntajes que el usuario alcanzó en cada una de sus lecciones.
- **Calcular Puntaje Total:** se calcula el puntaje total de la evaluación
- **Mostrar Calificación:** despliega los puntajes obtenidos.

3.2.2.4 Módulo de Logros

El Usuario de Siveuis puede ver los resultados de todas sus evaluaciones para determinar sus falencias y fortalezas. El caso de uso **Ver Logros** incluye **Ver Puntajes**, es decir, puede ver los resultados de las evaluaciones que ha presentado hasta ese momento.

3.2.2.5 Módulo de Ayuda

El módulo de ayuda le permite al usuario realizar una consulta sobre como utilizar de forma adecuada la herramienta. Brinda las opciones de:

- **Listar Contenido:** Muestra una lista de todos los temas contemplados en el módulo de ayuda.

3.2.2.6 Módulo de Comunicaciones

El módulo de comunicaciones contiene herramientas tales como Correo, Chat, Foro, Tablón Público y Tablón Privado. Tales herramientas permiten a los participantes de Siveuis la comunicación interna y promueve la discusión y/o debate de temas concernientes al curso; igualmente permite la publicación de mensajes tanto a los usuarios registrados como a los usuarios visitantes.

3.2.2.7 Módulo de Administrador

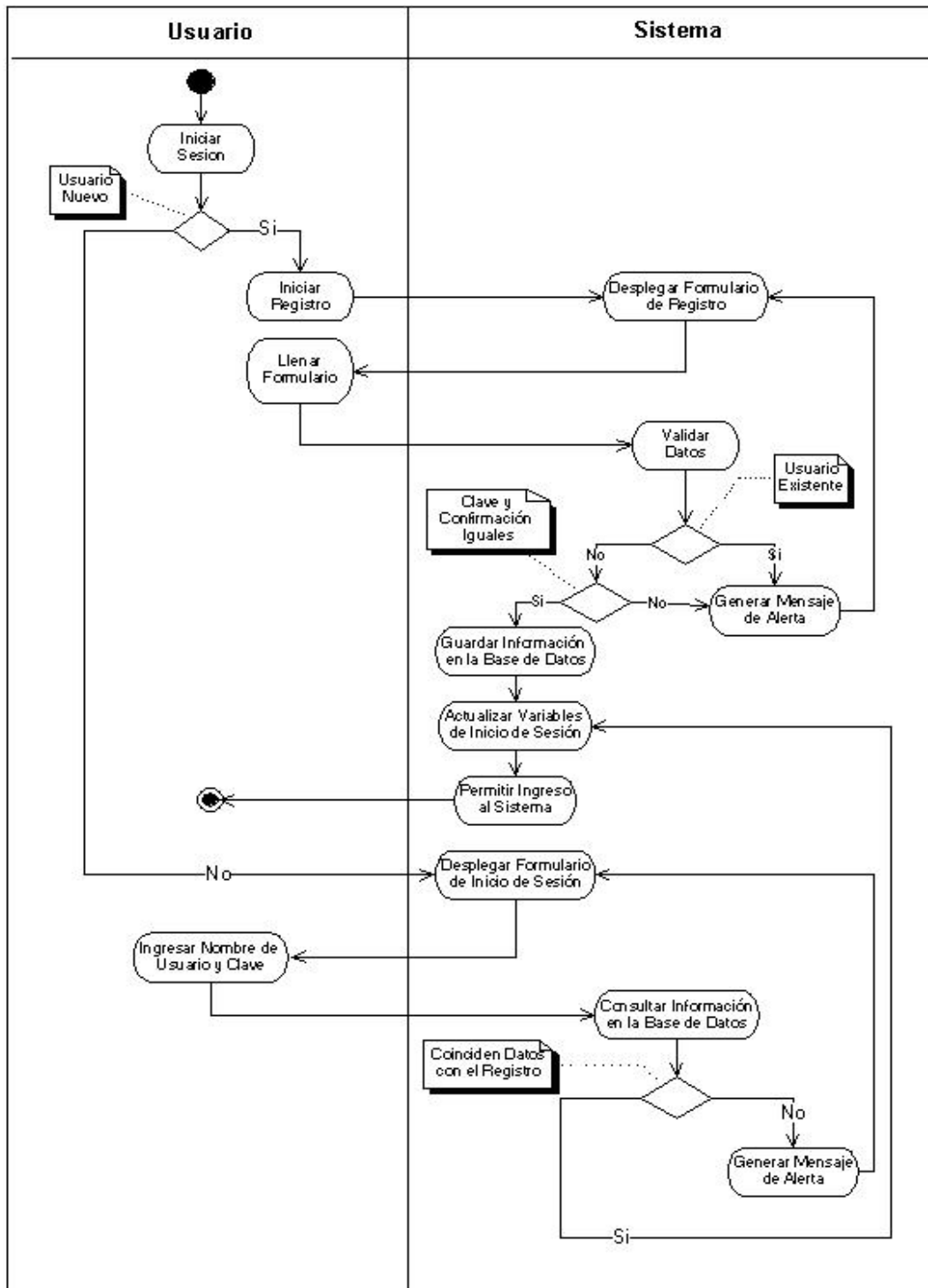
Esta parte de la plataforma es exclusiva para el administrador del sistema. Desde allí puede Validar usuarios nuevos, administrar el Foro, el tablón público y privado, alimentar y actualizar la base de datos tanto de las Evaluaciones como del EKG-semanal, monitorear el tráfico de Siveuis, actualizar los enlaces de interés y personalizar varios aspectos relacionados con Siveuis.

3.3 IMPLEMENTACIÓN

3.3.1 Diagramas de Actividad

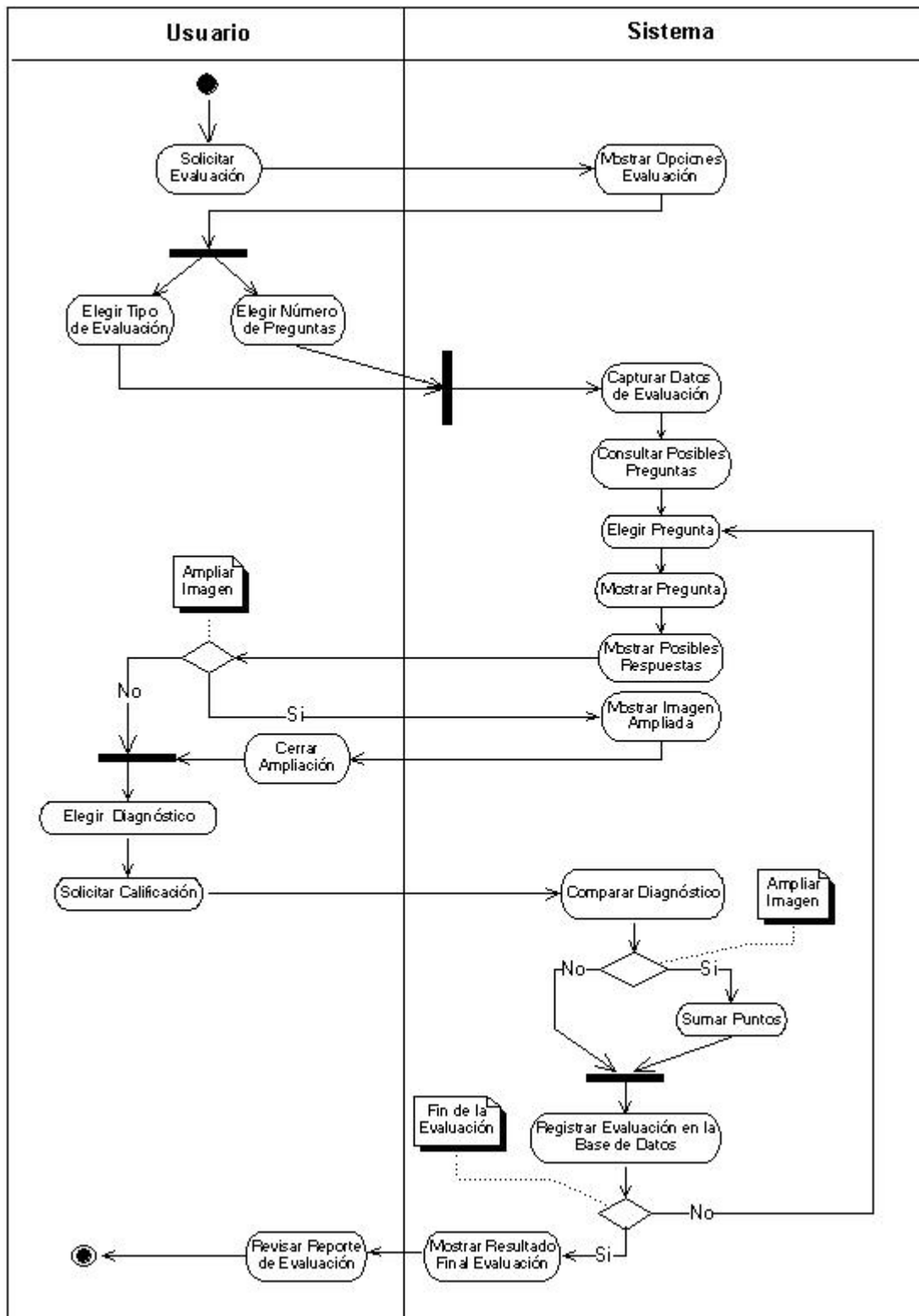
Modelo de Actividad para el Módulo de Iniciar Sesión

Figura 12 Diagrama de Actividad, Modulo de Iniciar Sesión



Modelo de Actividad para el Módulo de Evaluación

Figura 13 Diagrama de Actividad, Modulo de Evaluación



3.3.2 Aspectos de Optimización del Sistema

3.3.2.2 Optimización del Tiempo de Retorno

El Tiempo de Retorno (en adelante TR) es el tiempo que tarda el servidor en terminar de ejecutar los programas y entregar su resultado, lo cual influye directamente el tiempo que tarda el servidor en empezar a devolver los datos al navegador.

El tiempo que tarde el programa en terminar de generar todos los datos no solo influirá en la conexión con un usuario en particular, sino con el rendimiento de todo el sistema. A mayor TR, menor cantidad de conexiones simultáneas posibles y mayor carga de todo el sistema.

Para la optimización del TR este proyecto tratara dos aspectos:

1. El de disminuir el tiempo en el que se interpretan los programas.
2. Disminuir el trabajo de la cpu en los casos en los cuales la lógica del programa es muy compleja o se usan muchas llamadas de sistema.

Para afrontar el primer aspecto se debe tener en cuenta que para interpretar un programa, primero debe ser llevado de disco a memoria principal y que en algún instante puede estar sujeto a intercambio (un proceso inherentemente lento debido a que depende del tiempo que se tarda en mover los cabezales del disco).

La idea es eliminar estos dos procesos intentando mantener siempre los programas en memoria principal (RAM).

El segundo aspecto se tratara implementando una cache de código, que evitara interpretar mas de una vez el programa, es decir se interpretara una vez y se guarda la imagen del resultado en disco para una posterior utilización. Esta imagen también residirá en memoria principal para evitar el intercambio.

Para lograr mantener los programas en RAM se hace necesario trasladar a ella el árbol Web. Es decir tenemos que tratar la RAM como un dispositivo de bloques que permita escribir en el y tratarlo como un sistema de archivos normal.

El kernel de linux soporta una serie de sistemas de archivos que permiten trabajar con la RAM como si de un dispositivo de bloques se tratara, dentro de algunos estos sistemas de archivos tenemos ramfs, tmpfs y también se soportan los denominados ramdisk.

Decidimos trabajar con tmpfs puesto que permite especificar el tamaño desde la línea de comandos y solo usa una cantidad de memoria igual al tamaño de todos los archivos que residan en ella, es decir si se especifica un tamaño de 50 Mb pero el contenido total es de 43Mb únicamente se utilizara esa cantidad de RAM.

Adicional a ello, tmpfs no permite el intercambio, con lo que se debe hacer una estimación del crecimiento del sistema de archivos, pues el espacio reservado de RAM debe tener la capacidad suficiente para los programas con sus respectivos datos.

También es de notar que puesto que los datos están en memoria volátil, una vez se reinicie la maquina estos se perderán, lo cual nos enfrenta al problema de implementar alguna forma de guardar en disco los cambios efectuados sobre la imagen que reside memoria.

Se opto por implementar un demonio que este permanentemente vigilando el estado de la imagen y corra un script cada vez que sucede un cambio, este script sincroniza las imágenes utilizando rsync.

Para determinar cuando sucede un cambio se utilizo FAM, aplicación tipo cliente-servidor que permite registrar un archivo o directorio y emite un evento cada que sucede un cambio en el, luego nos interesan solo tres tipos de eventos:

- Creación
- Modificación
- Eliminación.

De esta forma se pueden tener sincronizadas los directorios (original e imagen) sin la intervención del administrador y asegurar que solo se sincronizara cuando es realmente necesario.

Para ver el detalle de la implementación refiérase al anexo A

La cache de código se implemento con TurckMMcache, el cual acelera la velocidad de ejecución de los scripts php, pues guarda una copia de ellos en estado 'compilado' con lo que se evita la sobrecarga por compilación.

El directorio de los scripts compilados reside en RAM con lo que ganamos aun más velocidad en su ejecución. TurckMMcache tiene soporte para sesiones permitiendo así diferenciar la dirección de destino de los scripts proporcionando una copia de los datos requeridos para cada navegador en particular y no provocara problemas de resultados si dos navegadores visitan la pagina al mismo tiempo(cada cual recibe sus propios datos).

TurckMMcache es un software que se compila e instala como un modulo de php, logrando que la aplicación se haga para todo los scripts que son interpretados sin necesidad de introducir cambios en ellos.

El tiempo de vida de los scripts 'compilados' es configurable, evitando así mantener imágenes muy antiguas de los scripts y además cada vez que un script es modificado se reemplaza la versión guardada con una nueva compilación.

Para ver el detalle de la implementación refiérase al anexo B.

3.3.2.2 Optimización de la Transferencia de Datos

El tiempo de visualización es el tiempo efectivo que el usuario tiene que esperar para ver el contenido completo de una página Web, como ya se mencionó es generalmente un factor muy importante a la hora de evaluar un sitio.

Logrando una disminución del TR mejoramos el rendimiento del servidor, disminuyendo el tiempo de visualización. Pero aun este se ve afectado por el ancho de banda de la conexión entre el cliente y el servidor, que generalmente es un factor externo difícil de manejar (a no ser que ambos se encuentran en una Intranet).

Pero aun se puede intentar hacer algo para mejorar esta situación. Una vez el servidor entrega los datos al navegador permanecen estáticos hasta que este los visualiza por completo, lo que permite manipularlos para tratar de aminorar su tiempo de descarga. Si los datos se comprimen lograremos tamaño mas pequeño de paquete con lo cual viajara mas rápido, dejando en manos del navegador la tarea de la descompresión. El proceso de descompresión es transparente al usuario y no repercute en la forma como se visualiza la información pues los datos se descomprimen de forma casi inmediata.

En conclusión podremos aprovechar mejor el ancho de banda y disminuir el tiempo de visualización de la información. Cabe anotar que la compresión solo se

logra con texto, no siendo posible en imágenes, sonido y video (flash), luego una página cargada con este tipo de elementos no se beneficiara de este proceso.

La compresión esta soportada por la especificación del protocolo HTTP 1.1 en el cual además se hace referencia a los métodos para compresión:

- gzip
- deflate⁶

A pesar de que la compresión HTTP esta definida desde la especificación 1.1 (1997) de este protocolo, hay en la actualidad algunos navegadores que no soportan (realmente muy pocos, en realidad solo versiones antiguas de estos), como es el hecho de IE 3 el cual es el navegador de Windows 95.

Luego es entonces importante determinar que navegadores soportan compresión (gzip o deflate), y para esto nos valdremos de la negociación de contenidos de las cabeceras HTTP, puesto que solo los clientes con soporte HTTP 1.1 envían peticiones de archivos en uno de estos formatos. La cabecera enviada para este objetivo es **Accept-Encoding**: que explícitamente nos informa sobre que tipos de codificación son aceptados por el navegador.

El efecto de la compresión funciona como en el siguiente ejemplo:

El Tiempo de descarga de una pagina esta dado por el número de bytes sobre la velocidad de conexión

$$TD = \text{\#bytes} / \text{\#kbs}$$

Si tomamos una página promedio de 8 KB (se elige 8 KB para el ejemplo pues en general una típica página Web en promedio tiene de 8 a 15 KB en texto excluyendo imágenes, animaciones, etc.) y una conexión de 32 kbps tenemos:

⁶ Formato de compresión de datos

Sin compresión:

$$TD = (8*1024 \text{ bytes}) / (32*1000/8 \text{ bytes/segundos}) = 2.048 \text{ segundos}$$

Con compresión:

$$TD = (4*1024 \text{ bytes}) / (32*1000/8 \text{ bytes/segundos}) = 1.024 \text{ segundos}$$

(Tiempos calculados ignorando la latencia de la red y cualquier espera introducida por el servidor Web y/o el navegador)

En este ejemplo suponemos un radio de compresión de 50%. Este radio depende del nivel de compresión que se elija, a un mayor nivel más ciclos de procesador, lo que tiende a sobrecargar el sistema.

Podría pensarse que el tiempo que se ahorra en descargar el archivo se invierte ahora en la compresión del mismo, pero en pruebas realizadas (un archivo de 4476 bytes compreso a un nivel 1 obtiene un radio de 76.77% y a nivel 9 obtiene un radio de 76.99%), al igual que en la literatura, se obtiene que el nivel de compresión y el radio de compresión no se comportan de forma directamente proporcional.

En conclusión para implementar la compresión HTTP no es necesario utilizar un nivel de compresión muy alto, por consiguiente el tiempo que se utiliza para lograr la compresión no introduce demoras significativas a la hora de descargar el archivo.

Para ver el detalle de la implementación refiérase al anexo C

3.3.2.3 Usuarios de Base de Datos Para el Correo

El sistema de correo de Siveuis estaba basado, como en la mayoría de los Unix tradicionales, en cuentas de usuario del sistema y con Sendmail⁷ como MTA. Este esquema aunque completamente funcional, tiene ciertos detalles que son susceptibles de mejorar.

El crear cuentas de usuarios del sistema genera archivos passwd y shadow grandes y difíciles de manejar (en el caso de autenticación contra archivos del sistema), demasiados directorios para spool de correo, entradas inútiles en /home (si se permite la creación de los directorios hogar), en fin no se justifica crear cuentas de usuario que en realidad nunca se van a usar.

Se decidió entonces utilizar bases de datos para la creación de los usuarios del correo (de forma similar a las cuentas de correo creadas en Hotmail y yahoo), con lo que conseguimos automatizar el proceso y embeberlo dentro de la creación de las cuentas para el curso sin necesidad de hacer llamadas al sistema (no se hace llamado a las herramientas del sistema para la creación de usuarios).

Para lograrlo solo se debió hacer que postfix⁸ y el teapop⁹ se comunicaran con el servidor de base de datos postgres.

Cabe anotar que para la primera vez que se hizo esta prueba se debió compilar postfix con soporte para postgres ya que por defecto no lo tiene, pero al finalizar

⁷ Sendmail: Es un popular MTA en internet. Se afirma de el que es el agente mas usado corriendo sobre sistemas UNIX, aunque se cuestiona su alto numero de alertas de seguridad (solucionadas a las pocas horas), además de no ser sencillo de configurar.

⁸ Servidor MTA modular, robusto y estable que presenta una alternativa a sendmail

⁹ Servidor POP3

este proyecto ya se consigue un modulo en formato rpm¹⁰ para esta integración, evitando así el proceso de compilar.

Para poder realizar esta tarea se debió cambiar los componentes prioritarios del antiguo sistema. Sendmail se cambio por postfix e imap se cambio por pop3, lo que derivo en el cambio del cliente MUA por uebimiau¹¹

El cambio de MTA fue aprovechado para implementar algunos aspectos de seguridad en el mismo. Postfix soporta dentro de su configuración las siguientes directivas de configuración:

- **smtpd_client_restrictions:** desde donde aceptar conexiones hacia nuestro servidor. se usa por defecto cuando se usa RCPT TO.
- **smtpd_sender_restrictions:** desde donde permitir comandos MAIL FROM:
- **smtpd_helo_restrictions:** que hosts estan permitidos para hacer HELO
- **smtpd_recipient_restrictions:** restringe los comandos RCPT TO:
- **smtpd_etrn_restrictions:** restringe que dominios pueden realizar comandos ETRN.

Además al igual que sendmail permite especificarle listas negras (rbl), con esto podemos lograr un simple control anti-spam.

Pero también se puede aprovechar la versatilidad de las expresiones regulares puesto que postfix tiene soporte para ellas. Solo es cuestión de agregar algunas directivas

¹⁰ Red Hat Package Manager: Manejador de paquetes de la distribución Linux Red Hat, Software para instalación y desinstalación de paquetes de Linux.

¹¹ MUA basado en Web, de rápida interfaz basada en javascript y desarrollado con plantillas.

- **body_checks:** son chequeos del cuerpo del mensaje. Para encontrar palabras claves en el cuerpo del mensaje se utilizan entradas de la siguiente forma:

[Expresión regular] [Acción]

Así podemos detectar palabras "interesantes" en el cuerpo del mensaje. Por ejemplo, si queremos que se rechacen los correos que contengan:

la palabra "hola"

las palabras "adios mundo cruel"

las líneas que terminen con "999"

Usamos la siguiente tabla de body_checks:

/hola/ REJECT

/adios\ mundo\ cruel/ REJECT

/999\$/ REJECT

- **header_checks:** Funciona exactamente igual que body_checks, con la diferencia que solo afecta a los encabezados (MAIL FROM:, RCPT TO: y demás).

De esta forma podemos detectar algunos mensajes indeseables y, volviendo un poco más complejas las expresiones regulares, uno que otro virus.

Esta protección aunque es eficiente carece de la robustez del software desarrollado para este fin, como es el caso de spamassassin¹² y ClamAV¹³.

¹² Software para detección y filtrado de spam

¹³ Software para la detección de virus

Por eso se decidió integrarlos con postfix para lograr un sistema de correo potente y seguro.

Para integrar ClamAV a postfix usamos la opción de filtrar contenido (**content_filter**), la cual esta ligada a un proceso “pipe”(poner referencia) lo que permite enviar mensajes “hacia afuera” de postfix. Decidimos usar este método para evitar la carga de otro proceso smtp trabajando y consumiendo memoria. El spamassassin lo integramos al script que hace el filtrado ClamAV.

Para mejorar el rendimiento usamos las versiones servidores de estos programas, esto evita cargar todas las definiciones de virus y reglas antispam para filtrar un solo email, pues simplemente se contactan los demonios y son ellos los que brindan la información.

Para ver el detalle de la implementación refiérase al anexo D.

3.3.2.4 Monitoreo y Análisis del Sistema

Una de las muchas tareas importantes que corresponden al administrar un servidor es la monitorización del sistema, es imprescindible conocer en todo momento qué está ocurriendo con el, con la red y así detectar cualquier problema que pueda surgir, esto es posible mediante el **Simple Network Management Protocol(SNMP)** y otras herramientas como **Multi Router Traffic Grapher(MRTG¹⁴)** que permiten obtener y visualizar información de numerosos parámetros del sistema.

Simple Network Management Protocol (SNMP) es un protocolo de gestión de red muy utilizado, que permite obtener información de dispositivos de red, memoria

¹⁴ Herramienta para supervisar la carga de tráfico en los enlaces de red, empleando páginas html.

libre, uso de la CPU, detección de errores, establecer alarmas, estado de funcionamiento, etc. Por ejemplo, en la gestión de un hub, SNMP podría desconectar automáticamente los nodos que estén corrompiendo la red, o se podrían establecer alarmas para alertar al administrador de la red cuando en un dispositivo el tráfico de datos supere el umbral establecido, o se podrían buscar IPs duplicadas, etc.

La mayoría de los fabricantes de dispositivos de red soportan SNMP, para ello unos agentes localizados en el dispositivo recogen la información y la registran en una base de datos en forma de árbol, llamada MIB (Management Information Base). Los MIB tienen un formato estándar, de forma que aún siendo de fabricantes distintos, las herramientas SNMP puedan obtener información del dispositivo.

El protocolo SNMP está formado por un agente que se instala en los nodos que se desean monitorizar y un gestor que se instala en el ordenador encargado de monitorizar la red. El gestor es el que obtiene la información de los agentes. El gestor solicita a los agentes información sobre los dispositivos gestionados, y los agentes responden a dicha solicitud. Esto último tiene una excepción, mediante el comando SNMP trap, los agentes pueden enviar datos no solicitados al gestor, por ejemplo cuando hay un fallo eléctrico.

SNMP funciona bajo TCP/IP, lo cual significa que desde un sistema central se puede gestionar cualquier ordenador de la LAN, WAN o internet.

MRTG usa SNMP para recolectar los datos de tráfico de un determinado dispositivo (routers o servidores). Los gráficos generados con MRTG, además de una vista diaria detallada, representan también el tráfico de los últimos siete días, las cuatro últimas semanas y los últimos doce meses. Esto es posible porque MRTG mantiene un archivo de todos los datos que ha obtenido del dispositivo de

red. Este archivo es consolidado automáticamente, así que no crece con el tiempo, pero contiene todos los datos relevantes del tráfico de los últimos dos años. Todo esto se realiza de una manera eficiente. Por lo tanto, usted puede monitorear 200 o más sistemas de red desde cualquier máquina Unix.

MRTG también puede usar scripts de shell, perl, php, etc. para recibir información del sistema, gracias esta opción se puede utilizar el poder de otras aplicaciones como es el caso de iptables¹⁵. Con iptables podemos establecer contadores para los servidores que deseamos monitorizar, en este caso lo usamos para obtener información del servidor Web(bytes de entrada y salida).

Para ver el detalle de la implementación refiérase al anexo E

3.3.2.5 Sistema de Alimentación Ininterrumpida SAI Administrable Vía Web

Un servidor que debe estar continuamente trabajando, debe tener un mecanismo para evitar caídas del mismo por falta de fluido eléctrico. Esto se implementa con colocar al equipo una UPS y configurar el software necesario para administrar el SAI.

Pero es deseable poder observar el estado del mismo sin necesidad de estar frente al equipo. Aprovechando que las aplicaciones Web están siendo mucho mas frecuentemente utilizadas y que es cada vez es mas el software que se migra a entorno Web es factible utilizar un software en entorno Web para administrar el SAI.

Cada SAI viene con un software de administración, pero generalmente estos funcionan solo para Windows. En el ambiente Unix/Linux existen dos utilidades

¹⁵ Firewall embebido en el kernel de Linux

para administración de SAI, una es el proyecto NUT¹⁶ y la otra es el software de APCTM¹⁷(parece ser el único fabricante que hace software para linux).

El proyecto NUT intenta crear un software administrado que sea compatible con la mayoría de SAI del mercado, suministrando drivers genéricos que funcionan para un particular tipo de SAI.

NUT se puede descargar en formato rpm o bien los fuentes para compilar.

Una vez instalado en el sistema se debe configurar y decirle que tipo de SAI va a usar y seleccionar el driver adecuado, en algunos casos también debe decirse con que tipo de cable se conecta el SAI al equipo(solo para conexiones por puerto serial).

En nuestro caso usamos un SAI APC Back-UPS Office de 350VA con un cable 940-0119A y conector rs232. Para comunicarse con este modelo se usa el driver genericups tipo 12(los tipos para el SAI se encuentran en la documentación proporcionada por NUT).

El tipo de cable que proporciona APC con este modelo tiene algunas debilidades puesto que no se implementa la conexión correcta para obtener información y para enviar comandos al SAI. El cable solo permite enviar la señal de apagado y reconocer si el SAI esta trabajando con la batería o con fluido eléctrico. A pesar de esto se configuraron los servicios Web.

Un vez configurado se puede obtener alguna información, pero debido a que la comunicación entre el equipo y el SAI esta restringida no se pudieron implementar los script necesarios para obtener y personalizar la información del SAI.

¹⁶ Network UPS Tools: Software genérico para administración de SAI

¹⁷ American Power Conversion: Fabricante de SAI

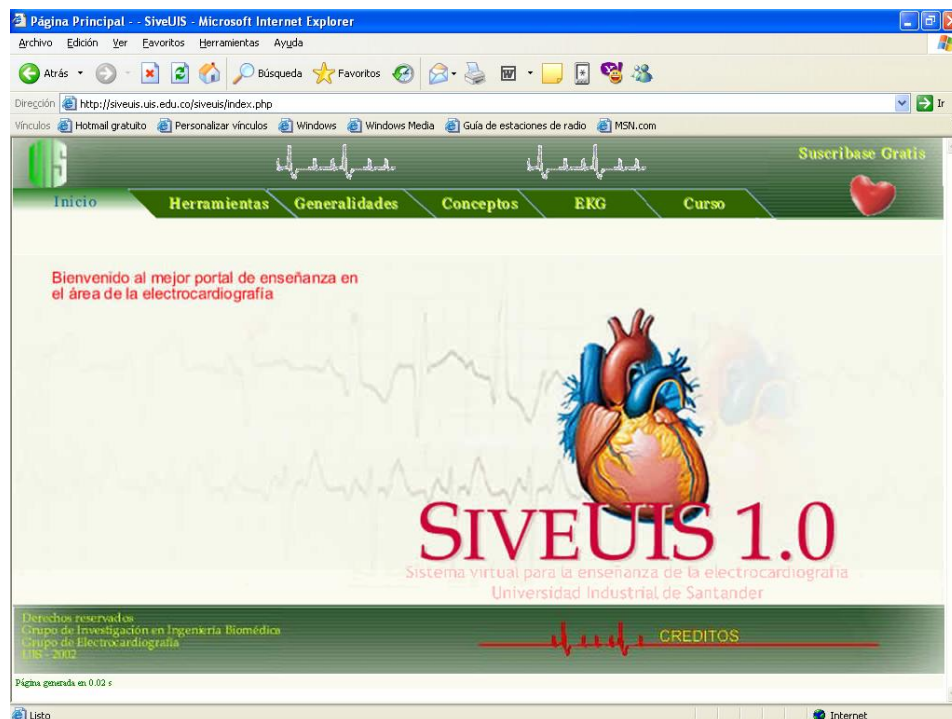
Pero la comunicación ofrecida por el cable es suficiente para asegurar que ante un fallo del fluido eléctrico la estabilidad del sistema no estará comprometida, pues se garantiza que la señal de poweroff(apagado) será recibida.

En conclusión un SAI administrable por Web es una realidad, de hecho ya existe software que lo implementa pero este depende en gran medida del hardware que se habilite, como es el caso de un SAI tipo USB. Pero el software que se consigue es demasiado general y no permite ningún grado de personalización, es por esto que si la comunicación es completa entre servidor y SAI(es decir se soportan todos los comandos) es preferible y factible crear scripts para esta administración.

3.3.3 Funcionamiento de Siveuis

A continuación se presenta una descripción de las principales funcionalidades de Siveuis.

Página Principal de Siveuis



Esta es la página inicial de Siveuis, allí se pueden encontrar las opciones de la parte pública del sitio aunque los usuarios registrados también pueden ser partícipes de ella. Dentro de sus aspectos encontramos: Buzón de Sugerencias, Tablón de Anuncios, Registro de Usuarios, Misión y Visión del Grupo, El Electro de la Semana (Sección donde semanalmente se presentará una patología diferente y los usuarios intentaran dar un correcto diagnóstico) y También se puede encontrar el primer capítulo del curso: Historia de la electrocardiografía.

Área Privada Siveuis

Index - SiveUIS 1.0 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección <http://siveuis.uis.edu.co/siveuis/privado/index.php> Ir

Vínculos Hotmail gratuito Personalizar vínculos Windows Windows Media Guía de estaciones de radio MSN.com

Cerrar sesión Herramientas Ayudas Contenidos Evaluación Personal SiveUIS 1.0

Bienvenido al mejor portal de enseñanza en el área de la electrocardiografía.

Con ayuda de esta sección usted podrá mejorar sus conocimientos en el área.

Para salir de esta sección y volver a la portada de nuestro sitio haga click en el boton Cerrar sesión.

Curso de Electrocardiografía

Derechos reservados
Grupo de Investigación en Ingeniería Biomédica
Grupo de Electrocardiografía
1997-2002

La Parte privada de Siveuis alberga varios menús tales como: Herramientas, Ayuda, Contenidos, Evaluación y Personal.

Menú Herramientas: Contiene las funciones Correo Electrónico, Foro de Discusión Web, Lista de Discusión por Email, Chat, Buzón de Sugerencias y Tablón de anuncios.

Menú Ayudas: Contiene las opciones Enlaces de Interés, Artículos y Documentos y Manual del Curso.

Menú Contenidos: Contiene las opciones Ejemplos Interactivos y Ejercicios.

Menú Evaluación: Contiene las opciones Evaluación Curso y Revisión de Notas.

Menú Personal: Contiene las opciones Directorio de Participantes e Información Personal.

Índice de temas del Curso de Electrocardiografía



The screenshot shows a Microsoft Internet Explorer browser window displaying the website for the 'Curso de Electrocardiografía'. The browser's address bar shows the URL 'http://sis25.uis.edu.co/siveuis/cursos/1/index.htm'. The website has a green and white color scheme with a navigation menu at the top containing 'Inicio', 'Herramientas', 'Generalidades', 'Conceptos', 'EKG', and 'Curso'. A 'Suscribirse Gratis' button with a heart icon is also visible. The main content area features the title 'Curso de Electrocardiografía' in a large, bold, green font. Below the title, there is a paragraph explaining the course's philosophy: 'La filosofía del curso es brindar al participante las herramientas básicas para la adecuada interpretación del Electrocardiograma y su aplicación clínica, generando un mejoramiento en el diagnóstico y tratamiento de las principales patologías cardiovasculares.' To the right of this text is a small graphic of a heart and a stethoscope. Below the philosophy text is a section titled 'INDICE GLOBAL' which lists nine topics: 2. ANATOMÍA DEL CORAZÓN, 3. ELECTROFISIOLOGÍA CARDIACA, 4. ELECTROCARDIOGRAMA NORMAL, 5. ENFERMEDAD CORONARIA, 6. ARRITMIAS CARDIACAS, 7. TRASTORNOS DE LA CONDUCCIÓN CARDIACA, 8. HIPERTROFIA CARDIACA, and 9. ELECTROCARDIOGRAFÍA PEDIÁTRICA. A short introductory paragraph on the right side of the page states: 'El curso está diseñado para iniciar desde los conceptos históricos de la Electrocardiografía, pasando por los conceptos de anatomía cardíaca básica, Fisiología cardiovascular, y generación de potenciales eléctricos cardíacos, hasta llegar a las claves diagnósticas y terapéuticas de los diversos trastornos eléctricos cardíacos.'

Entre los temas que comprenden el curso podemos observar: Historia de la electrocardiografía, Electrofisiología Cardíaca, Electrocardiograma Normal, Enfermedad Coronaria, Arritmias Cardíacas, Trastornos de la conducción cardíaca, Hipertrofia Cardíaca y Electrocardiografía Pediátrica.

Evaluación de EKG

Historia Tiempo restante para responder 2:22

Paciente de 19 años de sexo masculino quien luego de estar de pie durante una presentación escolar, presenta subitamente síncope con pérdida de la conciencia transitoria por 20 segundos y recuperación espontanea sin secuelas posteriores.

Isquemia Subepicardica Inferior

Bloqueo AV de segundo grado Mobitz I

Taquicardia Sinusal

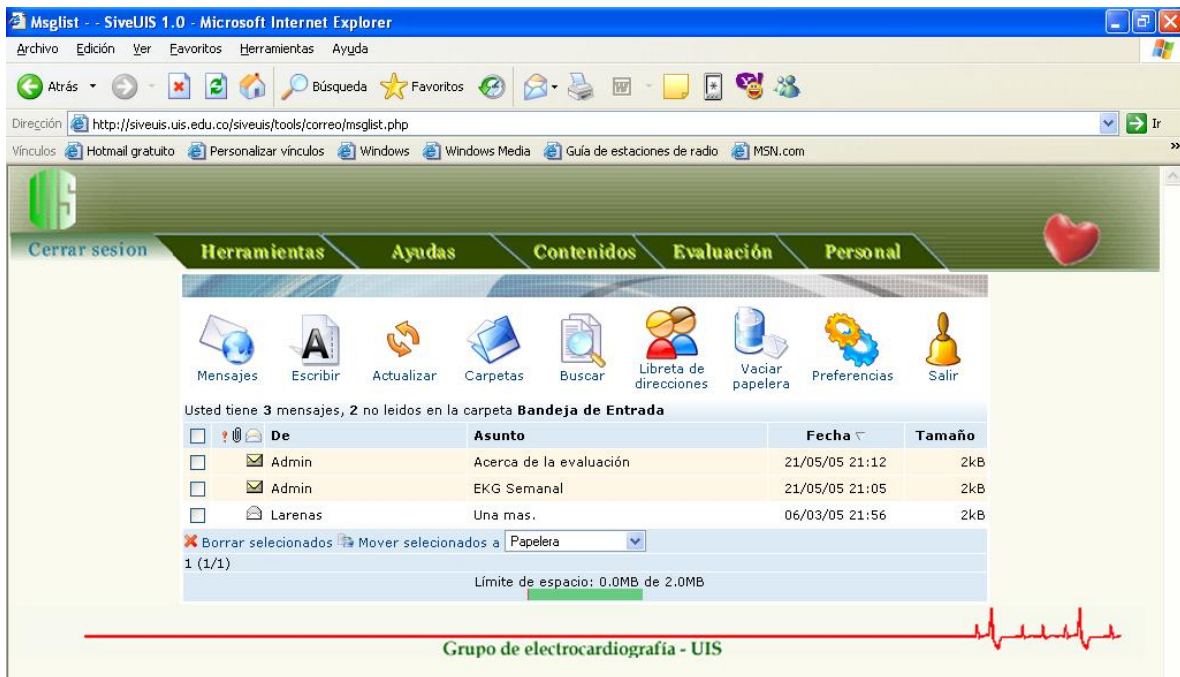
Electrocardiograma normal

Hipertrofia Ventricular Izquierda

El usuario va a encontrar evaluaciones de los distintos temas que componen el curso. Para determinadas evaluaciones va a encontrar graficas que puede ver en tamaño más grande con solo hacer un clic sobre esta. Otro factor que hay que tener en cuenta es el tiempo de respuesta del usuario a las preguntas ya que el

sistema solo permite tres minutos¹⁸ para responder determinada pregunta, después de pasado este tiempo el sistema ignora esta pregunta y busca aleatoriamente la siguiente.

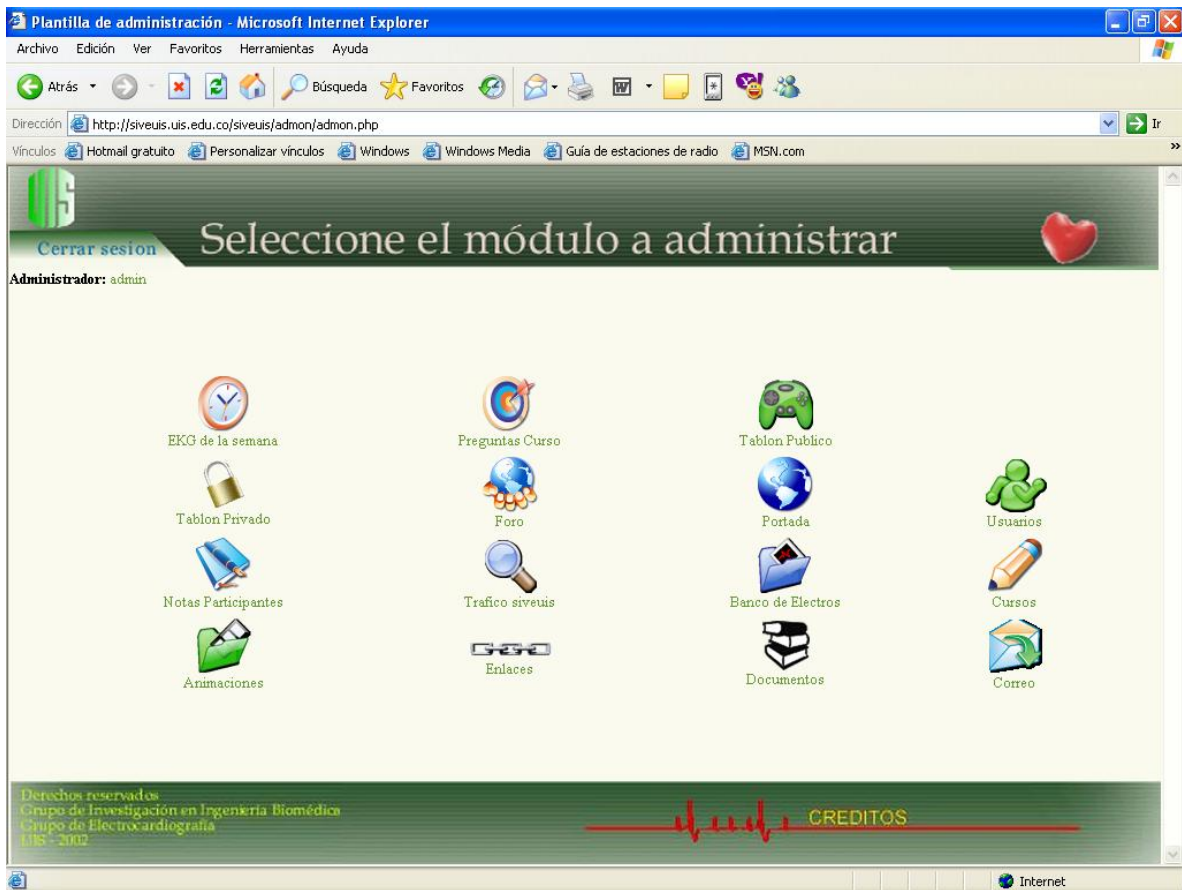
Correo Electrónico Siveuis



Mediante el Correo Electrónico de Siveuis se puede mantener la comunicación entre los usuarios ya que permite enviar correo a un integrante o a todos los integrantes del curso. Este sistema permite las opciones normales de un cliente de correo tales como: Administración de mensajes, Envío de archivos adjuntos, Creación de carpetas, etc.

Modulo Administración

¹⁸ Este valor es configurable por medio del Administrador del Sistema



El módulo administrar permite gestionar las diferentes particularidades de la plataforma incluyendo Administración de Usuarios, Administración de Contenidos, Administración del Foro, Monitoreo del Tráfico del Servidor, así como la administración de la base de datos de la evaluación.

4. CONCLUSIONES

- Utilizar un sistema operativo libre como lo linux y BSD presenta una serie de ventajas sobre utilizar software propietario, tales como mayor control del kernel, una gama mas amplia de aplicaciones disponibles para reemplazar un software especifico, no genera dependencia de versiones, el soporte es gratuito y muy diversificado, el sistema es realmente robusto y escalable si se esta dispuesto a invertir tiempo en configurar el SO y, sobre todo la facilidad de explorar y conocer aspectos realmente importantes de un sistema operativo.
- Proyectos OpenSource como lo son apache y php son herramientas suficientemente robustas para implementar aplicaciones, no necesariamente Web, de tipo comercial y educativo que satisfagan las necesidades del mercado, esto lo demuestra el hecho de que el uso de apache en la red Web sea de casi del 70%(cifras de mayo 2005).
- El implementar técnicas que intenten reducir la carga del servidor Web tanto en la generación como en la transmisión de los datos permiten optimizar los tiempos necesarios para acceder a una pagina Web, lo que demuestra que no solo se depende del ancho de banda sino también de la respuesta del servidor.
- A pesar de que la compresión HTTP puede reducir en promedio de un 65% a un 85% el tamaño de las páginas aumentando la velocidad de descarga de 3 a 6 veces, no es muy eficiente si estas están sobrecargadas de imágenes, audio o video, una tendencia bastante generalizada en el Web.
- El manejo de SNMP es casi que una obligación para un administrador de redes o servidores, puesto que permite obtener información detallada y prácticamente en tiempo real de casi todos los componentes del sistema permitiéndole administrarlos en determinadas ocasiones. Su configuración puede

parecer complicada, pero solo se debe conocer un poco las ramas MIB y configurar las comunidades. Para este fin en el mercado existen alternativas tanto libres como propietarias que brindan un frontend para su configuración y administración.

- Se desarrollo una herramienta que brinda los conceptos básicos para que un usuario aprenda a interpretar las principales patologías del electrocardiograma permitiéndole además establecer su nivel de percepción mediante un sistema de evaluación y realizar consultas eficientes sobre temas particulares relacionados con la electrocardiografía básica.
- El desarrollo de plataformas computacionales abren las posibilidades de enseñanza virtual, apoyan los procesos educativos como motores del desarrollo humano y ofrecen un amplio campo de aplicación en la Universidad.
- El proceso unificado de desarrollo de software ofrece un marco de trabajo para transformar las necesidades de los usuarios en un sistema software, proporcionándonos una guía ordenada para el desarrollo de las actividades y la posibilidad de incluir nuevos requisitos durante la elaboración del proyecto y que surgen cuando se realiza un trabajo interdisciplinario.
- El lenguaje unificado de modelado (UML) permite estructurar el diseño del software de una forma clara y precisa, describiendo la herramienta por medio de diagramas para obtener de esta forma beneficios en el tiempo de análisis y la eficiencia de los procesos.
- Las tecnologías multimediales de vanguardia incluidas en el desarrollo del software permiten presentar los contenidos de la herramienta de forma clara e ilustrativa para el usuario.

- Se creó un puesto de trabajo de tipo interdisciplinario para el grupo GIB Y ECG donde los integrantes administran el sistema, permitiendo la actualización y manejo de contenidos pedagógicos del curso

5. RECOMENDACIONES

- Obtener hardware de mejores especificaciones para aprovechar las ventajas de algunas implementaciones como el SAI y el manejo del servidor Web en RAM.
- Disminuir el promedio de imágenes que implementa la interfase de Siveuis y reemplazarlas por definiciones CSS, para disminuir aun más el tamaño de sus páginas y aprovechar la compresión HTTP. Esto lograría además una mejor integración con las resoluciones de los monitores actuales (típicamente 1024x768 y 1280x1024), evitando la creación de imágenes para suplir estas resoluciones y aumentar el peso del árbol Web.
- Implementar un sistema de sincronización para:
 - La base de datos mediante un cluster.
 - El árbol Web con rsync.

En un servidor de respaldo y utilizar alguna de las técnicas de balanceo de carga, para dividir las tareas del servidor principal, maximizando las capacidades del proceso de datos y de ejecución de tareas.

- El diseño de la herramienta permite que el usuario explore a su criterio el contenido de las diferentes lecciones que la componen, pero se recomienda realizar la navegación de los temas de forma secuencial, para garantizar un desarrollo estructurado del aprendizaje.
- Para enriquecer el contenido de este software educativo puede agregarse un módulo de tratamiento de afecciones cardiacas que complemente el estudio

que se hace sobre la identificación de patologías presentes en el trazado electrocardiográfico.

- En el contenido de la herramienta solo se han incluido las patologías que hacen parte de la electrocardiografía básica, se recomienda para futuros desarrollos los temas concernientes a enfermedades cardiacas avanzadas dirigidos a especialistas en la materia.
- Para el desarrollo de futuras versiones de este software, se puede aumentar las lecciones de electrocardiografía, para generar una versión avanzada del software.
- Recomendamos tener conocimientos previos en electrocardiografía básica, base de datos y herramientas multimedia para el desarrollo de futuras versiones del software.
- Para ampliar el banco de datos de este software existen dos aplicaciones complementarias, que facilitan el ingreso de nuevos electrocardiogramas problema, preguntas teóricas y términos para el diccionario. Estas aplicaciones no se instalan con el software y su uso se restringe al grupo de EKG de la Facultad de Salud UIS.
- En la Universidad debe promoverse entre los futuros profesionales el trabajo interdisciplinario, estableciendo estrategias para integrar los diferentes grupos de investigación con el fin de elaborar aplicaciones encaminadas a satisfacer necesidades computacionales en otras áreas del conocimiento.

6. GLOSARIO

CSS

Las hojas de estilo en cascada (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en html.

Cracker

Es alguien que viola la seguridad de un sistema informático para fines de beneficio personal o simple diversión. El termino fue creado por los hacckers alrededor de 1985 como defensa la mal empleo del termino hacker. Es una actividad ilegal.

Firewall

Básicamente un firewall se considera como cualquiera de los métodos de proteger una red, separándola de otra red en la cual no se confía. El mecanismo real por el cual este objetivo se cumple varía ampliamente, pero en principio, el firewall puede ser descrito como un par de mecanismos: uno que existe para bloquear el tráfico, y otro que existe para permitirlo. Algunos firewalls ponen mucho esfuerzo en denegar el tráfico, mientras otros lo ponen en permitirlo.

Hacker

Neologismo usado para referirse a un experto y apasionado en una o varias ramas de la computación y las telecomunicaciones.

HTML(HyperText Markup Language)

Lenguaje de marcado de hipertexto. Formato estándar de los documentos que circulan en la World Wide Web (WWW); se utiliza desde 1989.

HTTP(HyperText Transfer Protocol)

Protocolo de transferencia de hipertexto. Es el método utilizado para transferir ficheros hipertexto por Internet. En el World Wide Web, las páginas escritas en HTML utilizan el hipertexto para enlazar con otros documentos. Al pulsar en un hipertexto, se salta a otra página Web, fichero de sonido, o imagen. La transferencia hipertexto es simplemente la transferencia de ficheros hipertexto de un ordenador a otro. El protocolo de transferencia hipertexto es el conjunto de reglas utilizadas por los ordenadores para transferir ficheros hipertexto, páginas Web, por Internet.

IMAP(Internet Mail Access Protocol)

Este protocolo representa un mecanismo de comunicación para que los clientes de correo interactúen con servidores, y manipulen buzones. A pesar de que el protocolo de acceso a correo es el Post Office Protocol (POP), el cual también cubre las necesidades de acceso a correo remoto. IMAP ofrece un súper conjunto de las características de POP, ya que permite una interacción mucho más compleja y provee un modelo de acceso más eficiente que el modelo POP.

MTA(mail transfer agent)

MTA es el agente de transferencia de correo, un programa encargado de entregar un correo electrónico al destinatario, o bien de rutearlo a otro MTA.

MUA(mail user agent)

MUA es el agente de correo de usuario, un programa que maneja la recepción, composición y el envío de mensajes.

OpenSource

Código abierto (*open source* en inglés) es el término por el que se conoce al software distribuido y desarrollado en una determinada forma. Este término empezó a utilizarse en 1998 por algunos usuarios de la comunidad del software

libre, tratando de usarlo como reemplazo al ambiguo nombre original, en inglés, del software libre (*free software*).

Open Relay

Un MTA que permite el envío de correos de un determinado servidor o todos hacia cualquier dominio, susceptible de ser usado para enviar spam.

POP3

Post Office Protocol versión 3, protocolo para la gestión de correo en Internet.

RSYNC

Implementación de un algoritmo para replicar eficientemente un sistema de archivos.

SMTP(Simple Mail Transfer Protocol)

Protocolo simple de transferencia de correo, protocolo para la recepción y envío de mensajes entre MTA's.

SNMP(Simple Network Management Protocol)

Protocolo simple de administracion de redes, protocolo Internet para el control de redes y componentes.

Spam

Se considera mensaje spam al que está fuera de la temática del grupo, es decir que es "off topic" , y que además es comercial, por tanto el que lo envía está intentando sacar con él un beneficio económico.

Spammer

Persona que envía spam a los grupos de news o a los buzones de correo electrónico

SAI

Sistema de Alimentación Ininterrumpida, UPS en ingles, es un equipo y el software que le permitirán seguir trabajando con normalidad en caso de corte del suministro eléctrico. Además protegerá a sus equipos de subidas o caídas de tensión, variación de frecuencia, distorsiones armónicas, ruido eléctrico, y fluctuaciones que pueden causarles graves averías.

TCP/IP

Conjunto básico de protocolos de comunicación de Internet, que permiten la transmisión de información en redes de computadoras.

W3C

(World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los navegadores.

7. BIBLIOGRAFÍA

JHONSON W., DAVID. JHONSON T., ROGER. JHONSON, Edythe. Los nuevos círculos del aprendizaje colaborativo. Primera edición. Aique. Argentina, 1999.

FOWLER, Martín SCOTT, Kendall. UML Gota a Gota. Editorial Pearson Educación, 1999.

GALVIS PANQUEVA, Alvaro H. Ingeniería de Software Educativo, Santafé de Bogotá, Ediciones Uniandes, 1994.

JACOBSON, Ivar. BOOCH, Grady. RUMBAUGH, James. El Lenguaje Unificado de Modelado. Editorial Pearson Educación S.A. Madrid 1999.

MACROMEDIA. Utilización de FLASH™5. Primera Edición septiembre de 2000

RUEDA, Oscar Leonel. Programa de Entrenamiento en Electrocardiografía. Facultad de Salud – Universidad Industrial de Santander. Publicaciones UIS.

GUALDRON, Ana. MENCO, Ricardo. Software Educativo Multimedia para el Aprendizaje de la Electrocardiografía “Ekg-Tutor” – Universidad Industrial de Santander.

<http://es.wikipedia.org>

http://people.ee.ethz.ch/~oetiker/webtools/mrtg/es_es

<http://www.postfix.org>

<http://www.clamav.net>

<http://spamassassin.apache.org>

<http://www.apache.org>

<http://www.postgresql.org>

<http://www.toontown.org/teapop>

<http://www.linuxfocus.org>

<http://bulma.net>

<http://www.php.net>

ANEXOS

Anexo A. Optimizacion del Tiempo de Retorno

Observación. Estas instrucciones están basadas en **Linux Mandrake 10.0 Oficial**, pero llevarlas a otra distribución puede ser muy similar.

Llevar el árbol Web a disco RAM:

Decidir con que tipo de sistema de archivos se va a trabajar, en este caso se decidió trabajar con tmpfs, y verificar que el kernel implementa el soporte para este. Para verificar observamos el archivo especial filesystems que se encuentra en el directorio /proc.

Ejecute el siguiente comando:

```
cat /proc/filesystems
```

debe devolver una línea similar a esta (entre otras muchas).

```
nodev tmpfs
```

Si esta línea no aparece el sistema de archivos no es soportado, luego hay que recompilar el kernel con soporte para el tmpfs o elegir otro que este soportado.

Ubicar un punto de montaje o sea el directorio donde se alojara el sistema de archivos. En este caso elegimos /var/ramdisk/ .

Este directorio puede o no estar vacío, puesto que no se formateara no hay problema de pérdidas.

A partir de este punto debe trabajar como el súper usuario root.

Montar el tmpfs en el directorio elegido, sugerimos indicar el tamaño que tendrá el sistema de archivos, en este caso 70M, ya que por defecto al montar el tmpfs ocupa la mitad de la memoria total. Para montar ejecute

```
mount -t tmpfs -o size=70M tmpfs /var/ramdisk
```

Después verifique que efectivamente el tmpfs fue montado

```
df -h
```

debe obtener una línea similar a

```
tmpfs 70M          0          70M          0%          /var/ramdisk
```

Una vez montado debe hacer una copia del árbol Web, /var/www/ para nuestro caso, en el directorio recién montado. Para este propósito puede usar varias opciones tal como el cpio, tar o inclusive el mismo cp, pero recomendamos el tar por su fiabilidad (utilizado para respaldo en cinta) y simplicidad de ejecución. Puesto que nosotros tenemos un directorio de cache para los scripts compilados de php, omitimos su contenido

```
tar --exclude mmcach* -C /var/www/ -c . | tar -xC /var/ramdisk/
```

Esta instrucción ejecutara una copia de todo /var/www a /var/ramdisk, dejandolo idéntico al original.

Puede querer verificar que efectivamente se ha trasladado todo el contenido, ejecute

```
du -sh /var/www
du -sh /var/ramdisk
```

el resultado debe ser que reportan el mismo tamaño. Ahora debe decirsele al servidor apache que hay un nuevo documento raiz o arbol Web y que debe utilizarlo. Cambie la linea

```
DocumentRoot /var/www/html
```

en /etc/httpd/conf/httpd2.conf por

```
DocumentRoot /var/ramdisk/html
```

para dar soporte a la ejecución de scripts, busque las líneas donde se menciona la ruta /var/www y cámbielas por /var/ramdisk en este archivo y en /etc/httpd/conf/commonhttpd2.conf. Una vez haya esto hecho reinicie o inicie el servicio apache

```
apachectl restart
```

No debe haber problemas al arrancar, pero si sucede alguno lo mas seguro es que se cometi6 un error en la redacción de estos archivos, así que a revisar cuidadosamente.

En este momento ya debe estar funcionando el servidor Web en la imagen RAM, ahora falta lograr la sincronización.

Como ya se menciona, para lograr la sincronización utilizaremos FAM. Puesto que FAM es un servicio de rpc lo primero que debemos comprobar es que el servicio portmapper esta funcionando, esto es muy importante pues siempre debe empezar primero el rpc. Compruébelo de la siguiente forma

```
rpcinfo -p
```

debe obtener una lista de los servicios que esta usando rpc, incluido el mismo portmapper, de lo contrario obtendrá una línea similar a

```
rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused
```

si esto sucede inicie el servicio

```
service portmap start
```

una vez compruebe que esta funcionando puede seguir con el siguiente paso.

FAM es un servicio del súper servidor xinetd, entonces debe buscar el archivo fam en el directorio /etc/xinetd.d/, este archivo tiene una apariencia como esta (similar a la de todos los servicios de xinetd)

```
# default: on
# description: FAM is a file monitoring daemon. It can # be used to get reports
when files change.

service sgi_fam
{
    disable          = no
    type             = RPC UNLISTED
    rpc_version      = 2
    rpc_number       = 391002
    socket_type      = stream
    protocol         = tcp
    wait             = yes
    user             = root
    group            = nogroup
    server           = /usr/bin/fam
    bind             = 127.0.0.1
    flags            = NOLIBWRAP
}
```

de este archivo es importante que:

disable = no, para que arranque cuando arranque xinetd (activo)

bind = 127.0.0.1, para que solo escuche por la interfase local, que no escuche peticiones de otras maquinas.

Los demás parámetros indican la configuración del servicio. Ahora puede reiniciar el servicio xinetd para que FAM también se inicie, una vez reinicie compruebe que FAM arranco

```
rpcinfo -p
```

debe obtener una línea similar a

```
391002 2 tcp 32895 sgi_fam
```

si esto sucede FAM esta listo para registrar directorios.

Es momento de iniciar el cliente. Este cliente esta diseñado para registrar un directorio y solo reportar algunos eventos.

Veamos un poco de FAM.

FAM, el Monitor de Alteración de Archivos, provee un API que las aplicaciones pueden usar para ser notificadas cuando archivos o directorios específicos han cambiado. FAM viene en dos partes: fam, el *demonio* que escucha las peticiones y envía las notificaciones, y libfam, una librería que la aplicación cliente puede usar para comunicarse con FAM.(<http://oss.sgi.com/projects/fam/faq.html>).

El demonio FAM envía los siguiente eventos cuando algo sucede en el directorio registrado

FAMChanged: Un archivo es cambiado o actualizado.

FAMDeleted: Un archivo es borrado.

FAMStartExecuting: Un programa empieza la ejecución con este como el directorio de trabajo.

FAMStopExecuting: Termina la ejecución de un programa en este directorio.

FAMCreated: Archivo creado en el directorio.

FAMMoved: Este evento realmente nunca ocurre.

FAMAcknowledge: Ruta errónea o el monitor termino.

FAMExists: Registra los archivos que debe monitorizar.

FAMEndExist: Una vez se termina de registrar el elemento.

Basados en estos eventos la aplicación cliente que se construyo (en realidad se modifiko un programa de la documentación del sitio oficial de fam), captura FAMChanged, FAMCreated, FAMDeleted y hace un llamado al sistema para ejecutar el script :

```
/root/bin/save_www
```

encargado de mantener sincronizados los directorios. Para iniciar el cliente ejecute

```
/root/bin/watch_dir /var/ramdisk
```

una vez lo ejecute este le dirá que la conexión se hizo y que se registro el directorio, a partir de este momento se garantiza que si un archivo o directorio es actualizado, borrado o creado en RAM, se hará lo mismo en el disco duro.

Crear o verificar que el script `save_www` existe. Este es un script de shell que realiza un llamado a `rsync` (Tiene muchas ventajas: posibilidad de descubrir diferencias entre las copias local y remota y bajar solo las diferencias, recuperación de una copia parcial, compresión de los datos enviados, cifrado de datos (usando **ssh**) y facilidad de uso y configuración, entre otras.) para mantener las copias sincronizadas. Se ejecuta de la siguiente forma:

```
rsync -auz --exclude=tmpcache /var/ramdisk/ /var/www
```

esta instrucción le dice a `rsync` que sea recursivo, preserve enlaces blandos, preserve permisos, preserve la fecha, preserve dispositivos (solo root), preserve el grupo, comprima y que conserve archivos creados en el directorio destino que también existan en el fuente pero que su fecha sea posterior a la de los del fuente.

Después de ejecutar estos pasos se debe tener un servidor ejecutándose en RAM y sincronizado con una imagen de el en disco duro.

Para automatizar estas tareas se crearon los siguientes script de shell:

```
/root/bin/ramdisk
```

```
#!/bin/sh
#Script encargado de iniciar, terminar y reportar el estado de la ejecución
#del servidor web en RAM.
#Recibe como parametros:
#  iniciar - Iniciar la ejecución
#  terminar - Terminar la ejecución
#  estado - Reportar si esta o no ejecutandose y desde cuando
function Resultado(){
    if [ $? -eq 0 ]
    then
        echo -e " [\033[0;32mbien\033[0m]";
        return 0
    else
        echo -e " [\033[0;31mfallo\033[0m]";
    fi
}
```

```

        echo "Abortando ejecución";
        if [ $# -eq 0 ]
        then
            Terminar 1
        fi
        exit
    fi
}
function Iniciar(){
    #Montar la particion en el punto seleccionado con tmpfs
    echo -ne "Iniciando disco RAM"
    /bin/mount -t tmpfs -o size=70M tmpfs /var/ramdisk &>/dev/null
    Resultado
    #Descomprimir la imagen de www en el directorio
    echo -ne 'Sincronizando directorios'
    /bin/tar --exclude=mmcach* -C /var/www/ -c . | /bin/tar -xC /var/ramdisk/
&>/dev/null
    Resultado
    #Copiar los archivos de configuracion nuevos de apache
    echo -ne "Cargando configuración servidor web"
    /bin/cp -f /etc/httpd/conf/httpd2.conf.ram /etc/httpd/conf/httpd2.conf && \
    /bin/cp -f /etc/httpd/conf/commonhttpd.conf.ram
/etc/httpd/conf/commonhttpd.conf
    Resultado
    #Reiniciar apache
    echo -ne "Reiniciando servidor web"
    /usr/sbin/apachectl restart &>/dev/null
    Resultado
    #Iniciar vigia del directorio
    echo -ne "Iniciando demonio de sincronización"
    /root/bin/watch_dir /var/ramdisk/ &>/dev/null &
    Resultado
    sleep 10s
    /bin/date > /var/run/webram
    return 0
}
function Terminar(){
    #Restaurar configuracion inicial apache
    echo -ne "Restaurando configuración inicial servidor web"
    /bin/cp -f /etc/httpd/conf/httpd2.conf.normal /etc/httpd/conf/httpd2.conf && \
    /bin/cp -f /etc/httpd/conf/commonhttpd.conf.normal
/etc/httpd/conf/commonhttpd.conf
    Resultado 1
    #Reiniciar servidor configuracion original

```

```

echo -ne "Reiniciando servidor web"
/usr/sbin/apachectl restart &>/dev/null
Resultado 1
#Ejecutar rsync para sincronizar
if [ $# -eq 0 ]
then
    echo -ne "Sincronizando..."
    /root/bin/save_www
    Resultado 1
fi
#Terminar demonio de sincronizacion
echo -ne "Terminando demonio de sincronizacion"
/usr/bin/killall watch_dir
Resultado 1
#Desmontar disco ram
echo -ne "Terminando disco RAM"
/bin/umount /var/ramdisk
Resultado 1
/bin/rm -f /var/run/webram
return 0
}
function Estado(){
    if [ -f /var/run/webram ]
    then
        echo -ne "Web ram funcionando desde: "
        /bin/cat /var/run/webram
    else
        echo -e "Web ram detenido"
    fi
}
function Uso(){
    echo -e "Uso: $0 iniciar|terminar|estado"
}
case "$1" in
    'iniciar')
        Iniciar
        ;;
    'terminar')
        Terminar
        ;;
    'estado')
        Estado
        ;;
    *)

```

Uso

```
;;  
esac  
  
/root/bin/save_www  
  
#!/bin/sh  
#Script para realizar la sincronizacion y reportarlo en un archivo  
#si se pudo llevar a cabo  
dir="/var/ramdisk/";  
dst="/var/www";  
comando="rsync -aуз --exclude=tmpcache $dir $dst";  
fecha=$(date +"%Y-%m-%d %H:%M");  
if ($comando)  
then  
    echo "Se ha creado una copia de seguridad de $dir en $fecha" >>  
    /var/log/wwwramdisk;  
fi
```

Anexo B. Cache de Código

Descargar el paquete con los fuentes de TurckMMcache de <http://turck-mmcache.sourceforge.net>, en formato tar.bz2, descomprimirlo en un directorio temporal

```
tar jxf turck-mmcache-2.4.6.tar.bz2 -C /tmp
```

TurckMMcache viene para compilar, ya que es necesario crear los módulos de acuerdo a la versión de php y apache que se tenga instalada. Para esta versión se requiere apache 1.3, mod_php 4.1, autoconf, automake, libtool, m4 que en una instalación normal de Mandrake 10.0 se instalan por defecto.

Se requiere de php-devel que son las herramientas de php para crear módulos, este paquete por lo general no se instala porque viene en el cd 4 de la distribución, cd que no se consigue en la versión download. Entonces tendremos que conseguirlo en la red en formato rpm o bajar los fuentes de php y compilarlo como un módulo (TurckMMcache solo funciona cuando php se ha compilado como módulo de apache), de esta forma las herramientas también se instalan.

Para este caso específico necesitamos php-devel-4.2.3-10 que conseguimos en <http://rpmfind.rediris.es>. Una vez lo conseguimos procedemos a instalarlo (como root).

```
rpm -Uvh php-devel-4.2.3-10.i686.rpm
```

Entrar en el directorio, creado al descomprimir y desempaquetar los fuentes de TurckMMcache, y ejecutar las siguientes instrucciones

```
phpize (para crear el configure a partir de ac4)
./configure --enable-mmcache=shared --with-phpconfig=
$PHP_PREFIX /bin/php-config * (habilitar memoria compartida)
make (compilar)
make install
```

después de esto se instala un módulo llamado mmcache.so en /usr/lib/php/extensions, el cual es el encargado de llevar a cabo la "compilación" de los scripts php. Si sucede algún problema en el proceso anterior, lo más seguro es que sea por dependencias, habrá que resolverlas.

4- Una vez que se construye el módulo hay que configurar php para que lo cargue y utilice. Esto se hace en el archivo /etc/php.ini, buscar la sección correspondiente a los módulos y agregar

```
extension="mmcache.so"
```

```
mmcache.shm_size="16"  
mmcache.cache_dir="/var/ramdisk/tmpcache"  
mmcache.enable="1"  
mmcache.optimizer="1"  
mmcache.check_mtime="1"  
mmcache.debug="0"  
mmcache.filter=""  
mmcache.shm_max="0"  
mmcache.shm_ttl="0"  
mmcache.shm_prune_period="0"  
mmcache.shm_only="0"  
mmcache.compress="1"
```

Esta configuración le dice a php que cargue el modulo con los parámetros que se especifican después de la línea “extensión”. Los parámetros indican lo siguiente

Utilice 16M de memoria compartida.

Guarde las copias en /var/ramdisk/tmpcache.

Habilite el modulo.

Habilite el optimizador.

Verifique cambios en el archivo original.

Deshabilite la salida de errores al archivo log del servidor apache.

Crear cache de todos los archivos php.

No colocar limite al tamaño de los archivos en memoria compartida.

No restringir el tiempo de un archivo en memoria.

Use disco y memoria compartida para guardar cache.

Guarde cache de contenidos en formato gzip y deflate.

Después de agregar esta configuración a /etc/php.ini se debe reiniciar apache

```
service httpd restart
```

Ahora se hace una petición de cualquier pagina .php al servidor, se comprueba que se brinda sin problemas y que además se crea un archivo en el directorio elegido para guardar las copias “compiladas”.

Anexo C. Optimización de la transferencia de Datos

La compresión HTTP en apache se puede implementar directamente con el modulo mod_gzip, el cual es muy configurable, pero una vez configurado los parámetros no pueden ser cambiados en tiempo de ejecución, habría que reiniciar el servidor. Por eso se decidió utilizar a php como la herramienta para lograr la compresión y así obtener mayor dominio sobre el proceso de compresión para cada pagina generada. Al utilizar la compresión deseábamos obtener las siguientes características:

- Determinar el nivel de compresión de acuerdo a la carga de la cpu.
- Negociación de contenidos.
- Eliminar espacios en blanco y líneas innecesarias.
- Determinar los tipos de archivos a servir y decidir si comprimir o no.
- Hacer cambios en la configuración sin reiniciar el servidor.

PHP proporciona una serie de funciones para manejar datos comprimidos a través de la librería zlib, basado en estas funciones se modifico un script que implementa una clase para lograr la compresión.

Para llevar a cabo la compresión activamos la opción de guardar el resultado del php en el buffer para después poder manejarlo completo, esto se hace colocando al principio de cada script la función

```
ob_start();
```

o en el /etc/php.ini buscar la línea output_buffering y colocarla en On

```
output_buffering=On;
```

y al final del script hacer un llamado a la clase que ejecuta la compresión de código.

La clase primero verifica que el cliente acepta compresión, a través de la negociación de cabeceras, la siguiente es la cabecera de una petición HTTP enviada por IE

```
GET HTTP/1.1
Accept: application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Accept-Language: es
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
Host: www.argo.es
Connection: Keep-Alive
```

en la línea que esta en negrilla y subrayada observamos que el IE nos dice que soporta codificación gzip y deflate.

Obtenemos el valor del buffer y si el cliente acepta compresión le aplicamos la función gzcompress()

```
$contenido=ob_get_contents();  
$contenidogzip=gzcompress($contenido,$nivel);
```

de lo contrario lo dejamos tal cual. El nivel de compresión se calcula de acuerdo a la carga de la cpu en ese instante, haciendo un llamado /proc/loadavg

```
cat /proc/loadavg
```

y pasándole el resultado a php.

Se genera la cabecera HTTP para decirle al cliente que el contenido va en formato gzip y cual es su longitud

```
Content-Encoding:gzip  
Vary:Accept Encoding  
Content-Length:longitud
```

la longitud se calcula con strlen(\$contenidogzip) y finalmente se envía el resultado al navegador (echo).

Las líneas de código que implementan la compresión HTTP en siveuis, además de las que permiten enviar las imágenes según la resolución del cliente y calcular el tiempo en ejecutar un script son las siguientes:

```
<?php  
function TiempoTotal($color="#fafaef"){  
    global $TI;  
    $ti=explode(" ",$TI);  
    $tf=explode(" ",microtime());  
    $tt=($tf[1]-$ti[1])+($tf[0]-$ti[0]);  
    echo "<table width=100% heigth=2px><tr><td bgcolor=$color><font  
color=green size=1px>P&acute;gina generada en ".number_format($tt,2)."  
s</font></td></tr></table>";  
}  
if(!isset($_COOKIE[resolucion])){  
    function EscribirCookie(){  
        $js=<<<eof  
        <script lenguaje=javascript>
```

```

function SetCookieClient(){
    var ancho=screen.width;
    document.cookie="resolucion="+ancho+";path=/;domain=uis.edu.co";
    location.href="$_SERVER[REQUEST_URI]";
}
SetCookieClient();
</script>
eof;
echo $js;
exit;
}
EscribirCookie();
}
function Imagen($imagen,$ancho=true,$alto=true){
    if($imagen=="")
        return "&nbsp;";

$img="/siveuis".str_replace("graficos","graficos".$_COOKIE[resolucion],$imagen);
$prop=getimagesize($_SERVER[DOCUMENT_ROOT].$img);
$imagen="/siveuis".$imagen;
if(!$ancho)
    echo "'$imagen' height='$prop[1]'";
else if(!$alto)
    echo "'$imagen' width='$prop[0]'";
else
    echo "'$imagen' width='$prop[0]' height='$prop[1]'";
}
function ValAdmon(){
    if(strpos($_SERVER[REQUEST_URI],"siveuis/admon")!=false){
        if(!isset($_COOKIE[admin]) || trim($_COOKIE[admin])==""){
            header("Location:/siveuis/index.php");
            exit;
        }
    }
}
}
class gzip_encode
{
    var $version;
    var $level;
    var $encoding;
    var $crc;
    var $size;
    var $gzsize;
    function gzip_encode($level = 3, $debug = false)

```

```

{
    if (!function_exists('gzcompress')) {
        trigger_error('gzcompress not found, ' .
            'zlib needs to be installed for gzip_encode',
            E_USER_WARNING);
        return;
    }
    if (!function_exists('crc32')) {
        trigger_error('crc32() not found, ' .
            'PHP >= 4.0.1 needed for gzip_encode', E_USER_WARNING);
        return;
    }
    if (headers_sent()) return;
    if (connection_status() != 0) return;
    $encoding = $this->gzip_accepted();
    if (!$encoding) return;
    $this->encoding = $encoding;
    if ($level === true) {
        $level = $this->get_complevel();
    }
    $this->level = $level;
    $contents = ob_get_contents();
    $contents = str_replace("graficos",$this->LeerCookie(),$contents);
    if($GLOBALS[noblink]!="no")
    {
        $contents=str_replace("\n", ' ', $contents);
        $contents=ereg_replace('[:space:]]+', ' ', $contents);
    }
    if ($contents === false) return;
    $gzdata = "\x1f\x8b\x08\x00\x00\x00\x00"; // gzip header
    $size = strlen($contents);
    $crc = crc32($contents);
    $gzdata .= gzcompress($contents, $level);
    $gzdata = substr($gzdata, 0, strlen($gzdata) - 4); // fix crc bug
    $gzdata .= pack("V",$crc) . pack("V", $size);
    $this->size = $size;
    $this->crc = $crc;
    $this->gzsize = strlen($gzdata);
    if ($debug) {
        return;
    }
    ob_end_clean();
    Header('Content-Encoding: ' . $encoding);
    Header('Vary: Accept-Encoding');
}

```

```

Header('Content-Length: ' . strlen($gzdata));
Header('X-Content-Encoded-By:      class.gzip_encode      '.$this-
>_version);
    echo $gzdata;
}

function gzip_accepted()
{
    if (strpos($_SERVER[HTTP_ACCEPT_ENCODING], 'gzip') ===
false) return false;
    if (strpos($_SERVER[HTTP_ACCEPT_ENCODING], 'x-gzip') ===
false) {
        $encoding = 'gzip';
    }
    else {
        $encoding = 'x-gzip';
    }
    // Test file type. I wish I could get HTTP response headers.
    $magic = substr(ob_get_contents(),0,4);
    if (substr($magic,0,2) === '^_') {
        // gzip data
        $encoding = false;
    } else if (substr($magic,0,3) === 'GIF') {
        // gif images
        $encoding = "gzip";
    } else if (substr($magic,0,2) === "\xFF\xD8") {
        // jpeg images
        $encoding = false;
    } else if (substr($magic,0,4) === "\x89PNG") {
        // png images
        $encoding = false;
    } else if (substr($magic,0,3) === 'FWS') {
        // Don't gzip Shockwave Flash files. Flash on windows incorrectly
        // claims it accepts gzip'd content.
        $encoding = false;
    } else if (substr($magic,0,2) === 'PK') {
        // pk zip file
        $encoding = false;
    }
    return $encoding;
}

function get_complevel()
{
    $uname = posix_uname();

```

```

switch ($uname['sysname']) {
case 'Linux':
    $cl = (1 - $this->linux_loadavg()) * 10;
    $level = (int)max(min(9, $cl), 0);
    break;
case 'FreeBSD':
    $cl = (1 - $this->freebsd_loadavg()) * 10;
    $level = (int)max(min(9, $cl), 0);
    break;
default:
    $level = 3;
    break;
}
return $level;
}
function linux_loadavg()
{
    $buffer = "0 0 0";
    $f = fopen("/proc/loadavg","r");
    if (!feof($f)) {
        $buffer = fgets($f, 1024);
    }
    fclose($f);
    $load = explode(" ", $buffer);
    return max((float)$load[0], (float)$load[1], (float)$load[2]);
}
function freebsd_loadavg()
{
    $buffer= `uptime`;
    ereg("averag(es)e: ([0-9][.][0-9][0-9]), ([0-9][.][0-9][0-9]), ([0-9][.][0-9][0-9]*)", $buffer, $load);
    return max((float)$load[2], (float)$load[3], (float)$load[4]);
}
function LeerCookie(){
    if( isset($_COOKIE[resolucion]) && !empty($_COOKIE[resolucion]) )
        return "graficos".$_COOKIE[resolucion];
}
}
?>

```

Anexo D. Usuarios de base de datos para el Correo

Se deben conseguir postfix-pcre-2.1.0-0.20040209.18mdk.i586.rpm y postfix-pgsql-2.1.0-0.20040209.18mdk.i586.rpm e instalarlos, estos dos paquetes son los módulos de postfix para soporte de expresiones regulares y tablas de postgres.

Una vez instalados se debe crear la base de datos que albergará a los usuarios, en este caso se llama "correo", con las siguientes tablas:

alias: Tabla para albergar los datos indispensables del usuario, la dirección de correo, el directorio donde almacenara los correos y si tiene o no acceso

```
id int2 primary key
alias varchar(255) not null
maildir varchar(255) not null
acceso char(2) default 'OK'
```

alias_remos: En esta tabla se le indica a postfix si el correo debe ser entregado a un usuario local o de otro dominio

```
alias varchar(255) not null
rcpt varchar(255) not null
```

siveuis: Esta es básicamente para que el servidor pop3 pueda hacer la autenticación

```
usuario varchar(255) not null
password varchar(255) not null
maildir varchar(255) not null
activo int2 default 1
```

transporte: Se especifican los dominios por los cuales respondera postfix, estos son virtuales

```
dominio varchar(255) not null
transporte char(8)
acceso char(2) default 'OK'
```

Esta tabla se debe llenar así:

```
insert into transporte values('siveuis.uis.edu.co','virtual:','OK');
```

Despues de crear la base de datos se crean los archivos de consulta (tablas de diccionario como las llama postfix) que contactaran a postfix con postgres y le dicen que busque los

transporte.cf: dominios virtuales por los cuales responde en la tabla transporte.

```
user=postfix
password=correo
dbname=correo
table=transporte
select_field=transporte
where_field=dominio
hosts=127.0.0.1
```

uids.cf: identificadores de usuarios en la tabla alias.

```
user=postfix
password=correo
dbname=correo
table=alias
select_field=id
where_field=alias
hosts=127.0.0.1
```

alias.cf: usuarios que pertenecen al dominio

```
user=postfix
password=correo
dbname=correo
table=alias
select_field=maildir
where_field=alias
hosts=127.0.0.1
```

alias_remos.cf: si la entrega es remota o local.

```
user=postfix
password=correo
dbname=correo
table=alias_remos
select_field=rcpt
where_field=alias
hosts=127.0.0.1
```

En este punto estamos listos para decirle a postfix que utilice la infraestructura que se adecuo para enviar y recibir el correo. Esto se hace en el archivo principal de configuración /etc/postfix/main.cf. Aquí se deben agregar las siguientes líneas:

```
transport_maps = pgsq:/etc/postfix/pgsql/transporte.cf
virtual_mailbox_base = /home/postfix
virtual_uid_maps = pgsq:/etc/postfix/pgsql/uids.cf
virtual_gid_maps = pgsq:/etc/postfix/pgsql/uids.cf
virtual_mailbox_maps = pgsq:/etc/postfix/pgsql/alias.cf
virtual_maps = pgsq:/etc/postfix/pgsql/alias_remos.cf
```

Con lo cual postfix lee los datos de postgres, específicamente de la base de datos correo.

Es bueno aclarar que postfix trata todo el correo entrante como si fuera del usuario que corre el proceso, en este caso postfix, depositándolo en diferentes colas de correo dentro del directorio base(la información para encontrar este directorio se le suministró con la directiva **virtual_mailbox_base** de la sección anterior en main.cf). Es por eso que el usuario debe ser creado con un directorio hogar. Cuando se hace la instalación el usuario postfix es creado, pero sin directorio hogar y con un uid por debajo de 500, (Para que un usuario normal pueda recibir correo debe tener un uid por encima de 500) entonces se debe proceder a eliminar y crear al usuario postfix con el mismo uid que se inserto en el campo id de la tabla alias (el valor de este campo siempre es el mismo), escoger uno que este por encima de 500 y que por supuesto este libre.

```
userdel -r postfix
adduser -r postfix -u 508
```

Se debe tener cuenta que el usuario postfix debe ser el que ejecute el servicio de lo contrario los correos serán rechazados.

Después de realizar este procedimiento se puede iniciar/reiniciar el servicio postfix, para que se carguen las nuevas configuraciones. Para probar la configuración insertamos un usuario en la base de datos correo

```
insert into alias values(508,'usuario@siveuis.uis.edu.co','siveuis/usuario','OK');
insert
alias_remos('usuario@siveuis.uis.edu.co','usuario@siveuis.uis.edu.co');
insert into siveuis('usuario','password','siveuis/usuario',1);
```

Luego de crear el usuario iniciamos una sesión telnet al puerto 25 y al finalizar verificamos que el usuario haya recibido el correo observando si se creó un archivo llamado usuario dentro del directorio /home/postfix/siveuis (directorio creado con anterioridad), este archivo debe tener el contenido del mensaje enviado a través del telnet.

Para que los usuarios puedan gestionar su correo debemos instruir al servidor POP3 para que lea los datos de la base de datos. Se debe descargar aplicación teapop y compilarla con soporte para postgresql. Dentro del directorio teapop ejecutar

```
./configure --prefix=/usr --with-pgsql=/usr/lib/pgsql --sysconfdir=/etc/teapop
make
make install
```

Después de instalar se creará un archivo en /etc llamado teapop.passwd, al final de este añadir la siguiente línea (este archivo está muy bien documentado)

```
siveuis.uis.edu.co:192.168.24.109:pgsql:/home/postfix:0:postfix:postfix:127.0.0.1:5
432:correo:postfix::siveuis:usuario:password:maildir:
```

una vez hecha esta configuración iniciamos una sesión POP3 haciendo telnet al puerto 110, teniendo cuidado de escribir usuario@siveuis.uis.edu.co cuando se pida el usuario. En esta sesión comprobamos que el servidor lee los datos de postgres y que encuentra y administra la cola de correo.

Llegados a este punto solo nos queda instalar y configurar un cliente de correo, en este caso uebimiau, que es un webmail desarrollado en php-javascript y con soporte para POP3. Para instalar solo es necesario descargarlo, desempaquetarlo en algún lugar dentro del árbol web y modificar el archivo config.php que está dentro del directorio inc del raíz del paquete, según los datos de nuestro dominio, ip y puerto para POP3 (110 por defecto). Tomamos nuestro navegador y escribimos la url del lugar en donde está el correo autenticamos y uebimiau ya debe estar gestionando el correo de los usuarios.
Anexamos fragmento de config.php

```
$smtp_server = "192.168.24.109";

$mail_protocol = "pop3";
$mail_port      = 110;

$mail_detect_remove   = "www.";
$mail_detect_prefix   = "mail.";
$mail_detect_login_type = "%user%@%domain%";
```

```
$mail_servers[] = Array(  
    "domain" => "siveuis.uis.edu.co",  
    "server" => "192.168.24.109",  
    "login_type" => "%user%@%domain%",  
);  
  
$default_mail_server = "siveuis.uis.edu.co";  
$one_for_all_login_type = "%user%@%domain%";
```

Las configuraciones realizadas deben tener un correo completamente funcional. Pero para obtener un correo mucho mas robusto se podría agregar controles UCE y conectar postfix con ClamAV y spamassassin.

Anexo E. Monitoreo y Análisis del Sistema

Primero se debe instalar y configurar net-snmp-5.1-7mdk, el paquete para manejo del protocolo snmp. Luego de instalar se crea un archivo en /etc/snmp/snmpd.conf en este se debe configurar la comunidad (clave) que tendrá acceso al agente snmp. Para siveuis se definió una comunidad de solo lectura y con acceso solo desde el equipo local (localhost) llamada gvigia, con un nombre seguro vigia y accesible solo por el grupo publico, quien al final será el que acceda al agente. A continuación se muestra un extracto del archivo de configuración de siveuis

```
#####
# define RO community
rocommunity gvigia
# First, map the community name "public" into a "security name"

#   sec.name source      community
com2sec vigia localhost gvigia

#####
# Second, map the security name into a group name:

#   groupName securityModel securityName
group public v1      vigia
group public v2c     vigia

#####
# Third, create a view for us to let the group have rights to:
#   name      incl/excl subtree mask(optional)
view systemview included system

#####
# Finally, grant the group read-only access to the systemview view.

#   group      context sec.model sec.level prefix read write notif
access public "" any noauth exact systemview none none
```

Después de configurar las comunidades se inicia el servicio snmpd y se prueba de la siguiente forma

```
snmpget localhost -c gvigia sys.Descr.0
```

lo que nos debe devolver el tipo de sistema que especificamos en snmpd.conf. Después de verificar que snmp esta configurado y corriendo empezamos la configuración de mrtg.

Primero descargar el rpm de mrtg e instalarlo, una vez instalado tendremos a nuestra disposición los comandos `cfgmaker`, `mrtg` e `indexmaker` que respectivamente utilizaremos así:

con **cfgmaker** crearemos el archivo único de configuración de mrtg, `mrtg.cfg`,

```
cfgmaker gvigia@localhost > /home/httpd/html/mrtg/mrtg.cfg
```

*para acceder a los dispositivos snmp. De este archivo cambiaremos las directivas **WorkDir**: /var/www/html/mrtgu, directorio en donde se almacenan los graficos y paginas html generados por mrtg y que se publican en el web.*

Language: spanish, lenguaje con el que se mostraran las paginas.

RunAsDaemon: yes, correr como un demonio

Interval:10, cada cuantos minutos recogerá estadísticas el mrtg.

Refresh:300, cada cuantos segundos se refrescara la pagina de resultados.

El archivo de configuración inicial solo obtiene datos de snmp, luego hay que agregar algunas entradas para obtener datos de scripts. En siveuis se utilizaron los siguientes script adicionales

servidorweb: carga del servidor web apache (bytes de entrada y salida)

freemen.sh: Cantidad de memoria libre.

stats.pl: Número de procesos en ejecución, carga de la cpu, usuarios conectados, tiempo en servicio del servidor.

El resultado del archivo `mrtg.cfg` para siveuis es este

```
# Created by
# /usr/bin/cfgmaker gvigia@localhost
### Global Config Options
# for UNIX
WorkDir: /var/www/html/mrtgu

### Global Defaults

# to get bits instead of bytes and graphs growing to the right
Options[_]: growright, bits

EnableIPv6: no

#Directiva para correr como demonio
RunAsDaemon:yes
Interval:10

#Numero de segundos para actualización automatica
Refresh: 600
```

```
#Lenguaje
Language: spanish

#Tamaño de las imagenes
XSize[_]:300
YSize[_]:75

#Colores de fondo
Background[_]: #fafaef
#####
####
System: siveuis.uis.edu.co
Description: "Linux"
Contact: gdead <gdead@siveuis.uis.edu.co>
Location: Bucaramanga/Santander-Colombia
#####
####

Target[localhost_3]: 3:gvigia@localhost:
SetEnv[localhost_3]: MRTG_INT_IP="192.168.24.109"
MRTG_INT_DESCR="eth1"
MaxBytes[localhost_3]: 1250000
Title[localhost_3]: siveuis - Tráfico tarjeta de red
PageTop[localhost_3]: <H1>siveuis - Tráfico tarjeta de red</H1>
<TABLE>
  <TR><TD>Sistema:</TD> <TD>siveuis.uis.edu.co en
Bucaramanga/Santander-Colombia</TD></TR>
  <TR><TD>Administrador:</TD> <TD>gdead
&lt;gdead@siveuis.uis.edu.co&gt;</TD></TR>
  <TR><TD>Descripción:</TD><TD>eth1 </TD></TR>
  <TR><TD>Velocidad Máxima:</TD> <TD>1250.0 kBytes/s</TD></TR>
  <TR><TD>Ip:</TD> <TD>192.168.24.109 (sis25.uis.edu.co)</TD></TR>
</TABLE>

#####
####
Target[siveuis-web]: `/var/www/html/mrtgu/scripts/servidorweb`
WithPeak[siveuis-web]: my
MaxBytes[siveuis-web]: 125000
Options[siveuis-web]: growright,noinfo,unknaszero
Title[siveuis-web]: siveuis - Tráfico servidor web
PageTop[siveuis-web]: <H1>siveuis - Tráfico servidor web</H1>
```

```
#####
#####
#Información sistema
#####
#####

###Memoria utilizada
Target[siveuis-freemem]: `/var/www/html/mrtgu/scripts/freemem.sh`
Options[siveuis-freemem]: nopercen,t,growright,gauge,noinfo,unknaszero
Title[siveuis-freemem]: siveuis - Memoria libre
PageTop[siveuis-freemem]: <h1>siveuis - Memoria libre</h1>
MaxBytes[siveuis-freemem]: 1000000000
YLegend[siveuis-freemem]: bytes
ShortLegend[siveuis-freemem]: bytes
LegendI[siveuis-freemem]: &nbsp;Memoria libre:
LegendO[siveuis-freemem]:
Legend1[siveuis-freemem]: Memoria libre en bytes (no incluida swap).

###Procesos en ejecución
Target[siveuis-procs]: `/var/www/html/mrtgu/scripts/stat.pl procs`
Options[siveuis-procs]: nopercen,t,growright,gauge,noinfo,unknaszero
Title[siveuis-procs]: siveuis - Número procesos en ejecución
PageTop[siveuis-procs]: <h1>siveuis - Número procesos en ejecución</h1>
MaxBytes[siveuis-procs]: 1000000
YLegend[siveuis-procs]: procs
ShortLegend[siveuis-procs]: procs
LegendI[siveuis-procs]: &nbsp;Procs:
LegendO[siveuis-procs]:
Legend1[siveuis-procs]: Número procesos en ejecución
```

Una vez se tiene listo el archivo de configuración obtenemos las estadísticas y generamos los gráficos, páginas HTML y los logs con **mrtg**

```
mrtg /var/www/html/mrtgu/mrtg.cfg
```

para visualizar los datos necesitamos una página índice la cual crearemos con **indexmaker**

```
indexmaker /var/www/html/mrtgu/mrtg.cfg > /var/www/html/index.html
```

Después de crear la página índice apuntamos el navegador a index.html de mrtg y ya debemos poder observar algunas gráficas mostrándonos datos que se actualizarán cada 10 minutos.

A continuación se muestran los scripts usados en mrtg

stat.pl

```
#!/usr/bin/perl
#
# Based on original versión by Jon Anhold <jon@snoopy.net>
# Modifications by Jeremy Weatherford <xidus@xidus.net>
# This code is licensed under the GNU General Public License
#
# License text may be found at: http://www.gnu.org/copyleft/gpl.html
#

# stat <stat> <name>

$s = $ARGV[0];
$n = $s;
$n = $ARGV[1] if $ARGV[1];

$uptime = `uptime`;

$uptime =~ /up (.?)/;
$up = $1;

if ($s eq "load") {
    $l = `cat /proc/loadavg`;
    $l =~ /([0-9.]+)/; $l =~ /([0-9.]+)/;
    $data = $l * 100;
} elsif ($s eq "users") {
    $l = `w | wc -l`;
    $l =~ /(d+)/;
    $data = $l - 2;
} elsif ($s eq "procs") {
    $l = `ps auxc | wc -l`;
    $l =~ /(d+)/;
    $data = $l - 3;
} elsif ($s eq "uptime") {
    $data = $up;
    $i = ($data =~ /(.?) day/);
    $data = 0;
    $data = int($1) if $i;
} else {
    $data = "0";
    $n = "unknown stat requested";
}
```

```
}  
  
$data = int($data);  
  
print "$data\n";print "$data\n";  
print "$up\n";  
print "$n\n";
```

freemen.sh

```
#!/bin/sh  
# freemem  
# Memoria Libre en Linux  
# GarZa  
# garza@thelinuxmaster.org  
  
statname="freemem"  
uptime=unused  
  
stat=`free -b |grep Mem: | awk '{print $4}'`  
  
echo $stat  
echo $stat  
echo $statname  
echo $uptime
```

servidorweb

```
#!/bin/sh  
statname="Servidor Web"  
uptime=`uptime | awk '{print $3}'`" horas funcionando"  
entrada=`sudo /sbin/iptables -L -n -x -v | grep WebIn | awk '{print $2}' | grep -v  
WebIn`  
salida=`sudo /sbin/iptables -L -n -x -v | grep WebOut | awk '{print $2}' | grep -v  
WebOut`  
echo $entrada  
echo $salida  
echo $uptime  
echo $statname
```

Para usar este script se crearon las cadena WebIn y WebOut con iptables y se permitió el acceso de apache a iptables con sudo.