1. Planejamento e Design do Projeto

A. Pesquisa e Escolha de Tecnologias

- Linguagem de Programação: Java.

- Frameworks: Considere usar frameworks para comunicação em tempo real, como WebRTC (comunicação P2P) e Socket.IO (mensagens em tempo real).

- Protocolos de Descentralização: Pesquise protocolos como \*\*WebTorrent\*\* ou \*\*IPFS\*\* para compartilhamento e mensagens descentralizadas.

#### B. \*\*Definição da Arquitetura\*\*

- \*\*Modelo Descentralizado\*\*: Decida como os usuários se conectarão, seja por conexões P2P diretas ou utilizando um servidor de sinalização descentralizado.

- \*\*Estrutura de Dados\*\*: Projete as estruturas de dados para usuários, mensagens e histórico de conversas.

- \*\*Criptografia\*\*: Escolha um método de criptografia. \*\*AES\*\* pode ser usado para criptografia de mensagens, enquanto \*\*RSA\*\* ou \*\*Diffie-Hellman\*\* para troca de chaves.

---

### 2. \*\*Implementação\*\*

#### A. \*\*Cadastro e Autenticação de Usuários\*\*

- Implemente um processo de cadastro que gere pares de chaves públicas/privadas para os usuários.

- Considere uma solução de identidade descentralizada, onde a identidade do usuário não dependa de um servidor centralizado.

#### B. \*\*Funcionalidade de Mensagens\*\*

- \*\*Envio de Mensagens\*\*: Implemente a funcionalidade de enviar mensagens criptografadas entre usuários. As mensagens devem ser armazenadas de forma que apenas o destinatário possa lê-las.

- \*\*Recepção de Mensagens\*\*: Construa a funcionalidade para receber mensagens e exibir notificações quando uma nova mensagem chegar.

#### C. \*\*Interface do Usuário\*\*

- Crie uma interface de cliente que permita visualizar as conversas. Isso pode ser um aplicativo simples para \*\*web\*\* ou \*\*mobile\*\*.

- A interface deve exibir o histórico de conversas de um usuário e permitir que ele veja as mensagens de uma conversa específica.

---

### 3. \*\*Recursos de Segurança\*\*

#### A. \*\*Criptografia de Ponta a Ponta\*\*

- Assegure que todas as mensagens sejam criptografadas antes de saírem do remetente e só sejam descriptografadas pelo destinatário.

#### B. \*\*Integridade e Autenticidade das Mensagens\*\*

- Utilize \*\*MACs\*\* (Message Authentication Codes) ou assinaturas digitais para garantir a integridade e autenticidade das mensagens.

#### C. \*\*Gerenciamento de Chaves\*\*

- Implemente um gerenciamento de chaves seguro para garantir a troca de chaves de maneira protegida. Protocolos como \*\*Diffie-Hellman\*\* podem ser usados para a troca de chaves.

---

### 4. \*\*Teste e Avaliação de Desempenho\*\*

#### A. \*\*Teste Funcional\*\*

- Realize testes para garantir que os requisitos básicos estão sendo atendidos, como envio/recebimento de mensagens e exibição do histórico de conversas.

#### B. \*\*Teste de Segurança\*\*

- Execute testes de segurança para identificar possíveis vulnerabilidades na criptografia e no processo de envio/recebimento de mensagens.

#### C. \*\*Métricas de Desempenho\*\*

- Meça a latência (tempo de envio e recebimento das mensagens), a carga de rede (sobrecarga de comunicação) e o armazenamento necessário (sobrecarga de armazenamento). Use ferramentas de benchmark para avaliar o desempenho da sua aplicação.

---

### 5. \*\*Documentação e Relatório Final\*\*

#### A. \*\*Documentação do Projeto\*\*

- Documente a arquitetura, as escolhas tecnológicas e os detalhes da implementação do aplicativo.

- Descreva os desafios enfrentados e como foram resolvidos.

#### B. \*\*Análise de Segurança\*\*

- Explique as garantias de segurança oferecidas pelo seu app. Detalhe como você assegura a \*\*privacidade\*\*, \*\*integridade\*\* e \*\*autenticidade\*\* das mensagens.

- Discuta o nível de descentralização alcançado e como isso contribui para a segurança.

#### C. \*\*Contribuições dos Membros do Grupo\*\*

- No relatório, indique claramente as contribuições de cada membro do grupo.

---

### 6. \*\*Entrega\*\*

- Prepare o código para submissão (limpo e bem comentado).

- Compile o relatório final, mantendo o limite máximo de 3 páginas, conforme as diretrizes.

---

### Requisitos Específicos

1. \*\*Decentralização\*\*:

- O sistema deve ser o mais descentralizado possível. Isso significa que a comunicação entre os pares não deve depender de um servidor central.

- Utilize protocolos P2P como WebRTC para estabelecer conexões diretas.

2. \*\*Criptografia de Ponta a Ponta\*\*:

- \*\*Troca de Chaves\*\*: Use algoritmos como \*\*Diffie-Hellman\*\* para gerar chaves seguras e permitir a troca entre usuários.

- \*\*Criptografia de Mensagens\*\*: As mensagens devem ser criptografadas usando \*\*AES\*\* ou outra forma de criptografia simétrica eficiente.

- \*\*Autenticação\*\*: Utilize \*\*assinaturas digitais\*\* para garantir que as mensagens sejam originadas do remetente correto e não foram adulteradas.

3. \*\*Interface de Cliente\*\*:

- A interface do usuário deve mostrar uma lista de conversas e permitir que o usuário visualize mensagens de uma conversa específica.

- Deve ser possível enviar e receber mensagens pela interface.

4. \*\*Garantias de Fiabilidade\*\*:

- As mensagens devem ser entregues na ordem correta e sem falhas. Isso pode ser garantido através de um protocolo que assegura a confiabilidade da comunicação.