|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing icon  Description automatically generated | | Curso de TeSP em  Programação de Sistemas de Informação  Tópicos de Segurança | |
|  |  | |  |
| **Ano Letivo 2023/2024** | **Av. Periódica** | | **1º Ano, 2º Semestre** |

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

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

|  |  |  |
| --- | --- | --- |
| **Turno**: PL1/2/3 | **Grupo**: <A> | **Docente**: Nuno Simões |
| **Nº 2231438** | André Eusébio Antunes Barroso | |
| **Nº 2232356** | Joel Gomes Barbeiro | |
| **Nº 2232494** | Pedro Rafael dos Santos Lourenço | |

**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 tem como objetivo o desenvolvimento de um “chat” com troca de mensagens de forma segura, utilizando a linguagem de programação C# com .NET framework na plataforma Visual Studio 2022.

O sistema será dividido em módulo cliente e módulo servidor, com funcionalidades específicas para cada um. A segurança da comunicação será uma prioridade, com a utilização de algoritmos criptográficos e autenticação para garantir a integridade e confidencialidade das mensagens trocadas.

O projeto irá envolver a implementação de sockets TCP/IP em .NET, ProtocolSI, algoritmos criptográficos, autenticação e interfaces gráficas personalizadas.

Este relatório terá como objetivo documentar os requisitos funcionais e não-funcionais, contextualizando cada um adaptado ao nosso projeto, bem como ser utilizado como uma espécie de check-list para verificarmos quando cada requisito é implementado no projeto.

No final do relatório teremos também alguns wireframes para dar uma ilustração inicial de como nós delineamos que seria a User Interface do nosso “Chat”.

# Especificação do Sistema

O chat deve permitir o envio e a receção de mensagens, mas mais do que isso, tornar esse mesmo envio e receção seguros, através de criptografia, envio de chave publica e autenticação do utilizador através de credenciais.

Este sistema vai ser dividido em dois módulos, o cliente e outro módulo servidor. Ao que esses dois vão estar distintamente separados por funções, já que, o cliente vai ter UI (User Interface) e o módulo servidor não, também as autenticações e as bases de dados estão apenas do lado do servidor para garantir que esses conceitos ficam separados do cliente.

Vamos utilizar a biblioteca ProtocolSI, utilizado para criptografia, envio e receção de dados entre cliente e servidor.

## Especificação de Requisitos

### Requisitos Funcionais (RF)

<Contextualização de requisitos funcionais a implementar>

<etapa II – marcar com um X na coluna “Implementado”>

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RF-01 | O sistema deve permitir o envio e a receção de mensagens | ALTA |  |
| RF-02 | O sistema deve permitir a autenticação do utilizador com base em credenciais | ALTA | X |
| RF-03 | O sistema deve assegurar que as comunicações são executadas de forma segura | ALTA |  |
| RF-04 | O sistema deve utilizar uma base de dados para verificar as credenciais providenciadas pelo utilizador | MÉDIA | X |
| RF-05 | O sistema deve ter implementado um módulo de servidor que gere conexões de clientes, distribui mensagens e gere autenticação e registo de utilizadores | ALTA |  |
| RF-06 | O sistema deve ter implementado um módulo de cliente que se conecta ao servidor, envia e recebe mensagens e gere a interação do utilizador. | ALTA |  |
| RF-07 | O sistema deve permitir a criação de um ficheiro de registo (log.txt) do sistema para guardar todos os dados processados pelo servidor. | ALTA |  |
| RF-08 | O sistema deve permitir que o cliente envie a sua chave pública para o servidor. | ALTA |  |
| RF-09 | O sistema deve permitir que o cliente e o servidor enviem e recebam mensagens encriptadas. | ALTA |  |
| RF-10 | O sistema deve permitir que o servidor guarde a chave pública de cada cliente. | ALTA |  |
| RF-11 | O sistema deve permitir que o servidor autentique um utilizador registado no sistema. | ALTA |  |
|  |  |  |  |

Tabela 1 Requisitos Funcionais

### Requisitos Não Funcionais (RNF)

<Contextualização de requisitos não funcionais a implementar >

#### Requisitos Não Funcionais de Usabilidade

<Contextualização de requisitos não funcionais de usabilidade a implementar>

<etapa II – marcar com um X na coluna “Implementado”>

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-USA-01 | A interface do utilizador deve ser intuitiva e fácil de usar, garantindo uma boa experiência | ALTA | X |
| RNF-USA-02 | O sistema deve fornecer feedback claro e imediato ao utilizador. | ALTA |  |
|  |  |  |  |
|  |  |  |  |

Tabela 2 Requisitos Não Funcionais de Usabilidade

#### Requisitos Não Funcionais de Fiabilidade

<Contextualização de requisitos não funcionais de fiabilidade a implementar>

<etapa II – marcar com um X na coluna “Implementado”>

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-FIA-01 | O sistema deve ser robusto e confiável, evitando falhas e garantindo a disponibilidade do serviço. | ALTA | X |
| RNF-FIA-02 | O sistema deve ser testado regularmente para garantir o seu correto funcionamento. | ALTA |  |
|  |  |  |  |
|  |  |  |  |

Tabela 3 Requisitos Não Funcionais de Fiabilidade

#### Requisitos Não Funcionais de Segurança

<Contextualização de requisitos não funcionais de segurança a implementar>

<etapa II – marcar com um X na coluna “Implementado”>

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-SEG-01 | O sistema deve garantir a segurança das comunicações, utilizando criptografia para proteger os dados transmitidos | ALTA |  |
| RNF-SEG-02 | O sistema deve utilizar algoritmos criptográficos em .NET que devem ser implementados para garantir a segurança das comunicações. | ALTA |  |
| RNF-SEG-03 | O sistema deve estabelecer a troca de chaves públicas entre cliente e servidor para garantir a segurança da comunicação. | ALTA |  |
| RNF-SEG-04 | O sistema deve validar todas as mensagens trocadas com recurso a assinaturas digitais | MEDIA |  |
| RNF-SEG-05 | O sistema deve proteger contra injeção de código ou SQL. | ALTA |  |

Tabela 4 Requisitos Não Funcionais de Segurança

#### Requisitos Não Funcionais de Eficiência

<Contextualização de requisitos não funcionais de eficiência a implementar>

<etapa II – marcar com um X na coluna “Implementado”>

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-EFI-01 | O sistema deve ser capaz de lidar com pelo menos dois clientes | ALTA |  |
| RNF-EFI-02 | O sistema deve ser capaz de suportar vários clientes simultaneamente e processar mensagens em tempo real. | ALTA |  |
| RNF-EFI-03 | O sistema deve ser capaz de trocar várias mensagens de vários clientes sem o servidor falhar | ALTA |  |
|  |  |  |  |

Tabela 5 Requisitos Não Funcionais de Eficiência

#### Requisitos Não Funcionais de Disponibilidade

<Contextualização de requisitos não funcionais de disponibilidade a implementar>

<etapa II – marcar com um X na coluna “Implementado”>

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-DIS-01 | O sistema deve ter um tempo de inatividade mínimo para que os utilizadores não sejam afetados. | ALTA |  |
| RNF-DIS-02 | O sistema deve fornecer mecanismos de backup e recuperação de dados em caso de perda ou corrupção. | ALTA |  |
| RNF-DIS-03 | O sistema deve garantir a disponibilidade contínua do serviço, mesmo em caso de falhas ou manutenção. | MEDIA |  |
|  |  |  |  |

Tabela 6 Requisitos Não Funcionais de Disponibilidade

#### Requisitos Não Funcionais de Ambiente

<Contextualização de requisitos não funcionais de ambiente a implementar>

<etapa II – marcar com um X na coluna “Implementado”>

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-AMB-01 | O sistema deve ser executado no SO Windows | ALTA |  |
| RNF-AMB-02 | O sistema deve ser independente de hardware específico. | MEDIA |  |
|  |  |  |  |
|  |  |  |  |

Tabela 7 Requisitos Não Funcionais de Ambiente

#### Requisitos Não Funcionais de Desenvolvimento

<Contextualização de requisitos não funcionais de desenvolvimento a implementar>

<etapa II – marcar com um X na coluna “Implementado”>

|  |  |  |  |
| --- | --- | --- | --- |
| # ID | Descrição | Prioridade | Implementado |
| RNF-DES-01 | O sistema deve ser desenvolvido em C# usando Windows Forms, Console Application | ALTA |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabela 8 Requisitos Não Funcionais de Desenvolvimento

### Wireframes UI

Neste capítulo damos a conhecer a primeira versão do nosso projeto antes de ser desenvolvido recorrendo a wireframes, deste modo dando a conhecer o que a equipa delineou como design para o projeto inicial.

Numa fase posterior, serão anexadas imagens da aplicação na sua versão inicial e na sua versão final, dando assim a conhecer todo o processo de evolução pelo qual o nosso projeto passou.

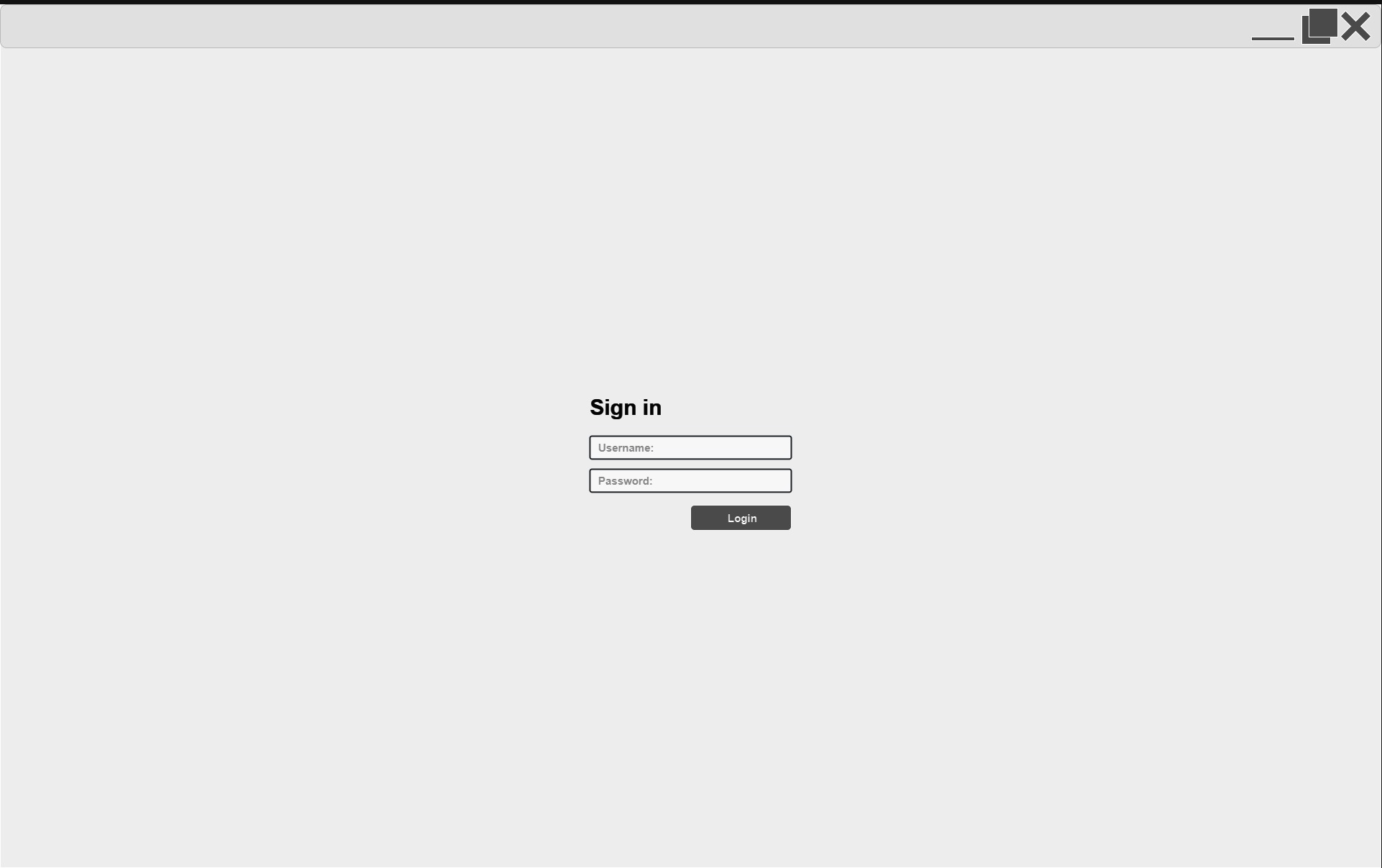


Figura 1 - Wireframe da página de Login

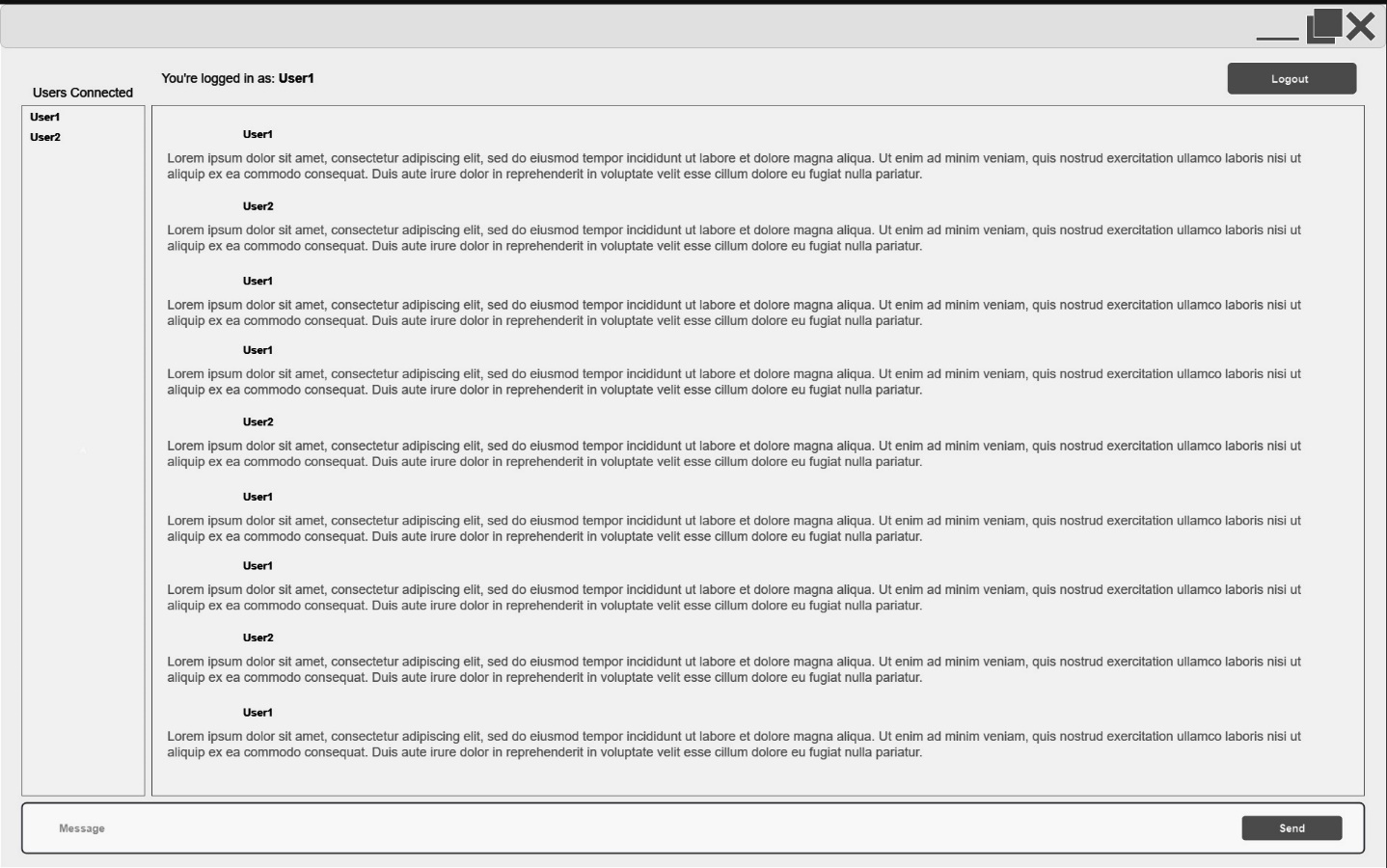


Figura 2 - Wireframe da página de chat

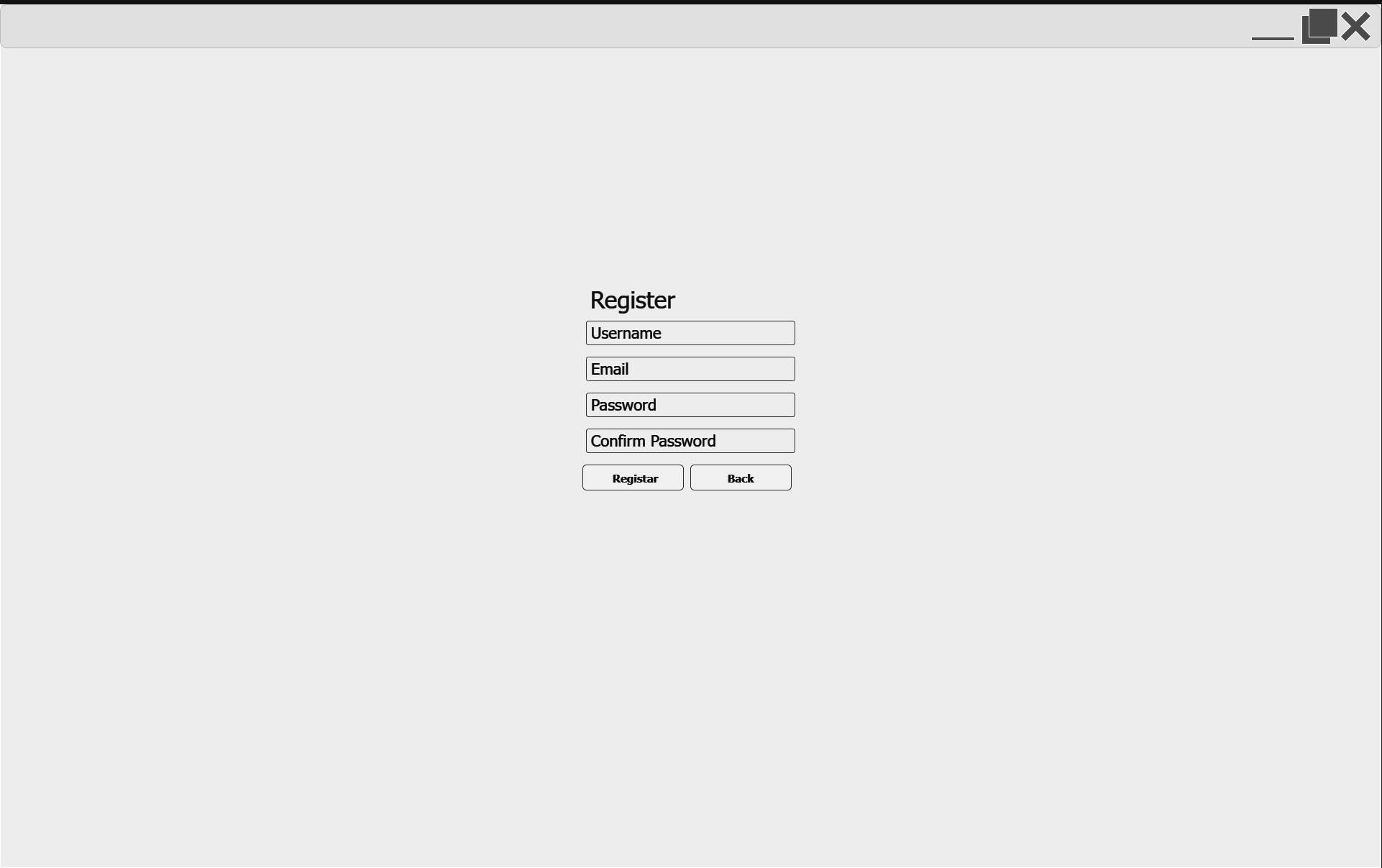


Figura 3 – Wireframe da Página de Registo

# Conclusão

Neste projeto de criação dum chat desenvolvido em C# .NET Framework utilizámos as bibliotecas do ProtocolSI para facilitar a ligação entre servidor e cliente utilizando os seus métodos e a Entity Framework para a criação e fácil manuseamento da base de dados, para além disso, o mesmo permitiu abordar vários tópicos relacionados com a cibersegurança, através do que aprendemos durante as aulas e por pesquisa própria na internet.

Ficámos a conhecer a importância da autenticação e validação de dados tal como alguns dos princípios das boas práticas da cibersegurança.

Percebemos também para a demanda do mercado atual o quão importante é a confidencialidade e integridade das aplicações onde cada vez mais a proteção de dados é um fator decisivo para a qualidade do produto final.

Explorámos protocolos de rede, manipulação de pacotes e *peer to peer communication*, aprendemos bastante sobre o protocolo TCP/IP o qual constatámos que é uma base para muitos dos outros protocolos de envio de dados e como estes são distribuídos em packets de bytes através do NetworkStream.

Com a possibilidade da criação de threads, conseguindo assim o processamento em paralelo é nos permitido tratar de vários clientes quase em simultâneo acelerando assim a resposta dada a cada cliente da parte do servidor e isolando a sua comunicação a essa thread

Além disso, implementámos um sistema de autenticação e registo de credenciais para os utilizadores, garantindo que apenas utilizadores autorizados tenham acesso ao chat.

Concluindo assim a primeira etapa deste trabalho demonstrou-se bastante benéfico para a solidificação de novos conhecimentos nesta área.