

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



**Transmisión de audio y video en un entorno de tiempo real usando
el marco de trabajo jWebSocket.**

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

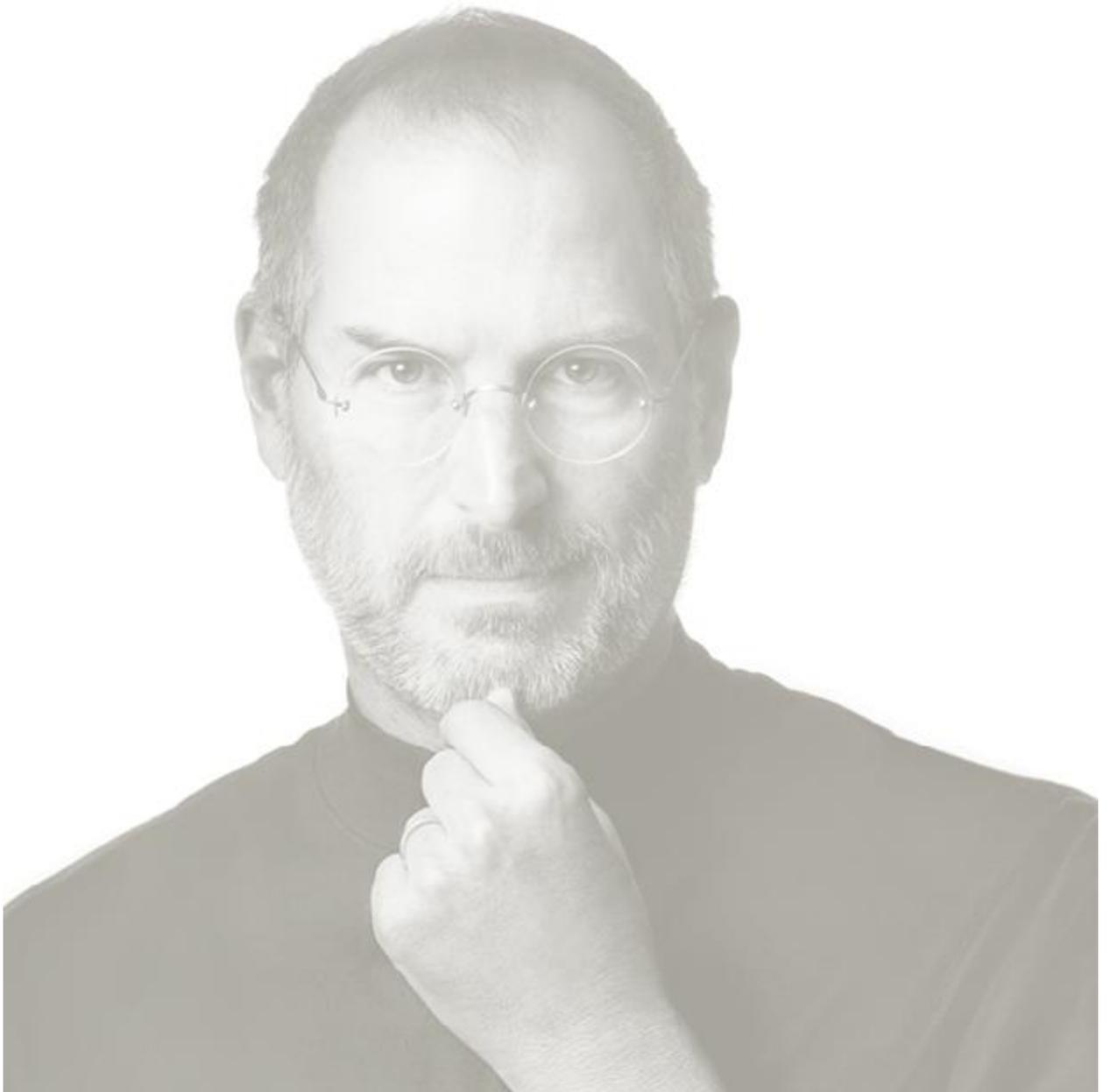
Autor: Alexander Rojas Hernández

Tutores: Msc. Yamila Vigil Regalado

Lic. Gilberto Ramón Justiniani Fernández

Artemisa, Cuba

Junio 2012



*“Estoy convencido de que la mitad de lo que separa a los emprendedores exitosos de los que no triunfan es la **perseverancia**.” (Steve Jobs)*

Declaración de Autoría

Declaro ser el autor de la presente tesis y autorizo a la Facultad Regional "Mártires de Artemisa", de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Alexander Rojas Hernández

Msc. Yamila Vigil Regalado

Firma del Autor

Firma del Tutor.

Agradecimientos

Quiero agradecer eternamente a mi madre querida, por entregarme siempre su amor incondicional, por verme siempre como su niño, aunque no lo sea ya, por luchar tanto para que no me falte nada y por darme siempre la fuerza para levantar el ánimo.

A mi padre luchador, siempre atento a mi desempeño y a mis responsabilidades como estudiante y como persona.

A mi familia en general, sin ellos no hubiera sido posible todos mis resultados a lo largo de la carrera.

A todos los profesores que han tomado parte en mi educación, a mi tutora y cotutor, a todos mis compañeros y verdaderos amigos que han estado conmigo en los buenos y malos momentos, a mi novia Dianelis, alias la kuki, por aguantarme tanto y por su paciencia. Muy especiales al siempre grupo 2 y a los locos del cuarto. A mis senseis Rolando Santamaría y Osvaldo Aguilar que gracias a ellos he aprendido muchas cosas buenas. Al proyecto de jWebSocket en general y a todos sus miembros que me dieron un gran impulso y experiencia colaborativa.

Especiales agradecimientos a Adis Bárbara y a su familia que lo largo de la carrera se han preocupado por mi trayectoria y bienestar, a todo el colectivo y comunidad del blog NarutoK@ge por ayudar a mejorarlo y por su gran colaboración en post de un mejor servicio para todos.

Dedicatoria

A mis tias lindas Bertha Lidia y Luisa Maria, que en paz descansen, por darme mucho apoyo y amor en todo momento.

A mi hermanita querida y a mis preciosos sobrinos.

A mi abuelito y a mi prima del alma.

Resumen

En la actualidad se ha difundido en la Web la utilización de transmisión de audio y video online en la cual los usuarios pueden visualizar contenido multimedia de forma dinámica. Además de las potencialidades que brinda la tecnología streaming y del conveniente uso que se le atribuye, se ve afectado en muchas ocasiones la integridad, por lo que afecta la calidad del servicio debido al gran número de conexiones y la concurrencia de usuarios conectados a Internet, ya sea desde las computadoras personales como desde los dispositivos móviles. Por lo tanto se hace necesario más movilidad, operatividad y tiempo real en la Web en general.

El marco de trabajo `jWebSocket` 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.

Por el gran auge y masificación en la utilización de la tecnología streaming en las aplicaciones web se identificó la necesidad de desarrollar dos aplicaciones, una aplicación estacionaria para la Web y otra nativa para los dispositivos móviles. Ambas utilizan el marco de trabajo `jWebSocket`. En el caso de la aplicación estacionaria se utiliza en el lado del cliente `Java Media Framework(JMF)` con la cual se puede acceder a los controladores de audio y video instalados en la pc. Por parte de la aplicación nativa se utiliza `PhoneGap`, un marco de trabajo para acceder a los controladores del dispositivo móvil. Estas aplicaciones demostrativas proveen un marco de trabajo de una herramienta poderosa que le permite desarrollar aplicaciones web con una mayor seguridad en la transferencia y la comunicación con los procesos del streaming en la web.

El resultado de las aplicaciones es satisfactorio ya que se logra una excelente comunicación entre los distintos tipos de clientes y el servidor, logrando así dar los primeros pasos en lograr una verdadera transmisión en tiempo real.

Índice

Introducción.....	1
Capítulo 1: Fundamentación teórica	9
1.1 Conceptos asociados al dominio del problema	9
1.2 Análisis de Soluciones Existentes	12
1.3 Metodología a emplear para el desarrollo de la solución	14
1.3.1 Metodologías Ágiles.....	15
1.4 Lenguajes de programación y modelado	17
1.5 Tecnologías usadas para el desarrollo de la solución.....	19
1.5.1 Marcos de trabajo.....	19
1.5.2 Marco de Trabajo del lado del Servidor	19
1.5.3 Marcos de Trabajo del lado del Cliente	21
1.6 Herramientas a emplear para el desarrollo de la solución	24
1.6.1 Herramienta CASE.....	24
Capítulo 2: Características, análisis y diseño del sistema.....	31
2.1 Planificación del proyecto por roles.....	33
2.2 Modelo de Negocio.....	35
2.3 Lista de Reserva del Producto.....	36
2.4 Historias de Usuario y Tareas de Ingeniería.....	40
2.5 Plan de Release.....	48
2.6 Arquitectura de Software.....	49
2.7 Diseño con Metáfora.....	50
2.8 Diagrama de componentes.....	51
Capítulo 3: Implementación y Validación del Sistema	54
3.1 Diagrama de despliegue	54
3.2 Validación de la Solución Propuesta	55
3.2.1 Aplicaciones Demostrativas	56
3.3 Casos de Pruebas	56
“Casos de pruebas para la HU Crear una envoltura genérica”.....	60

Introducción

“Casos de pruebas para la HU Realización de un visualizador”	60
3.4 Certificación de Calidad de Software	61
3.4.1 Valoración del Cliente	62
3.5 Resultados Obtenidos	62
3.6 Funcionalidades Obtenidas	62
3.7 Aporte Social y Económico	63
Recomendaciones	66
Bibliografía Referenciada	67
Bibliografía Consultada	70
Glosario de Términos	71
Anexos	72
Anexo 1: Captura de Imágenes.	72

INTRODUCCIÓN

La radio y la televisión han sido los principales medios de comunicación en todo el mundo, ya que permiten llevar al espectador una variada gama de contenidos audiovisuales de manera ágil, con los cuales se busca entretener, informar y educar a la audiencia. Inicialmente se utiliza tecnología analógica para su captura, transmisión, almacenamiento y reproducción; esta tecnología está presente desde hace muchos años, y ha crecido con la televisión convencional. Un sucesor natural de esta ha sido la tecnología digital que junto a la masificación de los computadores han permitido la transmisión digital de contenido multimedia.

Con el transcurso de los años se ha experimentado un considerado avance en la velocidad de transmisión de las redes de computadoras. Esta evolución ha sido apoyado además por las actuales capacidades de procesamiento de los ordenadores y los distintos protocolos de transmisión. De esta manera las aplicaciones de video y sonido sobre redes IP son más populares y más fáciles de implementar en comparación con años atrás.

La tecnología *streaming*¹ ha influido en el incremento de este tipo de aplicaciones, permitiendo la transferencia de contenido audiovisual por la red desde un servidor hacia sus clientes. Antes de que la tecnología *streaming* apareciera, la difusión de contenido multimedia a través de Internet necesariamente implicaba tener que descargar completamente el archivo contenedor al disco duro local. Como los archivos de audio y especialmente los de video son por lo general grandes en capacidad, su descarga y acceso como paquetes completos se vuelve una operación muy lenta, sin embargo, con la tecnología del *streaming*, el tiempo de descarga disminuye en gran medida y además permite la visualización en línea . (Laguna Torres, 2009)

Streaming es una secuencia de imágenes en movimiento que son enviadas en un formato comprimido sobre la red y es mostrado por un visualizador. La transmisión

¹ Streaming: Tecnología que permite la transmisión de audio y video sobre la red.

del flujo íntegro es streaming video pero con sonido, ya sea uno u otro, el usuario no tiene que esperar que descargar un archivo completo antes de ver el video o escuchar el sonido. En lugar de eso los datos son enviados en un flujo continuo y son visualizados a medida que llegue. La recepción más efectiva del flujo de audio y video requiere de alguna tecnología de frecuencia de banda ancha. Un paquete es la unidad de dato que esta enrutado entre un origen y un destinatario en Internet. (Universe., 2011)

Un protocolo muy utilizado para la transmisión de audio y video es User Datagram Protocol (UDP) que a diferencia de paquetes como en el protocolo TCP utiliza el término datagrama. Para poder poner a disposición de los clientes aplicaciones que usen esta transmisión, es necesario un servidor streaming, los que se encargan de realizar una serie de operaciones sobre los datos que se van a transmitir. Estos servidores capaces de enviar información audiovisual a través de una red de datos han evolucionado con mucha rapidez. Se ha podido presenciar desde los inicios de la Web los usuarios han querido comunicarse donde quiera que se encuentren y cada vez rápido, no solamente se socializa o comparte información mediante correos electrónicos o faxes. Con el creciente desarrollo que han tenido las Redes Sociales ejemplo Facebook² como herramientas para una comunicación entre dos o más usuarios mediante mensajería instantánea, sino que también se han incorporado nuevas tecnologías como streaming, la cual permite a los usuarios una comunicación de forma visual utilizando para esto las cámaras digitales. Además, con el uso de las aplicaciones nativas³ de los dispositivos móviles, ha potenciado el uso de la tecnología streaming el creciente desarrollo de la telefonía móvil la cual provee un amplio consumo por parte de los usuarios. Actualmente ocupa un 39% en ancho de banda en las redes de los celulares (Allot Communicatios-2011). Esto se ve referenciado en sitios como YouTube⁴ que permite a los usuarios compartir videos digitales a través de Internet.

² <http://www.facebook.com>

³ Nativa: Término con el cual se nombran las aplicaciones propias del dispositivo móvil

⁴ <Http://www.youtube.com>

Para el desarrollo de esta tecnología se han ido creando diferentes protocolos los cuales han aumentando increíblemente el flujo de este tipo de transmisión. Real Time Streaming Protocol (RTSP) usa una combinación de los protocolos TCP y UDP y RTP recupera varias funcionalidades para el mantenimiento de estado de sesión entre el servidor y el cliente. En otras palabras RTSP permite enviar peticiones simultáneas en ambas direcciones el cual es un buen mecanismo una ventaja sobre otros protocolos. Sin embargo este protocolo es vulnerable a la pérdida de datos, retraso de la transmisión, no presenta un mecanismo que asegure un transporte oportuno de los paquetes. (Wes, 2008)

Hoy día, el desarrollo de la tecnología streaming en la Web está dirigido a lograr una mayor interactividad, eficiencia y tiempo real. Con este objetivo surge el protocolo WebSocket, idea impulsada por distintos grupos internacionales. WebSocket es una tecnología que proporciona un canal de comunicación bidireccional sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores Web pero puede utilizarse por cualquier aplicación cliente/servidor. Websockets no constituye solamente una nueva forma de comunicación entre el cliente y el servidor, sino un cambio de paradigma para el desarrollo de aplicaciones web estacionarias y móviles.

Los principales servidores que soportan Websockets para el desarrollo de aplicaciones hoy día son, la Pasarela Websockets de Kaazing⁵, Jetty WebSocketServlet⁶, Socket.IO⁷, django-websocket del proyecto Python⁸ y jWebSocket⁹. Este último es un marco de trabajo de código abierto para el desarrollo de aplicaciones web estacionarias y nativas para los dispositivos móviles basado en Java en el lado del servidor y en JavaScript del lado del cliente (Framework Approach for WebSockets, 2011). jWebSocket es una nueva tecnología

⁵ <http://kaa5zing.com/>

⁶ <http://www.eclipse.org/jetty/>

⁷ <http://socket.io/>

⁸ <http://pypi.python.org/pypi/django-websocket>

⁹ <https://jwebsocket.org/>

orientada al desarrollo de aplicaciones basadas en WebSockets que proporcionan altos niveles de velocidad, escalabilidad, seguridad y el trabajo en tiempo real (Framework Approach for WebSockets, 2011). Este marco de trabajo está diseñado además para funcionar como un servidor web, proporcionando un conjunto importantes de funcionalidades que lo hacen más robusto y flexible.

En todos los escenarios donde hoy se hace uso de la tecnología *streaming* mediante el protocolo RTP se evidencia un gran desarrollo y mejora en cuanto a usabilidad por parte de los usuarios. Sin embargo la calidad y eficiencia del servicio muchas veces se ven comprometidas por las vulnerabilidades anteriormente expuestas y de las cuales se deriva la siguiente situación problemática.

La transmisión de audio y video mientras más solicitud de conexiones recibe, requiere una mayor velocidad de procesamiento de respuesta. Hoy día existen muchas limitantes con respecto a la velocidad con que se procesan estos datos generando dificultades en la integridad de la información. La utilización de protocolos como RTP hace que existan diferentes irregularidades en cuanto la calidad de la transmisión ya que no garantiza la entrega total de los mismos afectando así la integridad de la información. La pérdida y la inestabilidad de muchos paquetes de contenido multimedia cuando aumenta el número de conexiones trae consigo que muchas veces presente fallos en el flujo de comunicación. Estos problemas acarrearán que no sea posible un mejor aprovechamiento de la transmisión de audio y vídeo en múltiples entornos que lo requieren. De esta manera no logra satisfacer las necesidades para el intercambio de conocimiento en línea, el trabajo colaborativo y la comunicación permanente entre distintos usuarios. Esto implica que no pueda ser utilizado en áreas, donde el compartir experiencias en tiempo real es de vital importancia. Con el surgimiento del protocolo WebSocket se garantiza un mayor nivel de comunicación entre el cliente y el servidor ya que trabaja sobre un único canal socket de una forma bidireccional y full duplex por lo que se solucionan los problemas que tiene el protocolo RTP.

De lo planteado anteriormente nace la siguiente interrogante que nos define el **problema científico**:

¿Cómo garantizar mayor integridad de la información en las aplicaciones que transmiten audio y video en tiempo real en la Web con el uso del protocolo WebSocket?

Por tanto el presente trabajo centra su **objeto de estudio** en la transmisión de audio y video en tiempo real en la Web, delimitándose como **campo de acción** la transmisión de audio y video en tiempo real en la Web con el uso del protocolo WebSocket.

Para dar solución al problema anterior se plantea como **objetivo general**: Desarrollar una aplicación web y una nativa con el marco de trabajo jWebSocket que garantice mayor integridad de la información en las aplicaciones que transmiten audio y video en tiempo real en la Web.

Para dar cumplimiento al objetivo general se definen las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos-metodológicos de la transmisión de audio y video en tiempo real en la Web?
- ¿Cuál es la situación actual de la transmisión de audio y video en tiempo real que se hace mediante la Web?
- ¿Cómo desarrollar una aplicación web y una nativa con el marco de trabajo jWebSocket que garantice mayor integridad de la información en las aplicaciones que transmiten audio y video en tiempo real en la Web?
- ¿Cómo comprobar la capacidad y la potencialidad de las aplicaciones propuestas para la transmisión de audio y video en aplicaciones web y nativas desarrolladas con jWebSocket?

Para dar cumplimiento a los objetivos propuestos se planificaron las siguientes:

Tareas de la investigación:

1. Fundamentación teórico-metodológico de la transmisión de audio y video en tiempo real en la Web.
2. Análisis de la situación actual de la transmisión de audio y video en tiempo real en aplicaciones web y nativas.
3. Desarrollo de una aplicación web y una nativa para la transmisión de audio y video en aplicaciones web y nativas desarrolladas con el marco de trabajo WebSocket.
4. Comprobar la capacidad y potencialidad de la aplicaciones para transmisión de audio y video mediante pruebas de aceptación y otros métodos asociados a las Ciencias Informáticas.

Los **métodos científicos** utilizados en esta investigación fueron:

- **Histórico-Lógico:** Permite analizar la trayectoria completa del proceso de administración de aplicaciones distribuidas, así como el estudio histórico del mismo que permite ver deficiencias y proponer soluciones acorde a las necesidades.
- **Análisis Documental:** Permite realizar el estudio de una variada documentación referente a la gestión de tarjetas inteligentes en aplicaciones 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 investigación.
- **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.

Introducción

- **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 definen las siguientes **variables de la investigación:**

Variables Dependientes:

- Nivel de integridad de la información en las aplicaciones que transmiten audio y video en tiempo real en la Web.

Variable Independiente:

- Aplicación web y nativa para la transmisión de audio y video en tiempo real en la Web utilizando marco de trabajo jWebSocket.

Posible Resultado

- Aplicación web y nativa para la transmisión de audio y video en tiempo real en la Web utilizando marco de trabajo jWebSocket.

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 sobre el proceso de transmisión de audio y vídeo en la Web 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 de las aplicaciones. Se elaboran y documentan las pruebas realizadas a la solución propuesta para demostrar el cumplimiento de los requerimientos de la misma.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

El presente capítulo tiene como objetivo, tratar los principales conceptos y aspectos más significativos relacionados con la temática abordada desde distintos enfoques. Un estudio acerca del estado del arte de la transmisión de audio y video en tiempo real en la Web y en específico de la transmisión de audio y video en tiempo real en la Web con el uso del protocolo WebSocket. Se analizan además las diferentes metodologías de desarrollo de software, lenguajes de programación, tecnologías y herramientas que serán usados para el desarrollo de la solución, siempre teniendo en cuenta la necesidad del uso de aplicaciones de código abierto.

1.1 Conceptos asociados al dominio del problema

Son muchas las definiciones de streaming en la Web dadas por prestigiosas revistas, organizaciones y por diferentes autores. Se encuentra el enfoque brindado en una de las conferencias de la W3C para la cual streaming es el *“continuo envío, recibo y presentación al usuario final de paquetes, los cuales son codificados y decodificados constantemente sobre la red.”* (Chen, See-Mong, H. Campbell, & Li, 2005). Existen otros criterios en los cuales se hace referencia a esta tecnología como la transferencia de contenido audiovisual por la red desde un servidor hacia sus clientes. Streaming una *“tecnología que se utiliza para aligerar la descarga y ejecución de audio y video en la web, ya que permite escuchar y visualizar los archivos mientras se están descargando.”* (Morales Venegas & otros, 2005).

Para la siguiente investigación y a partir de las definiciones antes mencionadas se define como streaming a la tecnología que permite transferir datos codificados sobre la red de manera que se puedan procesar en un flujo regular y continuo. Posteriormente los paquetes son decodificados y visualizarlos en el cliente.

Para un mejor entendimiento de los procesos por los que transita el flujo de audio y video, se describen a continuación cada una de las etapas.

Inicialmente está la captura del contenido audiovisual, donde es convertido en datos digitales, aún sin codificación y compresión. El codificador utiliza herramientas denominadas genéricamente como encoders¹⁰, los cuales poseen los algoritmos suficientes para permitir disminuir el tamaño de la información y así optimizar el consumo de ancho de banda. Posteriormente el servidor procesa los datos multimedia en cortos espacios de tiempo y soporta funciones de control interactivas siendo además el responsable de suministrar los servicios de audio y video en modo sincronizado. El flujo continuo de datos es distribuido por la red a los diferentes clientes ya sea a través de unicast, multicast o broadcast. En la etapa del decodificador los datos son descodificados y descomprimidos de acuerdo al formato contenedor y a los *codecs* usados en el proceso de en codificación. Por último el reproductor es el encargado de mostrar todo el contenido multimedia a partir de los datos enviados.

El proceso de transmisión de los datos se basan en 3 tipos, entre ellos se encuentra Unidifusión o Unicast, Multidifusión o Multicast y por último Difusión o BroadCast. A continuación se explican los distintos procesos de transmisión.

- **Unidifusión (Unicast)** : Se basa en un proceso de envío de una información en una o más unidades de datos (datagramas IP) desde una máquina origen a una única máquina destinataria o receptor final. Por tanto, es una transmisión punto a punto con cada destinatario. Si se desea enviar la misma información y hay “n” destinatarios, habrá “n” comunicaciones punto a punto independiente o “n” copias de la misma información enviadas desde la máquina origen. (YÁGÜEZ, 2005)
- **Multidifusión (Multicast)**: Cuando la información es distribuida a un grupo determinado de usuarios simultáneamente, se está usando el esquema de distribución Multicast. El servidor envía un solo flujo para todos los miembros

¹⁰ **Encoder**. Es un dispositivo que convierte una señal RGB (red, green and blue) en una señal NTSC. Es utilizado para grabar clips de video digital a un videocasete .

del grupo, esto significa que se usa el mismo ancho de banda para enviar el contenido a uno o a 100 clientes. (Quintero, Pablo, Castro, & Cristian, 2006)

- **Difusión (Broadcast):** Se basa en un único proceso de envío, independientemente del número de potenciales máquinas receptoras, de una misma información en una o más unidades de datos (datagramas IP) desde un origen a todas las máquinas de una red de área local. (Pérez Agüera, Sánchez Jiménez, & Caldera Serrano, 2004).

Desde los inicios de la tecnología streaming se ha trabajado sobre la base de lograr una transmisión eficiente y rápida, de modo que términos como el de tiempo real es de vital relevancia para lograr tales fines. Para esta investigación se utiliza el concepto de tiempo real brindado por Lambert M. Surhone en el cual plantea que 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).

El principal factor para lograr *streaming* en tiempo real es que no exista un almacenador intermedio de los paquetes de datos, solamente pequeños buffers en el decodificador, pero la señal es esencialmente transmitida todo el camino desde la cámara digital hasta el visualizador. La información no se almacena en el disco local, a menos que el usuario lo permita. (Austerberry, 2008). Streaming en tiempo real emplea protocolos como *Real Time Protocol(RTP)*¹¹ para lograr una mejor calidad de servicio. Dicho protocolo se crea específicamente para la transmisión de audio y video, gracias a que incluye en su cabecera informaciones que sincronizan imagen y sonido. Trabaja sobre UDP¹² y tiene características especiales para el trabajo con sistemas en tiempo real. En un principio fue diseñado para emisiones *multicast* de tráfico en tiempo real ,aunque también se puede utilizar en emisiones

¹¹ RTP es un protocolo doble basado en IP que proporciona soporte para el transporte de datos en tiempo real (flujos de vídeo y de audio)

¹² UDP es un protocolo del nivel de transporte basado en el intercambio de datagramas que proporciona una sencilla interfaz entre la capa de red y la capa de aplicación

unicast, y puede ser utilizado para el video bajo demanda y servicios interactivos.

Aunque los protocolos que actualmente son utilizados para dar soporte y brindar servicios de la tecnología streaming tienen grandes potencialidades de transmisión, no logran cumplir con ciertos parámetros los cuales son imprescindibles para lograr una integridad en el servicio. En el caso de RTP no asegura la entrega continua de información, ni la de todos los paquetes y no puede evitar la entrega desordenada de los mismos. Por estas razones se hace necesario introducir nuevas técnicas basadas en el protocolo Websocket para lograr un perfeccionamiento en cuanto al tiempo real de la tecnología streaming.

Para esto es vital explicar el término del protocolo Websocket, que según Petter Lubbers es *“una tecnología que proporciona un canal de comunicación bidireccional y full-duplex¹³ sobre un unico socket¹⁴ TCP. Soluciona las limitaciones del protocolo HTTP, al establecer una comunicación full-duplex (TCP) entre el cliente y el servidor, sustituyendo la comunicación half-duplex¹⁵ (HTTP)”*. Se reduce, en grandes proporciones, el tráfico en la red teniendo en cuenta que al establecer la comunicación websockets entre el cliente y el servidor solo hay un envío de 2 bits, eliminando las cabeceras HTTP, (HTML5 Websockets and communication, 2010).

1.2 Análisis de Soluciones Existentes

La historia de la transmisión de audio y video se puede ver en tres líneas de tiempo separadas. Una relativa a las tecnologías de streaming, otra junto a la velocidad de conexión y el dispositivo de destino, y la final en relación con el crecimiento de los terceros proveedores de servicios de streaming.

En el surgimiento de la tecnología Streaming donde Real Networks fue pionera en la transmisión de audio y video de los mercados, la difusión del primer evento de

¹³ Full-Duplex: Cualidad de los elementos que permiten la entrada y salida de datos de forma simultánea.

¹⁴ Socket: Método para la comunicación entre un programa del cliente y un programa del servidor en una red.

¹⁵ Half-Duplex: Significa que el método o protocolo de envío de información es bidireccional pero no simultáneo.

audio a través de Internet un juego de béisbol entre los Yankees y los Marineros de Seattle - en 1995, el lanzamiento de la primera tecnología de streaming de video en 1997. Posteriormente devino la llamada era de Microsoft dando el Reproductor de Windows Media una posición dominante en equipos de sobremesa y portátiles disponibles. Desde principios de los años 2000 hasta alrededor del 2007, Windows Media era el formato más utilizado en Internet y en intranets la mayoría de las empresas. Durante este mismo período, el diseño del sitio web fue la transición de HTML a Flash, que ofrecen mayor interactividad y flexibilidad de diseño. Aunque tenía un componente de Flash video, los codecs iniciales ofrecen una calidad de video y la sincronización de los pobres incompleta de audio/ video. Por último la era de Flash donde Macromedia podría igualar la calidad de video de Microsoft en un reproductor marcable que pudiera integrarse con el resto de un sitio basado en Flash, y fue verdaderamente multiplataforma y cerca de todas partes, fue cuando el streaming en la web fue tomando más auge.

Son muchas las empresas y los escenarios en que actualmente se hace uso de esta tecnología, se pueden contemplar en proyectos de colaboración en línea y aprendizaje o también como un medio de entretenimiento e interacción directa entre los usuarios. Se destacan entre ellas las redes sociales como Facebook. El mismo en el año 2011 introdujo una aplicación la cual permite a los usuarios una comunicación más directa mediante la webcam de la computadora. La misma está desarrollada en Flex¹⁶ en el cliente y como servidor de Flash utilizan Red5¹⁷ bajo el protocolo RTMP¹⁸. Otros de los escenarios donde el uso de la tecnología streaming es visible se encuentra en los sitios de visualización y descarga videos como YouTube, donde el consumo diario por los usuarios es notable. En Cuba, aunque en menor porción, se evidencia el uso de streaming en sitios como Cubasi, entre otros. Los mismos trabajan con servidores de streaming los cuales ofrecen video

¹⁶ Flex: Adobe Flex es una herramienta derivada de Flash utilizado para crear aplicaciones RIA

¹⁷ Red5: Servidor de Flash de código abierto, es utilizado como alternativa a Flash Media Server.

¹⁸ RTMP: Real Time Media Protocol, protocolo para la transmisión de audio y video.

bajo demanda a los usuarios y utilizan protocolos como RTP y MMS¹⁹ para la transmisión. Por otra parte se encuentran los sitios que utilizan la tecnología streaming para ofrecer conferencias en línea y presentan un trabajo colaborativo entre los integrantes. Streaming posee con un amplio campo de acción y desarrollo, el cual se ve reflejado día tras día en la Internet, ya que atrae a los usuarios y también como parte de la misma competencia comercial. Los protocolos de se utilizan para esta transmisión de audio y video mientras más solicitud de conexiones recibe, requiere una mayor velocidad de procesamiento de respuesta. Hoy día existen muchas limitantes con respecto a la rapidez con que se procesan estos datos generando dificultades en la integridad de la información. La utilización de protocolos como RTP hace que existan diferentes irregularidades en cuanto la calidad de la transmisión ya que no garantiza la entrega total de los mismos afectando así la integridad de la información. La pérdida y la inestabilidad de muchos paquetes de contenido multimedia cuando aumenta el número de conexiones trae consigo que muchas veces presente fallos en el flujo de comunicación. Estos problemas acarrearán que no sea posible un mejor aprovechamiento de la transmisión de audio y video en múltiples entornos que lo requieren. De esta manera no logra satisfacer las necesidades para el intercambio de conocimiento en línea, el trabajo colaborativo y la comunicación permanente entre distintos usuarios. Esto implica que no pueda ser utilizado en áreas, donde el compartir experiencias en tiempo real es de vital importancia. Con el surgimiento del protocolo WebSocket se garantiza un mayor nivel de comunicación entre el cliente y el servidor ya que trabaja sobre un único canal socket de una forma bidireccional y full duplex por lo que se solucionan los problemas que tiene el protocolo RTP.

1.3 Metodología a emplear para el desarrollo de la solución

Todo proceso de desarrollo de una aplicación informática debe estar regido y orientado por una metodología de desarrollo de software, que guíe los procesos y

¹⁹ MMS: Multimedia Media Service, utilizado para enviar mensajes incluyendo contenido multimedia.

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 cortos tiempo.

1.3.1 Metodologías Ágiles

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. Estas logran que se minimicen los riesgos desarrollando software en corto tiempo. Entre las metodologías ágiles mas destacas se encuentra SCRUM y XP.

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. Sus principales características son: desarrollo mediante iteraciones y reuniones sistemáticas a lo largo del proyecto. (Schwaber, y otros, 2011)

Programación Extrema²⁰

XP es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Esta consiste en una programación rápida o extrema y es utilizada para proyectos de corto plazo, con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. (Wells, 2009) Esta metodología es basada en pruebas unitarias, la re-fabricación y la programación en pares.

El ciclo de vida ideal de esta metodología consta de 6 fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

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

²⁰ Programación Extrema: (*eXtreme Programming* en inglés, habitualmente resumido como XP)

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 añade 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. (Peñalver, Meneses, & García, 2010)

Después de analizar la metodología RUP y observar las metodologías ágiles se ha llegado a la conclusión de que no existe una metodología universal que indique cómo crear todo tipo de software. Cuando un equipo decide construir un producto, 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. (Peñalver, Meneses, & García, 2010). Debido a las grandes ventajas que proporcionan las metodologías ágiles se propone para el desarrollo de este trabajo el uso de SXP.

1.4 Lenguajes de programación y modelado

Lenguaje Modelado Unificado.

Lenguaje Modelado Unificado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. 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

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*).

JavaScript

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. En cuanto a sus limitaciones, JavaScript fue diseñado de forma que se ejecutara en un entorno muy limitado que permitiera a los usuarios confiar en la ejecución de los scripts. De esta forma, los scripts de JavaScript no pueden comunicarse con recursos que no pertenezcan al mismo dominio desde el que se descargó el script.

HTML

El Lenguaje para Marcado de Hipertexto HTML es un lenguaje estático para el desarrollo de sitios web permitiendo estructurar las páginas web, fue desarrollado por el World Wide Web Consortium (W3C) y se ha convertido en un estándar reconocido mundialmente constituyendo el idioma común para todos los

navegadores. Los archivos pueden tener las extensiones htm y html.

1.5 Tecnologías usadas para el desarrollo de la solución

1.5.1 Marcos de trabajo

Un Marco de Trabajo (framework) es un esquema, esqueleto o patrón para el desarrollo y/o la implementación de una aplicación. (Sánchez, 2006) En otras palabras, un marco de trabajo se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta, puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Los marcos de trabajo son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel para proveer un sistema funcional.

Existen multitud de marcos de trabajo orientados a diferentes lenguajes, funcionalidades, entre otros. Aunque la elección de uno de ellos puede ser una tarea complicada, lo más probable es que a largo plazo sólo los mejor definidos o más utilizados permanezcan.

1.5.2 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 nativas para los dispositivos móviles basado en Java en el lado del servidor. En el lado del cliente actualmente se puede trabajar con JavaScript, C# y Python. jWebSocket establece un modelo de token, los mismos 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. En la primera opción jWebsocket proporciona un archivo *.jar*, ofreciendo la ventaja de ejecutarse fácilmente desde una línea de comandos e integrarse a la biblioteca de una aplicación existente de Java. En la actualidad hay algunos servidores que ya soportan Websockets y otros que no, por lo que jWebsocket se integra a servidores como Tomcat o Apache para lograr una comunicación Websockets. En caso de que los servidores soporten de manera nativa Websockets, como el caso de Jetty o GlassFish, se incluyen las funciones de comunicación del marco de trabajo jWebsocket, pero los motores internos se apagan y el anfitrión se utiliza. Esto asegura que no haya mecanismos de seguridad adicionales.

jWebsocket como servidor web proporciona un conjunto importantes 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. En un clúster los plug-ins se pueden utilizar como servicios, por lo que jWebsocket perfectamente es compatible con Service Oriented Architectures (SOA) en un entorno totalmente basado en eventos. 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.(Schulze, 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.

1.5.3 Marcos de Trabajo del lado del Cliente

Sencha Touch

Sencha Touch es un marco de trabajo de JavaScript interfaz de usuario especialmente construido para móviles web. Puede ser utilizado por los desarrolladores web para desarrollar la interfaz de usuario para aplicaciones web móviles que se ven y se sienten como aplicaciones nativas de dispositivos móviles compatibles. Está totalmente basado en estándares web como HTML5, CSS3 y táctil JavaScript. Sencha Touch es un producto de Sencha que se formó tras la combinación de las populares bibliotecas de Javascript Ext JS, jQTouch y Raphael . (Raphael Eyvindr, 2011)

Ventajas:

- Posee un grupo de componentes y eventos ricos en interacción.
- Integración con PhoneGap.
- Soporta almacenadores locales para los componentes, cuenta con Yahoo Query Language.

Desventajas:

- Las aplicaciones desarrolladas con este marco de trabajo aún no proveen una gran experiencia de usuario.
- Tiene algunas dificultades aún en la administración de la sincronización de los datos.

(Clevenger, 2011)

jQuery Mobile

jQuery Mobile está basado en el popular marco de trabajo de JavaScript jQuery. jQuery Mobile está designado para crear aplicaciones web para dispositivos móviles, la cuales se son soportadas por todos los dispositivos. Tiene desarrolladas capacidades para interfaz de usuario y brinda al desarrollador un variedad gama de diseños, componentes de interfaz de usuario e interacciones, así como una API para aplicar y extender del mismo. (Reid, 2012)

Ventajas

- Menos tiempo diseñando e implementando.
- Incorpora capas prediseñadas, controles y componentes.
- Brinda transiciones atractivas y efectos para tu sitio móvil.
- Fácil adaptación a una larga variedad de dispositivos móviles.

Desventajas

- Reemplaza algunas etiquetas HTML con un nuevo marcado el cual requiere jQuery Mobile para la apariencia y comportamiento de la aplicación
- Aplicar jQuery Mobile a muchos proyectos web los hace ver muy similares.

PhoneGap

PhoneGap es un marco de trabajo open source para la creación de aplicaciones nativas móviles utilizando HTML, CSS y Javascript para iPhone, Android, BlackBerry, Palm webOS, y Symbian WRT (Nokia). PhoneGap es perfecto para la transformación de una aplicación web móvil para una aplicación nativa. Es fácil de usar para desarrolladores web. Para utilizar PhoneGap, un desarrollador web tendrá que aprender a construir utilizando uno o más dispositivos y herramientas de SDK, pero todo el código de la aplicación puede ser HTML, CSS y JavaScript.

(Allen, Graupera, & Lundrigan)

Ventajas

- Multiplataforma (iPhone, Android, BlackBerry, Palm webOS, y Symbian WRT (Nokia), Bada).
- Funcionalidad de Aplicaciones Nativas las cuales son fáciles de usar y soportadas por varios dispositivos.
- Creación de plugins extensibles para aumentar las potencialidades del mismo y en correspondencia a las necesidades del usuario.

Desventajas

La principal desventaja es que la aplicación web dejará de funcionar y debe ser instalada por el usuario, lo que implica que no será accesible desde la web.

(Raasch, 2012)

Java Media Framework

El Java Media Framework(JMF) es una biblioteca Java que permite audio, video y otros medios de comunicación basados en el tiempo que se añade a las aplicaciones Java y applets. Este paquete opcional, que puede capturar, reproducir, transmitir, y convierte múltiples formatos de medios, se extiende la Plataforma Java, Edición Estandard y permite el desarrollo de aplicaciones multimedia multi-plataforma. (P. Miller, F. Vandome, & McBrewster, 2010)

JMF soporta los formatos comunes de audio y video como AIFF, AU, AVI, GSM, MIDI, MPEG, QuickTime, RMF y WAV. Soporta la transmisión de audio y video utilizando los protocolos RTP/ RTCP. A pesar de que funciona con applets, tanto para capturar como para mostrar, posee una atractiva plataforma para su implementación y adaptables mecanismos de transmisión.

Flex

Adobe Flex es una herramienta derivada de Flash utilizado para crear aplicaciones RIA , posee un lenguaje declarativo de marcado llamado MXML para la estructura de la aplicación, además, al igual que Flash utiliza Action Script 3.0. Para la interfaz hace uso de CSS y posee su propio editor Flex Builder. Para lograr streaming utiliza Flash Media Server mediante el protocolo RTMP.

Provee contenedor de capas , control de formularios, formateadores y validadores, y librerías de animación las cuales potencian mucho la interfaz del usuario. Además no brinda funcionalidades más allá de las que tiene, o sea, no se pueden agregarle extensiones adicionales.(Tapper, Labriola, Boles, & Talbo, 2008)

Después de analizar los distintos tipos de marcos de trabajo para el cliente, se llegó a la decisión de utilizar PhoneGap para el desarrollo de la aplicación de dispositivos móviles ya que soporta múltiples plataformas como iPhone, Android, BlackBerry,

Palm webOS, y Symbian WRT (Nokia),Bada), además es fácil de usar y presenta potencialidades las cuales puede aprovechar el usuario.

Para el desarrollo de la aplicación estacionaria se seleccionó Java Media Framework ya que se trabaja en el mismo lenguaje, tanto en el cliente como en el servidor y hay un mejor dominio de los elementos técnicos, además que brinda un API la cual ha sido utilizada por bastante tiempo en la Web y aún se sigue utilizando.

1.6 Herramientas a emplear para el desarrollo de la solución

1.6.1 Herramienta CASE

Las herramientas Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador (CASE) 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.

Dichas herramientas se encuentran en una continua evolución, por lo que existe una gran variedad de proveedores y productos, cada uno de ellos con diferentes aplicaciones y especificaciones. A continuación se hace una caracterización de algunas de ellas que permitirá adquirir los elementos necesarios para determinar cuál es la más idónea para especificar y diseñar la solución propuesta. (Alfaro, 2011)

Visual Paradigm

Visual Paradigm es una poderosa herramienta CASE que hace uso del Lenguaje Modelado Unificado (UML) con soporte multiplataforma, que proporciona un ciclo de vida completo del desarrollo de software, excelentes facilidades de interoperabilidad con otras aplicaciones, compatibilidad entre versiones, así como dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Dicha herramienta brinda una serie de facilidades que se mencionan a continuación: (Visual Paradigm)

- 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.

Anteriormente se analizaron algunas de las herramientas CASE existentes en el mundo para el modelado de software, ejemplo Rational Rose. 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.

Herramientas de Control de Versiones

Un sistema de control de versiones es un sistema de gestión de archivos y directorios, cuya principal característica es mantener el historial de cambios y modificaciones que se han realizado sobre dichos archivos a lo largo del tiempo. Es importante decir que estos sistemas no solo se limitan a gestionar archivos de texto sino que también gestionan documentos, imágenes y ficheros de todo tipo. Los sistemas de control de versiones utilizados a nivel internacional con gran aceptación y popularidad guardan toda la información en un repositorio central accesible a través de la red, permitiendo el trabajo colaborativo entre varios puestos de trabajo y a su vez proporcionando una mayor seguridad y disponibilidad de los datos. 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.

Subversion

Subversion es un sistema de control de versiones completamente equipado que fue originalmente diseñado para reemplazar a CVS. Desde entonces se ha expandido más allá de su objetivo original, pero su modelo básico, el diseño y la interfaz fueron fuertemente influenciados por CVS por lo que debido a estas particularidades los usuarios de CVS se sienten muy cómodos al interactuar con Subversion. (Apache Software Foundation, 2011)

Este sistema presenta varias características importantes las cuales aparecen a continuación.

- Los directorios son versionados.
- Resolución de conflictos de forma interactiva.
- Gestiona de manera eficaz los archivos binarios.
- Bloqueo de archivos.
- Vinculaciones para lenguajes de programación.
- Vinculación de varios repositorios.
- Soporte para desarrolladores.
- Desarrollo Paralelo.

Teniendo en cuenta las características expuestas, 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.

Herramientas del Cliente de Control de Versiones

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

TortoiseSVN

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

Características de TortoiseSVN:

Integración con el Shell de Windows: TortoiseSVN se integra perfectamente en el Shell de Windows (por ejemplo, el explorador). Esto significa que puede seguir trabajando con las herramientas que ya se conocen.

- No está obligado a usar el explorador de Windows. Los menús contextuales de TortoiseSVN también funcionan en otros administradores de archivos y en el diálogo Fichero/Abrir que es común a la mayoría de aplicaciones estándar de Windows. Sin embargo, debe tener en cuenta que TortoiseSVN está desarrollado con la mirada puesta en hacerle extensión del Explorador de Windows. Por este motivo, puede que en otras aplicaciones la integración no sea tan completa y que, por ejemplo, los íconos sobreimpresionados en las carpetas no se muestren.

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)

Características:

- 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 Raptidsvn, a pesar de todas las ventajas y funcionalidades que ofrece Tortoise, se decide usar RapidSVN ya que esta 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.

Entorno integrado de desarrollo - IDE

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

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. Esta plataforma, típicamente ha sido usada para desarrollar entornos integrados de desarrollo, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte del IDE. Eclipse basa su funcionalidad en módulos (en inglés plug-in) que se adaptan a las necesidades del programador. Este mecanismo de módulos es una plataforma ligera para componentes de software que permite el uso de diferentes lenguajes de programación como son Java, C/C++ y Python (Eclipse, 2011). Helios es una liberación particular de Eclipse la cual además de Java, tiene una integración con el SDK de Android y un conjunto de herramientas para poder realizar aplicaciones Android. (Felker & Dobbs, 2010).

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 una de las soluciones propuestas 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 Java Standard Edition

(JavaSE7) con las características de Java Development Kit (JDK7). Esta versión también ofrece una integración mejorada con el servidor Oracle WebLogic, así como soporte para Oracle Database y GlassFish3.1. Otros puntos destacados incluyen soporte para Maven3 y HTML5, así como mejoras en el editor de Java (NetBeans, 2011).

Para la aplicación para dispositivos móviles se seleccionó Eclipse Helios porque además de Java, tiene una integración con el SDK de Android y un conjunto de herramientas para poder realizar aplicaciones Android. (Felker & Dobbs, 2010)

Conclusiones del capítulo

En este capítulo se abordaron los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. Se analizaron principalmente lo referente a la tecnología streaming, así como su funcionamiento, además de los principales protocolos de transmisión de datos más usados. Además se explicó las etapas por las que transcurre el proceso de streaming y se identificaron algunas aplicaciones que hacen uso de esta tecnología. Con el mismo se logra tener un mejor entendimiento de los procesos de captura y de transmisión tanto del video como del sonido digital. También se identificaron los conceptos de tiempo real en la Web y del protocolo WebSocket.

CAPÍTULO 2: CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

Para encaminar el desarrollo de la aplicación hacia un sistema correcto es preciso conocer su contexto e identificar las condiciones y funciones que debe cumplir.

Para fundamentar las características del sistema se puntualiza, en el presente capítulo, cómo debe funcionar el mismo y se destaca sus características distintivas lo que permite una comprensión general de la aplicación a desarrollar. Para ello se describen con detalles los procesos del negocio que lo soportan, además se presentan los requisitos Funcionales y no Funcionales mediante la Lista de Reserva del Producto para cumplir el objetivo planteado. Se confeccionan las Historias de Usuarios y las Tareas de la Ingeniería asociadas a las mismas y se muestran los modelos necesarios para la creación del diseño de la aplicación a realizar.

Características y Principales Funcionalidades de la aplicación.

La transmisión de audio y video utilizando el marco de trabajo `WebSocket` garantiza un flujo de comunicación en tiempo real bajo el protocolo `WebSocket`. Los sistemas que actualmente utilizan la tecnología streaming están basados en una estructura cliente – servidor utilizando el protocolo de comunicación `Real Time Protocol (RTP)`, el cual provee una buena sincronización de imágenes y sonido. Sin embargo el uso de este protocolo para esta comunicación tiene sus desventajas: no garantiza un continuo recibo de la información, ni la de todos los paquetes. No puede evitar la entrega desordenada de los mismos. A pesar de que se apoya en el protocolo `UDP` el cual es bastante rápido para la transmisión, no garantiza altos niveles de precisión en la entrega de la información, viéndose afectado la integridad de la misma.

La solución propuesta consiste en una aplicación web y una nativa que garantiza la transmisión de audio y video en tiempo real en la Web mediante el protocolo WebSocket. Para la aplicación nativa para dispositivos móviles fue desarrollada del lado del cliente con el marco de trabajo PhoneGap. La aplicación estacionaria por su parte utiliza el marco Java Media framework para su desarrollo. Ambas están implementadas en el lado del servidor con el marco de trabajo jWebSocket garantizando la comunicación mediante el protocolo Websocket.

En el caso de la aplicación nativa se accede a los componentes de captura del dispositivo móvil como la imagen, el video y el audio, se codifican los datos y se envían hacia el servidor. En la aplicación estacionaria el proceso de captura es desde la webcam y se realiza un proceso similar. Estas aplicaciones garantizan tiempo real utilizando el protocolo WebSocket y un trabajo bidireccional de alta precisión, se garantiza la seguridad e integridad de la información ya que el servidor controla que el cliente esté conectado y enviando datos.

Los aspectos antes mencionados: tiempo real y comunicación bidireccional, fomentan un nuevo paradigma de comunicación en la Web. Esto se logra haciendo uso del marco de trabajo jWebSocket, una nueva tecnología orientada al desarrollo de aplicaciones web sobre el protocolo de comunicación WebSocket.

Actualmente existen soluciones que utilizan la tecnología streaming mediante la Web, sin embargo estas aplicaciones no usan el protocolo WebSocket para la comunicación, el cual proporciona altos niveles de seguridad, escalabilidad y velocidad.

Dentro de las funcionalidades del sistema se especifica primeramente la aplicación estacionaria desarrollada para la transmisión de audio y video utilizando el marco de trabajo jWebSocket presenta las siguientes funcionalidades:

- Captura de video desde la webcam
- Visualización del contenido desde un applet java
- Codificación de datos capturados

- Envío de datos al servidor
- Difusión del servidor y recibo de los datos en otro cliente.

La aplicación nativa para teléfonos inteligentes desarrollada para la transmisión de audio y video utilizando el marco de trabajo jWebSocket presenta las siguientes funcionalidades:

- Captura de los controladores de imagen, audio y video del dispositivo móvil.
- Codificación de los datos.
- Envío de datos al servidor
- Decodificación y visualización de los datos enviados.

2.1 Planificación del proyecto por roles.

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

Tabla # 1: Planificación del Proyecto por Roles

Rol	Responsabilidad	Nombre
Gerente (Manager)	Dirige y controla las tareas del equipo. Toma las decisiones finales. Participa en la selección de objetivos y requerimientos. Controla el progreso y da seguimiento a cada iteración. Evalúa si los objetivos son alcanzables con las restricciones	Alexander Schulze

Capítulo 2

	de tiempo y recursos presentes.	
Cliente	Participa en las tareas que involucran la lista de reserva del producto.	Alexander Schulze
Jefe de Proyecto	Controla las tareas del equipo y la toma las decisiones.	MSc. Yamila Vigil Regalado
Especialista	Es necesario que conozca a fondo el proceso para el desarrollo de software. Es una especialización que está activa, el miembro del grupo de trabajo que la desempeña siempre está ejecutándola y alcanzando un grado mayor de conocimientos en el tema, en este caso como Diseñador Gráfico.	Rebecca Schulze
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.	Alexander Rojas
Analista	Escribe las prueba de usuario y las pruebas funcionales para validar su implementación.	Alexander Rojas
Diseñador	Encargados de diseño del sistema; así como los prototipos de interfaces máximos responsables	Alexander Rojas

	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 las pruebas	Alexander Rojas
Arquitecto	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema. Ayuda en el diseño de las metáforas.	Alexander Rojas

2.2 Modelo de Negocio.

Dentro de los artefactos que genera la metodología SXP se encuentra la plantilla del Modelo Historias de Usuario del Negocio la que se muestra en la Figura # 1, donde se definen las características específicas del negocio en cuestión.

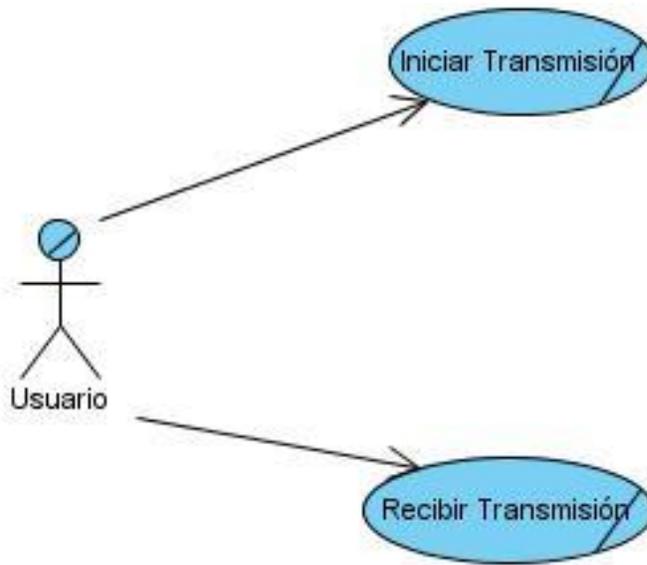


Figura # 1 Modelo de Historia de Usuario del Negocio

2.3 Lista de Reserva del Producto.

La LRP es un proceso importante para la descripción de los requisitos del sistema, con ella se llega a un acuerdo entre el cliente y los desarrolladores. Se lleva a cabo el desarrollo de un software es necesario contar con una descripción detallada de los requerimientos o requisitos, para conocer, qué debe hacer el sistema, y la forma en que debe hacerlo. Los requerimientos del software constituyen características o capacidades que el sistema debe cumplir. Estos pueden clasificarse en:

Requerimientos funcionales (RF)

Definen las funcionalidades que el sistema debe realizar, son capacidades o condiciones que el sistema debe cumplir, el punto de partida para identificar qué debe hacer el sistema. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. Con su definición se busca establecer un común entendimiento con el cliente.

Requerimientos no funcionales (RNF)

Características del sistema que hacen al software atractivo para el usuario, son fundamentales en el éxito del producto y normalmente están vinculados a requerimientos funcionales, es decir, una vez que se conozca lo que el sistema

debe hacer, se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. En resumen, son propiedades o cualidades que el producto debe tener. A continuación se presenta la Tabla # 2 que muestra la lista priorizada contenida en el LRP perteneciente a la solución propuesta.

Tabla # 2 Listas de Reservas del Producto

Prioridad	Ítem *	Descripción	Estimación	Estimado por
Muy Alta				
	1	Capturar de la señal de video y de audio desde la webcam de un ordenador.	2 semanas	Desarrollador
	2	Acceder al componente de captura de video y audio desde un dispositivo móvil utilizando PhoneGap	2 semanas	Desarrollador
Alta				
	3	Crear una capa capaz de codificar los datos y enviarlos al servidor de jWebSocket.	2 semanas	Desarrollador
	4	Crear una envoltura genérica del lado del servidor el cual decodifique esos datos y los envíe al cliente.	2 semanas	Desarrollador
	5	Realizar de un visualizador en el cual se muestre los datos decodificados, tanto para aplicaciones nativas como web.	2 semanas	Desarrollador
Media				
Baja				
Requisitos no Funcionales				

Capítulo 2

6	La aplicación está dirigida a usuarios con conocimientos básicos de informática y un correcto dominio del negocio en cuestión, por lo cual debe estar diseñada para ser de fácil manejo por el cliente.		
7	El Sistema es una aplicación web		
8	El sistema se mantendrá disponible 24 horas diarias durante los 7 días de la semana.		
9	El mantenimiento de la aplicación se realizará en tiempo de ejecución sin afectar la disponibilidad ni el rendimiento de los servicios.		
10	Garantizar la integridad y consistencia de los datos.		
11	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.		
12	Se documentará la aplicación con diferentes manuales con el objetivo de explicar el uso de la aplicación para garantizar el soporte de la misma.		
13	En el lado del servidor el lenguaje de programación a utilizar es Java, empleando el Framework jWebSocket y el IDE NetBeans 7.0.1.		

Capítulo 2

14	En el lado del cliente se hará uso de los lenguajes Java, JavaScript, HTML5 y CSS3, empleando el Framework Java Media Framework, PhoneGap para las aplicaciones nativas de SmartPhones y el IDE NetBeans 7.0.1. aplicación para garantizar el soporte de la misma.		
15	La metodología de desarrollo a seguir es SXP y para la modelación se utilizará la herramienta Visual Paradigm 3.4.		
16	La interfaz deberá ser consistente con el mundo real, de manera que los conceptos manejados sean conocidos por los usuarios, para que les sea fácil su uso y aprendizaje.		
17	Lograr un diseño amigable y de fácil uso.		
18	Todas las interfaces de usuario que se definan para el sistema respetarán los patrones de diseño establecidos para la organización.		
19	Los requisitos mínimos de hardware que tiene que tener el cliente para el correcto funcionamiento de la aplicación son: 512 MB de memoria RAM, microprocesador Pentium IV, tarjeta red 100 mbps.		
20	El cliente debe contar con un navegador que soporte el protocolo WebSocket.		

21	El servidor de aplicaciones debe tener instalado la máquina virtual de Java OpenJDK 7 y el servidor de jWebSocket.		
22	El código de la aplicación será liberado bajo la Licencia Pública General Reducida de GNU (LGPL).		
23	El código y la documentación de la aplicación deberán cumplir estrictamente las normas de calidad estipuladas por el grupo de Calisoft de la UCI.		

2.4 Historias de Usuario y Tareas de Ingeniería.

Las historias de usuarios (HU) son la forma en que se especifican los requisitos del sistema, es decir, mediante tareas describen lo que el sistema debe hacer en la metodología SXP. Las HU no deben ser descritas en más de tres líneas e idealmente es el cliente quien las redacta y prioriza por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica. Luego de esto se le suma un tiempo estimado de desarrollo que lo define el propio equipo del proyecto. En la claridad de su descripción radica el éxito del proyecto motivo por el cual el contenido que ellas abarcan debe ser concreto y sencillo.

A continuación se muestra un ejemplo de HU y sus respectivas tareas:

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Capturar la señal de video y de audio.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alexander Rojas Hernández	Iteración Asignada: 2

Capítulo 2

Prioridad en Negocio: Muy Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: La presente historia de usuario permitirá la captura de la señal de video y de audio desde la webcam de un ordenador.	
Observaciones: Para que esto sea posible el usuario debe utilizar la aplicación web “Transmisión de audio y video utilizando como servidor jWebSocket” .	
Prototipode interfaz: Ver Anexo 1 Captura de Imagenes	

Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario: HU_1
Nombre Tarea: Investigar como capturar la señal de video y de audio.	
Tipo de Tarea : Investigación	Puntos Estimados: 2 semana
Fecha Inicio: 1/11/2011	Fecha Fin: 16/11/2011
Programador Responsable: Alexander Rojas Hernández	
Descripción: Se realiza una investigación en el framework en busca de las clases encargadas de escribir la información capturada desde la webcam.	

Tarea de Ingeniería

Número Tarea: 1.2	Número Historia de Usuario: HU_1
Nombre Tarea: Capturar la señal de video y de audio.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 1/11/2011	Fecha Fin: 16/11/2011
Programador Responsable:: Alexander Rojas Hernández	
Descripción: Primeramente se lleva cabo la implementación de un método en el cual se busca en los controladores instalados en la computadora y posteriormente se inicia la captura de la señal de audio y video.	

Tarea de Ingeniería	
Número Tarea: 1.3	Número Historia de Usuario: HU_1
Nombre Tarea: Capturar la señal de video y de audio.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 1/11/2011	Fecha Fin: 16/11/2011
Programador Responsable:: Alexander Rojas Hernández	

Descripción: Se codifica la información y posteriormente se envía para el servidor.

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Acceder al componente de captura de video y audio.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alexander Rojas Hernández	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: La presente historia de usuario permitirá acceder al componente de captura de video y audio desde un dispositivo móvil utilizando PhoneGap.	
Observaciones: Para que esto sea posible el usuario debe utilizar un teléfono inteligente y ejecutar la aplicación nativa “Transmisión de audio y video utilizando como servidor websocket” .	
Prototipo de interfaz: Anexo 2: Vista Inicial de la Aplicación Nativa	

Tarea de Ingeniería	
Número Tarea: 2.1	Número Historia de Usuario: HU_2
Nombre Tarea: Investigar como acceder al componente de captura de video y audio.	

Tipo de Tarea : Investigación	Puntos Estimados: 2 semana
Fecha Inicio: 17/11/2011	Fecha Fin: 02/12/2011
Programador Responsable:: Alexander Rojas Hernández	
Descripción: Se realizó una investigación acerca del framework PhoneGap y del trabajo con SDK de Android	

Tarea de Ingeniería	
Número Tarea: 2.2	Número Historia de Usuario: HU_2
Nombre Tarea: Acceder al componente de captura de video y audio.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 17/11/2011	Fecha Fin: 02/12/2011
Programador Responsable:: Alexander Rojas Hernández	
Descripción: Se realizó la implementación para acceder a los componentes de de imágenes, video y audio del teléfono inteligente.	

Historia de Usuario

Número: HU_3	Nombre Historia de Usuario: Crear una capa codificadora de datos.	
Modificación de Historia de Usuario Número: Ninguna		
Usuario: Alexander Rojas Hernández.	Iteración Asignada: 2	
Prioridad en Negocio: Alta	Puntos Estimados: 2 semanas	
Riesgo en Desarrollo: Alto	Puntos Reales:	
Descripción: La presente historia de usuario tiene como objetivo la creación de una capa genérica capaz de codificar los datos y enviarlos al servidor de jWebSocket.		
Observaciones: Para que esto sea posible se debe capturar de forma satisfactoria los datos a transmitir.		
Prototipo de interfaz: Ninguna		

Tarea de Ingeniería	
Número Tarea: 1.3	Número Historia de Usuario: HU_1
Nombre Tarea: Capturar la señal de video y de audio.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 1/11/2011	Fecha Fin: 16/11/2011
Programador Responsable:: Alexander Rojas Hernández	

Descripción: Se codifica la información y posteriormente se envía para el servidor.

Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Crear una envoltura genérica.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alexander Rojas Hernández.	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: La presente historia de usuario tiene como objetivo la creación de una envoltura genérica del lado del servidor el cual decodifique esos datos y los envíe al cliente.	
Observaciones: Para que esto sea posible se debe capturar de forma satisfactoria los datos a transmitir.	
Prototipo de interfaz: Ninguna	

Tarea de Ingeniería	
Número Tarea: 4.1	Número Historia de Usuario: HU_4
Nombre Tarea: Crear una envoltura genérica.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 semana

Fecha Inicio: 11/01/2012	Fecha Fin: 25/01/2012
Programador Responsable: Alexander Rojas Hernández	
Descripción: Desarrollo de la capa base para la transmisión del datos desde el servidor hacia el cliente.	

Historia de Usuario	
Número: HU_5	Nombre Historia de Usuario: Realización de un visualizador.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alexander Rojas Hernández.	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 4 semana
Riesgo en Desarrollo: Alto	Puntos Reales
Descripción: La presente historia de usuario tiene como objetivo la realización de un visualizador en el cual se muestre los datos decodificados, tanto para aplicaciones nativas como web.	
Observaciones: Para que esto sea posible se debe recibir de forma satisfactoria los datos que vienen del servidor.	
Prototipo de interfaz: Ver Anexo 3 Visualizador de la Aplicación Nativa	

Número Tarea: 5.1	Número Historia de Usuario: HU_5
--------------------------	---

Nombre Tarea: Realización de un visualizador.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 semana
Fecha Inicio: 26/01/2012	Fecha Fin: 11/02/2012
Programador Responsable: Alexander Rojas Hernández	
Descripción: Se realizó la implementación de un visualizador el cual muestre el contenido enviado del servidor.	

2.5 Plan de Release.

Cuando se tiene definidas las historias de usuario es preciso crear el Plan de Release donde se indiquen que historias se crearán para cada iteración del proyecto y las fechas en las que se publicarán estas versiones.

Elaborar un Plan de Release es necesario porque:

- ✓ Ayuda al cliente y al equipo de trabajo, a decidir cuánto se debe desarrollar y cuánto tiempo se tardará antes de tener un producto entregable.
- ✓ Exterioriza sobre lo que se puede desarrollar y en qué tiempo (para que el equipo de desarrollo pueda hacerse una idea).
- ✓ Proporciona una idea general del progreso.

A continuación en la se muestra el PPlan de Release:

Tabla # 3 Plan de Release

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
---------	-----------------------------	------------------------------	----------------

Iteración 1	En esta iteración se va a desarrollar las primeras funcionalidades que permitan el continuo desarrollo de la aplicación, las mismas tienen prioridad muy alta.	HU_1, HU_2	4 semanas
Iteración 2	En esta iteración se dará continuidad a las funcionalidades correspondientes para el desarrollo de la aplicación, las mismas tienen prioridad alta.	HU_3, HU_4, HU_5	8 semanas

2.6 Arquitectura de Software.

Con la evolución de los años se han ido desarrollando y descubriendo guías generales de aprendizaje para el desarrollo de programas. A estas guías se les nombrado de Arquitectura de Software, pues estas modelan la forma de desarrollar un determinado software informático. Esta, también dominada Arquitectura lógica está integrada por un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencias necesarias para guiar el desarrollo de un software. Establece las normas para que un equipo de desarrollo trabaje sobre una misma línea común permitiendo alcanzar los objetivos específicos de un sistema informático. A la hora de desarrollar un software se selecciona o se crea una

arquitectura que rige el desarrollo del mismo. Algunas de las diferentes arquitecturas de software que existen son:

- ✓ Monolítica: Donde el software se estructura en grupos funcionales muy acoplados.
- ✓ Cliente-Servidor: Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
- ✓ Arquitectura de tres niveles: Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

Para el desarrollo de las aplicaciones se utilizó la arquitectura en capas. Se tiene una capa de presentación donde se encuentran los distintos clientes que inician los procesos de transmisión. En la capa de negocio se encuentra el servidor el cual procesa y distribuye todos los datos enviados hacia los clientes finales.

Fue seleccionado la arquitectura en capas y no otra ya que permite hacer mucho más extensible y adaptable la estructura del proyecto en general.

2.7 Diseño con Metáfora.

El Diseño con Metáforas describe como debería funcionar el sistema y define que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. No es más que el diseño de la solución, lo cual genera el artefacto conocido como Modelo de Diseño cuyo objetivo es describir los elementos físicos del sistema mediante paquetes para su posterior implementación.

A continuación se representa en la Fig2 y Fig3 el diagrama de paquetes de las aplicaciones:

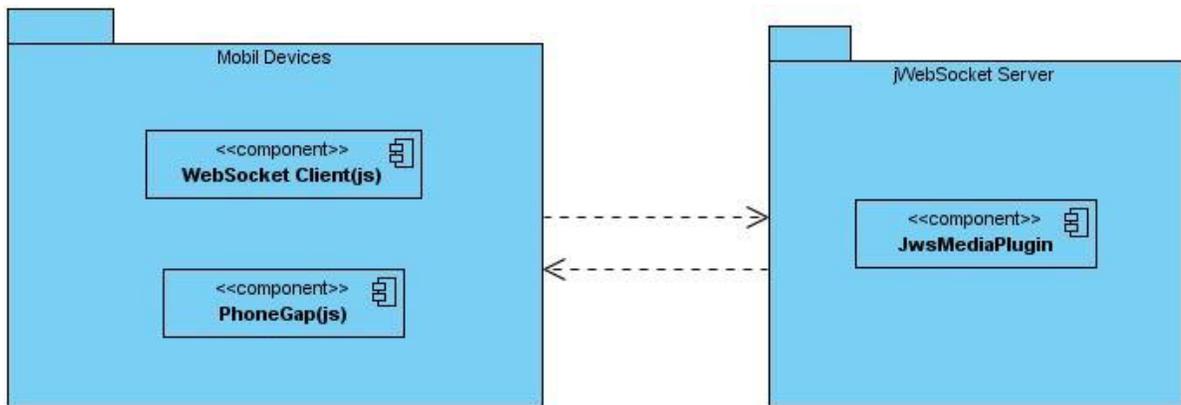


Figura # 2 : Diagrama de paquetes Aplicación Nativa.

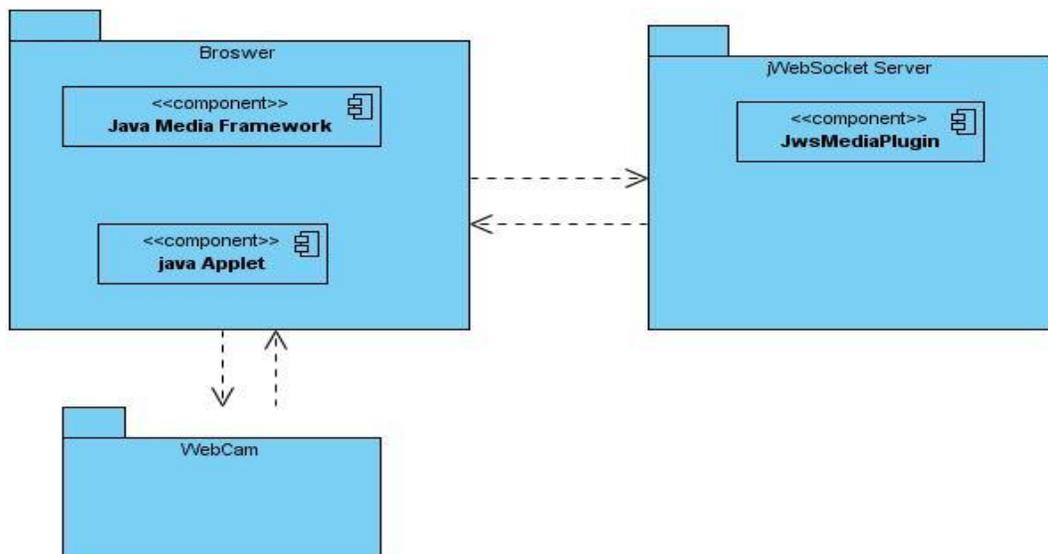


Figura # 3:Diagrama de paquetes Aplicación Estacionaria.

2.8 Diagrama de componentes.

Los diagramas de componentes modelan la estructura del software en un conjunto de componentes, agrupados en ocasiones por paquetes. Muestran la organización y las dependencias lógicas entre un conjunto de paquetes y componentes de software, siendo éstos últimos componentes fuentes, binarios o ejecutables. Los componentes de software presentan tipo, que indica si son útiles en tiempo de compilación, enlace o ejecución. Además estos diagramas de componentes cubren

la vista de la implementación estática y se relacionan con los diagramas de clases ya que un componente suele tener una o más clases, interfaces o colaboraciones, dejando así una traza con el modelo de diseño.

En el desarrollo de la aplicación estacionaria el marco de trabajo Java Media Framework es utilizado como marco de trabajo para el manejo de los procesos de captura, codificación, transmisión y decodificación de audio y video a bajo nivel de procesamiento.

A continuación se muestra en la Fig4 una vista general del modelo de implementación expresado en diagramas de componentes.

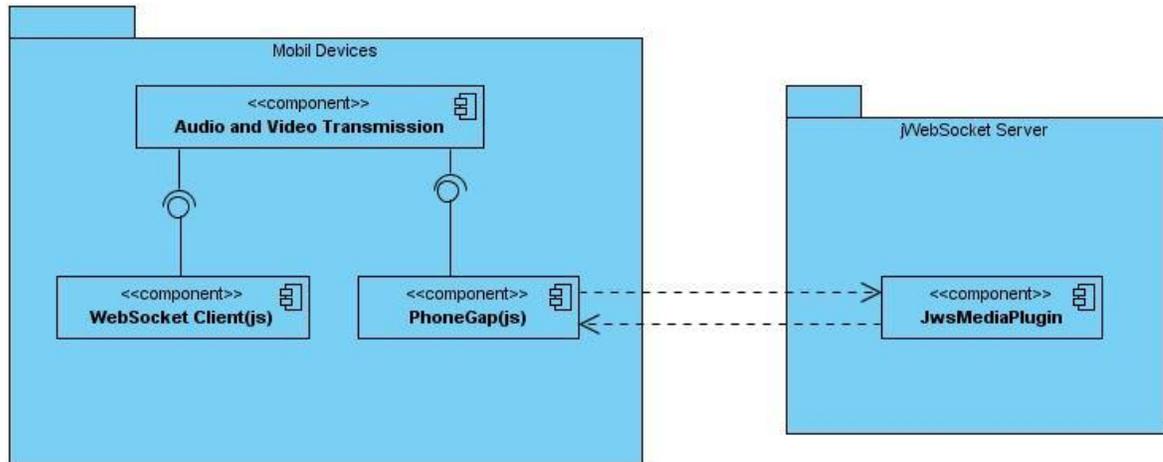


Figura # 4: Diagrama de componentes de la Aplicación Nativa

El uso del marco de trabajo PhoneGap es el punto de acceso a los distintos controladores del dispositivo móvil permitiendo la captura y el procesamiento de la imagen, el audio y el video. El mismo puede ser extendido e integrado con otras librerías que trabajan con este tipo de dispositivos. A continuación en la fig5 se hace una descripción en forma diagrama de componentes.

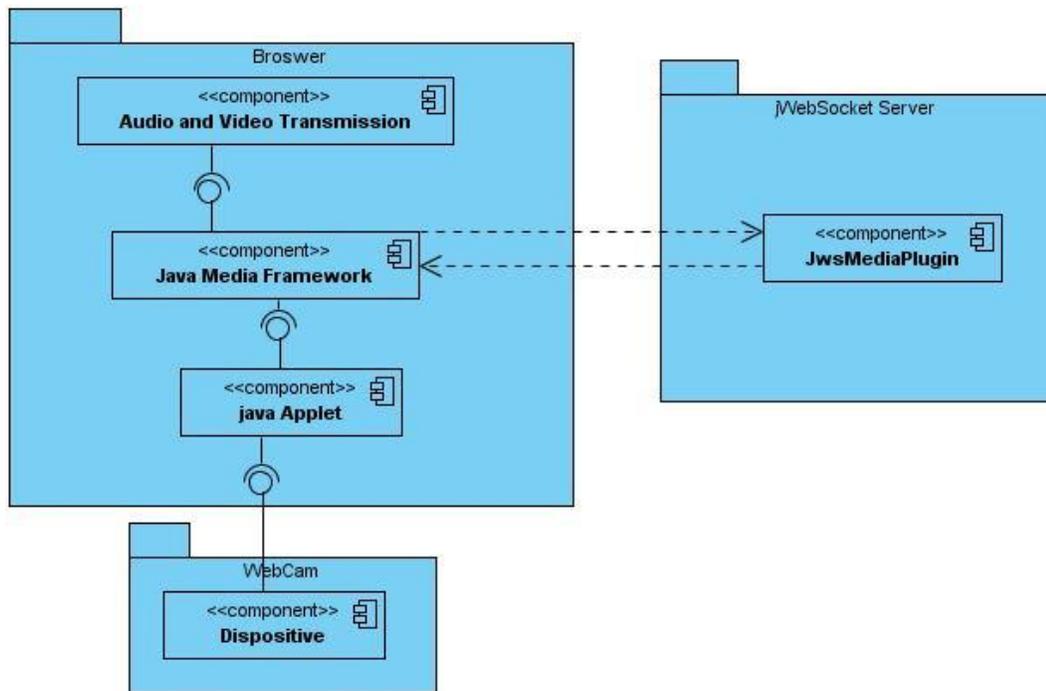


Figura # 5 Diagrama de componentes de la Aplicación Estacionaria

La utilización de jWebSocket como base de difusión para la tecnología streaming provee a los usuarios la creación de aplicaciones con altos niveles de seguridad en la entrega de los paquetes, garantizando una confiabilidad en el servicio en entornos donde se necesiten mucha concurrencia de usuarios y alta velocidad en el servicio. También se demuestra las altas prestaciones que brinda el protocolo WebSocket para este flujo de transmisión.

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.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Introducción

En el presente capítulo se documenta la 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 además de las pruebas se dan a conocer los resultados obtenidos hasta el momento.

3.1 Diagrama de despliegue

El diagrama de despliegue es un artefacto que modela la arquitectura en tiempo de ejecución de un sistema. 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:

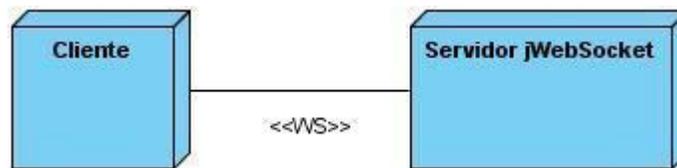


Figura # 6 Diagrama de Despliegue

A continuación se brindan elementos descriptivos de cada nodo del diagrama de despliegue.

Cliente: El cliente se conectará por medio del 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 es el nodo que gestiona las solicitudes que vienen del cliente, este es el encargado de procesar y difundir todos los paquetes hacia los clientes finales.

3.2 Validación de la Solución Propuesta

Para realizar la validación de la presente investigación, la cual tiene como objetivo desarrollar una aplicación web y una nativa con el marco de trabajo jWebSocket que garantice mayor integridad de la información en las aplicaciones que transmiten audio y video en tiempo real en la Web, se decide trazar una estrategia de validación basada en las siguientes etapas.

La primera etapa consiste en realizar dos aplicaciones demostrativas en las cuales se realice el proceso de transmisión de audio y video utilizando el marco de trabajo jWebSocket. La siguiente etapa consiste en diseñar y ejecutar un conjunto de casos de pruebas funcionales a las historias de usuarios definidas para la solución propuesta, con el objetivo de mostrar su correcto funcionamiento.

Otra de las etapas que se lleva a cabo es el proceso de certificación de calidad de software, realizado por el grupo de calidad del Centro de Desarrollo de la Facultad Regional “Mártires de Artemisa”. Dicho grupo es el encargado de certificar la funcionalidad, estandarización y limpieza del código así como los artefactos documentales generados teniendo en cuenta la metodología de desarrollo SXP seleccionada en la presente investigación.

Por último, se define la utilización del método de valoración del cliente el cual permite la evaluación de las aplicaciones desarrolladas por parte de Alexander Schulze, líder internacional de la comunidad jWebSocket y arquitecto principal del proyecto. En esta etapa de validación se obtiene un aval de Alexander Schulze como experto internacional en la temática, certificando de esta forma la capacidad de integración de la solución desarrollada con el marco de trabajo jWebSocket a

nivel internacional, así como la usabilidad real para los clientes potenciales y desarrolladores de esta comunidad.

A continuación se realiza una descripción más detallada de cada una de las etapas de la estrategia de validación trazada a las cuales fue sometida la librería desarrollada.

3.2.1 Aplicaciones Demostrativas

Con el objetivo de demostrar el cumplimiento de las funcionalidades del proceso de transmisión de audio y video se decidió desarrollar dos aplicaciones demostrativas, una estacionaria para la web y otra nativa para los dispositivos móviles.

La aplicación demostrativa de transmisión de audio y video, realizada para la Web utilizando Java Media Framework y otra nativa para teléfonos inteligentes utilizando PhoneGap, ambas con el marco de trabajo WebSocket permiten el envío de datos codificados hacia el servidor y posteriormente el recibo, decodificación y visualización de los mismos. En el caso de la aplicación nativa se accede a los componentes de captura del dispositivo móvil como la imagen, el video y el audio, se codifican los datos y se envían hacia el servidor.

En la aplicación estacionaria el proceso de captura es desde la webcam y se realiza un proceso similar. Estas aplicaciones garantizan tiempo real utilizando el protocolo WebSocket y un trabajo bidireccional de alta precisión, se garantiza la seguridad e integridad de la información ya que el servidor controla que el cliente esté conectado y enviando datos.

3.3 Casos de Pruebas

Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al cliente. Las mismas proporcionan 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. 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 web y nativa de transmisión de audio y video con el objetivo de verificar el buen funcionamiento de la misma.

- ✓ “Casos de Prueba para la HU Capturar señal de audio y video”.

Caso de Prueba de Aceptación	
Código Caso de Prueba: jws-1-1	Nombre Historia de Usuario: Capturar señal de audio y video
Nombre de la persona que realiza la prueba: Alexander Rojas Hernández	
Descripción de la Prueba: Esta prueba consiste en la comprobación de la instalación en la pc de los distintos controladores de audio y video.	
Condiciones de Ejecución: Es necesario que la pc soporte los distintos controladores necesarios para el proceso de captura.	
Entrada / Pasos de ejecución: El usuario debe observar en el navegador no se muestra un mensaje de error de que faltan los controladores.	
Resultado Esperado: Reconoce que se encuentran instalados los controladores.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: jws-1-2	Nombre Historia de Usuario: Capturar señal de audio y video
Nombre de la persona que realiza la prueba: Alexander Rojas Hernández	
Descripción de la Prueba: Esta prueba consiste en obtener el flujo de datos capturados desde el ordenador.	

Condiciones de Ejecución: Para poder realizar la prueba se tiene que tener instalada una webcam para el video y un micrófono para el audio, además de los distintos controladores instalados.
Entrada / Pasos de ejecución: Para verificar esto el usuario tiene que observar si cuando se ejecuta el applet obtiene alguna imagen
Resultado Esperado: : Obtención de la captura se muestra en el applet view del netbeans, asi como también en el navegador.
Evaluación de la Prueba: Satisfactoria.

- ✓ “Casos de pruebas para la HU Acceder al componente de captura de audio y video“

Caso de Prueba de Aceptación	
Código Caso de Prueba: jws-2-1	Nombre Historia de Usuario: Acceder al componente de captura de audio y video.
Nombre de la persona que realiza la prueba: Alexander Rojas Hernández	
Descripción de la Prueba: Esta prueba consiste en verificar que el dispositivo móvil soporte el trabajo con el marco de trabajo PhoneGap	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que poseer un teléfono inteligente que soporte la utilización del marco de trabajo PhoneGap para acceder a los distintos controladores del dispositivo.	
Entrada / Pasos de ejecución: Para verificar esto el usuario debe ejecutar la aplicación del teléfono y comprobar que se ejecuta de forma correcta.	
Resultado Esperado: Un mensaje al iniciar la aplicación.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: jws-2-2	Nombre Historia de Usuario: Acceder al componente de captura de audio y video.
Nombre de la persona que realiza la prueba: Alexander Rojas Hernández	
Descripción de la Prueba: Esta prueba consiste en capturar la señal de audio y video desde un dispositivo móvil.	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que poseer un teléfono inteligente que soporte la utilización del marco de trabajo PhoneGap para acceder a los distintos controladores del dispositivo.	
Entrada / Pasos de ejecución: Para verificar esto el usuario debe ejecutar la aplicación del teléfono y presionar el botón correspondiente al proceso de captura que desee.	
Resultado Esperado: La imagen, video o audio es visualizado en el dispositivo.	
Evaluación de la Prueba: Satisfactoria.	

✓ "Casos de pruebas para la HU Crear capa codificadora de datos".

Caso de Prueba de Aceptación	
Código Caso de Prueba: jws-3-1	Nombre Historia de Usuario: Crear capa codificadora de datos
Nombre de la persona que realiza la prueba: Alexander Rojas Hernández	
Descripción de la Prueba: Esta prueba consiste en lograr de que los datos se codifiquen y se envíen de forma correcta en el flujo de transmisión.	
Condiciones de Ejecución: Para que esta prueba se ejecute es necesario que el servidor se mantenga en ejecución.	
Entrada / Pasos de ejecución: Para la verificación de esta prueba el flujo de captura de la señal debe ser constante.	

Resultado Esperado: Se debe mantener un constante flujo de los paquetes hacia el servidor

Evaluación de la Prueba: Satisfactoria.

“Casos de pruebas para la HU Crear una envoltura genérica”.

Caso de Prueba de Aceptación	
Código Caso de Prueba: jws-4-1	Nombre Historia de Usuario: Crear una envoltura genérica
Nombre de la persona que realiza la prueba: Alexander Rojas Hernández	
Descripción de la Prueba: Esta prueba consiste en verificar que los datos están llegando codificados al servidor.	
Condiciones de Ejecución: Para que esta prueba se ejecute es necesario que el servidor se mantenga en ejecución.	
Entrada / Pasos de ejecución: Para la verificación de esta prueba el flujo de envío se tiene que mantener.	
Resultado Esperado: La transmisión de audio y video se mantiene constante.	
Evaluación de la Prueba: Satisfactoria.	

“Casos de pruebas para la HU Realización de un visualizador”.

Caso de Prueba de Aceptación	
Código Caso de Prueba: jws-5-1	Nombre Historia de Usuario: Realización de un visualizador
Nombre de la persona que realiza la prueba: Alexander Rojas Hernández	

Descripción de la Prueba: Esta prueba consiste en verificar que los datos están llegando al otro cliente.

Condiciones de Ejecución: Para que esta prueba se ejecute es necesario que el servidor se mantenga en ejecución y que los datos enviados lleguen al cliente final.

Entrada / Pasos de ejecución: Para la verificación de esta prueba se debe mostrar los datos decodificados.

Resultado Esperado: Se muestra en el navegador los paquetes recibidos.

Evaluación de la Prueba: Satisfactoria.

3.4 Certificación de Calidad de Software

El grupo de calidad del Centro de Desarrollo de la Facultad Regional “Mártires de Artemisa” es el encargado de certificar la calidad de los artefactos documentales generados teniendo en cuenta la metodología de desarrollo SXP seleccionada en la presente investigación. Este proceso fue llevado a cabo por la asesora de calidad del Centro, Ing. Maidel Ojeda Castro, teniendo en cuenta los siguientes artefactos:

- Plantilla Modelo de Historia de Usuario del Negocio
- Plantilla lista de reserva del producto (LRP)
- Plantilla de Historia de Usuario
- Plantilla de Arquitectura de Software SXP
- Plantilla Tarea de Ingeniería
- Plantilla de Releases
- Plantilla Estándar de Código
- Plantilla Caso de Prueba de Aceptación
- Plantilla Manual de Usuario
- Plantilla Manual de Instalación

- Plantilla Manual de Desarrollo

Por otra parte, se puso a disposición del Ing. Domma Moreno Dager, asesor de tecnología del Centro, el código fuente del módulo desarrollado. Este fue sometido a un proceso de revisión donde se hizo énfasis en la calidad, estandarización y limpieza del código así como en el correcto funcionamiento de la librería teniendo en cuenta los requisitos funcionales definidos por el cliente.

3.4.1 Valoración del Cliente

La presente investigación fue sometida al criterio del cliente Alexander Schulze, considerado un usuario con buena experiencia en esta temática ya que es el principal arquitecto e impulsor del proyecto jWebSocket y líder internacional de esta comunidad. Con este objetivo fueron incorporados al servidor SubVersion de jWebSocket internacional el código fuente y los manuales de Usuario, Desarrollador y Administración en inglés que garantizan un mayor entendimiento de la librería. De esta forma el cliente certifica la capacidad de integración de la solución desarrollada con el marco de trabajo jWebSocket a nivel internacional, así como la usabilidad real para los clientes potenciales y desarrolladores de esta comunidad.

3.5 Resultados Obtenidos

Por el resultado satisfactorio de los casos de prueba realizados a la aplicación web y una nativa de transmisión de audio y video queda disponible su versión 1.0. Se obtuvo un sistema que cumple con las principales especificaciones para permitir la utilización de la tecnología streaming utilizando el protocolo Websocket.

3.6 Funcionalidades Obtenidas

Entre las principales funcionalidades obtenidas en la aplicaciones desarrolladas para la transmisión de audio y video se pueden mencionar:

- ✓ Se realiza la captura de audio y video.
- ✓ Se accede de forma satisfactoria a los controladores de captura de un dispositivo móvil.
- ✓ Se codifican los datos a bajo nivel para su envío.

- ✓ El servidor realiza la difusión correcta de los paquetes.
- ✓ Se reciben los paquetes de forma correcta.
- ✓ Se visualizan la información.

3.7 Aporte Social y Económico

Capturar y transmitir audio y video en tiempo real constituye al igual que el resto de las funcionalidades un elemento trascendental para la educación a distancia. A través de aplicaciones que contengan esta implementación se pueden transmitir conferencias, reuniones de trabajo, intercambio de distintos temas y áreas del conocimiento, de una manera personalizada y con alta calidad en la visualización de los videos y el audio. Brindar la posibilidad de intervenir en una discusión de criterios con compañeros de estudio, no solo desde una PC sino camino a la universidad, sentado en un parque o camino a casa, proporciona facilidad y flexibilidad en el aprendizaje. La flexibilidad en las vías y formas de lograr un proceso de aprendizaje constituye un ente importante, pues los individuos de la sociedad actual del conocimiento se enfrentan de manera permanente a un proceso de asimilación de mayores y nuevos contenidos cada día.

Además de estas funcionalidades específicas de WebSocket para el desarrollo de aplicaciones web educativas, el marco cuenta con elementos generales que favorecen además su uso para el sector educacional. El nivel de escalabilidad de las conexiones soportadas por los servidores WebSocket permiten alta concurrencia de usuarios en todas las actividades colaborativas y educativas a distancia.

El desarrollo de esta aplicación no supone grandes gastos, ya sea monetario, de tiempo o material de oficina. Además las tecnologías utilizadas para el desarrollo de las aplicaciones son basadas en software libre. Por lo que haciendo un balance del costo-beneficio de la investigación y atendiendo al resultado que esta tributa se puede concluir que es factible de realizar, garantizando el aumento de la calidad y en los servicios.

Conclusiones del capítulo

En este capítulo se realizó el diagrama de despliegue que indica la situación física de los componentes lógicos desarrollados. Además se realizaron casos de pruebas que guiaron la calidad de la aplicación desarrollada. De esta manera los casos de prueba mostraron que las funciones de la aplicación son operativas y que se produce un resultado correcto. Demostrándose así que el sistema se encuentra en óptimas condiciones para el proceso de colaboración en línea en tiempo real.

Conclusiones

Con la realización del presente trabajo de diploma se dio respuesta a cada una de las preguntas científicas planteadas, obteniéndose de manera general las siguientes conclusiones:

- La realización de la fundamentación teórico-metodológica permitió conceptualizar el proceso de transmisión de audio y video que brinda a los usuarios la posibilidad de una comunicación en tiempo real usando el protocolo Websocket.
- Las soluciones actuales que hacen uso de la tecnología streaming en la Web permiten el trabajo colaborativo, video conferencias pero no trabajan en tiempo real utilizando el protocolo WebSocket.
- Se desarrolló una aplicación web y una nativa para la transmisión de audio y video utilizando el protocolo WebSocket en aplicaciones web desarrolladas con el marco de trabajo jWebSocket y a su vez garantiza un posible aumento del nivel de usabilidad del marco de trabajo en el desarrollo de este tipo de aplicaciones.
- Se comprobó la capacidad y potencialidad de la aplicación web y de la nativa para la transmisión de audio y video en tiempo real usando el protocolo Websocket a través de la estrategia de validación trazada para la presente investigación, garantizando que su código fuente y documentación poseen la calidad requerida, además puede ser integrada al entorno de desarrollo de jWebSocket y utilizada por su comunidad.

RECOMENDACIONES

Se recomienda para posteriores versiones de la aplicación web y la nativa para transmisión de audio y video:

- El trabajo en el cliente de Java Media Framework para potenciar el proceso de visualización de la transmisión.
- Realizar soporte para la transmisión de audio y video en servidores de streaming que son utilizados en la universidad.
- Seguir potenciando el trabajo con PhoneGap a medida que se siga actualizando para un mejor trabajo con el marco de trabajo WebSocket.

BIBLIOGRAFÍA REFERENCIADA

Allen, Sarah, Graupera, Vidal and Lundrigan, Lee. 2010. *Pro Smartphone Cross-Platform Development*. 2010.

Apache Software Foundation. 2011. Apache Subversion. *Apache Subversion*. [Online] 12 10, 2011. [Cited: 12 10, 2011.] <http://subversion.apache.org/>.

Austerberry, David. 2008. *The technology of video and audio streaming*. 2008.

Clevenger, Nathan. 2011. *IPad in the Enterprise: Developing and Deploying Business Applications*. 2011.

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

Felker, Donn and Dobbs, Joshua. 2010. *Android Application Development For Dummies*. 2010.

Figuroa, Roberth G, Solís, Camilo J and Cabrera, Armando A. 2011. Entorno Virtual de Aprendizaje. [Online] 2011. [Cited: 12 13, 2011.]

Global Mobile Broadband Traffic Report. **MobileTrends, Allot. 2011.** 2011, Allot MobileTrends.

Grupo Soluciones Innova. [Online] [Cited: 12 11, 2011.]

<http://www.rational.com.ar/herramientas/roseenterprise.html>.

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

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

2009. Introduction to video on the Internet. *Computer e Commerce*. [Online] 2009. <http://www.computerecommerce.com/web-video-primer.aspx>.

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-lle-no-de-ventajas/>, 2011.

Bibliografía

- Laguna Torres, Angel. 2009.** *Propuesta de Servidor Streaming de software libre para la captura y transmisión de video y sonido digital.* 2009.
- Morales Venegas, Hernán and otros. 2005.** *Streaming.* 2005.
- Murphey, Rebecca. 2010.** *jQuery Fundamentals.* s.l. : Autoedición, 2010.
- NetBeans. 2011.** *NetBeans.* [Online] Oracle Corporation, 2011.
<http://netbeans.org/features/index.html>.
- Nourie, Dana. 2005.** *Oracle Sun Developer Network (SDN).* [Online] Oracle Corporation, Mayo 24, 2005. <http://java.sun.com/developer/technicalArticles/tools/intro.html>.
- P. Miller, Frederic, F. Vandome, Agnes and McBrewster, John. 2010.** *Java Media Framework.* 2010.
- Pérez Agüera, José Ramón, Sánchez Jiménez, Rodrigo and Caldera Serrano, Jorge. 2004.** *ADAPTACIÓN DE TECNOLOGÍAS STREAM Y XML A CENTROS DE DOCUMENTACIÓN EN TELEVISIÓN.* 2004.
- Alvarez, Miguel Angel. 2009.** s.l. : <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>, 2009.
- Quintero, Ortiz, et al. 2006.** *EVALUACION DE SERVIDORES DE STREAMING DE VIDEO ORIENTADO A DISPOSITIVOS MOVILES.* 2006.
- Raasch, Jon. 2012.** *Smashing WebKit.* 2012.
- Raphael Eyvindr, Lennox. 2011.** *Sencha Touch.* 2011.
- RapidSVN. 2011.** RapidSVN. [Online] 2011. [Cited: 12 13, 2011.]
<http://www.rapidsvn.org/>.
- Real Time Video and Audio in the World Wide Web.* **Chen, Zhigang, et al. 2005.** 2005.
- Reid, Jon. 2012.** *JQuery Mobile.* 2012.
- Schwaber, Ken and Beedle, Mike. 2011.** *Agile Software Development with SCRUM.* s.l. : Prentice Hall, 2011. 0130676349.
- SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE.* **Peñalver, G, Meneses, A and García, S. 2010.** Chile : 1er congreso Iberoamericano de Ingeniería de Proyectos, 2010.
- Tapper, Jeff, et al. 2008.** *Training from the Source.* 2008.

Bibliografía

- team, TortoiseSVN. 2011.** TortoiseSVN. [Online] 2011. [Cited: 12 13, 2011.]
<http://tortoisesvn.net/>.
- Universe., "RealNetworks Inc." Funding. 2011.** RealNetworks-Inc-Company-History.
[Online] 2011.
Visual Paradigm. [Online] [Cited: 12 11, 2011.] <http://www.visual-paradigm.com>.
- Wells, Don. 2009.** Extreme Programming: A gentle introduction. *Extreme Programming*.
[Online] 07 28, 2009. [Cited: 02 17, 2012.] <http://www.extremeprogramming.org/>.
- Wes, Simpson. 2008.** *Video over IP: IPTV, Internet video, H.264, P2P, web RV, and streaming.* . 2008.
What is Streaming? **Ozer, Jan. 2011.** 2011, Streaming Media.
- YÁGÜEZ, JAVIER. 2005.** *Librounimultidifu.* 2005.
- Zhu, Ce, Li, Yue nan and Niu, Xiamu. 2010.** *Streaming Media Architectures, Techniques, and Applications: Recent Advances.* 2010.

BIBLIOGRAFÍA CONSULTADA

¿Qué es un 'framework'? **Sánchez, Jordi. 2006.** Valencia :

<http://jordisan.net/blog/2006/que-es-un-framework>, 2006.

Broulik, Brad. 2011. *Pro JQuery Mobile*. 2011.

Comparación de Frameworks en Javascript. **Tavárez, David. 2009.** s.l. :

<http://www.maestrosdelweb.com/editorial/comparacion-frameworks-javascript/>, 2009.

Eguíluz Pérez, Javier. 2009. *CSS avanzado*.

http://www.librosweb.es/css_avanzado/capitulo5/el_framework_yui.html : s.n., 2009.

Extjs. 2009. www.extjs.es. [Online] 2009. [Cited: 12 10, 2011.] <http://www.extjs.es>.

Proyecto jQuery. 2010. License. *jQuery*. [Online] 2010. [Cited: 12 8, 2011.]

<http://jquery.org/license/>.

Qué es jQuery, para qué sirve y qué ventajas tiene el utilizar este framework Javascript.

Sencha. 2011. Ext JS 4 JavaScript Framework for Rich Apps in Every Browser. *Sencha*.

[Online] 2011. [Cited: 12 14, 2011.] <http://www.sencha.com/products/extjs/>.

Carga de clases dinámicamente con ExtJS 4. **Villa, Crysfel. 2011.** s.l. :

<http://www.quizpot.com/2011/10/carga-de-clases-dinamicamente-con-extjs-4/>, 2011.

Glosario de Términos

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

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

CSS (Cascading Style Sheets): Hojas de Estilo en Cascada

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

DHTML (Dynamic HTML): HTML Dinámico).

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

GPL: Licencia Pública GNU.

GNU General Public License

HTTP (Hypertext Transfer Protocol): Protocolo de Transferencia de Hipertexto.

TCP (Transmission Control Protocol): Protocolo de Control de Transmisión.

UDP(User Datagram Protocol): Protocolo de Datagrama de Usuario

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

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

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

SysML (Systems Modeling Language): Lenguaje de Modelo de Sistemas

BPMN (Business Process Modeling Notation): Notación para el Modelado de Procesos de Negocio)

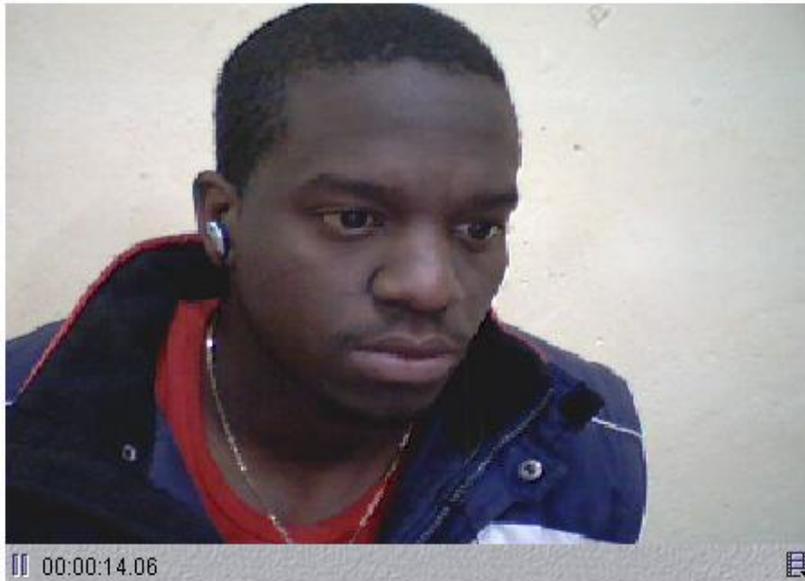
XMI (XML Metadata Interchange): Intercambio de Metadatos XML.

SDK(Software Development Kit): Kit de Desarrollo de Software.

Anexos

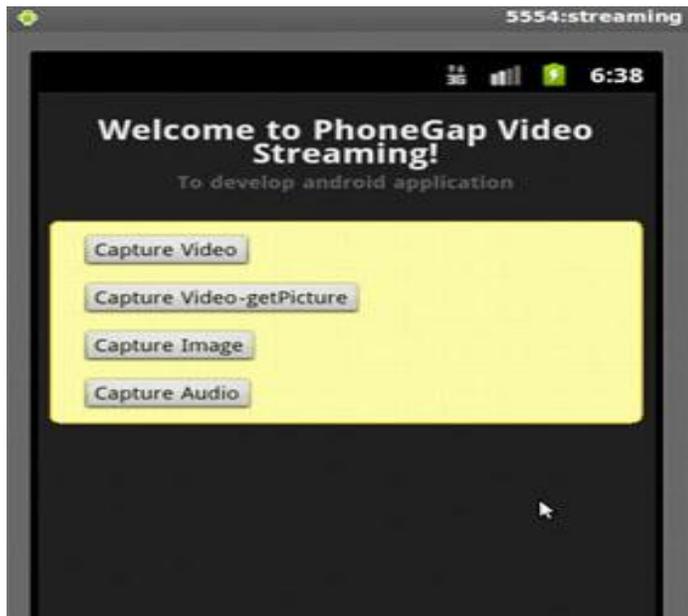
Anexo 1: Captura de Imágenes.

JMF Media Capture



Generated by NetBeans IDE

Anexo 2: Vista Inicial de la Aplicación Nativa



Anexo 3 Visualizador de la Aplicación Nativa

