

**Universidad de las Ciencias Informáticas
Facultad Regional
“Mártires de Artemisa”**



Título: Monitoreo en tiempo real del estado de los recursos del sistema usando el marco de trabajo jWebSocket.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Orlando Miranda Gómez

Tutor(a):

Msc. Yamila Vigíl Regalado

Co. Tutor:

Lic. Gilberto Ramón Justiniani Fernández

Artemisa, Cuba, Junio 2012.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad Regional “Mártires de Artemisa” de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Orlando Miranda Gómez

Lic. Gilberto R. Justiniani Fernández

Firma del Autor

Firma del Co-Tutor

Msc. Yamila Vigil Regalado

Firma del Tutor



“Lo que sabemos es una gota de agua; lo que ignoramos es el océano.”

Isaac Newton

AGRADECIMIENTOS

A mi hermano Osvaldo Miranda Gómez que además de formar parte importante en la realización de esta investigación ha sido mi guía a seguir en la vida profesional y personal.

A mi novia Mayra Fva Marañón Rodríguez que me ha dado siempre su amor y cariño todo el tiempo y estar a mi lado para darme fuerza y valor para culminar todos estos años de largos estudios y esfuerzos.

A mis compañeros de grupo y de proyecto ¡WebSocket junto con sus líderes Alexander y Rebecca Schulze por darme siempre su apoyo y ayuda incondicional.

A mi profesora y amiga Yuneisy Barrios Pérez por su apoyo incondicional en cada momento que lo necesite.

A mis tutores Yamila Vigil Regalado y Gilberto Ramón Justiniani Fernández quienes influyeron mucho en la preparación y realización de este trabajo y todos los profesores que han sido clave en mi preparación y formación.

DEDICATORIA

A mi madre Vicenta Gómez Salas por darme el cariño, la dedicación y confianza en la vida, y de esta forma ayudar en formarme en la persona que hoy soy, por estar siempre a mi lado y nunca abandonarme cuando más lo necesitaba.

A mi padre Osvaldo Miranda Arencibia que aunque no lo tengo presente físicamente siempre estuvo conmigo como ejemplo y camino a seguir. Por aconsejarme, prepararme y apoyarme ante cualquier situación o antes de dar algún paso importante.

A mi familia por estar siempre presente cuando los necesité.

RESUMEN

El constante desarrollo de las computadoras ha permitido la creación y uso de programas y aplicaciones que hacen un seguimiento a su funcionamiento. Estas aplicaciones evalúan y monitorean los recursos o componentes físicos para así mostrar un resumen de sus estados de forma interactiva.

En la actualidad existen aplicaciones de escritorio y web que realizan operaciones para el control de las computadoras. Sin embargo con la evolución de la Web en cuanto al acceso de la información, es necesario el monitoreo en tiempo real de los estados de recursos de una computadora atendiendo a las nuevas exigencias de interoperabilidad y conectividad permanente en la Web. En relación a esto surge entre otros el marco de trabajo `jQueryWebSocket`, el cuál es una tecnología orientada al desarrollo de aplicaciones basadas en el protocolo `WebSocket`. Este protocolo proporciona altos niveles de seguridad, velocidad y trabajo en tiempo real, además de brindar un canal de comunicación bidireccional.

La presente investigación muestra el proceso de desarrollo en su primera versión. Esta es una solución demostrativa que tiene como objetivo monitorear el estado de los recursos de una computadora en tiempo real usando el marco de trabajo `jQueryWebSocket`. Dicha aplicación va a permitir a los usuarios aumentar el nivel de disponibilidad de los recursos de una computadora en la Web en tiempo real. El usuario además podrá hacer uso de esta herramienta a través de estaciones fijas y móviles. En esta investigación se realiza una estrategia de validación pasando por etapas como pruebas funcionales, certificación de calidad de software y valoración del cliente.

ÍNDICE GENERAL

| | |
|---|----|
| INTRODUCCIÓN..... | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE | 8 |
| 1.1. Fundamentos Teóricos | 8 |
| 1.2. Tendencias y tecnologías actuales..... | 10 |
| 1.3. Metodología de desarrollo del software para la solución..... | 13 |
| 1.4. Tecnologías usadas para el desarrollo de la solución. | 15 |
| 1.5. Lenguajes de programación y modelado para el desarrollo de la solución. | 20 |
| 1.6. Herramientas a emplear para el desarrollo de la solución. | 22 |
| 1.7. Entorno Integrado de Desarrollo. | 27 |
| 1.8. Herramienta para la construcción del proyecto. | 28 |
| 1.9. Otras librerías utilizadas en la solución..... | 28 |
| CAPÍTULO 2: CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA..... | 30 |
| 2.1. Propuesta de solución. | 30 |
| 2.2. Planificación del proyecto por roles..... | 31 |
| 2.3. Modelo de dominio. | 33 |
| 2.4. Lista de Reserva del Producto. | 34 |
| 2.5. Historias de Usuarios y Tareas de Ingenierías..... | 38 |
| 2.6. Plan de Release..... | 47 |
| 2.7. Descripción de la arquitectura. | 47 |
| 2.8. Diseño con metáforas..... | 48 |
| 2.9. Diseño de componentes..... | 50 |
| CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA..... | 52 |
| 3.1. Diagrama de despliegue..... | 52 |
| 3.2. Validación de la investigación..... | 53 |
| 3.3. Resultados obtenidos. | 63 |
| 3.4. Funcionalidades obtenidas..... | 64 |
| 3.5. Aporte social y económico..... | 64 |
| CONCLUSIONES GENERALES..... | 66 |
| RECOMENDACIONES..... | 67 |
| BIBLIOGRAFÍA REFERENCIADA..... | 68 |
| BIBLIOGRAFÍA GENERAL..... | 71 |
| ANEXOS..... | 72 |
| GLOSARIO DE TÉRMINOS..... | 73 |

INDICE DE TABLAS Y FIGURAS

Tabla # 1: Planificación del proyecto por roles31

Tabla # 2 Lista de Reserva del Producto.....34

Tabla # 3 HU Obtener uso de la memoria.....38

Tabla # 4 Tarea Obtener consumo de la memoria RAM.39

Tabla # 5 Tarea Obtener consumo de la memoria Swap.40

Tabla # 6 Tarea Mostrar uso de la memoria RAM.40

Tabla # 7 Tarea Mostrar el uso de la memoria Swap.40

Tabla # 8 Tarea Corregir el alcance del uso de la memoria.....41

Tabla # 9 HU Obtener consumo del microprocesador.41

Tabla # 10 Tarea Obtener carga del microprocesador.....42

Tabla # 11 Tarea Mostrar el uso del microprocesador.....42

Tabla # 12 HU Obtener uso del disco duro del sistema.....43

Tabla # 13 Tarea Obtener uso y espacio del disco duro.44

Tabla # 14 Tarea Mostrar el uso del disco duro.44

Tabla # 15 Tarea Notificar consumo de los recursos.45

Tabla # 16 Tarea Mejorar muestra de uso y espacio del disco duro.45

Tabla # 17 HU Notificar consumo elevado de los recursos.46

Tabla # 18 Tarea Notificar por correo.46

Tabla # 19 Plan de Releases.....47

Tabla # 20 Casos de Prueba para la HU Obtener uso de la memoria.54

Tabla # 21 Casos de pruebas para la HU Obtener consumo del microprocesador.57

Tabla # 22 Casos de pruebas para la HU Obtener uso del disco duro del sistema.....59

Tabla # 23 Casos de pruebas para la HU Notificar consumo elevado de los recursos61

Fig. 1 Modelo de dominio.34

Fig. 2 Arquitectura de la aplicación.....48

Fig. 3 Diagrama de paquetes.49

Fig. 4 Diagrama de Componentes.....50

Fig. 5 Diagrama de despliegue.....52

INTRODUCCIÓN

El surgimiento y desarrollo de las computadoras han sido caracterizados por generaciones, las cuales han ampliado el acceso y almacenamiento de la información y el uso de ellas en cualquier campo de trabajo. Estas generaciones fueron perfeccionando cada vez más los recursos físicos que la conforman en cuanto al uso de la tecnología de la información.

Los recursos fundamentales de un sistema de computadora permiten responder y ejecutar una serie de programas que hacen que sea más interactiva la comunicación con el usuario. Estos recursos son de suma importancia porque no solo determinan el correcto funcionamiento de una computadora sino también la capacidad en la velocidad de transferencia y procesamiento de datos. Necesitan en su mayoría de una configuración para que el sistema operativo los detecte de forma correcta. Dentro de estos elementos que conforman el hardware de un sistema informático se encuentran los discos duros, las *Random Access Memory* (RAM), en español Memoria de Acceso Aleatorio; la memoria de intercambio o Swap, los periféricos de entrada/salida, microprocesador o *Central Processing Unit* (CPU), en español Unidad Central de Procesamiento.

En estos días, el número de aplicaciones y servicios que soportan las computadoras son cada vez más complejas, lo que a su vez hace que evolucionen los componentes que permiten realizar estas aplicaciones. Por esto son desarrollados programas para el monitoreo de las mismas, que se han convertido en un aspecto esencial en el hardware informático. El monitoreo de recursos es una utilidad importante, la cual ayuda a los diferentes usuarios realizar un seguimiento de sus sistemas, controlar su funcionamiento y rendimiento, es además una recolección sistemática y el análisis de la información de los recursos en uso. Por tales características el monitoreo tiene como objetivo mejorar la eficiencia y la eficacia de un proyecto u organización. (Monitoring and Evaluation, 2001)

El monitoreo informático no es más que el flujo de información que se genera desde y hacia la estructura informática. Contemplando hardware, software e infraestructura con el objeto de relevar el estado en el que se encuentran los distintos parámetros operativos. Hoy en día el concepto de monitoreo de sistemas se ha vuelto muy popular, ya que ofrece muchas ventajas a los administradores de sistemas, directores ejecutivos y otros usuarios. Estas aplicaciones se utilizan para proporcionar informaciones y supervisar un grupo de actividades que realizan los componentes antes mencionados. Además permiten mantener en buena salud el estado de los componentes del sistema operativo, esto ayuda a mejorar la disponibilidad y el rendimiento de los mismos.

Con la aparición de la *World Wide Web* (WWW), instrumento de uso cotidiano en la sociedad, se ha comparado con otros medios de comunicaciones por las altas demandas de escalabilidad y disponibilidad. Por eso en la actualidad resulta ser el medio más flexible y económico para la comunicación, el comercio y los negocios, acceso a información y servicios. La ampliación de las formas de acceso y administración de sistemas remotos, facilitan la obtención del control total de un sistema conectado a la Web. Debido a esto comienzan a aparecer sistemas de monitoreo en la Web, los que permiten observar lo que ocurre en los sistemas de una empresa u hogar, desde cualquier parte del mundo. Otras de estas aplicaciones permiten observar el funcionamiento de las redes, de los sitios Web, el estado de los recursos de una empresa, entre otros.

Anteriormente, con las transformaciones que ha tenido Internet en cuanto al acceso a la información, la misma era obtenida y representada en la Web de forma estática y estadística, es decir, se iba mostrando los datos dentro de un período determinado. Sin embargo el uso intensivo cada día más de la Web trae consigo mayor interactividad en la observación de informaciones que varía constantemente y mayor necesidad de monitorear los recursos. Es evidente entonces la importancia de tomar decisiones a tiempo y garantizar la disponibilidad de los servicios de la

Web que se brindan en las computadoras o servidores.

Como resultados de los grandes desarrollos de las tecnologías que surgen en estos momentos aparecen las técnicas de desarrollo de la Web para lograr el acceso a informaciones en tiempo real. Mediante esto se propician la eficiencia y rapidez de la información. Tiempo real en la Web significa lograr dos elementos: el cliente recibe mensajes del servidor sin solicitud previa y los usuarios finales reciben actualizaciones de forma simultánea. (Framework Approach for WebSockets, 2011)

El protocolo que permite una comunicación en tiempo real en la Web es el protocolo WebSocket. Este define los procedimientos para actualizar la conexión a través de *Hypertext Transfer Protocol* (HTTP), en español Protocolo de Transferencia de Hipertexto; a una conexión mediante WebSockets totalmente bidireccional usando *Transmission Control Protocol* (TCP), Protocolo de Control de Trasmisiones. WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y *full-duplex* sobre un único *socket* TCP. No constituye solamente una nueva forma de comunicación entre el cliente y el servidor, sino un cambio de paradigma para el desarrollo de aplicaciones web estacionarias y móviles. Lograr tiempos reales, significa que el usuario reciba de manera inmediata toda la información sin previa solicitud.

Los principales servidores que soportan Websockets para el desarrollo de aplicaciones hoy día son, la Pasarela Websockets de Kaazing¹, Jetty WebSocketServlet², Socket.IO³, django-websocket del proyecto Python⁴ y jWebSocket⁵. El marco de trabajo jWebSocket que proporciona un mayor nivel de robustez y confiabilidad, brinda la posibilidad de obtener informaciones de recursos en tiempo real usando el protocolo WebSocket. El marco jWebSocket es una nueva

1 <http://kaazing.com/>

2 <http://www.eclipse.org/jetty/>

3 <http://socket.io/>

4 <http://pypi.python.org/pypi/django-websocket>

5 <https://jwebsocket.org/>

tecnología orientada al desarrollo de aplicaciones basadas en WebSockets que proporcionan altos niveles de velocidad, escalabilidad y seguridad. (Framework Approach for WebSockets, 2011)

De acuerdo con los razonamientos que se han venido realizando surge la siguiente **situación problemática:**

El monitoreo de los recursos de computadoras es de suma importancia para identificar problemas a tiempo evitando pérdidas y garantizar la disponibilidad de cada recurso. Significa entonces que la falta de evaluación rápida y estable de una computadora tiene como consecuencia que se pueda afectar la disponibilidad del estado de los recursos de una computadora y no tengan un rendimiento útil.

Si los recursos de la computadora presenta algún fallo, el monitoreo que se realiza a través del protocolo http no garantiza el tiempo real de la información. Lo que hace que no se cuente con información actualizada y provoque una carga excesiva o fallas en los recursos críticos afectando la disponibilidad del funcionamiento de los mismos.

Origina que el administrador no pueda hallar exactamente el punto donde ocurre una falla y pueda asegurar que los recursos de la computadora estén en completo funcionamiento. Provocando problemas potenciales que llevan al deterioro de los recursos y se conviertan en tiempos de inactividad en la disponibilidad de los mismos, dando lugar a un impacto negativo en los usuarios finales.

A partir de lo antes expuesto surge el **problema científico:** ¿Cómo garantizar mayor disponibilidad de los recursos de computadoras mediante el monitoreo en tiempo real en la Web? En correspondencia a esto surge como **objeto de estudio:** monitoreo del estado de los recursos de una computadora.

La investigación se enmarca en el **campo de acción:** monitoreo del estado de los recursos de una computadora mediante la Web en tiempo real con el marco de trabajo jWebSocket. Para darle solución se trazó como **objetivo general:**

Desarrollar una aplicación web con el marco de trabajo jWebSocket que garantice mayor disponibilidad de los recursos de una computadora mediante el monitoreo en tiempo real.

Para cumplir el objetivo general se definieron las siguientes **preguntas científicas**:

1. ¿Cuáles son los fundamentos teórico-metodológicos del proceso de monitoreo del estado de los recursos de una computadora en la Web con el marco de trabajo jWebSocket?
2. ¿Cuál es la situación actual del monitoreo de recursos de computadoras?
3. ¿Cómo desarrollar una aplicación web con el marco de trabajo jWebSocket que garantice mayor disponibilidad de los recursos de una computadora mediante el monitoreo en tiempo real?
4. ¿Cómo comprobar la capacidad de la aplicación web de monitoreo de los recursos de una computadora en tiempo real usando el marco de trabajo jWebSocket?

Variable Independiente: Aplicación web con el marco de trabajo jWebSocket.

Variable Dependiente: Nivel de disponibilidad de los recursos de una computadora mediante el monitoreo en tiempo real.

Para la realización correcta de este trabajo se plantean las siguientes **tareas de la investigación**:

- ✓ Fundamentación teórico-metodológico del proceso de monitoreo del estado de los recursos de una computadora en la Web con el marco de trabajo jWebSocket.
- ✓ Estudio del estado del arte de los sistemas de monitoreo de computadoras.
- ✓ Desarrollo de una aplicación web que garantice mayor disponibilidad de los recursos de una computadora mediante el monitoreo en tiempo real con el marco de trabajo jWebSocket.

- ✓ Comprobación de la capacidad de la aplicación web de monitoreo de los recursos de una computadora en tiempo real usando el marco de trabajo `WebSocket`.

Para el desarrollo de la investigación se utilizaron **métodos científicos** importantes como los teóricos y empíricos. Dentro de los métodos teóricos se han destacados los siguientes:

Histórico-Lógico: Este método permitió hacer un estudio a la evolución y desarrollo del objeto de investigación. Además de conocer algunas herramientas que permiten el monitoreo de los recursos de una computadora y así obtener una mayor comprensión del estado actual de estas aplicaciones que ayude a desarrollar una aplicación a partir de la situación problemática.

Análisis-Síntesis: Para analizar los problemas existentes por los cuales se quiere lograr un producto con las características que se requieren. También facilitó el estudio por separado de cada una de las herramientas que monitorizan los recursos de una computadora y lograr una valoración de los resultados de la investigación en curso.

Análisis Documental: Con este método se hizo un estudio de la documentación referente a las aplicaciones de monitoreo de computadoras utilizadas actualmente, para obtener a través de estas el análisis y las sugerencias que pudieran ser incorporadas a la investigación.

Al finalizar se espera (**resultados esperados**):

- ✓ Aplicación web con el marco de trabajo `WebSocket` que garantice mayor disponibilidad de los servicios de una computadora mediante el monitoreo en tiempo real de sus recursos en la Web.

La presente investigación está estructurada con una introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas, bibliografías,

un glosario de términos en donde se explicaran de forma detallada los términos técnicos utilizados en la elaboración del mismo.

Los capítulos están resumidos de la siguiente forma:

Capítulo 1. Fundamentación Teórica y Estado del Arte

Se realiza la fundamentación teórica de la investigación. Se expone un estudio del estado del arte del proceso de monitoreo de computadoras en la actualidad, tanto a nivel nacional como internacional. Se investiga sobre metodologías y herramientas para el desarrollo de software.

Capítulo 2. Características, Análisis y Diseño del Sistema

Brinda una fundamentación de la solución propuesta, a partir de la cual se describen las actividades de análisis de la solución, seguidas por la descripción de los procesos del sistema y de la etapa de diseño.

Capítulo 3. Implementación y Validación del Sistema

Se describe la etapa de implementación que conlleva a la obtención de la aplicación. Además como parte de la estrategia de validación trazada se elaboran y documentan las pruebas funcionales, certificación de calidad de software y valoración de la solución propuesta por parte del cliente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

Introducción

En este capítulo se constituye la fundamentación teórica donde se presentan los conceptos relacionados al presente trabajo. Se hace un estudio general de aplicaciones y sistemas similares existentes en el mundo que monitorizan los recursos que utiliza una computadora. Se resaltan los principales conceptos sobre las tecnologías estudiadas y utilizadas, las herramientas informáticas definidas para el desarrollo, lenguajes de programación, controladores de versiones y navegadores web. Además se hace un estudio de las metodologías de desarrollo de software candidatas a seguir para el desarrollo de la aplicación.

1.1. Fundamentos Teóricos

El monitoreo es una forma de evaluación o apreciación, que permite determinar qué programas o componentes están funcionando y cuáles no, logrando de esta forma hacer ajustes para su mejor desempeño. Además permite evaluar los recursos de una computadora. Entre las principales definiciones estudiadas alrededor del término monitoreo esta la que plantea:

“Monitoreo es llegar donde se tiene que llegar, ver lo que se tiene que ver e informar lo que se tiene que informar”. (El Monitor Informa, 2009)

La palabra monitoreo es un término que no se encuentra incluido en el diccionario de la Real Academia Española (RAE). Su origen se encuentra en monitor, que es un dispositivo que toma imágenes de instalaciones filmadoras o sensores y que permite visualizar algo en una pantalla. (RAE, 2009-2010) Puede además tratarse de un dispositivo de salida que exhibe los resultados del procesamiento realizado por una computadora.

Esto permite inferir que monitoreo es la acción y efecto de monitorear, es decir, es un término usado para nombrar a la supervisión o el control de componentes o

aplicaciones a través de un sistema y luego ser evaluado.

Realizar un monitoreo en la Web proporciona constante vigilancia sobre muchas aplicaciones, sitios web, computadoras y otros servicios de Internet, además de brindar información sobre sus desempeños. El monitoreo es un componente que en su esencia constituye una herramienta práctica para la recolección de datos en diferentes momentos dados del desarrollo de un proceso.

Los monitores de recursos trabajan coherentemente para facilitar la recogida y el tratamiento oportuno de los datos, por ejemplo, el cálculo de la temperatura del procesador y que indica al usuario la temperatura máxima aceptable. La supervisión basada en la Web se ha convertido en un nuevo paradigma de control.

La definición de monitoreo ha sido enfocada por autores como Leana Espinosa y Brücke-Le pont, quienes plantean que a través de un monitoreo se observa y analiza procesos, resultados de programas o proyectos durante su desarrollo y los entornos, en los cuales se ejecutan. El monitoreo ayuda a conducir programas o proyectos y permite garantizar la calidad, eficacia y sostenibilidad de los mismos. Se cuenta con una variedad amplia de herramientas para la observación y el análisis. Los datos obtenidos permiten conocer los avances de los proyectos y programas e indican cuando la forma de conducción requiere cambios. (ESPINOZA Corrales, y otros, 2007) (PONT, 2009)

La unión de las definiciones de monitoreo anteriores fueron adoptadas en la presente investigación. Brindando un mejor entendimiento de los conceptos relacionados con el objeto de investigación.

En la actualidad la comunicación en la Web exige un intercambio de experiencias en tiempo real, los usuarios desean recibir retroalimentación inmediata de todas las acciones que realizan. El tiempo real en la Web es un conjunto de tecnologías y prácticas que permiten a los usuarios recibir información tan pronto como se publique por sus autores en lugar de comprobar una fuente de información

periódicamente. (SURHONE, y otros, 2010)

Tiempo real en la Web es además lograr dos elementos: el cliente recibe mensajes del servidor sin solicitud previa y los usuarios finales reciben actualizaciones de forma simultánea. (Framework Approach for WebSockets, 2011) Esta definición fue asumida como principal concepto para la investigación ya que es más preciso y concreto, además fue definida por el líder del proyecto jWebSocket.

En rasgos generales se define que un sistema de monitoreo de recursos en tiempo real consiste en la observación sistemática y periódica del curso de uno o más parámetros para detectar deficiencias, obstáculos, eventuales anomalías y/o necesidades de ajuste. Además se refiere a la recolección continua, el examen y uso de la información sobre las actividades de los recursos de una computadora, para identificar los problemas, como el mal funcionamiento, y tomando las acciones correctivas para cumplir los objetivos establecidos.

1.2. Tendencias y tecnologías actuales.

En la actualidad existen diversas aplicaciones que permiten obtener información de un equipo, estos son conocidos como Sistemas Integrales de Información en Tiempo Real (SIITR). Los sistemas permiten apoyar a los operadores en la supervisión de los procesos, ya sea en condiciones normales o de emergencia, e incluso algunos sistemas ofrecen herramientas para llevar a cabo el diagnóstico y predicción de eventos. Los SIITR ofrecen funciones de presentación de información en diferentes formatos como lista de variables, diagramas de barra, gráficas de tendencia, guías de operación y alarmas, principalmente.

Estos sistemas adquieren de forma confiable la información, la procesan para validarla y la presentan de forma oportuna en el formato adecuado. Estas características los han convertido en herramientas valiosas para mejorar la eficiencia y rentabilidad de los procesos donde se instalan.

Se puede constatar entre las tendencias que presentan los sistemas de monitoreos

la visualización de la información. La cual no sólo se hace en forma local sino también a nivel remoto, vía aplicaciones web, y contarán con mecanismos de interoperabilidad con sistemas de gestión de la empresa tipo *Enterprise Resource Planning* (ERP). (Tecnologías en evolución de los Sistemas Integrales de Información en Tiempo Real, 2002)

1.2.1. Análisis de otras soluciones existentes.

Aplicaciones de escritorios

En la actualidad se cuenta con varias herramientas o aplicaciones que permiten el monitoreo de recursos de computadoras. En su mayoría son nativas de los sistemas operativos que utilizan las interfaces basadas en menús de texto. Además basan sus interfaces de usuario en el principio Windows, donde el empleo de estos elementos permite al usuario interactuar con el sistema.

Entre ellas se encuentra la aplicación Monitor del sistema de Linux permite mostrar información básica y monitorizar los procesos, el uso de los recursos del sistema. Además muestra información básica acerca los procesos activos y cómo están relacionados entre ellos. Muestra el uso actual de los recursos del sistema como son tiempo CPU, memoria y espacio de intercambio y uso de la red y lista todos los sistemas de archivos montados junto con información básica acerca de cada uno.

Otra aplicación es el Administrador de tareas de Windows que proporciona información acerca de los programas y procesos que se están ejecutando en el equipo. También muestra las medidas de rendimiento utilizadas normalmente para los procesos. Este puede evaluar la actividad de los procesos en ejecución y ver gráficos y datos acerca de la utilización de la CPU y de la memoria.

Además se encuentra como pequeña aplicación propietaria *Superbar Monitor*⁶, que brinda todos los archivos *.dll* necesarios para el monitoreo de diferentes áreas de una *Personal Computer* (PC), en español Computadora Personal. Solo se debe

⁶ <http://superbarmonitor.de/>

seleccionar los que se desean usar en cada caso y las estarán visibles con un indicador de colores mostrando el estado de determinados elementos. Los elementos disponibles son: batería, CPU, disco, memoria y volumen.

Aplicaciones web.

En la actualidad son múltiples las soluciones que de una forma u otra se relacionan con el tema del presente trabajo, a continuación se reflejan algunas aplicaciones web:

La herramienta multiplataforma *Munin*⁷ es utilizada en el monitoreo basado en la Web. Permite monitorizar muchos parámetros y visualizarlos en cómodas gráficas. *Munin* tiene una arquitectura maestro / nodo en el que el maestro se conecta a todos los nodos a intervalos regulares, y les solicita los datos. A continuación almacena los datos en los archivos, y si es necesario actualiza los gráficos. Esta herramienta monitorea el uso de recursos de cada máquina, recursos como disco, red, uso de CPU, RAM y carga.

Por otro lado se encuentra *Nagios*⁸, el cual es un software de monitorización de equipos y servicios de red. Tiene entre sus objetivos permitir a los administradores poseer el control de los recursos de la red y los problemas que se presentan en la infraestructura antes de que los usuarios identifiquen alguna dificultad. *Nagios* es un sistema de código abierto muy poderoso, implementado para monitorear cualquier objeto que esté conectado a la red, como son: servidores, *switch*, *router*, computadoras e impresoras. Proporciona los registros históricos de las interrupciones, las notificaciones, y la respuesta de alerta para su posterior análisis.

Otra herramienta es *OCSInventory*⁹ (*Open Computers and Software Inventory*, Equipos Abiertos y Software de Gestión de Inventario) que permite tener un inventario centralizado de software y hardware. Tiene una interfaz de

⁷ <http://munin-monitoring.org/>

⁸ <http://www.nagios.org/>

⁹ <http://www.ocsinventory-ng.org/fr/>

administración, que permite a los administradores consultar la base de datos del servidor a través de HTTP. Además muestra el detalle de cada servidor incluyendo: CPU, RAM, red, placa madre, video, sonido, entre otros. También muestra los datos del sistema operativo, programas instalados, versión de BIOS, detalles de red o redes activas.

Todo este grupo de aplicaciones analizadas poseen limitaciones comunes. Una de ellas es la observación activa de los recursos en la web que se realiza a través del protocolo HTTP, el cual no proporciona interoperabilidad en tiempo real en el desarrollo del monitoreo. Por tales motivos es necesario crear una aplicación que permita al usuario observar de forma detallada el uso y estado de los recursos que usa el servidor o computadora en la web usando el protocolo WebSocket, la cual será mostrada en cualquier cliente de computadora y en un celular.

1.3. Metodología de desarrollo del software para la solución.

Para la presente investigación se hizo un estudio de las principales metodologías de desarrollo para estructurar, planificar y controlar el proceso de desarrollo. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente.

Existen varios tipos de metodologías que pueden ser tradicionales y ágiles. Las metodologías tradicionales o pesadas se usan con mayor énfasis en la planificación, control del proyecto, especificación precisa de requisitos y modelado. Por su parte las ágiles tienen la capacidad de proveer respuestas rápidas y ser adaptables al cambio.

1.3.1. Proceso Unificado de Desarrollo.

Dentro de las metodologías robustas se investigó Proceso Unificado de Rational (acrónimo en inglés RUP, *Rational Unified Process*); la cual es un proceso formal para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es cumplir con los requerimientos fundamentales del usuario final para

asegurar la producción de software de alta calidad. Es guiado por casos de uso y centrado en la arquitectura, iterativo e incremental y utiliza UML como lenguaje de notación. (FIGUEROA, y otros, 2011)

Esta metodología no es la más apropiada para esta investigación, ya que está indicada para proyectos grandes donde se requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.

1.3.2. SCRUM

Una metodología que forma parte del grupo de las ágiles es SCRUM, que es ideal para la gestión de proyectos y un equipo que trabaje eficientemente y tenga siempre medidos los progresos. Para esta investigación es apropiada ya que proporciona una entrega de un producto funcional al finalizar cada Sprint, da la posibilidad de ajustar la funcionalidad en base a la necesidad de negocio del cliente y brinda una visualización del proyecto día a día. (ITZCOALT Alvarez, y otros, 2007)

Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto. (ALBALADEJO)

1.3.3. XP

XP (*eXtreme Programing*, Programación extrema) es una metodología como su nombre lo indica para una programación rápida o extrema, que tiene como parte del equipo al usuario final para llegar al éxito del proyecto basándose en la simplicidad, la comunicación y el reciclado continuo de código.

Los objetivos de XP están relacionados con este trabajo porque se le brinda satisfacción al cliente tratándole de brindar el software que él necesita y cuando lo necesita. El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están

involucrados en el desarrollo del software. (SOLÍS, 2003)

1.3.4. SXP

La metodología que más encaja para el desarrollo de esta investigación es la SXP, que está compuesta por las metodologías SCRUM y XP, que permiten actualizar los procesos de desarrollo de software para el mejoramiento de su producción, a partir de la introducción de procedimientos ágiles.

SXP está especialmente indicada para proyectos con pequeños equipos de trabajo, en constante cambio de requisitos o requisitos imprecisos, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Fomenta el trabajo en equipo, con un objetivo claro, permitiendo el seguimiento y control de las tareas a realizar. (PEÑALVER, y otros, 2010)

Por todos los grandes beneficios y ventajas que proporcionan las metodologías ágiles explicadas anteriormente, se propone para el desarrollo de esta investigación el uso de la metodología SXP, que además por todo lo visto resulta oportuna ya que tiene la combinación de todas las ventajas de SCRUM y XP para pequeños proyectos que tengan equipos que trabajen eficientemente para usuarios que requieren de una entrega rápida y eficiente de los resultados.

1.4. Tecnologías usadas para el desarrollo de la solución.

1.4.1. Marcos de trabajo.

Un Marco de Trabajo, en inglés *framework*; es un esquema, esqueleto o patrón para el desarrollo y/o la implementación de una aplicación. (¿Qué es un 'framework'?, 2006)

En otras palabras, un marco de trabajo puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar, unir los diferentes componentes de un proyecto con el intento de facilitar el desarrollo de software.

1.4.2. Marcos de Trabajo del lado del Cliente.

1.4.2.1. Marco de Trabajo Ext JS.

Es un marco de trabajo para construir Aplicaciones de Internet Enriquecidas (acrónimo en inglés RIA). Está basado en librerías Java Script de código abierto, ligera y de alto rendimiento. Incluye también tecnologías como Ajax, DHTML, XML, XSLT, JSON, CSS y DOM. Tiene incluidos la mayoría de los controles de los formularios Web incluyendo tablas para mostrar datos y elementos semejantes a la programación *desktop* como los formularios, paneles, barras de herramientas, menús y muchos otros. (Extjs, 2009)

Además proporciona los gráficos más avanzados y capacidades gráficas de cualquier marco de Java Script, sin depender de *plugins*, ofreciendo una visualización perfecta de píxeles en cualquier navegador sobre cualquier sistema operativo. (Sencha, 2011)

Ext js provee en la Web una experiencia de usuario muy parecida o igual a la que se tiene en las aplicaciones de escritorio pero para obtener esto una gran cantidad de código Java Script tiene que ser cargado, algo en torno a 300-400 Kb sin comprimir.

1.4.2.2. jQuery.

jQuery es una biblioteca de Java Script que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

jQuery es software libre y de código abierto, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en Java Script logrando grandes resultados en menos tiempo y espacio. (MURPHEY, 2010)

Consiste en un único fichero Java Script que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. La característica principal de la biblioteca es que

permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Entre sus características se encuentran:

- ✓ Selección de elementos DOM.
- ✓ Eventos.
- ✓ Manipulación de la hoja de estilos CSS.
- ✓ Efectos y animaciones.
- ✓ Animaciones personalizadas. (Proyecto jQuery, 2010)

Así pues, este marco de trabajo, ofrece una infraestructura con la que se tendrá mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente como también es ideal si se quiere agregar algunos ajustes AJAX / JS a una interfaz de usuario existente.

1.4.2.3. YUI (Yahoo! User Interface).

YUI es un marco de trabajo utilizado para desarrollar en la web con librerías JavaScript y CSS para crear interfaces de usuario. Contiene un conjunto de utilidades y controles que se utilizan para crear aplicaciones web ricas, interactivas y dinámicas complejas, usando técnicas como DOM scripting, DHTML y AJAX. YUI es escalable, rápido y robusto. Construido por los ingenieros de interfaz de Yahoo! y colaboradores de todo el mundo, es una biblioteca JavaScript de potencia industrial para los profesionales que aman JavaScript. (Inc., 2011)

Cuenta con el respaldo de Yahoo!, que utiliza su librería en muchas de sus páginas vistas a diario. YUI se ejecuta correctamente en todos los navegadores modernos e incluso en algún navegador obsoleto. (EGUÍLUZ Pérez, 2009)

Sin embargo YUI en comparación con los otros marcos tiene como desventajas que requiere de más código para implementar una funcionalidad por lo tanto toma más tiempo para elaborar una solución. Más bien YUI se puede utilizar cuando se necesita trabajar en una aplicación realmente grande y con un conjunto de

*widgets*¹⁰, ya que tiene muchos ejemplos de este último.

1.4.2.4. Bindows.

Bindows es un marco de trabajo Ajax para el desarrollo de soluciones ricas Ajax y aplicaciones web, con una apariencia idéntica a Windows. El nombre Bindows proviene de la combinación de Inteligencia de Negocio (BI por sus siglas en español) y Windows.

Con Bindows las aplicaciones web de negocios tendrán exactamente la misma apariencia de un Windows GUI, es decir, que no hay necesidad de volver a entrenar a un usuario para una interface nueva, ya que los usuarios ya conocen el tipo de interfaz de Windows.

Las aplicaciones Bindows se descargan significativamente más rápido que las aplicaciones de escritorio o *desktop*. La avanzada y dinámica arquitectura de memoria de Bindows permite el uso de cualquier tamaño de información desde muy pequeña al infinito, mientras que las aplicaciones *desktop* se limitan por el cliente RAM. Este marco de trabajo se caracteriza de cero mantenimientos, no hace falta instalación, ningún pre requisito de configuración, y todos los usuarios estarán usando la última aplicación por defecto. Cabe decir que Bindows es de actualizaciones fáciles, por lo que sólo se actualizan los servidores, las nuevas versiones se proveen directamente a los usuarios automáticamente. (CELIS, 2003-2011)

Entre otras de sus características se encuentra:

- ✓ La mayoría de los procesos se realiza en las máquinas de los clientes.
- ✓ La información se actualiza usando puramente XML lo que hace que sea extremadamente limpio y responde siempre aún en casos de un gran número de usuarios.

¹⁰ Es una pequeña aplicación o programa para proveer información visual.

- ✓ Trabaja con cualquier servidor (Java, .NET, PHP).
- ✓ Cualquier sistema operativo puede ser usado en el cliente (siempre y cuando incluya un navegador).
- ✓ Todos los lenguajes soportados, Bindows soporta Unicode.

Como resultado de todo lo antes planteado se escoge para la realización de esta aplicación los marcos de trabajos jQuery en su versión 1.5.1 que permite establecer una conexión eficaz con el servidor y Bindows que además de todo lo visto anteriormente tiene una librería llamada Bindows Gauges versión 1.1 beta 3 que ofrece numerosos tipos de medidores que permitirá mostrar la información que el servidor envíe. La librería Bindows Gauges se escoge ante los demás ya que además de ser una librería de Java Script tiene varias ventajas, como mejor y más agradable visualización de los datos a mostrar, es mucho más fácil la integración y utilización con jWebSocket.

1.4.3. Marco de Trabajo del lado del Servidor.

1.4.3.1. Marco de Trabajo jWebSocket.

El marco de trabajo jWebSocket es de código abierto para el desarrollo de aplicaciones web estacionarias y móviles basado en Java en el lado del servidor y permitiendo desarrollar clientes con lenguajes como Java Script, Python y C#. Este marco establece un modelo de *token* que son datos abstractos que a través de una estructura jerárquica y una API proporcionan métodos de acceso a los contenidos. El cliente nativo soporta el intercambio de paquetes en los formatos JSON, XML y CSV, que en entornos específicos se pueden utilizar sin la necesidad de manejarlos a través de *tokens*. El cliente jWebsocket tiene una arquitectura de *plug-in* que permite aumentar con facilidad sus funcionalidades.

El servidor jWebSocket está diseñado para funcionar como servidor de comunicaciones o como servidor web, brindando total flexibilidad. En la primera opción jWebSocket proporciona un archivo *.jar*, ofreciendo la ventaja de ejecutarse

fácilmente desde una línea de comandos e integrarse a la biblioteca de una aplicación existente de Java. En la actualidad hay algunos servidores que ya soportan WebSockets y otros que no, por lo que jWebSocket se integra a servidores como Tomcat o Apache para lograr una comunicación WebSockets. En caso de que los servidores soporten de manera nativa WebSockets, como el caso de Jetty o GlassFish, se incluyen las funciones de comunicación del marco de trabajo jWebSocket, pero los motores internos se apagan y el anfitrión se utiliza. Esto asegura que no haya mecanismos de seguridad adicionales.

Estas características muestran la fortaleza y flexibilidad del marco de trabajo para el desarrollo de aplicaciones web estacionarias y móviles, multiplataforma, multisectorial y compatible con todos los navegadores. (Framework Approach for WebSockets, 2011)

El proyecto jWebSocket conformado además por un equipo de estudiantes de la Facultad Regional de la UCI “Mártires de Artemisa”, constituye alrededor de un 50% de los integrantes del grupo de desarrollo de este marco de trabajo. La presente investigación es parte del proyecto jWebSocket, precisamente por esta razón la aplicación a desarrollar se lleva a cabo sobre este marco de trabajo con la versión 1.0.

1.5. Lenguajes de programación y modelado para el desarrollo de la solución.

1.5.1. Java

Esta solución será desarrollada con Java 7 ya que se escogió como marco de trabajo jWebSocket y su base es con este lenguaje.

Java es un lenguaje moderno, de alto nivel, que recoge los elementos de programación que típicamente se encuentran en todos los lenguajes, permitiendo la realización de programas profesionales. La tecnología Java está compuesta básicamente por dos elementos: el lenguaje y su plataforma. Actualmente este lenguaje es el más utilizado para el desarrollo de aplicaciones por las ventajas que

brindan sus características. (GOSLING, y otros, 2005)

Además es un lenguaje multiplataforma, lo que permite que el código funcione en cualquier sistema operativo que tenga instalada la máquina virtual java. Java tiene un modelado de objetos simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

1.5.2. Java Script

Java Script es utilizado para desarrollar el cliente de esta solución ya que es un lenguaje para realizar acciones dentro del ámbito de una página web y es compatible con la mayoría de los navegadores existentes. Aunque no es un lenguaje orientado a objetos se pueden implementar muchas de las características de este paradigma y aplicar diversos patrones de código y diseño permitiendo crear aplicaciones web con un alto grado de calidad. (GOODMAN, 2001)

Los programas escritos con Java Script se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (EGUÍLUZ Pérez, 2009) Java Script es el lenguaje de programación utilizado en el lado del cliente del marco de trabajo *jQuery* seleccionado para la presente investigación.

1.5.3. HTML

El Lenguaje para Marcado de Hipertexto HTML es un lenguaje estático para el desarrollo de sitios web permitiendo estructurar las páginas web, fue desarrollado por el *World Wide Web Consortium (W3C)* y se ha convertido en un estándar reconocido mundialmente constituyendo el idioma común para todos los navegadores. Los archivos pueden tener las extensiones *HTM* y *HTML*. (LUJAN, 2002)

Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. (EGUÍLUZ Pérez, 2009) *HTML5* es el lenguaje utilizado para crear las

interfaces o vistas de la solución propuesta.

1.5.4. Hojas de Estilo en Cascada

Estilos de Hojas en Cascada (CSS por sus siglas en inglés) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Para definir la posición de cada elemento dentro de la página, entre otros. (EGUÍLUZ Pérez, 2009) CSS3 es el lenguaje utilizado para controlar el aspecto de las interfaces o vistas de la solución propuesta.

1.5.5. Lenguaje Modelado Unificado

El Lenguaje Modelado Unificado (UML), en inglés *Unified Modeling Language*; es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (RUMBAUGH, y otros, 2007) La presente investigación utiliza la herramienta de modelado Visual Paradigm versión 6.4 para trabajar con UML.

1.6. Herramientas a emplear para el desarrollo de la solución.

1.6.1. Herramienta CASE.

Las herramientas Ingeniería de Software Asistida por Ordenador (CASE por sus siglas en inglés) propician un conjunto de métodos y técnicas automatizadas que brindan ayuda y dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todo el ciclo de vida del desarrollo de un software, reduciendo el esfuerzo, el costo y el tiempo. (Herramientas Case, 2011)

Dichas herramientas se encuentran en una continua evolución, por lo que existe una gran variedad de proveedores y productos. A continuación se hace una caracterización de algunas de las más usadas en la actualidad que permitirá adquirir los elementos necesarios para determinar cuál es la más idónea para especificar y diseñar la solución propuesta.

1.6.1.1. Rational Rose.

Rational Rose es una herramienta CASE que da soporte al modelado visual con UML cubriendo todo el ciclo de vida de un proyecto. Es compatible con la metodología RUP que permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo del software. Se enmarca dentro del desarrollo de modelado para fines académicos, investigativos y comerciales. Además brinda estas facilidades: capacidad de análisis de calidad de código, provee visualización, modelado y las herramientas para desarrollar aplicaciones web, modelado UML para trabajar en diseños de base de datos y capacidad para integrarse con sistemas de control de versiones. (GONZÁLEZ Blanco, y otros)

Esta herramienta comparada con Visual Paradigm tiene menos estabilidad de ejecución en diferentes sistemas operativos y menos trabajo con un modelo UML utilizando el mismo programa sin importar el sistema operativo y sin afectar en absoluto el trabajo echo

1.6.1.2. Visual Paradigm.

Visual Paradigm es una poderosa herramienta CASE que hace uso UML con soporte multiplataforma, que proporciona un ciclo de vida completo del desarrollo de software, excelentes facilidades de interoperabilidad con otras aplicaciones, compatibilidad entre versiones, así como dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Esta herramienta brinda una serie de facilidades como un generador de mapeo de objetos-relacionales para los lenguajes de programación Java, .NET y PHP,

integración con herramientas Java (Eclipse/IBM WebSphere, JBuilder, NetBeans IDE, Oracle JDeveloper y BEA Weblogic). Brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas. Integración con herramientas para el control de versiones. (Visual Paradigm International Ltd., 2011)

Anteriormente se analizaron algunas de las herramientas CASE existentes en el mundo para el modelado de software, y se selecciona como herramienta CASE a Visual Paradigm con la versión 6.4. Esta herramienta a pesar de no ser de código abierto, cuenta con una licencia comercial la cual posee la universidad y en la cual se ha usado más en la docencia y producción, ya que es mejor en computadoras menos potentes.

1.6.2. Herramientas de Control de Versiones.

Un sistema de control de versiones es un sistema de gestión de archivos y directorios, cuya principal característica es mantener el historial de cambios y modificaciones que se han realizado sobre dichos archivos a lo largo del tiempo. Los sistemas de control de versiones permiten el trabajo colaborativo entre varios puestos de trabajo a través de la red y a su vez proporcionando una mayor seguridad y disponibilidad de los datos. A continuación se hace una caracterización de algunos sistemas de control de versiones con el objetivo de determinar el más idóneo para garantizar mayor seguridad y disponibilidad de los datos.

1.6.2.1. Git

Git es un sistema de control de versiones, diseñado para el trabajo con proyectos de cualquier tamaño con gran rapidez y eficiencia. Tiene una forma diferente y revolucionaria en su sistema de guardado haciendo que cada copia de trabajo sea un repositorio en sí mismo y contenga todo el historial de modificaciones. Esta es una característica que lo hace muy eficiente porque se puede disponer de toda la información necesaria para trabajar sin conexión y sincronizar los cambios una vez restablecida la conexión.

Entre sus características se encuentran: eficiencia, desarrollo no lineal que evita la separación del código en ramas de trabajo que pueden o no incorporarse al código base, limpieza del espacio de trabajo y autenticación criptográfica del repositorio. (CHACON) Este sistema no se escoge para el desarrollo de la solución ya que no se cuenta con experiencias en el uso de la misma.

1.6.2.2. Subversion

Subversion es un sistema de control de versiones completamente equipado. El modelo básico, el diseño y la interfaz de esta herramienta fueron fuertemente influenciados por CVS (Sistema de Control de Versiones) por lo que debido a estas particularidades los usuarios de CVS se sienten muy cómodos al interactuar con Subversion. (Apache Software Foundation, 2011)

Subversion permite versionar los directorios, tiene resolución de conflictos de forma interactiva, bloqueo de archivos, vinculaciones para lenguajes de programación y vinculación de varios repositorios. Esta última permite a los desarrolladores tener acceso a todas las funcionalidades de Subversion desde diferentes Entornos Integrados de Desarrollo (IDE por sus siglas en inglés) como Eclipse y NetBeans, entre otros.

Después de haber observado las características expuestas de los sistemas mencionados, se selecciona Subversion versión 1.6.12 como herramienta de control de versiones para ser utilizado en el desarrollo de la aplicación, ya que permite la integración con NetBeans y la posibilidad de desarrollar en paralelo. También influyó en la decisión, la experiencia de trabajo por parte del equipo de desarrollo con esta herramienta y su fácil uso.

1.6.3. Herramientas del Cliente de Control de Versiones.

Los clientes permiten la gestión de cambios que se realizan sobre los elementos de algún producto. Permitiendo una conexión entre él como cliente y el servidor, para un mejor manejo de los archivos locales.

1.6.3.1. TortoiseSVN

Es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. TortoiseSVN maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. (team, 2011)

Los menús contextuales de TortoiseSVN funcionan en otros administradores de archivos que son comunes a la mayoría de aplicaciones estándar de Windows. Sin embargo, se tuvo en cuenta que TortoiseSVN está desarrollado para sistemas privativos y por eso no se escoge como sistema de versiones para el desarrollo del trabajo.

1.6.3.2. RapidSVN

Es una plataforma de interfaz gráfica de usuario, para el sistema de revisión de Subversion. RapidSVN está licenciado bajo la v3 de GNU *General Public License*. Utiliza las mejores características de los clientes de otras arquitecturas de control de versiones. Si bien es bastante fácil para los nuevos usuarios de Subversion trabajar con él, también debe ser lo suficientemente potente como para que los usuarios con experiencia sean aún más productivos. (RapidSVN, 2011)

Características de RapidSVN:

- ✓ **Simple:** Proporciona una interfaz fácil de usar para las características de Subversion.
- ✓ **Eficiente:** Simple para los principiantes pero lo suficientemente flexible como para aumentar la productividad para los usuarios de Subversion.
- ✓ **Portátil:** Se ejecuta en cualquier plataforma: Linux, Windows, Mac OS / X, Solaris.

Hecha la observación anterior de las herramientas para el control de versiones, y analizando las ventajas y funcionalidades que ofrecen, se decide usar Rapsidsvn versión 0.12.0 ya que es una herramienta libre, y fácil de usar, tanto por quienes ya conocen. Subversion como para quienes empiezan, pudiendo acceder a direcciones SVN, subir y descargar contenido, sincronizarlo con el servidor original, comprobar su estado, crear y fusionar direcciones.

1.7. Entorno Integrado de Desarrollo.

Un IDE es un entorno de programación que integra varias herramientas con el objetivo de facilitar el desarrollo de software sobre uno o varios lenguajes de programación. La mayoría de los IDE cuentan con herramientas tales como: editor de código, herramientas para el rastreo de código, compilador, depurador y constructor de interfaz gráfica. (NOURIE, 2005)

1.7.1. Eclipse

Eclipse es un entorno integrado de desarrollo de código abierto y multiplataforma. En un principio Eclipse fue desarrollado por IBM y posteriormente su desarrollo fue llevado a cabo por Eclipse Foundation. Esta plataforma, típicamente ha sido usada para desarrollar entornos integrados de desarrollo, como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte del IDE. Eclipse basa su funcionalidad en módulos (*plug-in*) que se adaptan a las necesidades del programador. Este mecanismo de módulos es una plataforma ligera para componentes de software que permite el uso de diferentes lenguajes de programación como son Java, C/C++ y Python. (Eclipse, 2011)

Eclipse no tiene un soporte para *Java Development Kit* (JDK) como NetBeans. (Oracle Corporation, 2012) Cuenta con autocompletado de código, además de generar los *get* y los *set*, pero a su vez no cuenta con herramientas que te ayudan a hacer la interfaz gráfica: botones, *label* y *textfield*. Aunque en el desarrollo de esta solución no se utilice estas últimas características hacen que este IDE se vea menos completo que otros.

1.7.2. NetBeans

NetBeans IDE es una herramienta desarrollada por Sun Microsystems. Está completamente escrito en Java, por lo que puede ser utilizado desde cualquier sistema operativo compatible con la máquina virtual de Java. Permite el desarrollo de aplicaciones de escritorio, web y móviles. Brinda soporte a varios lenguajes de programación como Java, PHP, C/C++, Groovy, Python, Java Script, entre otros. Es un producto libre y gratuito sin restricciones de uso. Esta es una herramienta para escribir, compilar, depurar y ejecutar programas. Cuenta con un depurador, perfilador de integración, herramientas para refactorizaciones, completamiento de código y control de versiones de archivos. (NetBeans, 2011)

Para el desarrollo de la solución propuesta se selecciona como entorno integrado de desarrollo a NetBeans IDE debido a que se tiene una mayor experiencia y familiarización con esta herramienta. Además su versión 7.0.1 introduce un soporte para el desarrollo con la especificación JavaSE7 (*Java Standard Edition*) con las características de JDK7. Esta versión también ofrece una integración mejorada con el servidor Oracle WebLogic, así como soporte para Oracle Database y GlassFish3.1. Otros puntos destacados incluyen soporte para Maven3 y HTML5, así como mejoras en el editor de Java (NetBeans, 2011)

1.8. Herramienta para la construcción del proyecto.

En esta investigación se usa Maven3 ya que es una herramienta de código abierto para administrar proyectos de software. Brinda una estructura consistente de proyectos y herramientas necesarias para la gestión avanzada de dependencias, compilar y ejecutar pruebas. (CHAVARRÍA, 2010) Además Maven es capaz de descargar nuevas actualizaciones de las bibliotecas de las que depende un proyecto dado.

1.9. Otras librerías utilizadas en la solución.

El Sistema recolector de Información y Reportero (Sigar), en inglés *System Information Gatherer And Reporter*, es una librería bajo licencia GNU GPL que

permite obtener datos de nuestro computador. Además es una API multi-plataforma que a través de lenguajes de la biblioteca y de línea de comandos puede acceder a la herramienta del sistema operativo y la información a nivel de hardware en Java, Perl y NET.

Sigar en su versión 1.6.4 incluye soporte para Linux, FreeBSD, Windows, Solaris, AIX, HP-UX y Mac OS X a través de una variedad de versiones y arquitecturas. Brinda acceso portátil a datos de inventario y monitoreo, incluyendo: información de la CPU, espacio en disco disponible, cantidad de espacio que consume un directorio en específico, datos de la RAM, información de red y de los procesos que se están ejecutando en el sistema, nombre y versión del sistema operativo. Esta información está disponible en la mayoría de sistemas operativos, pero cada sistema operativo tiene su propia forma de prestarla. (MORGAN, 2010)

El API de Sigar proporciona una serie de métodos que permiten obtener información del sistema operativo; consta de una librería de Java (*.jar*) y otras librerías nativas específicas para cada sistema operativo soportado (*libsigar-amd64-linux.so*, para Linux de 64 bits o *sigar-x86-winnt.dll* para Windows de 32 bits, por poner un ejemplo). (Cristian, 2006-2010)

Conclusiones del capítulo.

A partir del análisis de las tendencias actuales, las tecnologías y herramientas utilizadas en la web para monitorizar recursos de las computadoras, en este capítulo se hizo una breve descripción y estudio de ellas para buscar sistemas similares. Se seleccionaron las más apropiadas en función de sus características que puedan ayudar a la solución del problema.

Se hizo un estudio a los principales conceptos relacionados con el objeto de investigación que fueron dados por autores como Leana Espinosa y el diccionario de la RAE.

CAPÍTULO 2: CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción.

En este capítulo se realiza la implementación de la solución de problema y se generan los artefactos correspondientes a la fase de desarrollo. Para definir las características del sistema se realiza el modelado del dominio con el objetivo de comprender su contexto; se hace la propuesta del sistema, describiendo cómo debe funcionar y destacando sus características distintivas; se especifican además sus requisitos funcionales y no funcionales. Se explica la dinámica del proyecto a través de historias de usuarios, prototipos de interfaz de usuario y otros modelos auxiliares.

2.1. Propuesta de solución.

La aplicación web que se propone en la presente investigación se titulará Monitoring Demo que debe proporcionarle al usuario el estado de los recursos de su computadora, mostrando de forma dinámica una descripción del funcionamiento de cada uno de ellos. Esta aplicación es desarrollada con el marco de trabajo `jQueryWebSocket` que permite el desarrollo de aplicaciones web en tiempo, aportando altos niveles de velocidad, escalabilidad y tiempo real. Significa que se podrá visualizar el estado de los recursos de su computadora desde otra o desde un celular, tan solo teniendo un navegador web que soporte el protocolo `WebSocket` para garantizar la comunicación bidireccional.

La aplicación demostrativa proporciona un alto nivel de seguridad ya que permite notificar al usuario de la computadora, a través de correo, en caso de que exista algún consumo elevado de algunos de los recursos. Garantizando así que no exista ninguna falla o deterioro de los mismos y de esta forma aumentar sus niveles de disponibilidad en el funcionamiento mediante la Web.

Para esto se realizará un diseño que sea bueno, fácil de entender y específico que permita adquirir una comprensión de los aspectos relacionados con los requisitos

Capítulo II: Características, análisis y diseño del sistema

funcionales y no funcionales del sistema, que sirven de base para el comienzo del desarrollo del mismo. Además la aplicación debe estar diseñada para ser asequible para cualquier usuario.

Independientemente de que el objetivo de esta aplicación es que permita visualizar el estado de los recursos y la notificación en caso de consumo elevado de los mismos, se pretende desarrollar las funcionalidades necesarias para que el trabajo con esta herramienta sea cómodo y confortable para el usuario en su navegación.

Para acceder a la aplicación Monitoring Demo será necesario que el usuario pueda utilizar un navegador que soporte el protocolo WebSocket. De esta forma puede monitorizar su computadora o laptop desde cualquier otra y desde un celular, en donde la información será enviada por el servidor jWebSocket, el cual estará instalado en su computadora personal. Será necesario para el uso de esta aplicación tener una conexión a Internet y utilizar cualquiera de los navegadores que soportan el protocolo de comunicación WebSocket. Algunos de ellos son Google Chrome 4, Safari 5, Mozilla Firefox 8, Opera 11 e Internet Explorer 10.

2.2. Planificación del proyecto por roles.

Como se muestra en la Tabla # 1 SXP define diferentes roles que intervienen en el ciclo de desarrollo de la solución propuesta. Estos se enfocan en la organización y comunicación entre los miembros del proyecto para lograr un exitoso resultado en el desarrollo de software.

Tabla # 1: Planificación del proyecto por roles.

| Rol | Responsabilidad | Nombre |
|---------------------------|--|--------------------------|
| Líder del Proyecto | Debe asegurar que el proyecto se está llevando a cabo de acuerdo con las prácticas y que todo funciona según lo planeado. Coordina y facilita las reuniones. Asegura que se consiguen los objetivos de cada iteración. | Yamila Vigil Regalado |

Capítulo II: Características, análisis y diseño del sistema

| | | |
|---------------------|--|---|
| Gerente | Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto. Participa en la selección de objetivos y requerimientos. Tiene la responsabilidad de controlar el progreso y trabaja junto con el Jefe de Proyecto en la reducción de la Lista de Reserva del Producto. | Alexander Schulze |
| Especialista | Es necesario que conozca a fondo el proceso para el desarrollo de software. Es una especialización que está activa, el miembro del grupo de trabajo que la desempeña siempre está ejecutándola y alcanzando un grado mayor de conocimientos en el tema, en este caso como Diseñador Gráfico. | Rebecca Schulze Eduardo Bourzac Hernández |
| Consultor | Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas, además aportan ideas y experiencias para el beneficio del sistema en desarrollo. | Alexander Schulze Rebecca Schulze |
| Cliente | Participa en las tareas que involucran la lista de reserva del producto. | Alexander Schulze |
| Programador | Elabora el código de las nuevas funcionalidades a implementar. Escribe las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y el resto del equipo. | Orlando Miranda Gómez |
| Analista | Escribe las historias de usuario y las pruebas funcionales para validar su implementación. | Orlando Miranda Gómez |

Capítulo II: Características, análisis y diseño del sistema

| | | |
|-----------------------------|--|--------------------------|
| Diseñador | Encargado del diseño del sistema y de los prototipos de interfaces, son los máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción. | Orlando Miranda Gómez |
| Encargado de Pruebas | Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas. | Orlando Miranda Gómez |
| Arquitecto | Se vincula con el analista y el diseñador ya que su trabajo tiene que ver con la estructura y el diseño del sistema. Ayuda en el diseño de las metáforas. | Orlando Miranda Gómez |

2.3. Modelo de dominio.

Dentro de las actividades más importantes definidas en la metodología SXP se encuentra la planilla del modelo historias de usuario del negocio donde se definen las características particulares del negocio y la interacción del sistema con los clientes. En caso que en el negocio no quede claramente definido entre los clientes y los ejecutores del proyecto, se genera el modelo de dominio.

El Modelo de Dominio es la representación visual de los elementos significativos y sus interacciones para resolver un problema específico. En la presente investigación es utilizado como base para permitir un mejor entendimiento de la aplicación a desarrollar.

A continuación en la Fig. 1 se muestra el modelo de dominio del sistema propuesto.

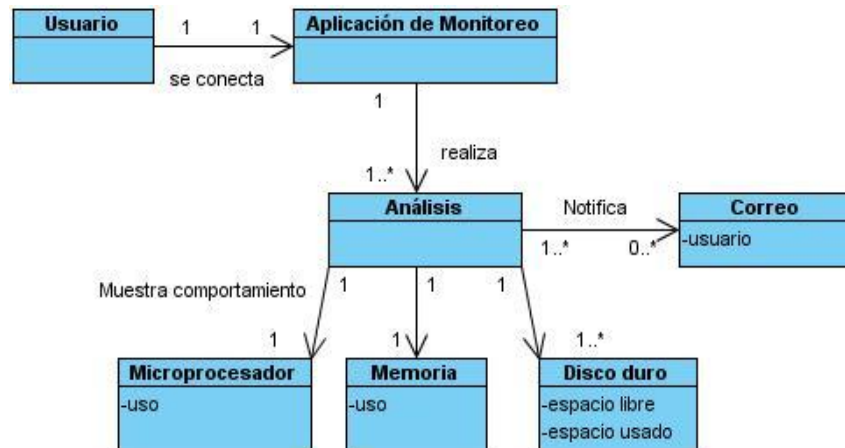


Fig. 1 Modelo de dominio.

2.4. Lista de Reserva del Producto.

La Lista de Reserva del producto (LRP) se utiliza para recoger los requisitos del producto; es un artefacto que recoge en lenguaje natural las funcionalidades que debe cumplir el sistema para su desarrollo y está conformado por una lista priorizada que define el trabajo que se va a realizar en el proyecto. Tiene como objetivo principal asegurar que el producto resultante sea más correcto y útil después de haber terminado la lista.

Esta lista como se observa en la Tabla # 2 puede estar conformada por requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas.

Tabla # 2 Lista de Reserva del Producto.

| Prioridad | Ítem * | Descripción | Estimación | Estimado por |
|-----------------|--------|--|------------|--------------|
| Muy Alta | | | | |
| | 1. | Obtener el uso en por ciento de la memoria RAM. | 1 semana | Analista |
| | 2. | Obtener el uso en por ciento de la memoria Swap (Intercambio). | 1 semana | Analista |
| | 3. | Enviar correo para notificar alto consumo | 2 semanas | Analista |

Capítulo II: Características, análisis y diseño del sistema

| | | | | |
|---|-----|---|-----------|----------|
| | | de los recursos. | | |
| Alta | | | | |
| | 4. | Obtener el consumo en por ciento del microprocesador. | 3 semanas | Analista |
| Media | | | | |
| | 5. | Obtener el uso del disco duro del sistema (Total, usado y libre). | 1 semana | Analista |
| Baja | | | | |
| | 6. | Mostrar el uso de la memoria RAM. | 1 semana | Analista |
| | 7. | Mostrar el uso de la memoria Swap | 2 días | Analista |
| | 8. | Mostrar el uso del CPU. | 3 días | Analista |
| | 9. | Mostrar el uso del disco duro (Espacio total, usado y libre). | 1 día | Analista |
| RNF (Requisitos no funcionales). | | | | |
| | 10. | La aplicación está dirigida a usuarios con conocimientos básicos de informática y un correcto dominio del negocio en cuestión, por lo cual debe estar diseñada para ser de fácil manejo por el cliente. | | |
| | 11. | La información estará disponible las 24 horas del día. | | |
| | 12. | Garantizar la integridad y disponibilidad de los datos. | | |
| | 13. | Lograr un diseño amigable. | | |
| | 14. | Se debe realizar la aplicación de forma versionable que | | |

Capítulo II: Características, análisis y diseño del sistema

| | | | | |
|--|-----|---|--|--|
| | | permita darle mantenimientos al sistema a fin de aumentar las funcionalidades y/o corregir los errores del mismo a través de versiones posteriores. | | |
| | 15. | Se documentará la aplicación con diferentes manuales con el objetivo de explicar el uso de la aplicación para garantizar el soporte de la misma. | | |
| | 16. | En el lado del servidor el lenguaje de programación a utilizar es Java, empleando el Framework jWebSocket, la librería Hyperic Sigar y el IDE NetBeans 7.0.1. | | |
| | 17. | En el lado del cliente se hará uso de los lenguajes Java Script, HTML, XML y CSS, empleando el Framework jQuery, la librería Bindows Gauges usando el IDE NetBeans 7.0.1. | | |
| | 18. | La metodología de desarrollo a seguir es SXP y para la modelación se utilizará la herramienta Visual Paradigm 3.4. | | |

Capítulo II: Características, análisis y diseño del sistema

| | | | | |
|--|-----|--|--|--|
| | 19. | Todos los textos y mensajes en pantalla aparecerán en idioma inglés. | | |
| | 20. | La interfaz deberá ser consistente con el mundo real, de manera que los conceptos manejados sean conocidos por los usuarios, para que les sea fácil su uso y aprendizaje. | | |
| | 21. | La resolución para la cual se debe diseñar la aplicación es 754 píxeles de ancho con una altura variable. | | |
| | 22. | Los requisitos mínimos de hardware para el correcto funcionamiento de la aplicación son: 512 MB de memoria RAM, microprocesador Pentium IV, tarjeta red 100 Mbps | | |
| | 23. | El cliente debe contar con un navegador que soporte el protocolo WebSocket. | | |
| | 24. | El servidor de aplicaciones debe tener instalado la máquina virtual de Java OpenJDK 6, Maven 2 y el servidor de jWebSocket. Además de copiar el archivo de la librería Sigar del sistema operativo en el | | |

| | | | | |
|--|-----|---|--|--|
| | | directorio del Open JDK. | | |
| | 25. | El código de la aplicación será liberado bajo la Licencia Pública General Reducida de GNU (LGPL) | | |
| | 26. | El código y la documentación de la aplicación deberán cumplir estrictamente las normas de calidad estipuladas por el grupo de Calisoft de la UCI. | | |

2.5. Historias de Usuarios y Tareas de Ingenierías.

Las historias de usuarios en la metodología SXP se utilizan para especificar los casos de usos. Describen las tareas que el sistema debe hacer, cuestión que depende en gran medida de las especificaciones realizadas por el cliente. Se escriben con un lenguaje natural y con palabras concisas para no exceder su tamaño en unas pocas líneas de texto.

A continuación se dan a conocer las distintas historias de usuarios que están presentes en el sistema; así como su relación con los prototipos de interfaz usuario y las tareas asociadas a cada historia de usuario.

Tabla # 3 HU Obtener uso de la memoria.

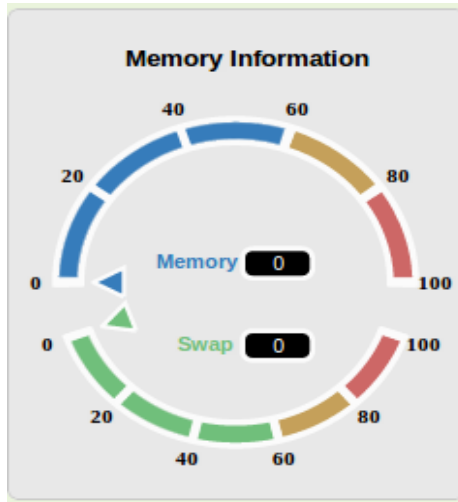
| Historia de Usuario | |
|--|---|
| Número: HU_1 | Nombre Historia de Usuario: Obtener uso de la memoria. |
| Modificación de Historia de Usuario Número: Ninguna | |
| Usuario: Orlando Miranda Gómez | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 3 semanas |
| Riesgo en Desarrollo: Alto | Puntos Reales: 3 semanas |

Capítulo II: Características, análisis y diseño del sistema

Descripción: La presente historia de usuario tiene como objetivo obtener el consumo de la memoria RAM y de la memoria de intercambio (Swap) de una máquina en tiempo real. La información será dada en por ciento y en caso de que los valores se encuentren entre 90 y 100 se le notificará al administrador.

Observaciones: Ninguna

Prototipo de interfaz:



Para el desarrollo de la anterior HU se diseñan las tareas de ingenierías siguientes:

Tabla # 4 Tarea Obtener consumo de la memoria RAM.

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 1.1 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Obtener consumo de la memoria RAM. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semanas |
| Fecha Inicio: 7/11/11 | Fecha Fin: 11/11/11 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: Esta tarea de ingeniería tiene como objetivo obtener toda la información necesaria del uso de la memoria física del sistema o computadora. | |

Capítulo II: Características, análisis y diseño del sistema

Tabla # 5 Tarea Obtener consumo de la memoria Swap.

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 1.2 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Obtener consumo de la memoria Swap. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semana |
| Fecha Inicio: 14/11/11 | Fecha Fin: 18/11/11 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: Esta tarea de ingeniería tiene como objetivo obtener toda la información necesaria del uso de la memoria Swap de la computadora en caso de que tenga Linux como sistema operativo. | |

Tabla # 6 Tarea Mostrar uso de la memoria RAM.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 1.3 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Mostrar el uso de la memoria RAM. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 2 días |
| Fecha Inicio: 01/02/12 | Fecha Fin: 03/02/12 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: Esta tarea de ingeniería tiene como objetivo mostrar los datos de la memoria RAM en por ciento y como resultado que funcione correctamente el medidor que lo muestra. | |

Tabla # 7 Tarea Mostrar el uso de la memoria Swap.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 1.4 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Mostrar el uso de la memoria Swap. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 2 días |

Capítulo II: Características, análisis y diseño del sistema

| | |
|--|----------------------------|
| Fecha Inicio: 06/02/12 | Fecha Fin: 08/02/12 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: Esta tarea de ingeniería tiene como objetivo mostrar los datos en por ciento y hacer que el reloj funcione de acuerdo a ellos. | |

Tabla # 8 Tarea Corregir el alcance del uso de la memoria.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 1.5 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Corregir el alcance del uso de la memoria. | |
| Tipo de Tarea : Corrección | Puntos Estimados: 2 días |
| Fecha Inicio: 08/02/12 | Fecha Fin: 10/02/12 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: La presente tarea tiene como objetivo corregir un error en el medidor (reloj) que permite mostrar los datos de la memoria RAM. Este medidor estaba teniendo problemas a la hora de mostrar los datos. | |

A continuación se presenta la HU Obtener consumo del microprocesador y las tareas de ingenierías relacionadas.

Tabla # 9 HU Obtener consumo del microprocesador.

| Historia de Usuario | |
|--|---|
| Número: HU_2 | Nombre Historia de Usuario: Obtener consumo del microprocesador. |
| Modificación de Historia de Usuario Número: Ninguna | |
| Usuario: Orlando Miranda Gómez | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 3 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 3 |
| Descripción: La presente historia de usuario tiene como objetivo obtener el | |

Capítulo II: Características, análisis y diseño del sistema

consumo del microprocesador de una computadora. En caso de estar entre el 90 y 100 por ciento se le notificará al administrador.

Observaciones: Ninguna

Prototipo de interfaz:

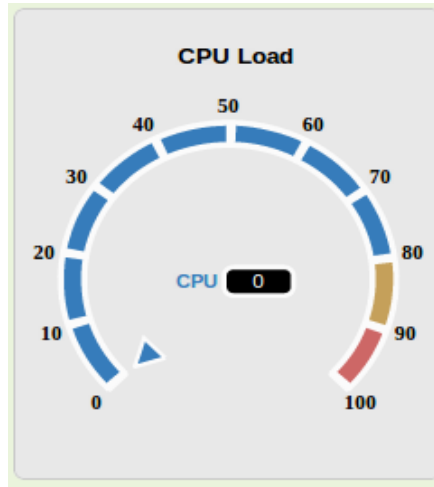


Tabla # 10 Tarea Obtener carga del microprocesador.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 2.1 | Número Historia de Usuario: HU_2 |
| Nombre Tarea: Obtener carga del microprocesador. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 2 semanas |
| Fecha Inicio: 21/11/11 | Fecha Fin: 02/12/11 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: La presente tarea de ingeniería tiene como objetivo obtener el uso que realiza la computadora sobre el microprocesador. | |

Tabla # 11 Tarea Mostrar el uso del microprocesador.

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 2.2 | Número Historia de Usuario: HU_2 |
| Nombre Tarea: Mostrar el uso del microprocesador. | |

Capítulo II: Características, análisis y diseño del sistema

| | |
|--|-----------------------------------|
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semana |
| Fecha Inicio: 13/02/12 | Fecha Fin: 17/02/12 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: Esta tarea de ingeniería tiene como objetivo mostrar los datos obtenidos en cuanto al uso del microprocesador de la computadora. La información se mostrará en por ciento. | |

Para el desarrollo de la HU Obtener uso del disco duro del sistema se diseña la siguiente tabla y las tareas de ingenierías relacionadas.

Tabla # 12 HU Obtener uso del disco duro del sistema.

| Historia de Usuario | |
|---|--|
| Número: HU_3 | Nombre Historia de Usuario: Obtener uso del disco duro del sistema. |
| Modificación de Historia de Usuario Número: Ninguna | |
| Usuario: Orlando Miranda Gómez | Iteración Asignada: 2 |
| Prioridad en Negocio: Media | Puntos Estimados: 2 |
| Riesgo en Desarrollo: Medio | Puntos Reales: 1 |
| Descripción: La presente historia de usuario tiene como objetivo mostrar el espacio total y utilizado del disco duro donde se encuentra instalado el sistema de la computadora. Se podrá mostrar su estado en distintas medidas, ejemplo: Kb, Mb y Gb. | |
| Observaciones: Ninguna | |

Prototipo de interfaz:

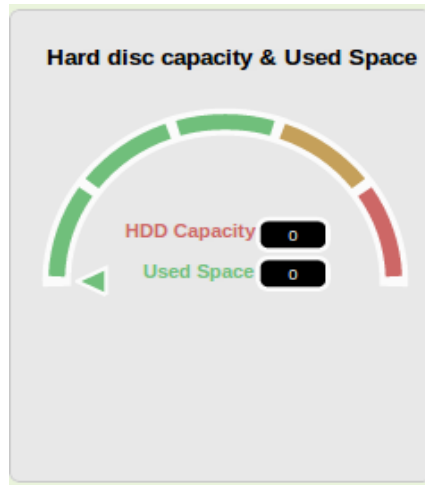


Tabla # 13 Tarea Obtener uso y espacio del disco duro.

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 3.1 | Número Historia de Usuario: HU_3 |
| Nombre Tarea: Obtener uso y espacio del disco duro. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 3 días |
| Fecha Inicio: 05/12/11 | Fecha Fin: 07/12/11 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: La presente tarea tiene como objetivo obtener el espacio total y utilizado del disco duro que utiliza el sistema operativo. Esto significa que se obtendrá la información respecto a la partición donde se encuentra el sistema instalado. | |

Tabla # 14 Tarea Mostrar el uso del disco duro.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 3.2 | Número Historia de Usuario: HU_3 |
| Nombre Tarea: Mostrar el uso del disco duro. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 2 días |

Capítulo II: Características, análisis y diseño del sistema

| | |
|---|----------------------------|
| Fecha Inicio: 08/12/11 | Fecha Fin: 09/12/11 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: Esta tarea de ingeniería tiene como objetivo mostrar los datos que se obtienen del disco duro, ya sea espacio utilizado o libre, estos se mostrarán según la unidad de medida en que se encuentre, es decir en Kb, Mb o Gb. En caso de que el espacio libre del disco sea menos de 5Gb se le notificará al administrador de la computadora. | |

Tabla # 15 Tarea Notificar consumo de los recursos.

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 3.2 | Número Historia de Usuario: HU_3 |
| Nombre Tarea: Notificar consumo de los recursos. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 5 semanas |
| Fecha Inicio: 12/12/11 | Fecha Fin: 03/02/12 |
| Programador Responsable: Orlando Miranda Gómez | |
| Descripción: La presente tarea tiene como objetivo informar en el momento debido al usuario o administrador de la computadora por vía de correo o mensajes SMS, de que uno o varios de los recursos monitorizados están en un alto nivel de su consumo o presentan algún error y poder prevenir esto. | |

Tabla # 16 Tarea Mejorar muestra de uso y espacio del disco duro.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 3.3 | Número Historia de Usuario: HU_3 |
| Nombre Tarea: Mejorar muestra de uso y espacio del disco duro. | |
| Tipo de Tarea : Mejora | Puntos Estimados: 2 días |
| Fecha Inicio: 20/02/12 | Fecha Fin: 22/02/12 |
| Programador Responsable: Orlando Miranda Gómez | |

Capítulo II: Características, análisis y diseño del sistema

Descripción: La presente tarea tiene como objetivo mejorar el diseño de la aplicación en cuanto a la visualización de la información del disco duro para lograr un mejor entendimiento visual para el usuario. Aquí se hace una modificación de los colores del reloj del disco duro.

Tabla # 17 HU Notificar consumo elevado de los recursos.

| Historia de Usuario | |
|--|---|
| Número: HU_4 | Nombre Historia de Usuario: Notificar consumo elevado de los recursos. |
| Modificación de Historia de Usuario Número: Ninguna | |
| Usuario: Orlando Miranda Gómez | Iteración Asignada: 3 |
| Prioridad en Negocio: Alta | Puntos Estimados: 7 |
| Riesgo en Desarrollo: Alta | Puntos Reales: 6 |
| Descripción: La presente historia de usuario tiene como objetivo notificar al usuario o administrador de la computadora a través de correo o mensajes SMS, para alertar que algunos de los recursos monitorizados están en un alto nivel de su consumo y poder prevenir esto. | |
| Observaciones: Ninguna | |
| Prototipo de interfaz: Esta historia de usuario no tiene prototipo. | |

Tabla # 18 Tarea Notificar por correo.

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 4.1 | Número Historia de Usuario: HU_4 |
| Nombre Tarea: Notificar por correo. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semana |
| Fecha Inicio: 06/02/12 | Fecha Fin: 10/02/12 |
| Programador Responsable: Orlando Miranda Gómez | |

Descripción: La presente tarea tiene como objetivo notificar al usuario o administrador de la computadora a través de correo, para alertar que algunos de los recursos monitorizados están en un alto nivel de su consumo y poder prevenir esto.

2.6. Plan de Releases.

En el plan de *release* se define cuales son las historias de usuario más significativas y se ubican en las iteraciones según esta prioridad. Permite dividir el proceso de desarrollo de software en iteraciones, planificando el trabajo a realizar en cada una de ellas. También tiene como entrada la relación de historias de usuario definidas previamente.

Como resultado se llegó a la planificación que se muestra en la Tabla # 19:

Tabla # 19 Plan de Releases.

| Releases | Descripción de la iteración | Orden de la HU a implementar | Duración total |
|-------------|---|------------------------------|----------------|
| Iteración 2 | En esta iteración se desarrollará una primera versión de la aplicación. El objetivo principal de esta iteración es obtener toda la información necesaria de los recursos. | HU_1 HU_2 HU_3 | 7 semanas |
| Iteración 3 | En esta iteración se va a notificar al usuario por vía de correos para informar si hay algunos recursos que tenga un elevado nivel de consumo. | HU_4 | 6 semanas |

2.7. Descripción de la arquitectura.

La arquitectura de una aplicación es importante en la construcción del mismo ya que brinda una visión más estructurada posible del sistema, establece el papel de las herramientas y los patrones de diseño. Con el objetivo de desarrollar la

aplicación Monitoring Demo se utiliza una arquitectura en capas como se muestra en la Fig. 2. Se escoge esta arquitectura porque es más fácil para el desarrollo de una capa sin que influyan cambios en otra. Esta arquitectura tiene como objetivo primordial separar la lógica de negocios de la lógica de diseño.



Fig. 2 Arquitectura de la aplicación.

En la capa de presentación se tiene los navegadores soportados por el protocolo WebSocket. Muestra una interfaz grafica amigable al usuario donde le comunica la información necesaria. Esta capa se comunica con la capa de negocio. La capa de negocio está representada por el servidor jWebSocket que proporcionara el monitoreo y se le envía los resultados al usuario después del proceso.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

2.8. Diseño con metáforas.

El diseño con metáforas conforma el vocabulario para realizar la descripción del problema, es el diseño de la solución más simple que pueda funcionar y ser

implementado en determinados momentos del proyecto.

Es además el diseño de la solución más simple que pueda funcionar y ser implementado en un momento dado del proyecto; lo cual genera el artefacto conocido como Modelo de Diseño. Este diseño está integrado por un Diagrama de Paquetes que muestran los elementos físicos del sistema así como las relaciones existentes entre ellos y las dependencias lógicas entre paquetes de software. Además muestra las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados. Estos muestran además la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables.

A continuación se representa el diagrama de paquetes para el sistema que se propone Fig. 3:

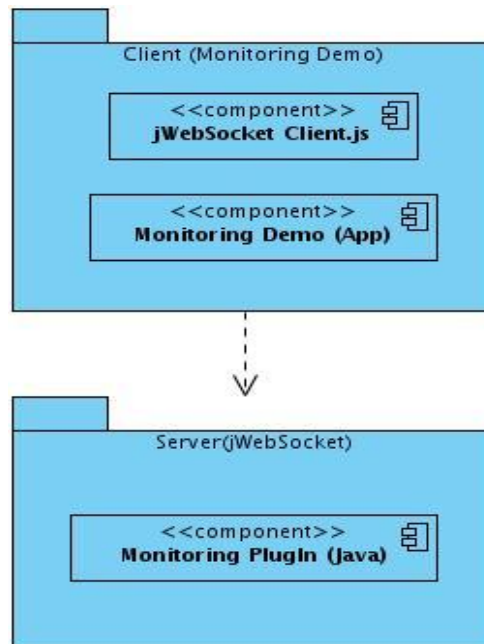


Fig. 3 Diagrama de paquetes.

A continuación se describen los elementos del diagrama anterior:

En el paquete **Client** se tienen los componentes Monitoring Demo que representa la aplicación y jWebSocket Client que constituye el cliente del marco de trabajo jWebSocket. Esta capa del cliente de Monitoring Demo se relaciona con la capa del servidor de jWebSocket.

El paquete **Server** contiene el *plug-in* de la aplicación desarrollado con jWebSocket que permite obtener información de la computadora.

2.9. Diseño de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas.

A continuación en la Fig. 4 se representa el diagrama de componentes para el sistema que se propone.

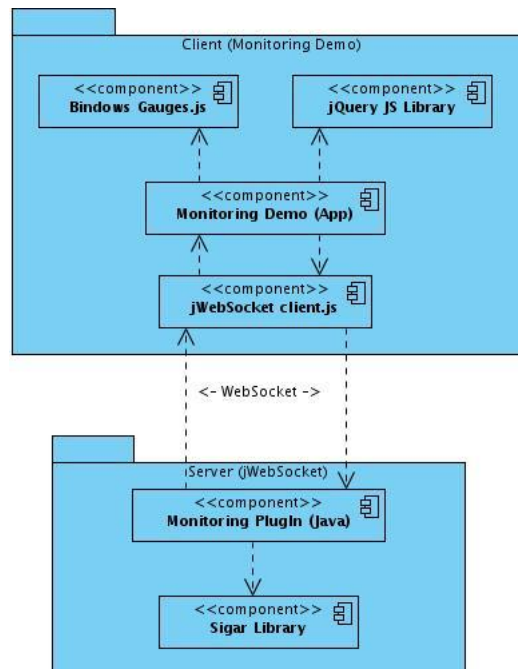


Fig. 4 Diagrama de Componentes.

A continuación se describen los elementos del diagrama anterior:

- ✓ La aplicación Monitoring Demo funciona mediante Bindows Gauges que es una librería Java Script y utiliza como plantillas para las pantallas archivos XML llamados *Application Description Files (ADF)*, en español Archivos de descripción de aplicación; y se ejecuta sólo del lado del cliente.
- ✓ La aplicación Monitoring Demo recibe los datos a través del cliente *jQuery* y consume funcionalidades de la librería *jQuery*.
- ✓ El cliente de *jQuery* se comunica con el servidor por medio del protocolo *WebSocket*.
- ✓ El servidor de *jQuery* contiene el *plug-in* de la aplicación que se comunica con la librería *Sigar* encargada de obtener cualquier tipo de información de los recursos de la computadora.

Conclusiones del capítulo.

En la realización de este capítulo se definieron el diseño propuesto por el analista permitiendo adquirir una comprensión de todo lo relacionado con los requisitos funcionales y no funcionales del sistema, quedando aprobados para servir de base para el comienzo del desarrollo del mismo. Se obtiene la herramienta Monitoring Demo como propuesta de solución para la visualización de los estados de los recursos de la computadora en tiempo real utilizando el marco de trabajo *jQuery*.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Introducción.

En este capítulo se generan los artefactos correspondientes a la fase de desarrollo y se comienza a realizar la implementación de la solución que fue propuesta en el capítulo anterior. Además se genera el diagrama de despliegue del sistema desarrollado. También se le realizan al sistema pruebas funcionales. Con el objetivo de lograr un producto con calidad que a su vez cumpla con el funcionamiento esperado y permita al cliente determinar su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

3.1. Diagrama de despliegue.

El diagrama de despliegue permite observar cómo funciona la estructura de la distribución física del sistema. Es un tipo de diagrama del Lenguaje Unificado de Modelado que muestran las relaciones físicas de los distintos nodos que componen un sistema y sus relaciones. Describen la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. A continuación en la Fig. 5 se muestra el diagrama de despliegue donde se distribuyen las funcionalidades entre los nodos que conforman el sistema.



Fig. 5 Diagrama de despliegue.

Cliente: El cliente muestra la información mediante la librería Bindows Gauges a través de Java Script, esta información es dada por medio del protocolo WebSocket para lo cual solo necesitará un navegador que soporte este protocolo sobre cualquier sistema operativo.

Servidor: El servidor jWebSocket se conectara a la librería Sigar mediante el *plugin* Monitoring para obtener toda la información del sistema de la computadora.

3.2. Validación de la investigación.

En esta investigación se decide trazar una estrategia para validar la misma. Teniendo como objetivo desarrollar una aplicación web con el marco de trabajo jWebSocket que garantice mayor disponibilidad de los recursos de una computadora mediante el monitoreo en tiempo real en la Web. La validación está basada en tres etapas que se mencionan a continuación.

En la primera etapa se realizan pruebas funcionales vinculadas a las historias de usuarios. Estas pruebas se hacen con el objetivo de verificar los requisitos funcionales que debe cumplir la solución.

Seguidamente se pasa a un proceso de certificación de calidad del software realizado por el grupo de calidad del centro de desarrollo de la facultad regional. Este grupo de calidad hace la certificación principalmente basada en la funcionalidad, estandarización y limpieza del código y revisa los artefactos documentales que son generados por la metodología SXP, seleccionada para el desarrollo de esta investigación.

Como última etapa se decide poner a consideración la aplicación desarrollada a criterio del cliente Alexander Schulze. En esta etapa de validación se consigue que este especialista en la temática verifique la funcionalidad de la aplicación integrada a jWebSocket y su usabilidad para los usuarios de la comunidad.

3.2.1 Pruebas funcionales.

Se denominan Pruebas Funcionales o *Functional Testing*, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software.

Durante el desarrollo de esta aplicación se realizaron un conjunto de pruebas a las que fue sometido el sistema para comprobar el funcionamiento de acuerdo a las

Capítulo III: Implementación y validación del sistema

historias de usuario como se muestran a continuación.

Tabla # 20 Casos de Prueba para la HU Obtener uso de la memoria.

| Caso de Prueba Funcional | |
|--|--|
| Código Caso de Prueba: jws-1-1 | Nombre Historia de Usuario: Obtener uso de la memoria |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en obtener la información exacta de la memoria en el momento. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para verificar esto el usuario tiene que observar si la aguja del reloj “ <i>Memory Information</i> ” se mueve hacia alguna posición. | |
| Resultado Esperado: Obtención del dato de la memoria en el momento. | |
| Evaluación de la Prueba: Satisfactoria. | |

| Caso de Prueba Funcional | |
|--|--|
| Código Caso de Prueba: jws-1-2 | Nombre Historia de Usuario: Obtener uso de la memoria |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en verificar si los datos obtenidos corresponden con los que brinda el monitor del sistema que trae el sistema operativo por defecto. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para verificar esto el usuario tiene que observar el funcionamiento de esta aplicación web y compararlo con el monitor del | |

Capítulo III: Implementación y validación del sistema

sistema.

Resultado Esperado: Los datos que se muestran oscilan en el intervalo que muestra el monitor del sistema.

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba Funcional

| | |
|---------------------------------------|--|
| Código Caso de Prueba: jws-1-3 | Nombre Historia de Usuario: Obtener uso de la memoria |
|---------------------------------------|--|

Nombre de la persona que realiza la prueba: Orlando Miranda Gómez

Descripción de la Prueba: Esta prueba consiste en verificar si los datos se obtienen de igual forma cuando se visualizan a la vez en navegadores abiertos en otras computadoras.

Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación.

Entrada / Pasos de ejecución: Para verificar esto el usuario debe abrir esta aplicación web desde distintas computadoras.

Resultado Esperado: Se muestran los mismos datos en todos los navegadores a igual velocidad.

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba Funcional

| | |
|---------------------------------------|--|
| Código Caso de Prueba: jws-1-4 | Nombre Historia de Usuario: Obtener uso de la memoria |
|---------------------------------------|--|

Nombre de la persona que realiza la prueba: Orlando Miranda Gómez

Descripción de la Prueba: Esta prueba consiste en verificar que la sucesión de datos enviados por el servidor sean actualizados en el cliente de forma continua.

Capítulo III: Implementación y validación del sistema

Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación.

Entrada / Pasos de ejecución: Para verificar esto el usuario debe observar si las agujas correspondientes con el reloj de memoria se mueven continuamente.

Resultado Esperado: Se puede observar el movimiento del reloj mostrando los datos del consumo de la memoria.

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba Funcional

| | |
|---------------------------------------|--|
| Código Caso de Prueba: jws-1-5 | Nombre Historia de Usuario: Obtener uso de la memoria |
|---------------------------------------|--|

Nombre de la persona que realiza la prueba: Orlando Miranda Gómez

Descripción de la Prueba: Esta prueba consiste en verificar que el usuario pueda observar el uso de la memoria física y Swap en caso de que tenga como sistema operativo GNU/Linux.

Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación.

Entrada / Pasos de ejecución: Para observar el desempeño que está realizando la memoria RAM de la computadora el usuario solo tendrá que ver la parte superior del reloj con el nombre “*Memory Information*”.

Resultado Esperado: El usuario puede observar el estado de la memoria RAM.

Evaluación de la Prueba: Satisfactoria.

Capítulo III: Implementación y validación del sistema

Tabla # 21 Casos de pruebas para la HU Obtener consumo del microprocesador.

| Caso de Prueba Funcional | |
|--|--|
| Código Caso de Prueba: jws-2-1 | Nombre Historia de Usuario: Obtener consumo del microprocesador |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en obtener la información exacta del microprocesador en el momento. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para verificar esto el usuario tiene que observar si la aguja del reloj “ <i>CPU Load</i> ” se mueve hacia alguna posición. | |
| Resultado Esperado: Obtención del dato del microprocesador en el momento. | |
| Evaluación de la Prueba: Satisfactoria. | |

| Caso de Prueba Funcional | |
|--|--|
| Código Caso de Prueba: jws-2-2 | Nombre Historia de Usuario: Obtener consumo del microprocesador |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en verificar si los datos obtenidos del CPU corresponden con los que brinda el monitor del sistema que trae el sistema operativo por defecto. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para verificar esto el usuario tiene que observar el funcionamiento de esta aplicación web y compararlo con el monitor del sistema. | |

Capítulo III: Implementación y validación del sistema

Resultado Esperado: Los datos que se muestran oscilan en el intervalo que muestra el monitor del sistema.

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba Funcional

| | |
|---------------------------------------|--|
| Código Caso de Prueba: jws-2-3 | Nombre Historia de Usuario: Obtener consumo del microprocesador |
|---------------------------------------|--|

Nombre de la persona que realiza la prueba: Orlando Miranda Gómez

Descripción de la Prueba: Esta prueba consiste en verificar que los datos del CPU se obtengan de igual forma cuando se visualizan al mismo tiempo en navegadores abiertos en otras computadoras.

Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación.

Entrada / Pasos de ejecución: Para verificar esto el usuario debe abrir esta aplicación web desde distintas computadoras.

Resultado Esperado: Se muestran el estado del CPU en todos los navegadores a igual velocidad.

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba Funcional

| | |
|---------------------------------------|--|
| Código Caso de Prueba: jws-2-4 | Nombre Historia de Usuario: Obtener consumo del microprocesador |
|---------------------------------------|--|

Nombre de la persona que realiza la prueba: Orlando Miranda Gómez

Descripción de la Prueba: Esta prueba consiste en verificar que la sucesión de datos del microprocesador enviados por el servidor sean actualizados en el cliente de forma continua.

Capítulo III: Implementación y validación del sistema

Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación.

Entrada / Pasos de ejecución: Para verificar esto el usuario debe observar si la aguja del reloj “CPU Load” se mueven continuamente.

Resultado Esperado: Se puede observar el movimiento del reloj mostrando los datos del uso del micro.

Evaluación de la Prueba: Satisfactoria.

Tabla # 22 Casos de pruebas para la HU Obtener uso del disco duro del sistema.

| Caso de Prueba Funcional | |
|--|---|
| Código Caso de Prueba: jws-3-1 | Nombre Historia de Usuario: Obtener uso del disco duro del sistema |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en mostrarle al usuario los estados del disco duro. Estos estados serán libres y ocupados de la partición usada por el sistema operativo. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para observar el estado del disco duro del sistema el usuario debe ir al reloj con el nombre de “ <i>Hard disc capacity & Used Space</i> ”. | |
| Resultado Esperado: El usuario puede observar el estado del disco duro. | |
| Evaluación de la Prueba: Satisfactoria. | |

| Caso de Prueba Funcional | |
|---------------------------------------|--|
| Código Caso de Prueba: jws-3-2 | Nombre Historia de Usuario: Obtener uso |

Capítulo III: Implementación y validación del sistema

| | |
|--|----------------------------|
| | del disco duro del sistema |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en que el usuario pueda observar la capacidad total y espacio usado de la partición del disco duro en donde se encuentra el sistema operativo. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para observar la capacidad total del disco duro el usuario debe fijarse en el valor que aparece en el cuadro “ <i>HDD Capacity</i> ”. | |
| Resultado Esperado: El usuario puede observar el total del disco duro utilizado para el sistema. | |
| Evaluación de la Prueba: Satisfactoria. | |

| Caso de Prueba Funcional | |
|--|---|
| Código Caso de Prueba: jws-3-3 | Nombre Historia de Usuario: Obtener uso del disco duro del sistema |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en verificar que la cantidad total y usada del disco duro sea la correcta. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para observar el estado del disco duro el usuario debe comprobar que los datos mostrados del disco correspondan a las propiedades que tiene la partición en la que está instalado el sistema operativo. | |
| Resultado Esperado: El usuario puede observar el estado del disco duro. | |

Capítulo III: Implementación y validación del sistema

Evaluación de la Prueba: Satisfactoria.

Tabla # 23 Casos de pruebas para la HU Notificar consumo elevado de los recursos.

| Caso de Prueba Funcional | |
|--|--|
| Código Caso de Prueba: jws-4-1 | Nombre Historia de Usuario: Notificar consumo elevado de los recursos |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en verificar si es notificado el usuario administrador de la computadora de un alto consumo del microprocesador. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para la notificación del consumo elevado del microprocesador el valor debe estar entre el 90 y 100 por ciento. Después el usuario deberá revisar su correo electrónico para recibir un correo con un aviso de alerta. | |
| Resultado Esperado: Le llega un correo de aviso al usuario. | |
| Evaluación de la Prueba: Satisfactoria. | |

| Caso de Prueba Funcional | |
|--|--|
| Código Caso de Prueba: jws-4-2 | Nombre Historia de Usuario: Notificar consumo elevado de los recursos |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en verificar si es notificado el usuario administrador de la computadora de un alto consumo de la memoria. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |

| |
|---|
| Entrada / Pasos de ejecución: Para esta notificación de consumo elevado el valor de la memoria debe estar entre el 90 y 100 por ciento. Después el usuario deberá revisar su correo electrónico para recibir un correo con un aviso de alerta. |
| Resultado Esperado: Le llega un correo de aviso al usuario. |
| Evaluación de la Prueba: Satisfactoria. |

| Caso de Prueba Funcional | |
|--|--|
| Código Caso de Prueba: jws-4-3 | Nombre Historia de Usuario: Notificar consumo elevado de los recursos |
| Nombre de la persona que realiza la prueba: Orlando Miranda Gómez | |
| Descripción de la Prueba: Esta prueba consiste en verificar si es notificado el usuario administrador de la computadora de poco espacio disponible de disco duro. | |
| Condiciones de Ejecución: Para realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y un usuario tiene que estar autenticado en la aplicación. | |
| Entrada / Pasos de ejecución: Para esta notificación de consumo elevado debe quedar 5 Gb espacio libre del disco. Después el usuario deberá revisar su correo electrónico para recibir un correo con un aviso de alerta. | |
| Resultado Esperado: Le llega un correo de aviso al usuario. | |
| Evaluación de la Prueba: Satisfactoria. | |

3.2.2 Certificación de calidad.

Los artefactos generados por la metodología SXP relacionados a la investigación se pusieron a consideración del grupo de calidad de software del centro para lograr su certificación. Estos artefactos como se muestran a continuación explican todo el proceso de análisis, diseño e implementación de la aplicación:

- ✓ Plantilla de Arquitectura de Software.
- ✓ Plantilla de Estándar de Código.

- ✓ Modelo de diseño.
- ✓ Plantilla de releases.
- ✓ Plantilla de tareas de ingeniería.
- ✓ Plantilla de Historias de Usuario.
- ✓ Plantilla de Lista de Reserva del Producto.
- ✓ Casos de pruebas de aceptación.

Este grupo de calidad conformado por los ingenieros Maidel Ojeda Castro y Domma Moreno Dargel quienes llevaron además un proceso de revisión de la estandarización, calidad y funcionamiento del código fuente en colaboración de Alexander Schulze quien es también asesor de tecnología.

3.2.3 Valoración del cliente.

Para validar la presente investigación se contó con el criterio del especialista Alexander Schulze, quien es líder de la comunidad de jWebSocket internacional y arquitecto principal del proyecto jWebSocket. Para garantizar la revisión por parte del cliente se crea un SVN internacional de jWebSocket para subir la aplicación y el código de la misma, junto con los manuales de instalación, usuario y desarrollo en inglés. Estos manuales tienen como objetivos documentar los procesos de instalación y descarga, brindar una perspectiva de cómo se usa y lograr un entendimiento del código fuente de la aplicación respectivamente.

Alexander Schulze es director ejecutivo y director general de la empresa In-notrade GmbH. Desarrollador de software y consultor de Tecnologías de la Información durante más de 20 años. Es entrenador de especialistas en Tecnologías de la Información y Máster en SCRUM. Participa activamente como ponente en conferencias internacionales y ha publicado varios libros y artículos sobre diversos temas de las Tecnologías de la Información.

3.3. Resultados obtenidos.

Debido al resultado satisfactorio de las pruebas funcionales realizadas se

demuestra el cumplimiento de los requerimientos identificados. Por parte del Grupo de Certificación de Calidad se obtuvo resultados positivos garantizando que la aplicación y su documentación poseen la calidad requerida (Ver Anexo # 1 Certificación de Calidad de Software.). A través de la valoración del cliente se obtuvo un aval de calidad. Este certificado demuestra que la aplicación y su documentación en inglés tienen el nivel suficiente para ser utilizados por la Comunidad e integrados al proyecto internacional de jWebSocket (Ver Anexo # 2 Valoración del cliente.).

Los resultados que fueron obtenidos mediante estas pruebas realizadas permiten resaltar que está disponible la aplicación en su primera versión 1.0, lo cual significa que se puede esperar resultados en versiones posteriores. Se obtuvo un sistema que cumple con todas las especificaciones y funcionalidades para el chequeo de los recursos de una computadora mediante la Web en tiempo real mejorando los niveles de disponibilidad de la información que proporciona la aplicación.

3.4. Funcionalidades obtenidas.

Entre las principales funcionales que posee Monitoring Demo en su versión 1.0 se pueden mencionar:

- ✓ Permite visualizar el uso del microprocesador de su computadora a través de la Web.
- ✓ Muestra el consumo de la memoria RAM y la Swap.
- ✓ Visualización del espacio libre y utilizado del disco duro. Este espacio de es la partición donde se encuentra instalado el sistema operativo.
- ✓ Notificación mediante correo electrónico del uso elevado de los recursos mencionados anteriormente.

3.5. Aporte social y económico.

Monitoring Demo es una aplicación de monitoreo de recursos en tiempo real desarrollada con el marco de trabajo jWebSocket. Podrá ser utilizado en cualquier

sector de la sociedad ya que tiene mucha importancia en la protección de los componentes de una computadora haciendo uso de la Web, garantizando mayor nivel de disponibilidad de los recursos de una computadora.

Esta aplicación muestra la capacidad de `WebSocket` en el desarrollo de aplicaciones que permiten el monitoreo de los recursos de computadoras en tiempo real mediante la Web, ya que facilita el seguimiento continuo y estable de los recursos. Teniendo como una de las ventajas mayores es que puede ser visualizado desde cualquier lugar del mundo, tan solo teniendo una conexión a Internet y desde cualquier dispositivo tanto estacionario como móvil.

Esta aplicación web puede ser usada por cualquier institución que tenga la intención de mantener en constante vigilancia el estado de cualquiera de sus computadoras. Por el motivo de que le estén brindando un servicio importante en esta institución y requiera de su completo funcionamiento. Además es una aplicación de código abierto, lo que permite que no haya costos para su uso y se pueda modificar y ampliar sus funcionalidades.

Conclusiones del capítulo.

En este capítulo al concluir el desarrollo de las pruebas al sistema quedó demostrado con el análisis de los resultados obtenidos, que las funcionalidades alcanzadas por `Monitoring Demo`, se han desarrollado de acuerdo a los requerimientos definidos en la etapa inicial. Proporcionando así informaciones de gran importancia para el mantenimiento, soporte y eficiencia de los recursos físicos de una computadora en tiempo real. Se realizó el diagrama de despliegue que indica la situación física de los componentes lógicos desarrollados.

CONCLUSIONES GENERALES

Al culminar la presente investigación se obtiene como conclusiones las respuestas a las preguntas científicas definidas al inicio de esta:

- ✓ La realización de la fundamentación teórico-metodológica permitió la conceptualización del monitoreo de computadoras, facilitando además el flujo de información de los recursos de estas en la Web.
- ✓ Existen diversos tipos de monitoreo web con muchas funcionalidades pero la mayoría son privativos o no trabajan en tiempo real.
- ✓ La aplicación web Monitoring Demo desarrollada permite el monitoreo de computadoras en tiempo real usando jWebSocket y garantiza mayor nivel de disponibilidad de sus recursos mediante la Web.
- ✓ Se comprobó la capacidad y potencialidad de la aplicación desarrollada para el monitoreo de recursos de computadoras en tiempo real, usando estrategias de validación para la presente investigación garantizando la calidad de la documentación y su código fuente.

RECOMENDACIONES

- ✓ Incluir más componentes físicos de la computadora a monitorizar.
- ✓ Desarrollar de ser posible en otras versiones el envío de notificaciones a través de SMS.

BIBLIOGRAFÍA REFERENCIADA

¿Qué es un 'framework'? **SÁNCHEZ, Jordi. 2006.** Valencia : <http://jordisan.net/blog/2006/que-es-un-framework>, 2006.

ALBALADEJO, Xavier. proyectos agiles.org. [En línea] ScrumAlliance. [Citado el: 5 de 2 de 2012.] <http://www.proyectosagiles.org/que-es-scrum>.

Apache Software Foundation. 2011. Apache Subversion. *Apache Subversion*. [En línea] 10 de 12 de 2011. [Citado el: 10 de 12 de 2011.] <http://subversion.apache.org/>.

CELIS, Leonardo Alberto. 2003-2011. The leading object-oriented platform for developing AJAX enterprise applications. [En línea] MB Technologies, 2003-2011. <http://www.bindows.net/worldwide/LatinAmerica.html>.

Cristian. 2006-2010. Capturar informacion sistema operativo java. [En línea] 2006-2010. [Citado el: 20 de 10 de 2011.] <http://casidiablo.net/capturar-informacion-sistema-operativo-java/>.

CHACON, Scott. Git the fast version control system. [En línea] Software Freedom Conservancy. [Citado el: 11 de 12 de 2011.] <http://git-scm.com/about>.

CHAVARRÍA, Erick Camacho. 2010. *Tutorial Introduccion a Maven 3*. s.l. : javaHispano, 2010. http://www.javahispano.org/storage/contenidos/Tutorial_de_Maven_3_Erick_Camacho.pdf.

Eclipse. 2011. The Eclipse Foundation Open Source Community Website. [En línea] Eclipse Foundation, 2011. <http://www.eclipse.org/>.

EGUÍLUZ Pérez, Javier. 2009. *Introducción a CSS*. 2009.

—. 2009. *Introducción a JavaScript*. 2009.

—. 2009. *Introducción a XHTML*. 2009.

El Monitor Informa. **SERRANO, Marlene Piñarreal. 2009.** Cusco : s.n., 2009.

ESPINOZA Corrales, Leana y VAN De Velde, Herman. 2007. Monitoreo, Seguimiento y Evaluación de Proyectos Sociales. Managua : s.n., 2007.

Extjs. 2009. www.extjs.es. [En línea] 2009. [Citado el: 10 de 12 de 2011.] <http://www.extjs.es>.

FIGUEROA, Roberth G, SOLÍS, Camilo J y CABRERA, Armando A. 2011. Entorno Virtual de Aprendizaje. [En línea] 2011. [Citado el: 13 de 12 de 2011.] <http://eva.uci.cu/mod/resource/view.php?id=9303&subdir=/Metodologias/RUP>.

Framework Approach for WebSockets. **SCHULZE, Alexander.** 2011. Web Technologies & Internet Applications (WebTech 2011) : http://dl.globalstf.org/index.php?page=shop.product_details&flypage=flypage_image_s.tpl&product_id=528&category_id=42&option=com_virtuemart&Itemid=4&vmcchk=1&Itemid=4, 2011.

GONZÁLEZ Blanco, Rubén y PÉREZ Tobalina, Sergio. *LESE-2 Introducción a Rational Rose.* Catalunya, Barcelona : s.n. http://www.google.com/cu/url?sa=t&rct=j&q=rational+rose&source=web&cd=8&ved=0CFwQFjAH&url=http%3A%2F%2Fwww.essi.upc.edu%2F~ese%2Fweb%2Fdocuments%2Flab%2F0304Q2%2Flessons%2Flese-2%2FLESE-2%2520-%2520Introduccion%2520a%2520Rational%2520Rose.ppt&ei=RQdET_fo.

GOODMAN, Danny. 2001. *JavaScript Bible.* s.l. : Gold Edition, 2001.

GOSLING, James, y otros. 2005. *The Java language specification.* s.l. : Addison-Wesley, 2005. 0-321-24678-0.

ALFARO, Félix Murillo. 2011. *Herramientas Case.* COLECCION CULTURA INFORMATICA.

Inc., Yahoo! 2011. YUI Library. Yahoo! [En línea] 2011. [Citado el: 14 de 12 de 2011.] <http://developer.yahoo.com/yui/>.

ITZCOALT Alvarez, M. y JOIZ, Net. 2007. *Desarrollo Ágil con SCRUM.* 2007.

LUJAN, Mora Sergio. 2002. *Programación de aplicaciones web: historia, principios básico y clientes web.* s.l. : Club Universitario 1ra edición, 2002.

MOKATE, Karen Marie. 2000. *EL MONITOREO Y LA EVALUACIÓN: herramientas indispensables de la gerencia social.* 2000.

SHAPIRO, J. 2001. *Monitoring and Evaluation.* Johannesburg : s.n., 2001.

MORGAN, Ryan. 2010. SIGAR - System Information Gatherer And Reporter Home. [En línea] 20 de 12 de 2010. [Citado el: 15 de 10 de 2011.] <http://support.hyperic.com/display/SIGAR/Home>.

MURPHEY, Rebecca. 2010. *jQuery Fundamentals.* s.l. : Autoedición, 2010.

NetBeans. 2011. *NetBeans.* [En línea] Oracle Corporation, 2011. <http://netbeans.org/features/index.html>.

NOURIE, Dana. 2005. Oracle Sun Developer Network (SDN). [En línea] Oracle Corporation, 24 de mayo de 2005. <http://java.sun.com/developer/technicalArticles/tools/intro.html>.

Oracle Corporation. 2012. NetBeans. [En línea] 2012. [Citado el: 28 de 2 de 2012.] <http://netbeans.org/features/platform/compare.html>.

PEÑALVER, G, MENESES, A y GARCÍA, S. 2010. *SXP, Metodología Ágil para el*

Desarrollo de Software. Chile : 1er congreso Iberoamericano de Ingeniería de Proyectos, 2010. pág. 12. Universidad de las Ciencias Informáticas.

PONT, Brücke Le. 2009. Sistema de Monitoreo. Zürich : s.n., 2009.

Proyecto jQuery. 2010. Browser Compatibility. *jQuery*. [En línea] 2010. [Citado el: 8 de 12 de 2011.] http://docs.jquery.com/Browser_Compatibility.

RAE. 2009-2010. Real Academia Española. [En línea] 2009-2010. [Citado el: 5 de 12 de 2011.] http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=Monitoreo.

RapidSVN. 2011. RapidSVN. [En línea] 2011. [Citado el: 13 de 12 de 2011.] <http://www.rapidsvn.org/>.

RUMBAUGH, James y JACOBSON, Ivar. 2007. *El Lenguaje Unificado de Modelado. Manual de Referencia*. s.l. : ADDISON-WESLEY, 2007. 9788478290871.

Sencha. 2011. Ext JS 4 JavaScript Framework for Rich Apps in Every Browser. *Sencha*. [En línea] 2011. [Citado el: 14 de 12 de 2011.] <http://www.sencha.com/products/extjs/>.

SOLÍS, Manuel Calero. 2003. Una explicación de la programación extrema (XP). Madrid : s.n., 2003.

SURHONE, Lambert M, T Tennoe, Mariam y HENSSONOW, Susan F. 2010. *Real-Time Web*. s.l. : VDM Verlag, 2010. 9786133432239.

team, TortoiseSVN. 2011. TortoiseSVN. [En línea] 2011. [Citado el: 13 de 12 de 2011.] <http://tortoisesvn.net/>.

Tecnologías en evolución de los Sistemas Integrales de Información en Tiempo Real. **VILLAVICENCIO, Alejandro. 2002.** 2002.

Visual Paradigm International Ltd. 2011. Visual Paradigm for UML. *Visual Paradigm*. [En línea] 2011. [Citado el: 17 de 02 de 2012.] <http://www.visual-paradigm.com/product/vpuml/>.

BIBLIOGRAFÍA GENERAL

- ABARCA, Nanci, y otros.** *Seguridad de Redes*.
- FERNÁNDEZ, Juan Manuel Martínez.** *Aplicaciones Web en tiempo real*. 2010. <http://www.meetup.com/HTML5-Ciudad-de-Mexico>.
- GALARZA M., Fernando.** *Sistema para monitoreo y control de uso de software en la empresa*. Cuenca: s.n., 2007. Tesis previa a la obtención del Título de: Ingeniero de Sistemas.
- MOKATE, Karen Marie.** *EL MONITOREO Y LA EVALUACIÓN: herramientas indispensables de la gerencia social*. 2000.
- RUSSELL, Alex, y otros.** The Bayeux Specification. [En línea] The Dojo Foundation, 2007.
- 2006.** *Sistema De Monitoreo Con Java*. S.l.: Revista Digital Universitaria, 2006. Vol. 7.
- SOLÍS, Manuel Calero.** Una explicación de la programación extrema (XP). Madrid: s.n., 2003. pág. 11. Vol. V Encuentro usuarios xBase, <http://www.apolosoftware.com/>.
- 2010.** *System Monitoring Plug-in for Hosts for Operating System and Hardware*. S.l.: Oracle and/or its affiliates, 2010. http://www.oracle.com/enterprise_manager.
- TELLERÍA de Esteban, Miguel.** *Monitoreo de CPU y memoria*. 2010-2011. <http://www.ctr.unican.es>.
- VALLE, Otto y RIVERA, Otto.** *Monitoreo e indicadores*. S.l.: Educación Inicial y Derechos de la Infancia. pág. 20. Organización de Estados Iberoamericanos.
- 2006.** *What Lies Beneath: Host Server Provisioning, Monitoring, and Management Using Oracle Enterprise Manager Grid Control*. S.l.: Oracle Corporation, 2006.

ANEXOS

Anexo # 1 Certificación de Calidad de Software.

Anexo # 2 Valoración del cliente.

GLOSARIO DE TÉRMINOS

AJAX (Asynchronous Java Script And XML): Java Script Asíncrono y XML.

CASE (Computer Aided Software Engineering): Ingeniería de Software Asistida por Ordenador.

CSV (Comma Separated Values): Valores separados por Comas.

DHTML (Dynamic HTML): HTML Dinámico).

DOM (Document Object Model): Modelo de Objetos del Documento.

GNU: GNU es el nombre de un completo sistema de software compatible con Unix. Se denominó así, según se afirma en el Proyecto GNU, principalmente para marcar la diferencia ya que GNU era un acrónimo recursivo de "GNU No es Unix".

GPL: Licencia Pública GNU.

HTML (HyperText Markup Language): Lenguaje de Marcado de Hipertexto.

IETF (Internet Engineering Task Force): Grupo Especial sobre Ingeniería de Internet.

JDT (Java Development Toolkit): Herramientas de Desarrollo Java.

JSON (JavaScript Object Notation): Notación de Objetos de JavaScript.

MIT: Licencia del Instituto Tecnológico de Massachusetts.

XML (eXtensible Markup Language): Lenguaje de Marcas Extensible.

XSLT (Extensible Stylesheet Language Transformations): Lenguaje de Transformaciones Extensible de Hojas de Estilos.