

Análise de Requisitos

Tópicos de Segurança

David Domingues, 2220897

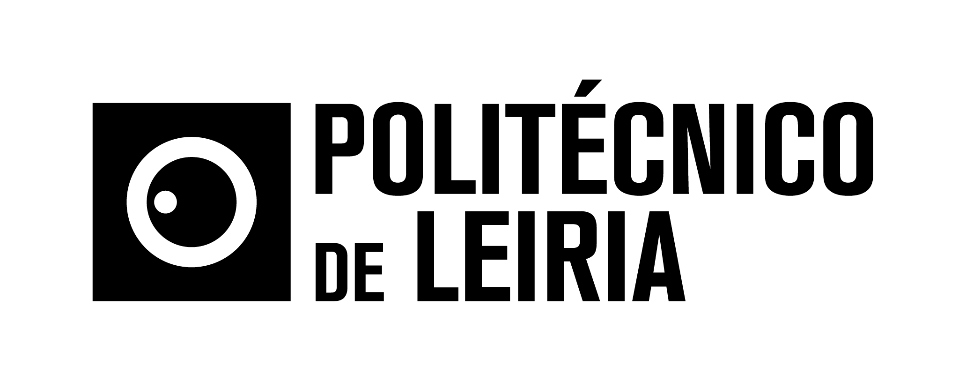
Hugo Gomes,

Rúben Soares,

2022/2023

Curso Técnico de Ensino Superior Profissional de Programação de Sistemas de Informação

Abril de 2023



Análise de Requisitos

Tópicos de Segurança

David Domingues, 2220897

Hugo Gomes,

Rúben Soares,

Ano curricular

Curso Técnico de Ensino Superior Profissional de Programação de Sistemas de Informação

Trabalho/relatório elaborado sob supervisão do Professor Nuno Simões

Abril de 2023

**Índice**

**Índice de Figuras**

**Índice de tabelas**

Não foi encontrada nenhuma entrada do índice de ilustrações.

.

# Introdução

Com o crescente uso das tecnologias de comunicação, a segurança das informações compartilhadas através da internet é uma preocupação. Para atender a essa, o objetivo deste projeto é desenvolver um chat com troca de mensagens de forma segura utilizando a linguagem de programação C#, com recurso aos componentes Windows forms, Console Application e Web App do Visual Studio Communnity. O projeto será composto por dois módulos: um módulo cliente e um módulo servidor, cada um deles com as suas próprias características.

Com o desenvolvimento deste projeto, será possível a implementação de uma comunicação segura entre cliente e servidor, envio e recebimento de mensagens de forma segura. O resultado será um chat seguro, com interface gráfica.

# Especificação do Sistema

Este projeto será constituído pelo módulo cliente e módulo servidor.

O módulo cliente, com User Interface (UI), permitirá que os usuários se comuniquem com outros, enviando e recebendo mensagens. Além disso, os usuários poderão enviar a sua chave, autenticar-se no servidor fornecendo credenciais, validar todas a mensagens trocadas com recurso a assinaturas digitais e tornar todas as comunicações o mais seguras possível.

O módulo servidor, por sua vez, não terá UI, e permite que os usuários se conectem ao servidor, autentiquem-se, recebam ligações de clientes, guardem a chave pública do cliente, validem as assinaturas do cliente e enviem e recebam mensagens de forma segura.

# Especificação de Requisitos

Para o projeto ser desenvolvido com uma estrutura bem definida, será necessário ter os requisitos mínimos exigidos no anunciado, assim será necessário que aplicação implemente com sucesso e a suas prioridades para cada função especifica. Nas tabelas abaixo estão divididos os requisitos funcionais e não funcionais. Para avaliação da prioridade das implementações, utilizaremos a escala de 1 a 5 em que 1 é a prioridade maior e 5 é a menor.

### Requisitos Funcionais

Requisitos funcionais são as funcionalidades e características que um sistema deve possuir para atender às necessidades dos utilizadores. Eles descrevem o que o sistema deve fazer ou o que o utilizador deve ser capaz de realizar ao utilizar o sistema.

Os requisitos funcionais geralmente são descritos em termos de ações ou funções específicas que o sistema deve ser capaz de realizar.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RF-01 | O sistema deve permitir trocar mensagens entre um cliente e um servidor | 1 | x |
| RF-02 | O sistema deve permitir trocar ficheiros em qualquer formato e tamanho entre um cliente e um servidor | 1 | x |
| RF-03 | O sistema deve permitir ao cliente enviar a sua chave publica | 1 | x |
| RF-04 | O sistema deve permitir ao cliente enviar e receber as mensagens de conversação de ficheiros | 1 | x |
| RF-05 | O sistema deve permitir ao servidor receber ligações de cliente (n clientes para um servidor) | 2 | x |
| RF-06 | O sistema deve permitir ao servidor receber as mensagens ficheiros do cliente | 1 | x |
| RF-07 | O sistema deve permitir que o cliente se autentique | 1 | x |

Tabela - Requisitos Funcionais

### Requisitos Não funcionais

Os requisitos não funcionais são características e restrições que o sistema deve atender, além das funcionalidades básicas que ele deve possuir. Eles descrevem as qualidades ou atributos que o sistema deve conter como por exemplo, usabilidade, fiabilidade, segurança entre outros abaixo mencionados.

#### Requisitos Não Funcionais de Usabilidade

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-USA-01 | O sistema deve conter um módulo Cliente e um módulo Servidor | 1 | x |
| RNF-USA-02 | A UI deve ter um design simples intuitivo | 1 | x |
| RNF-USA-03 | A UI deve estar em português | 2 | x |

Tabela - Requisitos Não Funcionais de Usabilidade

#### Requisitos Não Funcionais de Fiabilidade

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-FIA-01 | Em 1000 (por exemplo) conversações, o chat não deve falhar mais de 2 vezes | 2 |  |
| RNF-FIA-02 | O chat tem de estar disponível 24h/7dias | 1 | x |

Tabela - Requisitos Não Funcionais de Fiabilidade

#### Requisitos Não Funcionais de Segurança

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-SEG-01 | O sistema deve tornar todas as comunicações o mais seguras possíveis | 1 |  |
| RNF-SEG-02 | O sistema deve permitir ao cliente autenticar-se no servidor fornecendo as credenciais (username e password) | 1 | X |
| RNF-SEG-03 | O sistema deve validar todas as mensagens e ficheiros trocados com recurso a assinaturas digitais | 1 |  |
| RNF-SEG-04 | O servidor deve guardar a chave pública do cliente, na base de dados | 1 |  |
| RNF-SEG-05 | O servidor deve autenticar um utilizador já registado no sistema caso a verificação das credenciais seja correta | 1 |  |
| RNF-SEG-06 | O servidor deve validar as assinaturas do cliente | 1 |  |
| RNF-SEG-07 | O servidor ao receber a chave publica do cliente deve criar uma chave simétrica, cifrá-la com a chave publica do cliente e enviá-la | 1 |  |
| RNF-SEG-08 | O servidor deve verificar a entrada do cliente no chat com a chave simétrica | 1 |  |
| RNF-SEG-09 | O sistema deve criar um ficheiro .txt do sistema para guardar todos os dados processados pelo servidor | 1 |  |

Tabela - Requisitos Não Funcionais de Segurança

#### Requisitos Não Funcionais de Eficiência

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-EFI-01 | O tempo de resposta do sistema não deve ultrapassar 30 segundos | 3 |  |

Tabela - Requisitos Não Funcionais de Eficiência

#### Requisitos Não Funcionais de Disponibilidade

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-DIS-01 | O servidor deve estar disponível apenas para utilizadores registados que se autentiquem com sucesso no mesmo | 1 |  |
| RNF-DIS-02 | O servidor deve estar disponível 24h/7dias | 2 |  |

Tabela - Requisitos Não Funcionais de Disponibilidade

#### Requisitos Não Funcionais de Ambiente

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-AMB-01 | O sistema deverá correr em qualquer sistema operativo | 4 |  |

Tabela - Requisitos Não Funcionais de Ambiente

#### Requisitos Não Funcionais de Desenvolvimento

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-DES-01 | Deverá ser utilizada a biblioteca ProtocolSI.dll | 1 | x |
| RNF-DES-02 | A linguagem de programação usada deve ser C# | 1 | x |
| RNF-DES-03 | Utilização de criptografia assimétrica | 1 |  |
| RNF-DES-04 | Utilização de criptografia simétrica | 1 |  |
| RNF-DES-05 | O código deve estar correta e devidamente comentado | 2 |  |

Tabela - Requisitos Não Funcionais de Desenvolvimento

# Apresentação da UI

Uma imagem com diagrama

Descrição gerada automaticamenteComo a *interface* com que o comum utilizador interage é de extrema importância. Em seguida, estão apresentados *wireframes* da idealização do da *interface*.

Figura - Página Inicial da Aplicação

Uma imagem com diagrama

Descrição gerada automaticamente

Figura - Página De Inicio de Sessão/Registo

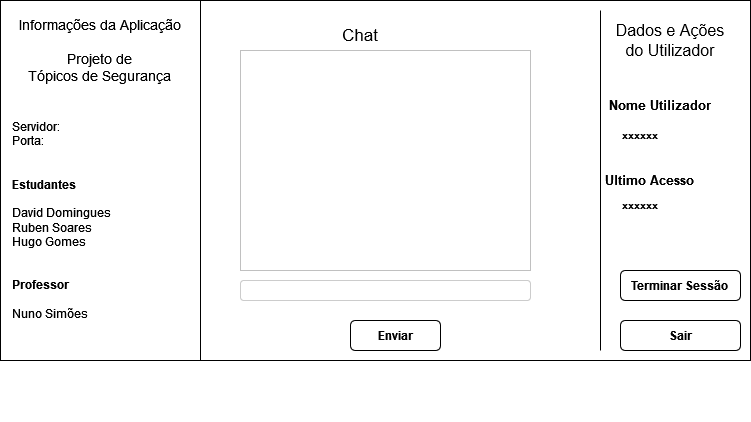


Figura - Página do Chat

# Casos de Uso

Os casos de uso são os cenários que poderão acontecer no decorrer da aplicação. Na figura seguinte, está apresentado o caso de uso que mais irá acontecer.

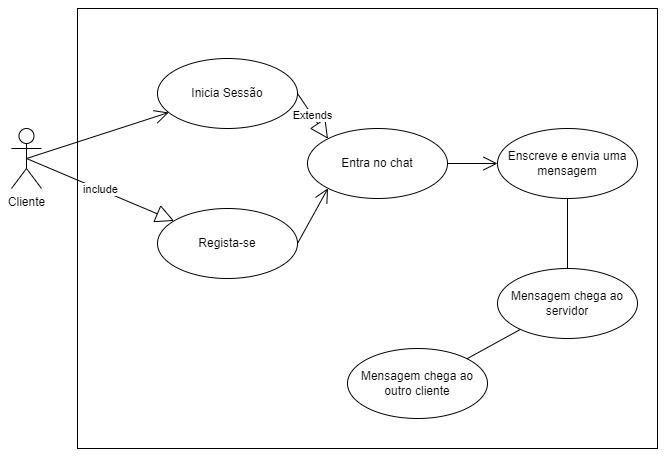


Figura - Caso de uso

# Conclusão

Em suma, o desenvolvimento de um chat com troca de mensagens de forma segura em C#, com módulos cliente e servidor, é uma tarefa desafiadora, mas que oferece uma grande oportunidade de aprendizagem. Ao longo deste projeto, foi possível aprender a implementação de uma comunicação segura entre o cliente e o servidor, envio e recebimento de mensagens de forma segura e autenticação de utilizadores.

Com a conclusão deste projeto, espera-se que o módulo cliente permita aos utilizadores que enviem e recebam mensagens, autentiquem-se no servidor, validem as mensagens trocadas e tornem todas as comunicações o mais seguras possível. Além disso, espera-se que o módulo servidor permita que os utilizadores se conectem ao servidor, autentiquem-se, validem as assinaturas dos clientes e enviem e recebam mensagens de forma segura.