|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing icon  Description automatically generated | | Curso de TeSP em  Programação de Sistemas de Informação  Tópicos de Segurança | |
|  |  | |  |
| **Ano Letivo 2024/2025** | **Av. Periódica** | | **1º Ano, 2º Semestre** |

|  |
| --- |
| **PROJETO DE TÓPICOS DE SEGURANÇA** |

**Relatório de Especificação de Requisitos do projeto de Tópicos de Segurança**

|  |  |  |
| --- | --- | --- |
| **Turno**: PL1 | **Grupo**: G | **Docente**: Nuno Salvador |
| **Nº 2241872** | Luís Abreu | |
| **Nº 2241610** | Tiago Carmo | |
| **Nº 2241594** | Rafael Campos | |

**Índice**

[1 Introdução 5](#_Toc67476478)

[2 Especificação do Sistema 6](#_Toc67476479)

[2.1 Especificação de Requisitos 6](#_Toc67476480)

[2.1.1 Requisitos Funcionais (RF) 6](#_Toc67476481)

[2.1.2 Requisitos Não Funcionais (RNF) 7](#_Toc67476482)

[2.1.3 Wireframes UI 14](#_Toc67476483)

[3 Conclusão 15](#_Toc67476484)

**Índice de tabelas**

[Tabela 1 Requisitos Funcionais 5](#_Toc67471061)

[Tabela 2 Requisitos Não Funcionais de Usabilidade 6](#_Toc67471062)

[Tabela 3 Requisitos Não Funcionais de Fiabilidade 7](#_Toc67471063)

[Tabela 4 Requisitos Não Funcionais de Segurança 8](#_Toc67471064)

[Tabela 5 Requisitos Não Funcionais de Eficiência 9](#_Toc67471065)

[Tabela 6 Requisitos Não Funcionais de Disponibilidade 10](#_Toc67471066)

[Tabela 7 Requisitos Não Funcionais de Ambiente 11](#_Toc67471067)

[Tabela 8 Requisitos Não Funcionais de Desenvolvimento 12](#_Toc67471068)

# Introdução

Este projeto foi desenvolvido no âmbito da unidade curricular de Tópicos de Segurança. O objetivo é criar uma aplicação de chat segura, dividida em dois módulos: um cliente com interface gráfica e um servidor sem interface.

A aplicação vai ser feita em C#, com o uso do Windows Forms e uma aplicação web. Vai ser usada criptografia (simétrica e assimétrica), autenticação e assinaturas digitais para garantir que as mensagens são trocadas de forma segura.

# Especificação do Sistema

Este projeto tem como objetivo criar uma aplicação de chat onde as mensagens trocadas entre o cliente e o servidor sejam seguras. Onde vamos usar criptografia para proteger as mensagens e autenticação para garantir que só entra quem tem permissão.

## Especificação de Requisitos

### Requisitos Funcionais (RF)

Os requisitos funcionais são as funções que o sistema deve ter. Em baixo estão os principais. Na fase II, vamos marcar os que foram feitos.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RF-01 | O cliente deve ser capaz de enviar a sua chave pública ao servidor | Alta |  |
| RF-02 | O servidor deve guardar a chave pública de cada cliente | Alta |  |
| RF-03 | O sistema deve permitir autenticação do utilizador com credenciais | Alta |  |
| RF-04 | O servidor deve validar a assinatura digital das mensagens recebidas | Alta |  |
| RF-05 | As mensagens trocadas devem ser cifradas com chave simétrica | Alta |  |
| RF-06 | O cliente deve conseguir enviar e receber mensagens de texto | Média |  |

Tabela 1 Requisitos Funcionais

### Requisitos Não Funcionais (RNF)

Os requisitos não funcionais são regras que garantem qualidade, segurança e bom funcionamento do sistema.

#### Requisitos Não Funcionais de Usabilidade

Os requisitos de usabilidade garantem que o sistema seja fácil e rápido de usar pelos utilizadores.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-USA-01 | |  | | --- | |  |  |  | | --- | | A interface deve ser simples e fácil de entender | | Alta |  |
| RNF-USA-02 | O sistema deve permitir acesso rápido ao chat | Alta |  |
| RNF-USA-03 | As mensagens trocadas devem aparecer de forma clara | Média |  |
| RNF-USA-04 | O login deve ser fácil de realizar | Alta |  |

Tabela 2 Requisitos Não Funcionais de Usabilidade

#### Requisitos Não Funcionais de Fiabilidade

Os requisitos de fiabilidade garantem que o sistema funcione de forma segura, estável e sem erros.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-FIA-01 | sistema deve estar sempre disponível enquanto o servidor está ativo | Alta |  |
| RNF-FIA-02 | O sistema deve conseguir recuperar se houver falha de comunicação | Alta |  |
| RNF-FIA-03 | O servidor deve conseguir lidar com vários clientes sem falhar | Alta |  |
| RNF-FIA-04 | As mensagens não devem ser perdidas em caso de erro temporário | Média |  |

Tabela 3 Requisitos Não Funcionais de Fiabilidade

#### Requisitos Não Funcionais de Segurança

Os requisitos de segurança garantem que os dados dos utilizadores e as mensagens trocadas estejam protegidos.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-SEG-01 | Todas as mensagens devem ser cifradas antes de serem enviadas | Alta |  |
| RNF-SEG-02 | O login dos utilizadores deve ser protegido com autenticação segura | Alta |  |
| RNF-SEG-03 | As chaves públicas dos clientes devem ser guardadas de forma segura | Alta |  |
| RNF-SEG-04 | O servidor deve proteger o ficheiro de logs contra acessos não autorizados | Média |  |

Tabela 4 Requisitos Não Funcionais de Segurança

#### Requisitos Não Funcionais de Eficiência

Os requisitos de eficiência garantem que o sistema funcione rápido e use bem os recursos.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-EFI-01 | O sistema deve responder rapidamente a pedidos de envio e receção de mensagens | Alta |  |
| RNF-EFI-02 | A utilização de CPU e memória deve ser otimizada | Média |  |
| RNF-EFI-03 | A cifragem e decifragem de mensagens deve ser feita de forma rápida | Alta |  |
| RNF-EFI-04 | O login dos utilizadores deve ser feito sem atrasos | Média |  |

Tabela 5 Requisitos Não Funcionais de Eficiência

#### Requisitos Não Funcionais de Disponibilidade

Os requisitos de disponibilidade garantem que o sistema esteja acessível e funcional quando necessário.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-DIS-01 | O sistema deve estar disponível para os utilizadores a maior parte do tempo | Alta |  |
| RNF-DIS-02 | O servidor deve aceitar ligações de novos clientes sem falhar | Alta |  |
| RNF-DIS-03 | Em caso de falha, o sistema deve voltar a ficar disponível rapidamente | Alta |  |
| RNF-DIS-04 | O sistema deve permitir que vários utilizadores estejam ligados ao mesmo tempo | Média |  |

Tabela 6 Requisitos Não Funcionais de Disponibilidade

#### Requisitos Não Funcionais de Ambiente

Os requisitos de ambiente definem onde e em que condições o sistema deve funcionar corretamente.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-AMB-01 | O sistema deve funcionar em computadores com Windows | Alta |  |
| RNF-AMB-02 | A aplicação cliente deve correr em Windows Forms | Alta |  |
| RNF-AMB-03 | O servidor deve funcionar numa aplicação de consola em C# | Alta |  |
| RNF-AMB-04 | O sistema deve funcionar corretamente numa rede local | Média |  |

Tabela 7 Requisitos Não Funcionais de Ambiente

#### Requisitos Não Funcionais de Desenvolvimento

Estes requisitos indicam como o sistema deve ser construído, incluindo ferramentas e boas práticas de programação.

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-DES-01 | O projeto deve ser desenvolvido em C# | Alta |  |
| RNF-DES-02 | O cliente deve ser feito com Windows Forms | Alta |  |
| RNF-DES-03 | O servidor deve ser feito em aplicação de consola | Alta |  |
| RNF-DES-04 | O código deve estar comentado e organizado | Alta |  |

Tabela 8 Requisitos Não Funcionais de Desenvolvimento

### Wireframes UI

No desenvolvimento do nosso design, os membros do grupo reuniram-se para chegar a uma ideia e como a seguir-mos-ia. As cores foram decididas em grupo e o logo feito com ajuda de ferramentas online, o Canva.



Os icons e a imagem foram retirados de uma base de dados gratuita.



# Conclusão

O desenvolvimento deste projeto permitiu criar uma aplicação de chat segura, onde atinje os objetivos definidos na fase inicial. Desenvolveram-se dois módulos principais: um cliente com interface gráfica e um servidor sem interface, implementado em C# com recurso ao Windows Forms.

Entre os aspetos mais relevantes do trabalho por agora destacam-se:

**Requisitos Funcionais e Não Funcionais:** O sistema foi feito para satisfazer todos os requisitos funcionais (como o envio e a receção de mensagens seguras) e não funcionais, onde inclui a usabilidade, a fiabilidade, a eficiência, a disponibilidade, o ambiente e as boas práticas de desenvolvimento.

**Usabilidade e Experiência do Utilizador:** A aplicação foi pensada para ser intuitiva, rápida e acessível, para facilitar a utilização de qualquer utilizador, independentemente da sua experiência.

**Colaboração e Design:** O design da interface surgiu de um esforço conjunto, com decisões partilhadas sobre cores, ícones e logótipo, com o uso do Canva para criar uma identidade visual apelativa e funcional.

Esta experiência permitiu aplicar, de forma prática, conceitos de design e de como iriamos seguir com o desenvolvimento do projeto.