



Universidad de las Ciencias  
Informáticas

## **Facultad Regional Mártires de Artemisa**

Control remoto de computadoras en un ambiente Web de tiempo real  
usando el marco de trabajo jWebSocket.

Trabajo de Diploma en opción del título de Ingeniero en Ciencias  
Informáticas.

**Autor:**

**Yasmany Nuñez Broch**

**Tutores:**

**MSc. Yamila Vigil Regalado**

**Lic. Gilberto Ramón Justiniani Fernández**

Artemisa, Cuba, Junio 2012

## **DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de este trabajo y autorizo al Centro de Desarrollo de 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 2012.

---

**Yasmany Nuñez Broch**

Firma del Autor

---

**Yamila Vigil Regalado**

Firma del Tutor

---

**Lic. Gilberto Ramón Justiniani**

Firma del Co-tutor



*"El único autógrafo digno de un hombre es el que deja escrito con sus obras".*

*José Martí*

## **AGRADECIMIENTOS**

Quiero agradecer a todas las amistades que se preocuparon por la realización de este trabajo.

A mi familia por apoyarme en todo momento.

A mi tutora por los consejos y las correcciones aportadas a este documento.

## **DEDICATORIA**

A mi madre por todos estos años de sacrificio pero llenos de amor.

A toda mi familia en especial a mis abuelos maternos por ser mi punto de apoyo en los momentos difíciles y por inculcarme los más sinceros valores que una persona puede portar.

A mi esposa por dejarme entrar en su corazón.

## **RESUMEN**

El desarrollo de las aplicaciones web fomentó la migración de las antes llamadas aplicaciones de escritorio al nuevo mundo de las redes e Internet. El hecho que estas aplicaciones no dependen de ningún sistema operativo, no requieren configuración de hardware y poseen facilidades de acceso, hizo que fueran utilizadas para una gran diversidad de tareas. El control remoto de computadoras fue un proceso que no quedó exento y de igual manera fue llevado a la Web.

La interactividad en la Web exige un intercambio de experiencias en tiempo real y en la actualidad no es posible realizar el control remoto de computadoras en tiempo real a través de la Web. Esto trae consigo un bajo nivel de usabilidad y confiabilidad en las aplicaciones destinadas para estos fines.

Por los beneficios que brinda realizar el control remoto de computadoras a través de la Web, se identifica la necesidad de desarrollar una aplicación web que permita el control remoto de computadoras en tiempo real a través de la Web utilizando el marco de trabajo `WebSocket`. Este no es más que una tecnología orientada al desarrollo de aplicaciones basadas en el protocolo de comunicación `WebSocket`. Este protocolo proporciona un canal de comunicación bidireccional, permitiendo la entrada y salida de datos de manera simultánea.

## ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA .....</b>	<b>8</b>
1.1. Fundamentos Teórico-Metodológicos .....	8
1.2. Análisis de Soluciones Existentes .....	10
1.3. Metodología usada para el desarrollo de la solución .....	12
1.4. Herramientas y Tecnologías usadas para el desarrollo de la solución.....	16
1.4.1. Lenguajes de Programación y Modelado.....	16
1.4.2. Otros Lenguajes.....	17
1.4.3. Marco de Trabajo del lado del Servidor .....	18
1.4.4. Marcos de Trabajo del lado del Cliente.....	19
1.4.5. Herramienta CASE .....	20
1.4.6. Herramientas de Control de Versiones .....	22
1.4.7. Cliente de Control de Versiones .....	23
1.4.8. Entorno Integrado de Desarrollo - IDE.....	25
<b>CAPÍTULO 2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA.....</b>	<b>27</b>
2.1. Propuesta de Solución .....	27
2.2. Planificación del Proyecto por Roles .....	28
2.3. Modelo de Dominio .....	31
2.4. Lista de Reserva del Producto (LRP) .....	32
2.5. Historias de Usuario y Tareas de Ingeniería .....	36
2.6. Plan de Releases .....	43
2.7. Descripción de la Arquitectura de Software.....	44
2.8. Diseño con Metáforas.....	45

2.9. Diagrama de Componentes.....	46
<b>CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA.....</b>	<b>49</b>
3.1. Implementación .....	49
3.2. Validación de la Investigación .....	50
3.3. Resultados Obtenidos .....	58
3.4. Funcionalidades Obtenidas .....	58
3.5. Diagrama de Despliegue .....	59
3.6. Aporte Social y Económico.....	60
<b>CONCLUSIONES GENERALES .....</b>	<b>61</b>
<b>RECOMENDACIONES .....</b>	<b>62</b>
<b>BIBLIOGRAFÍA REFERENCIADA .....</b>	<b>63</b>
<b>ANEXOS .....</b>	<b>66</b>



## INTRODUCCIÓN

La era de la informática ha abierto un sinfín de oportunidades, para mejorar los procesos de manera eficaz, oportuna y eficiente. En el mundo de hoy las tecnologías presentan un gran avance dado el desarrollo científico-técnico alcanzado, es por ello que la informatización de la sociedad ha crecido a escala mundial. El desarrollo de las tecnologías de red facilitó la interconexión de computadoras lo que dio paso al intercambio de datos y al surgimiento de métodos para el manejo remoto de las computadoras.

El control remoto de computadoras permite acceder a los datos almacenados en el computador, monitorizar el estado de sus recursos y realizar disímiles configuraciones en los archivos del sistema sin la necesidad de estar físicamente operando en ella. Estos son algunos de los usos que normalmente se le atribuyen al control remoto de computadoras. Muchos son los beneficios que aportan realizar este tipo de operaciones. Tener acceso desde el hogar o desde otro sitio como por ejemplo cuando las personas se encuentran de viaje.

En los inicios de la computación solo existían aplicaciones del tipo consola, posteriormente aparecieron las aplicaciones de escritorio, y como toda evolución, en los años 90 con el nacimiento de Internet fueron surgiendo lo que hoy conocemos como aplicaciones web. En ese entonces se limitaban a ser simples páginas de texto estático, pero que con el tiempo han ido tomando fuerza hasta llegar a lo que conocemos hoy en día.

Una de las principales ventajas que presentan las aplicaciones web ante las aplicaciones de escritorio o consola es el hecho de que no dependen de ningún sistema operativo ni configuración de hardware específica. Para su ejecución simplemente basta con teclear su dirección URL en cualquier navegador web. De igual manera sus actualizaciones se hacen de una manera muy sencilla, sin necesidad de hacer descargas, instalaciones o comprar físicamente el producto.

Esta flexibilidad ha sido uno de los principales motivos por los que cada vez son más utilizadas para una creciente diversidad de tareas. Hoy en día Internet

constituye el medio de comunicación más utilizado gracias entre otras cosas a la facilidad de acceso desde los diferentes dispositivos móviles. No solo se consulta e intercambia información a través de Internet, sino que se utiliza para realizar la informatización de complejos procesos.

Por el uso intensivo de la Web hoy en día debido a las potencialidades que brinda a los usuarios, estos exigen mayor interactividad e intercambio de experiencias en tiempo real. Los usuarios desean recibir notificaciones inmediatas de todas las acciones que realizan, así como el resultado de las tareas que han sido ejecutadas, las cuales solicitan un alto nivel de concurrencia. Esto hace que la comunicación no se establezca siempre ante una solicitud del usuario, sino que la comunicación sea proactiva, que la Web pueda comunicarse con ellos también sin una acción precedente.

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. (Schulze, 2011) HTTP fue intencionalmente diseñado como un simple protocolo de solicitud-respuesta. Este esquema no proporciona una comunicación bidireccional entre el cliente y el servidor. Esto ocasiona que si se genera un nuevo evento en el servidor, este no es capaz de enviarle directamente un mensaje al cliente de manera asíncrona. En todos los casos el cliente debe realizar una solicitud, para así el servidor pueda enviar una nueva información, ocasionando demoras en la entrega de datos actualizados al usuario final. (Egli, 2011).

Para brindar solución a la comunicación web en tiempo real surge HTML5 WebSocket. WebSocket es una idea impulsada por distintos grupos internacionales entre los que se destaca la W3C (WWW Council) quien establece las normas oficiales para los estándares en la Web. El primer borrador del protocolo Websockets se publicó el 9 de enero del 2009 y ha ido evolucionando hasta la actualidad. Websockets soluciona hoy las limitaciones del protocolo HTTP, así como las técnicas y tecnologías diseñadas anteriormente. Este protocolo establece

una comunicación *full-duplex*<sup>1</sup>(TCP) entre el cliente y el servidor, sustituyendo la comunicación *half-duplex*<sup>2</sup>(HTTP).

Websockets 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. Muchos autores plantean, ante las nuevas exigencias de la Web, que en poco tiempo Websockets y sus beneficios remplazarán al protocolo HTTP y sus limitaciones.

Los principales servidores que soportan Websockets para el desarrollo de aplicaciones hoy día son, la Pasarela Websockets de Kaazing<sup>3</sup>, Jetty WebSocketServlet<sup>4</sup>, Socket.IO<sup>5</sup>, django-websocket del proyecto Python<sup>6</sup> y jWebSocket<sup>7</sup>.

JWebSocket es un framework 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. El cliente nativo soporta el intercambio de paquetes en los formatos JSON, XML y CSV que en entornos específicos se pueden utilizar sin la necesidad de manejarlos a través de tokens. El cliente jWebSocket tiene una arquitectura de *plugin*<sup>8</sup> que permite aumentar con facilidad sus funcionalidades. (Schulze, 2011)

Existe un determinado número de aplicaciones web que simulan el comportamiento de una consola del sistema. La mayoría de ellas son utilizadas en ambientes de

---

<sup>1</sup> full-duplex: Cualidad de los elementos que permiten la entrada y salida de datos de forma simultánea.

<sup>2</sup> half-duplex: Significa que el método o protocolo de envío de información es bidireccional pero no simultáneo.

<sup>3</sup> <http://kaazing.com/>

<sup>4</sup> <http://www.eclipse.org/jetty/>

<sup>5</sup> <http://socket.io/>

<sup>6</sup> <http://pypi.python.org/pypi/django-websocket>

<sup>7</sup> <https://jWebSocket.org/>

<sup>8</sup> plugin: Conjunto de componentes de software que añade capacidades específicas.

servidores web para realizar el control de los mismos. Cada una de ellas son creadas con un objetivo específico, pero presentan el problema de no recrear un correcto ambiente de tiempo real.

Debido a los hechos anteriormente planteados surge la siguiente **situación problemática**:

Las aplicaciones web que realizan el control remoto de computadoras a través de la Web lo hacen utilizando técnicas avanzadas de comunicación con el servidor para mejorar la capacidad de respuesta del mismo. Estas técnicas sobrecargan al servidor con peticiones y aumentan considerablemente el uso de los recursos de la red.

En la actualidad no es posible realizar el manejo remoto de computadoras en tiempo real a través de la Web. Esto trae consigo un bajo nivel de usabilidad en las aplicaciones destinadas para estos fines, al no recrear un correcto ambiente de tiempo real, es decir que no muestran los procesos consultados en un corto intervalo de tiempo. Ante ello el control remoto de computadoras no logra la eficiencia requerida debido al alto nivel de concurrencia que solicitan las tareas que van a ser ejecutadas. Finalmente estos problemas generan una poca confiabilidad por parte de los usuarios finales al usar este tipo de aplicaciones.

Por los motivos antes expuestos se formuló el siguiente **problema científico de la investigación**: ¿Cómo garantizar mayor eficiencia en el control remoto de computadoras a través de la Web en tiempo real?

La investigación tiene como **objeto de estudio** el control remoto de computadoras, cuyo **campo de acción** es el control remoto de computadoras en la Web en tiempo real usando el marco de trabajo jWebSocket.

Para dar solución al problema científico se propone como **objetivo general**: Desarrollar una aplicación web utilizando el marco de trabajo jWebSocket que garantice mayor nivel de eficiencia en el proceso de control remoto de computadoras en tiempo real.

Del objetivo general se derivan las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teórico-metodológicos del control remoto de computadoras en tiempo real a través de la Web?
- ¿Cuál es la situación actual del control remoto de computadoras en tiempo real a través de la Web?
- ¿Cómo desarrollar una aplicación web para el control remoto de computadoras en tiempo real con WebSocket que garantice mayor nivel de eficiencia en el proceso de control remoto de computadoras?
- ¿Cuáles son los resultados obtenidos al utilizar la aplicación web desarrollada con WebSocket para garantizar mayor nivel de eficiencia en el proceso de control remoto de computadoras?

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

- Fundamentación teórico-metodológica de la investigación
- Análisis de la situación actual del control remoto de computadoras
- Desarrollo de una aplicación web para el control remoto de computadoras en tiempo real con WebSocket que garantice mayor nivel de eficiencia en el proceso de control remoto de computadoras
- Validación de la aplicación web desarrollada con WebSocket para garantizar mayor nivel de eficiencia en el proceso de control remoto de computadoras

Para una correcta comprensión y realización del trabajo se hace uso de los siguientes **métodos de la investigación científica**:

#### **Métodos Teóricos:**

**Histórico-Lógico:** Permite analizar la trayectoria completa acerca del control remoto de computadoras a través de la web, así como el estudio histórico de las mismas que permite ver deficiencias y proponer soluciones acorde a las necesidades.

**Análisis documental:** Mediante este método se hizo un estudio de una variada documentación referente al control remoto, en específico al control de computadoras a través de la web y las herramientas utilizadas actualmente para lograrlo, con el objetivo de obtener a través de estas 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 el módulo, así como las experiencias de otros desarrolladores en el trabajo con las mismas.

**Analítico-Sintético:** 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 adquiriera una mayor preparación sobre el tema en cuestión.

**Modelación:** Este método permite realizar una representación de la situación que se analiza. Permite obtener mediante diagramas y objetos una mayor comprensión del problema y desarrollar un modelo para la aplicación a desarrollar a partir de la situación problemática.

Se cuentan con las siguientes variables:

**Variable Independiente:** Aplicación web para el control remoto de computadoras en tiempo real utilizando el marco de trabajo jWebSocket.

**Variable Dependiente:** Eficiencia en el proceso de control remoto de computadoras en tiempo real.

**Posibles resultados:**

- Aplicación web desarrollada con el marco de trabajo jWebSocket que garantice mayor nivel de eficiencia en el proceso de control remoto de computadoras.
- Informe detallado con toda la base teórico-práctica sobre la cual se sustenta la solución propuesta.

Para una mejor comprensión de la investigación, el contenido ha sido desglosado en tres capítulos, además de las conclusiones generales, recomendaciones, referencias bibliográficas y bibliografía utilizada, glosario de términos en el cual se detallan los términos técnicos y poco claros utilizados en la elaboración del documento, y los anexos que complementan el trabajo realizado.

Los capítulos han sido estructurados de la siguiente manera:

**Capítulo 1.** Fundamentación Teórica: Se realiza la fundamentación teórica de la investigación. Se expone un estudio del estado del arte de las aplicaciones Web para el control remoto de computadoras en la actualidad, tanto a nivel nacional como internacional.

**Capítulo 2.** Características, Análisis y Diseño del Sistema: Brinda una fundamentación de la solución propuesta, a partir de la cual se describen las actividades de análisis de la solución, seguidas por la descripción de los procesos del sistema y de la etapa de diseño.

**Capítulo 3.** Implementación y Validación del Sistema: Se describe la etapa de implementación que conlleva a la obtención del software. Se elaboran y documentan las pruebas funcionales realizadas, la certificación de calidad de software y la valoración del cliente sobre la aplicación propuesta.

# CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

## Introducción

El presente capítulo tiene como objetivo, tratar los principales conceptos y aspectos más significativos relacionados con las principales temáticas abordadas en diferentes fuentes bibliográficas acerca del estado del arte que presenta el tema que se investiga. Se proporciona una panorámica general acerca del control remoto de computadoras y en específico del control remoto de computadoras en tiempo real. También se analiza la metodología, tecnologías y herramientas utilizadas para el desarrollo de la solución, siempre teniendo en cuenta la necesidad del uso de herramientas y marcos de trabajo de código abierto.

### 1.1. Fundamentos Teórico-Methodológicos

El auge de las aplicaciones web en los últimos años ha remodelado el uso de Internet en la actualidad. El surgimiento de la Web 2.0 les dio a los usuarios un mayor protagonismo, pero la necesidad de aumentar la velocidad de comunicación entre ellos ha dejado implantado frenos en el desarrollo de este tipo de aplicaciones. Al mismo tiempo la experiencia de los usuarios en el uso de las aplicaciones web exige el surgimiento de una comunicación en tiempo real.

La definición de **tiempo real** en la Web dada por el autor Phil Wolff establece que la Web en tiempo real es rápida, fluida, torrencial y sin fronteras. Rápido significa que la visión del mundo alrededor es siempre fresca. Fluida significa que la información viene a gotas en lugar de a cubos. Torrencial ya que el volumen de cambios es abrumador, sin filtros y sin porteros. Sin fronteras para que la información, los amigos y la experiencia llegue desde todas partes de internet. Web en tiempo real cambia el modo de sentir la web, se hace más inmediato, interpersonal, completo y humano. (Wolff, 2009)

También contamos con la definición de los autores Lambert M. Surhone, Mariam T. Tennoe y Susan F. Henssonow. Ellos plantean que el tiempo real en la Web se brinda a través de 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 actualizaciones periódicamente. (Surhone, y otros, 2010).

Otro punto de vista es el de Alexander Schulze, para el cual el tiempo real en la Web se posibilita cuando el cliente recibe mensajes del servidor sin solicitud previa y los usuarios finales reciben actualizaciones de forma simultánea. (Framework Approach for WebSockets, 2011). Para esta investigación, a partir de los conceptos dados anteriormente, se asume la definición de tiempo real en la web dada por Alexander Schulze ya que es el líder internacional del proyecto `jWebSocket` y presenta elementos importantes para el objetivo de la investigación.

Lograr tiempo real en la Web mediante el protocolo HTTP es posible, sin embargo se logra mediante técnicas y tecnologías que aumentan considerablemente el uso de los recursos de la red, esto es debido a que no fue ese el objetivo para el cual fue diseñado. El lanzamiento de la nueva versión del lenguaje HTML da soporte al protocolo `Websocket`. Según Petter Lubbers, **Websocket** es una tecnología que proporciona un canal de comunicación bidireccional y *full-duplex* sobre un único socket TCP (Protocolo de Control de Transmisión). Este protocolo soluciona las limitaciones del protocolo HTTP, al establecer una comunicación *full-duplex* (TCP) entre el cliente y el servidor, sustituyendo la comunicación *half-duplex* (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. (HTML5 Websockets and communication, 2010)

Otro concepto define que `Websocket` es un mecanismo de handshake seguido de intercambios de mensajes sobre la capa TCP. El objetivo de esta tecnología es proveer un mecanismo para aplicaciones basadas en navegadores que necesitan comunicación bidireccional con el servidor en vez de tener que realizar múltiples conexiones HTTP. (Hybi, 2011)

Según la IETF (Internet Engineering Task Force) `Websocket` permite la comunicación bidireccional entre un cliente que ejecuta código no confiable en un ambiente controlado y un anfitrión remoto que ha optado por comunicaciones

desde ese código. El modelo de seguridad utilizado para ello es el modelo de seguridad basado en el origen comúnmente utilizado por los navegadores web. Este protocolo consiste en un apretón de manos de apertura, seguida por la estructuración en capas del mensaje básico a través de TCP. El objetivo de esta tecnología es proporcionar un mecanismo para aplicaciones basadas en navegadores que necesitan una comunicación bidireccional con los servidores que no dependen de la apertura de múltiples conexiones HTTP. (Fette, y otros, 2011)

Para esta investigación, se asume la definición de WebSocket dada por Petter Lubbers ya que plantea elementos concretos. Otro concepto importante relacionado con el estudio de la investigación es la definición de **computadora remota**. Durante el estudio se analizaron distintos conceptos asociados a este término. Finalmente se asume la definición emitida por Nadia Journey donde establece que una computadora remota es una computadora común que posee un programa informático funcionamiento de forma regular que permite que el equipo pueda ser controlado a distancia, es decir sin la necesidad de acceso físico. El software le permite controlar el ordenador remoto, y usted puede hacer todas las tareas normales con el ordenador como si fuera delante de usted. Esta selección viene dada porque incluye elementos más completos enfocados al objetivo de la investigación. (Journey, 2011)

## **1.2. Análisis de Soluciones Existentes**

Poco a poco las aplicaciones web han ido cobrando terreno con respecto a las llamadas aplicaciones de escritorio. El vertiginoso desarrollo de Internet favoreció el crecimiento de las nuevas aplicaciones permitiendo que se crearan para un sinfín de escenarios. Esto dio lugar a que las aplicaciones web incorporaran las funcionalidades de las de escritorio. Las consolas de los diferentes sistemas operativos no se han quedado fuera en este proceso de migración y con el pasar de los años han surgido numerosas soluciones que permiten el control remoto a través de la Web simulando una consola del sistema. A continuación se presentan las principales herramientas analizadas en la investigación que garantizan el control remoto de computadoras en la Web.

**HTRSH** (Consola Remota de Híper Texto) es similar a una consola remota pero que trabaja en la Web usando el protocolo HTTP o HTTPS para la transferencia de datos. Está destinado para usuarios que no puedan acceder a alguna máquina que esté conectada a Internet. La razón de esto puede ser falta de disponibilidad de acceso externo a Internet, las normas estrictas del firewall, NAT (Traductor de Direcciones de Red) y entre otras. HTRSH permite administrar un servidor a través de su navegador web desde cualquier lugar.

La aplicación HTRSH consiste en dos aplicaciones que trabajan en conjunto. La parte del servidor y la parte de la interfaz. Para su uso se debe configurar y ejecutar el servidor en el host que el usuario va a acceder. Por ejemplo, puede ser el ordenador de casa o trabajo. El programa se está desarrollando y no ha sido liberado todavía.

El servidor es una aplicación multiplataforma que está siendo escrita con la norma ANSI C usando la librería multi-plataforma cURL (Cygwin para la versión Windows). El cliente es una aplicación multiplataforma escrita con PHP y la biblioteca Prototype de Java Script haciendo un amplio uso de AJAX para mejorar la capacidad de respuesta.

Otra solución similar es **Web Console**, que como su nombre indica es una aplicación basada en la web diseñada para ejecutar comandos en servidores con sistemas UNIX o Windows. También permite subir o descargar archivos del servidor y editar directamente archivos de texto.

Este software es de código abierto y está escrito en el lenguaje de programación Perl usando técnicas AJAX. Es muy ligera y no requiere de ninguna Base de Datos para su funcionamiento.

Por otra parte, **consoleFISH** pretende eliminar las restricciones impuestas por los firewalls o proxy para manejar un servidor SSH. Esta herramienta web está diseñada para que los administradores de los servidores puedan hacer resguardo de los datos de manera remota. Una gran desventaja de este sistema es que la información que se maneja viaja a través de la red de manera insegura,

permitiendo que teóricamente se pueda leer. Otra inconveniente es que a pesar de ser un producto de código abierto, la creación explícita de una cuenta de usuario personal para su uso debe ser comprada.

De manera general todas las aplicaciones web destinadas para el control remoto de computadoras presentan la dificultad que no trabajan en tiempo real. Una de las razones es el uso del protocolo HTTP para la transferencia de datos entre el cliente y el servidor. A pesar de usar técnicas avanzadas de comunicación como AJAX aún no logran satisfacer los requisitos de los usuarios de la ya inminente Web 3.0. La sobrecarga de peticiones al servidor y el ancho de banda utilizado son factores que también propician este resultado.

Por lo tanto se hace necesario ir en la búsqueda de una investigación que logre realizar el control remoto de computadoras en tiempo real a través de la web. Esto será posible gracias al surgimiento del protocolo WebSocket, el cual con su estándar de comunicación bidireccional viene a resolver las limitantes del antes mencionado HTTP.

### **1.3. Metodología usada para el desarrollo de la solución**

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 - RUP**

RUP (Proceso Unificado de Desarrollo) fue desarrollado por Rational Software, 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.

En esta metodología se pueden identificar 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 llevar a obtener la capacidad operacional inicial.
- Transición: el objetivo es llegar a obtener el despliegue del proyecto.

Las metodologías robustas centran su atención en llevar una documentación exhaustiva y en cumplir con un plan de proyecto. Otra característica importante dentro de este enfoque son los altos costos al implementar un cambio, por lo cual no ofrecen una buena solución para proyectos donde el entorno es volátil. (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)

Los beneficios de usar SCRUM como metodología se ve reflejada tanto en el lado del cliente como en el de los desarrolladores. Para los clientes ya que los pone contentos porque recibieron lo que necesitaban y esperaban y para los desarrolladores por la satisfacción de haber realizado el trabajo de una manera eficiente.

Una de las principales ventajas del modelo SCRUM es su flexibilidad y capacidad de adaptación tanto como cambian las necesidades del trabajo. Proporciona mecanismos de control para la planificación de una versión del producto. Esto significa que el proyecto puede ser alterado y modificado en función de las necesidades cambiantes, y al final se las arregla para establecer y ofrecer la versión más adecuada, que surge de la posibilidad de ajustar el trabajo a las cambiantes expectativas una vez que el proyecto está en marcha. (SCRUM methodology, 2009)

### **Programación Extrema - XP**

XP (Programación Extrema) 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)

XP mejora un proyecto de software en cinco aspectos esenciales, la comunicación, la sencillez, la retroalimentación, el respeto y el coraje. Los programadores están en constante comunicación con sus clientes y colegas programadores. Mantienen su diseño simple y limpio. Reciben retroalimentación probando su software a partir del primer día. Entregan el sistema a los clientes tan pronto como sea posible y ponen en práctica los cambios como se sugiere. Cada pequeño éxito profundiza su respeto por las contribuciones únicas de los miembros del equipo.

## **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.

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

Teniendo en cuenta las ventajas que ofrecen las metodologías ágiles y que no existe una metodología universal que indique cómo crear todo tipo de software se

propone para el desarrollo de este trabajo el uso de SXP ya que incorpora las mejores características de las metodologías XP y SCRUM. A la hora de construir una aplicación se debe escoger la metodología a utilizar teniendo en cuenta las características, complejidad, envergadura del proyecto y el tipo de contrato establecido para el mismo.

#### **1.4. Herramientas y Tecnologías usadas para el desarrollo de la solución**

##### **1.4.1. Lenguajes de Programación y Modelado**

###### **Lenguaje de Modelado Unificado - UML**

UML (Lenguaje Unificado de Modelado) es el lenguaje de modelado seleccionado para el desarrollo de la presente investigación debido a que la metodología seleccionada fue SXP la cual lo define como estándar para el modelado. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. 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. Se decide utilizar UML como lenguaje de modelado en su versión 2.0.

###### **Java**

El lenguaje Java se utiliza para desarrollar la aplicación en el lado del servidor. El marco de trabajo jWebSocket está desarrollado con dicho lenguaje por lo cual se asume para el desarrollo de 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.



Java tiene la asignación de memoria fuerte y un mecanismo automático de recolección de basura. Proporciona la gestión de excepciones de gran alcance y el mecanismo de comprobación de tipos. El compilador comprueba si el programa contiene algún error y el intérprete comprueba cualquier error en tiempo de ejecución y hace que el sistema sea seguro contra accidentes. Todas las características anteriores hacen que el lenguaje Java sea muy robusto. Los protocolos más utilizados, como HTTP y FTP se han desarrollado en Java.

### **JavaScript**

Actualmente solo se encuentran disponibles para la comunicación con el servidor jWebSocket clientes escritos en los lenguajes Java y JavaScript. Debido a que la solución está basada en la Web se decidió utilizar el lenguaje JavaScript para implementar el lado del cliente.

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. (Eguíluz Pérez, 2009)

### **1.4.2. Otros Lenguajes**

#### **Lenguaje de Marcado de Hipertexto - HTML**

Para el desarrollo de la aplicación se utiliza el lenguaje de marcado HTML (HyperText Markup Language), el cual es el lenguaje más utilizado para la creación de páginas web. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas

aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. (Eguíluz Pérez, 2009)

## **Hojas de Estilo en Cascada - CSS**

CSS (Hojas de estilo en cascada) 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)

### **1.4.3. 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 jWebSocket tiene una arquitectura de plug-in que permite aumentar con facilidad sus funcionalidades.

El servidor jWebSocket está diseñado para funcionar como servidor de comunicaciones o como servidor web, brindando total flexibilidad.

jWebSocket 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)

jWebSocket 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 jWebSocket, precisamente por esta razón la aplicación a desarrollar se lleva a cabo sobre este marco de trabajo.

#### **1.4.4. Marcos de Trabajo del lado del Cliente**

##### **Marco de Trabajo jQuery**

jQuery es una biblioteca o marco de trabajo 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 de 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. 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)

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Además posibilita la interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un componente básico de XPath. Posee varias utilidades para obtener información del navegador, operar con objetos y vectores, funciones, entre otras. Es compatible con los navegadores Mozilla

Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.  
(Proyecto jQuery, 2010)

### **Ventajas:**

- Como buen JavaScript se puede trabajar separado del HTML, haciéndolo más fácil de mantener y enriquecer, aumentando la productividad del proyecto.
- Se considera su sintaxis liviana, teniendo en cuenta que la sencillez y poca extensión de código es fundamental para los desarrollos.
- Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del marco de trabajo.
- Tiene una comunidad de creadores de componentes muy amplia, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar interfaces de usuario, galerías, votaciones, efectos diversos, entre otros.

(jQuery, la librería Javascript por excelencia. Un framework Javascript lleno de ventajas, 2011)

Existen otros marcos de trabajo JavaScript como Moo Tools, Prototype, YUI y Ext JS pero para la presente investigación se decide utilizar jQuery como marco de trabajo del lado del cliente ya que tiene una amplia gama de complementos disponibles para cubrir necesidades específicas y para esta investigación se necesitan utilizar algunos de ellos. Además provee de un excelente mecanismo para la captura de eventos, soporta una gran cantidad de selectores como los CSS y permite agregar o quitar fácilmente, los atributos de cualquier elemento HTML.

### **1.4.5. 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.

## 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. Dicha herramienta brinda una serie de facilidades que se mencionan a continuación:

- Soporta un conjunto de estándares entre los que se encuentran UML, SysML, BPMN, XML y XMI.
- Soporte de modelado UML, modelado de procesos de negocios y un generador de mapeo de objetos-relacionales para los lenguajes de programación Java, .NET y PHP.
- Integración con herramientas Java (Eclipse/IBM WebSphere ,JBuilder , NetBeans IDE , Oracle JDeveloper y BEA Weblogic).
- Permite la generación de código y la ingeniería inversa para un conjunto de lenguajes entre los que se encuentran Java, C++, CORBA IDL, PHP, XML Schema, Ada y Python. Cualquiera de los cambios en el código existente puede reflejarse en el modelo y viceversa.
- Ofrece herramientas para la generación de reportes en formatos html, pdf y doc.
- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- Interoperabilidad e integración. Permite la integración con un conjunto de herramientas (Visio drawing, Rational Rose, ERwin Data Modeler Project, Microsoft Excel y Microsoft Word document) e intercambiar diagramas UML y modelos usando representaciones industriales comunes.
- Modelado de base de datos. Proporciona una mayor documentación de la

base de datos y diagramas de mapeo de relación de objetos.

- Integración con herramientas para el control de versiones.
- Diseño de prototipo de Interfaz de Usuario. Permite insertar información adicional a los diagramas mediante notas y comentarios para describir sus elementos lo que facilita la revisión de los prototipos así como el trabajo en equipo.

(Visual Paradigm International Ltd, 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.

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.

#### **1.4.6. 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. A continuación se hace una caracterización de algunos sistemas de control de

versiones con el objetivo de determinar el más idóneo para garantizar mayor seguridad y disponibilidad de los datos.

### **Subversión**

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)

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. Posee soporte para desarrolladores y desarrollo paralelo.

Teniendo en cuenta las características expuestas y después de haber analizado otras herramientas de control de versiones, se selecciona Subversion como herramienta de control de versiones para ser utilizado en el desarrollo de la aplicación, ya que permite la integración con NetBeans y la posibilidad de desarrollar en paralelo. También influyó en la decisión, la experiencia de trabajo por parte del equipo de desarrollo con esta herramienta y su fácil uso.

#### **1.4.7. 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.

## **RapidSVN**

Es una plataforma de interfaz gráfica de usuario, para el sistema de revisión de Subversion. Este proyecto también incluye un cliente de Subversion C + + API. 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)

Se caracteriza por ser:

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

Se analizaron algunas de las herramientas, como por ejemplo Tortoise y Rapidsvn, a pesar de todas las ventajas y funcionalidades que ofrece Tortoise, se decide usar Rapidsvn ya que ésta es libre, y fácil de usar, tanto por quienes ya conocen Subversion como para quienes empiezan, pudiendo acceder a direcciones SVN, subir y descargar contenido, sincronizarlo con el servidor original, comprobar su estado, crear y fusionar direcciones. Puede ser ejecutado en cualquier plataforma.



#### **1.4.8. Entorno Integrado de Desarrollo - IDE**

Un IDE (Entorno Integrado de Desarrollo) 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.

##### **NetBeans**

NetBeans IDE es una herramienta desarrollada por Sun Microsystems. Está completamente escrito en Java, por lo que puede ser utilizado desde cualquier sistema operativo compatible con la máquina virtual de Java. Permite el desarrollo de aplicaciones de escritorio, web y móviles. Brinda soporte a varios lenguajes de programación como Java, PHP, C/C++, Groovy, Python, JavaScript, entre otros. Es un producto libre y gratuito sin restricciones de uso. 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)

##### **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.

Las características anteriormente expuestas sobre NetBeans IDE y Eclipse SDK demuestran las potencialidades de ambos para el desarrollo de aplicaciones Java. Sin embargo, para el desarrollo de la solución propuesta se selecciona como entorno integrado de desarrollo a NetBeans IDE debido a que se tiene una mayor experiencia y familiarización con esta herramienta. Además su versión 7.0.1 introduce un soporte para el desarrollo con la especificación JavaSE7 (Java Standard Edition) con las características de JDK7(Java Development Kit).

### **Conclusiones del capítulo**

En el presente capítulo se abordaron los conceptos fundamentales que orientan la presente investigación. Se consolidó el basamento teórico mediante un análisis de la actualidad de las aplicaciones web de control remoto de computadoras, evidenciándose las dificultades de estas. Además se muestran los resultados de un estudio realizado con el objetivo de seleccionar las herramientas, tecnologías y metodología a utilizar, permitiendo sentar las bases para el desarrollo de la aplicación web que posibilita el control remoto de computadoras en tiempo real usando el marco de trabajo `WebSocket`. Se definió la metodología SXP, así como los lenguajes de programación Java del lado del servidor y JavaScript del lado del cliente. Se seleccionaron las herramientas NetBeans, Visual Paradigm y Subversion y los marcos de trabajo `WebSocket` y `jQuery` para el desarrollo de la solución.

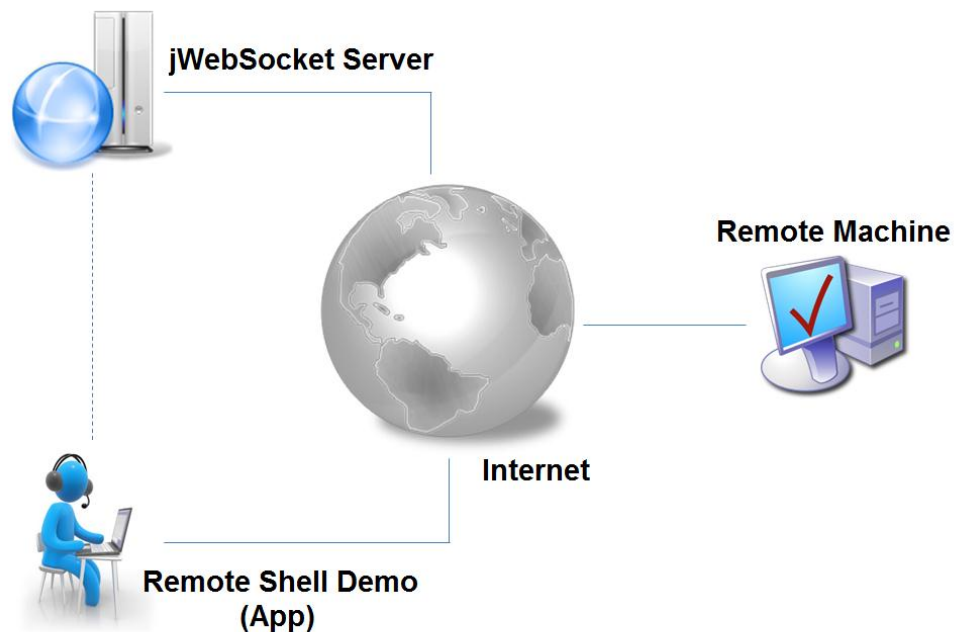
## CAPÍTULO 2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA

### Introducción

En el presente capítulo se describe cómo debe funcionar la aplicación web, destacando sus características distintivas. Se realiza el modelado del dominio con el objetivo de comprender su contexto. Además se detallan brevemente los artefactos de la metodología de desarrollo SXP, propuesta en el capítulo anterior. Describiéndose 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

El control remoto de computadoras a través de la Web no se realiza en tiempo real. Por esta razón se desarrolla una aplicación web utilizando el marco de trabajo jWebSocket que garantice mayor nivel de eficiencia en el proceso de control remoto de computadoras en tiempo real. A continuación en la Figura 1 se presenta un diagrama de la solución propuesta:



**Fig.1** Propuesta de Solución. Fuente: Elaboración propia.

La aplicación denominada Remote Shell App, permite a los usuarios realizar disímiles configuraciones en el sistema de archivos y tareas de monitorización a una computadora remota. Utiliza una interfaz web, la cual hace que sea independiente al sistema operativo desde donde va a ser utilizada. A diferencia de otras aplicaciones similares de control remoto de computadoras que utilizan técnicas avanzadas para simular tiempo real en la Web, esta trabaja en tiempo real debido a que está basada en el protocolo de comunicación WebSocket. Se desarrolla sobre el marco de trabajo jWebSocket, permitiendo que el usuario pruebe el potencial de jWebSocket para realizar aplicaciones de control remoto en tiempo real.

Una vez autenticado el usuario con el servidor jWebSocket, podrá insertar los datos para conectarse con la maquina remota, personalizar la consola para luego interactuar con ella y ejecutar los comandos que le permitan realizar sus tareas. La aplicación es accesible desde cualquier dispositivo, una computadora personal, una laptop o un teléfono celular.

La interfaz principal de la aplicación está diseñada para ser amigable y de fácil uso para cualquier usuario, respetando los patrones de diseño establecidos para el proyecto. Se pretende desarrollar las funcionalidades necesarias para que el trabajo con esta herramienta sea cómodo y confortable para el usuario. La aplicación eleva los niveles de eficiencia en el proceso de control remoto de computadoras ya que le da rapidez y fluidez a la actividad.

## **2.2. Planificación del Proyecto por Roles**

La selección de los roles que intervienen en el ciclo de desarrollo de la solución propuesta es vital para la organización de las responsabilidades y mantener una buena comunicación entre los miembros del proyecto. A continuación se muestran en la Tabla 1 los roles planificados y sus responsabilidades:

**Tabla 1:** Planificación del Proyecto por Roles

Rol	Responsabilidad	Nombre
-----	-----------------	--------

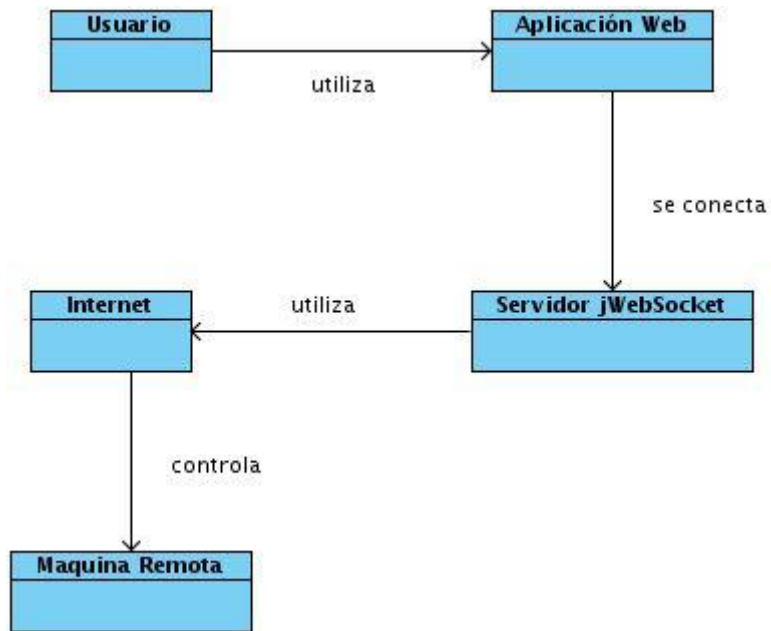
Líder del Proyecto	Debe asegurar que el proyecto se está llevando a cabo de acuerdo con las prácticas y que todo funciona según lo planeado. Su principal trabajo es remover impedimentos y reducir riesgos del producto. Coordina y facilita las reuniones. Asegura que se consiguen los objetivos de cada iteración.	Yamila Vigil Regalado
Gerente	Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto. Participa en la selección de objetivos y requerimientos. Tiene la responsabilidad de controlar el progreso y trabaja junto con el Jefe de Proyecto en la reducción de la Lista de Reserva del Producto.	Alexander Schulze
Especialista (Diseñador Gráfico)	Conoce a fondo el proceso para el desarrollo de software. Es una especialización que está activa, el miembro del grupo de trabajo que la desempeña siempre está ejecutándola y alcanzando un grado mayor de conocimientos en el tema, en este caso como Diseñador	Rebecca Schulze

	Gráfico.	
Consultores	Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas, además aportan ideas y experiencias para el beneficio del sistema en desarrollo. Esta es una especialización menos activa, quien la ejecuta funciona en este rol por un corto período de tiempo.	Alexander Schulze Rebecca Schulze
Cliente	Participa en las tareas que involucran la lista de reserva del producto.	Alexander Schulze
Programador	Elabora el código de las nuevas funcionalidades a implementar. Escribe las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y el resto del equipo.	Yasmany Nuñez Broch
Analista	Escribe las historias de usuario y las pruebas funcionales para validar su implementación.	Yasmany Nuñez Broch
Diseñador	Encargado del diseño del sistema y de los prototipos de interfaces, son los máximos	Yasmany Nuñez Broch

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

### 2.3. Modelo de Dominio

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, así como la forma en que interactúa el sistema con los clientes y viceversa. Pero si dicho negocio no está bien definido entre los clientes y los ejecutores del proyecto; entonces es generado el llamado Modelo de Dominio del sistema que se propone a continuación:



**Fig. 2** Modelo de Dominio. Fuente: Elaboración propia.

**Usuario:** Persona que desea autenticarse en la aplicación web para manejar la computadora remota.

**Aplicación Web:** Se refiere a la aplicación destinada para servir de intermediaria entre el usuario y la computadora remota.

**Servidor jWebSocket:** Controla las acciones emitidas por el cliente y maneja la lógica de respuesta de las operaciones lanzadas.

**Internet:** Medio mediante el cual el servidor jWebSocket se comunica con la computadora remota.

**Máquina Remota:** Dispositivo que desea ser controlado a distancia utilizando una interfaz web.

## 2.4. Lista de Reserva del Producto (LRP)

Una actividad importante definida en la metodología SXP es la Lista de Reserva del Producto ya que recoge de manera priorizada todo el trabajo a desarrollar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son suficientes para una iteración. Esta lista puede crecer y



modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente. Con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto. Puede estar conformada por requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas.

**Tabla 1:** Lista de Reserva del Producto

Prioridad	Ítem *	Descripción	Estimación	Estimado por
<b>Muy Alta</b>				
	1	Permitir conectarse remotamente a una PC	3 semanas	Analista
	2	Permitir ejecutar comandos	3 semanas	Analista
	3	Mostrar resultados en el lado del cliente	3 semanas	Analista
<b>Alta</b>				
<b>Media</b>				
	4	Seleccionar color de fondo.	1 semana	Analista
	5	Seleccionar estilo para el texto.	1 semana	Analista
<b>Baja</b>				
<b>RNF (Requisitos No Funcionales)</b>				
	1	La aplicación está dirigida a usuarios con conocimientos		Analista

		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.		
	2	El Sistema es una aplicación web		Analista
	3	El sistema se mantendrá disponible 24 horas diarias durante los 7 días de la semana.		Analista
	4	El mantenimiento de la aplicación se realizará en tiempo de ejecución sin afectar la disponibilidad ni el rendimiento de los servicios.		Analista
	5	Garantizar la integridad y consistencia de los datos.		Analista
	6	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.		Analista
	7	Se documentará la aplicación		Analista

		con diferentes manuales con el objetivo de explicar el uso de la aplicación para garantizar el soporte de la misma.		
	8	En el lado del servidor el lenguaje de programación a utilizar es Java, empleando el Framework jWebSocket y el IDE NetBeans 7.0.1.		Analista
	9	En el lado del cliente se hará uso de los lenguajes JavaScript, HTML y CSS, empleando el Framework jQuery y el IDE NetBeans 7.0.1.		Analista
	10	La metodología de desarrollo a seguir es SXP y para la modelación se utilizará la herramienta Visual Paradigm 3.4.		Analista
	11	Para el desarrollo de la aplicación en el lado del cliente se utilizará una arquitectura en capas MVC.		Analista
	12	Los requisitos mínimos de hardware que tiene que tener el cliente para el correcto funcionamiento de la		Analista

		aplicación son: 512 MB de memoria RAM, microprocesador Pentium IV, tarjeta de red 100 mbps.		
	13	El cliente debe contar con un navegador que soporte el protocolo WebSocket.		Analista
	14	El servidor de aplicaciones debe tener instalado la máquina virtual de Java OpenJDK 7 y el servidor de jWebSocket.		Analista
	15	El código de la aplicación será liberado bajo la Licencia Pública General Reducida de GNU (LGPL).		Analista
	16	El código y la documentación de la aplicación deberán cumplir estrictamente las normas de calidad estipuladas por el grupo de Calisoft de la UCI.		Analista

## 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. Se escriben con un lenguaje natural y con palabras concisas para no exceder su tamaño en unas pocas líneas de texto. Van a ser la guía para la construcción posterior de las pruebas de aceptación comprobando de esta manera la correcta implementación de las historias de

usuario. A continuación se dan a conocer las historias de usuario que están presentes en el sistema y las tareas de ingeniería asociadas a cada de ellas.

1. Ejecutar conexión remota

<b>Historia de Usuario</b>	
<b>Número:</b> HU_1	<b>Nombre Historia de Usuario:</b> Ejecutar conexión remota.
<b>Modificación de Historia de Usuario Número:</b> ninguna	
<b>Usuario:</b> Yasmany Nuñez Broch	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Muy Alta	<b>Puntos Estimados:</b> 3 semanas
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 3 semanas
<b>Descripción:</b> La presente historia de usuario tiene como objetivo permitir que el usuario pueda conectarse a computadoras de manera remota a través de la web en tiempo real.	
<b>Observaciones:</b> Para que esto sea posible el usuario debe autenticarse con el usuario del host al cual quiere manejar de forma remota.	
<b>Prototipo de interface:</b>  Ver Anexo 1	

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1.1	<b>Número Historia de Usuario:</b> HU_1
<b>Nombre Tarea:</b> Investigar las maneras de conectarse remoto a una PC.	
<b>Tipo de Tarea :</b> Investigación	<b>Puntos Estimados:</b> 2 días
<b>Fecha Inicio:</b> 01/11/2011	<b>Fecha Fin:</b> 02/11/2011

<b>Programador Responsable:</b> Yasmany Nuñez Broch
<b>Descripción:</b> La tarea tiene como objetivo realizar un estudio sobre el proceso de cómo conectarse remoto a una PC.

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.2	<b>Número Historia de Usuario:</b> HU_1
<b>Nombre Tarea:</b> Conectarse remotamente a una PC.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 13 días
<b>Fecha Inicio:</b> 03/11/2011	<b>Fecha Fin:</b> 21/11/2011
<b>Programador Responsable::</b> Yasmany Nuñez Broch	
<b>Descripción:</b> Implementar los métodos que gestionan la conexión con otras computadoras. El usuario introduce los datos necesarios para autenticarse en la maquina remota para establecer la comunicación con la misma.	

## 2. Ejecutar comandos

Historia de Usuario	
<b>Número:</b> HU_2	<b>Nombre Historia de Usuario:</b> Ejecutar comandos
<b>Modificación de Historia de Usuario Número:</b> ninguna	
<b>Usuario:</b> Yasmany Nuñez Broch	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Muy Alta	<b>Puntos Estimados:</b> 3 semanas
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 3 semanas
<b>Descripción:</b> La presente historia de usuario tiene como objetivo permitir que el	

usuario pueda ejecutar comandos en el host al cual se conectó previamente.
<b>Observaciones:</b> ninguna
<b>Prototipo de interface:</b> ninguna

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.1	<b>Número Historia de Usuario:</b> HU_2
<b>Nombre Tarea:</b> Investigar los nombres y los tipos de comandos que existen en los diferentes sistemas operativos.	
<b>Tipo de Tarea :</b> Investigación	<b>Puntos Estimados:</b> 2 días
<b>Fecha Inicio:</b> 22/11/2011	<b>Fecha Fin:</b> 23/11/2011
<b>Programador Responsable:</b> Yasmany Nuñez Broch	
<b>Descripción:</b> La tarea tiene como objetivo realizar un estudio sobre los diferentes tipos de comandos que existen en los sistemas operativos actuales más usados como UNIX, Windows y MacOS.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.2	<b>Número Historia de Usuario:</b> HU_2
<b>Nombre Tarea:</b> Ejecutar comandos	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 13 días
<b>Fecha Inicio:</b> 24/11/2011	<b>Fecha Fin:</b> 12/12/2011
<b>Programador Responsable::</b> Yasmany Nuñez Broch	
<b>Descripción:</b> Implementar las funcionalidades que permiten la ejecución de los comandos enviados desde el cliente. El usuario debe teclear los comandos, los	

cuales serán enviados al servidor, el cuál se encarga de ejecutarlos en la maquina remota.

### 3. Mostrar resultados

<b>Historia de Usuario</b>	
<b>Número:</b> HU_3	<b>Nombre Historia de Usuario:</b> Mostrar resultados.
<b>Modificación de Historia de Usuario Número:</b> ninguna	
<b>Usuario:</b> Yasmany Nuñez Broch	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Muy Alta	<b>Puntos Estimados:</b> 3 semanas
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 3 semanas
<b>Descripción:</b> La presente historia de usuario tiene como objetivo mostrar los resultados de la ejecución de los comandos en la maquina remota. De la misma manera tener actualizada la interfaz del usuario.	
<b>Observaciones:</b> ninguna	
<b>Prototipo de interface:</b>  Ver Anexo 2	

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 3.1	<b>Número Historia de Usuario:</b> HU_3
<b>Nombre Tarea:</b> Investigar cómo funciona jQuery.	
<b>Tipo de Tarea :</b> Investigación	<b>Puntos Estimados:</b> 2 días
<b>Fecha Inicio:</b> 13/12/2011	<b>Fecha Fin:</b> 14/12/2011



<b>Programador Responsable:</b> Yasmany Nuñez Broch
<b>Descripción:</b> La tarea tiene como objetivo realizar un estudio sobre el funcionamiento y estructura de la librería de JavaScript para realizar las labores de presentación en el lado del cliente.

Tarea de Ingeniería	
<b>Número Tarea:</b> 3.2	<b>Número Historia de Usuario:</b> HU_3
<b>Nombre Tarea:</b> Mostrar resultados en el lado del cliente.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 13 días
<b>Fecha Inicio:</b> 15/12/2011	<b>Fecha Fin:</b> 20/01/2012
<b>Programador Responsable::</b> Yasmany Nuñez Broch	
<b>Descripción:</b> Implementar el método que captura la respuesta enviada por el servidor y la visualiza en el navegador.	

#### 4. Ejecutar operaciones del middleware

Historia de Usuario	
<b>Número:</b> HU_4	<b>Nombre Historia de Usuario:</b> Configurar consola.
<b>Modificación de Historia de Usuario Número:</b> ninguna	
<b>Usuario:</b> Yasmany Nuñez Broch	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 3 semanas
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 3 semanas
<b>Descripción:</b> La presente historia tiene como objetivo proporcionar una serie de parámetros que el usuario pueda configurar, como el color de fondo, el tamaño	

del búfer y el estilo del texto.
<b>Observaciones: ninguna</b>
<b>Prototipo de interface:</b>  Ver Anexo 3

Tarea de Ingeniería	
<b>Número Tarea:</b> 4.1	<b>Número Historia de Usuario:</b> HU_4
<b>Nombre Tarea:</b> Investigar técnicas avanzadas en jQuery.	
<b>Tipo de Tarea :</b> Investigación	<b>Puntos Estimados:</b> 2 días
<b>Fecha Inicio:</b> 23/01/2012	<b>Fecha Fin:</b> 24/01/2012
<b>Programador Responsable:</b> Yasmany Nuñez Broch	
<b>Descripción:</b> La tarea tiene como objetivo realizar un estudio de las técnicas más avanzadas del marco de trabajo jQuery para manejar los elementos de una página web de manera óptima.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 4.2	<b>Número Historia de Usuario:</b> HU_4
<b>Nombre Tarea:</b> Seleccionar color de fondo.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 6 días
<b>Fecha Inicio:</b> 25/01/2012	<b>Fecha Fin:</b> 01/02/2012
<b>Programador Responsable::</b> Yasmany Nuñez Broch	
<b>Descripción:</b> Implementar la funcionalidad que le permite al usuario cambiar el	

color de fondo de la consola.
-------------------------------

Tarea de Ingeniería	
<b>Número Tarea:</b> 4.3	<b>Número Historia de Usuario:</b> HU_4
<b>Nombre Tarea:</b> Seleccionar estilo para el texto.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 7 días
<b>Fecha Inicio:</b> 02/02/2012	<b>Fecha Fin:</b> 10/02/2012
<b>Programador Responsable::</b> Yasmany Nuñez Broch	
<b>Descripción:</b> Implementar la funcionalidad que le permite al usuario cambiar el color de la fuente de la consola.	

## 2.6. Plan de Releases

En el ciclo de vida de un proyecto que usa la metodología SXP se define el plan de releases o iteraciones para realizar las entregas intermedias y la entrega final. El objetivo es mostrar la duración, prioridad y orden en que serán implementadas las historias de usuario dentro de cada iteración. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha historia. Para la solución se definieron 4 historias de usuarios divididas en 2 iteraciones, las cuales tienen una duración total de 12 semanas. Como resultado de la priorización de historias de usuario se llegó a la siguiente planificación:

**Tabla 3:** Plan de Releases

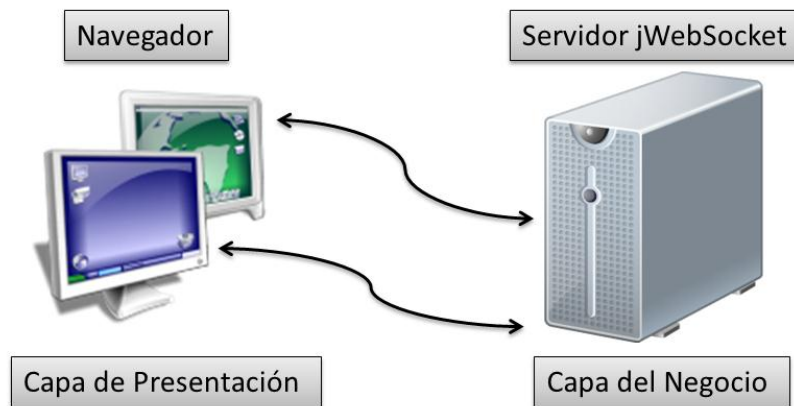
Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 2	En esta iteración se va a desarrollar una primera versión de la aplicación para la cual se	HU_1 HU_2	8 semanas

	implementarán todas las historias de usuario que constituyen funcionalidades fundamentales de la aplicación. Estas tienen prioridad muy alta.	HU_3	
Iteración 3	En esta iteración se va a desarrollar una segunda versión de la aplicación para lo cual se implementará la historias de usuario con categoría alta.	HU_4	4 semanas

## 2.7. Descripción de la Arquitectura de Software

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

Para desarrollar la aplicación Remote Shell App se utiliza una arquitectura en capas. La programación por capas es una arquitectura cliente-servidor que tiene como objetivo primordial la separación de la lógica del negocio de la lógica del diseño. Como podemos apreciar en la Fig. 3 se cuenta con 2 capas la de presentación y la capa del negocio.



**Fig. 3** Arquitectura de la Solución. Fuente: Elaboración propia.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y en caso de que sobrevenga algún cambio, sólo se afecta al nivel requerido sin tener que revisar entre código mezclado. Además distribuye el trabajo de creación de una aplicación por niveles, permitiendo que cada grupo de trabajo esté totalmente abstraído del resto de los niveles.

## 2.8. Diseño con Metáforas

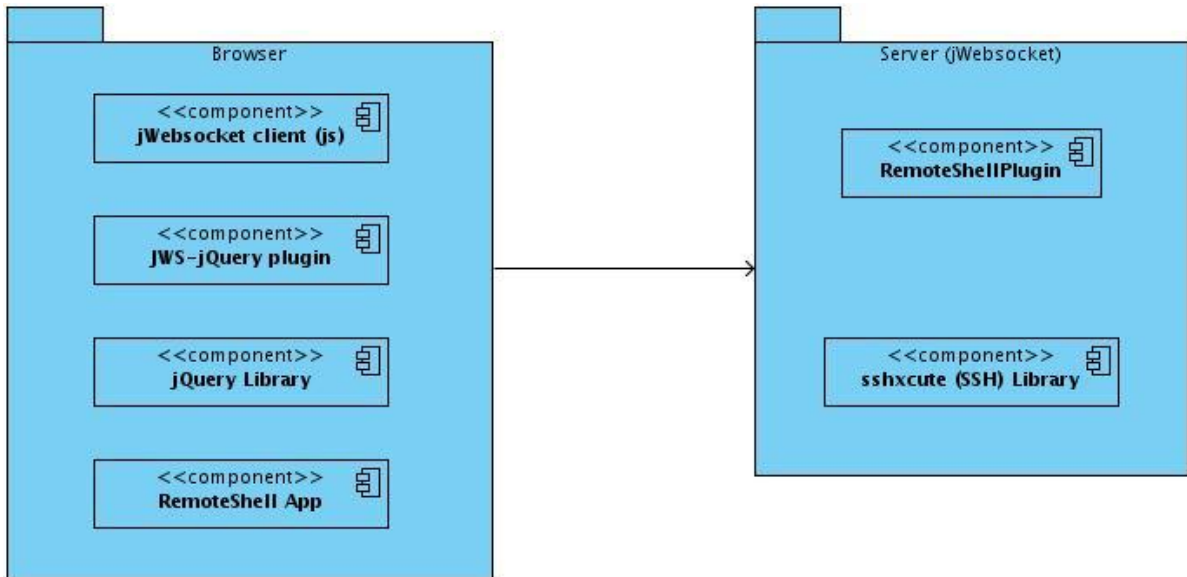
Debido a que SXP está basada en XP, y dicha metodología define un término llamado metáfora, lo cual según Martin Fowler es una historia compartida que describe como debería funcionar el sistema.

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 representa en la Fig. 3 el diagrama de paquetes para el sistema que se propone:



**Fig. 4** Diagrama de Paquetes. Fuente: Elaboración propia.

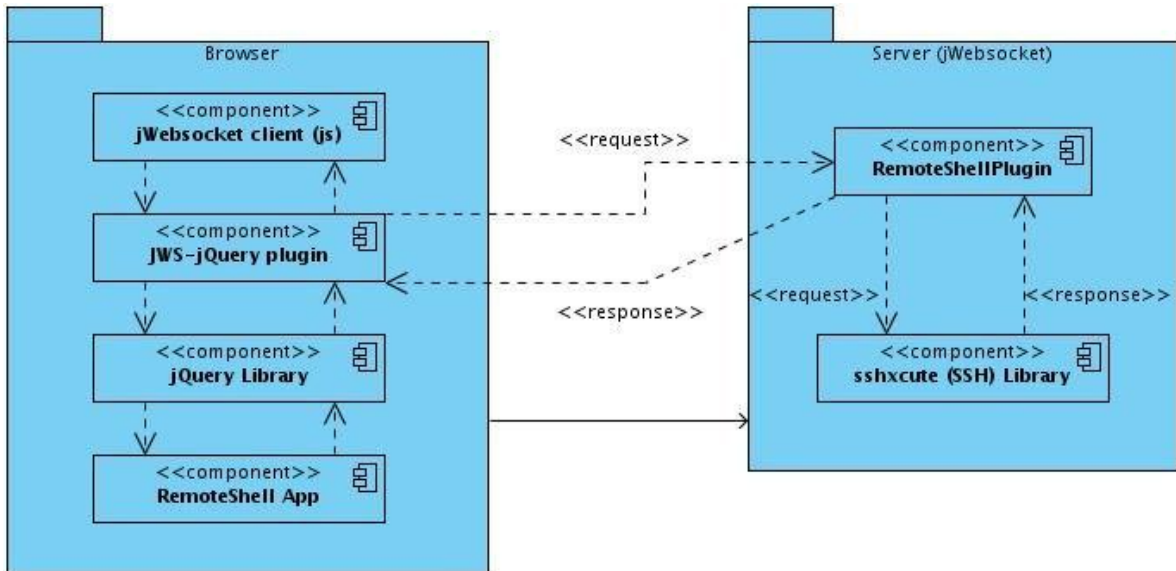
- En la capa del browser se tienen los componentes RemoteShell App que representa la aplicación, jWebSocket Client que constituye el cliente del marco de trabajo jWebSocket, la librería jQuery y el plugin de JWS-jQuery que permite conectarse con el servidor jWebSocket utilizando la librería jQuery. Esta capa del browser se relaciona con la capa del servidor de jWebSocket.
- La capa del servidor de jWebSocket contiene el plugin de la aplicación y la librería sshxcute que permite ejecutar los comandos que son enviados desde la capa del Browser.

## 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 el diagrama de componentes para el sistema que se propone:



**Fig. 5** Diagrama de Componentes. Fuente: Elaboración propia.

- El cliente debe contar con un navegador que soporte el protocolo websocket.
- La aplicación envía y recibe los datos a través del plugin JWS-jQuery el cual utiliza el cliente de jWebSocket escrito en JavaScript.
- El cliente de jWebSocket se comunica con su servidor por medio del protocolo WebSocket.
- El servidor de jWebSocket contiene el plugin de la aplicación el cual gestiona la comunicación entre el cliente y la maquina remota.
- El plugin de la aplicación en el servidor utiliza la librería sshxcute para ejecutar los comandos que vienen del cliente, obtiene los resultados de la ejecución de los mismos y los envía al cliente donde son finalmente mostrados.

## **Conclusiones del Capítulo**

En este capítulo se definieron las características y funcionalidades del sistema que se necesita desarrollar. Quedaron aprobados los requisitos funcionales y no funcionales necesarios para obtener un sistema eficiente, se describieron las historias de usuario y tareas de ingeniería que se deben implementar, el plan de releases para establecer el cronograma de trabajo y se realizó el modelado del diagrama de paquetes y de componentes que permiten una mejor comprensión de la solución propuesta.



## CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

### Introducción

En el presente capítulo se describe la etapa de implementación de la solución propuesta en el capítulo anterior, 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 se dan a conocer los resultados obtenidos hasta el momento y el aporte social y económico de la solución.

### 3.1. Implementación

La aplicación Remote Shell App está desarrollada sobre el marco de trabajo jWebSocket. Las librerías y clases clientes para la aplicación están disponibles solamente para el lenguaje JavaScript y el directorio donde se alojan los archivos del cliente se estructura de la siguiente manera:



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

- **jwsRemoteShell.js:** Permite manejar las funcionalidades de la consola con los datos que llegan del servidor.

Se hace uso además de archivos html y css para la interfaz de usuario de la aplicación:

- **jwsRemoteShell.html:** En este archivo está definido todo el código HTML de la aplicación.

- **jwsCharts.css:** Incluye el css correspondiente al HTML de la aplicación.

La librería utilizada en el lado del servidor es **sshxcute** la cual es utilizada para realizar conexiones remotas usando el protocolo SSH. Al mismo tiempo permite ejecutar comandos una vez establecida la conexión.

El método más significativo en el lado del servidor se describe a continuación:

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

En el lado del cliente:

- init: function(): Obtiene los elementos del DOM e inicializa variables auxiliares.
- registerEvents: function(): Maneja los eventos lanzados desde el servidor y controla el funcionamiento de las acciones.
- history: function(): Realiza el almacenamiento del historial de la consola.

### 3.2. Validación de la Investigación

Para validar la presente investigación que tiene como objetivo desarrollar una aplicación web utilizando el marco de trabajo jWebSocket que garantice mayor nivel de eficiencia en el proceso de control remoto de computadoras en tiempo real, se decide trazar una estrategia de validación basadas en las siguientes etapas:

En la primera etapa se realizan pruebas funcionales alrededor de las historias de usuario que garantizan el cumplimiento de los requisitos funcionales que debe cumplir la aplicación desarrollada.

Seguidamente se pasa a un proceso de certificación de calidad de software por parte del Grupo de Calidad del Centro de Desarrollo de la Facultad Regional "Mártires de Artemisa". Este grupo hace una certificación basándose principalmente en la funcionalidad, la estandarización, organización y limpieza del código de la aplicación. Además revisa los artefactos documentales que son generados por la metodología de desarrollo SXP seleccionada para la presente investigación.

Por último, se realiza una valoración de la aplicación por parte del cliente Alexander Schulze. En esta etapa de validación se persigue que el líder internacional de la comunidad jWebSocket certifique que la aplicación desarrollada pueda ser integrada al marco de trabajo y la real usabilidad que tiene esta para los usuarios potenciales y desarrolladores de la comunidad de jWebSocket.

### Pruebas Funcionales

Las pruebas funcionales son definidas por el equipo de desarrollo. La utilización de estas proporciona grandes ventajas, permitiendo a los programadores medir la calidad de su trabajo y garantizar la entrega de un producto con calidad y en correspondencia con las necesidades del cliente. Se definieron casos de prueba para todas las historias de usuario, 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 Remote Shell App.

- **Casos de pruebas para la HU Ejecutar conexión remota**

<b>Caso de Prueba</b>	
<b>Código Caso de Prueba:</b> jws-1-1	<b>Nombre Historia de Usuario:</b> Ejecutar conexión remota.
<b>Nombre de la persona que realiza la prueba:</b> Yasmany Nuñez Broch	
<b>Descripción de la Prueba:</b> Esta prueba consiste en verificar que el usuario pueda autenticarse con una maquina remota.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket.</li> <li>• El servidor de la aplicación tiene que estar ejecutándose.</li> <li>• El usuario debe estar autenticado con el servidor jWebSocket.</li> </ul>	
<b>Entrada / Pasos de ejecución:</b> Para establecer conexión con la maquina remota el usuario debe introducir su nombre, contraseña y la dirección que	

posee la maquina en la red. Luego presionar el botón “Login”.
<b>Resultado Esperado:</b> El usuario establece conexión con la maquina remota.
<b>Evaluación de la Prueba:</b> Satisfactoria.

Caso de Prueba	
<b>Código Caso de Prueba:</b> jws-1-2	<b>Nombre Historia de Usuario:</b> Ejecutar conexión remota.
<b>Nombre de la persona que realiza la prueba:</b> Yasmany Nuñez Broch	
<b>Descripción de la Prueba:</b> Esta prueba consiste en verificar que el usuario no pueda autenticarse con una maquina remota.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"> <li>• Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket.</li> <li>• El servidor de la aplicación tiene que estar ejecutándose.</li> <li>• El usuario debe estar autenticado con el servidor jWebSocket.</li> <li>• El usuario introduce sus credenciales de manera errónea.</li> </ul>	
<b>Entrada / Pasos de ejecución:</b> Para establecer conexión con la maquina remota el usuario debe introducir su nombre, contraseña y la dirección que posee la maquina en la red. Alguno de esos datos deben estar incorrectos. Luego presionar el botón “Login” y el sistema emitir una notificación que indique que es imposible conectar con la maquina remota.	
<b>Resultado Esperado:</b> El usuario no puede establecer conexión con la maquina remota.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Caso de Prueba
----------------

<b>Código Caso de Prueba:</b> jws-1-3	<b>Nombre Historia de Usuario:</b> Ejecutar conexión remota.
<b>Nombre de la persona que realiza la prueba:</b> Yasmany Nuñez Broch	
<b>Descripción de la Prueba:</b> Esta prueba consiste en verificar que el usuario pueda desconectarse de la maquina remota.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"> <li>• Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket.</li> <li>• El servidor de la aplicación tiene que estar ejecutándose.</li> <li>• El usuario debe estar autenticado con el servidor jWebSocket y en la aplicación.</li> </ul>	
<b>Entrada / Pasos de ejecución:</b> Para desconectarse el usuario debe hacer click en botón “Disconnect”.	
<b>Resultado Esperado:</b> El usuario pierde la conexión con la maquina remota.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

- **Casos de pruebas para la HU Ejecutar comandos**

Caso de Prueba	
<b>Código Caso de Prueba:</b> jws-2-1	<b>Nombre Historia de Usuario:</b> Ejecutar comandos
<b>Nombre de la persona que realiza la prueba:</b> Yasmany Nuñez Broch	
<b>Descripción de la Prueba:</b> Esta prueba consiste en verificar que se puedan ejecutar los comandos enviados por el cliente.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"> <li>• Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket.</li> </ul>	

<ul style="list-style-type: none"> <li>• El servidor de la aplicación tiene que estar ejecutándose.</li> <li>• El usuario debe estar autenticado en la maquina remota.</li> </ul>
<b>Entrada / Pasos de ejecución:</b> Para ejecutar comandos en la aplicación el usuario debe escribir en el área de la consola la orden a ejecutar.
<b>Resultado Esperado:</b> Se ejecutan en la maquina remota el comando insertado por el usuario.
<b>Evaluación de la Prueba:</b> Satisfactoria.

Caso de Prueba	
<b>Código Caso de Prueba:</b> jws-2-2	<b>Nombre Historia de Usuario:</b> Ejecutar comandos
<b>Nombre de la persona que realiza la prueba:</b> Yasmany Nuñez Broch	
<b>Descripción de la Prueba:</b> Esta prueba consiste en verificar que no se puedan ejecutar los comandos enviados por el cliente si antes no estaba autenticado en el sistema.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"> <li>• Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket.</li> <li>• El servidor de la aplicación tiene que estar ejecutándose.</li> <li>• El usuario no debe estar autenticado en la maquina remota.</li> </ul>	
<b>Entrada / Pasos de ejecución:</b> Para ejecutar comandos en la aplicación el usuario debe escribir en el área de la consola la orden a ejecutar y el sistema notificará que debe estar autenticado.	
<b>Resultado Esperado:</b> No se ejecuta el comando insertado por el usuario en la maquina remota.	

**Evaluación de la Prueba:** Satisfactoria.

- **Caso de pruebas para la HU Mostrar resultados**

<b>Caso de Prueba</b>	
<b>Código Caso de Prueba:</b> jws-3-1	<b>Nombre Historia de Usuario:</b> Mostrar Resultados
<b>Nombre de la persona que realiza la prueba:</b> Yasmany Nuñez Broch	
<b>Descripción de la Prueba:</b> Esta prueba consiste en verificar que se muestren los resultados de las ejecuciones de los comandos en el lado del cliente.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"><li>• Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket.</li><li>• El servidor de la aplicación tiene que estar ejecutándose.</li><li>• Se debe haber ejecutado algún comando.</li></ul>	
<b>Entrada / Pasos de ejecución:</b> Para que se muestren los resultados de los comandos el usuario debe introducir la orden en el área de la consola y presionar la tecla "Enter".	
<b>Resultado Esperado:</b> Se muestran los resultados esperados.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

- **Casos de pruebas para la HU Configurar consola**

<b>Caso de Prueba</b>	
<b>Código Caso de Prueba:</b> jws-4-1	<b>Nombre Historia de Usuario:</b> Configurar consola
<b>Nombre de la persona que realiza la prueba:</b> Yasmany Nuñez Broch	
<b>Descripción de la Prueba:</b> Esta prueba consiste en verificar que la consola cambie el color de fondo cuando el usuario seleccione una de las opciones de la	

paleta.
<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>• Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket.</li> <li>• El servidor de la aplicación tiene que estar ejecutándose.</li> </ul>
<p><b>Entrada / Pasos de ejecución:</b> Para cambiar el color de fondo de la consola se debe hacer click en una de las opciones que se encuentra en el recuadro “Color settings”.</p>
<p><b>Resultado Esperado:</b> Se cambia el color de fondo de la consola con el color seleccionado.</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria.</p>

Caso de Prueba	
<b>Código Caso de Prueba:</b> jws-4-2	<b>Nombre Historia de Usuario:</b> Configurar consola
<b>Nombre de la persona que realiza la prueba:</b> Yasmany Nuñez Broch	
<b>Descripción de la Prueba:</b> Esta prueba consiste en verificar que la consola cambie el color del texto cuando el usuario seleccione una de las opciones de la paleta.	
<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>• Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket.</li> <li>• El servidor de la aplicación tiene que estar ejecutándose.</li> </ul>	
<p><b>Entrada / Pasos de ejecución:</b> Para cambiar el color del texto que aparece en la consola se debe hacer click en una de las opciones que se encuentra en el recuadro “Color settings”.</p>	



<b>Resultado Esperado:</b> Se cambia el color del texto que aparece en la consola con el color seleccionado.
<b>Evaluación de la Prueba:</b> Satisfactoria.

### **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 “Mártires de Artemisa” 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 Releases, Estándar de Código, Caso de Prueba de Aceptación, Manual de Usuario, Manual de Instalación, Manual de desarrollo. También se puso a consideración del grupo de calidad el código fuente de la aplicación, este grupo llega a un proceso de revisión en el cual se analiza detalladamente la estandarización del código. El proceso de revisión de los artefactos se lleva 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.

### **Valoración del cliente**

Con el objetivo de comprobar la calidad y efectividad de los resultados, la aplicación desarrollada durante la presente investigación se puso a consideración del cliente Alexander Schulze, líder de la comunidad internacional y principal impulsor del proyecto jWebSocket.

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.

### **3.3. Resultados Obtenidos**

Al realizar las pruebas funcionales a la aplicación Remote Shell App se obtuvieron resultados satisfactorios, ya que ésta demuestra que la aplicación cumple con todas las funcionalidades que permite el control remoto de computadoras en tiempo real a través de 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 asegura que la aplicación va a ser utilizada por la comunidad de jWebSocket e integrada con el proyecto internacional jWebSocket. Todos estos elementos unidos van a tributar a aumentar los niveles de eficiencia en el proceso del control remoto de computadoras.

### **3.4. Funcionalidades Obtenidas**

Entre las principales funcionalidades que posee la aplicación Remote Shell App en su versión 1.0 se pueden mencionar:

- Conectarse remotamente a una computadora.
- Ejecutar comandos en dicha computadora.
- Mostrar los resultados de ejecución de los comandos en el cliente.
- Seleccionar el color de fondo para la consola.
- Seleccionar el color de fondo para el texto.

### 3.5. 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 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. 6** Diagrama de Despliegue. Fuente: Elaboración propia.

A continuación se describen los elementos del diagrama anterior:

- **Cliente:** 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.
- **Servidor jWebSocket:** El servidor de jWebSocket se conectará a la maquina remota a través del protocolo TCP/IP para ejecutar los comandos enviados por el cliente.

- **Maquina remota:** El servidor de jWebSocket se conectará a la maquina remota a través del protocolo TCP/IP para ejecutar los comandos enviados desde el cliente.

### **3.6. Aporte Social y Económico**

La aplicación Remote Shell App fue desarrollada utilizando el marco de trabajo jWebSocket. El cual es un marco de trabajo de código abierto lo que hace que la aplicación pueda ser distribuida para comunidades internacionales de desarrollo fomentando de esa manera el desarrollo colaborativo. Esta permite mejorar el proceso del control remoto de computadoras, llevándolo a la Web para que funcione en tiempo real. Al mismo tiempo muestra la capacidad de jWebSocket para enfrentar el desarrollo de aplicaciones web que trabajen en tiempo real.

Al poder ser accedida desde cualquier dispositivo con conexión a Internet y desde cualquier lugar del mundo, disminuye los costos de inversión de las empresas e instituciones a la hora de adquirir nuevos equipos de cómputo. Esto genera un mayor impacto a la hora de controlar los recursos de servidores en Internet que brindan servicios, ya que es el medio en el que más se podrá utilizar esta aplicación.

### **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 funcionales a la aplicación, mostrando que el sistema cumple con todas las funcionalidades requeridas, dándole una calidad máxima a la solución propuesta. Se verificó la calidad de la aplicación y de su documentación mediante la certificación de calidad de software y la valoración por parte del cliente. Quedó demostrada que la aplicación Remote Shell App se encuentra en óptimas condiciones para realizar el control remoto de computadoras a través de la web en tiempo real.

## **CONCLUSIONES GENERALES**

Con la realización de la presente investigación se dio respuesta a las preguntas científicas planteadas, obteniéndose de manera general las siguientes conclusiones:

- Se realizó la conceptualización del proceso de control remoto de computadoras que brinda a los usuarios la posibilidad controlar una PC remota sin estar operando en ella físicamente.
- Actualmente las aplicaciones de control remoto de computadoras permiten realizar configuraciones en los archivos del sistema entre otras cosas pero no trabajan en tiempo real.
- La aplicación web Remote Shell App posibilita el control remoto de computadoras en tiempo real utilizando el marco de trabajo jWebSocket.
- Se le realizó una estrategia de validación a la aplicación basada en diferentes etapas como son las pruebas funcionales, la certificación de calidad de software y la valoración del cliente para comprobar su funcionamiento.

## **RECOMENDACIONES**

Se recomienda desarrollar una segunda versión de la aplicación Remote Shell App, incluyéndole nuevas funcionalidades que tributen a aumentar el uso de la aplicación en diferentes sectores de la población.

## BIBLIOGRAFÍA REFERENCIADA

**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/>.

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

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

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

**Fette, I. y Melnikov, A. 2011.** The WebSocket Protocol. *Internet Engineering Task Force (IETF).* [En línea] 12 de 2011. [Citado el: 17 de 02 de 2012.] <http://tools.ietf.org/html/rfc6455>. 2070-1721.

**Figueroa, Roberth, Solís, Camilo y Cabrera, Armando. 2011.** *Metodologías tradicionales vs. Metodologías ágiles.* s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, 2011.

*Framework Approach for WebSockets.* **Schulze, Alexander. 2011.** s.l. : Web Technologies & Internet Applications (WebTech 2011), 2011. [http://dl.globalstf.org/index.php?page=shop.product\\_details&flypage=flypage\\_images.tpl&product\\_id=528&category\\_id=42&option=com\\_virtuemart&Itemid=4&vmcchk=1&Itemid=4..](http://dl.globalstf.org/index.php?page=shop.product_details&flypage=flypage_images.tpl&product_id=528&category_id=42&option=com_virtuemart&Itemid=4&vmcchk=1&Itemid=4..)

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

**Hybi. 2011.** The WebSocket protocol. *The WebSocket protocol.* [En línea] 30 de septiembre de 2011. [Citado el: 01 de diciembre de 2011.] <http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-17>.

**Jorney, Nadia. 2011.** eHow. eHow. [En línea] 2011. [http://www.ehow.com/about\\_6505443\\_definition-remote-computer.html](http://www.ehow.com/about_6505443_definition-remote-computer.html).

*jQuery, la librería Javascript por excelencia. Un framework Javascript lleno de ventajas.* **Soriano, Javier. 2011.** Valencia : <http://www.dicreato.com/blog/jquery-la-libreria-javascript-por-excelencia-un-framework-javascript-lleno-de-ventajas/>, 2011.

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

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

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

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

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

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

**SCRUM methodology. 2009.** SCRUM methodology. *SCRUM methodology.* [En línea] 2009. <http://www.scrummethodology.org/>.

**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.

**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/>.

**Wells, Don. 2009.** Extreme Programming: A gentle introduction. *Extreme Programming.* [En línea] 28 de 07 de 2009. [Citado el: 17 de 02 de 2012.] <http://www.extremeprogramming.org/>.



**Wolff, Phil. 2009.** Read Write Web. [En línea] Septiembre de 2009. [Citado el: 21 de 02 de 2012.] [http://www.readwriteweb.com/archives/explaining\\_the\\_real-time\\_web\\_in\\_100\\_words\\_or\\_less.php](http://www.readwriteweb.com/archives/explaining_the_real-time_web_in_100_words_or_less.php).

## ANEXOS

### Anexo 1: Prototipo de interfaz para Ejecutar conexión remota



Remote Shell Login

Hostname

Username

Password

Login

### Anexo 2: Prototipo de interfaz para Mostrar Resultados

```
> ls /home/yasmany/Desktop
Estudios
test.txt
```

### Anexo 3: Prototipo de interfaz para Configurar Consola



Color Settings

Font

Background