Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

Algoritmos y Estructuras de Datos II (CE 2103)

Documento de propuesta

Tercer proyecto: Criptografía

Profesor:

Jose Isaac Ramírez Herrera

Estudiantes:

Jestim Felix Espinoza Morales

1. Descripción general del proyecto:

En la actualidad, la comunicación digital se ha convertido en una herramienta esencial en diversos ámbitos, desde el personal hasta el profesional. La capacidad de intercambiar información de forma rápida y eficiente es crucial para la colaboración, el aprendizaje y la productividad. En este contexto, los sistemas de chat desempeñan un papel fundamental al facilitar la comunicación en tiempo real entre usuarios.

Este proyecto se enfoca en el desarrollo de un sistema de chat cliente-servidor que permita a los usuarios no solo intercambiar mensajes de texto, sino también transferir archivos de diferentes tipos. El sistema se implementará utilizando el lenguaje de programación C++ y la biblioteca SFML, la cual proporciona un conjunto de herramientas para la gestión de redes, gráficos, audio y más.

La elección de C++ se basa en su eficiencia, rendimiento y capacidad para desarrollar aplicaciones de alto nivel. Por otro lado, SFML ofrece una API sencilla y multiplataforma, lo que facilita el desarrollo y la portabilidad del sistema.

El problema que se pretende abordar con este proyecto es la falta de una herramienta de comunicación versátil que combine la capacidad de chat en tiempo real con la transferencia eficiente de archivos. Si bien existen numerosas aplicaciones de mensajería disponibles, muchas de ellas se centran principalmente en el intercambio de mensajes de texto o presentan limitaciones en cuanto a la transferencia de archivos.

Este proyecto busca desarrollar una solución que integre ambas funcionalidades de manera robusta y eficiente, proporcionando a los usuarios una herramienta completa para la comunicación digital. Además, se sentarán las bases para una futura implementación de seguridad que garantice la confidencialidad e integridad de la información intercambiada.

2. Justificación:

La necesidad de una herramienta de comunicación que combine el chat en tiempo real con la transferencia de archivos es evidente en diversos contextos. En el ámbito educativo, por ejemplo, facilita la interacción entre estudiantes y profesores, permitiendo el intercambio de materiales de estudio, tareas y retroalimentación. En el ámbito empresarial, agiliza la comunicación interna y la colaboración en proyectos, optimizando el flujo de trabajo y la productividad.

El desarrollo de este sistema de chat tiene una pertinencia significativa, ya que se alinea con las demandas actuales de la sociedad de la información. La creciente necesidad de comunicación instantánea y el intercambio de archivos de diferentes formatos hacen que este proyecto sea relevante y oportuno.

El enfoque propuesto, basado en C++ y SFML, es adecuado para el desarrollo de un sistema robusto, eficiente y escalable. C++ ofrece un alto rendimiento y control sobre los recursos del sistema, mientras que SFML simplifica el desarrollo de aplicaciones multimedia y de red.

Los beneficiarios potenciales de este proyecto son diversos. Estudiantes, profesores, profesionales y cualquier persona que requiera una herramienta de comunicación versátil se beneficiará de las funcionalidades del sistema.

Los beneficios económicos se traducen en una mayor eficiencia en la comunicación, lo que puede impactar positivamente en la productividad y la reducción de costos. Desde el punto de vista tecnológico, el proyecto contribuye al desarrollo de soluciones de software innovadoras y al uso de tecnologías de vanguardia. A nivel social, facilita la comunicación y la colaboración entre individuos, promoviendo la interacción y el intercambio de conocimientos.

3. Marco Teórico/Estado del Arte

El desarrollo de este proyecto se basa en los siguientes conceptos y tecnologías:

Arquitectura Cliente-Servidor: Este modelo de arquitectura distribuida define la relación entre dos entidades: el cliente, que solicita un servicio, y el servidor, que lo proporciona. En un sistema de chat, el servidor actúa como intermediario, recibiendo mensajes de un cliente y enviándolos a otro. Esta arquitectura permite la escalabilidad, ya que múltiples clientes pueden conectarse al mismo servidor, y la centralización de recursos, como el almacenamiento de historiales de chat y la gestión de usuarios. (Tanenbaum & Van Steen, 2017)

Lenguaje de Programación C++: C++ es un lenguaje de programación de propósito general que combina características de programación procedural y orientada a objetos. Su eficiencia, rendimiento y capacidad para trabajar a bajo nivel lo convierten en una excelente opción para desarrollar aplicaciones de red que requieren un control preciso sobre los recursos del sistema, como la gestión de memoria y el manejo de conexiones. (Donnelly, 2019; Josuttis, 2020)

Biblioteca SFML: SFML (Simple and Fast Multimedia Library) es una biblioteca multiplataforma que proporciona una API simple y consistente para manejar gráficos, audio, redes y entrada (SFML Developers, n.d.). SFML facilita el desarrollo de aplicaciones multimedia interactivas, como juegos y sistemas de chat, al abstraer las complejidades de la programación a bajo nivel y ofrecer una interfaz fácil de usar. En este proyecto, SFML se utilizará para la gestión de la red, permitiendo la creación de sockets, la conexión entre el cliente y el servidor, y el envío y recepción de datos.

Protocolos de Red: Los protocolos de red son un conjunto de reglas y estándares que rigen la comunicación entre dispositivos en una red. TCP/IP (Transmission Control Protocol/Internet Protocol) es un conjunto de protocolos fundamentales para Internet. TCP proporciona una comunicación confiable y orientada a la conexión, asegurando que los datos se transmitan en orden y sin errores. IP se encarga del direccionamiento y enrutamiento de los paquetes de datos a través de la red. En este proyecto, TCP/IP se utilizará para establecer una conexión estable y confiable entre el cliente y el servidor, garantizando la correcta transmisión de mensajes y archivos. (Forouzan, 2017)

Para la futura implementación de la seguridad se considerarán:

Criptografía Simétrica: La criptografía simétrica utiliza la misma clave para cifrar y descifrar información. AES (Advanced Encryption Standard) es un algoritmo de cifrado simétrico ampliamente utilizado y considerado seguro. Su robustez y eficiencia lo hacen ideal para proteger la confidencialidad de los mensajes y archivos transmitidos en el sistema de chat. (Schneier, 2015)

Criptografía Asimétrica: La criptografía asimétrica utiliza un par de claves: una clave pública para cifrar y una clave privada para descifrar. RSA (Rivest-Shamir-Adleman) es uno de los algoritmos de criptografía asimétrica más utilizados. En un sistema de chat, RSA puede utilizarse para el intercambio seguro de claves simétricas, la autenticación de usuarios y la firma digital de mensajes, garantizando la integridad y autenticidad de la información. (Schneier, 2015)

Funciones Hash: Las funciones hash son algoritmos que toman una entrada de datos de cualquier tamaño y producen una salida de tamaño fijo (hash). SHA-256 (Secure Hash Algorithm 256-bit) es una función hash criptográfica que genera un hash único para cada mensaje. En un sistema de chat, SHA-256 puede utilizarse para verificar la integridad de los mensajes, asegurando que no han sido modificados durante la transmisión. (Stallings, 2022)

Protocolos de Comunicación Segura: TLS/SSL (Transport Layer Security/Secure Sockets Layer) son protocolos criptográficos que proporcionan una comunicación segura a través de una red. TLS/SSL se utiliza para establecer una conexión segura entre el cliente y el servidor, cifrando los datos transmitidos y autenticando las partes involucradas. En un sistema de chat, TLS/SSL puede utilizarse para proteger la confidencialidad e integridad de los mensajes y archivos, previniendo ataques de intermediarios y garantizando la seguridad de la comunicación. (Stallings, 2022)

Referencias:

- Donnelly, J. (2019). C++ Crash Course: A Fast-Paced Introduction.
- Josuttis, N. M. (2020). The C++ Standard Library: A Tutorial and Reference (2nd Edition). Addison-Wesley Professional.
- Schneier, B. (2015). Cryptography Engineering: Design Principles and Practical Applications. Wiley.
- Stallings, W. (2022). Cryptography and Network Security: Principles and Practice (8th Edition). Pearson Education.
- Forouzan, B. A. (2017). TCP/IP Protocol Suite (5th Edition). McGraw-Hill Education.
- Tanenbaum, A. S., & Van Steen, M. (2017). Distributed systems: principles and paradigms (3rd ed.). Pearson Education.
- SFML Developers. (n.d.). *Simple and Fast Multimedia Library*. https://www.sfml-dev.org/documentation/2.6.2/

6. Plan de Acción:

Objetivo Específico	**Producto**	**Actividades**	**Período**	**Asignacion**
Implementar comunicación cliente-servidor	Módulo de comunicación	Implementar la conexión TCP, el envío y recepción de mensajes.	Semana 1-2	Jestim
Implementar transferencia de archivos	Implementar la conexión TCP, el envío y recepción de mensajes.	Implementar la segmentación de archivos, el envío y recepción de bloques.	Semana 3-4	Jestim
Investigación de seguridad	Módulo de transferencia	Investigar e implementar algoritmos de criptografía como AES, RSA y SHA-256	Semana 5	Jestim

7. Definición del cronograma:

Actividad	**Fecha de inicio**	**Fecha de fin**
Implementar comunicación cliente-servidor	18 de octubre de 2024	1 de noviembre de 2024
Implementar transferencia de archivos	4 de noviembre de 2024	15 de noviembre de 2024

Investigación de seguridad e implementación	18 de noviembre de 2024	22 de noviembre de 2024

8. Divulgación y transferencia de tecnología:

Se elaborará un informe técnico que documente el desarrollo del proyecto y los resultados obtenidos. El código fuente del proyecto se publicará en un repositorio público en GitHub.

9. Presupuesto:

Sabiendo que cada hora de implementación cuesta 40$ y que cada actividad no dura más de 6 horas:

Actividad	**Horas**	**Costo $**
Implementar comunicación cliente-servidor	4	160
Implementar transferencia de archivos	6	240
Investigación de seguridad e implementación	3	120

Costo total en $: 520