



Facultad Regional Mártires de Artemisa

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

Título:

Representación de datos en tiempo real
mediante gráficos usando el marco de trabajo
jWebSocket.

Autor:

Merly López Barroso

Tutor:

MSc. Yamila Vigil Regalado

Cotutor:

Dr. Raúl René Crespo Heredia

Artemisa, Julio 2012

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora 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 _____.

Merly López Barroso

Firma del Autor

Msc. Yamila Vigil Regalado

Firma del Tutor

Dr. Raúl René Crespo Heredia

Firma del Cotutor

AGRADECIMIENTOS

Agradezco a mis padres, por su apoyo incondicional, por todo el esfuerzo realizado para poder culminar mis estudios satisfactoriamente, por todo el amor que me han brindado siempre y por la confianza que depositaron en mí.

A toda mi familia por su preocupación y apoyo, principalmente a mis hermanos, a mi tía y a mis abuelitos por todo su cariño y dedicación.

A mis amigos por todos los momentos compartidos a lo largo de estos cinco años de carrera. A mi equipo de trabajo por toda la colaboración durante estos meses de trabajo, en especial a Victor Antonio Barzana Crespo quien me ha ayudado en todo momento en la realización de mi tesis.

A mis tutores por toda su ayuda y orientación durante la realización de esta investigación.

A mi novio por todo su amor y cariño, por demostrarme siempre su confianza en la culminación satisfactoria de mi tesis.

Agradezco a todos los que de una manera u otra hicieron posibles la realización de mi sueño.

DEDICATORIA

Dedico este trabajo de Diploma a mis padres por toda la confianza que depositaron en mí, por su dedicación y amor incondicional. Siempre han sido mi fuente de inspiración, a ellos va dirigido todo lo bueno que soy capaz de hacer y de lograr. Gracias por existir.

A mi hermanito por apoyarme e incitarme a ser mejor cada día, le dedico los resultados de todo este esfuerzo que él me ha ayudado a alcanzar.

A mi familia por su preocupación y apoyo en todo momento.

RESUMEN

Con la invención de nuevas tecnologías, ha surgido la necesidad de analizar y dar solución a los problemas de la información. Una manera de representar la información es a través de gráficos. La Web es el mayor medio de comunicación hoy día, por lo que se hace necesario una mayor interactividad y movilidad para lograr tiempo real. Debido a esto se hace necesario representar gráficamente los datos en tiempo real. Para esto surge el nuevo protocolo de comunicación WebSocket. Uno de los marcos de trabajo que utilizan este protocolo es `jQueryWebSocket`.

Por la importancia que posee la representación gráfica de datos en la Web, se ha identificado la necesidad de aportarle al marco de trabajo esta nueva funcionalidad. Esto trae consigo altos beneficios, debido a que los desarrolladores logran identificar con claridad las potencialidades y las bondades que ofrece este marco de trabajo para la representación gráfica. Empresas y desarrolladores que necesitan graficar datos en tiempo real logran reconocer a `jQueryWebSocket` como una vía eficiente para desarrollar aplicaciones con estas funcionalidades para mayor satisfacción de sus clientes. Por otro lado la aplicación web de representación gráfica desarrollada en la investigación aumenta las cuotas de mercado para el marco de trabajo y eleva en gran medida el número de usuarios y clientes potenciales a nivel internacional.

Por lo que se desarrolla la aplicación web Charting Demo en su primera versión la cual posibilita la representación gráfica de datos en tiempo real garantizando mayor nivel de usabilidad del marco de trabajo `jQueryWebSocket`.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	8
1.1. Fundamentos Teóricos	8
1.2. Estado del Arte	10
1.3. Metodología a utilizar	14
1.4. Lenguajes de programación y modelado	18
1.4.1. Lenguaje Modelado Unificado	18
1.4.2. Java	18
1.4.3. JavaScript.....	20
1.5. Otros lenguajes	20
1.5.1. HTML.....	20
1.5.2. Hojas de Estilo en Cascada (CSS)	20
1.6. Tecnologías usadas para el desarrollo de la solución	21
1.6.1. Marco de Trabajo del lado del Servidor	21
1.6.2. Marcos de Trabajo del lado del Cliente.....	23
1.7. Herramientas a emplear para el desarrollo de la solución	24
1.7.1. Herramienta CASE	24
1.7.2. Herramientas de Control de Versiones	26
1.7.3. Herramientas del Cliente de Control de Versiones	27
1.8. Entorno integrado de desarrollo IDE	28
1.9. Sistema Gestor de Base de Datos	29
CAPÍTULO 2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA.....	33
2.1 Propuesta de Solución	33
2.2 Planificación del Proyecto por Roles	34
2.3 Modelo de Historias de Usuario del Negocio.....	34
2.4 Lista de Reserva del Producto (LRP)	36
2.5 Historias de Usuario y Tareas de Ingeniería	40

2.6	Plan de Releases	51
2.7	Arquitectura de Software de la Solución.....	52
2.8	Diseño con Metáforas.....	52
2.9	Diagrama de Componentes.....	54
CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA.....		56
3.1	Implementación	56
3.2	Validación de la Investigación.....	58
3.2.1	Casos de Pruebas	59
3.2.2	Certificación de Calidad del Software	60
3.2.3	Valoración del Cliente	61
3.2.4	Resultados Obtenidos.....	61
3.3	Funcionalidades Obtenidas	62
3.4	Diagrama de Despliegue	62
3.5	Aporte Social y Económico.....	63
CONCLUSIONES.....		65
RECOMENDACIONES		66
BIBLIOGRAFÍA REFERENCIADA		67

ÍNDICE DE TABLAS

Tabla 1:	Actor y Trabajador del Negocio	35
Tabla 2:	Lista de Reserva del Producto	36
Tabla 3:	Historia de Usuario Graficar Recursos de la máquina	41
Tabla 4:	Tarea de Ingeniería Graficar uso de cpu	42
Tabla 5:	Tarea de Ingeniería Graficar uso de memoria	42
Tabla 6:	Tarea de Ingeniería Graficar uso de swap	43
Tabla 7:	Tarea de Ingeniería Graficar espacio libre de disco duro	43
Tabla 8:	Tarea de Ingeniería Graficar espacio en uso de disco duro	44
Tabla 9:	Historia de Usuario Graficar Cantidad de Usuarios Conectados	44
Tabla 10:	Tarea de Ingeniería Graficar cantidad de usuarios conectados	45

Tabla 11: Historia de Usuario Graficar Intervalo de uso.....	46
Tabla 12: Tarea de Ingeniería Graficar peticiones por hora	47
Tabla 13:Tarea de Ingeniería Graficar peticiones por días	48
Tabla 14: Tarea de Ingeniería Graficar peticiones por mes.....	48
Tabla 15: Historia de Usuario Graficar Plug-ins en Uso.....	49
Tabla 16: Tarea de Ingeniería Graficar plug-ins en uso	50
Tabla 17: Plan de Releases	51
Tabla 18: Casos de pruebas para la HU Graficar recursos de la máquina.	59

ÍNDICE DE FIGURAS

Fig. 1 Modelo de Historia de Usuario del Negocio. Fuente: Elaboración propia.....	35
Fig. 2 : Arquitectura de la Solución. Fuente: Elaboración propia.	52
Fig. 3 Diagrama de Paquetes de la Solución Propuesta. Fuente: Elaboración propia.	53
Fig. 4 Diagrama de Componentes de la Solución Propuesta. Fuente: Elaboración propia.	54
Fig. 5: Descripción del cliente de la aplicación Stock Ticker Demo. Fuente: Elaboración propia.....	56
Fig. 6: Descripción del servidor de la aplicación Charting Demo. Fuente: Elaboración propia.	57
Fig. 7 Diagrama de Despliegue de la Solución Propuesta. Fuente: Elaboración propia.	63

INTRODUCCIÓN

Hoy día la información constituye uno de los elementos más importantes de la sociedad. El cúmulo de información es cada vez mayor en esta sociedad, a diario aumenta en altos niveles enriqueciendo a la humanidad con vastos conocimientos. Actualmente es considerada el activo intangible máspreciado. El crecimiento de la información durante la última década ha sido exponencial y se espera continúe ese comportamiento en el futuro. Ante esto surge la necesidad de disponer no solo de eficaces sistemas de almacenamiento, sino de establecer una estrategia concreta que marque pautas perfectamente definidas sobre la gestión de la información.

En la actualidad, la expresión sociedad de la información, se debe al gran uso que se hace de la información. Sin embargo, su origen se remonta a los años 70', cuando se comienza a percibir que la sociedad industrial empieza a evolucionar hacia un modelo de sociedad distinto, en la que el control y la optimización de los procesos industriales eran reemplazados por el procesamiento y manejo de la información. La expresión sociedad de la información designa una forma nueva de organización de la economía y la sociedad.

La información juega un papel primordial en el proceso de toma de decisiones. En las organizaciones que se encuentran sometidas constantemente a la toma de decisiones, la información adquiere un rol fundamental, y por ello un valor inigualable.

Debido a la invención de nuevas tecnologías a través del tiempo, ha surgido la necesidad de analizar y dar solución a los problemas de la información. Una manera de representar la información es mediante gráficos. La representación gráfica es parte del proceso de cambios y avances tecnológicos de la actualidad, dando lugar a nuevas posibilidades en los medios de comunicación. Actualmente, es una parte esencial para el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), mediante su utilización se pueden alcanzar altos niveles de eficiencia y productividad.

La representación de datos de forma gráfica ayuda a presentar la información de forma sencilla donde los resultados son fáciles de entender. Con el uso de los gráficos se agiliza el proceso, se obtiene un mayor nivel de conocimiento y se eleva

la visión y el alcance que tiene la información en sí. Al representar gráficamente los datos, el cliente puede identificar una gran cantidad de información en un período muy corto de tiempo. El uso profesional de los elementos gráficos aumenta la confianza y credibilidad del cliente hacia el producto.

La Web es la mayor vía de comunicación hoy día, es el principal medio para los negocios y el *marketing*. Hoy no constituye solo un medio para documentación e información estática, sino que la información se encuentra en constante cambio y actualización. Ante este hecho la representación en la Web posee alto grado de relevancia, dado que permite una rápida interpretación de la información para la toma de decisiones.

Las representaciones gráficas en la Web permiten a los usuarios de Internet interpretar mejor determinada información, haciéndola más entendible e interesante. Además permite a todos disponer de ella en cualquier lugar del mundo.

Para representar datos gráficamente en la Web lo primero es decidir qué tipo de gráfico se utiliza para representar la información. Esto depende en gran medida de la naturaleza de los datos y de la relación entre los elementos que componen la información. Por ejemplo, si los datos varían en el tiempo, entonces un gráfico de líneas o un gráfico de barras verticales serían mucho más apropiados que un gráfico circular. Sin embargo, si los datos representan las cifras de ventas por región, entonces tal vez un gráfico circular sería la mejor opción. Existen diversidad de gráficas, cada una de ellas posee sus particularidades. Las más comunes son las gráficas de barras o columnas, las de líneas, las de tartas o pastel así como barras apiladas.

En la era de Internet se han dado varios pasos en la accesibilidad a la información, sin embargo hoy la Web no brinda todas las facilidades que necesitan los usuarios para la representación de la información de manera gráfica. Anteriormente la Web fue concebida de manera estática, por lo que la información mantuvo este mismo estado. El flujo de datos variaba constantemente y se hacía muy engorroso mantener actualizada la información de las gráficas sin tener que estar constantemente actualizando el navegador, trayendo consigo demoras en el análisis de los datos y un bajo nivel de confidencialidad de los mismos.

El mundo actual está en constantes transformaciones y con él las nuevas tecnologías. En la evolución de la Web para lograr tiempos reales, han surgido grandes mejoras tecnológicas, que propician la eficiencia y rapidez de la información en la Web. 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)

Para brindar solución a la comunicación Web en tiempo real surge el protocolo WebSocket que define los procedimientos para actualizar la conexión a través de HTTP a una conexión mediante WebSocket totalmente bidireccional usando TCP. WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y *full-duplex*¹ sobre un único socket TCP. Este protocolo está diseñado para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor.

WebSocket no constituye hoy para la Web 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 en la Web, significa que el usuario reciba de manera inmediata toda la información sin previa solicitud.

Los principales servidores que soportan WebSocket para el desarrollo de aplicaciones hoy día son, la Pasarela WebSocket de Kaazing², JettyWebSocketServlet³, Socket.IO⁴, django-WebSocket del proyecto Python⁵ y jWebSocket⁶.

El marco de trabajo jWebSocket proporciona un mayor nivel de robustez y confiabilidad. jWebSocket es una nueva tecnología orientada al desarrollo de aplicaciones basadas en WebSocket que proporcionan altos niveles de velocidad, escalabilidad y seguridad (Framework Approach for WebSockets, 2011).

¹ Full-Duplex: Cualidad de los elementos que permiten la entrada y salida de datos de forma simultánea.

² <http://kaazing.com/>

³ <http://www.eclipse.org/jetty/>

⁴ <http://socket.io/>

⁵ <http://pypi.python.org/pypi/django-websocket>

⁶ <https://jwebsocket.org/>

Hoy el marco de trabajo cuenta con el soporte a los requerimientos necesarios para generar gráficos cuyas fuentes de datos generan nuevas muestras en intervalos muy cortos. Dicho de otro modo jWebSocket soporta la generación de gráficos con niveles de actualización de los datos en tiempo real. Sin embargo el proyecto jWebSocket no cuenta con una aplicación que demuestre su valía a la comunidad de desarrollo para servir de base tecnológica en el tema de la generación de gráficos con actualizaciones de datos en tiempo real.

A raíz de lo anteriormente planteado surge la siguiente **situación problemática**:

Hoy día no existe una vía para mostrar la capacidad del marco de trabajo jWebSocket en el desarrollo de aplicaciones que grafiquen información real con altos niveles de variación en el tiempo. Los desarrolladores no identifican con claridad las potencialidades y las bondades que ofrece el marco de trabajo jWebSocket para la representación gráfica. Esto trae como consecuencia que empresas y desarrolladores que necesitan graficar datos en tiempo real, no reconocen al marco jWebSocket como una vía eficiente para desarrollar aplicaciones con estas funcionalidades para mayor satisfacción de sus clientes. De esta manera no se logran altos niveles de usabilidad del marco jWebSocket para el desarrollo de aplicaciones web de representación gráfica. El desconocimiento de las potencialidades de jWebSocket para la representación gráfica en tiempo real trae consigo además pérdidas de cuotas de mercado para el marco de trabajo y afecta el número de usuarios y clientes potenciales a nivel internacional.

La anterior situación problemática conduce al siguiente **problema científico**:

¿Cómo garantizar mayor nivel de usabilidad del marco de trabajo jWebSocket en el desarrollo de aplicaciones que representen información gráfica en tiempo real en la Web?

Teniendo en cuenta todo lo anterior se define como **objeto de estudio**: Representación de información gráfica en tiempo real en la Web.

Dentro del objeto de estudio anterior se define como **campo de acción**: Representación de información gráfica en tiempo real en la Web para el marco de trabajo jWebSocket.

Por lo que se plantea como **objetivo general**: Desarrollar una aplicación web para la representación gráfica de información en tiempo real usando WebSocket que garantice mayor nivel de usabilidad del marco de trabajo en el desarrollo de aplicaciones gráficas en la Web.

De este se derivan las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teórico-metodológicos de la representación de información gráfica en tiempo real en la Web?
- ¿Cuál es la situación actual de la representación de información gráfica en tiempo real en la Web?
- ¿Cómo desarrollar una aplicación web para la representación gráfica de información en tiempo real con WebSocket que garantice mayor nivel de usabilidad del marco de trabajo en el desarrollo de aplicaciones gráficas en la Web?
- ¿Cuáles son los resultados obtenidos al utilizar la aplicación web desarrollada para garantizar mayor nivel de usabilidad del marco de trabajo WebSocket en el desarrollo de aplicaciones gráficas en la Web?

Para dar cumplimiento al objetivo general y las preguntas científicas se plantearon las siguientes **tareas de la investigación**:

1. Fundamentación teórico-metodológica sobre la representación de información gráfica en tiempo real en la Web.
2. Estado de la situación actual sobre la representación de información gráfica en tiempo real en la Web.
3. Desarrollo de una aplicación web para la representación gráfica de información en tiempo real con WebSocket que garantice mayor nivel de usabilidad del marco de trabajo en el desarrollo de aplicaciones gráficas en la Web.
4. Validación de la aplicación web desarrollada para garantizar mayor nivel de usabilidad del marco de trabajo WebSocket en el desarrollo de aplicaciones gráficas en la Web.

Métodos de la Investigación

Métodos teóricos:

- **Histórico-Lógico:** Permite analizar la trayectoria completa acerca de la representación gráfica en las aplicaciones web, así como el estudio histórico de las mismas que permite ver deficiencias y proponer soluciones acorde a las necesidades.
- **Análisis y Síntesis:** Mediante este método se va a analizar toda la teoría recopilada a través de los diferentes medios bibliográficos que pueda servir para desarrollar mejor el diseño del sistema, y poder aplicar así estos conocimientos en la práctica de manera que se adquiera una mayor preparación sobre el tema en cuestión.
- **Análisis Documental:** Mediante este método se realizó un estudio de una variada documentación referente a la representación gráfica, con el objetivo de obtener a través de éstas el análisis, las experiencias y las sugerencias que pudieran ser incorporadas a la tesis. Con el uso de este método se estudió además la documentación referente a las potenciales tecnologías a utilizar para desarrollar la aplicación, así como las experiencias de otros desarrolladores en el trabajo con las mismas.
- **Modelación:** Este método permite realizar una representación de la situación que se analiza. Permite obtener mediante diagramas y objetos una mayor comprensión del problema a resolver y desarrollar un modelo para la aplicación a desarrollar a partir de la situación problemática.

Método Particular:

- **Certificación de calidad de software:** Este método se utiliza en la liberación que realiza el equipo de calidad de software del Centro de Desarrollo de la “Facultad Regional Mártires de Artemisa”. Mediante este método se certifica la capacidad funcional de la aplicación desarrollada y su usabilidad.

Se cuentan con las siguientes variables:

Variable Independiente: Aplicación web para la representación gráfica de información en tiempo real usando el marco de trabajo WebSocket.

Variable Dependiente: Nivel de usabilidad del marco de trabajo WebSocket en el desarrollo de aplicaciones gráficas en la Web.

Posibles resultados:

➤ Aplicación web para la representación gráfica de información en tiempo real usando el marco de trabajo WebSocket.

Para lograr una mayor comprensión de la investigación, se ha desglosado el contenido en tres capítulos.

➤ **Capítulo 1. Fundamentación Teórica:** Se realiza la fundamentación teórica de la investigación. Se expone un estudio del estado del arte sobre el proceso de representación gráfica de datos en la actualidad, tanto a nivel nacional como internacional.

➤ **Capítulo 2. Desarrollo de la solución:** 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 las etapas de diseño e implementación que conllevan a la obtención de la aplicación.

➤ **Capítulo 3. Pruebas y validación de la solución:** Se elaboran y documentan las pruebas realizadas a la solución propuesta para demostrar el cumplimiento de los requerimientos de la misma. Se realiza un análisis de los resultados de la aplicación en un entorno de tiempo real, comparando indicadores antes y luego de la solución.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Introducción

El objetivo fundamental del presente capítulo es dejar plasmado las definiciones más significativas que sustentan la presente investigación. Se analizan las soluciones existentes con características similares. Se presenta la metodología que se va a poner en práctica, así como los lenguajes de programación, tecnologías y herramientas para el desarrollo de la solución, siempre teniendo en cuenta la necesidad del uso de tecnologías de código abierto.

1.1. Fundamentos Teóricos

En la presente investigación se definen un conjunto de conceptos con el objetivo de establecer las bases teórico-metodológicas de la investigación. Para ello se parte de la definición y los elementos que se tienen en cuenta para representar información de forma gráfica. Seguidamente se establece cómo cumplir con tiempos reales en la Web, con el objetivo de mostrar los elementos para garantizar la representación de información gráfica en tiempo real en la Web.

Hoy día se han dado varias definiciones de gráficos o representación gráfica. Uno de ellos define a los gráficos comerciales como representaciones visuales de procesos comerciales y evolución de variables de negocios. Éstos representan el resultado de ecuaciones que describen el movimiento de diferentes aspectos de la vida económica. Los gráficos estadísticos son de este tipo. (Asinsten)

Otro de los conceptos asociado a gráficos plantea que los gráficos científicos son representaciones visuales de procesos naturales, de lo medido por diferentes instrumentos de experimentación o control, del relevamiento de fenómenos del más variado tipo. (Asinsten)

Por otra parte se ha dado a conocer como representación gráfica a un “constructo desarrollado en un contexto cultural específico que interviene como mediador en la interpretación de los datos... una actividad relacionada con un complejo rango de elementos y procesos” (Monteiro, 2006) y es considerado como una herramienta de razonamiento para aprender algo nuevo acerca del contexto que representa, obteniendo nueva información o aprendizaje de los datos. (Pfannkuch, 2006) Otros

autores por su parte consideran que los gráficos son esenciales para la representación, reducción y análisis de datos en el razonamiento y pensamiento estadístico. (Shaughnessy, 2007)

También se ha definido que los gráficos o representaciones gráficas o diagramas, utilizados como sinónimos por Henry (1995), se basan en datos cuantitativos y tienen un doble propósito: visualizar la presentación de datos y estudiar y analizar la naturaleza de los datos. Todas las fases del tratamiento estadístico (examinar y describir los datos, analizar e interpretar, y resumir los datos) o del razonamiento y pensamiento estadístico, requieren del apoyo gráfico, el cual resulta en la mayoría de los casos esencial, tal y como ha sido suficientemente reconocido por diversos autores. (Curcio, 1989)(Henry, 1995)(Garfield, 2003)(Shaughnessy, 2007)

Por otro lado William Playfair plantea que un buen gráfico proporciona una explicación más adecuada de los hechos que una mera lista de datos o tablas. Sirve para simplificar lo complejo, permite al cerebro una mayor retención y es un instrumento visual de ayuda a hombres ocupados. Por último, los gráficos permiten ver relaciones aparentemente inexistentes entre variables, que suelen quedar ocultas entre la multitud de datos y cifras, de difícil comparación de otro modo. (Pérez, 2009)

Basándose en definiciones dadas por autores reconocidos, el autor de esta investigación plantea que la representación gráfica en la Web es una herramienta que permite de manera dinámica interpretar mejor la información que se analiza haciéndola más entendible e interesante. Además permite disponer de ella en cualquier lugar del mundo.

Hoy día la interactividad en la Web exige un intercambio de experiencias en tiempo real, los usuarios desean recibir retroalimentación inmediata de todas las acciones que realizan, por lo que se hace necesario definir qué es tiempo real. La definición de tiempo real en la Web ha sido dada entre otros autores por Lambert M Surhone, Mariam T Tennoe y Susan F Henssonow. Ellos definen tiempo real en la Web como 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)

Por otro lado Alexander Schulze emite un concepto para tiempo real en la Web, el cual plantea 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) En la presente investigación se asume la definición dada por Alexander Schulze debido a que es el concepto que más se adecúa a la investigación.

Para brindar solución a la comunicación Web en tiempo real surge un nuevo protocolo de comunicación HTML5 WebSocket. Entre las definiciones de WebSocket se encuentra la que expresa que el protocolo WebSocket define los procedimientos para actualizar la conexión a través de HTTP a una conexión mediante WebSocket totalmente bidireccional usando TCP. A través del procedimiento *handshake* el cliente envía una petición HTTP GET para establecer una comunicación WebSocket con el servidor. Posteriormente la comunicación permanece activa hasta que se cierra, permitiendo intercambiar mensajes entre el cliente y el servidor.(HTML 5 WebSockets in Theory and Practice. Conference Kaazing, 2011)

Por otro lado Petter Lubbers define un concepto diferente para WebSocket, el cual plantea que es una tecnología que proporciona un canal de comunicación bidireccional y *full-duplex* sobre un único socket TCP (Transmission Control Protocol, Protocolo de Control de Transmisión). Soluciona las limitaciones del protocolo HTTP, al establecer una comunicación *full-duplex* (TCP) entre el cliente y el servidor, sustituyendo la comunicación *half-duplex*⁷ (HTTP). Se reduce, en grandes proporciones, el tráfico en la red teniendo en cuenta que al establecer la comunicación WebSocket entre el cliente y el servidor solo hay un envío de 2 bytes, eliminando las cabeceras HTTP.(Lubbers, 2010). La presente investigación asume la definición emitida por Petter Lubbers, puesto que este autor define de manera más completa este protocolo de comunicación.

1.2. Estado del Arte

El mundo de la representación de información gráfica en tiempo real en la Web está adquiriendo una gran importancia a lo largo de los últimos años. Hoy día se pueden

⁷ Half-Duplex: Significa que el método o protocolo de envío de información es bidireccional pero no simultáneo.

encontrar en el mercado numerosos desarrolladores que generan este tipo de aplicaciones.

Actualmente existen protocolos de comunicación que prometen dotar sustancialmente a las aplicaciones Web. Uno de estos protocolos, WebSocket, define los procedimientos para actualizar la conexión a través de HTTP a una conexión mediante WebSocket totalmente bidireccional usando TCP, solucionando los problemas que presenta AJAX o Comet. WebSocket establece una comunicación *full-duplex* (TCP) entre el cliente y el servidor, sustituyendo la comunicación *half-duplex* (HTTP). El ancho de banda ahorrado tiene una proporción de 500:1 y una latencia de 3:1, resultados que hacen que grandes empresas estén muy interesadas en este protocolo (WebSocket für alle, 2011).

En los últimos años han surgido numerosas aplicaciones que permiten a los usuarios analizar la información visualmente de manera más efectiva, rápida, precisa y confiable. Este tipo de aplicaciones resultan muy útiles, ayudando en muchos casos a la toma de decisiones y a un mejor manejo de los datos que se están analizando.

Las aplicaciones que grafican datos en tiempo real poseen características distintivas y ventajas muy particulares. Una de ellas es **DataHub QuickTrend**, es un programa en tiempo real para mostrar cualquier dato que se pueda leer desde OPC⁸, DDE⁹, TCP¹⁰ o una fuente de datos encriptados a través de gráficas. Las características de DataHub QuickTrend permiten construir gráficos en tiempo real, simplemente eligiendo los datos que se desean visualizar.

Dichos datos podrían ser perdidos en el caso de que la aplicación del gráfico estuviera comunicándose por ejemplo en modo *polling*. Esta técnica consiste en realizar sondeos al servidor en intervalos regulares de tiempo, con el objetivo de lograr mayor nivel de actualización en el lado del cliente. Esta técnica implica un conjunto de peticiones innecesarias al servidor, aumentando considerablemente el uso del ancho de banda y la latencia de la comunicación en la Web.

⁸ OPC: contiene un acrónimo en un acrónimo. Control de procesos OLE. El acrónimo "OLE" se refiere a la vinculación e incrustación.

⁹ DDE: Intercambio dinámico de datos.

¹⁰ TCP: Protocolo de control de transmisión.

Esta aplicación utiliza la modalidad continua AJAX, tecnología que se introduce en la Web para lograr tiempos reales. Con el uso de AJAX, ante una actualización en el cliente no es necesario actualizar todo el contenido de una página Web, sino que brinda la posibilidad con *XHR*¹¹ y JavaScript en el cliente, actualizar solo los elementos que han sufrido variación. Esto trae consigo mayor rendimiento en las aplicaciones Web. Sin embargo, es una tecnología basada en una solicitud del cliente, por lo que no brinda solución a las principales limitaciones del protocolo HTTP. La tecnología AJAX no supera a las características de latencia y ancho de banda que proporciona WebSocket, ya que el ancho de banda mínimo de AJAX es muy superior al ancho de banda de WebSocket.

Otra de las soluciones existentes en la actualidad es **Lightstreamer**. Esta solución es escalable y fiable para empujar datos a las aplicaciones dinámicas de Internet basado en el paradigma Comet. Este paradigma intenta lograr una comunicación bidireccional manteniendo una conexión permanente y una petición HTTP de larga duración, donde los eventos del lado del servidor pueden ser enviados al navegador y se puede realizar peticiones al servidor en una conexión de reciente apertura. El mantenimiento de estas dos conexiones introduce una sobrecarga significativa en términos de consumo de recursos, que se traduce en una latencia adicional, lo cual afecta el rendimiento del sistema. Los clientes y las aplicaciones pueden recibir datos en tiempo real de Lightstreamer pero a un alto precio y consumo de recursos. Éste se ha utilizado en muchos sistemas de producción, donde las características de escalabilidad, el bajo impacto de la red, la gestión de ancho de banda, y otras características avanzadas, han demostrado ser fundamental.

Las características que posee Lightstreamer no son comparables con las que proporciona WebSocket, lo que logra que se descarte la posibilidad de utilizarla.

Por otro lado, **Adobe Flex Charting** ofrece una abundante biblioteca de gráficos y diagramas interactivos que permiten los tableros de datos sofisticados y el análisis interactivo de datos. Los componentes de los gráficos, proporcionados dinámicamente al cliente y con total compatibilidad con la agrupación de datos y el

¹¹ XHR: XML Http Request significa que es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web.

modelo de objetos de Flex, facilitan la inserción de información detallada, el paso del puntero del ratón por encima de elementos y otras interacciones. Con esto se logra que sus gráficos sean aún más intuitivos que antes. Todos los gráficos se pueden ampliar para ofrecer más funciones, pero, además, los desarrolladores pueden utilizar las clases básicas para crear tipos de gráficos personalizados.

Esta solución está basada en la plataforma Adobe Flash Builder y utiliza el protocolo Real Time Message Protocol (RTMP) para comunicaciones en tiempo real. El RTMP es un método para la comunicación de los flujos de datos que tienen en tiempo real la latencia y requisitos de rendimiento, tales como streaming de vídeo y audio. Éste es un protocolo basado en TCP que mantiene conexiones persistentes. Sin embargo, esta tecnología trae consigo numerosos problemas que hacen que los usuarios comiencen a dudar en cuanto a su utilización. Uno de ellos es la prolongada demora al cargar las páginas, lo que provoca mayor latencia, logrando que los usuarios accedan hacia otros sitios más rápidos. No sucede así con el protocolo WebSocket, que proporciona rapidez en la carga de los datos, mostrando con eficiencia las representaciones gráficas de la información. La velocidad e inmediatez de los datos de este protocolo supera en gran medida al RTMP, superándolo enormemente en cuanto a latencia, ya que la agilidad y velocidad de la información reduce la latencia en gran medida, ventaja que distingue al protocolo WebSocket del resto de los protocolos de comunicación. Por otra parte un gran número de usuarios de Internet no pueden visualizar correctamente una página web que utilice la tecnología flash, aún existen navegadores que no dan soporte a esta tecnología y en diversos casos se ignoran los estándares de accesibilidad, lo que provocaría la salida inmediata del mercado. Razones para no tomar esta opción como una posible solución a la hora de representarla información gráficamente en tiempo real en la Web.

Uno de los principales servidores que soportan WebSocket para el desarrollo de aplicaciones hoy día es la Pasarela WebSocket de Kaazing desarrollada por la empresa Kaazing Corporation¹². Fue diseñada para ser un servidor web de transmisión continua y de *duplex* completo.

¹² <http://kaazing.com/about/contact.html>

Kaazing es una solución empresarial que permite comunicación bidireccional y de alta velocidad en la Web, utilizando el protocolo WebSocket. Entre sus principales ventajas destacan extrema escalabilidad, alta disponibilidad, seguridad y mecanismos de respaldo para que todos los navegadores y dispositivos móviles puedan ser conectados a las aplicaciones. (Kaazing Corporation, 2012) Sin embargo la licencia de este producto es comercial y esto impide que la comunidad de desarrolladores lo adopte rápidamente como tecnología para desarrollar sus aplicaciones.

Actualmente son numerosas las aplicaciones que visualizan datos en tiempo real, sin embargo existen inconvenientes tales como la latencia, el ancho de banda, los recursos del sistema y la licencia de los productos que hacen necesaria una solución más completa. Hoy día ya existe una herramienta que permite representar gráficamente datos en tiempo real, solo que no es identificada por muchos usuarios que desean representar estos datos de forma gráfica, por lo que se hace necesario desarrollar una aplicación web para la representación gráfica de información en tiempo real con jWebSocket que garantice mayor nivel de usabilidad del marco de trabajo en el desarrollo de aplicaciones gráficas en la Web.

1.3. Metodología a utilizar

En una aplicación informática, el proceso de desarrollo debe estar regido y orientado por una metodología de desarrollo de software, que guíe los procesos y permita tener un registro detallado del avance de la investigación. Las metodologías pueden ser robustas o ágiles. Las metodologías robustas o pesadas están concebidas para guiar el proceso de desarrollo de los software de gran envergadura, cuando un proyecto requiere de gran cantidad de documentación, vaya a ser realizado en un tiempo considerablemente largo y existe la posibilidad de que pase por las manos de varios equipos de trabajo. Por su parte, las metodologías ágiles intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en los clientes y los resultados. Se basan en promover iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto, logrando que se minimicen los riesgos desarrollando software en corto tiempo. (Metodologías Tradicionales Vs. Metodologías Ágiles, 2011)

Proceso Unificado de Desarrollo

Proceso Unificado de Desarrollo (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso formal que provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales, respetando cronograma y presupuesto. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura, iterativo e incremental y utiliza UML como lenguaje de notación.

Consta de 4 fases principales:

- **Inicio:** el objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** en esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** en esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- **Transición:** el objetivo es llegar a obtener el despliegue del proyecto.

(Figueroa, y otros, 2011)

SCRUM

Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza de forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nuevas funcionalidades. SCRUM se enfoca en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. (Schwaber, y otros, 2011)

SCRUM permite la creación de equipos auto organizado impulsando la localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de SCRUM es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser

fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, SCRUM adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Programación Extrema

Programación Extrema (Extreme Programming en inglés, habitualmente resumido como XP) es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Esta consiste en una programación rápida o extrema y es utilizada para proyectos de corto plazo, con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. (Wells, 2009)

Las características fundamentales que presenta esta metodología son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y discutido mientras se escribe, es más importante que la posible pérdida de productividad inmediata.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir una nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.

- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

SXP

SXP está compuesta por las metodologías SCRUM y XP, ofreciendo una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva. Esta metodología fomenta el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

SXP consta de 4 fases principales:

- **Planificación-Definición:** se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** se realiza la implementación del sistema hasta que esté listo para ser entregado.
- **Entrega:** es la puesta en marcha.
- **Mantenimiento:** se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, Diseño, Implementación, Pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añada una nueva funcionalidad.

Fundamentos de la selección

Para la presente investigación se hace uso de la metodología SXP porque cumple con las características que se requieren para el tipo de proyecto que se está desarrollando. Está especialmente indicada para el trabajo con pequeños equipos de desarrollo, donde exista un constante cambio de requerimientos o requisitos imprecisos, donde exista un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad, como es el caso del proyecto que responde a esta investigación. Fomenta el trabajo en equipo, con un objetivo claro, permitiendo el seguimiento y control de las tareas a realizar. (SXP, Metodología Ágil para el Desarrollo de Software, 2010)

1.4. Lenguajes de programación y modelado

1.4.1. Lenguaje Modelado Unificado

Teniendo en cuenta que fue seleccionada la metodología de desarrollo de software SXP para el desarrollo de la presente investigación y que además define a UML como lenguaje de modelado, este es asumido para modelar y diseñar la solución propuesta.

Lenguaje Modelado Unificado (UML, por sus siglas en inglés, Unified Modeling Language). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. (Rumbaugh, y otros, 2007)

Se decide utilizar UML en su versión 2.0 para el modelado de esta investigación, debido a que es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

1.4.2. Java

Para desarrollar la aplicación se utiliza Java como lenguaje de programación en el lado del servidor, este es utilizado en el marco de trabajo jWebSocket sobre el cual se desarrolla la solución.

Java es un lenguaje moderno, de alto nivel, que recoge los elementos de

programación que típicamente se encuentran en todos los lenguajes de programación, permitiendo la realización de programas profesionales. La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. Con plataforma se refiere a la máquina virtual de Java (*Java Virtual Machine*). Actualmente este lenguaje es el más utilizado para el desarrollo de aplicaciones por las ventajas que brindan sus características.(Gosling, y otros, 2005)

Posee grandes ventajas, interpretada y compilada a la vez: genera ficheros de clases compiladas, pero estas clases compiladas, son en realidad interpretadas por la máquina virtual de java. Siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando. Es multiplataforma: el mismo código java que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java. Esta es una de las principales características que favorece el crecimiento y difusión del lenguaje. Es seguro: la máquina virtual, al ejecutar el código java, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros. Es distribuido: Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets, establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas. Es orientado a objetos: Java fue diseñado como un lenguaje orientado a objetos desde su creación. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos que manipulan esos datos. Esto le permite crear programas modulares y código reutilizable. Es robusto: exhaustivo de la fiabilidad. Java pone mucho énfasis en el control temprano de posibles errores, como compiladores de Java son capaces de detectar muchos problemas que en primer lugar aparecen durante el tiempo de ejecución en otros idiomas. Es multiproceso: hoy día ya se ven terriblemente limitadas las aplicaciones que sólo pueden ejecutar una acción a la vez. Java soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

1.4.3. JavaScript

Para desarrollar la aplicación se utiliza JavaScript como lenguaje de programación en el lado del cliente, este es utilizado en el marco de trabajo jWebSocket sobre el cual se desarrolla la solución.

JavaScript es un lenguaje utilizado para realizar acciones dentro del ámbito de una página web. 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. Existen un conjunto de librerías JavaScript que facilitan la implementación de aplicaciones con las más diversas funcionalidades. Entre las características admirables de este lenguaje se destaca su compatibilidad con la mayoría de los navegadores existentes.(Goodman, 2001)

1.5. Otros lenguajes

1.5.1. HTML

Para el desarrollo de la aplicación para representar datos gráficamente con el marco de trabajo jWebSocket se hace uso del lenguaje de marcado HTML (HyperText Markup Language). Este es el lenguaje con el que se crean la mayoría de páginas web.

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)

1.5.2. Hojas de Estilo en Cascada (CSS)

CSS 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. Una vez creados los contenidos de una página web, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos,

posición de cada elemento dentro de la página, entre otros.(Eguíluz Pérez, 2009)

En la presente investigación se hace uso de CSS para dar estilos a las páginas de la aplicación.

1.6. Tecnologías usadas para el desarrollo de la solución

Marcos de trabajo

Para el desarrollo de la solución propuesta se hace uso del marco de trabajo jWebSocket.

Un Marco de Trabajo (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 se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta, puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Los marcos de trabajo son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel para proveer un sistema funcional.

El marco de trabajo jWebSocket forma parte del campo de acción en el cual se basa la presente investigación. Por lo tanto es considerado una tecnología imprescindible para el desarrollo de la aplicación que se propone.

1.6.1. Marco de Trabajo del lado del Servidor

Marco de Trabajo jWebSocket

jWebSocket es un marco de trabajo de código abierto para el desarrollo de aplicaciones web estacionarias y móviles basado en Java en el lado del servidor y en JavaScript del lado del cliente. jWebSocket establece un modelo de token. Los tokens son datos abstractos que a través de una estructura jerárquica y una API proporcionan métodos de acceso a los contenidos. Con el objetivo de realizar una abstracción en la manipulación de los diferentes formatos, el marco de trabajo convierte los paquetes de datos entrantes y salientes en tokens. 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 `jQueryWebSocket` tiene una arquitectura de plug-in que permite aumentar con facilidad sus funcionalidades.

El servidor `jQueryWebSocket` está diseñado para funcionar como servidor de comunicaciones o como servidor web, brindando total flexibilidad. `jQueryWebSocket` como servidor web proporciona un conjunto importante de funcionalidades y su arquitectura extensible mediante plug-in permite añadir fácilmente características adicionales a un sistema independiente. Por otra parte los administradores pueden configurar el servidor exactamente como sea necesario y dejar a un lado todos los módulos que no necesiten. 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)

En la actualidad existen tecnologías que brindan la posibilidad de graficar en tiempo real utilizando el protocolo de comunicación `WebSocket`, una de ellas es precisamente el marco de trabajo `jQueryWebSocket`, que proporciona un mayor nivel de robustez y confiabilidad a la hora de representar los datos en la Web. `jQueryWebSocket` es una nueva tecnología orientada al desarrollo de aplicaciones basadas en `WebSocket` que proporcionan altos niveles de velocidad, escalabilidad y seguridad (Framework Approach for WebSockets, 2011). Permite crear innovadoras aplicaciones HTML5, ofrece además una amplia gama de funcionalidades desde el intercambio de bajo nivel basado en tokens, hasta la sincronización de interfaz gráfica de usuario, llamadas a procedimiento remoto y mucho más. (WebSocket für alle, 2011)

`jQueryWebSocket` es un proyecto del cual 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 `jQueryWebSocket`, precisamente por esta razón la aplicación a desarrollar se lleva a cabo sobre este marco de trabajo.

1.6.2. Marcos de Trabajo del lado del Cliente

jQuery

jQuery es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. (Murphey, 2010)

Moo Tools

Es un marco de trabajo de JavaScript que se caracteriza por ser liviano, compacto, modular y orientado a objetos. Está diseñado para desarrolladores de JavaScript de nivel intermedio y avanzado. Le permite escribir código potente, flexible y multi-navegador con su API elegante, bien documentada y coherente. (Proietti, 2009)

Marco de Trabajo Ext JS

Es un marco de trabajo para construir Aplicaciones de Internet Enriquecidas (RIA por sus siglas en inglés). Está basado en librerías JavaScript de código abierto, ligera y de alto rendimiento, compatible con la mayoría de los navegadores que permiten crear páginas e interfaces webs dinámicas. (Extjs, 2009)

Ext JS es neutral al lenguaje que se use en el servidor. Siempre que el resultado se envíe a la página en el formato adecuado, Ext JS no se preocupará de lo que pase en el servidor.

Ext JS 4

Es un gran avance de los marcos JavaScript. Con mayor funcionalidad, plug-in gratuito para gráficos, y una nueva arquitectura MVC. Permite crear aplicaciones web increíbles para todos los navegadores. Trae un paquete de datos completamente nuevo que permite a los desarrolladores utilizar una arquitectura

modelo-vista-controlador en la construcción de su aplicación. Al permitir la separación de la gestión de datos, la lógica y elementos de la interfaz, Ext JS 4 hace que sea fácil, incluso para los grandes equipos de desarrollo, trabajar de manera independiente sin preocupaciones.

Un paquete de datos completamente nuevo anima a los desarrolladores a aprovechar funciones como el desplazamiento de cuadrícula infinita para construir un nuevo nivel de interactividad en las aplicaciones web.

Ext JS 4 permite a los desarrolladores entregar sobre una increíble variedad de navegadores y sistemas operativos con el mismo código. En los navegadores modernos utiliza las características de HTML5. Hace que sea fácil crear una aplicación poderosa en la web, independientemente del navegador que el cliente usa.

Ext JS 4 proporciona los gráficos más avanzados y capacidades gráficas de cualquier marco de JavaScript, sin depender de plug-ins, ofreciendo una visualización perfecta de píxeles en cualquier navegador sobre cualquier sistema operativo. Aprovechando SVG y VML, Ext JS 4 con su paquete de gráficos, permite a los desarrolladores diseñar y programar sus gráficos en cualquier navegador. (Sencha, 2011)

Fundamentos de la selección

A pesar de que existen diversos marcos de trabajo de JavaScript para el desarrollo de este tipo de aplicación como los mencionados anteriormente, se utiliza el marco de trabajo Ext JS 4 por las múltiples ventajas y funcionalidades que proporciona, el diseño está completamente separado de la funcionalidad, existe una buena y amplia documentación, es sencillo el trabajo con gráficos y la vinculación con el marco de trabajo jWebSocket, facilitando en gran medida el desarrollo de la aplicación.

1.7. Herramientas a emplear para el desarrollo de la solución

1.7.1. Herramienta CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) 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, cada uno de ellos con diferentes aplicaciones y especificaciones. (Herramientas Case, 2011)

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.(Herramientas Case, 2011)

Visual Paradigm

Visual Paradigm es una poderosa herramienta CASE que hace uso del lenguaje modelado unificado (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.

Fundamentos de la selección

Anteriormente se analizaron algunas de las herramientas CASE existentes en el mundo para el modelado de software, y teniendo en cuenta que se desea desarrollar un trabajo bajo las políticas de software libre, se selecciona como herramienta CASE a Visual Paradigm. Esta herramienta a pesar de no ser libre, cuenta con una licencia comercial la cual posee la Universidad de las Ciencias Informáticas (UCI).

Soporta UML, lo cual permite modelar los procesos de negocios y generar el mapeo de objetos-relacionales para el lenguaje de programación Java, lenguaje

utilizado del lado del servidor por el marco de trabajo `jWebSocket` seleccionado para la presente investigación. Permite la integración con herramientas Java como NetBeans IDE. Permite la generación de código y la ingeniería inversa para un conjunto de lenguajes entre los que se encuentra Java. (Visual Paradigm International Ltd, 2011)

1.7.2. Herramientas de Control de Versiones

Para el desarrollo de un proyecto de software de esta envergadura se hace indispensable la utilización de una herramienta para el control de versiones debido a las necesidades de controlar los cambios realizados al código fuente.

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. Es importante decir que estos sistemas no solo se limitan a gestionar archivos de texto sino que también gestionan documentos, imágenes y ficheros de todo tipo. Los sistemas de control de versiones utilizados a nivel internacional con gran aceptación y popularidad guardan toda la información en un repositorio central accesible a través de la red, permitiendo el trabajo colaborativo entre varios puestos de trabajo y a su vez proporcionando una mayor seguridad y disponibilidad de los datos.

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.

Subversion

Subversion es un sistema de control de versiones completamente equipado que fue originalmente diseñado para reemplazar a CVS. Desde entonces se ha expandido

más allá de su objetivo original, pero su modelo básico, el diseño y la interfaz fueron fuertemente influenciados por CVS por lo que debido a estas particularidades los usuarios de CVS se sienten muy cómodos al interactuar con Subversion. (Apache Software Foundation, 2011)

Subversión puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. (Ben Collins-Sussman, 2004)

Fundamentos de la selección

A pesar de que existen otras herramientas de control de versiones, se selecciona Subversion como herramienta de control de versiones para ser utilizado en el desarrollo de la aplicación debido a que este sistema presenta varias características importantes. Los directorios son versionados. La resolución de conflictos es de forma interactiva. Gestiona de manera eficaz los archivos binarios. Bloquea los archivos. Vincula los lenguajes de programación. Vincula varios repositorios. Éste 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.

1.7.3. 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.

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)

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)

Fundamentos de la selección

Actualmente existen herramientas de control de versiones del lado del cliente, a pesar de todas las ventajas y funcionalidades que ofrece Tortoise, se decide usar Raptidsvn ya que ésta es libre, y proporciona una interfaz fácil de usar, es simple para los principiantes tanto para 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. Puede ser ejecutado en cualquier plataforma.

1.8. Entorno integrado de desarrollo IDE

Un IDE (acrónimo en inglés de Integrated Development Environment) 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 IDEs cuentan con herramientas tales como: editor de código, herramientas para el rastreo de código, compilador, depurador y constructor de interfaz gráfica. A continuación se hace una caracterización de varios IDEs que permitirá adquirir los elementos necesarios para determinar cuál es el más idóneo para el desarrollo de solución propuesta.

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, una organización independiente sin ánimo de

lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. Eclipse basa su funcionalidad en módulos (en inglés 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).

NetBeans

NetBeans IDE es una herramienta desarrollada por Sun Microsystems. Este entorno integrado de desarrollo es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Su misión consiste en evitar tareas repetitivas, facilitar la escritura correcta de código, disminuir el tiempo de depuración e incrementar la productividad del desarrollador. Cuenta con un depurador, perfilador de integración, herramientas para refactorizaciones, completamiento de código y control de versiones de archivos.(NetBeans, 2011)

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 entre ellos Java.

Fundamentos de la selección

A pesar de que existen otros IDE capaces de desarrollar aplicaciones Java como es el caso de Eclipse SDK, para el desarrollo de la solución propuesta se selecciona como entorno integrado de desarrollo a NetBeans IDE, ya que es Java el lenguaje a utilizar, además de ser un producto libre y gratuito sin restricciones de uso. 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 con las características de JDK7.

1.9. Sistema Gestor de Base de Datos

Los sistemas de gestión de bases de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Su propósito es manejar de forma clara, sencilla y ordenada un

conjunto de datos que posteriormente se convertirán en información relevante para una organización.

Las ventajas que proporcionan los gestores de base de datos son:

- Proveen facilidades para la manipulación de grandes volúmenes de datos.
- Simplifican la programación de equipos de consistencia.
- Manejando las políticas de respaldo adecuadas, garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores correctamente, etc.
- Organizan los datos con un impacto mínimo en el código de los programas.
- Disminuyen drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Existen bases de datos relacionales y documentales (o no relacionales). El auge de la Web está mirando las bases de datos documentales como clave en sus negocios, por su mayor capacidad de escalabilidad, rendimiento y alta disponibilidad que los actuales sistemas de bases de datos relacionales.

Características de las bases de datos documentales:

- A diferencia de las bases de datos relacionales, están libres de esquemas y de estructuras. Por tanto, la información no se divide en columnas fijas y filas, sino en documentos que pueden tener libertad de información, como cualquier cantidad de campos o de tipos de campos, e incluso de contener en el propio campo una lista de valores, un documento o una colección de documentos.
- La información almacenada en un documento está formada por pares de clave/valor.
- En cuanto a estructura de datos, son libre de esquemas. Al no estar sujeta a estructuras ni relaciones, los cambios de diseño son fáciles de adoptar y con un impacto mínimo.
- Son muy sencillas y potencian la escalabilidad.

- Son muy eficientes a la hora de explotar grandes volúmenes de información textual.
- Ganan rapidez dedicándose más en exclusiva al almacenamiento y la recuperación de la información, no consumen tiempos en la integridad o en la revisión de las reglas.
- La gran ventaja de no poseer esquema, es lo que las hace muy eficientes al trabajar con grandes cantidades de documentos.
- Los campos vacíos no se añaden al documento, optimizando el espacio de almacenamiento.
- Brindan la posibilidad de embeber documentos y colecciones dentro de documentos.
- Poseen un lenguaje de consulta natural.

Mongo DB

Mongo es el sistema de base de datos desarrollada en 10gen por Geir Magnusson y Dwight Merriman. Mongo es una base de datos orientada a documentos JSON, salvo que está diseñada para ser una verdadera base de datos de objetos, más que para un almacenamiento de clave/valor puro. Originalmente, 10gen enfocó poner juntos una pila completa de servicios web, aunque sin embargo, más recientemente ha tenido que ser reenfocado principalmente en la base de datos Mongo.

MongoDB es escalable, de alto desempeño, de código abierto, y el almacenamiento de datos está orientado a documentos, con un estilo JSON proporcionando esquemas dinámicos que ofrecen simplicidad y poder. Soporta índices sobre cualquier atributo, tal y como se acostumbra a ver. Rápidas actualizaciones en el contexto.

Fundamentos de la Selección

Existen numerosas bases de datos como son PostgreSQL, SimpleDB, CouchDB, Cassandra, entre otras. Sin embargo se hace uso de la base de datos documental MongoDB por las ventajas que proporciona, ya que responden a las necesidades de la presente investigación, la velocidad de respuesta con que cuenta esta base de datos, facilita que se logre representar datos en la Web en tiempo real. Supera

en gran medida a las restantes bases de datos ya que es el enlace perfecto entre el almacenamiento clave/valor (que son rápidos y altamente veloces) y los sistemas tradicionales de RDBMS que proporcionan consultas ricas y una profunda funcionalidad. (10gen, Inc., 2011)

Conclusiones del capítulo

En el presente capítulo, luego de realizar una profunda investigación, se expusieron las definiciones más significativas que sustentan la presente investigación. Se analizaron las soluciones existentes con características similares. Además se definió la metodología SXP que es la que se va a poner en práctica, así como los lenguajes de programación Java del lado del servidor y JavaScript del lado del cliente y como gestor de base de datos MongoDB. Se seleccionaron las herramientas tales como NetBeans, Visual Paradigm y Subversion así como las tecnologías jWebSocket y Ext JS 4 para el desarrollo de la solución.

CAPÍTULO 2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En el presente capítulo se describe las funcionalidades del sistema, resaltando sus características. La metodología de desarrollo SXP posee artefactos, los cuales son brevemente detallados y explicados para lograr un mejor entendimiento de los requisitos con los que debe cumplir la aplicación. Se define el modelo de negocio para lograr entender mejor su contexto y se describen los Requisitos Funcionales y No Funcionales del sistema, las historias de usuario y las tareas de ingeniería asociadas a las mismas.

2.1 Propuesta de Solución

Hoy día ya se logra realizar en tiempo real la representación de información gráfica en la Web, lo que a un elevado costo en cuanto a recursos del sistema, como se plantea en el estudio del estado del arte realizado en la presente investigación. Sin embargo, el marco de trabajo `jWebSocket` es capaz de lograr la representación gráfica solucionando las limitaciones que presenta el protocolo HTTP. Para ello se desarrolla la aplicación web `Charting Demo` que representa gráficamente la información en tiempo real usando `jWebSocket`, garantizando un mayor nivel de usabilidad del marco de trabajo en el desarrollo de aplicaciones gráficas en la Web. La aplicación permite a los usuarios ver el comportamiento de los recursos de la computadora donde se está consultando la aplicación, del servidor de `jWebSocket`, muestra el número de usuarios que accedan a la aplicación en un instante de tiempo así como la información estadística del uso de cada plug-in de `jWebSocket`. El comportamiento de los recursos de la computadora se refleja a partir del CPU, de la memoria, del espacio utilizado y libre del disco duro, y del swap mediante una gráfica de barras. Se logra ver el comportamiento de los usuarios que acceden a la aplicación a través de una gráfica de líneas mostrando la cantidad de usuarios conectados en un minuto, así como el comportamiento de los números de peticiones en el servidor a intervalos configurables. Se visualiza la cantidad de peticiones al servidor cada hora del día, cada día del mes y cada mes del año. Además analiza el uso de los plug-ins que estén siendo utilizados.

La aplicación presenta interfaces sencillas. Son muy fáciles de utilizar y acordes

con los patrones de diseño establecidos para el proyecto.

El impacto que tiene la aplicación Charting Demo consiste en mostrar la capacidad del marco de trabajo jWebSocket en el desarrollo de aplicaciones que grafiquen información real con altos niveles de variación en el tiempo. Los desarrolladores identifican con claridad las potencialidades y las bondades que ofrece este marco de trabajo para la representación gráfica. Esto trae como consecuencia que empresas y desarrolladores que necesitan graficar datos en tiempo real, reconozcan al marco jWebSocket como una vía eficiente para desarrollar aplicaciones con estas funcionalidades para mayor satisfacción de sus clientes. El conocimiento de las potencialidades de jWebSocket para la representación gráfica en tiempo real trae consigo además mayor usabilidad del marco de trabajo, lo cual logra un aumento de cuotas de mercado para jWebSocket y eleva significativamente el número de usuarios y clientes potenciales a nivel internacional.

2.2 Planificación del Proyecto por Roles

Para lograr una mayor organización y eficiencia en el desarrollo de la solución propuesta se hace necesario la definición de los diferentes roles que intervienen en el proceso de desarrollo de software. De esta forma se le asigna a cada integrante del proyecto una responsabilidad con el objetivo de coordinar e integrar sus esfuerzos para lograr un objetivo común. En el Anexo 1 se muestra la asignación de roles pertenecientes al proyecto, así como las principales responsabilidades de cada uno de estos.

2.3 Modelo de Historias de Usuario del Negocio

Dentro de los artefactos que genera la metodología SXP se encuentra la plantilla del Modelo Historias de Usuario del Negocio, donde se definen las características específicas del negocio en cuestión.

Este modelo describe los pilares fundamentales del proceso de representación gráfica en la Web. Representar los datos gráficamente en la Web posibilita que los usuarios analicen la información, y seleccionen los datos que desean representar así como el tipo de gráfica que será utilizada para mostrar esta información. El usuario es el que actúa sobre el proceso para el cual está dirigido la representación

gráfica en la Web. Este usuario es quien actúa sobre las historias de usuario del negocio Analizar información que no es más que llegar a conclusiones a través del proceso de representación gráfica de la información, la historia de usuario Seleccionar los datos a graficar consiste en escoger la información que se desea reflejar mediante gráficos y Seleccionar tipo de gráficas no es más que elegir el tipo de gráfica con la cual va a ser representada la información seleccionada, que son los elementos que se realizan durante este proceso.

A continuación se muestra en la Tabla 1 los actores y trabajadores del negocio:

Tabla 1: Actor y Trabajador del Negocio

Actor	Descripción
Usuario	Es el que actúa sobre el proceso para el cual está dirigido la representación gráfica en la Web.
Trabajador	Descripción
Sistema	Aplicación Charting Demo la cual posibilita la representación gráfica de la información.

Seguidamente se muestra en Fig. 1 el diagrama de historia de usuario del negocio correspondiente:

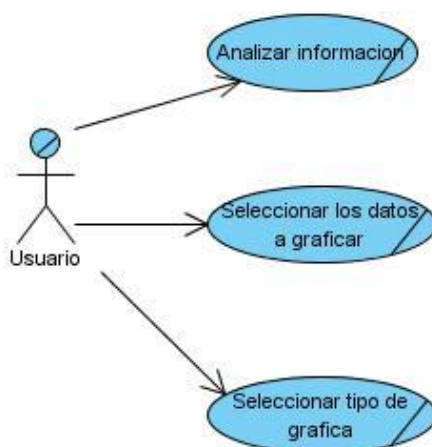


Fig. 1 Modelo de Historia de Usuario del Negocio. Fuente: Elaboración propia.

2.4 Lista de Reserva del Producto (LRP)

La lista de Reserva del Producto es una lista con prioridad en la cual queda reflejada todas las tareas a desarrollar en el proyecto durante el tiempo de desarrollo.

La presente investigación cuenta en su totalidad con 10 requisitos funcionales y 24 requisitos no funcionales como se muestra en la Tabla 2, los cuales se detallan a continuación:

Tabla 2: Lista de Reserva del Producto

Prioridad	Ítem *	Descripción	Estimación	Estimado por
Muy Alta				
	1	Graficar uso de CPU.	3 días	Analista
	2	Graficar uso de memoria.	3 días	Analista
	3	Graficar uso de swap.	3 días	Analista
	4	Graficar espacio libre de disco duro.	3 días	Analista
	5	Graficar espacio en uso de disco duro.	3 días	Analista
Alta				
	6	Graficar cantidad de usuarios conectados.	15 días	Analista
	7	Graficar peticiones por hora.	5 días	Analista
	8	Graficar peticiones por días.	5 días	Analista
	9	Graficar peticiones por meses.	5 días	Analista
	10	Graficar plug-ins en	15 días	Analista

		USO.		
Media				
Baja				
RNF (Requisitos No Funcionales)				
	11	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.		
	12	El Sistema es una aplicación web.		
	13	El sistema se mantendrá disponible 24 horas diarias durante los 7 días de la semana.		
	14	El mantenimiento de la aplicación se realizará en tiempo de ejecución sin afectar la disponibilidad ni el rendimiento de los servicios.		
	15	Garantizar la integridad y consistencia de los datos.		
	16	Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.		
	17	Se debe realizar la aplicación de forma versionable que permita darle mantenimientos al		

		sistema a fin de aumentar las funcionalidades y/o corregir los errores del mismo a través de versiones posteriores.		
	18	Establecer pautas para la codificación y el diseño de la BD que permita mantener un código uniforme a la hora de realizar modificaciones en el mismo.		
	19	Se documentará la aplicación con diferentes manuales con el objetivo de explicar el uso de la aplicación para garantizar el soporte de la misma.		
	20	En el lado del servidor el lenguaje de programación a utilizar es Java, empleando el Framework jWebSocket y el IDE NetBeans 7.0.1.		
	21	En el lado del cliente se hará uso de los lenguajes JavaScript y HTML, empleando el Framework Sencha y el IDE NetBeans 7.0.1.		
	22	La metodología de desarrollo a seguir es SXP y para la modelación se utilizará la		

		herramienta Visual Paradigm 3.4.		
	23	El gestor de BD a utilizar es MongoDB v2.0.		
	24	Todos los textos y mensajes en pantalla aparecerán en idioma inglés.		
	25	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.		
	26	Lograr un diseño amigable y de fácil uso.		
	27	Todas las interfaces de usuario que se definan para el sistema respetarán los patrones de diseño establecidos para la organización.		
	28	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.		
	29	Los requisitos mínimos de hardware para el servidor de Base de Datos son: 2GB		

		de RAM.		
	30	El cliente debe contar con un navegador que soporte el protocolo WebSocket.		
	31	El servidor de aplicaciones debe tener instalado la máquina virtual de Java OpenJDK 7 y el servidor de jWebSocket.		
	32	El servidor de base de datos requiere un sistema operativo de 64 bits.		
	33	El código de la aplicación será liberado bajo la Licencia Pública General Reducida de GNU (LGPL)		
	34	El código y la documentación de la aplicación deberán cumplir estrictamente las normas de calidad estipuladas por el grupo de Calisoft de la UCI.		

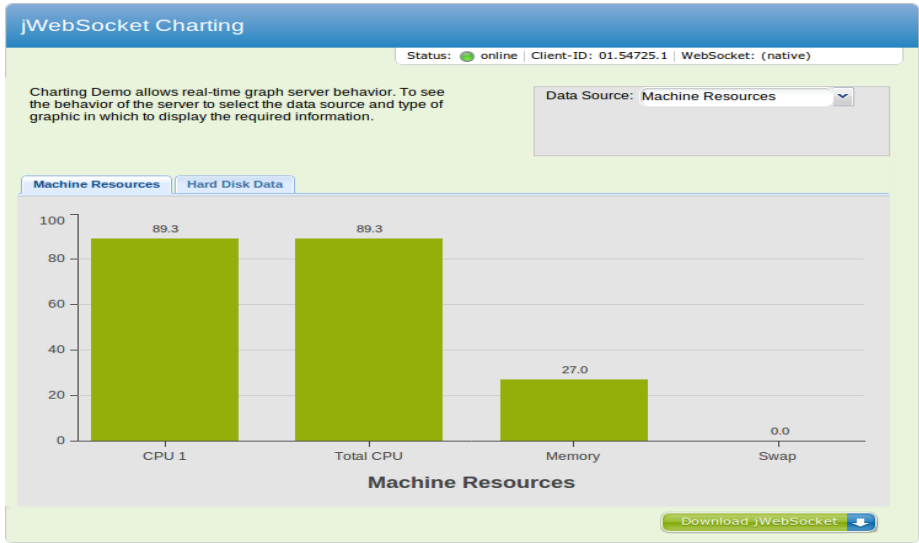
2.5 Historias de Usuario y Tareas de Ingeniería

Las historias de usuario en la metodología de desarrollo SXP son las que describen las tareas que el sistema debe hacer, cuestión que depende en gran medida de las especificaciones realizadas por el cliente.

Las historias de usuarios engloban los requisitos funcionales con los que debe cumplir la aplicación. Para un total de 10 requisitos funcionales se agruparon 4 historias de usuario para dar cumplimiento a las funcionalidades. A continuación se

muestran cada una de las historias de usuarios con sus respectivas tareas asociadas:

Tabla 3: Historia de Usuario Graficar Recursos de la máquina

Historia de Usuario	
Número:HU_1	Nombre Historia de Usuario: Graficar recursos de la máquina.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Merly López Barroso	Iteración Asignada:2
Prioridad en Negocio: Muy Alta	Puntos Estimados:4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 2
<p>Descripción:</p> <p>La presente historia de usuario tiene como objetivo permitir que el usuario pueda ver el comportamiento de la máquina en tiempo real. Se podrá observar como varían los recursos de la PC: uso del CPU, uso de la memoria, uso de la carga de red, espacio utilizado y libre del disco duro, y uso del swap.</p>	
Observaciones: Ninguna	
<p>Prototipo de interfase:</p>  <p>The screenshot shows a web application titled 'jWebSocket Charting'. At the top, there is a status bar indicating 'Status: online', 'Client-ID: 01.54725.1', and 'WebSocket: (native)'. Below this, a text box explains that the 'Charting Demo allows real-time graph server behavior'. To the right, there is a 'Data Source' dropdown menu set to 'Machine Resources'. The main area features a bar chart titled 'Machine Resources' with two tabs: 'Machine Resources' (selected) and 'Hard Disk Data'. The chart displays four bars: 'CPU 1' with a value of 89.3, 'Total CPU' with a value of 89.3, 'Memory' with a value of 27.0, and 'Swap' with a value of 0.0. The y-axis ranges from 0 to 100. At the bottom right, there is a 'Download jWebSocket' button with a download icon.</p>	

A continuación se muestran en las Tabla 4, Tabla 5, Tabla 6, Tabla 7 y Tabla 8 respectivamente la descripción de las tareas de ingeniería que se realizan para lograr el cumplimiento de la historia de usuario anterior:

Tabla 4: Tarea de Ingeniería Graficar uso de CPU

Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario: HU_1
Nombre Tarea: Graficar uso de CPU.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 5 días
Fecha Inicio: 12/09/2011	Fecha Fin: 16/09/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar uso de CPU. Este método muestra el comportamiento del CPU en % a través de una gráfica de barras. El servidor se sirve de la librería Sgar, la cual le proporciona el uso del mismo. El servidor envía al cliente el token con la información del CPU que se utiliza.</p>	

Tabla 5: Tarea de Ingeniería Graficar uso de memoria

Tarea de Ingeniería	
Número Tarea: 1.2	Número Historia de Usuario: HU_1
Nombre Tarea: Graficar uso de memoria.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 4 días
Fecha Inicio: 17/09/2011	Fecha Fin: 20/09/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar uso de memoria. Este método muestra el comportamiento de la memoria en % a través de una gráfica de barras. El</p>	

servidor se sirve de la librería Sgar, la cual le proporciona el uso de la misma. El servidor envía al cliente el token con la información de la memoria que se utiliza.

Tabla 6: Tarea de Ingeniería Graficar uso de swap

Tarea de Ingeniería	
Número Tarea: 1.3	Número Historia de Usuario: HU_1
Nombre Tarea: Graficar uso de swap.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 5 días
Fecha Inicio: 21/09/2011	Fecha Fin: 25/09/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar uso de swap. Este método muestra el comportamiento del swap en % a través de una gráfica de barras. El servidor se sirve de la librería Sgar, la cual le proporciona el uso del mismo. El servidor envía al cliente el token con la información del swap que se utiliza.</p>	

Tabla 7: Tarea de Ingeniería Graficar espacio libre de disco duro

Tarea de Ingeniería	
Número Tarea: 1.4	Número Historia de Usuario: HU_1
Nombre Tarea: Graficar espacio libre de disco duro.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 5 días
Fecha Inicio: 26/09/2011	Fecha Fin: 30/09/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar espacio libre de disco duro. Este método</p>	

muestra el espacio libre de disco duro de la máquina en % a través de una gráfica de pastel. El servidor se sirve de la librería Sgar, la cual le proporciona el uso del mismo. El servidor envía al cliente el token con la información del espacio libre que posee la máquina.

Tabla 8: Tarea de Ingeniería Graficar espacio en uso de disco duro

Tarea de Ingeniería	
Número Tarea: 1.5	Número Historia de Usuario: HU_1
Nombre Tarea: Graficar espacio en uso de disco duro.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 5 días
Fecha Inicio: 1/10/2011	Fecha Fin: 4/10/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar espacio en uso de disco duro. Este método muestra el espacio en uso de disco duro de la máquina en % a través de una gráfica de pastel. El servidor se sirve de la librería Sgar, la cual le proporciona el uso del mismo. El servidor envía al cliente el token con la información del espacio usado de la máquina.</p>	

Tabla 9: Historia de Usuario Graficar Cantidad de Usuarios Conectados

Historia de Usuario	
Número:HU_2	Nombre Historia de Usuario: Graficar cantidad de usuarios conectados.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Merly López Barroso	Iteración Asignada:2
Prioridad en Negocio: Alta	Puntos Estimados:4 semanas

Riesgo en Desarrollo: Alto	Puntos Reales: 3
<p>Descripción:</p> <p>La presente historia de usuario tiene como objetivo permitir que el usuario pueda ver el comportamiento de los usuarios en línea. Se podrá observar la cantidad de usuarios conectados.</p>	
Observaciones: Ninguna	
<p>Prototipo de interfase:</p>  <p>The screenshot shows a web application titled 'jWebSocket Charting'. At the top, it displays 'Status: online', 'Client-ID: 01.37710.3', and 'WebSocket: (native)'. Below this is a text box explaining the demo's purpose and a dropdown menu for 'Data Source' set to 'Online users'. The main area features a line graph titled 'Online Users' with 'Users Connected' on the y-axis (0-10) and 'Seconds' on the x-axis (0-56). The graph shows a fluctuating line starting at 6, rising to 7 at 12 seconds, 8 at 20 seconds, and peaking at 9 at 56 seconds. A tooltip at the end of the graph reads '56 Seconds 9 Connected'. A 'Download jWebSocket' button is at the bottom right.</p>	

A continuación se muestra en la Tabla 10 la descripción de la tarea de ingeniería que se realiza para lograr el cumplimiento de la historia de usuario anterior:

Tabla 10: Tarea de Ingeniería Graficar cantidad de usuarios conectados

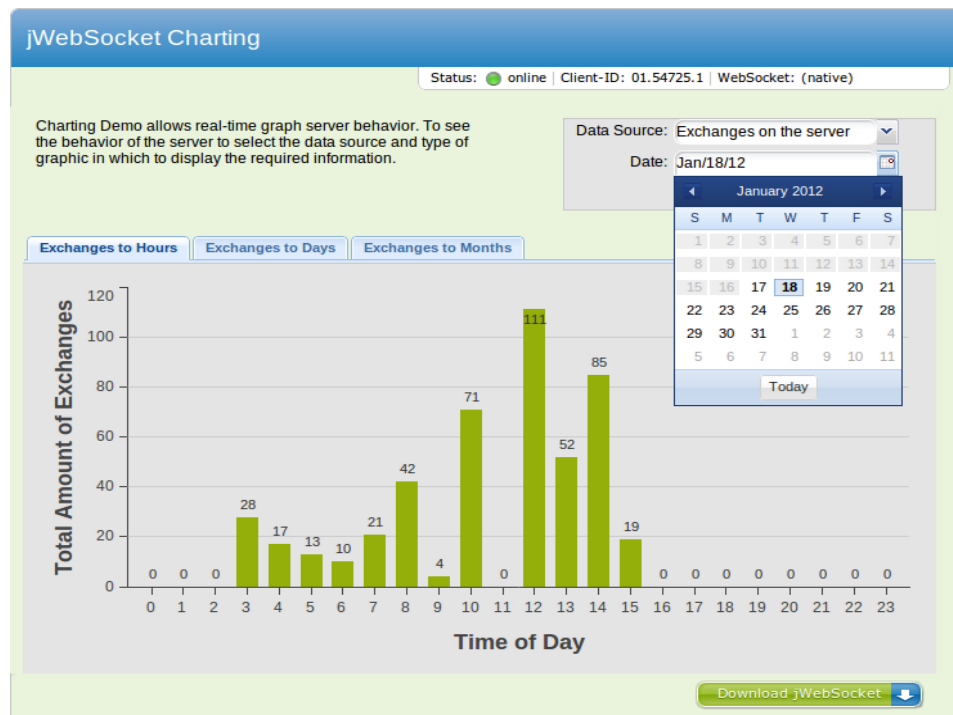
Tarea de Ingeniería	
Número Tarea: 2.1	Número Historia de Usuario: HU_2
Nombre Tarea: Graficar cantidad de usuarios conectados.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 24 días

Fecha Inicio: 5/10/2011	Fecha Fin: 28/10/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar la cantidad de usuarios conectados mostrando el comportamiento de los usuarios en línea. Este método muestra cantidad de usuarios conectados a través de una gráfica de barras. El servidor envía al cliente el token con la información de los usuarios en línea.</p>	

Tabla 11: Historia de Usuario Graficar Intervalo de uso

Historia de Usuario	
Número:HU_3	Nombre Historia de Usuario: Graficar intervalo de uso.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Merly López Barroso	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 2
<p>Descripción:</p> <p>La presente historia de usuario tiene como objetivo permitir que el usuario pueda ver el comportamiento de los números de peticiones en el servidor a intervalos configurables. Se podrá observar la cantidad de peticiones cada hora del día, cada día del mes y cada mes del año.</p>	
Observaciones: Ninguna	

Prototipo de interfase:



A continuación se muestran en las Tabla 12, Tabla 13 y Tabla 14 respectivamente la descripción de las tareas de ingeniería que se realizan para lograr el cumplimiento de la historia de usuario anterior:

Tabla 12: Tarea de Ingeniería Graficar peticiones por hora

Tarea de Ingeniería	
Número Tarea: 3.1	Número Historia de Usuario: HU_3
Nombre Tarea: Graficar peticiones por hora.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 8 días
Fecha Inicio: 12/11/2011	Fecha Fin: 19/11/2011
Programador Responsable: Merly López Barroso	
Descripción:	
Desarrollar la opción de graficar intervalo de uso por horas mostrando el comportamiento de los números de peticiones en el servidor durante las	

horas de un día seleccionado. A través de un calendario se selecciona el día que se desea y se brindará la información de las peticiones que se le hicieron al servidor ese día. Se hace una búsqueda en la base de datos, la cual guarda por días las peticiones hechas al servidor, y éste le proporciona al cliente el token con los datos requeridos.

Tabla 13:Tarea de Ingeniería Graficar peticiones por días

Tarea de Ingeniería	
Número Tarea: 3.2	Número Historia de Usuario: HU_3
Nombre Tarea: Graficar peticiones por días.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 8 días
Fecha Inicio: 22/11/2011	Fecha Fin: 29/11/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar intervalo de uso por días mostrando el comportamiento de los números de peticiones en el servidor durante los días de un mes seleccionado. A través de un combobox se selecciona el mes que se desea y se brindará la información de las peticiones que se le hicieron al servidor en los días pertenecientes a ese mes. Se hace una búsqueda en la base de datos, la cual guarda por días las peticiones hechas al servidor, y éste le proporciona al cliente el token con los datos requeridos.</p>	

Tabla 14: Tarea de Ingeniería Graficar peticiones por mes

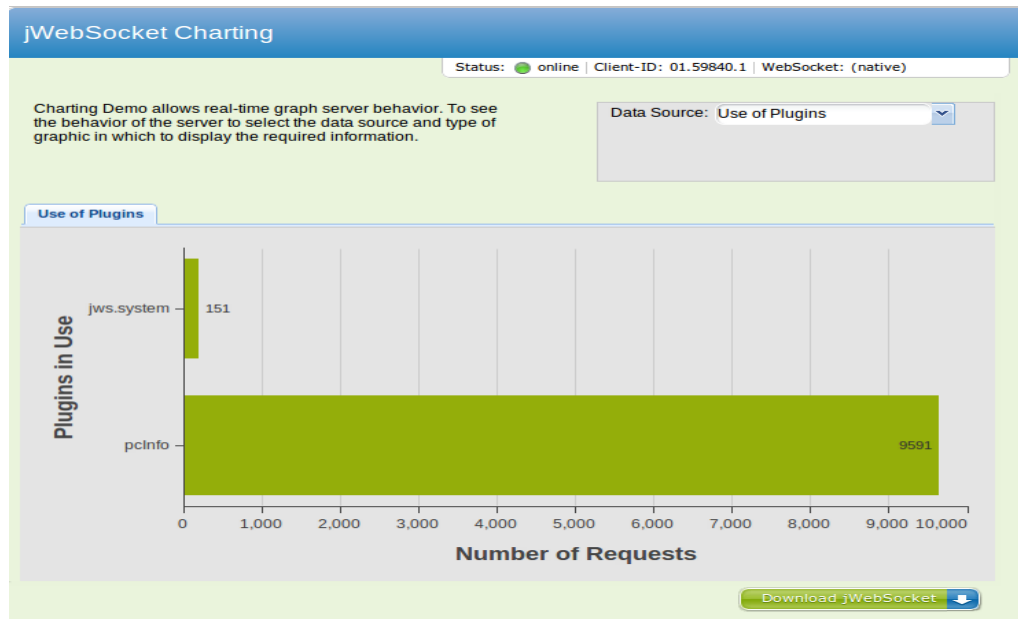
Tarea de Ingeniería	
Número Tarea: 3.3	Número Historia de Usuario: HU_3
Nombre Tarea: Graficar peticiones por mes.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 8 días

Fecha Inicio: 10/12/2011	Fecha Fin: 17/12/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar intervalo de uso por mes mostrando el comportamiento de los números de peticiones en el servidor durante los meses de un año seleccionado. A través de un combobox se selecciona el año que se desea y se brindará la información de las peticiones que se le hicieron al servidor en los meses pertenecientes a ese año. Se hace una búsqueda en la base de datos, la cual guarda por días las peticiones hechas al servidor, y éste le proporciona al cliente el token con los datos requeridos.</p>	

Tabla 15: Historia de Usuario Graficar Plug-ins en Uso

Historia de Usuario	
Número:HU_4	Nombre Historia de Usuario: Graficar plug-ins en uso.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Merly López Barroso	Iteración Asignada:2
Prioridad en Negocio: Alta	Puntos Estimados:3 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 3
<p>Descripción:</p> <p>La presente historia de usuario tiene como objetivo permitir que el usuario pueda ver el comportamiento de los plug-ins que se están utilizando. Se podrá observar la cantidad de peticiones hechas a cada uno de ellos.</p>	
Observaciones: Ninguna	

Prototipo de interfase:



A continuación se muestra en la Tabla 16 la descripción de la tarea de ingeniería que se realiza para lograr el cumplimiento de la historia de usuario anterior:

Tabla 16: Tarea de Ingeniería Graficar plug-ins en uso

Tarea de Ingeniería	
Número Tarea: 4.1	Número Historia de Usuario: HU_4
Nombre Tarea: Graficar plug-ins en uso.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 días
Fecha Inicio: 18/12/2011	Fecha Fin: 20/12/2011
Programador Responsable: Merly López Barroso	
<p>Descripción:</p> <p>Desarrollar la opción de graficar plug-ins en uso mostrando el comportamiento de la cantidad de peticiones que se le hacen a los plug-ins que se estén utilizando. Se hace una búsqueda en la base de datos, la cual guarda por id de los plug-ins las peticiones hechas a cada uno de estos, y el</p>	

servidor le proporciona al cliente el token con los datos requeridos.

2.6 Plan de Release

El Plan de Release o iteraciones permite realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente las cuales representan en su totalidad 4 HU. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha historia. Como resultado de la priorización de historias de usuario se llegó a la siguiente planificación que se muestra en la Tabla 17:

Tabla 17: Plan de Release

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 2	En esta iteración se va a desarrollar una primera versión de la aplicación para la cual se implementará la historia de usuario que constituye la funcionalidad fundamental de la aplicación. Esta tiene prioridad muy alta.	HU_1	4 semanas
Iteración 3	En esta iteración se va a desarrollar una segunda versión de la aplicación para la cual se implementarán las historias de usuarios que tienen prioridad alta.	HU_2 HU_3 HU_4	8 semanas

2.7 Arquitectura de Software de la Solución

La arquitectura de una aplicación es la vista conceptual de su estructura. Toda aplicación contiene código de presentación, de procesamiento y de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como esté distribuido su código.

Para desarrollar la aplicación se utiliza una arquitectura en capas. La programación por capas es una arquitectura cliente-servidor la cual tiene como objetivo primordial la separación de la lógica de negocios y la lógica de diseño. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que ocurra algún cambio sólo afectará dicho nivel, logrando obviar las demás capas del sistema. Permite además distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles. A continuación se presenta la Fig. 2 que muestra la arquitectura de la solución:



Fig. 2 : Arquitectura de la Solución. Fuente: Elaboración propia.

2.8 Diseño con Metáforas

La metodología SXP está basada en XP, y dicha metodología define un término llamado metáfora, el cual es considerado una historia compartida que describe cómo debería funcionar el sistema. La práctica de las metáforas consiste en formar

un conjunto de nombre que actúen como vocabulario para hablar sobre el dominio del problema. (Fowler, 2004)

.El Diseño con metáforas es sencillamente 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, que a su vez está compuesto por un diagrama de paquetes, el cual expone dicho diseño.

Los diagramas de paquetes describen los elementos físicos del sistema y sus relaciones. 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 muestra la Fig. 3 que representa el diagrama de paquetes para el sistema que se propone:

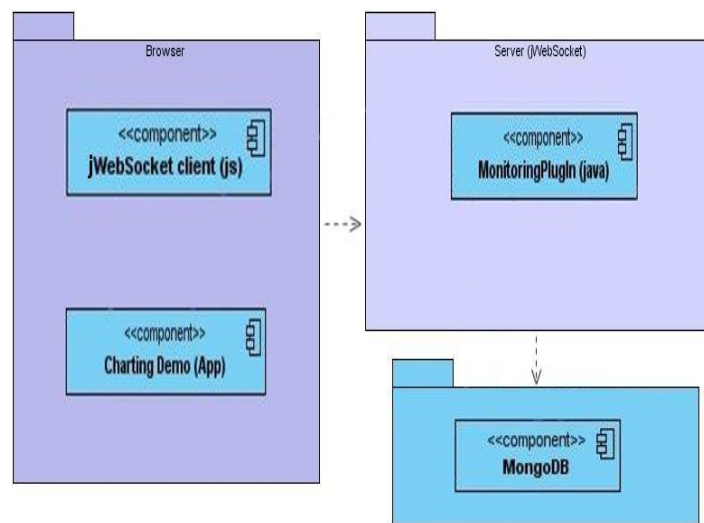


Fig. 3 Diagrama de Paquetes de la Solución Propuesta. Fuente: Elaboración propia.

En la capa del navegador se tienen los componentes jWebSocket Client y Charting Demo. jWebSocket Client constituye el cliente del marco de trabajo jWebSocket y Charting Demo representa la aplicación. Esta capa del navegador se relaciona con la capa del servidor de jWebSocket. Por otro lado la capa del servidor de jWebSocket

contiene el plug-in MonitorinPlug-in de la aplicación el cual se encarga de proveer al cliente la información y se relaciona con la capa del servidor de Base de Datos. Mientras que la capa del servidor de base de datos contiene la base de datos de MongoDB el cual interactúa con el servidor de jWebSocket, permitiendo persistir o consultar los datos de la aplicación.

2.9 Diagrama 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. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros.

A continuación se presenta la Fig. 4 que muestra el diagrama de componentes para el sistema que se propone:

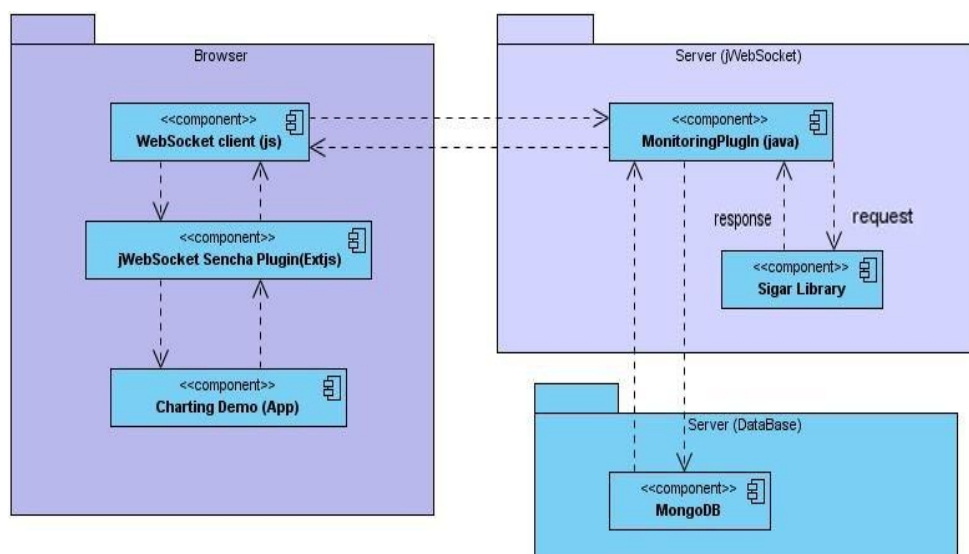


Fig. 4 Diagrama de Componentes de la Solución Propuesta. Fuente: Elaboración propia.

Los componentes son los plug-ins o filtros de las aplicaciones. El paquete de Browser contiene componentes con los que debe interactuar el cliente. El paquete Server (jWebSocket) contiene los componentes que interactúan en el servidor, el componente Sigar Library le provee al componente Monitoring Plug-in los recursos

de la máquina. El paquete Server (DataBase) contiene el componente MongoDB, el cual interactúa con el servidor para proporcionar los datos persistentes.

Conclusiones del Capítulo

En este capítulo se definieron las características y funcionalidades del sistema desarrollado. Quedaron aprobados los 10 requisitos funcionales y los 24 no funcionales, así como las 4 historias de usuario necesarias para obtener un sistema eficiente. Se obtuvo la aplicación Charting Demo como propuesta de solución para la representación gráfica de datos en tiempo real utilizando el marco de trabajo `jQueryWebSocket`.

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Introducción

En el presente capítulo la implementación de la solución propuesta en el capítulo anterior queda documentada, generándose el diagrama de despliegue del sistema desarrollado. Además se exponen los casos de pruebas o test de aceptación a las que fue sometida la aplicación en cada una de las iteraciones. El cumplimiento de estos casos de pruebas fue el hito para avanzar hacia la próxima iteración. En este capítulo además de las pruebas se dan a conocer los resultados obtenidos hasta el momento.

3.1 Implementación

La aplicación Charting Demo está desarrollada sobre el marco de trabajo jWebSocket. Las librerías y clases clientes para la aplicación Charting Demo están disponibles solamente para el lenguaje JavaScript y se encuentran almacenadas en la carpeta germanyClient/demos/jwsCharts.

A continuación se muestra en la Fig. 5 una descripción del cliente de la aplicación Charting Demo.



Fig. 5: Descripción del cliente de la aplicación Stock Ticker Demo. Fuente: Elaboración propia.

En el lado del cliente se hace uso del siguiente archivo javascript:

- **charting.js:** Permite representar gráficamente los datos que llegan del servidor.
- Se hace uso además de archivos htm y css para la interfaz de usuario de la aplicación:

- **jwsCharting.html:** En este archivo está definido todo el código HTML de la aplicación.
- **jwsCharts.css:** Incluye el css correspondiente al HTML de la aplicación.

En el lado del servidor la aplicación Charting Demo hace uso de la estructura de paquetes que se observa en la Fig. 6 :



Fig. 6: Descripción del servidor de la aplicación Charting Demo. Fuente: Elaboración propia.

- **org.jwebsocket.filter:** Contiene un filtro para registrar los plug-ins en uso.
- **org.jwebsocket.plugin-ins.monitoring:** Contiene el plug-in que implementa todas las funcionalidades de la aplicación y un listener para persistir datos en mongoDB.

Las librerías y herramientas utilizadas en el lado del servidor son las siguientes:

- **Librería Hyperic Sigar:** versión 1.6.4. Sigar es un sistema recolector de información y reportero, que permite obtener datos de la computadora. 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.
- **MongoDB:** Mongo es una base de datos orientada a documentos JSON, salvo que está diseñada para ser una verdadera base de datos de objetos, más que para un almacenamiento de clave/valor puro.

Los métodos más significativos en el lado del servidor se describen a continuación:

- processToken(): Recibe los mensajes entrantes de cualquier cliente WebSocket.

- broadcastServerXchgInfo(WebSocketConnector aConnector): Se encarga de enviar un token con la información de las peticiones que se le han hecho al servidor en el día actual.
- broadcastPlug-insInfo(WebSocketConnector aConnector): Se encarga de enviar un token con la información de las peticiones que se le han hecho a los plug-ins que están en uso.
- computerInfoToToken(): Devuelve un token con la información de los recursos de la máquina, tales como espacio de memoria usada, espacio de memoria libre, total de memoria, uso del swap, consumo de cpu, y su consumo total.

En el lado del cliente:

- plug-in.processToken = function(aToken): El cliente contiene una función la cual es la encargada de procesar todos los token que llegan del servidor, y según su tipo almacena la información correspondiente para luego ser mostrada gráficamente.

3.2 Validación de la Investigación

Para validar la presente investigación que tiene como objetivo desarrollar una aplicación web que mejore los niveles de usabilidad del marco de trabajo jWebScket se decidió trazar una estrategia de validación basadas en casos de prueba a la aplicación, certificación de calidad de software y uso de una valoración del cliente.

En la primera etapa se hacen casos de pruebas funcionales alrededor de las historias de usuarios que garantizan el cumplimiento de los requisitos funcionales que debe cumplir la aplicación que fue diseñada.

Seguidamente se efectúa un proceso de certificación de calidad de software por el grupo de calidad del Centro de Desarrollo de la Facultad Regional “Mártires de Artemisa”. Este grupo como agente externo al proyecto realiza un proceso de certificación basados principalmente en diferentes elementos funcionales y documentales, de acuerdo a la metodología SXP seleccionada para el desarrollo de la aplicación demostrativa.

Además de optar por la certificación del grupo de calidad, la aplicación es puesta a consideración del líder de la comunidad de jWebSocket Internacional Alexander Schulze. Esta etapa de validación tiene como objetivo lograr que dicho cliente ofrezca una certificación sobre la aplicación desarrollada en la integración al marco de trabajo jWebSocket y la real usabilidad que pueda tener dicha aplicación para los usuarios potenciales y desarrolladores de la comunidad de jWebSocket.

3.2.1 Casos de Pruebas

Se definieron casos de prueba funcionales para todas las historias de usuario teniendo en cuenta el cumplimiento de los requisitos funcionales de la aplicación, a continuación se dan a conocer las pruebas que se realizaron a cada una de las historias de usuario con las que cuenta la aplicación Charting Demo y los resultados obtenidos en el diseño de cada una. A continuación se muestra en la Tabla 18 los casos de pruebas realizados. En el Anexo 2 se podrán observar los restantes casos de pruebas.

Tabla 18: Casos de pruebas para la HU Graficar recursos de la máquina.

Caso de Prueba Funcional	
Código Caso de Prueba: jws-1-1	Nombre Historia de Usuario: Graficar recursos de la máquina.
Nombre de la persona que realiza la prueba: Merly López Barroso	
Descripción de la Prueba: Esta prueba consiste en verificar que se pueda representar gráficamente el uso del CPU, el total de CPU, el uso de la memoria y el uso del swap.	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket y el servidor de la aplicación tiene que estar ejecutándose.	
Entrada / Pasos de ejecución: Para representar gráficamente los recursos de la máquina se debe hacer clic en el menú “ <i>Data Source</i> ” que se encuentra en la esquina superior derecha de la aplicación, se seleccionará la opción “ <i>Machine Resources</i> ”, luego dirigirse al área	

central de la aplicación en la cual se presionará en la pestaña “ <i>Machine Resources</i> ” y se mostrarán en % mediante una gráfica de barras, el consumo de CPU, el total de CPU, el uso de la memoria y el uso del swap.
--

Resultado Esperado: Se muestran los datos mediante gráficas.

Evaluación de la Prueba: Satisfactoria.
--

3.2.2 Certificación de Calidad del Software

Para dar la certificación de calidad del software se puso a consideración del grupo de calidad del centro de desarrollo de la facultad regional el proyecto de software con los artefactos generados por la metodología SXP. Los principales artefactos generados son: Lista de Reserva del Producto (LRP), Modelo de Historia de Usuario del negocio, Historia de Usuario, Arquitectura de Software, Modelo Canónico de Datos, Tarea de Ingeniería, Plan de Release, Estándar de Código, Caso de Prueba Funcional, Manual de Usuario, Manual de Instalación y Manual de Desarrollo. Durante este proceso se verifica la claridad en la redacción de toda la documentación de forma tal que facilite el entendimiento de la misma por parte de otros desarrolladores o usuarios. A su vez estos artefactos deben cumplir con los estándares de calidad establecidos por la Universidad de la Ciencias Informáticas para garantizar de esta forma la continuidad del proyecto.

También se puso a consideración del grupo de calidad el código fuente de la aplicación, este grupo llegó a un proceso de revisión en el cual se analizó detalladamente la estandarización del código.

El proceso de revisión de los artefactos se llevó a cabo por la Ing. Maidel Ojeda Cruz, asesora de calidad del Centro de Desarrollo de la Facultad Regional “Mártires de Artemisa” y el proceso de revisión del código fuente fue desarrollado por el Ing. Domma Moreno Dager, asesor de tecnología del mismo centro. Se garantiza de esta manera la limpieza, el funcionamiento y la calidad de la estandarización del código.

3.2.3 Valoración del Cliente

Con el objetivo de comprobar el funcionamiento de la aplicación, la presente investigación se puso a consideración del cliente Alexander Schulze.

Toda la documentación así como el código implementado fue publicada en el Subversion de jWebSocket Internacional. Fue desarrollada una documentación para apoyar el proceso de valoración del cliente de la aplicación demostrativa. Para ello se confeccionó un manual de desarrollador, de usuario y de instalación en el idioma inglés para que fluyera mejor la comunicación y así garantizar un mayor entendimiento de la aplicación por parte del cliente. En la realización de este proceso fueron señaladas algunas imprecisiones y dificultades a la aplicación, las cuales fueron solucionadas y revisadas nuevamente por el cliente. Finalmente el cliente estuvo de acuerdo con la calidad y las funcionalidades de la aplicación.

3.2.4 Resultados Obtenidos

Al realizar los casos de pruebas a la aplicación Charting Demo se obtuvieron resultados satisfactorios, ya que ésta demuestra que la aplicación cumple con todas las funcionalidades que permite la representación de información gráfica en tiempo real en la Web garantizando así que la aplicación cumpla con las funcionalidades para la que fue diseñada.

Por otro lado la certificación de calidad mediante terceros arrojó resultados positivos alrededor de la calidad, limpieza y estandarización del código y de toda la documentación que se necesita, esto va a permitir que pueda ser usable y generalizada para otros usuarios y asegura además su disponibilidad ante futuras mejoras.

El criterio emitido por el cliente permite que la aplicación pueda ser utilizada por la comunidad de jWebSocket e integrado con el proyecto internacional jWebSocket.

Estos elementos unidos van a tributar a aumentar los niveles de usabilidad del marco de trabajo jWebSocket. Teniendo en cuenta los resultados satisfactorios de cada una de las etapas de la estrategia de validación asumida en la presente investigación, queda disponible la versión 1.0 de la aplicación Charting Demo.

3.3 Funcionalidades Obtenidas

Entre las principales funcionalidades que posee la aplicación Charting Demo en su versión 1.0 se pueden mencionar:

- Representar gráficamente el consumo de CPU, el total de CPU utilizado, el consumo de memoria y de swap.
- Representar gráficamente el espacio libre en disco duro y el espacio utilizado en disco duro.
- Representar gráficamente los usuarios que se encuentren conectados a la aplicación.
- Representar gráficamente las peticiones que se le hagan al servidor por horas.
- Representar gráficamente las peticiones que se le hagan al servidor por días.
- Representar gráficamente las peticiones que se le hagan al servidor por meses.
- Representar gráficamente los plug-ins en uso.

3.4 Diagrama de Despliegue

El diagrama de despliegue es un artefacto que modela la arquitectura en tiempo de ejecución de un sistema. El diagrama de despliegue es el complemento del diagrama de componente que, unidos, proveen la vista de implementación del sistema. Se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. En él se indica la situación física de los componentes lógicos desarrollados, lo que significa que sitúa el software en el hardware que lo contiene.

A continuación se muestra en la Fig. 7 el diagrama de despliegue, que representa la distribución física del sistema en términos de cómo se distribuirán las funcionalidades entre los nodos, donde cada nodo representa un recurso de cómputo, siendo estos procesadores o dispositivos hardware que se necesitan para el despliegue del sistema:

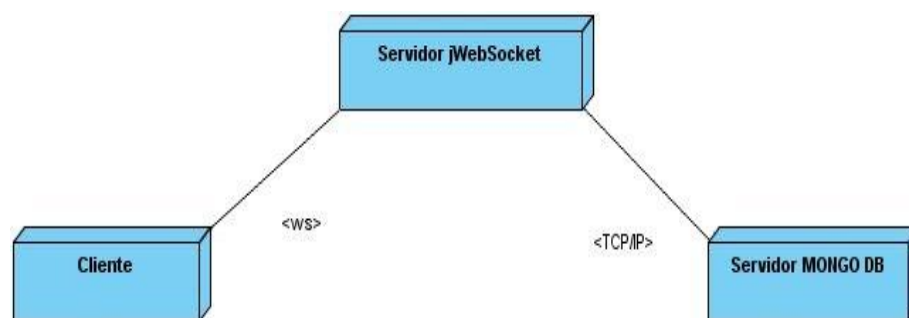


Fig. 7 Diagrama de Despliegue de la Solución Propuesta. Fuente: Elaboración propia.

El cliente se conectará mediante el protocolo WebSocket al Servidor jWebSocket para lo cual solo necesitará un navegador que soporte WebSocket sobre cualquier sistema operativo. Por otro lado el servidor de jWebSocket se conectará al servidor de base de datos de MongoDB a través del protocolo TCP/IP para persistir o consultar los datos. Mientras el servidor de base de datos interactúa con el servidor de jWebSocket para proporcionar los datos persistentes.

3.5 Aporte Social y Económico

La aplicación Charting Demo da aportes sociales significativos. Puede ser utilizada en la gran mayoría de los sectores de la sociedad. Es un elemento fundamental para la toma de decisiones ágiles y eficientes, esto representa una gran ventaja en el mercado, ya que permite conocer sus estadísticas en tiempo real, y tomar decisiones importantes a partir de ellas, brindando la posibilidad de analizar cómo se desenvuelven el resto de los competidores del mercado. Puede ser de gran ayuda a inversionistas y hombres de negocios para llevar el control de sus inversiones y estados financieros. También proporcionará gran ayuda para dar conferencias que requieran mostrar la evolución de determinada información. Los negocios en Internet juegan un papel primordial, debido a que hoy en día todo se mueve alrededor de la Web. Tiendas virtuales en Internet, tableros de cotizaciones, control de usuarios, control de recursos de servidores en Internet que brindan servicios, son algunos de los medios en los que más se podrá utilizar esta aplicación.

La aplicación Charting Demo logra representar gráficamente datos en tiempo real, por lo que aporta en gran medida al marco de trabajo, debido a que muestra la capacidad del marco de trabajo jWebSocket en el desarrollo de aplicaciones que grafiquen información real con altos niveles de variación en el tiempo. Los desarrolladores logran identificar con claridad las potencialidades y las bondades que ofrece el marco de trabajo jWebSocket para la representación gráfica. Esto provoca que empresas y desarrolladores que necesitan graficar datos en tiempo real, reconozcan al marco jWebSocket como una vía eficiente para desarrollar aplicaciones con estas funcionalidades para mayor satisfacción de sus clientes. Aumentando los niveles de usabilidad del marco de trabajo. Trae consigo además un aumento de cuotas de mercado para el marco de trabajo y eleva el número de usuarios y clientes potenciales a nivel internacional.

Si se hiciera uso de la aplicación Charting Demo se lograría una disminución de costos considerable. Las empresas y organizaciones que se dedican hoy a desarrollar soluciones que representan datos mediante gráficos en tiempo real se beneficiarían de las potencialidades que ofrece esta aplicación, así como de la ventaja de ser código abierto.

Conclusiones del Capítulo

Al concluir el presente capítulo se describió como quedó estructurada la implementación en el lado del cliente y del servidor de la aplicación, además se definió el diagrama de despliegue, en el cual queda reflejada la situación física de los componentes desarrollados. Se realizaron pruebas a la aplicación, mostrando al cliente que el sistema cumple con todas las funcionalidades requeridas, dándole una calidad máxima a la solución dada. Quedó demostrada que la aplicación Charting Demo se encuentra en óptimas condiciones para representar gráficamente datos en tiempo real logrando así aumentar en gran medida los niveles de usabilidad del marco de trabajo jWebSocket.

CONCLUSIONES

Con la realización de la presente investigación se hizo un estudio de los fundamentos teórico-metodológicos que permite definir que la representación gráfica en la Web es una herramienta que brinda la posibilidad de interpretar mejor la información que se analiza, haciéndola más entendible e interesante. Al realizar el estudio del arte de la investigación arrojó como resultado que las soluciones actuales que representan datos gráficamente presentan inconvenientes tales como la latencia, el ancho de banda, los recursos del sistema y la licencia de los productos que hacen necesaria ir en búsqueda de una aplicación que logre representar gráficamente datos en tiempo real utilizando el marco de trabajo `WebSocket`. Por tal motivo se desarrolló la aplicación web Charting Demo que posibilita la representación gráfica en tiempo real utilizando el marco de trabajo `WebSocket`, a dicha aplicación se le realizó una estrategia de validación basada en diferentes etapas como son las pruebas funcionales, la certificación de calidad de software y la valoración del cliente para garantizar un posible aumento de los niveles de usabilidad de `WebSocket`.

RECOMENDACIONES

Se recomienda para versiones posteriores de la aplicación Charting Demo incorporar nuevas funcionalidades que tributen a aumentar las potencialidades del proceso de representación gráfica de datos en la Web en tiempo real desarrolladas con el marco de trabajo `jQueryWebSocket`.

BIBLIOGRAFÍA REFERENCIADA

- Sánchez, Jordi. 2006.** ¿Qué es un 'framework'? Valencia : <http://jordisan.net/blog/2006/que-es-un-framework>, 2006.
- 10gen, Inc. 2011.** Agile and Scalable . *mongoDB*. [En línea] 2011. [Citado el: 16 de 02 de 2012.] <http://www.mongodb.org/>.
- . **2011.** What is MongoDB? *10gen*. [En línea] 2011. [Citado el: 16 de 02 de 2012.] <http://www.10gen.com/why-mongodb>.
- 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/>.
- Asinsten, Juan Carlos.** *Comunicación visual y tecnología de gráficos en computadora*.
- Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato. 2004.** Control de versiones con Subversion. [En línea] 2004. [Citado el: 22 de 02 de 2012.] <http://svnbook.red-bean.com/nightly/es/svn-ch-1-sect-1.html>.
- Carga de clases dinámicamente con ExtJS 4.* **Villa, Crysfel. 2011.** s.l. : <http://www.quizzpot.com/2011/10/carga-de-clases-dinamicamente-con-extjs-4/>, 2011.
- 2009.** *CENTRO DE INVESTIGACIÓN Y DESARROLLO*. Lima : s.n., 2009.
- Comet-HTML5-Websocket.* **Egli, Peter R. 2011.** Indigoo : <http://www.slideshare.net/PeterREgli/comet-websockets-html5>, 2011.
- Curcio. 1989.** *Developing Graph Comprehension. Elementary and Middle School*. 1989.
- D., Pittenger. 1995.** *Teaching students about graphs. Teaching of Psychology*. 1995.
- 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.
- Extjs. 2009.** www.extjs.es. [En línea] 2009. [Citado el: 10 de 12 de 2011.] <http://www.extjs.es>.
- Few, Stephen. 2007.** *Data Visualization Past, Present, and Future*. 2007.

Figuerola, 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>.

Fowler, Martin. 2004. Is Design Dead? [En línea] 5 de 2004. [Citado el: 27 de 3 de 2012.] <http://www.martinfowler.com/articles/designDead.html>.

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.

Friel, S. , Curcio, F. y Bright, G. 2001. *Making sense of graphs: Critical factors influencing. Comprehension and Instructional Implication.* 2001.

Friel, S., Curcio, F. y Bright, G. 2001. *Making sense of graphs: Critical factors.* 2001.

Garfield, J. 2003. *Assessing statistical reasoning. Statistical Education Research Journal.* 2003.

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

—. 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.

Henry, G. T. 1995. *Graphing Data. Techniques for Display and Analysis. Applied Social.* 1995.

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

HTML 5 WebSockets in Theory and Practice. Conference Kaazing. **Jacobi, J., & Fallows, J. R. 2011.** 2011.

HTML5 Websockets and communication. **Lubbers, Petter. 2010.** Java User Group Meeting : <http://www.slideshare.net/Kaazing/v2-peterlubberssfjugwebsocket>, 2010.

Hyder, Shahriar. 2011. *Taking a Quantum Leap with Html5 WebSocket: Taking bi-directional communication on the web to the next level.* 2011.

IBM's Corporate. 2011. Rational Rose Enterprise. IBM. [En línea] 2011. [Citado el: 17 de 02 de 2012.] <http://www-142.ibm.com/software/products/es/es/enterprise/>.

J., Pittenger D. 1995.*Teaching students about graphs. Teaching of Psychology.* 1995.

Kaazing Corporation. 2012. Kaazing, WebSocket Gateway. [En línea] 2012. [Citado el: 22 de 02 de 2012.] <http://kaazing.com/products/kaazing-websocket-gateway.html>.

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

M., Pfannkuch. 2006.*Statistics Education Research Journal.* 2006.

Metodologia Agil para el desarrollo de software. 2010. 2010.

Metodologías Tradicionales Vs. Metodologías Ágiles. **Figuerola, Roberth G., Solís, Camilo J. y Cabrera, Armando A. 2011.** Loja : <http://www.docstoc.com/docs/102328746/articulo-metodologia-de-sw-formato>, 2011.

Monteiro, C. y Ainley, J. 2006.*Proceedings of the Seventh International Conference on Teaching Statistics.* 2006.

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

Pérez, Alan García. 2009.*Guía para la presentación de gráficos estadísticos.* Lima : s.n., 2009.

Pfannkuch. 2006.*Statistics Education Research Journal.* 2006.

Pittenger, D. J. 1995.*Teaching students about graphs. Teaching of Psychology.* 1995.

Proceedings of the Seventh International Conference on Teaching Statistics.

Monteiro, C. y Ainley, J. 2006.2006.

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

2010.*Real-Time-Web.* 2010.

Research on statistic learning and reasoning. **Shaughnessy, J.M. 2007.** 2007.

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

Schwaber, Ken y Beedle, Mike. 2011.*Agile Software Development with SCRUM.* s.l. : Prentice Hall, 2011. 0130676349.

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/>.

Shaughnessy, J. M. 2007. *Research on statistic learning and reasoning. Second Handbook of Research on Mathematics Teaching and Learning*. 2007.

Shaughnessy, J.M. 2007. *Research on statistic learning and reasoning*. 2007.

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

SXP, Metodología Ágil para el Desarrollo de Software. **Peñalver, G, Meneses, A y García, S. 2010.** Chile : 1er congreso Iberoamericano de Ingeniería de Proyectos, 2010.

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

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/>.

WebSocket für alle. **Alexander Schulze, Rolando Santamaría Masó. 2011.** 2, Alemania : Mobile Developer Android, 2011.

Wells, Don. 2009. Extreme Programming: A gentle introduction. *Extreme Programming*. [En línea] 28 de 07 de 2009. [Citado el: 16 de 02 de 2012.] <http://www.extremeprogramming.org/>.

Wordpress.com. 2007. Librería ExtJs. *Desarrollo WEB y otras Hierbas*. [En línea] 06 de 08 de 2007. [Citado el: 13 de 12 de 2011.] <http://vargasti.wordpress.com/2007/08/06/libreria-extjs/>.