

PROJETO DE TÓPICOS DE SEGURANÇA

Relatório de Especificação de Requisitos do projeto de Tópicos de Segurança

Turno: PL2	Grupo: <A>	Docente: Nuno Simões
Nº 2241868	Eduardo Carvalho	
Nº 2240100	Diego Teixeira	
Nº 2240097	Miguel Filipe	

ÍNDICE

1	INTRODUÇÃO	4
2	ESPECIFICAÇÃO DO SISTEMA	5
2.1	Especificação de Requisitos	5
2.1.1	Requisitos Funcionais (RF)	5
2.1.2	Requisitos Não Funcionais (RNF)	7
2.1.3	Wireframes UI	14
3	CONCLUSÃO	16

ÍNDICE DE TABELAS

Tabela 1 Requisitos Funcionais	6
Tabela 2 Requisitos Não Funcionais de Usabilidade	7
Tabela 3 Requisitos Não Funcionais de Fiabilidade	8
Tabela 4 Requisitos Não Funcionais de Segurança	9
Tabela 5 Requisitos Não Funcionais de Eficiência.....	10
Tabela 6 Requisitos Não Funcionais de Disponibilidade	11
Tabela 7 Requisitos Não Funcionais de Ambiente	12
Tabela 8 Requisitos Não Funcionais de Desenvolvimento	13

ÍNDICE DE FIGURAS

Figura 1 - Wireframe do Ecrã de Chat	14
Figura 2 - Wireframe da Página de Login.....	15
Figura 3 - Wireframe da Página de Criar Conta	15

1 INTRODUÇÃO

No âmbito da disciplina de Tópicos de Segurança, temos como objetivo desenvolver um projeto que consiste na criação de um sistema de comunicação cliente-servidor, com foco na segurança da troca de mensagens entre utilizadores.

Na fase inicial, o sistema irá apenas permitir a comunicação entre clientes e servidor, sem mecanismos de autenticação ou cifragem, com o objetivo de garantir que a infraestrutura base está a funcionar. Mais à frente (fase II), o sistema será melhorado com a implementação de funcionalidades de segurança, como autenticação de utilizadores, troca de chaves, criptografia simétrica e assimétrica, e validação de integridade com assinaturas digitais.

Este relatório apresenta os requisitos funcionais e não funcionais definidos para o projeto, bem como os *wireframes* da interface gráfica e outras informações que documentam o processo de desenvolvimento.

2 ESPECIFICAÇÃO DO SISTEMA

O sistema proposto será desenvolvido em C#, recorrendo a sockets TCP/IP e à biblioteca ProtocolSI. Será composto por dois módulos principais: uma aplicação cliente com interface gráfica (Windows Forms) e uma aplicação servidor em consola.

Este sistema visa permitir a troca de mensagens de forma estruturada, inicialmente sem mecanismos de segurança implementados (Fase I), e mais tarde com autenticação, criptografia e validação de integridade (Fase II).

Abaixo são descritos os requisitos funcionais e não funcionais do sistema, tendo como base a sua implementação por fases.

2.1 Especificação de Requisitos

2.1.1 Requisitos Funcionais (RF)

5

Na Fase I, o objetivo é garantir a comunicação básica entre cliente e servidor. Para isso, é implementada uma aplicação cliente com interface gráfica (Windows *Forms*) e uma aplicação servidor em consola. A troca de mensagens é feita através de *sockets* TCP/IP e estruturada com o ProtocolSI. Esta fase serve para validar a infraestrutura base do sistema, ainda sem autenticação ou mecanismos de segurança.

Na Fase II, o sistema será evoluído com a implementação de funcionalidades de segurança, nomeadamente autenticação de utilizadores, cifragem de mensagens (criptografia simétrica e assimétrica), troca de chaves e validação de integridade através de assinaturas digitais. Também será incluída a criação de ficheiros de log.

A tabela seguinte apresenta todos os requisitos funcionais definidos para o sistema, com indicação dos que já se encontram implementados nesta etapa.

# ID	Descrição	Prioridade	Implementado
RF-01	Permitir ao cliente introduzir IP e <i>Port</i> para se ligar ao servidor	Alta	X
RF-02	Estabelecer ligação entre cliente-servidor via TCP/IP	Alta	X
RF-03	Enviar mensagens do cliente para o servidor	Alta	X
RF-04	Receber e apresentar mensagens do servidor no cliente.	Alta	X
RF-05	Utilizar o ProtocolSI para processar mensagens	Alta	X
RF-06	Suportar múltiplos clientes em simultâneo (<i>threads</i>)	Alta	X
RF-07	Apresentar mensagens recebidas no cliente	Alta	X
RF-08	Apresentar mensagens de erro em caso de falha.	Média	X
RF-09	Enviar a chave pública do cliente ao servidor	Alta	
RF-10	Enviar a chave simétrica do servidor cifrada com a chave pública do cliente	Alta	
RF-11	Enviar e receber mensagens cifradas com criptografia simétrica	Alta	
RF-12	Validar a integridade das mensagens com assinatura digital	Média	
RF-13	Autenticar o utilizador com <i>username</i> e <i>password</i>	Alta	
RF-15	Guardar mensagens e ações num ficheiro de log	Média	
RF-16	Registar novos utilizadores a partir do cliente	Baixa	
RF-17 (Extra)	Permitir ao cliente introduzir um nome para se identificar nas mensagens no chat	Média	X

Tabela 1 Requisitos Funcionais

2.1.2 Requisitos Não Funcionais (RNF)

Os requisitos não funcionais definem aspetos importantes do sistema que não estão diretamente ligados às funcionalidades principais, mas que garantem a sua qualidade geral. Estes requisitos estão relacionados com a usabilidade, fiabilidade, segurança, eficiência, entre outros, e asseguram que o sistema seja fácil de usar, confiável, seguro e adequado ao ambiente onde será utilizado.

Abaixo estão representados os vários tipos de requisitos não funcionais identificados para este projeto, organizados por categoria.

2.1.2.1 Requisitos Não Funcionais de Usabilidade

Os requisitos de usabilidade têm como objetivo garantir que a interface do cliente é fácil de utilizar, clara e acessível para o utilizador.

Como a aplicação cliente é desenvolvida com *Windows Forms*, é importante que a aplicação seja intuitiva e que a troca de mensagens seja apresentada de forma clara.

7

# ID	Descrição	Prioridade	Implementado
RNF-USA-01	A interface deve ser simples e intuitiva para qualquer utilizador	Alta	X
RNF-USA-02	As mensagens recebidas devem ser apresentadas de forma visível e clara	Alta	X
RNF-USA-03	O utilizador deve conseguir ligar-se ao servidor com poucos passos	Média	X
RNF-USA-04	Os botões e campos da interface devem estar devidamente identificados	Média	X

Tabela 2 Requisitos Não Funcionais de Usabilidade

2.1.2.2 Requisitos Não Funcionais de Fiabilidade

A fiabilidade está relacionada com a capacidade do sistema funcionar de forma consistente e sem falhas durante o uso.

Para este projeto, é importante garantir que a ligação entre cliente e servidor se mantém estável e que as mensagens são enviadas e recebidas corretamente, mesmo com múltiplos clientes ligados ao servidor.

# ID	Descrição	Prioridade	Implementado
RNF-FIA-01	O sistema deve manter a ligação ativa entre cliente e servidor durante a sessão	Alta	X
RNF-FIA-02	O servidor deve conseguir lidar com múltiplos clientes sem falhas	Ala	X
RNF-FIA-03	As mensagens devem ser entregues corretamente e por ordem	Alta	X

Tabela 3 Requisitos Não Funcionais de Fiabilidade

2.1.2.3 Requisitos Não Funcionais de Segurança

A segurança é um dos principais focos deste projeto. Apesar de não ser implementada na Fase I, a Fase II irá incluir mecanismos para proteger a troca de mensagens entre cliente e servidor.

Entre os objetivos estão a autenticação de utilizadores, a proteção da confidencialidade das mensagens com criptografia, e a verificação da integridade dos dados trocados.

# ID	Descrição	Prioridade	Implementado
RNF-SEG-01	O sistema deve garantir a confidencialidade das mensagens trocadas entre clientes	Alta	
RNF-SEG-02	O sistema deve autenticar os utilizadores antes de permitir o envio de mensagens	Alta	
RNF-SEG-03	A integridade das mensagens deve ser verificada através de assinaturas digitais	Média	
RNF-SEG-04	As credenciais dos utilizadores devem ser armazenadas de forma segura (ex: hash + salt)	Alta	

Tabela 4 Requisitos Não Funcionais de Segurança

2.1.2.4 Requisitos Não Funcionais de Eficiência

Os requisitos de eficiência referem-se ao desempenho do sistema, garantindo que as operações são realizadas de forma rápida e com um bom tempo de resposta.

É importante que a troca de mensagens seja praticamente imediata, mesmo com múltiplos clientes ligados ao servidor.

# ID	Descrição	Prioridade	Implementado
RNF-EFI-01	O sistema deve garantir um tempo de resposta curto na troca de mensagens	Alta	X
RNF-EFI-02	A aplicação deve funcionar sem atrasos perceptíveis com múltiplos clientes	Média	X
RNF-EFI-03	O servidor deve libertar recursos corretamente após a desconexão dos clientes	Média	X

Tabela 5 Requisitos Não Funcionais de Eficiência

2.1.2.5 Requisitos Não Funcionais de Disponibilidade

Os requisitos de disponibilidade garantem que o sistema está acessível e operacional sempre que necessário.

É importante que tanto o servidor como o cliente consigam manter-se ativos e disponíveis durante o tempo de execução da aplicação.

# ID	Descrição	Prioridade	Implementado
RNF-DIS-01	O servidor deve estar disponível enquanto houver clientes ligados	Alta	X
RNF-DIS-02	O cliente deve conseguir reconectar ao servidor em caso de falha temporária	Média	
RNF-DIS-03	O sistema deve manter a estabilidade mesmo com várias sessões ativas	Média	X

Tabela 6 Requisitos Não Funcionais de Disponibilidade

2.1.2.6 Requisitos Não Funcionais de Ambiente

Os requisitos de ambiente referem-se ao contexto técnico onde o sistema será desenvolvido, testado e utilizado.

Esta aplicação será executada em ambientes Windows, tanto no cliente como no servidor, utilizando a framework .NET e comunicação em rede local.

# ID	Descrição	Prioridade	Implementado
RNF-AMB-01	O cliente deve ser executado num sistema operativo Windows com .NET	Alta	X
RNF-AMB-02	O servidor deve ser executado numa consola Windows	Alta	X
RNF-AMB-03	A comunicação deve ocorrer em rede local TCP/IP	Alta	X

12

Tabela 7 Requisitos Não Funcionais de Ambiente

2.1.2.7 Requisitos Não Funcionais de Desenvolvimento

Os requisitos de desenvolvimento referem-se às práticas adotadas durante a criação do sistema, com o objetivo de manter o código organizado, compreensível e de fácil manutenção.

Estas práticas facilitam o trabalho em grupo e futuras atualizações do sistema.

# ID	Descrição	Prioridade	Implementado
RNF-DES-01	O código deve estar organizado em diferentes ficheiros	Alta	X
RNF-DES-02	O projeto deve seguir uma estrutura modular para facilitar manutenções	Média	X
RNF-DES-03	O código deve incluir comentários explicativos nas partes principais	Média	X

Tabela 8 Requisitos Não Funcionais de Desenvolvimento

2.1.3 Wireframes UI

As *wireframes* desenvolvidas têm como objetivo representar visualmente a estrutura e os principais componentes das interfaces do sistema, antes da sua implementação final. Estas servem como referência para o desenvolvimento da aplicação cliente, podendo sofrer alterações entre as fases do projeto e no desenvolvimento do produto final.

Foram criadas três *wireframes* principais:

1. Ecrã de Chat (Cliente) – Representa a interface principal da aplicação cliente, onde o utilizador pode introduzir o IP e a porta do servidor, visualizar mensagens recebidas e enviar mensagens para outros clientes através do servidor.

2. Página de Login – Prevista para a Fase II do projeto, esta interface permite ao utilizador introduzir o seu nome de utilizador e palavra-passe para autenticação no sistema.

3. Página de Criar Conta – Também planeada para a Fase II, esta interface possibilita a criação de uma nova conta de utilizador com nome de utilizador e palavra-passe.

14

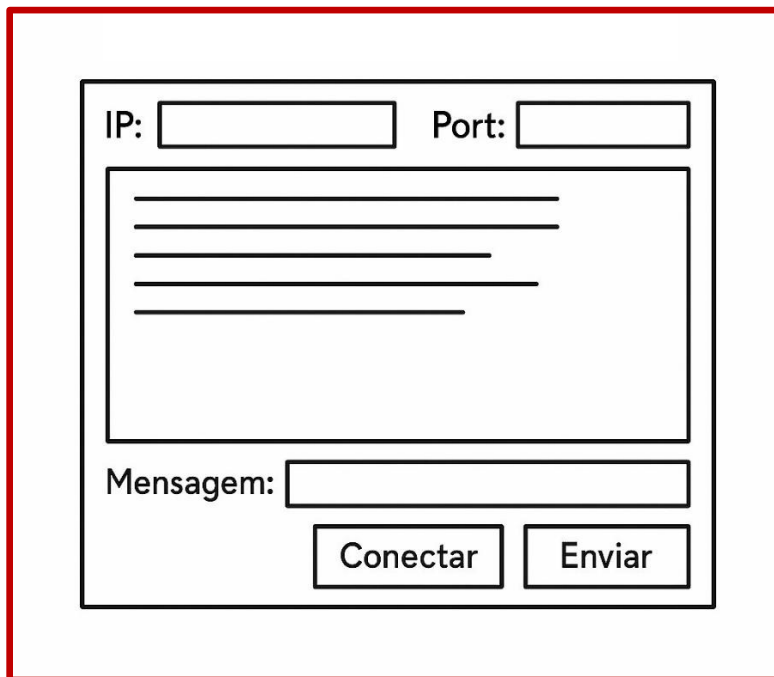
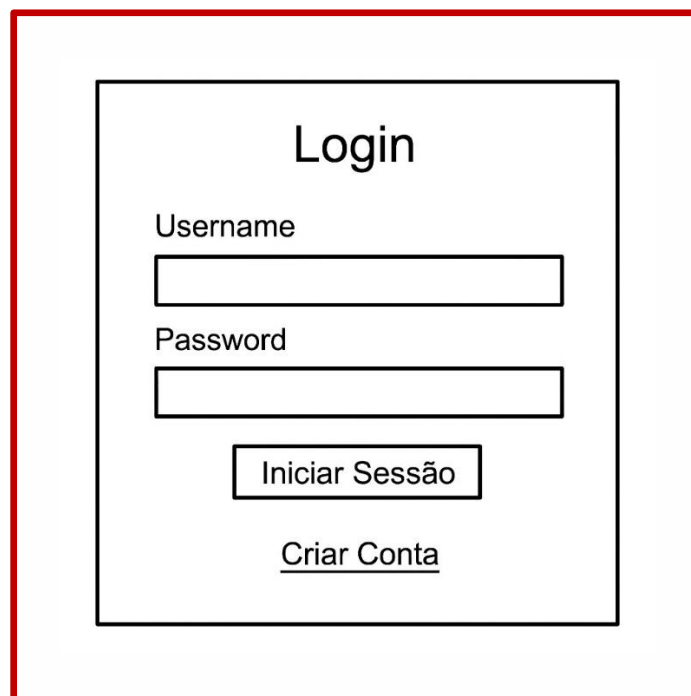
A wireframe diagram of a chat interface. It features a central rectangular frame with a red border. Inside, at the top, are two input fields labeled 'IP:' and 'Port:'. Below these is a large rectangular area representing a chat log, containing several horizontal lines of varying lengths to indicate text messages. At the bottom of the frame, there is a text input field labeled 'Mensagem:'. To the right of this field are two buttons: 'Conectar' and 'Enviar'.

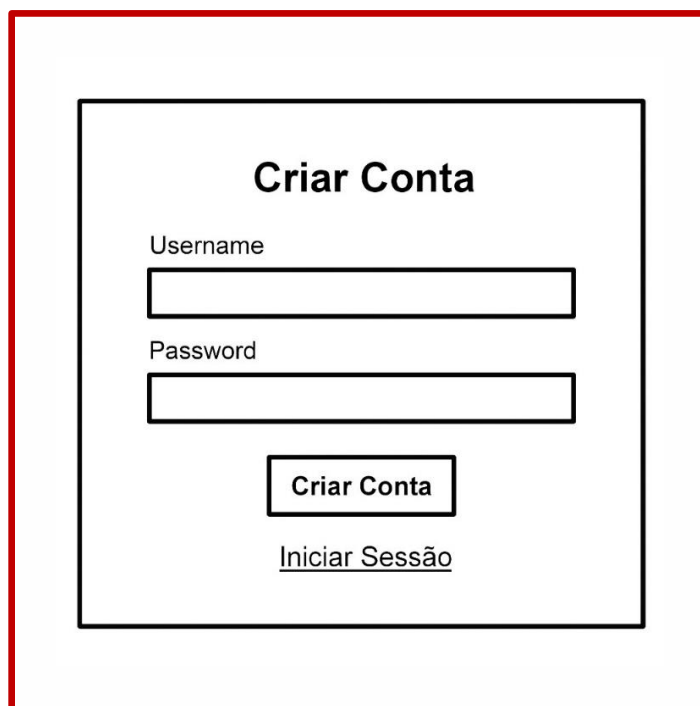
Figura 1 - Wireframe do Ecrã de Chat



A wireframe of a login page. It features a central white box with a black border. At the top, the word "Login" is centered in a large, bold, black font. Below it, the label "Username" is followed by a horizontal input field. Underneath, the label "Password" is followed by another horizontal input field. Below the password field, there is a rectangular button labeled "Iniciar Sessão". At the bottom, the text "Criar Conta" is displayed, indicating a link to the registration page.

Figura 2 - Wireframe da Página de Login

15



A wireframe of a "Criar Conta" (Create Account) page. It features a central white box with a black border. At the top, the text "Criar Conta" is centered in a large, bold, black font. Below it, the label "Username" is followed by a horizontal input field. Underneath, the label "Password" is followed by another horizontal input field. Below the password field, there is a rectangular button labeled "Criar Conta". At the bottom, the text "Iniciar Sessão" is displayed, indicating a link to the login page.

Figura 3 - Wireframe da Página de Criar Conta

3 CONCLUSÃO

Nesta primeira fase do projeto, foi desenvolvida a base do sistema de comunicação cliente-servidor, focando-se na troca de mensagens estruturadas através da biblioteca ProtocolSI. A aplicação cliente, com interface gráfica em *Windows Forms*, consegue comunicar com o servidor, que por sua vez suporta múltiplos clientes em simultâneo.

Para além dos requisitos definidos no enunciado, foram também implementadas funcionalidades adicionais com o objetivo de melhorar a experiência do utilizador. Entre elas, destaca-se uma caixa de texto que permite ao utilizador introduzir o seu nome, o qual será utilizado para identificar as mensagens trocadas no chat.

Foi igualmente adicionada uma Label que indica o estado da ligação (“Conectado” ou “Desconectado”), com alteração dinâmica da interface consoante o estado da conexão, permitindo ao utilizador saber de forma clara se está ligado ao servidor.

16

Embora nesta fase ainda não tenham sido implementadas funcionalidades de segurança, como autenticação e criptografia, estas já foram planeadas e descritas no relatório, estando previstas para a fase seguinte do projeto.

Foram ainda desenvolvidos *wireframes* que representam o conceito da interface atual e as interfaces futuras, permitindo uma visão clara da evolução prevista para o sistema. O projeto está preparado para avançar para a próxima fase, onde serão integrados os mecanismos de segurança.