UNIVERSIDAD DEL BÍO-BÍO

Análisis de algoritmos criptográficos en una red hibrida P2P.

10 de abril de 2014
Concepción - Chile

Nombre Alumno: Heber Gálvez Ojeda.
Profesor Guía: Patricio Galdames Sepúlveda.
Título al que se opta: Ingeniería Civil Informática.

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil Informática. El proyecto titulado "Análisis de algoritmos criptográficos en una red distribuida P2P", tiene como principal objetivo investigar y realizar comparativas en los tiempos de transmisión de datos digitales – vídeo -, al aplicar técnicas criptográficas a los datos transmitidos por algún protocolo como UDP o TCP, determinando el impacto que genera el hacer uso de estos algoritmos.

Consideramos como escenario de aplicación un sistema de transmisión de video almacenado, asistido por una red peer-to-peer. Para ello desarrollamos pruebas de rendimiento considerando tres entidades. La primera el servidor de video quien transmite el video encriptado a la red y mantiene registro de los clientes que pueden acceder a los videos. La segunda entidad es opcional, consiste de un peer que tan solo realiza el reenvío del video a un peer final. Este peer intermedio no visualiza el video y tan solo apoya al servidor en el proceso de difusión del video. Finalmente la tercera entidad que corresponde al computador del cliente quien descifra y reproduce el video. Todo el sistema de transmisión fue desarrollado con el lenguaje de programación Java y complementada con la aplicación cliente/servidor "VLC Media Player", la cual nos da la posibilidad de emitir y recibir el contenido multimedia a través de la red.

Las conclusiones obtenidas pasan por la mejor opción de cifrado usando el algoritmo AES, independiente del tipo de cifrado. Además se corrobora que el uso de criptografías de llave asimétrica incrementa el tiempo empleado para llevar a cabo las tareas de cifrado y descifrado.

Abstract

This project was carried out to fulfill the requirements of the University of Bio-Bio to obtain the professional title in Computer Engineering. This project, titled as "Analysis of cryptographic algorithms in a hybrid P2P network" pursues to measure and compare the overhead of using cryptographic techniques to provide confidentiality and integrity of a stored video being broadcasting through a P2P network.

Without losing generality, we measure performance of our system considering a test bed with three entities. A server encrypts and transmits a selected video to the P2P network. This server is also responsible to provide client authentication and video access control. The second entity is optional and it consists of a peer that simply forwards the video coming from the server towards other peers of the P2P networks. This intermediate peer does not display the video and its only purpose is to help the server to disseminate the video to others. Finally, the third entity is the client computer who receives the encrypted video, decodes it and plays it. All transmission system was developed with the Java programming language and complemented by the "VLC Media Player" application. VLC gives us the flexibility to send and receive multimedia content through the network using either TCP or UDP as the transport protocols.

Our study shows that regardless of the mode of encryption algorithm, AES added the lowest time overhead. We also confirmed that the execution of a asymmetric key cryptography technique like RSA to sign the video stream increases significantly the delay experienced by the final clients. We also measured the impact of UDP and TCP on the delay experienced by the clients.

Índice General

Índice Tablas

Índice Figuras

1 INTRODUCCIÓN

Multicasting es una técnica que opera a nivel de capa de red que busca la transmisión eficiente de datos desde un emisor a múltiples receptores ubicados en distintos puntos de una red. Esta técnica requiere que los routers que conforman esta red tengan capacidades de Multicasting. Por diversos motivos esta capacidad no se ha masificado en la Internet lo que ha motivado la implementación de Multicasting a nivel de capa de Aplicación.

Un escenario natural para la implementación de Multicasting a nivel de capa de aplicación es en una red peer-to-peer (P2P), en la cual los pares o computadores que conforman la red brindan sus recursos de hardware y de red a otros pares. Uno de los problemas que surgen al implementar Multicasting en una red P2P es garantizar la confidencialidad e integridad de los datos que son reenviados por los pares hacia otros pares de la red. En la literatura se propone el uso de técnicas criptográficas para mitigar los mencionados problemas de seguridad.

En este documento se presentan los resultados de una investigación enfocada al uso de distintos métodos criptográficos aplicados a la transmisión de video en una red P2P, con el objetivo de determinar el impacto existente en el uso de estos algoritmos, considerando como parámetro la medición del tiempo empleado para el cifrado de datos.

Se asume la existencia de un servidor el cual almacena y distribuye los archivos de video para cierto grupo de clientes, que pueden estar ubicados en diferentes zonas, como por ejemplo sus casas, o en este caso, en los laboratorios disponibles en nuestra universidad. Nuestra idea es que estos clientes pagan una cuota de membresía que les permite visualizar un video seleccionado. Es por ello que solo aquellos clientes autorizados son los únicos que pueden visualizar un video.

Bajo este planteamiento, la primera solución sería que los usuarios recibieran directamente los datos de transmisión desde el servidor, pero se ha probado que los recursos limitados con los que dispone el servidor, como son los de hardware y de la red, hacen que el flujo de datos digitales solo le permite servir a un grupo limitado de usuarios .

Es por ese motivo que se ha propuesto desarrollar e implementar esta investigación bajo una red P2P o Peer to Peer. Esta red se conforma con los mismos computadores de los usuarios, los cuales brindan sus recursos de hardware y de redes a sus pares, de esta forma se puede reducir la carga de trabajo y uso de red que afecta en un principio al servidor.

Durante los primeros capítulos de este documento, se comienza por detallar todo los aspectos asociados al lugar y contexto en donde se desarrolló el proyecto, especificando aspectos principales del centro de estudio.

Luego se establece una definición más completa del proyecto, en donde se indican los objetivos generales y específicos. Continuando con la especificación de requerimientos de software.

Luego de la especificación del proyecto, se da inicio a los capítulos introductorios asociados a la criptografía, comenzando con establecer algunos conceptos generales, explicando de forma clara los algoritmos empleados, técnicas existentes, etc.

Finalmente se presenta los resultados obtenidos en las pruebas de transferencia de video, usando los diversos algoritmos criptográficos, agregando también, las conclusiones respectivas a cada metodología utilizada. La descripción de aquellos capítulos más relevantes se presenta a continuación.

Cap2. En este capítulo se abarcan todos los aspectos asociados a la institución en donde se realiza esta investigación. Se definen los aspectos asociados por ejemplo la misión, visión y objetivos que existen en la entidad.

Cap3.Se plantean todos los objetivos del proyecto, más las herramientas necesarias para llevarlo a cabo.

Cap4. Se detallan todos los elementos que la aplicación requiere para que se puedan cumplir las expectativas planteadas.

Cap5. Se presenta al lector, una introducción respecto a la criptografía, abordando los temas más relacionados al objetivo de esta investigación.

Cap6. Se realiza una explicación del trabajo que se llevó a cabo, demostrando los elementos que fueron utilizados y necesarios para la construcción de los programas.

Cap7. Se muestran los resultados más relevantes y consistentes que fueron obtenidos a partir de las pruebas realizadas.

Cap8. Se presentan los requerimientos o parámetros necesarios que fueron abordaros y que permitieran la correcta configuración de los distintos programas creados y utilizados.

Cap9. Se establecen las conclusiones obtenidas a partir de todo el trabajo e investigación realizada.

Cap10. Se presenta la bibliografía ocupada para la investigación y desarrollo de este proyecto.

Cap11. Se presentan las descripciones y resultados (económicos) que sirven de base para determinar la viabilidad en el desarrollo del proyecto.

Cap12. Se realiza el análisis previo al desarrollo del proyecto como tal. Se muestran las herramientas como diagrama de flujo de datos, o casos de uso, que permiten representar el trabajo que se quiere realizar.

Cap13. Se habla de aquellos aspectos funcionales que se desarrollarán en el proyecto. Se apoya en herramientas como el árbol de descomposición funcional.

Cap14. Se expone la planificación final que se utilizó durante todo el periodo de desarrollo. Además se agregan el cálculo asociado a puntos de casos de uso que permite determinar una aproximación al tiempo empleado para el desarrollo del proyecto.

Cap15. Se detalla los aspectos asociados a la capacitación realizada a las personas involucradas en este trabajo.

Cap17. Se muestra un resumen estimativo de las horas utilizadas para desarrollar las diversas tareas propuestas a lo largo del periodo de desarrollo de proyecto.

Cap18. Se detallan los resultados obtenidos en las iteraciones asociadas a la metodología de trabajo escogida para el desarrollo de este proyecto.

2 DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN

2.1 Descripción de la institución.

- **Antecedentes generales de la institución.**
  - **Nombre**: Universidad del Bío-Bío, Facultad de Ciencias Empresariales (FACE).
  - **Dirección**: Concepción, Avenida Collao 1202.
  - **Rubro**: Educación.
  - **Servicios que ofrece**: La Universidad el Bío-Bío ofrece "Programas de Pregrado y acreditación de carreras", "Programas de Postgrado", "Programas de formación continua" y finalmente "Servicios de Investigación, Desarrollo e Innovación (DGI)".
- **Entorno.**
  - **Competencia directa**: En Concepción la principal competencia existente erradica en otras casas de estudio como la Universidad de Concepción, Universidad Técnica Federico Santa María, Universidad Católica de la Santísima Concepción, entre otras.

- **Cuota de mercado.**

  Según la información entregada por el DEMRE, la Universidad del Bío-Bío, entre los periodos 2011 – 2013, tuvo la siguiente cantidad de postulaciones efectivas.

	Año 2011	**Año 2012**	**Año 2013**
UBB	16505	14123	14839
Total postulaciones efectivas (CRUNCH)	365586	509331	474809

Tabla 1 – Cuota de mercado UBB ([2])

- **Misión.**
  - La universidad del Bío-Bío es una institución de educación superior, pública, estatal y autónoma , de carácter regional, que se ha propuesto como misión:
    - Formar profesionales de excelencia capaces de dar respuesta a los desafíos de futuro, con un modelo educativo cuyo propósito es la formación integral del estudiante a partir de su realidad y sus potencialidades, promoviendo la movilidad social y la realización personal.
    - Fomentar la generación de conocimiento avanzado mediante la realización y la integración de actividades de formación de postgrado e investigación fundamental, aplicada y de desarrollo, vinculadas con el sector productivo, orientadas a áreas estratégicas regionales y nacionales.
    - Contribuir al desarrollo armónico y sustentable de la Región del Biobío, a través de la aplicación del conocimiento, formación continua y extensión, contribuyendo a la innovación, productividad y competitividad de organizaciones, ampliando el capital cultural de las personas, actuando de manera interactiva con el entorno y procurando la igualdad de oportunidades.
    - Desarrollar una gestión académica y administrativa moderna, eficiente, eficaz y oportuna, centrada en el estudiante, con estándares de calidad certificada que le permiten destacarse a nivel nacional y avanzar en la internacionalización.

- **Visión.**
  - Universidad del Bío-Bío: Ser reconocida a nivel nacional como una Universidad estatal, pública, regional, autónoma, compleja e innovadora con énfasis en la formación de capital humano, vinculada al desarrollo sustentable de la Región del Bío-Bío y que aporta a la sociedad del conocimiento y al desarrollo armónico del país.

- **Objetivos de la institución.**
  - La Universidad del Bío-Bío tiene como objetivo el contribuir, mediante el cultivo del saber, de la educación superior, de la asistencia técnica y de la capacitación, a la formación de profesionales y al desarrollo regional en el territorio en el cual realiza sus actividades, sin perjuicio de poder extender sus actividades, si las condiciones así lo requieren al ámbito nacional e internacional.
    ([3]).

- **Estructura organizativa Universidad del Bío-Bío.**

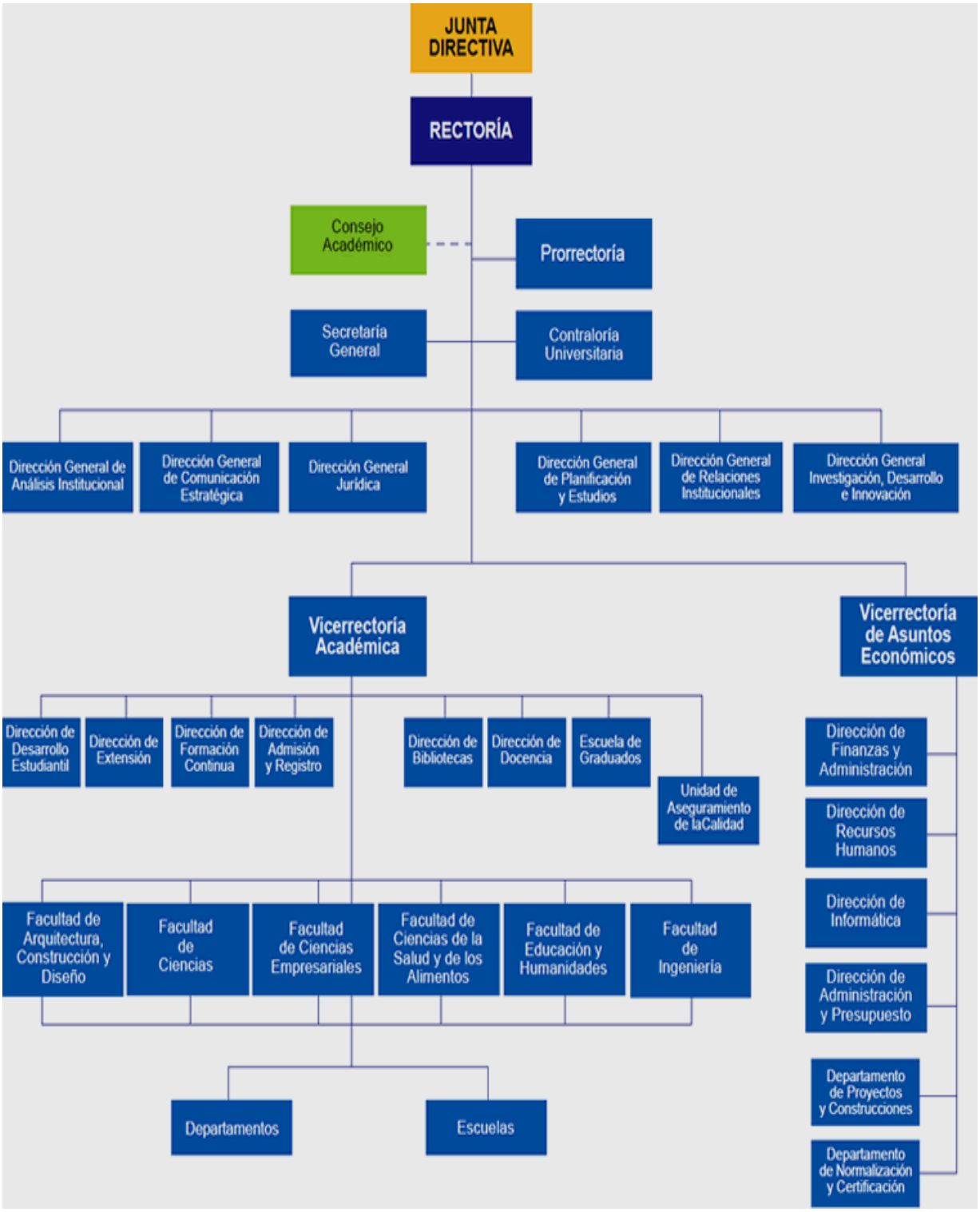

Figura 1 – Organigrama Universidad del Bío-Bío. [14]

- **Descripción de funciones más importantes.**
  - **Junta Directiva**.
    - Proponer al Presidente de la República la terna para la designación del Rector.
    - Fijar la política global de desarrollo de la Universidad y los planes de mediano y largo plazo para materializarla.
    - Aprobar el presupuesto anual, la planta de personal y las normas para fijación de las remuneraciones de la Universidad.
    - Nombrar a los funcionarios superiores o aprobar su renombramiento.
    - Aprobar la estructura orgánica de la Universidad.
    - Aprobar los préstamos con cargo a fondos de la Universidad.
    - Aprobar la creación, modificación o supresión de grados, diplomas y certificados, así como los títulos profesionales que correspondan.
    - Autorizar la compra, adquisición o enajenación de bienes raíces, la constitución, reserva y alzamiento de hipotecas y otros gravámenes que comprometan el patrimonio de la Universidad.
    - Aceptar donaciones y herencias destinadas a promover los fines de la corporación.
    - Nombrar a los profesores eméritos y honoris causa, miembros honorarios y otorgar otras distinciones.
    - Proponer al Presidente de la República la remoción del Rector por notable abandono de deberes.
    - Dirimir los conflictos de competencia que se susciten entre las diversas autoridades y organismos colegiados de la Universidad.
  - **Consejo Académico**.
    - Actuar como cuerpo consultivo del Rector en materias académicas.
    - Proponer al Rector las iniciativas que estime de utilidad para la buena marcha de la corporación.
    - Recomendar la creación de grados, títulos, diplomas y certificados, así como los planes de estudio conducentes a ellos.
    - Recomendar la creación, supresión o transformación de las Facultades y otras unidades u organismos académicos.
    - Proponer el calendario de actividades académicas.

- **Rector**.
  - Dirigir, promover y coordinar la actividad de la Universidad.
  - Representar a la Universidad, legal, judicial y extrajudicialmente.
  - Velar por el cumplimiento del presente Estatuto y reglamentación orgánica de la Universidad.
  - Proponer a la Junta Directiva los nombramientos de los funcionarios superiores de la Universidad contemplados en este Título a excepción de los señalados en sus artículos 18º y 19º. La remoción de tales funcionarios será facultad del Rector, salvo lo dispuesto en el artículo 25º del presente Estatuto.
  - Aprobar, en primera instancia, los cargos necesarios de académicos y funcionarios administrativos de la Universidad, solicitados por los Decanos de las Facultades y otros funcionarios con responsabilidad en la administración de la Universidad y, en virtud de ello, proponer a la Junta Directiva la planta de funcionarios de la Universidad y sus modificaciones.
  - Nombrar al personal académico y administrativo de la Universidad, conforme a los procedimientos establecidos en este Estatuto y los reglamentos respectivos.
  - Fijar el valor de la matrícula, aranceles y otros derechos que pueda percibir la Universidad de conformidad con lo dispuesto en el artículo 3º, letra e) de estos Estatutos.
  - Proponer a la Junta Directiva la política y normas de remuneraciones del cuerpo académico y de los funcionarios administrativos de la Universidad.
  - Proponer a la Directiva el presupuesto anual de la Universidad.
  - Aprobar el cupo anual de ingreso de estudiantes, previo informe del Consejo Académico.
  - Proponer a la Junta Directiva el nombramiento de profesores eméritos, profesores honoris causa, miembros honorarios y el otorgamiento de otras distinciones.
- **Prorrector.**
  - Subrogará al Rector en casos de ausencia o impedimento, con sus mismas facultades.
  - Tiene labores de coordinación en el desarrollo y gestión de los asuntos académicos y administrativos de la Universidad, sin perjuicio de otras que el Rector les encomiende (Art 17º).

- **Vicerrector Académico.**
  - Tiene la responsabilidad del desarrollo, administración de los asuntos académicos de la Universidad.
  - En caso de ausencia del Rector y Prorrector, él subrogará al Rector.
- **Vicerrector de Asuntos Económicos.**
  - Responsable de ejecutar la política administrativa y financiera de la Corporación, para un eficiente funcionamiento de sus actividades académicas.
- **Secretario General.**
  - Es el ministro de fe y vocero oficial de la Universidad.
  - Le corresponderá el cuidado, archivo y custodia de los documentos universitarios.
  - Le corresponderá la emisión de los documentos en que conste un título o grado otorgado por la Universidad.
  - Le corresponderá la suscripción de documentos y certificados oficiales emanados de la Universidad.
  - Le corresponderá el desempeño de todas las demás funciones inherentes a su cargo o que los reglamentos determinen.
  - Tendrá la custodia del sello de la Universidad, el que deberá ser estampado en los documentos que lo requieran.
- **Controlador Universitario.**
  - Tiene a su cargo el control de la legalidad de los actos de las autoridades de la Corporación.
  - Fiscalizará el ingreso y uso de los fondos.

Para ver los documentos originales de las distintas cargos y funciones de la universidad, puede visitar [4].

2.2 Descripción del área de estudio.

El área de estudio para este proyecto es la Facultad de Ciencias Empresariales, la cual pertenece al Departamento de Sistemas de Información.

- Énfasis en las funciones y cargos relacionados con el proyecto.
    - Profesor Patricio Galdames Sepúlveda: será el profesor guía de este proyecto. Su función es la entrega de la orientación necesaria para abarcar las funcionalidades necesarias para el desarrollo de los programas.
    - Alumno Heber Gálvez Ojeda, como alumno memorista, tiene la tarea de investigar, elaborar y desarrollar los programas planteados dentro de los objetivos del proyecto, con el objetivo de cumplir los requerimientos establecidos.
- **Objetivo del área.**

  El departamento de Sistemas de Información, tiene los siguientes objetivos:
    - El desarrollo académico de la Ciencias de la Computación e Informática y la participación activa en la formación, capacitación y asistencia técnica de profesionales en informática, particularmente en las áreas de la informática aplicada a la gestión.
    - Una creciente actividad de investigación relevante, buscando la formación de equipos en áreas donde se pueda destacar.
    - Un permanente perfeccionamiento de sus cuadros académicos y capacitación del personal administrativo, para el mejoramiento de la calidad.
    - Establecimiento de relaciones con el entorno, que permita una mayor pertinencia y contribución a su quehacer.

- **Estructura Organizativa y funciones.**
    - Mónica Caniupán Marileo, Directora Departamento Sistema de Información.
        - Karina Rojas Contreras, Jefa de Carrera Ingeniería Civil Informática.
        - Juan Carlos Parra Márquez, Jefe de carrera, Ingeniería Ejecución computación e informática.

2.3 Descripción de la problemática.

La problemática existente para este proyecto consiste en realizar la transmisión de videos de forma segura, empleando algoritmos criptográficos que garanticen la seguridad de los datos.
Para esto, se asume la existencia de un servidor, el cual almacenará y distribuirá los videos a los clientes. Aquí es donde surge el primer problema, ya que los usuarios podrían recibir la transmisión directamente desde el servidor, sin embargo, esto no es viable, pues los recursos que emplearía este servidor son limitados y no podría cumplir con las necesidades de todos los usuarios que quieran establecer la conexión.
Esto motiva que una primera solución sería trabajar bajo una red P2P. Para estimular la formación de esta red, el servidor podría considerar diversos incentivos, ya sean descuentos, acceso gratuito a videos, cuyo estudio y/o evaluación, no son parte de este trabajo.
La transmisión de video a través de este tipo de redes, presenta los siguientes desafíos:

- Dado que el streaming puede no provenir directamente desde el servidor principal, un usuario puede estar preocupado de saber si el video que recibió, es realmente auténtico, y no ha sido alterado en algún momento de la transmisión. En ese caso, el usuario podría presentar algún reclamo e idealmente incluir la posible fuente de alteración de los datos.
- Segundo, los usuarios que reenvían un video, desean notificar al servidor la cantidad de tráfico que han reenviado con el fin de recibir algún tipo de retribución. Por este motivo, el servidor debe poseer algún mecanismo que le permita estimar el tráfico que reenvía un usuario.
- Finalmente, solo aquellos usuarios que pagan su suscripción deben tener acceso a la visualización de los videos. En este sentido, puede existir un usuario que esté reenviando un video, pero que no ha pagado el costo asociado a la suscripción.

3 DEFINICIÓN PROYECTO

3.1 Objetivos del proyecto.

3.1.1 Objetivo General.

Evaluar el sobrecosto de tiempo que hacen uso los algoritmos criptográficos para garantizar la confiabilidad e integridad en una transmisión de videos a través de una red P2P.

3.1.2 Objetivos Específicos.

- Evaluar el retardo agregado por el protocolo de transmisión de datos seleccionado (UDP y TCP).
- Evaluar el retardo agregado por diversas herramientas criptográficas que han sido propuestas para garantizar confidencialidad e integridad de datos.
- Desarrollar una aplicación red a nivel de cliente, que permita verificar la autenticidad de un video retransmitido por un miembro de la red P2P.
- Desarrollar una aplicación a nivel de servidor que permita realizar una transmisión de videos hacia los clientes y que brinde confiabilidad e integridad a esa transmisión.

3.2 Ambiente de Ingeniería de Software.

- **Metodología de desarrollo**: la metodología de trabajo que se utilizó fue una metodología con entrega evolutiva, ya que a medida que transcurría el tiempo y el progreso de las aplicaciones, se iban agregando nuevos elementos que permitieran obtener el resultado planteado en un principio.
- **Técnicas y notaciones**: Se utilizó notaciones técnicas de manera tal que la explicación de los elementos fuera clara y entendible por quienes lean el documento.
- **Estándares de documentación, producto o proceso.**
    - El estándar que se planteó para el desarrollo de este documento, corresponde al establecido por la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío. Además, se plantea para la especificación de la bibliografía el estándar establecido por la biblioteca de la misma casa de estudios.

- **Herramientas de apoyo al desarrollo de software que serán utilizadas.**
    - **Netbeans/Eclipse**: Entorno de desarrollo que nos permite elaborar los códigos necesarios para la creación de las aplicaciones.
    - **Java**: Lenguaje de programación empleado por sus diversas ventajas frente a otro lenguaje como C.
    - **VLC Player**: reproductor multimedia, gratuito y multiplataforma, el que puede actuar como servidor y cliente para poder realizar las pruebas de transmisión de videos.
    - **SublimeText**: Editor de texto, orientado principalmente para el desarrollo y escritura de códigos de programación.

3.3 Definiciones, Siglas y Abreviaciones.

- **P2P**: Una red P2P, o Peer-to-peer, es una red la cual conecta una gran cantidad de ordenadores o nodos, con el objetivo de compartir diverso material digital, como videos, imágenes, música, entre otros.
- **Streaming**: Esta es una tecnología que permite aligerar la descarga de video y audio, ya que permite reproducirlos mientras estos aún están siendo descargados.
- **Servidor**: El servidor es un tipo de software/equipo el cual pertenece a una red, y que tiene como propósito entregar servicios a otros computadores, que en este contexto, se denominan clientes.
- **Clientes**: Los clientes son computadores o nodos, los cuales pertenecen a una red. Estos clientes reciben un servicio por parte de un servidor.
- **Red Cliente/Servidor**: Esta es una red de comunicaciones, compuesta por diferentes nodos (computadores), en donde aquellos nodos clientes están conectados a uno o más servidores. Estos servidores poseen los recursos y aplicaciones necesarias para responder de forma satisfactoria a las peticiones que realizan los clientes.
- **Socket**: Los sockets son una metodología la cual permite la comunicación entre un programa cliente, y un programa servidor en una red.
- **Datagrama**: Los datagramas son una unidad de transferencia básica que está asociada a una red de conmutación de paquetes en los que la entrega, la hora y el orden de llegada no está garantizada.

- **UDP**: Del inglés, User Datagram Protocol (Protocolo de Datagramas de usuario), es un protocolo no orientado a la conexión, esto quiere decir, que permite el envío de datagramas por la red, sin que exista previamente una conexión establecida.
- **TCP**: Del inglés Transfer Control Protocol (Protocolo de Control de Transmisión), es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. A diferencia de UDP, este protocolo es orientado a la conexión, es decir, permite que dos máquinas comunicadas puedan controlar el estado de la transmisión.
- **Multicasting:** Corresponde a la comunicación entre un emisor, y múltiples receptores dentro de una red.
- **AES**: Advanced Encryption Data.
- **DES**: Data Encryption Standard.
- **Triple DES:** Triple cifrado de DES.
- **CBC:** Cipher-block Chaining.
- **ECB**: Electronic Code-Book.
- **CTR:** Counter Mode.
- **IV:** Initialization vector o Vector de Inicialización,

4 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

Basado en [5].

4.1 Alcances.

Las aplicaciones desarrolladas durante este proyecto son tres.

- La primera aplicación, denominada “Servidor Java”, tiene como objetivo recibir los datos que son transmitidos por el servidor de video, o en este caso, el programa VLC Servidor. Esta aplicación java recibe el streaming, bajo un protocolo de transmisión como UDP o TCP, y un determinado puerto. A medida que los datos son recibidos por este programa, estos van siendo cifrados bajo algún método criptográfico que permita la confidencialidad de los datos. Finalmente los datos encriptados son enviados a una segunda aplicación Java.
- La segunda aplicación, denominada “intermediario java”, recibe los datos enviados por la primera aplicación, estableciendo el mismo protocolo de

transmisión y configurando el puerto según lo establezca el Servidor Java. Una vez que recibido los datos, los vuelve a re-enviar a la tercera aplicación java, sin realizar algún tipo de cambio.

- Finalmente, la tercera aplicación java, llamada "Cliente Java", obtiene los datos encriptados, que fueron re-enviados por la aplicación intermediaria. Estos datos son descencriptados, teniendo en consideración el uso del mismo método criptográfico empleado en un principio. Una vez que los datos son descencriptados, se obtiene el streaming originalmente enviado por el VLC Servidor, y por tanto ahora se transmiten al VLC cliente, quien espera por estos datos, bajo el protocolo de transferencia y puertos establecidos previamente.

Esta aplicación no considera el uso de archivos de fuentes externas como por ejemplo el uso de videos almacenados en los servidores de Youtube.

El proyecto tampoco está enfocado a realizar transmisiones en vivo, por ejemplo como lo hace Skype.

El objetivo es netamente de investigación, buscando métricas de comparación entre los algoritmos que se empleen. Por lo tanto, este proyecto no se ha desarrollado con fines lucrativos, si no que más bien didácticos.

4.2 Objetivo del software.

4.2.1 Objetivo General Software.

El objetivo general de las aplicaciones es tener un medio por el cual los datos que se transmiten desde el equipo servidor hacia el equipo cliente, puedan ser encriptados, garantizando la confidencialidad de los datos. Se debe tener en consideración, que las 3 aplicaciones desarrolladas deben ser "transparentes" para el usuario, esto quiere decir, que el usuario que reciba la transmisión, no se percate de la ejecución de las aplicaciones Java.

4.2.2 Objetivo Específico del Software.

De forma específica, el desarrollo de estas aplicaciones está enfocado en la medición de dos parámetros:

- El tiempo de procesamiento existente entre los equipos empleados.
- El tiempo de transferencia entre el "Servidor Java", el "intermediario java", y luego entre el "intermediario java" y el "cliente java".

Con los resultados que se obtengan se determinará:

- Tiempos de descarga.
- Tiempos entre segmentos, lo que equivale a lectura de datos del socket.
- Tasa de recepción de segmentos.

Para poder realizar el cálculo de estos tiempos, se considerará el uso de los siguientes algoritmos criptográficos, DES, Triple-Des y AES. Además se empleará el uso de los siguientes métodos de cifrado, CBC, CTR y ECB. Finalmente para asegurar una correcta transmisión de los datos se empleará RSA.

4.3 Descripción Global del Producto.

4.3.1 Interfaz de usuario.

La interfaz desarrollada tiene como requerimientos el poder establecer los parámetros asociados a:

- Algoritmo criptográfico: DES, Triple DES y AES.
- Protocolo de Comunicación: UDP o TCP.
- Tipo de Cifrado: ECB, CBC, CTR.

Esta interfaz se debe aplicar para los programas Cliente y Servidor Java, donde además se asume que el Servidor envía por algún medio la combinación necesaria de los parámetros con el objetivo de recibir los datos sin problemas.

4.3.2 Interfaz De Hardware.

- Características de los equipos empleados.
  - Sistema Operativo: Microsoft Windows XP o Ubuntu 12.04.
  - Procesador: Intel Core 2 Duo, 2.93 GHz.
  - Memoria RAM: 1,96 GB.
  - Firewall: Desactivado.

* Puertos empleados: Para el desarrollo de las pruebas, se emplearon los puertos habilitados tales como: 8080, 8081, 5678, 6789, 1234, 4321.
* Conexión entre equipos: la configuración de los equipos empleados es la configuración existente en los laboratorios de la FACE.

4.3.3 Interfaz Software.

- Nombre: VLC Media Player.
- Abreviación: VLC
- Número especificación o Versión: Reproductor multimedia VLC 2.0.8 Twoflower.
- Fuente: http://www.videolan.org

4.3.4 Interfaces de comunicación.

Los protocolos de comunicación con los cuales se trabajó en este proyecto corresponden a los siguientes:

- UDP.
- TCP.
- TCP/IP.

4.4 Requerimientos Específicos.

4.4.1 Requerimientos Funcionales del sistema.

Id	Nombre	Descripción
Req - 01	**Ingreso protocolo de transmisión.**	**El usuario debe seleccionar en los programas Java Cliente y/o Servidor, el tipo de protocolo de transmisión, UDP o TCP. Este protocolo debe ser el mismo en ambos programas, y además debe coincidir con los protocolos establecidos previamente en el programa VLC Cliente y Servidor.**
Req - 02	**Ingreso del tipo de cifrado.**	**El usuario debe seleccionar en los programas Java Cliente y/o Servidor, el tipo de cifrado que se empleara en la encriptación y desencriptación. Estos modos son: ECB, CBC, CTR. El modo seleccionado, debe ser el mismo para ambos programas Java.**
Req - 03	**Ingreso de algoritmo criptográfico**	**El cliente debe seleccionar uno de los tres algoritmos criptográficos (DES, 3DES, AES). El tipo de algoritmo seleccionado, debe ser el mismo en los programas java Cliente y java Servidor.**

Tabla 2 – Requerimientos funcionales del sistema.

4.4.2 Interfaces externas de entrada.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
DAT – 01	**Protocolo transmisión.**	**UDP o TCP.**
DAT – 02	**Algoritmo criptográfico.**	**DES, Triple DES, AES.**
DAT – 03	**Modo Cifrado.**	**ECB, CBC, CTR.**
DAT – 04	**Llave del algoritmo seleccionado.**	**Cuando se selecciona el algoritmo, se requiere un archivo que posee la llave de cifrado. El tamaño de esta llave varía según el algoritmo seleccionado.**
DAT – 05	**Vector de inicialización.**	**Se requiere de este vector, para el cifrado y descifrado cuando se dan ciertas combinaciones entre los algoritmos criptográficos y los tipos de cifrado.**

Tabla 3 – Interfaces externas de entrada.

4.4.3 Interfaces externas de Salida.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
S-01	Datos de streaming originales.	Corresponde a los datos que son transmitidos desde el programa VLC Servidor. Estos datos son transmitidos en formato binario.	Red de equipo comunicados
S-02	Datos de streaming encriptados.	Corresponde a los datos de streaming que han sido encriptados bajo algún algoritmo criptográfico. Estos datos son los transmitidos por el programa Java Servidor, y el programa Java intermediario.	Red de equipos comunicados.
S-03	Datos de Streaming descencriptados.	Corresponde a los datos de streaming que han llegado al programa Java cliente, y que se han descencriptados por alguno de los algoritmos criptográficos. Estos datos, son los transmitidos por el programa Java Cliente, hacia el programa VLC cliente.	Red de equipos comunicados Pantalla, visualización del video original.

Tabla 4 – Interfaces externas de salida

4.4.4 Atributos del producto

- USABILIDAD- OPERABILIDAD. La interfaz gráfica debe ser amigable, es decir, que el usuario solo debe seleccionar los parámetros necesarios para la transmisión y recepción de los datos, sin tener que preocuparse de otros factores, como por ejemplo el tamaño de la llave necesaria para aplicar la criptografía.

- EFICIENCIA, el tiempo de retardo que se pueda producir en la transferencia de los datos, no debe superar los 10 segundos, considerando que las pruebas y resultados obtenidos se realizaron en una red estable.

- PORTABILIDAD. Al ser los programas desarrollados con Java como lenguaje de programación, no debe existir problemas en ejecutarlo en sistemas operativos basados en Linux, siempre y cuando tengan los elementos de software necesarios para el correcto funcionamiento.

5 INTRODUCCIÓN A LA CRIPTOGRAFÍA

5.1 Reseña Histórica.

Primero que todo es de saber que la palabra "criptografía" proviene del griego "**krypto"** (oculto) y "**graphos"** (que equivale a escribir). De lo anterior se puede deducir un significado más preciso: "escritura oculta", o "escribir mensajes en clave secreta".

La criptografía, en un principio escrita, nace principalmente como una necesidad requerida por el hombre para poder transmitir información segura hacia terceras personas, evitando la intercepción de los mensajes por agentes externos. Lo que se busca es garantizar la privacidad de la información transmitida, *cifrando* el mensaje original con algún algoritmo criptográfico.

Es importante destacar que la criptografía solo corresponde a una parte de la comunicación secreta. Si se quiere enviar un mensaje cifrado con algún método criptográfico, es porque existe inseguridad/desconfianza en la transmisión, debido a factores como que el mensaje podría ser interceptado por terceras personas y utilizado para fines no deseados. Para que estos individuos puedan *descifrar* el mensaje transmitido, emplearán diversas técnicas y métodos los cuales en conjunto, forman una ciencia llamada "criptoanálisis".

Al conjunto de criptografía y criptoanálisis se le conoce como **Criptología.**

En toda comunicación secreta se presenta una lucha entre criptógrafos y criptoanalistas. El éxito de unos representa el fracaso de los otros. ([1]).

Ahora bien, los primeros registros del uso de sistemas criptográficos datan desde el año 1500 a.C, en donde los comerciantes asirios hacían uso de tablillas de arcilla en donde tallaban escritos y ciertas imágenes, las cuales establecían la forma de llevar a cabo sus transacciones comerciales, muchas veces, dichas tablillas eran colocadas dentro de un contenedor del mismo material para luego ser transportado.

Sin embargo, varias investigaciones establecen un patrón similar en el uso y evolución de la criptografía a través de las diferentes civilizaciones antiguas. A continuación, se detallan las más relevantes y conocidas.

- **El cifrador de Polybios**.

  Este método de cifrado surge a mediados del siglo II a.C, y su nombre se debe a un historiador griego. El procedimiento de cifrado, consistía en ir cambiando los caracteres del mensaje original, por un par de nuevos caracteres correspondientes según una tabla diseñada para ese propósito.

  Sin embargo, se consideró que como el tamaño del texto cifrado se duplicaba, con textos o números, este método no era tan bueno.

  A continuación, un ejemplo:

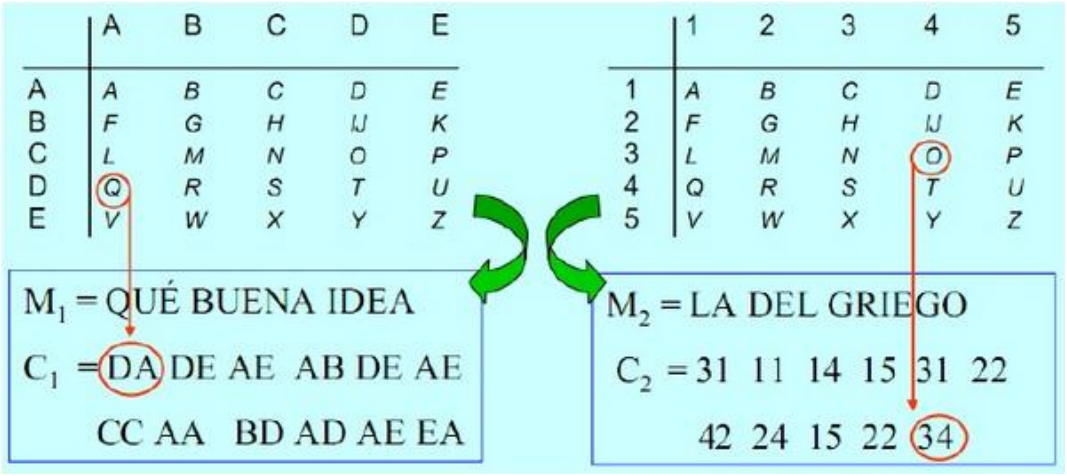

Figura 2 - Ejemplo Polybios. [15]

- **El cifrador de Cesar.**

  Esta técnica de cifrado debe ser una de las más conocidas. Su nombre proviene debido a que el sistema fue usado por el militar y político Julio Cesar, en el siglo I a.C.

  La técnica consiste en sustituir cada carácter del mensaje original, por otro carácter situado tres posiciones más adelante, en un determinado alfabeto.

  El método puede mejorarse cambiando el segundo diccionario (el desplazable), por uno que fuese aleatorio.

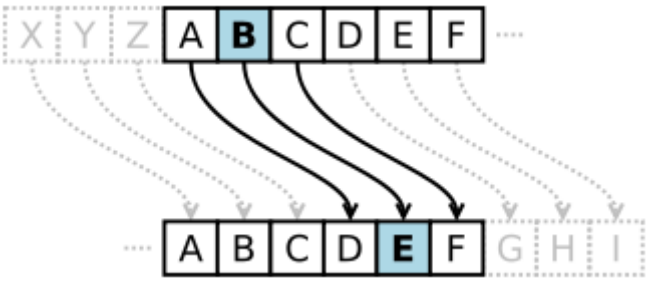

Figura 3 - Cifrado de Cesar.[16]

- **Las guerras mundiales**.

  Las guerras mundiales influyeron en el desarrollo de nuevos métodos que permitieran brindar seguridad a comunicaciones militares y diplomáticas para lo cual se emplearon técnicas como la telegrafía y la radiotecnia.

  En la primera guerra mundial, la ruptura del telegrama Zimmemann, en donde el ministro alemán buscaba convencer a México y Japón para invadir EE.UU, y que como consecuencia, hizo que EE.UU se decidiera a entrar en la guerra.

  Por otra parte, en la segunda guerra mundial, la oficina cripto-analítica británica liderada por Alan Turing, lograba romper el cifrado de la máquina alemana Enigma, mediante el uso de la maquina Colossus.

5.2 Algoritmos criptográficos de clave simétrica o privada

Este tipo de algoritmos se caracteriza por emplear una sola llave de seguridad, la cual es utilizada para cifrar y descifrar el dato o mensaje que se quiere transmitir. El conocimiento de la llave, debe ser compartido solo por el emisor y el receptor o receptores involucrados en la transmisión de la información, y por lo tanto es de gran importancia mantener el secreto de la llave. Una forma de representar el funcionamiento de este tipo de algoritmos, es el siguiente:

- MENSAJE + LLAVE = CODIGO (CIFRADO O ENCRIPTACIÓN).
- CODIGO + LLAVE = MENSAJE (DESCIFRADO O DESCENCRIPTACIÓN).

Algunas características necesarias para el correcto funcionamiento de este tipo de algoritmo son:

- Cuando el dato sea cifrado, no debe ser posible que entes externos a la comunicación, obtengan la llave, y por ende, no puedan recuperar el dato original.
- La complejidad para recuperar la llave de seguridad debe ser muy alta, aun cuando se tengan los datos cifrados y los datos descifrados.
- Como el funcionamiento de este tipo de algoritmos es público, la privacidad de la llave se hace realmente importante.
- Del punto anterior es que debe haber un compromiso o confianza entre el emisor y el o los receptores, de mantener la privacidad de la llave.
- La cantidad de llaves utilizadas, será proporcional, a la cantidad de datos que se quieran proteger.

Dentro de los algoritmos que están en esta categoría, destacan DES, Triple-DES, AES, IDEA, RC5. Para este proyecto, se utilizaron los tres primeros algoritmos, que a continuación se explican de forma general.

5.2.1 DES (Data Encryption Standard)

DES o estándar de cifrado de datos en español, es un algoritmo que fue desarrollado en un inicio por la compañía IBM por requerimiento del Instituto Nacional de Estandarización y Tecnología (NIST – National Institute of Standarts and Technology) de Estados Unidos, el cual en el año 1977 fue modificado y adoptado por el gobierno de ese mismo país como un estándar de cifrado de todas las informaciones que fuesen sensibles no clasificadas. Luego en el año 1980, el NIST estandarizó los diferentes modos de operación del algoritmo.

La idea general en la cual se basa DES, es aplicar diversas funciones a los datos que se quieren cifrar, con el objetivo de que sólo por medio de la llave y aplicando la forma inversa de las funciones, se puedan descifrar los datos.

Este algoritmo cumple con dos principios básicos de la criptografía, los cuales son:

- El secreto, el cual se consigue con el uso de la llave privada.
- La autenticidad, que también se consigue de la misma forma, pues sólo el verdadero emisor es quien crea el mensaje o los datos que enviará para que el receptor pueda descifrar con la llave establecida y compartida.

El algoritmo usa una llave de 64 bits, pero solo los 56 primeros bits son utilizados para el cifrado, es decir, la llave real es de ese tamaño. Mientras que los bits restantes son utilizados para comprobar errores durante el proceso.

Ventajas	Desventajas
Es uno de los algoritmos más probados, ya que es ampliamente usado y extendido.	La longitud de la clave no puede ser aleatoria. Por ende, no se puede aumentar el tamaño de la llave para dar más seguridad.

Su implementación es sencilla y rápida.	La longitud de 56 bits actualmente es demasiado corta, y por ende más vulnerable.
	Ya no es considerado como estándar debido a que en 1999 la seguridad fue quebrada por un ordenador.

Tabla 5 – Ventajas/Desventajas DES.

Existen diversos trabajos y libros [6], en los cuales se habla de manera mucho más detallada sobre los aspectos de DES, como por ejemplo la explicación de los diferentes métodos que permiten el cifrado de la información.

5.2.2 TRIPLE DES (Triple Data Encryption Standard).

La velocidad en que la tecnología evoluciona, implica y requiere que los sistemas de seguridad se adapten a esas necesidades. Es por ese motivo que DES ya no es mayormente utilizado, pues los tiempos empleados para romper su seguridad no son excesivamente altos.

A partir de lo anterior surge como solución Triple DES (3DES), que como su nombre indica, es el uso de DES pero tres veces, en donde la llave para el cifrado tiene una longitud de 128 bits, la cual se divide en dos llaves “A” y “B”.

Cuando los datos son recibidos, se hace uso de DES con la llave “A”, luego se repite usando la llave “B” para que finalmente se vuelva a utilizar la llave “A”. Si bien es cierto, hay un aumento en la seguridad, también existe un aumento en el uso de los recursos empleados para llevar a cabo el algoritmo.

Actualmente hay una variación de este algoritmo, el cual se llama DES-EDE3, usando tres claves diferentes y que usan una llave de 192 bits, lo que genera un sistema más fuerte.

5.2.3 AES (Triple Data Encryption Standard).

AES surge como el algoritmo sustituto de DES, pues como se mencionó anteriormente, en la actualidad DES ya no brinda suficiente seguridad para el cifrado de datos.

Un uso muy conocido de este algoritmo, es el que se da en los routers, ya que aquellas claves del tipo WPA, operan con el algoritmo AES como algoritmo de cifrado.

Este sistema criptográfico, opera con bloques y llaves de cifrado/descifrado que pueden ser variables, donde los tamaños de la llave pueden ser de 128 bits, 192 bits o 256 bits. En este proyecto y siguiendo los parámetros establecidos por el documento Java Security, Part 1: Cripto Basics de IBM, se ha usado un tamaño de llave de longitud de 128 bits.

Para tener una idea del nivel de seguridad que genera este algoritmo, es que si una computadora pudiera romper la seguridad de DES en un segundo, para romper la seguridad de AES, tardaría unos 149 billones de años.

Existen también diversos documentos donde se explica de forma más detallada y especifica el funcionamiento de este algoritmo. [7], [8].

En general, y de forma más gráfica el funcionamiento de los algoritmos simétricos es como se muestra a continuación.

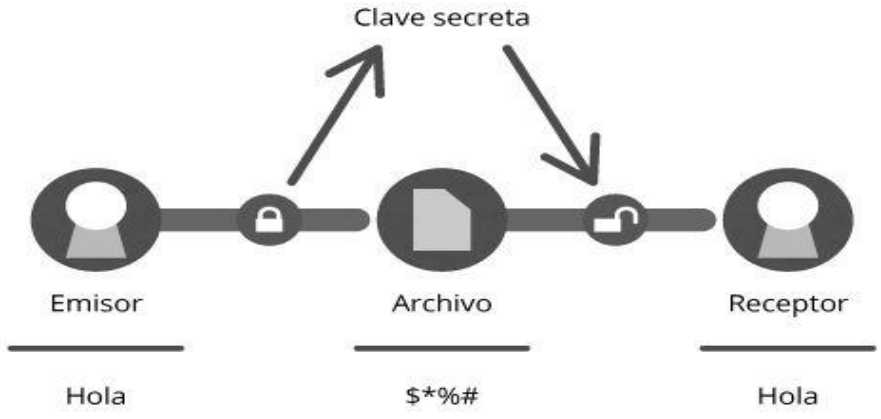

Figura 4 – Esquema funcionamiento algoritmo de llave privada.[17]

5.3 Técnicas para el cifrado de bloques

Los algoritmos anteriormente planteados, DES, Triple DES y AES, se caracterizan por ser cifrados en bloque, esto quiere decir que el flujo de datos original transmitido se separa en segmentos (bloques) de tamaño fijo, los que luego son cifrados. Las formas en que son gestionadas las piezas de esta separación, son denominadas cifrado por bloques. Cuando el cifrado es realizado, la unidad de cifrado por bloques toma como entrada el dato original, y produce un bloque de cifrado del mismo tamaño. La transformación exacta se controla por medio de otro parámetro, la llave secreta utilizada por el emisor y receptor.

Existen diversas técnicas, como por ejemplo: CBC, ECB, CTR, OFB, CFB, entre otras. A continuación se establece una breve reseña de las técnicas empleadas en este proyecto.

5.3.1 ECB (Electronic Codec Book mode).

Este método ha sido estandarizado por NIST (National Institute Standard and Technology), y es el más simple de todos, ya que divide el mensaje en bloques y luego los cifra por separado.

Una de las ventajas es la posibilidad de interferir el mensaje en bloques para poder cifrarlos en paralelo. Sin embargo esta técnica ya no es demasiado utilizada debido a sus grandes falencias. El cifrar cada bloque por separado trae como consecuencia que cada vez que se cifre un bloque con un determinado valor, siempre se obtendrá el mismo resultado. El error que presenta este tipo de cifrado, se puede apreciar a continuación.

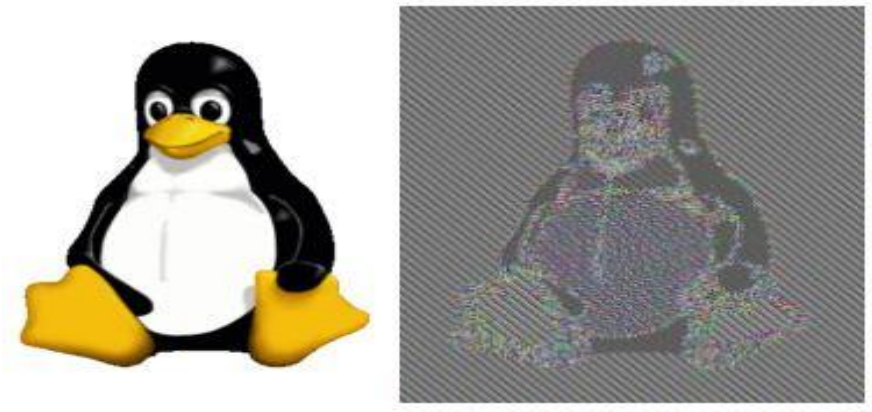

Figura 5– Ejemplo Cifrado de imagen con ECB. [18]

Como se puede apreciar en la imagen, y según lo explicado en el párrafo anterior, cifrar aquellos bloques que poseen el mismo valor, en este caso los matices de la imagen, generarán siempre la misma salida. Por este motivo es que la imagen cifrada (imagen derecha), es prácticamente similar a la original y entendible, por lo que la seguridad que se ha aplicado no es suficiente, pues aun cuando está cifrada, la imagen no varía en demasía a la original.

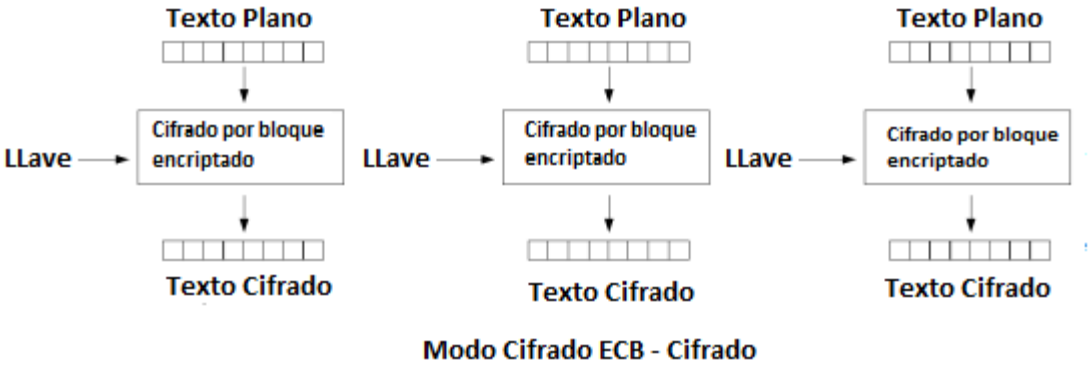

Figura 6 – Esquema cifrado ECB. [18]

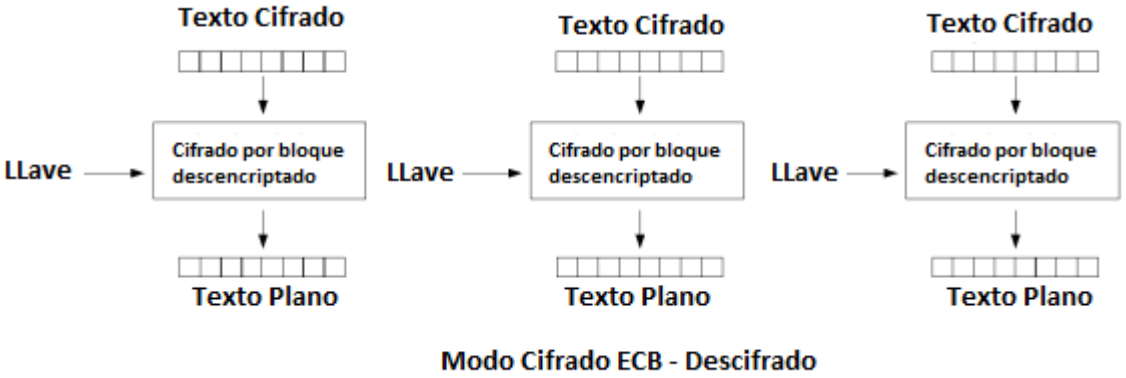

Figura 7 – Esquema descifrado ECB. [18]

5.3.2 CBC (Cipher Block Chainging mode)

Este método también ha sido estandarizado por NIST. Es una extensión del método ECB, donde se añade cierto grado de seguridad. CBC lo que hace, es dividir el mensaje en bloques y aplicar el operador lógico XOR (OR Exclusivo) con el objetivo de combinar el cifrado del bloque anterior, con el texto plano del bloque actual, es decir, cada bloque de texto cifrado dependerá de todo el texto plano que se ha procesado hasta ese punto.

Para el primer bloque, no existe un texto cifrado con el cual se pueda realizar la combinación. Para solucionar eso se utiliza un vector de inicialización (IV), el cual es un número aleatorio, no secuencial, que puede ser públicamente conocido.

Es muy importante el uso de este vector, ya que de no utilizarlo, los datos cifrados podrían ser recuperados por terceros, haciendo uso de un ataque del tipo diccionario.

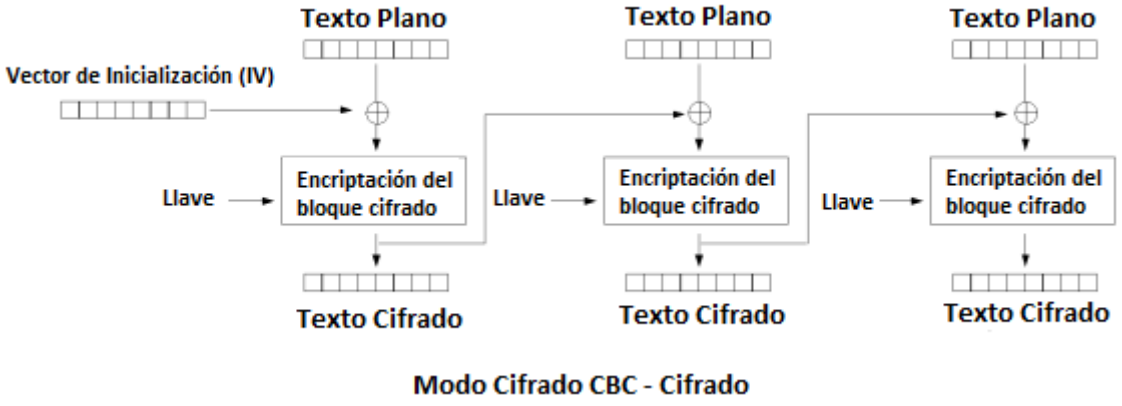

Figura 8 – Esquema cifrado CBC. [18]

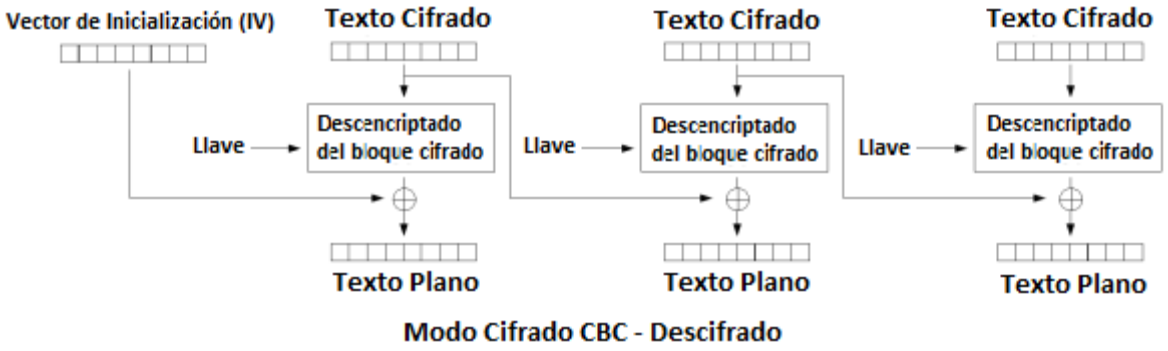

Figura 9 – Esquema Descifrado. [18]

5.3.3 CTR (Counter Mode)

A diferencia de ECB y CBC, cuyo sistema de trabajo es cifrado por bloques, este método simula un sistema de cifrado de flujos. Se hace uso de un cifrado de bloques para crear un flujo, el cual es pseudo aleatorio denominado KeyStream. Posteriormente el flujo generado se combina con el operador lógico XOR (OR exclusivo), lo que da origen al cifrado. Para la obtención del KeyStream, se necesita cifrar un contador, el que se combina con un número aleatorio, usando ECB y se va incrementando. Según la información recopilada, se recomienda que el valor que posea el contador se mantenga en secreto, aunque podría ser de conocimiento público. Además, tanto el emisor como el receptor de la información deben conocer los valores del número aleatorio empleado, y valor del contador.

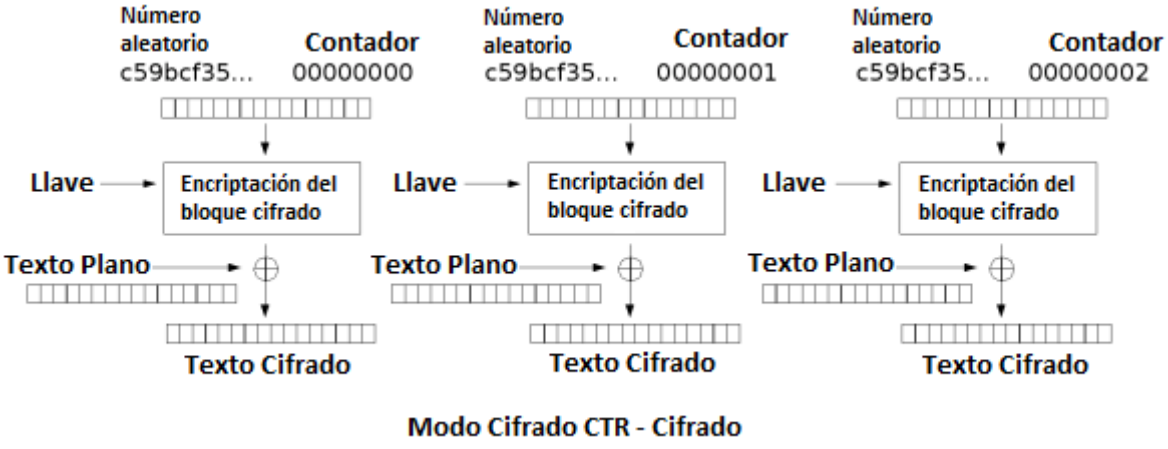

Figura 10 – Esquema Cifrado CTR. [18]

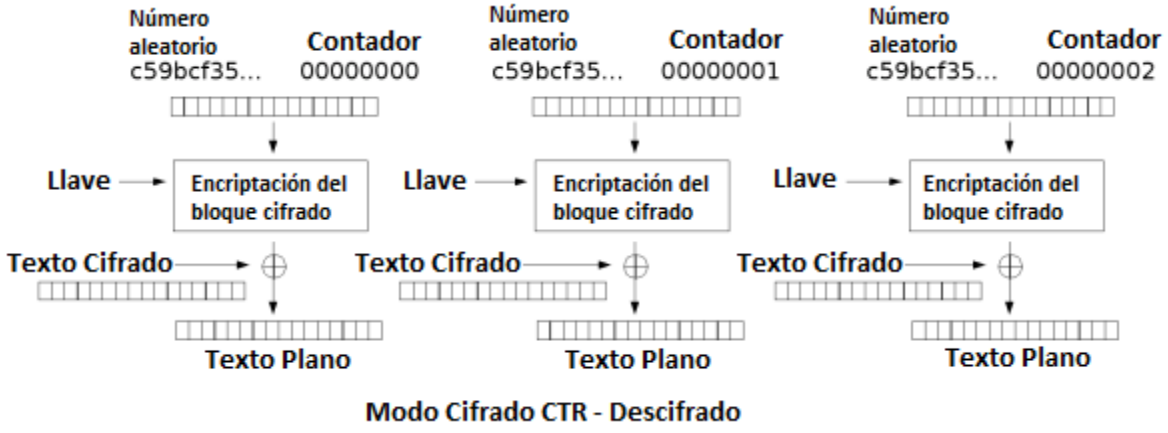

Figura 11 – Esquema Descifrado CTR. [18]

5.3.4 Padding.

El padding es una técnica de relleno de datos, empleada principalmente en los métodos de cifrado en bloques, tales como CBC o ECB.

Recordando que la separación de los datos en bloques, tienen un tamaño fijo, el padding se aplica cuando el largo del mensaje a encriptar, no es múltiplo del tamaño del bloque, por lo que es necesario "rellenar" ese bloque, para obtener el tamaño fijo establecido.

Al igual que los métodos de cifrado, existen diversos métodos para el uso de padding, tales como:

- No Padding.
- PKCS5 Padding.
- OAEP.
- SSL.

Para el cifrado de datos en este proyecto, se utilizó la segunda opción, recomendada en el documento Java Security. Mientras que para el descifrado de los datos, se estableció un No padding.

Se puede encontrar más información y ejemplos de uso en [9].

5.4 Algoritmos criptográficos de clave asimétrica o pública.

En este tipo de algoritmos, se emplean dos tipos de llaves para asegurar el mensaje o los datos, una llave la cual es pública, y la otra privada.

La llave pública que posee el usuario, puede ser distribuida a otras personas con quien desee compartir la información, mientras que por otra parte, la llave privada, es la que debe mantener en secreto.

Los métodos de criptografía, garantizan, que este par de llaves sólo se pueda generar una sola vez, evitando posibles fallas o copias, las que afectarían en la transmisión del mensaje, o que podrían involucrar a terceras personas, no asociadas a la comunicación.

El procedimiento que se utiliza para cifrar y descifrar la información que se quiere asegurar, consiste en:

- El emisor, cifra el dato que quiera transmitir al receptor, usando la clave pública del receptor.
- Se transmite el dato cifrado, por el canal de comunicación empleado.

- El receptor descifra el dato, empleando su llave privada. De esta forma es posible garantizar la confidencialidad del mensaje, ya que solamente el receptor podrá recuperar la información original.

Al igual que en los algoritmos de clave privada, el tamaño de las llaves es un factor importante que se debe considerar para poder garantizar un correcto cifrado de la información, ya que la seguridad, recae en la complejidad e imposibilidad de poder obtener la clave privada a partir de la clave pública. La variación en el tamaño de la llave de este tipo de algoritmos, varía desde los 512 bits hasta los 4096 bits. Las claves con un tamaño (mínimo) de 1024 bits, son consideradas seguras en la actualidad.

Algunos algoritmos asociados a las llaves públicas son: RSA, DSA, Diffie-Hellman, ElGamal. Para este proyecto, se ha utilizado el algoritmo RSA, donde a continuación se entrega una breve reseña.

5.4.1 RSA

El sistema RSA, fue desarrollado en el año 1978 en el MIT (Instituto Tecnológico de Massachusetts), por los señores Ronald Rivest, Leonard Adleman y Adi Shamir, y su nombre se debe a las iniciales de los apellidos de cada uno de sus creadores.

El sistema, permite el cifrado y el firmado digital. Los datos que son cifrados y enviados utilizando este algoritmo, son representados mediante números. El funcionamiento toma el producto de dos números primos seleccionados al azar y mantenidos en secreto, los que deben ser mayores que 10^{100}. Con el resultado obtenido, es que se genera la llave de cifrado y descifrado.

La base de la seguridad de RSA radica en el problema matemático de la factorización de números demasiado grandes, pues no existen maneras rápidas de poder obtener resultados favorables, haciendo uso de una computadora tradicional.

Actualmente de los algoritmos que usan clave pública, RSA es el más empleado, además se prevé que el tamaño de los números primos seleccionados para la generación de las llaves, aumente debido al incremento existente en la capacidad de cálculo y proceso de los computadores.

Al igual que con los métodos anteriores, existen diversos documentos y libros que explican de forma más clara y especifica el funcionamiento del algoritmo RSA, cuyo contenido no es explicado en este documento, pues la información es bastante extensa.

En general, el funcionamiento de este tipo de algoritmos, es como se gráfica en la siguiente imagen

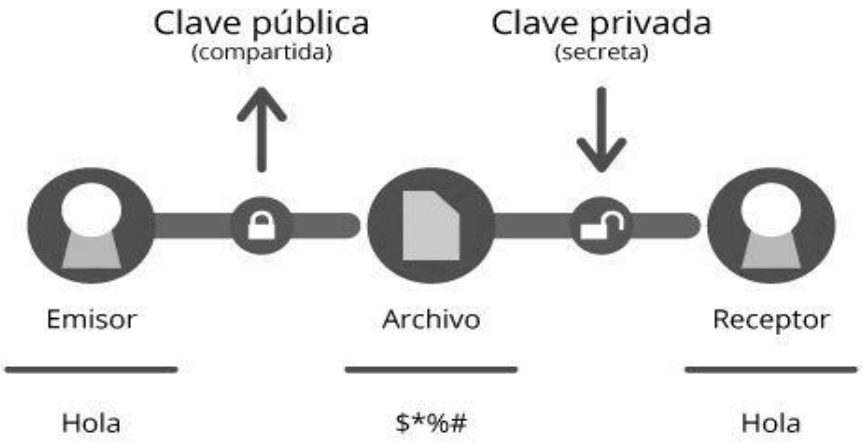

Figura 12 – Funcionamiento algoritmo de llave pública. [17]

5.5 Tabla comparativa algoritmos simétricos v/s algoritmos asimétricos.

Características	Algoritmo Llave Simétrica	Algoritmo Llave Asimétrica
Principal uso	Cifrado para cantidades grandes de datos.	Intercambio de las llaves, aplicación de firma digital.
Principales algoritmos	DES, Triple DES, AES.	RSA, Diffie-Hellman, DSA.
Velocidad de cifrado	Rápida.	Lenta**.**
Seguridad Llaves	Deben ser compartidas entre el emisor y receptor, manteniéndolas en secreto.	Se tiene la llave privada, conocida solo por una persona. Y la llave publica conocida por todos.
Distribución de llaves	El intercambio por un canal inseguro es complicado.	La llave privada jamás debe ser compartida, mientras que la llave pública puede ser compartida por cualquier medio.
Tamaño de	DES (56 bits).	

llaves	3DES (112 bits). AES (128 bits).	RSA (1024 hasta 2048 bits)
Seguridad que brindan.	Estos algoritmos entregan, confidencialidad, integridad y autenticación.	También otorgan confidencialidad, integridad y autentificación, no repudio.

Tabla 6 – Comparativa cifrado público y cifrado privado

6 DESARROLLO DE PROYECTO

6.1 Introducción desarrollo

La base fundamental de este proyecto, consiste en la comunicación de datos empleando el paradigma Cliente/Servidor. El canal de comunicación utilizado corresponde a la red de Internet, o en su defecto, el laboratorio de redes que dispone la universidad.

Los protocolos de comunicación empleados son UDP y TCP. Cada uno de estos posee características similares las cuales fueron estudiadas y comprendidas, de tal manera de desarrollar una correcta comunicación entre los equipos utilizados.

Nos apoyamos del programa VLC, el cual puede actuar como Servidor y Cliente, facilitando la emisión y recepción de videos.

Una vez que los programas fueron desarrollados, se emplearon y configuraron tres computadores para así poder obtener los resultados deseados. El primer equipo actúa como el servidor de video. El segundo actúa como un intermediario el cual tiene la función de re-transmitir los datos recibidos desde el servidor y que luego envía al tercer equipo, quien viene a ser el cliente de la red.

En un principio, el desarrollo de este proyecto se estaba llevando a cabo con el entorno de desarrollo Netbeans, el cual fue usado para la programación y ejecución de los archivos que se iban creando. Sin embargo, transcurrido el tiempo, se optó por dejar de usar esa plataforma para la ejecución de los archivos, y a cambio se comenzó a usar la consola de sistema de Windows. Este cambio se produjo porque Netbeans utiliza muchos más recursos para funcionar, lo que afectaba en la transmisión de los videos, ya que constantemente los programas dejaban de funcionar y la transmisión se cancelaba.

6.2 Planteamiento de desarrollo para protocolos UDP y TCP.

Gráficamente, el uso de los equipos es como se plantea en la siguiente imagen

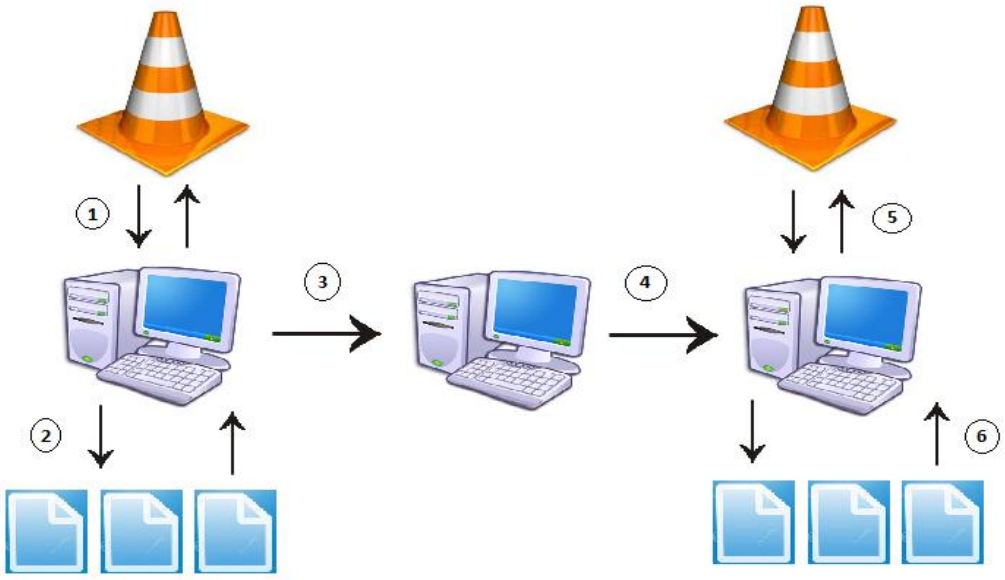

Figura 20 – Esquema de desarrollo del proyecto.

De izquierda a derecha se tienen los computadores: Servidor, Intermediario y Cliente.

6.2.1 Descripción etapas para protocolo UDP.

Para el protocolo UDP, el detalle de la enumeración para una mejor comprensión es como sigue a continuación.

1- El computador (Servidor) y VLC se comunican por un puerto establecido, y se utiliza la IP local del equipo para recibir la transmisión, ya que el programa VLC se encuentra en ese mismo nodo instalado.

2- El programa Java a nivel de Servidor, es configurado para poder comunicarse con el segundo nodo. Además, para las pruebas empleadas, las llaves de los algoritmos utilizadas son almacenadas como archivos en el equipo, también se debe seleccionar el tipo de vector de inicialización, el cual está ligado al tipo de algoritmo que se emplee. A esto se agrega, que los datos de transmisión recibidos del VLC Servidor, son recibidos y cifrados bajo algún protocolo para su posterior transmisión.

3- Los datos cifrados son enviados al nodo que actúa como intermediario. Para poder establecer esa comunicación es necesario definir los puertos por donde se envían los

datos y los puertos por donde se hace la recepción. Se debe considerar que esos puertos deben ser iguales. Además, el primer nodo, requiere la IP del nodo intermediario, de lo contrario no puede enviar los datos.

4- El nodo intermediario, reenvía los datos obtenidos hacia el tercer computador. Los datos no son modificados. Este nodo, requiere de la IP del tercer computador, y también se debe definir un puerto de comunicación.
5- Se tiene que establecer la comunicación entre el equipo y el VLC Cliente, por lo tanto, se debe definir otro puerto de comunicación, por donde el computador hará el envío de datos previo descifrado (6).
6- Con la ejecución del programa Java a nivel de cliente, los datos que son recibidos, son descifrados usando la llave del algoritmo correspondiente.

6.2.2 Descripción etapas para protocolo TCP.

La idea para el desarrollo de la aplicación Java para el protocolo TCP, son muy similares a los empleados en UDP. Sin embargo, al ser un protocolo distinto, sus características también cambian. Por ejemplo, para poder establecer la comunicación entre los diferentes elementos involucrados, se debe utilizar "ServerSocket", "Socket" (API Java).

1- El equipo debe, primero que todo, realizar una petición al programa VLC Servidor, para lo cual debe emplear el método GET (fuente). Cuando se tiene respuesta por parte del programa VLC, entonces se está habilitado para la recepción de datos. Es necesario también establecer el puerto y las IP's de comunicación necesarias para la recepción y envío de datos. [10]
2- Cuando los datos son obtenidos desde el programa VLC, se realiza la encriptación de estos, empleando una de las llaves de cifrado, asociada al algoritmo utilizado.
3- Se hace el envío de los datos cifrados, desde el nodo Servidor, hacia el nodo Intermedio.
4- El nodo Intermedio obtiene los datos y los vuelve a transmitir, esta vez, hacia el equipo Cliente.
5- Se debe establecer la comunicación entre el equipo y el VLC Cliente, para lo cual se emplea un puerto determinado y la IP del equipo (127.0.0.1). De esta forma los datos que llegan, pueden ser transmitidos y visualizados, previa desencriptación de los datos (6).
6- Los datos que llegan al equipo son descifrados, para lo cual se emplea una de las llaves, la que debe coincidir con el tipo de algoritmo que se está utilizando. Además, como en el

punto 2, se requiere el archivo asociado al vector de inicialización el cual es necesario para descifrar los datos.

6.3 Consideraciones generales

- Para ambos protocolos, hay características en los objetos de programación empleados, como por ejemplo el uso de "BufferedWriter", "BufferedReader", "DataOutputStream", "DataInputStream", "System.nanoTime()", "InetAddress", cuyas características, se pueden encontrar de forma detallada en la API de Java.
- Uno de los problemas que se presentaron al comienzo del desarrollo de las aplicaciones para ambos protocolos, fue el tamaño de datos máximo que soporta los elementos instanciados con "DatagramSocket" y "Socket", para UDP y TCP respectivamente. Para solucionar ese problema, fue necesario redimensionar el tamaño de recepción, empleando el método "setReceiveBufferSize", proporcionado por los elementos.

6.4 La biblioteca JCE.

La biblioteca JCE de Java brinda diferentes métodos para hacer uso de diversos algoritmos criptográficos. Los métodos y consideraciones necesarias para hacer un correcto uso de los algoritmos y que fueron utilizados en este proyecto, pueden verificarse en el documento [11].

- Las combinaciones de técnicas empleadas para el desarrollo de este proyecto son las siguientes:
    - DES + (CBC/CTR/ECB).
    - 3DES + (CBC/CTR/ECB).
    - AES + (CBC/CTR/ECB).

Con lo que se tiene un total de 9 combinaciones posibles para emplear. Mientras que para el uso del algoritmo RSA, se optó por utilizar sólo una combinación: RSA/ECB/PKCS1Padding, ya que el uso de esta técnica fue solamente para asegurar la veracidad de los datos

7 RESULTADOS OBTENIDOS

A continuación se presentan los resultados obtenidos que pueden ser expuestos en tres categorías que se describen a continuación. La desviación estándar que aparece en las tablas, equivalen a la dispersión que alcanzan los valores procesados en relación al promedio total obtenido.

7.1 Tiempos de procesamiento para el Cifrado y Descifrado de datos.

En este punto se exponen las gráficas que representan la variabilidad de tiempo empleado para el proceso en que los datos son cifrados (por parte del Servidor Java), y el proceso en los cuales son descifrados (por parte del Cliente Java).El desarrollo de estas pruebas es independiente del protocolo de comunicación empleado, por lo tanto se expondrán los gráficos cuya representación y visualización sean los más adecuadas para la comprensión.

- **Gráfico procesamiento de cifrado de datos (rango de 100 primeros datos).**

Tipo de algoritmo	Total de Datos (bytes)	Promedio (NanoSegundos)	Desviación estándar. (NanoSegundos)
AES	4021	154083,8304	71036,534
DES	4022	295293,5260	151391,421
Triple DES	4022	704515,8956	217553,363

Tabla 7 – Datos procesamiento cifrado de datos.

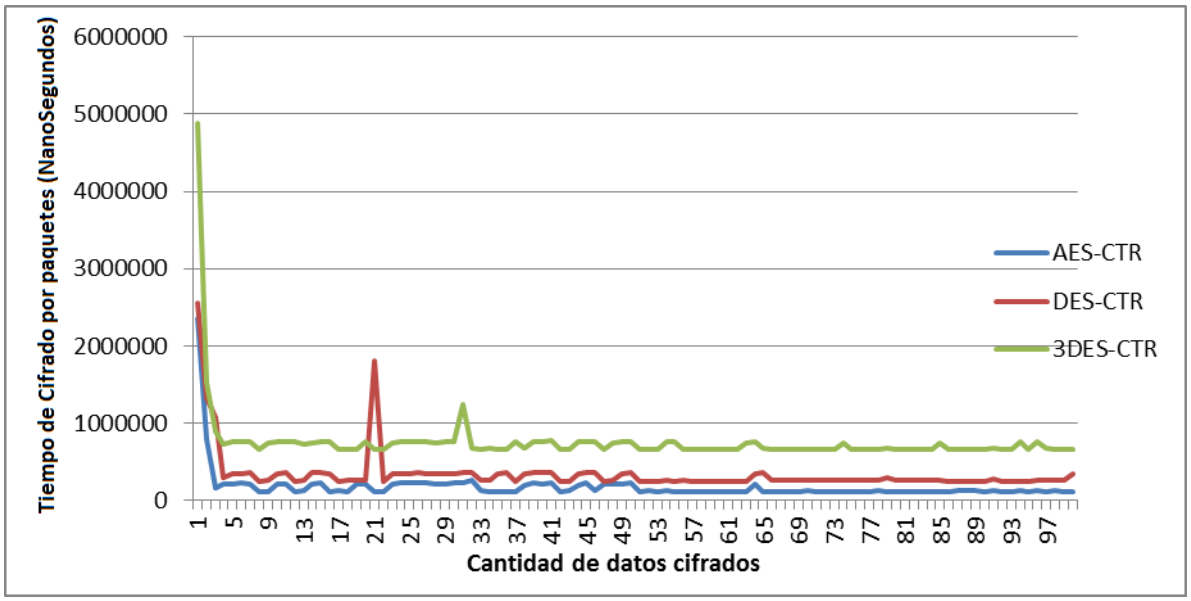

Figura 14 – Gráfico procesamiento cifrado de datos

- **Gráfico procesamiento de descifrado de datos (rango de 100 primeros datos).**

Tipo de algoritmo	Total de Datos (bytes)	Promedio (NanoSegundos)	Desviación estándar. (NanoSegundos)
AES	3994	138105,4031	34197,3958
DES	3996	281019,6759	48540,8105
Triple DES	4019	693425,01	71383,6087

Tabla 8 – Datos procesamiento descifrado de datos.

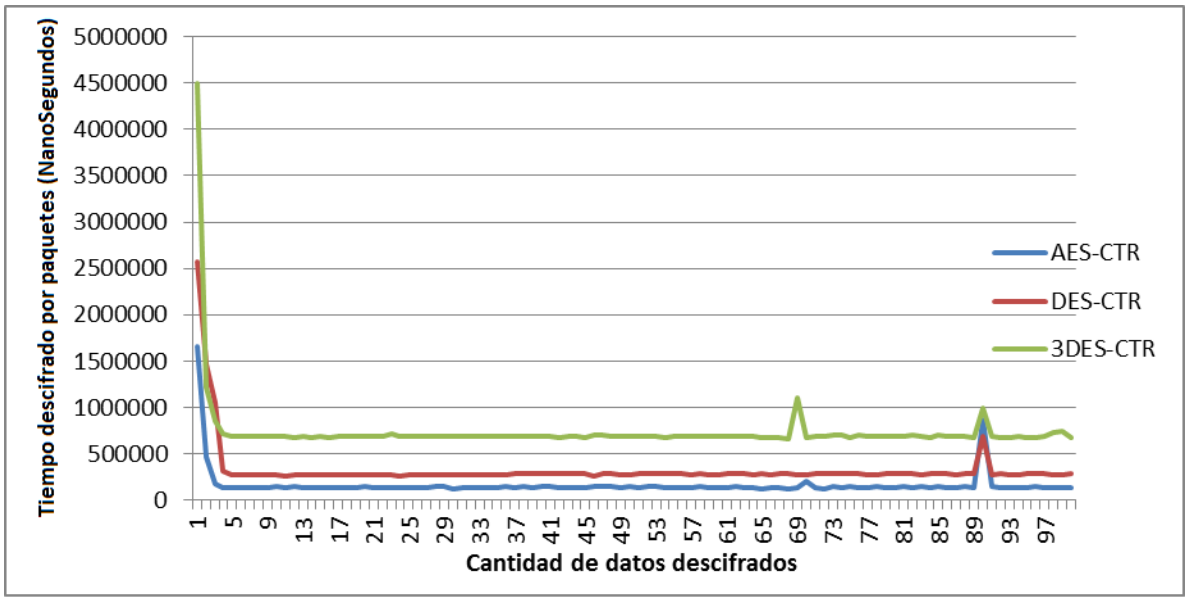

Figura 15 – Gráfico procesamiento descifrado de datos.

- Los gráficos expuestos anteriormente utilizan como cifrado la combinación de AES+CTR, el motivo de esta selección, es que durante la información recopilada se indica que no es conveniente utilizar ECB, pues esta metodología de cifrado es susceptible de ser quebrada [19] al contrario de CBC y CTR. Por otra parte, se toma como referencia el uso de AES, pues es el algoritmo que posee un proceso y variabilidad de tiempo mucho menores en comparación a DES y por consecuencia con Triple DES.

7.2 Tiempos entre transmisión y recepción de datos.

Para esta sección, se muestran los resultados considerando los protocolos de comunicación UDP y TCP. Esto, principalmente por tratarse de protocolos, con características y procedimientos diferentes, hace que finalmente los resultados posean cierta variabilidad.

El tiempo entre transmisión y recepción de datos es calculado considerando la diferencia existente entre el primer dato y el dato siguiente que viene después del proceso de encriptación o desencriptación según sea el caso (Cliente o Servidor).

Para la muestra de estos gráficos se considerará el uso de AES + CTR (pues como se mencionó anteriormente, es el algoritmo que mejor comportamiento tiene), versus, la recepción/transmisión de datos sin emplear algoritmos criptográficos.

Para el desarrollo de los gráficos números 16, 17, 18 y 19, se calculó la diferencia existente entre el tiempo empleado para la transmisión/recepción, utilizando el algoritmo AES y el tiempo empleado para la transmisión/recepción sin utilizar algún algoritmo criptográfico.

7.2.1 Tiempos entre transmisión/recepción para protocolo UDP.

- **Transmisión de datos.**

Tipo de algoritmo	Total de Datos (bytes)	Promedio (NanoSegundos)	Desviación estándar. (NanoSegundos)
AES	4018	15946776,05	27650910,6
Sin algoritmo	4019	15869526,14	50084451,1

Tabla 9 – Datos tiempos entre transmisión de datos UDP

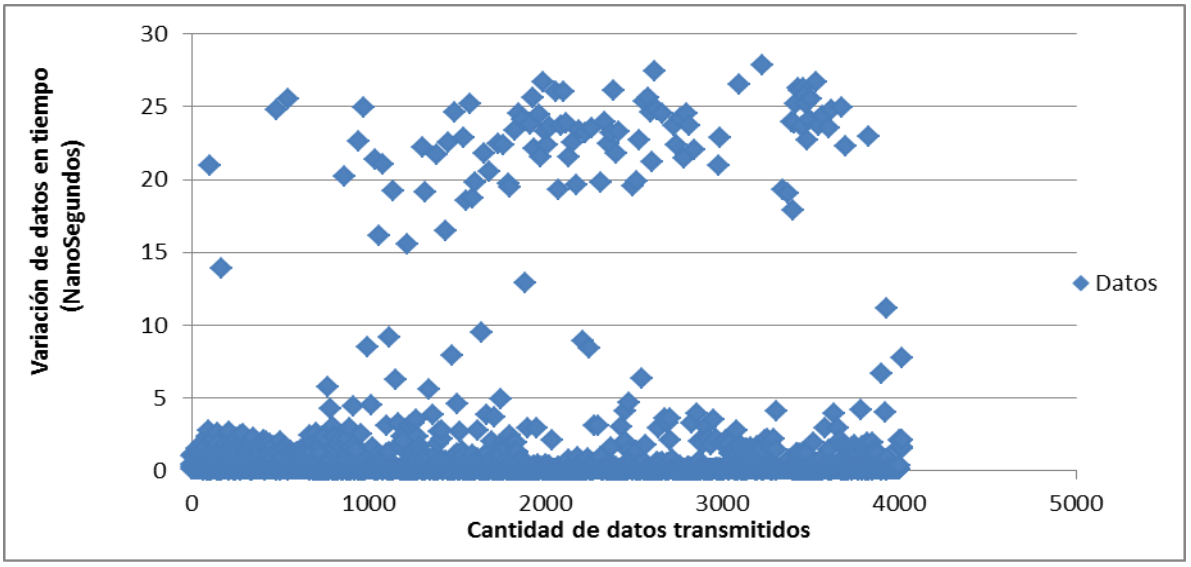

Figura 16 – Dispersión datos transmitidos por UDP.

- Como se puede apreciar en la tabla número 9, el promedio entre emplear y no emplear algoritmos es bastante mínima, lo cual se puede ver reflejado en el gráfico de dispersión de la figura 16, en el que se observa como los datos tienden a estar cerca del valor 0, lo cual tiene relación con la poca, o casi nula diferencia entre los promedios establecidos.

    - **Recepción de datos.**

Tipo de algoritmo	Total de Datos (bytes)	Promedio (NanoSegundos)	Desviación estándar. (NanoSegundos)
AES	4015	15863469,52	30596944,4
Sin algoritmo	4001	15862700,2	49861694,7

Tabla 10 – Datos tiempos entre recepción de datos UDP

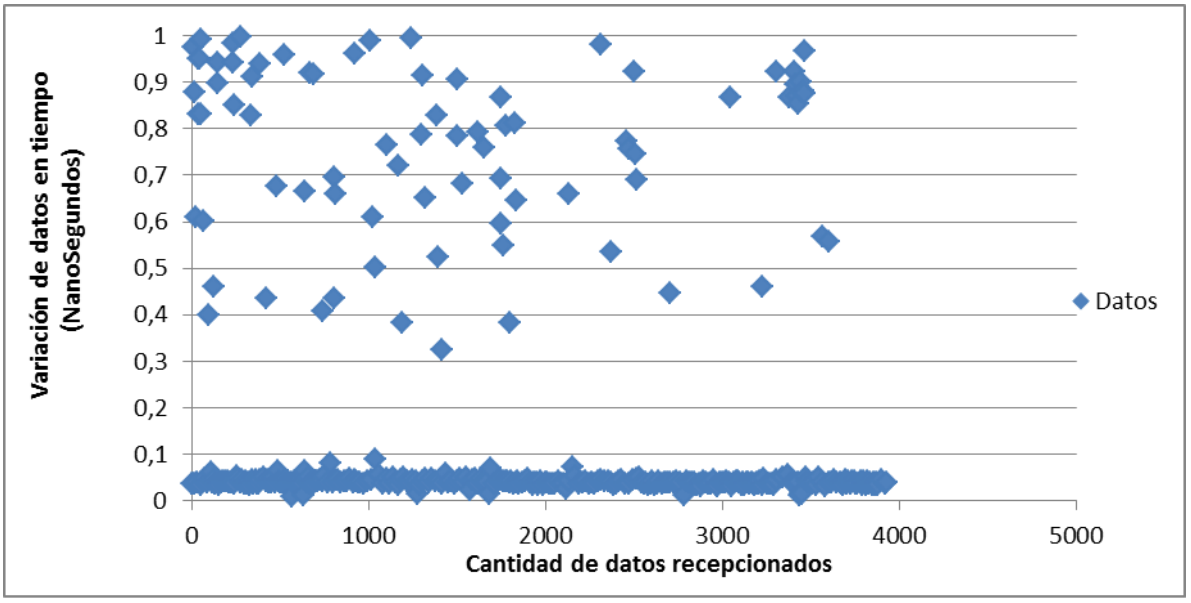

Figura 17 – Dispersión datos recepcionados por UDP.

- Para el caso de la recepción de los datos, ocurre algo bastante similar. La tendencia indica que los datos dispersos que se muestran en la figura 17 bordean y/o sobrepasan en ciertos momentos el valor de 0, que al igual que el punto anterior, se debe a la poca diferencia existente entre emplear o no cifrado de datos.

7.2.2 Tiempos entre transmisión/recepción para protocolo TCP.

- **Transmisión de datos.**

Tipo de algoritmo	Total de Datos (bytes)	Promedio (NanoSegundos)	Desviación estándar. (NanoSegundos)
AES	2706	23439409,43	33940971,2
Sin algoritmo	2707	23163559,56	75732697,7

Tabla 11 – Datos tiempos entre transmisión de datos TCP

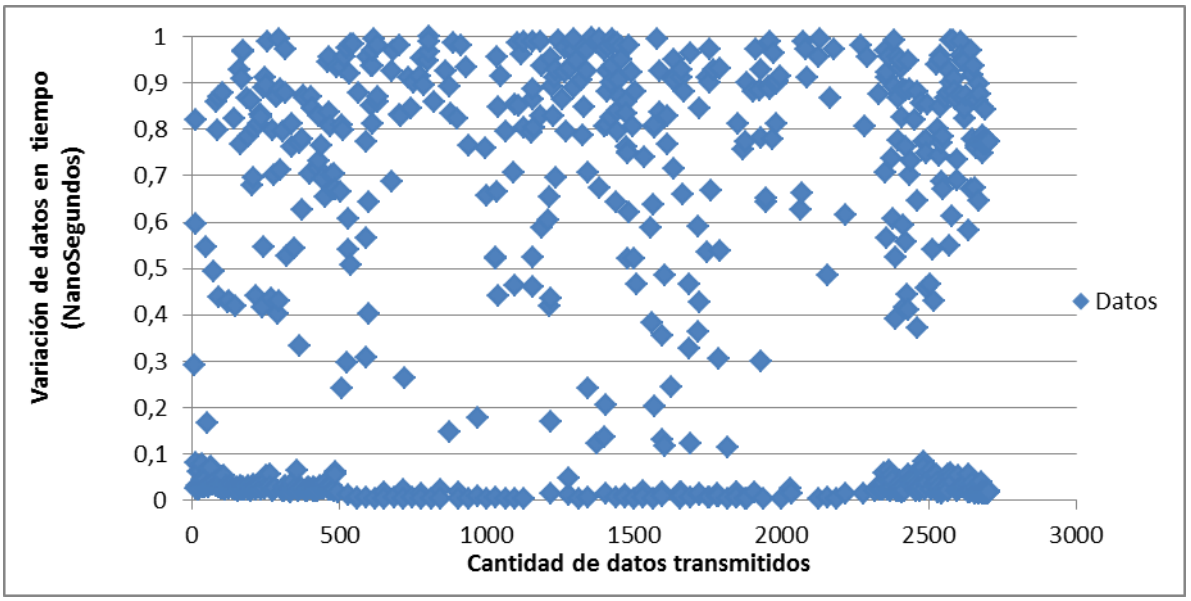

Figura 18 – Dispersión datos transmisión por TCP.

- En la figura 18, se puede apreciar que existe una mayor dispersión en la cantidad de datos transmitidos. Un aspecto importante que creemos que influye en estos resultados, es en las características propias que posee TCP ya que a diferencia de UDP, podemos decir que TCP es un protocolo que opera con funcionalidades más complejas que las de UDP, por ejemplo, TCP comprueba que todos los datos sean transmitidos, y en caso que exista un error en la llegada de datos, solicita nuevamente al servidor el envío del dato que falte. Este factor de influencia es relevante a la hora de apreciar los resultados. Pareciera que existen dos grupos para dispersión, los cuales tienden a acercarse a valores de 0 y 1.

- **Recepción de datos.**

Tipo de algoritmo	Total de Datos (bytes)	Promedio (NanoSegundos)	Desviación estándar. (NanoSegundos)
AES	2673	23957646,44	36397878,4
Sin algoritmo	2673	21757406,61	64620437,8

Tabla 12 – Datos tiempos entre recepción de datos TCP

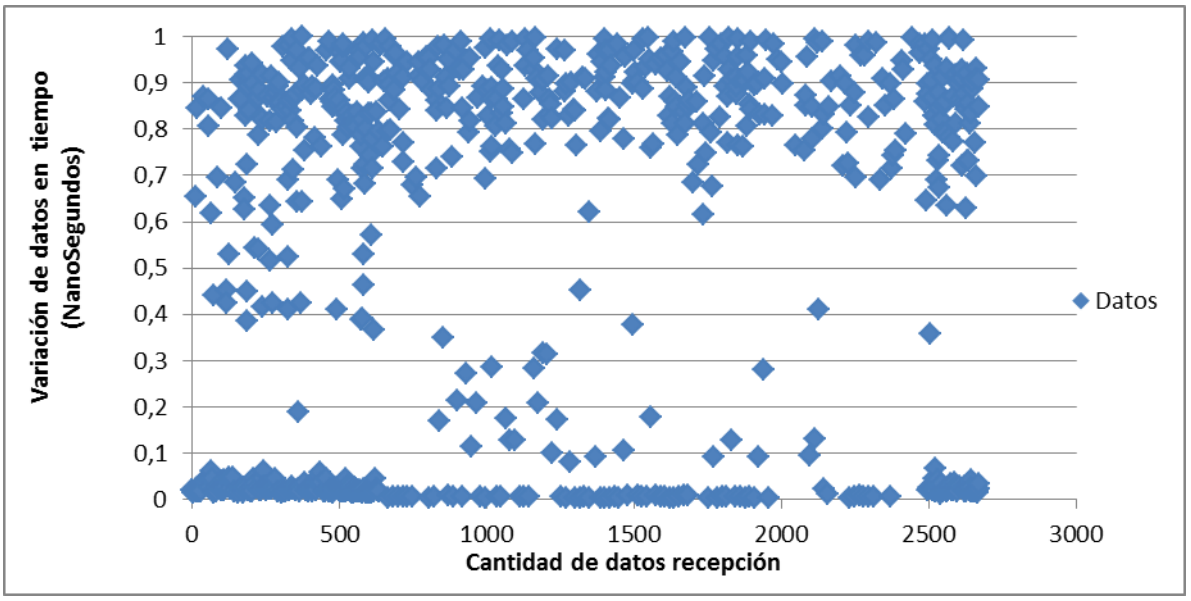

Figura 19 – Dispersión datos recepcionados por TCP

- Finalmente, y al igual que el gráfico de envío de datos, la figura 19 muestra que existe una amplia dispersión de los datos, los que tienden a "agruparse" cerca de los valores de 1 y 0. Los fundamentos por los cuales se presenta esta variabilidad, pasan por los mayores "precauciones" que posee TCP en lo que es transmisión/recepción de datos, a diferencia de UDP.

7.3 Tiempos procesamiento de RSA.

Al igual que el punto 7.1 de esta sección, los tiempos asociados al procesamiento de RSA son genéricos, y por tanto son independientes del tipo de protocolo de comunicación, por lo que sólo se mostrará aquella prueba que posee un mejor desempeño y por ende un resultado más estable.

Para este caso, se compara el uso de AES con la aplicación de criptografía asimétrica, y sin el uso de ésta, cuyos datos por tanto, equivalen a los presentados también en el punto 10.1.

Tipo de algoritmo	Total de Datos (bytes)	Promedio (NanoSegundos)	Desviación estándar. (NanoSegundos)
AES sin RSA	4021	154083,8304	71036,534
AES con RSA	4822	1511823,119	4350068,551

Tabla 13 – Datos tiempos de procesamiento para RSA.

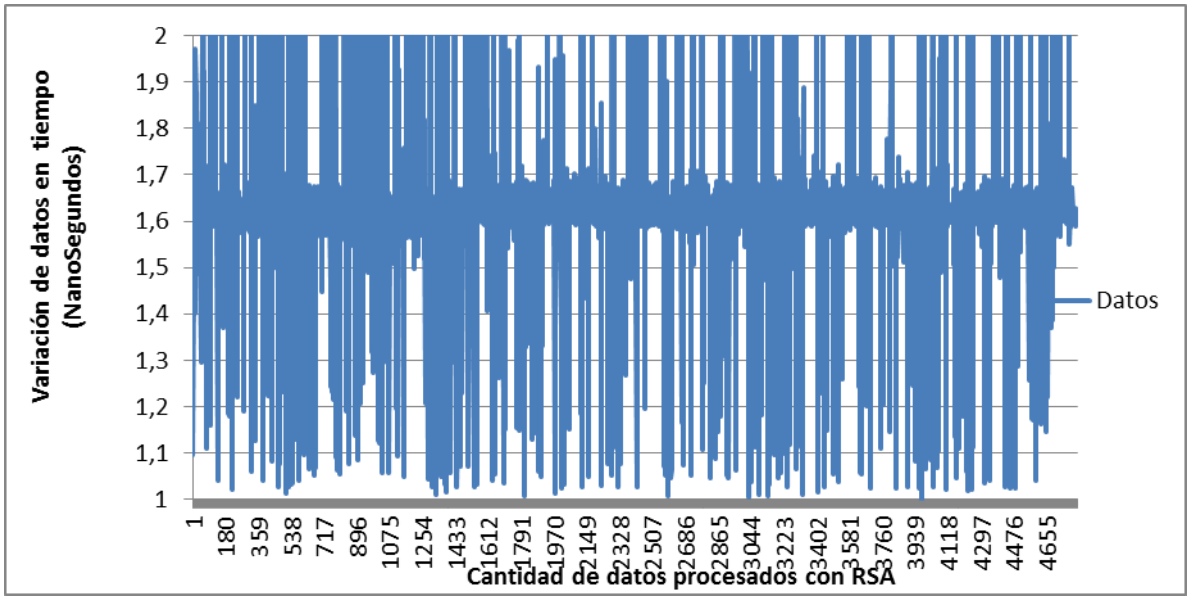

Figura 20 – Dispersión datos empleando RSA.

En la figura 20 se muestra la razón entre el tiempo de procesamiento en emplear y no utilizar RSA, en donde se aprecia que si bien existe una alta variabilidad de los datos, existe cierta tendencia y agrupamiento de datos para un valor de razón aproximado a 1,6. Esta concentración de datos implica que existe aproximadamente un 60% de impacto en el uso del algoritmo, lo cual ciertamente es acertado pues, el proceso y uso de RSA es mucho más costoso que emplear solamente los algoritmos simétricos.

7.4 Conclusiones de resultados obtenidos.

Las conclusiones que se han obtenidos a lo largo del desarrollo de las pruebas, demuestran primero que todo, que entre los algoritmos criptográficos de clave simétrica, AES posee un mejor rendimiento en cuanto al tiempo empleado, tanto para el caso de encriptar como descencriptar los datos. Se verificó también, que los modos de cifrado, es decir, CBC, ECB, y CTR, no tienen un mayor impacto en los procedimientos, y tampoco tienen mayor influencia en los tiempos finales obtenidos.

Existen diversos factores que se deben considerar para entender de buena forma toda la variabilidad existente en los gráficos expuestos. Por ejemplo, la cantidad de procesos que el o los computadores utilizados hayan estado ejecutando al momento de las pruebas, la conexión de red, que si bien es local, pues se emplearon los laboratorios, también puede haber tenido momentos de mayor o menor uso.

En cuanto a los protocolos de comunicación empleado, se observó que el uso de TCP impacta radicalmente la dispersión de datos analizados, pues como se comentó anteriormente, al ser un protocolo que garantiza una transferencia confiable de datos, que no es el caso de UDP, TCP debería tender a emplear mayor tiempo para la realización de sus operaciones. Mientras que como UDP es un protocolo no orientado a la conexión, no invierte tiempo en la ejecución de mecanismos de recuperación ante la corrupción o pérdida de datos ni tampoco control de flujo.

Finalmente, para el uso de criptografía asimétrica, especialmente RSA, se tiene que la empleabilidad de esta técnica, implica un alto costo en los recursos y tiempos que se obtuvieron, los cuales si bien, están en un rango de NanoSegundos, la suma de cada uno de estos momentos, puede llegar a significar un tiempo de procesamiento y uso mucho más radical que solo el empleo de algoritmos simétricos.

8 PRUEBAS

Adaptación basada en [12].

8.1 Elementos de prueba

- Programa nº1, Servidor Java.

  Este programa se encarga de recibir los datos asociados al video transmitido por el programa VLC Servidor. Cuando los datos son recepcionados, éstos se encriptan de acuerdo al algoritmo y modo de cifrado establecido. Finalmente, los datos cifrados, son transmitidos al programa intermediario Java.

- Programa nº2, Cliente Java.

  Este programa se encarga de recibir los datos encriptados que han sido re-enviados por el programa intermediario Java. Cuando los datos son recibidos, estos se descencriptan de acuerdo al algoritmo y modo de cifrado establecido, consiguiendo con esto los datos originales que ha enviado el programa VLC Servidor. Finalmente, los datos originales son transmitidos hacia el programa VLC Cliente, que es donde se visualizará el video.

8.2 Especificación de las pruebas.

Características a probar.	Nivel de prueba	Objetivo de la Prueba	Enfoque para la definición de casos de prueba	Técnicas para la definición de casos de prueba	Actividades de prueba	Criterios de cumplimiento
Funcionalidad	Aceptación	Se busca determinar el correcto funcionamiento de las combinaciones necesarias para el cifrado/descifrado, empleando la librería JCE.	Caja Negra	Configuración de parámetros.	Se requiere importar la librería JCE dentro del código desarrollado.	Recuperación de datos originales a partir de los datos encriptados.
Funcionalidad	Aceptación	Verificar la transmisión de los datos desde el programa VLC Servidor hacia el programa Java Servidor.	Caja Negra	Transmisión de datos.	El programa Java Servidor, y VLC servidor deben estar configurados con el nº de puerto establecido para la correcta transmisión de datos.	Recepción de datos enviados desde VLC Servidor.
Funcionalidad	Aceptación	Verificar la transmisión de los datos desde el programa VLC Servidor, pasando por los tres programas Java (Servidor-Intermediario-Cliente), hasta el programa VLC Cliente, sin emplear criptografía.	Caja Negra	Transmisión de datos	Se requiere establecer una configuración entre los puertos, direcciones IP necesarios.	Visualización completa del vídeo transmitido.
Funcionalidad	Aceptación	Verificar la transmisión de los datos desde el programa VLC Servidor, pasando por los tres programas Java (Servidor-Intermediario-Cliente), hasta el programa VLC Cliente, empleando métodos criptográficos.	Caja Negra	Transmisión de datos.	Se requiere establecer una configuración entre los puertos, direcciones IP y los parámetros de cifrado/descifrado.	Visualización completa del vídeo transmitido.

Tabla 14 – Especificación de pruebas.

8.3 Responsables de las pruebas

- Heber Gálvez Ojeda, Alumno.
- Patricio Galdámes Sepúlveda, Profesor Guía.

8.4 Calendario de pruebas

Calendarización de las distintas actividades de prueba que serán realizadas, ya sean por elementos, niveles o características.

De la planificación adjuntada en la carta Gantt, se tiene la siguiente sección desarrollada a las pruebas de la investigación.

GANTT project

Nombre	Fecha de inicio	Fecha de fin
Desarrollo de aplicación.	1/10/13	10/01/14
Aplicación prototipo para recepción de datos.	1/10/13	12/10/13
Iteración - Modificación de prototipo para protocolos de comunicación	14/10/13	24/10/13
Presentación de transmisión de datos por protocolo de comunicaci...	14/10/13	24/10/13
Estudio de librería JCE	25/10/13	28/10/13
Iteración - Agregar funcionalidad de criptografía simetrica	30/10/13	14/11/13
Modificación para programa con protocolo UDP.	30/10/13	8/11/13
Modificación para programa con protocolo TCP.	9/11/13	14/11/13
Desarrollo de primeras pruebas de funcionalidad	15/11/13	16/11/13
Iteración - Agregar funcionalidad de criptografía asimetrica	18/11/13	20/11/13
Modificación para programa con protocolo UDP	18/11/13	19/11/13
Modificación para programa con protocolo TCP	19/11/13	20/11/13
Desarrollo de segundas pruebas de funcionalidad.	22/11/13	27/11/13
Capacitación a profesor Guía	27/11/13	27/11/13
Configuración de equipos en laboratorio	29/11/13	10/01/14
Configuración para programas de protocolo UDP	29/11/13	4/12/13
Etapa de pruebas para obtención de resultados	5/12/13	13/12/13
Configuración para programas de protocolo TCP	16/12/13	20/12/13
Etapa de pruebas para obtención de resultados.	3/01/14	10/01/14
Validación y finalización de pruebas	10/01/14	10/01/14

Figura 21 – Calendario de pruebas, Carta Gantt.

8.5 Detalle de las pruebas.

Pruebas de Unidad.

Si bien, se han desarrollado tres programas en Java, los programas Cliente y Servidor son los que más importancia tienen en este proyecto. Ahora bien, independiente del tipo de protocolo que se esté empleando, de forma general, ambos programas requieren el mismo tipo de configuración para que cumplan sus objetivos. Por este motivo, se ha decidido establecer una sola tabla, la cual posee todas las pruebas empleadas.

8.5.1 <Transmisión Cliente/Servidor>

- Configuración de Hardware: Para las pruebas desarrolladas, no se realizaron mayores configuraciones que las ya existentes en los laboratorios de la universidad.
- Configuración de Software: Las configuraciones de software requeridas corresponden a definir los parámetros necesarios para la encriptación de los datos. Estos parámetros deben ser los mismos para los programas Cliente y Servidor. Los puertos de comunicación deben estar predefinidos, al igual que las direcciones IP. Por otra parte, se requiere la configuración para la transmisión y visualización de los datos en la aplicación VLC Media Player.
- Configuración del Sistema Operativo: Las pruebas desarrolladas en el laboratorio de redes, requiere de la desactivación del Cortafuegos (Firewall) de Windows, ya que de lo contrario, éste bloquea las entradas y salidas de los datos transmitidos.

ID Caso De Prueba	Características a Probar	Datos de Entrada							Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
		D1	D2	D3	D4							
P-01	Recepción de datos desde VLC Servidor.	Protocolo de comunicación	Algoritmo Criptográfico	Tipo de Cifrado	Datos de Transmisión				Datos originales de transmisión.	Datos originales de transmisión	Éxito.	La recepción de datos que se prueba es desde VLC Servidor, hacia el programa Java Servidor.
P-02	Transmisión de datos desde programa Servidor hacia programa intermediario.	Datos de transmisión de video encriptados.							Datos encriptados.	Datos encriptados	Éxito.	Se valida que los datos transmitidos no han sido alterados.
P-03	Descifrado de datos recibidos en programa Cliente.	Protocolo de comunicación	Algoritmo Criptográfico	Tipo de Cifrado	Datos de transmisión.				Visualización de video.	Visualización de video original.	Éxito.	El procedimiento responde de manera correcta, y se logra visualizar el video transmitido.
P-04	Configuración para la transmisión del video.	Protocolo de comunicación	Algoritmo Criptográfico	Tipo de Cifrado					Mensaje de validación.	Mensaje de validación.	Éxito.	Errores producidos solo por mala configuración, como por ejemplo la selección de llaves de cifrado.
P-05	Aplicación de firma digital RSA.	Protocolo de comunicación	Algoritmo Criptográfico	Tipo de Cifrado					Recepción de firma establecida	Recepción de firma establecida	Éxito	Se debe verificar el tamaño del buffer para la correcta transmisión de la firma.

Tabla 15 – Detalle de pruebas

8.6 Conclusiones de Prueba

Durante el proceso de pruebas, las conclusiones que se han obtenido pasan principalmente por la motivación requerida para configurar y ejecutar los programas desarrollados. Esto principalmente, porque se considera que no es una tarea que se debe realizar a la ligera, debe existir como en todo el proyecto, tiempo y una planificación adecuada que permita ir progresando de manera constante en la mejora de las funcionalidades de los programas desarrollados.

En cuanto a los resultados obtenidos, se puede apreciar claramente las diferencias existentes en la aplicación de las técnicas criptográficas empleadas. Como principal novedad está el desempeño del algoritmo AES, en lo que se refiere a cifrado y descifrado.

En general el rendimiento en función del tiempo, variará de acuerdo a la combinación de parámetros que se han establecidos para la encriptación de los datos.

9 CONCLUSIONES

Durante el período en el que se llevó a cabo esta investigación, y los objetivos planteados en primera instancia, se puede argumentar que hubo un correcto desempeño en el trabajo realizado. Los objetivos del proyecto y del sistema fueron cumplidos, lo que se puede verificar con todos los resultados que se obtuvieron en la etapa de pruebas.

Las herramientas empleadas fueron las adecuadas ya que brindaron en todo momento las funcionalidades requeridas. Java, como lenguaje de programación es muy versátil, más cuando existen diferentes librerías que entregan soluciones a las problemáticas que surgieron en el transcurso de la creación de los programas.

El desarrollo de una buena planificación al comienzo, y el uso de una metodología de trabajo, facilitan la forma de llevar a cabo las diversas tareas que nacen en este proyecto. Se puede validar la importancia que implica tener siempre un orden y consistencia para trabajar, lo que se traduce en el cumplimiento de las expectativas planteadas al momento de presentar este tema.

A modo académico, el desarrollo de este proyecto, sirvió para ahondar en un área más específica de la informática como es el tema de la seguridad basada en la criptografía. Se pudo poner en práctica muchos de los aspectos enseñados durante estos cinco años de estudio. En lo que se refiere a los aspectos prácticos, puedo decir que Java es un gran lenguaje de programación, del cual prácticamente se pueden desarrollar cualquier tipo de aplicaciones. La portabilidad, transparencia, y la gran cantidad de personas que lo emplean, hace que sea una poderosa herramienta para el desarrollo de softwares.

En lo personal, el proyecto me ha servido para demostrar aspectos asociados a valores humanos, como por ejemplo, la responsabilidad, persistencia, motivación, respeto. Todos esos factores, creo que tienen una gran importancia, pues fueron fundamentales para sacar adelante esta investigación.

10 BIBLIOGRAFÍA

[1] Caballero Gil, Pin, Introducción a la criptografía, 2ª edición actualizada, Madrid, Ra-Ma, 2002.

[2] DEMRE, Proceso de admisión 2013 Etapa de Selección [en línea], Santiago de Chile, < http://www.demre.cl/text/pdf/p2013/seleccion_proc_2013.pdf >, [consulta: 22 Enero 2014].

[3]Ministerio de Educación, Fija Estatuto de la Universidad del Bío-Bío, 19 Octubre de 1989, Santiago, 1989, 11p.

[4]Universidad del Bío-Bío, Estructura Orgánica de la Universidad del Bío-Bío,[en línea], <http://www.ubiobio.cl/miweb/web2012.php?id_pagina=3295>, [consulta 19 Enero 2014].

[5] IEEE, IEEE recommended practice for Software Specification, 06 Agosto 2002,1998, 40p.

[6]Jorge Sánchez Arriazu, Descripción del algoritmo DES (Data Encyption Standard), [en línea], < http://www.tierradelazaro.com/public/libros/des.pdf>, [Consulta: 21 Noviembre 2013].

[7]Adrian Pousa, Algoritmo de cifrado simétrico AES "Aceleración de tiempo de computo sobre arquitecturas multicore", [en línea], Argentina, <http://postgrado.info.unlp.edu.ar/Carreras/Especializaciones/Redes_y_Seguridad/Trabajos_Finales/Pousa_Adrian.pdf>, [consulta: 25 Octubre 2013].

[8]Jose de Jesús Angel Angel, AES – Advanced Encryption Standard, [en línea], México, <http://computacion.cs.cinvestav.mx/~jjangel/aes/AES_v2005_jjaa.pdf>, [consulta: 02 Octubre 2013].

[9]DI Managment Services, Using padding in Encryption, [en línea], <http://www.di-mgt.com.au/cryptopad.html>, [consulta: 18 Noviembre 2013].

[10] Centro de Innovación para la sociedad de la información, HTTP "Hyper Text Transfer Protocol", [en línea], <http://www.cicei.ulpgc.es/ocon/gsi/tut_tcpip/3376c426.html>, [consulta 29 Noviembre 2013].

[11]Brad Rubin, Java Security Part 1: Crypto Basics,[en línea], < https://www6.software.ibm.com/developerworks/education/j-sec1/j-sec1-pdf.pdf>, [consulta 12 Noviembre 2013].

[12] IEEE, Standard for Software and System Test Documentation, 18 Julio 2008,2008, 150p.

[13] Sergio Valero Orea, Estimación de proyectos de Software con puntos de casos de uso, [en línea], México, <http://www.utim.edu.mx/~svalero/docs/id45.pdf >, [consulta 6 Febrero 2014]

[14]Universidad del Bío-Bío, "Organigrama", [en línea], <http://ubiobio.cl/miweb/web2012.php?id_pagina=5152 >, imagen.

[15]Manuel Luna, Imagen Cifrado Polybios, [en línea], < http://manuelluna08.blogspot.com/> imagen.

[16]Wikipedia, Imagen Cifrado de César, [en línea], <http://commons.wikimedia.org/wiki/File:Caesar3.svg>, imagen.

[17]Pedro Gutiérrez, Imágenes cifrado con llave pública y privada, [en línea], < http://www.genbetadev.com/seguridad-informatica/manual-de-gpg-cifra-y-envia-datos-de-forma-segura>, Imagene.

[18] Wikipedia, Imágenes Estructura CBC, CTR, ECB, ejemplo uso de ECB, [en línea], <http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation>, Imágenes (x7).

[19]Mark Stamp, Information Security Principles and Practice, 2ª edición WILEY Publication, 2011.

11 ANEXO: ESTUDIO DE FACTIBILIDAD

11.1 Factibilidad técnica.

Software

Nombre	Versión	Licencia	Descripción
Netbeans IDE	7.3.1	CDDL y GNU General Public License.	Entorno de desarrollo.
VLC Media Player	2.0.8 o superior.	GNU General Public License.	Reproductor multimedia.
Java	1.7.0_40	Licencia de libre uso.	Lenguaje de programación.
Microsoft Windows	XP	Get Genuine Windows Agreements (Para instituciones académicas)	Sistema Operativo
Ubuntu	11.04 o superior.	Software libre.	Sistema Operativo.

Tabla 16 – Factibilidad técnica, Software.

- Para los recursos asociados a Hardware, se emplearon los equipos y la red con la que dispone la universidad, específicamente el laboratorio de redes, ubicado en el edificio tecnológico.

- Para el desarrollo del proyecto, se utilizó Java como lenguaje de programación, del cual existen los conocimientos básicos necesarios para el desarrollo de las aplicaciones. A medida que el proyecto iba en desarrollo, también se investigaron elementos fundamentales para la creación correcta de los códigos asociados a los archivos, los cuales debían cumplir ciertas características de acuerdo a los protocolos de comunicación que se utilizaron.
- En cuanto al uso del programa VLC, no hubo mayores complejidades en el uso de esta aplicación, pues su interfaz es lo bastante simple, como para poder trabajar sin inconvenientes.

11.2 Factibilidad operativa.

El impacto más relevante que se genera con esta investigación, está ligada a los resultados que se obtuvieron en las pruebas desarrolladas, ya que estos resultados pueden ser una base para un estudio más profundo respecto a los algoritmos criptográficos y la transmisión de datos multimedia a través de la red.

Además, el desarrollo de las aplicaciones podría ser una base para quien desee elaborar algún sistema, aplicación u otra investigación, cuyo foco de desarrollo este basado en el tema de este informe.

11.3 Factibilidad económica.

Como se mencionó anteriormente, este proyecto tiene como principal característica ser un trabajo de investigación y por ende no es un proyecto comercial.

Sin embargo, a modo de supuesto y considerando algunos aspectos básicos se pueden elaborar un análisis para determinar la viabilidad económica de este trabajo.

Primero que todo, para el flujo de caja empleado, se consideran los siguientes aspectos:

- Inversión Inicial.
- Ingresos, para lo cual se plantea el supuesto de ofrecer suscripciones a los usuarios por el servicio de streaming, y también ingresos asociados a publicidad a alguna compañía externa.
- Egresos, asociado a ciertos costos que deben ser cubiertos por el período de tiempo del proyecto

11.3.1 Flujo de Caja.

De acuerdo a la explicación anterior, un flujo de caja básico para cumplir los requerimientos de la factibilidad económica, sería como se presenta a continuación.

	Año 0	Año 1	Año 2	Año 3
Inversión Inicial				
Computadores x 3	$ 250.000			
Cables de red	$ 10.000			
Total Inversión	**$ 760.000**	**$ 0**	**$ 0**	**$ 0**
Ingresos				
Suscripciones		$ 250.000,00	$ 280.000,00	$ 330.000,00
Publicidad		$ 150.000,00	$ 230.000,00	$ 270.000,00
Total Ingresos		**$ 400.000,00**	**$ 510.000,00**	**$ 600.000,00**
Egresos				
Servicio Internet		$ 70.000,00	$ 95.000,00	$ 97.000,00
Electricidad (CV)		$ 75.000,00	$ 72.500,00	$ 81.000,00
Total Egresos		**$ 145.000,00**	**$ 167.500,00**	**$ 178.000,00**
Saldo Neto Actual	**-$ 760.000,00**	**$ 255.000,00**	**$ 342.500,00**	**$ 422.000,00**

Tabla 17 – Flujo de caja.

Ahora bien, para poder determinar la viabilidad de este proyecto, es necesario determinar los valores de VAN y TIR.

Haciendo uso de Excel, y tomando una de interés del mercado de un 5%, los resultados obtenidos son los siguientes.

Tasa Interés Activa :	5%
VAN	$ 158.054,21
TIR	15%

Con estos resultados, se puede concluir que el proyecto es viable de realizar.

Recordar que este flujo de caja y resultados solo es a modo de supuesto, y no se emplearon en el desarrollo de este proyecto.

11.4 Conclusión de la factibilidad

Como resultado de las factibilidades planteadas para este proyecto, se puede concluir que existieron en todo momento las herramientas y conocimientos necesarios para llevar a cabo esta investigación.

En cuanto al aspecto económico, si se considerara emplear esta investigación como base para el desarrollo de algún sistema más complejo, los resultados económicos debieran ser similares, en cuanto a viabilidad se refiere, puesto que para este caso, los elementos que se consideraron fueron simples y con valores cercanos a la realidad de los implementos utilizados.

12 ANEXO: ANÁLISIS

12.1 Diagrama de Flujo de Datos

- **Diagrama de Contexto.**

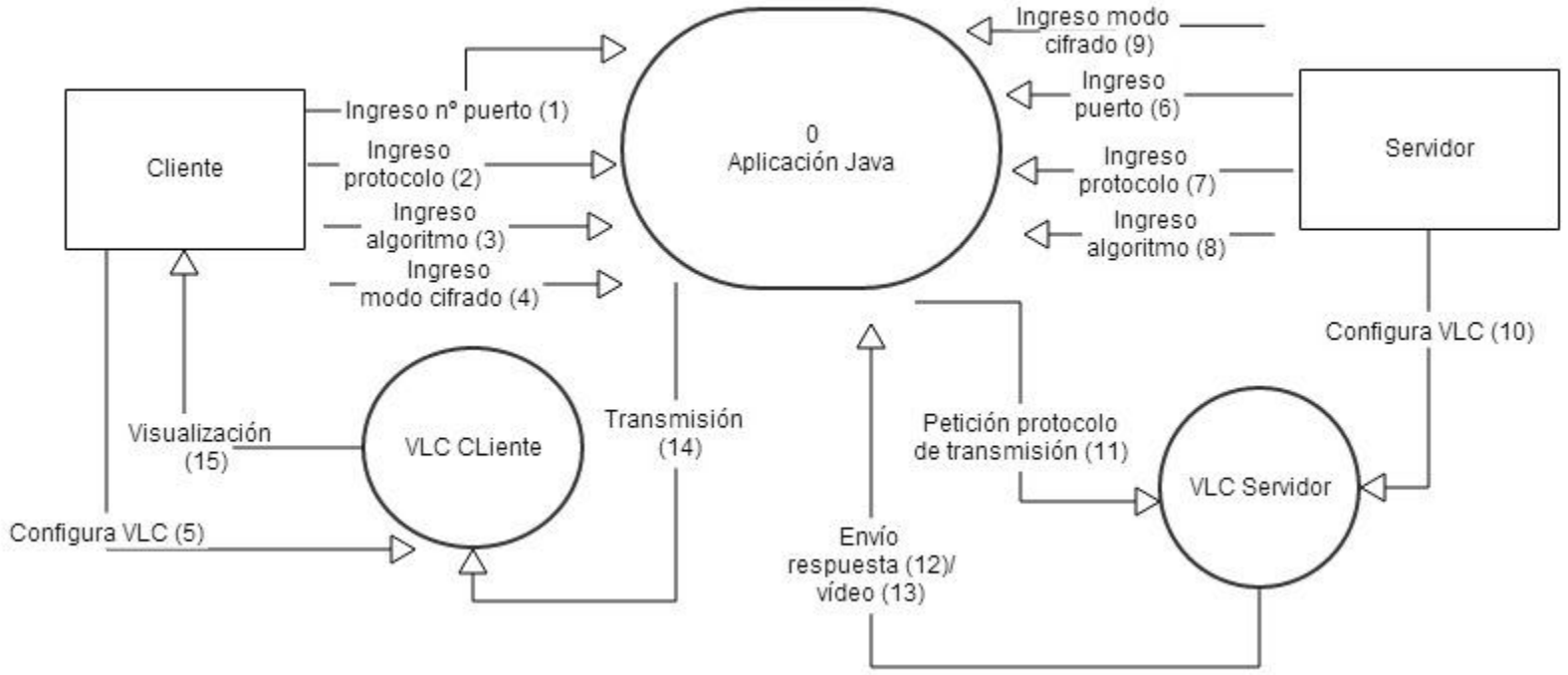

Figura 22 – Diagrama de Contexto.

- **Diagrama Superior.**

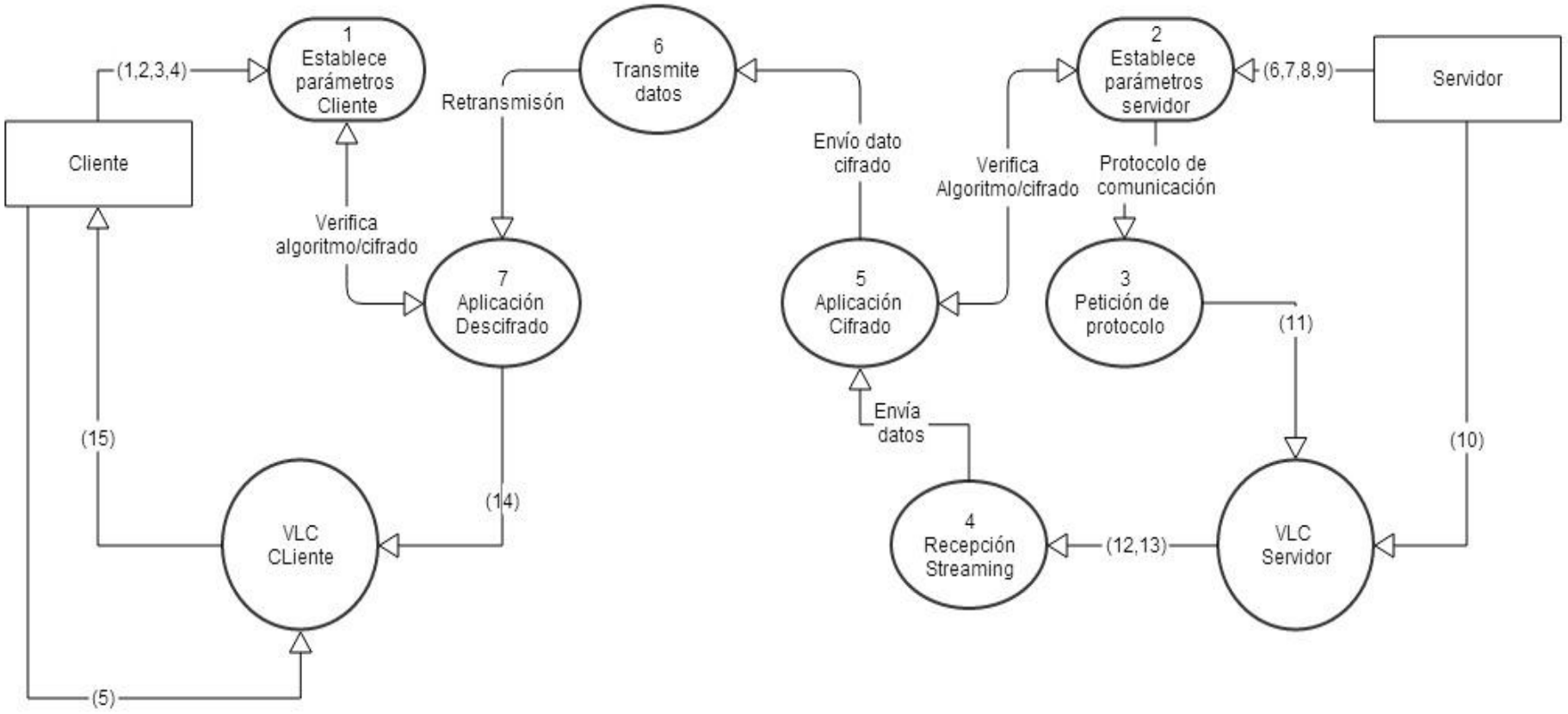

Figura 23 – Diagrama Superior.

- **Diagrama de Detalle.**

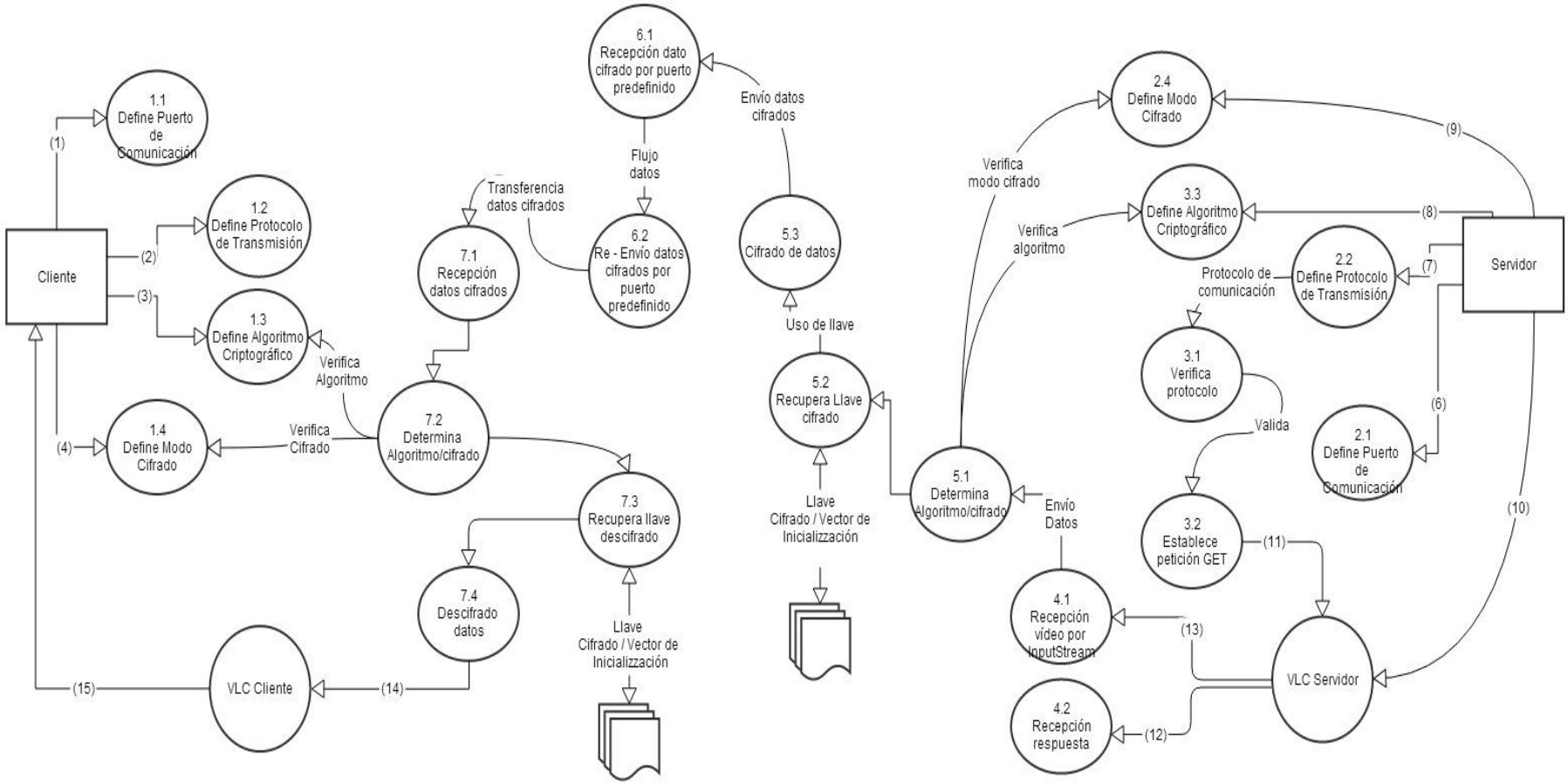

Figura 24 – Diagrama Detalle

12.2 Diagrama de casos de uso.

12.2.1 Actores

- **Cliente**.
    - **Rol**: Dentro de la aplicación, el cliente es quien recibe la transmisión de videos.
    - **Nivel de conocimientos requeridos**: EL cliente debe poseer un conocimiento técnico en un nivel medio, principalmente para la configuración de la interfaz de la aplicación y también la configuración del programa VLC (Cliente).
    - **Nivel de privilegios**: Nivel intermedio, ya que debe realizar las configuraciones necesarias y que han de ser indicadas por el actor Servidor.
- **Servidor**.
    - **Rol**: Dentro de la aplicación, el servidor es quien se encarga de transmitir el video hacia el cliente.
    - **Nivel de conocimientos requeridos**: Al igual que el cliente, el servidor requiere de un conocimiento técnico medio para la configuración en los atributos de la transmisión y la configuración asociada al programa VLC (Servidor).
    - **Nivel de privilegios**: Nivel Alto, ya que gestiona la configuración de las aplicaciones y además es quien selecciona el video requerido para la transmisión.
- **Intermediario**.
    - **Rol**: Dentro de la aplicación, se encarga de retransmitir los datos que llegan desde el programa servidor hacia el programa cliente.
    - **Nivel de conocimientos requeridos**: No aplica, pues se asume que este programa siempre está en ejecución y no requiere configuración previa.
    - **Nivel de privilegios**: Nivel bajo, pues solo retransmiten los datos, sin alterarlos.

12.2.2 Casos de Uso y descripción

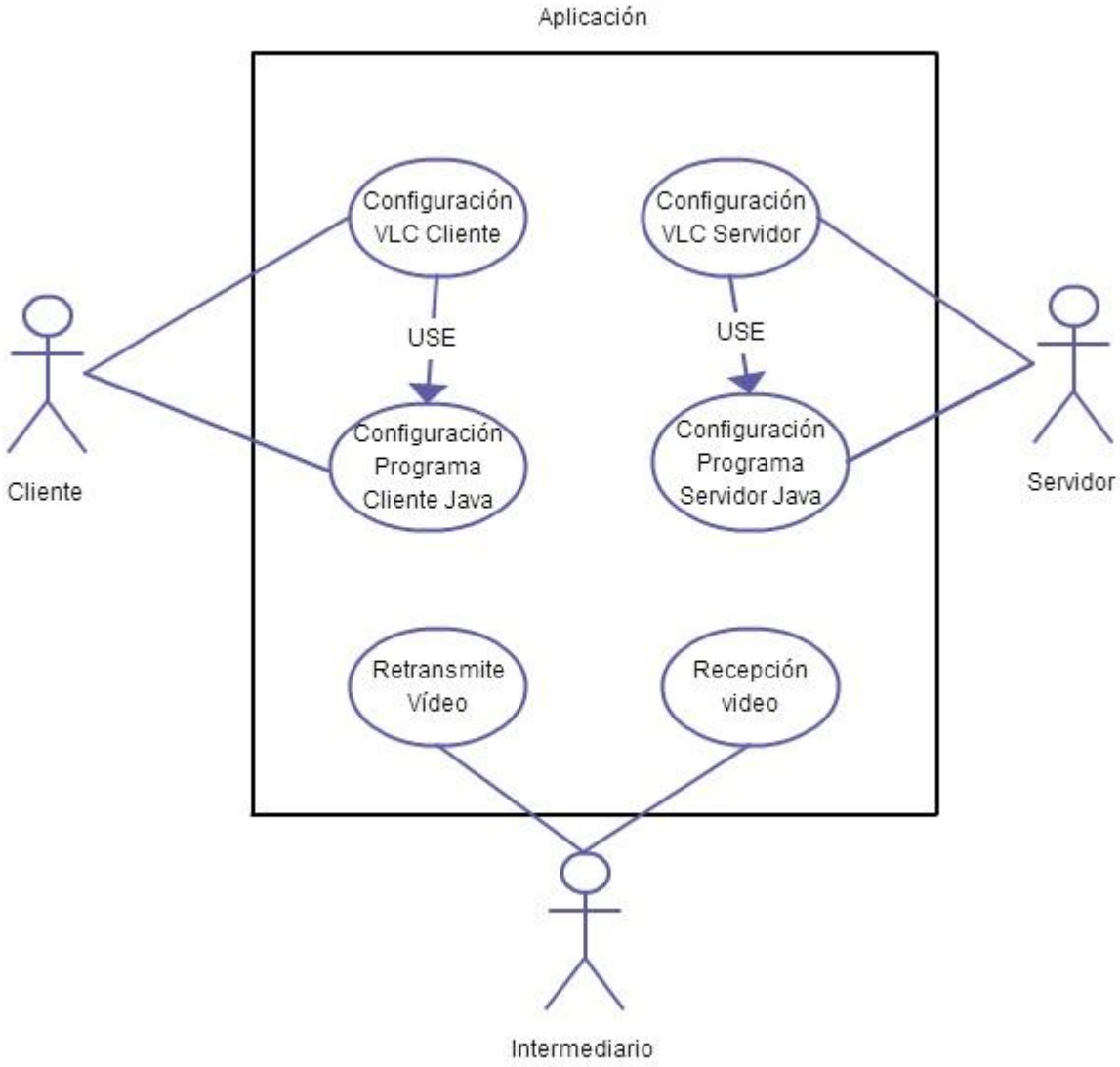

Figura 25 – Diagrama Casos de Uso.

12.2.3 Especificación de los Casos de Uso

12.2.3.1 Caso de Uso: <Configuración VLC Cliente>

- **Descripción**: El cliente configura el programa VLC con el puerto y protocolo de comunicación requeridos.
- **Pre-Condiciones**:
  - El Servidor, debe haber transmitido por algún medio, el puerto y protocolo de comunicación necesario para recibir la transmisión de video.
  - El Servidor debe estar "atento" al momento de transmitir los datos al término de la configuración de este ítem.
  - El programa intermediario debe estar en ejecución.
- **Flujo de Eventos Básicos.**

Al actor	El Programa
1 Ejecuta la aplicación VLC.	2 Se ejecuta y presenta las opciones en su barra de menú.
3 Selecciona del menú de VLC, la opción Medio, y luego la pestaña Red.	4 Abre una ventana en donde se debe establecer la ruta de origen del video transmitido.
5 Ingresa la dirección de acuerdo al protocolo y puerto determinados, y presiona el botón ejecutar.	6 Después de un par de segundos, comienza la visualización del video.

Tabla 18 – Flujo eventos básicos, configuración VLC cliente.

- **Flujo de Eventos Alternativo**.

Al actor	El Programa
5 Ingresa la dirección de acuerdo al protocolo y puerto determinados, y presiona el botón ejecutar.	6 Después de unos segundos, el programa notifica que no puede asociarse con la dirección establecida.
7 Se verifica que los datos de transmisión sean los correctos. En caso de error previo, se vuelven a ingresar y luego se presiona el botón ejecutar.	8 Después de un par de segundos, comienza la visualización del video.

Tabla 19 – Flujo eventos alternativos, Configuración VLC Cliente.

- Post-Condiciones: Luego de la total transmisión del video, el programa VLC vuelve a su estado inicial.

12.2.3.2 Caso de Uso: <Configuración programa Cliente Java>

- **Descripción**: El cliente configura el programa Cliente Java con el protocolo de comunicación, el tipo de cifrado y el algoritmo criptográfico requerido.
- **Pre-Condiciones**:
  - El Servidor, debe haber transmitido por algún medio, el protocolo de comunicación, el algoritmo criptográfico y el tipo de cifrado.
  - El Servidor, también debe estar en espera y preparado para la transmisión del video cuando la configuración del cliente se realice.
  - El programa intermediario debe estar en ejecución.
- **Flujo de Eventos Básicos**:

Al actor	El Programa
1 Ejecuta la aplicación Java Cliente.	2 Se ejecuta y presenta las opciones en la interfaz gráfica.
3 Selecciona la configuración requerida, interactuando con las listas desplegables que se presentan.	4 Queda en espera de la configuración.
5 Una vez seleccionada la configuración, el cliente "carga" esta configuración.	6 Muestra en una sección de la interfaz la configuración seleccionada.
7 El cliente ahora presiona el botón que da inicio a la ejecución del programa.	8 Comienza la ejecución del programa de acuerdo a la configuración, y se reciben los datos asociados a la transmisión.

Tabla 20 – Flujo eventos básicos, configuración programa java Cliente.

Flujo de Eventos Alternativo:

Al actor	El Programa
5 El cliente ahora presiona el botón que da inicio a la ejecución del programa.	6 Comienza la ejecución del programa, si no hay respuestas de comunicación se cancela la ejecución y se notifica al cliente.
7 Se verifica que los datos de transmisión sean los correctos. En caso de error previo, se vuelven a ingresar y luego se presiona el botón ejecutar.	8 Comienza la ejecución del programa de acuerdo a la configuración, y se reciben los datos asociados a la transmisión.

Tabla 21 – Flujo eventos alternativos, Configuración programa Java Cliente.

- Post-Condiciones: Luego de la total transmisión del video, la aplicación queda en espera de alguna nueva transmisión, siempre y cuando el Servidor esté listo para la transmisión.

12.2.3.3 Caso de Uso: <Configuración VLC Servidor>

- **Descripción**: El Servidor configura el programa VLC con el puerto y protocolo de comunicación requeridos.
- **Pre-Condiciones**:
  - El Cliente requiere la transmisión de un vídeo.
  - El programa Intermediario debe estar en ejecución para recibir los datos de emisión.
  - El programa intermediario debe estar en ejecución.
- **Flujo de Eventos Básicos**:

Al actor	El Programa
1 Ejecuta la aplicación VLC.	2 Se ejecuta y presenta las opciones en su barra de menú.
3 Selecciona del menú de VLC, la opción Emitir.	4 Abre una ventana en donde se debe seleccionar el video a transmitir.

5 Se da al botón "añadir", para cargar el video que se quiere emitir.	6 Carga el video, y vuelve a la ventana anterior.
7 Se presiona el botón "Emitir", y luego el botón "Siguiente".	8 Presenta una nueva ventana con una lista despegable, en donde se debe seleccionar el protocolo de comunicación que se empleará.
9 Se selecciona el protocolo de comunicación (UDP o TCP), y se presiona el botón añadir.	10 Se presenta una nueva ventana en donde se debe ingresar el puerto y la dirección hacia donde se transmitirá el video.
11 Se ingresa el puerto y la dirección requerida, y finalmente se presiona el botón "Emitir"	12 El programa comienza la transmisión de los datos.

Tabla 22 – Flujo eventos básicos configuración VLC Servidor.

Flujo de Eventos Alternativo:

Al actor	El Programa
9 Se selecciona el protocolo de comunicación (UDP o TCP), y se presiona el botón añadir.	10 Se presenta una nueva ventana en donde se debe ingresar el puerto y la dirección hacia donde se transmitirá el video.
11 Se ingresa el puerto y la dirección requerida, y finalmente se presiona el botón "Emitir"	12 Si el programa no comienza a emitir los datos, se cancela la emisión y se notifica que no se ha podido establecer la comunicación.
13 Se verifica que la configuración sea correcta. De haber errores, se ingresan los parámetros correctos, y se presiona el botón emitir.	14 El programa comienza la transmisión de los datos.

Tabla 23 – Flujo eventos alternativo configuración VLC Servidor.

- Post-Condiciones: Luego de la total transmisión del video, el programa VLC vuelve a su estado inicial.

12.2.3.4 Caso de Uso: <Configuración programa Servidor Java>

- **Descripción**: El actor configura el programa Servidor Java con el protocolo de comunicación, el tipo de cifrado y el algoritmo criptográfico requerido.
- **Pre-Condiciones**:
  - El cliente debe haber solicitado la transmisión de video.
  - El programa intermediario debe estar en ejecución.
  - El programa VLC Servidor debe estar configurado y preparado para emitir el video.
- **Flujo de Eventos Básicos**:

Al actor	El Programa
1 Ejecuta la aplicación Java Servidor.	2 Se ejecuta y presenta las opciones en la interfaz gráfica.
3 Selecciona la configuración requerida, interactuando con las listas desplegables que se presentan.	4 Queda en espera de la configuración.
5 Una vez seleccionada la configuración, el actor "carga" esta configuración.	6 Muestra en una sección de la interfaz la configuración seleccionada.
7 El actor ahora presiona el botón que da inicio a la ejecución del programa.	8 Comienza la ejecución del programa de acuerdo a la configuración, y se transmiten los datos asociados a la transmisión.

Tabla 24 – Flujo eventos básicos configuración programa Java Servidor.

- **Flujo de Eventos Alternativo**:

Al actor	El Programa
5 El actor ahora presiona el botón que da inicio a la ejecución del programa.	6 Comienza la ejecución del programa, si no hay respuestas de comunicación se cancela la ejecución y se notifica al cliente.
7 Se verifica que los datos de transmisión sean los correctos. En caso de error previo, se vuelven a ingresar y luego se presiona el botón ejecutar.	8 Comienza la ejecución del programa de acuerdo a la configuración, y se transmiten los datos que provengan desde el programa VLC Servidor.

Tabla 25 – Flujo eventos alternativos, configuración programa Java Servidor.

- Post-Condiciones: Luego de la total transmisión del video, la aplicación queda en espera de alguna nueva transmisión, siempre y cuando el programa VLC Servidor esté listo para la emisión y el cliente requiera de un video.

12.2.3.5 Caso de Uso: <Recepción Vídeo>

- **Descripción**: El programa intermediario recibe los datos cifrados provenientes desde el programa Servidor Java.
- **Pre-Condiciones**:
  - El programa VLC Servidor debe estar en ejecución y transmitiendo un video.
  - El programa Java Servidor debe estar en ejecución, recibiendo los datos originales desde VLC y retransmitiendo los datos cifrados.
- **Flujo de Eventos Básicos**:

Al actor	El Programa
1 Se da inicio al programa.	2 Se ejecuta y recibe los datos cifrados desde el programa Servidor.

Tabla 26 – Flujo eventos básicos, recepción de video.

- **Flujo de Eventos Alternativo**:

Al actor	El Programa
1 Se da inicio al programa.	2 Se ejecuta si no hay recepción de datos queda en espera.

Tabla 27 – Flujo eventos alternativos, recepción de video.

- Post-Condiciones: Luego de la total transmisión del video, la aplicación queda en espera de alguna nueva transmisión, siempre y cuando el programa VLC Servidor esté listo para la emisión.

12.2.3.6 Caso de Uso: <Retransmisión Video>

- **Descripción**: El programa intermediario vuelve a transmitir los datos cifrados provenientes desde el programa Servidor Java.
- **Pre-Condiciones**:
  - El programa VLC Servidor debe estar en ejecución y transmitiendo un video.
  - El programa Java Servidor debe estar en ejecución, recibiendo los datos originales desde VLC y retransmitiendo los datos cifrados.
  - El programa Cliente y el programa VLC cliente deben estar en ejecución, y en espera de datos.
- **Flujo de Eventos Básicos**:

Al actor	El Programa
1 Se da inicio al programa.	2 Se ejecuta y recibe los datos cifrados los que luego se retransmiten hacia el cliente.

Tabla 28 – Flujo eventos básicos, retransmisión de video.

- **Flujo de Eventos Alternativo**:

Al actor	El Programa
1 Se da inicio al programa.	2 Se ejecuta, si no hay recepción de datos queda en espera.

Tabla 29 – Flujo eventos alternativos, retransmisión de video.

- Post-Condiciones: Luego de la total transmisión del video, la aplicación queda en espera de alguna nueva transmisión, siempre y cuando el programa VLC Servidor esté listo para la emisión y los programas del cliente (Java y VLC), estén en espera de nuevos datos.

12.3 Modelamiento de datos.

Como se ha explicado en los puntos anteriores a este documento, el trabajo realizado está orientado a un tema de investigación, sin embargo, y a modo de proposición, se presenta el siguiente diagrama ER como base para lo que podría ser un futuro trabajo respecto al tema planteado en este informe.

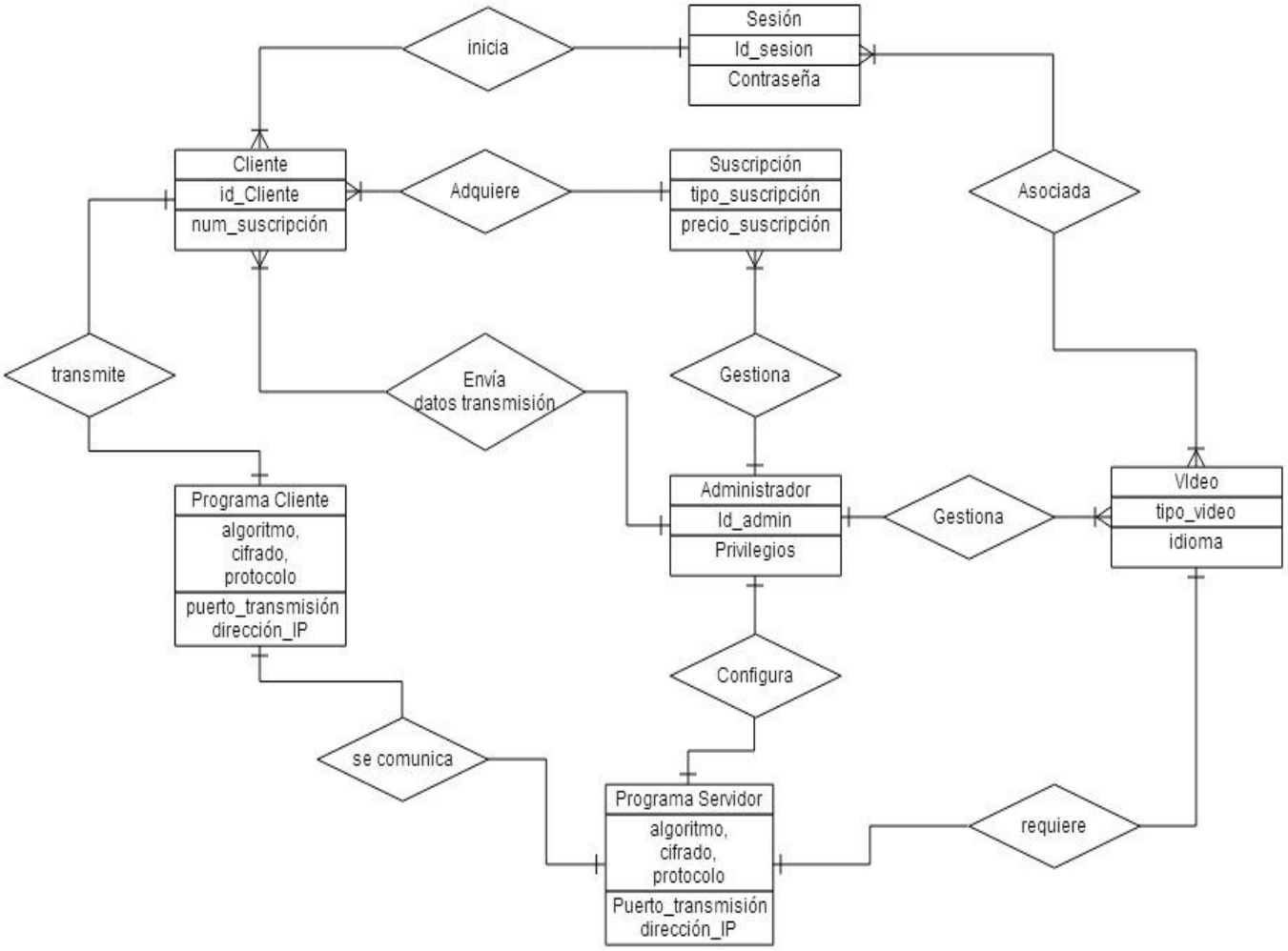

Figura 26 – Propuesta modelo ER.

13 ANEXO: DISEÑO

13.1 Diseño de Físico de la Base de datos.

Si bien se ha dado una propuesta, a modo de supuesto, de lo que es el modelo de ER, no se desarrolla un diseño físico de base de datos, pues el tema va más allá de los objetivos y requerimientos planteados en un principio para este trabajo de investigación.

13.2 Diseño de arquitectura funcional

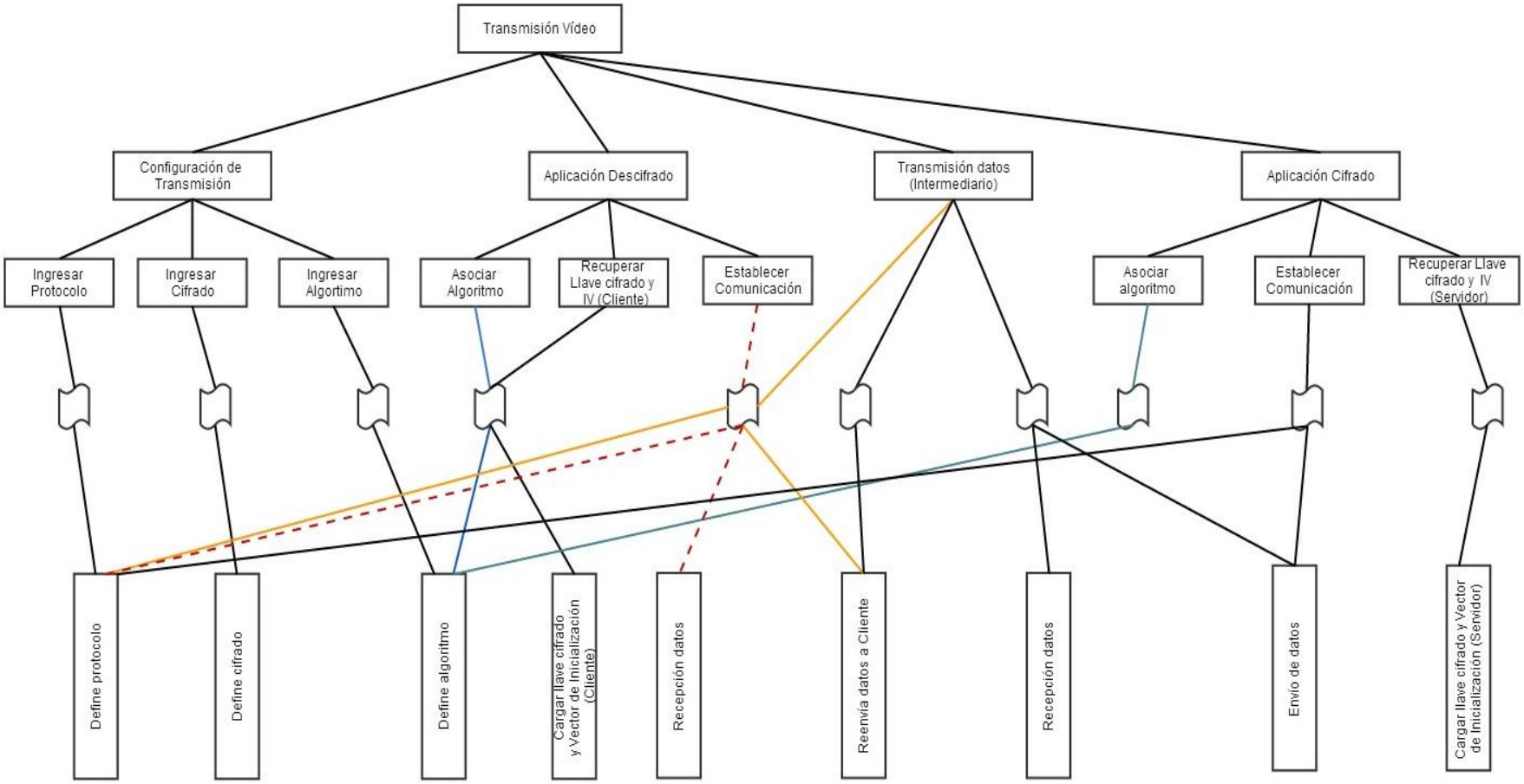

Figura 27 – Árbol de descomposición funcional

13.3 Diseño interfaz y navegación.

En un principio la interfaz puede ser como la siguiente imagen.

Figura 28 – Prototipo Interfaz Gráfica.

La descripción de los ítems es la siguiente:

1 - Label, establece el título para la selección del protocolo.

2 - Label, establece el título para el algoritmo criptográfico.

3 - Label, establece el título para el modo de cifrado

4 - ComboBox, selección del tipo de protocolo (UDP, TCP).

5 - ComboBox, selección del tipo de algoritmo (DES, Tiple DES, AES)..

6 - ComboBox, selección del modo de cifrado (ECB, CBC, CTR).

7 - Button, una vez seleccionado los parámetros, se presiona este botón para establecer los cambios en el código.

8 - Label, al presionar el Button nº 7, se visualiza la combinación de parámetros seleccionados.

9 - Label, de acuerdo al algoritmo seleccionado se visualiza el tamaño de la llave requerida para cifrar/descifrar los datos (este tema se ve más adelante en el informe).

10 - Button, una vez establecido los parámetros, este botón ejecuta la aplicación Servidor o Cliente según corresponda el programa.

13.4 Especificación de módulos

Los módulos de programa creados para esta aplicación se describen como sigue:

N° Módulo: 001		**Nombre Módulo**: Define Protocolo	
Parámetros de entrada: Tipo de protocolo de comunicación.		**Parámetros de Salida: Combinación de configuración seleccionada.**	
Nombre:	**Tipo de dato:**	**Nombre:**	**Tipo de dato:**
UDP o TCP	String	Protocolo/algoritmo/cifrado	String

N° Módulo: 002		**Nombre Módulo**: Define Cifrado	
Parámetros de entrada: Tipo de cifrado de datos.		**Parámetros de Salida: Combinación de configuración seleccionada.**	
Nombre:	**Tipo de dato:**	**Nombre:**	**Tipo de dato:**
ECB , CBC o CTR	String	Protocolo/algoritmo/cifrado	String

N° Módulo: 003		**Nombre Módulo**: Define algoritmo	
Parámetros de entrada: Tipo de cifrado de datos.		**Parámetros de Salida: Combinación de configuración seleccionada.**	
Nombre:	**Tipo de dato:**	**Nombre:**	**Tipo de dato:**
AES , DES , 3DES	String	Protocolo/algoritmo/cifrado	String
		Tamaño de llave decifrado	Int

N° Módulo: 004		**Nombre Módulo**: Cargar Llave de Cifrado y Vector de Inicialización (Cliente)	
Parámetros de entrada: Llave de cifrado y IV.		**Parámetros de Salida: Combinación de configuración seleccionada.**	
Nombre:	**Tipo de dato:**	**Nombre:**	**Tipo de dato:**
Llave cifrado	*.key		
Vector Inicialización	*.IV		

N° Módulo: 005		**Nombre Módulo**: Recepción Datos	
Parámetros de entrada: Datos Cifrados.		**Parámetros de Salida: Datos Descifrados.**	
Nombre:	**Tipo de dato:**	**Nombre:**	**Tipo de dato:**
ciphertext	byte[]	Buffer_descifrado	byte[]

N° Módulo: 006		**Nombre Módulo**: Reenvía datos a cliente	
Parámetros de entrada: Datos Cifrados.		**Parámetros de Salida: Datos Cifrados**	
Nombre:	**Tipo de dato:**	**Nombre:**	**Tipo de dato:**
buffer	byte[]	buffer	byte[]

N° Módulo: 007		**Nombre Módulo**: Recepción datos	
Parámetros de entrada: Datos Cifrados.		**Parámetros de Salida: Datos Cifrados**	
Nombre:	**Tipo de dato:**	**Nombre:**	**Tipo de dato:**
buffer	byte[]	buffer	byte[]

N° Módulo: 008	**Nombre Módulo**: Envío datos
Parámetros de entrada: Datos originales (desdeVLC)	**Parámetros de Salida: Datos Cifrados**

Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
buffer	byte[]	Buffer_cifrado	byte[]

N° Módulo: 009		Nombre Módulo: Cargar Llave de Cifrado y Vector de Inicialización (Servidor)	
Parámetros de entrada: Llave de cifrado y IV.		Parámetros de Salida: Combinación de configuración seleccionada.	
Nombre:	**Tipo de dato:**	**Nombre:**	**Tipo de dato:**
Llave cifrado	*.key		
Vector Inicialización	*.IV		

Tabla 30 – Especificación de módulos.

Para aquellas funcionalidades específicas de la librería empleada (JCE), se tienen la siguiente descripción:

- Nombre: Cipher.
- Objetivo: Realizar una instancia para poder encriptar o descencriptar un dato.
- Parámetros: Se requiere determinar, el algoritmo, el modo de cifrado y el padding requerido.
- Ejemplo:
    - Cipher cipher = Cipher.getInstance("TripleDes/CBC/PKCS5Padding")
    - cipher.init(Cipher.ENCRYPT_MODE, key,ivectorSpecv) // Encriptado
    - cipher.init(Cipher.DECRYPT_MODE, key,ivectorSpecv) // Descencriptado

14 ANEXO: PLANIFICACION INICIAL DEL PROYECTO

- A continuación se muestra la planificación de todo el proceso de investigación y trabajo asociado a este proyecto. La carta Gantt elaborada, toma en consideración la expansión de plazo inicial de entrega, cuyo cambio fue, desde el 27 de Diciembre del 2013, hasta el 28 de Febrero del 2014. Para mejor comprensión y entendimiento de la carta Gantt, esta será presentada en dos partes, la primera contiene la descripción de las tareas realizadas junto con sus respectivas fechas, mientras que la otra parte, expone el diagrama como tal.
- Fechas de planificación.

Nombre	Fecha de inicio	Fecha de fin
Entrega propuesta inicial	26/08/13	26/08/13
Respuesta propuesta inicial	30/08/13	30/08/13
Recopilación información gral.	2/09/13	28/09/13
Introducción criptografía	2/09/13	8/09/13
P2P, seguridad y streaming	9/09/13	16/09/13
Trabajo básico con VLC	20/09/13	28/09/13
Desarrollo de aplicación.	1/10/13	10/01/14
Aplicación prototipo para recepción de datos.	1/10/13	12/10/13
Iteración - Modificación de prototipo para protocolos de comunicación	14/10/13	24/10/13
Presentación de transmisión de datos por protocolo de comunicaci...	14/10/13	24/10/13
Estudio de librería JCE	25/10/13	28/10/13
Iteración - Agregar funcionalidad de criptografía simetrica	30/10/13	14/11/13
Modificación para programa con protocolo UDP.	30/10/13	8/11/13
Modificación para programa con protocolo TCP.	9/11/13	14/11/13
Desarrollo de primeras pruebas de funcionalidad	15/11/13	16/11/13
Iteración - Agregar funcionalidad de criptografía asimetrica	18/11/13	20/11/13
Modificación para programa con protocolo UDP	18/11/13	19/11/13
Modificación para programa con protocolo TCP	19/11/13	20/11/13
Desarrollo de segundas pruebas de funcionalidad.	22/11/13	27/11/13
Capacitación a profesor Guía	27/11/13	27/11/13
Configuración de equipos en laboratorio	29/11/13	10/01/14
Configuración para programas de protocolo UDP	29/11/13	4/12/13
Etapa de pruebas para obtención de resultados	5/12/13	13/12/13
Configuración para programas de protocolo TCP	16/12/13	20/12/13
Etapa de pruebas para obtención de resultados.	3/01/14	10/01/14
Validación y finalización de pruebas	10/01/14	10/01/14
Desarrollo informe	27/11/13	22/02/14
Creación.	27/11/13	21/02/14
Revisión/modificación	9/12/13	22/02/14
Entrega Informe final	28/02/14	28/02/14

Figura 29 – Fechas de planificación, Carta Gantt.

- Distribución de actividades en el tiempo.

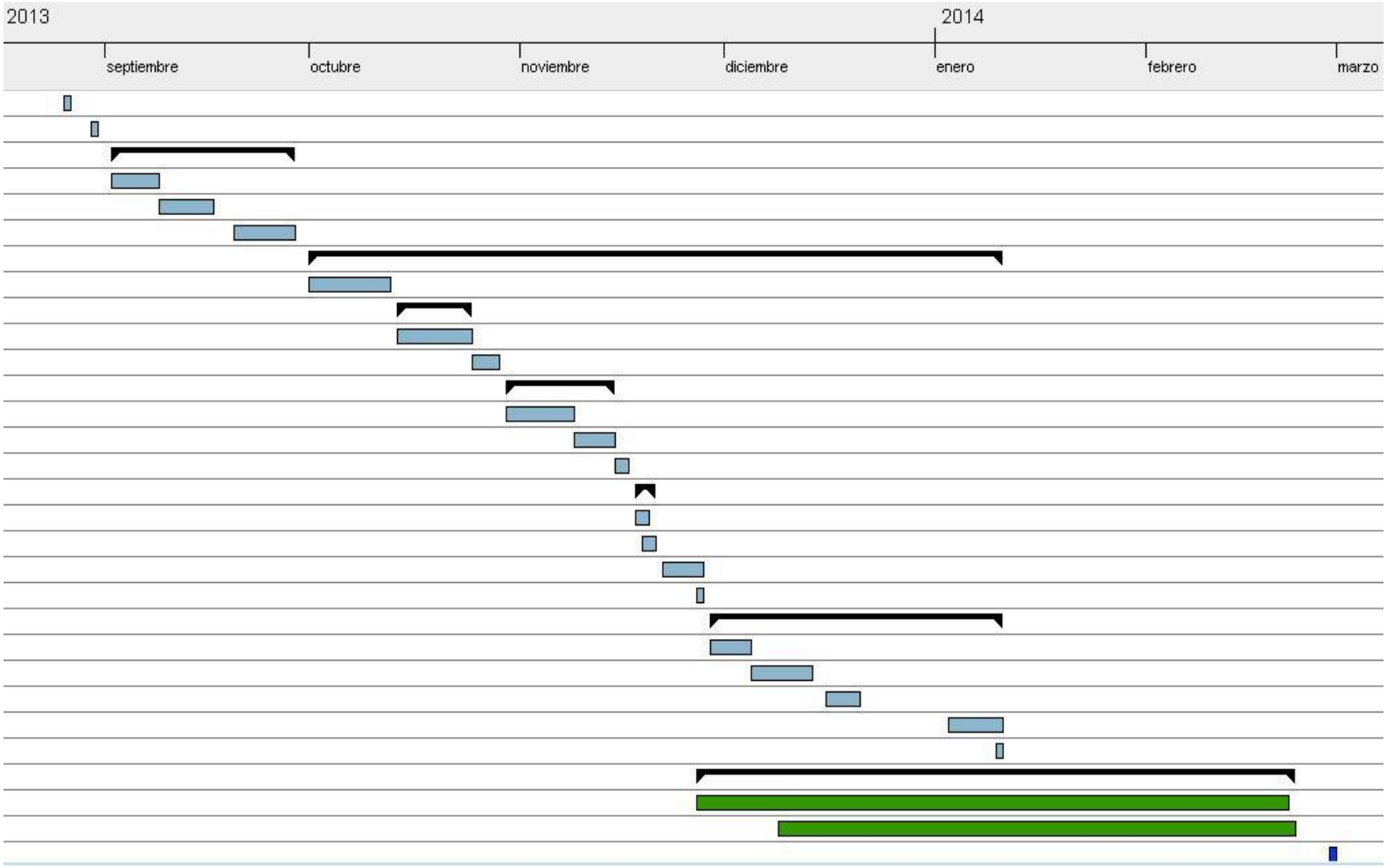

Figura 30 – Distribución de tereas en el tiempo, Carta Gant

14.1.1 Estimación inicial de tamaño

- **Puntos de casos de uso.**

Considerando las siguientes tablas de datos

Tipo de caso de uso			
5	Simple	Menos de 5 clases 5	3 transacciones o menos
10	Medio	5 a 10 clases 10	4 a 7 transacciones
15	Complejo	Más de 10 clases 18	Más de 7 transacciones

Tipo de actor D		
1	Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API).
2	Medio	Otro sistema interactuando a través de un protocolo (ej. TCP/IP) o una persona interactuando a través de una interfaz en modo texto
3	Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI).

Tabla 31 – Datos para cálculo punto casos de uso.

Se realiza la clasificación que se muestra a continuación.

- Clasificación de actores.

Tipo de Actor	Factor	Actor	Total
Simple	1	0	0
Medio	2	3	6
Complejo	3	0	0

Tabla 32– Clasificación de actores.

Total UAW	6

- Clasificación de casos de uso.

Tipo Caso de Uso	Factor	Caso de Uso	Total
Simple	5	6	30
Medio	10	0	0
Complejo	15	0	0

- **Tabla 33– Clasificación de casos de uso.**

Total UUCW	30

Con los resultados anteriores, se obtiene el valor asociado a los puntos de casos de uso sin ajustar (Unadjusted Use Case Point – UUCP).

UUCP = UUCW + UAW	36

Ahora se requiere calcular los valores asociados al Factor de complejidad técnica, y Factor de complejidad del entorno.

- **Factor de complejidad técnica.**

  Calculate TCF (Technical Complexity Factor)

Technical Factor	Multiplier
Distributed System	2
Application performance objectives, in either response or throughput	1
End-user efficiency (on-line)	1
Complex internal processing	1
Reusability, the code must be able to reuse in other applications	1
Installation ease	0,5
Operational ease, usability	0,5
Portability	2
Changeability	1
Concurrency	1
Special security features	1
Provide direct access for third parties	1
Special user training facilities	1

Tabla 34– Datos para el cálculo de complejidad técnica.

Descripción	Factor	Impacto percibido	Total
Sistema Distribuido	2	0	0
Rendimiento o tiempo de respuesta	1	4	4
Eficiencia del usuario final (on-line)	1	0	0
Procesamiento interno complejo	1	1	1
Código reutilizable	1	3	3
Facilidad de instalación	0,5	3	1,5
Facilidad de uso	0,5	4	2
Portabilidad	2	5	10
Facilidad de cambio	1	3	3
Concurrencia	1	2	2
Características especiales de seguridad	1	0	0
Provee acceso directo a 3ª personas	1	1	1
Facilidades especiales de entrenamiento a usuario	1	1	1

Tabla 35– Cálculo de complejidad técnica.

Total Factor Técnico	28,5

Factor de complejidad técnica (TCF)	
0,6+(0,01*Total factor Técnico)	0,885

- Factor de complejidad de entorno ambiental.

Environmental Factor	Multiplier
Familiar with Objectory + RUP	1,5
Application experience	0,5
Object Oriented experience	1
Analyst capability	0,5
Motivation	1
Stable requirements	2
Par time workers	-1
Difficult programming language	-1

Tabla 36– Datos para el cálculo de complejidad ambiental.

Descripción	Factor	Impacto percibido	Total
Familiaridad con modelo de proyecto usado	1,5	2	3
Experiencia en la aplicación	0,5	3	1,5
Experiencia en orientación a objetos	1	2	2
Capacidad de análisis	0,5	4	2
Motivación	1	5	5
Estabilidad de los requerimientos	2	1	2
Personal Part Time	-1	0	0
Dificultad en el lenguaje de programación	-1	2	-2

Tabla 37– Cálculo de complejidad ambiental.

Total factor entorno	13,5

Factor de complejidad entorno (EF)	
1,4+(-0,03*Total factor ambiental)	0,995

- Por definición se dice lo siguiente, si la suma entre el número de los primeros seis factores de entornos, cuyo valor es mayor a 3, y el número de los últimos factores de entorno es menor a 3.
  - Si es < 2 => LOE = 20.
  - Si esta entre 3 y 4 => LOE = 28.
  - Si es 4 >, hay que reconsiderar el proyecto. Se puede reducir los riesgos asociados con los factores de entorno.
- En este caso, se tiene que dentro de los primeros 6 factores de entorno, la motivación tiene un valor de 5 (Mayor a 3), y de los últimos dos factores, la dificultad en el lenguaje de programación tiene un valor de -2 (menor a 3). La suma de lo anterior es de un valor de 3. Por lo tanto se obtiene un valor de LOE igual a 28.

Para el cálculo de la complejidad técnica y la complejidad del entorno, los valores asociados al impacto percibido, deben estar en el siguiente rango.

Descripción	Valor
Irrelevante De	0 a 2.
Medio De	3 a 4.
Esencial	5

- Estimación de horas hombre.

  Según [13], el autor de la técnica, indica que se recomienda utilizar un valor de 20 horas por cada UUCP.

  Como se dio a conocer anteriormente, el valor de UUCP corresponde a 36. Por lo tanto, el valor estimado de horas hombre corresponde a:

Valor Horas hombre	720 hrs/hombre

  Considerando que se trabajó en promedio durante 5 días de la semana, por 7 horas, se tiene un total de 20,57 semanas (21 semanas aproximadamente), lo que equivale a unos 5 meses de trabajo para una persona.

14.1.2 Contabilización final del tamaño del Sw

- Contabilizar la cantidad de líneas de código implementadas en su software. Especifique junto a los valores la forma como fue calculado el valor (si se consideran líneas en blanco, comentarios o todas las líneas, si se consideran funciones o componentes reutilizados)

	Total líneas de código por programa
Cliente - UDP	123 líneas de código.
Intermediario – UDP	59 líneas de código
Servidor – UDP	88 líneas de código.
Cliente – TCP	168 líneas de código.
Intermediario - TCP	63 líneas de código.
Servidor - TCP	103 líneas de código.
Interfaz Gráfica	246 líneas de código.
Total líneas de código desarrolladas.	850 líneas de código. (aprox)

Tabla 38– Cálculo cantidad líneas de código desarrolladas.

- Equivalencias PCU con líneas de código.

  De acuerdo al resultado obtenido en el PCU, se puede afirmar que el tiempo estimado es correcto y adecuado para el trabajo desarrollado, considerando que hubo también un proceso de investigación y aprendizaje, el cual fue necesario para la creación de los distintos programas Java.

- Estimación de trabajo horas hombre.

  Como se mencionó en el punto anterior, en promedio el esfuerzo por horas hombre equivale aproximadamente a 720 horas/hombre. En este caso como el proyecto fue desarrollado por una sola persona, en total se tiene un tiempo de desarrollo aproximado a 5 meses.

15 ANEXO: PLAN DE CAPACITACIÓN Y ENTRENAMIENTO

- **Usuario a capacitar**: Profesor Guía, Patricio Galdámes Sepúlveda.
- **Tipo de capacitación o entrenamiento**: Configuración de programas Cliente y Servidor.
- **Funcionalidad o aspectos que serán abordados**:
  - La funcionalidad que se considera en la capacitación, corresponde a la configuración requerida por los programas tanto Cliente y Java, para la correcta transmisión de los datos.
- **Responsable**: Heber Gálvez Ojeda.
- **Tiempo estimado**: 5 horas aprox.
- **Calendario**: Como la capacitación no fue compleja, y sólo fue realizada a una persona, esta se desarrolló durante el período de una tarde. El día de desarrollo puede verse en la carta Gantt, adjuntada en el anexo nº 17.
- **Recursos requeridos /utilizados**
  - **Recursos de Hardware:** Para la capacitación, se emplearon los computadores que se encuentran en los laboratorios de la Facultad De Ciencias Empresariales (FACE).
  - **Recursos Software:** Se utilizó como sistema operativo Microsoft Windows XP, además de las herramientas asociadas a Java, es decir, Java Virtual Machine.

16 ANEXO: PLAN DE IMPLANTACIÓN Y PUESTA EN MARCHA

Al igual que en puntos anteriores, en este caso, este capítulo no aplica, pues el programa desarrollado está orientado al apoyo asociado a la investigación, y no como un sistema asociado al apoyo de negocios de una empresa.

17 ANEXO: RESUMEN ESFUERZO REQUERIDO

Actividades/fases	N° Horas
Investigación inicial del tema.	20 horas.
Estudio de criptografía y recursos necesarios.	15 horas.
Desarrollo de primeras aplicaciones bases.	40 horas.
Integración de nuevos requerimientos de desarrollo.	15 horas.
Desarrollo de programas (Protocolo UDP)	80 horas.
Actividades de prueba para programas de protocolo UDP.	45 horas.
Desarrollo de programa (Protocolo TCP)	90 Horas.
Actividades de prueba para programas de protocolo TCP	100 horas.
Etapa de pruebas finales para UDP.	50 horas.
Etapa de pruebas finales para TCP.	90 horas
TOTAL	545 horas aprox.

Tabla 39 – Resumen esfuerzo requerido.

De acuerdo a los datos obtenidos en el punto 17.1.1, en general se puede hablar de un alto grado de coherencia existente entre el tiempo aproximado obtenido con el uso de puntos de casos de uso, y el resumen de actividades recién planteado.

Considerando además, que existieron etapas y un trabajo previo, es decir, la etapa de desarrollo del informe previo a la aceptación del tema, se concluye que los plazos de desarrollo propuestos se han cumplido de manera correcta. Además, es importante recordar que se está considerando la ampliación en la entrega de este documento, por lo cual se queda en conformidad con los tiempos establecidos en la planificación desarrollada.

18 ANEXO: RESULTADOS DE ITERACIONES EN EL DESARROLLO

De acuerdo a lo planteado en la planificación de este proyecto, los resultados asociados a la metodología empleada son los siguientes.

- **Nombre Iteración**: Presentación de transmisión de datos por protocolo de comunicación.
- **Descripción**: En esta etapa se comienza a indagar en los aspectos asociados y requeridos para poder realizar la transmisión y recepción de los datos. Para llevar a cabo lo anterior se desarrollaron los primeros programas que tenían como finalidad determinar si efectivamente se recibían los datos provenientes desde el programa VLC Servidor, para luego reenviarlos, sin modificación, hacia la visualización en el programa VLC Cliente.

- **Nombre Iteración**: Desarrollo de primeras pruebas de funcionalidad.
- **Descripción**: Luego de haber verificado que los datos transmitidos eran recibidos sin problemas, y sin alteraciones, se comenzó la modificación de los programas incluyendo las funcionalidades de la librería JCE, con el objetivo de dar comienzo a la encriptación de los datos, empleando en primera instancia criptografía simétrica. Las pruebas tenían como objetivo el cifrar los datos que provenían desde la aplicación VLC Servidor. Una de las primeras pruebas fue la retransmisión de los datos cifrados, con lo cual la aplicación VLC Cliente no podía reproducir, pues los datos estaban alterados. Por lo tanto al aplicar la desencriptación de los datos, se pudo recuperar la información originalmente transmitida, y por ende, se daba por completada la etapa de prueba.

- **Nombre Iteración**: Desarrollo de segundas pruebas de funcionalidad.
- **Descripción**: Una vez completada las pruebas de transmisión de datos con criptografía simétrica, se pasó a la etapa de asegurar la información transmitida con algoritmos asimétricos, de forma específica, con RSA. Si bien el uso de este mecanismo no altera los datos asociados al video, fue requerido su uso para asegurar la transmisión.

- **Nombre Iteración**: Etapa de pruebas para obtención de resultados (UDP).
- **Descripción**: Una vez realizada la configuración de los equipos en el laboratorio, se llega al desarrollo de las pruebas que tienen como objetivo la obtención de resultados. En este caso las pruebas son para el protocolo de comunicación UDP.

- **Nombre Iteración**: Etapa de pruebas para obtención de resultados (TCP).
- **Descripción**: Al igual que el punto anterior, una vez realizada la configuración de los equipos, se comienza la obtención de los resultados para la transmisión de datos.