**Escola Superior de Tecnologia e Gestão**

**Licenciatura em Engenharia Informática**

**Sistemas Distribuídos**

**Ano Letivo 2024/25**

**Trabalho Prático Final**

**Elaborado em: 2025/06/15**

**Rodrigo Pereira, a2022155746**

**Guilherme Rodrigues, a2020154390**

**Índice**

[Lista de Figuras 1](#_Toc67772170)

[1 Introdução 2](#_Toc67772171)

[2 Estado da Arte 3](#_Toc67772172)

[3 Exemplos e Demonstrações 3](#_Toc67772173)

[4 Análise crítica 3](#_Toc67772174)

[5 Conclusão 3](#_Toc67772175)

[6 References 4](#_Toc67772176)

# Lista de Figuras

[Figura 1: Conexão á aplicação 4](#_Toc1)

[Figura 2: Falta de dados. 5](#_Toc2)

[Figura 3: Seleção da pasta de partilha de ficheiros 5](#_Toc3)

[Figura 4: Inicio de sessão na aplicação 1](#_Toc4)

[Figura 5: Tentativa de transferência sem seleção de um ficheiro 1](#_Toc5)

[Figura 6: Página dos logs 2](#_Toc6)

[Figura 7: Conexão de um segundo utilizador 2](#_Toc7)

[Figura 8: Conteudo da pasta partilhada 3](#_Toc8)

[Figura 9: Mensagem de upload 3](#_Toc9)

[Figura 10: Download do ficheiro concluído 4](#_Toc10)

[Figura 11: Verificação do ficheiro 4](#_Toc11)

# Introdução

Neste trabalho pretende-se criar uma aplicação com recurso a interface gráfica, onde seja possível vários clientes transferirem ficheiros entre si, através de uma tecnologia abordada durante as aulas desta unidade curricular, nomeadamente Sockets, Java RMI ou REST. Para o nosso trabalho, démos uso à tecnologia REST, já que inicialmente achámos mais simples e intuitiva para se trabalhar. Ou seja, um cliente acede a um servidor REST, e, através de endpoints, inicia e termina sessão, faz upload e download de ficheiros, e consegue aceder aos logs que o servidor envia automaticamente.

# Estado da Arte

Hoje em dia, as tecnologias de partilha de ficheiros estão a evoluir rapidamente. Cada vez mais pessoas e empresas procuram soluções que sejam seguras e descentralizadas. Uma das tecnologias mais conhecidas atualmente é o IPFS, que funciona de forma distribuída, sem precisar de servidores centrais. Ele usa um sistema de ficheiros peer-to-peer, parecido com torrents, para distribuir os dados de forma eficiente. Ao mesmo tempo, o WebRTC tem-se tornado uma ferramenta-chave para comunicação direta entre browsers, permitindo transferência de dados de baixa latência sem precisar de intermediários. Além disso, a inteligência artificial está a começar a entrar nos sistemas de transferência de ficheiros empresariais, ajudando a otimizar o uso da rede e a distribuir cargas de trabalho de forma mais inteligente. Com todos esses avanços, o futuro das partilhas de ficheiros parece apontar para um mundo onde a descentralização, a segurança de ponta a ponta e a eficiência energética são as principais prioridades das aplicações modernas.

# Exemplos e Demonstrações

* Conexão á aplicação

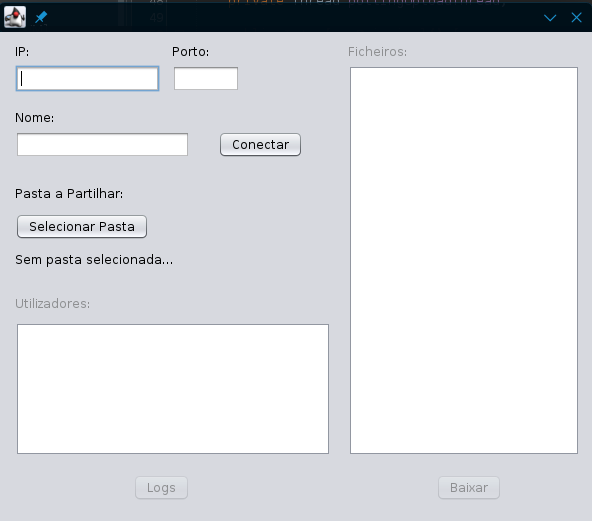


Figura 1: Conexão á aplicação

Ao iniciar a nossa aplicação a página inicial aparece com os seguintes campos disponíveis: IP, Porto, Nome e a seleção da pasta que pretendemos partilhar. Todos estes campos são de preenchimento obrigatório, em caso de falha, ou o não preenchimento de um deste o programa mostrará um erro.

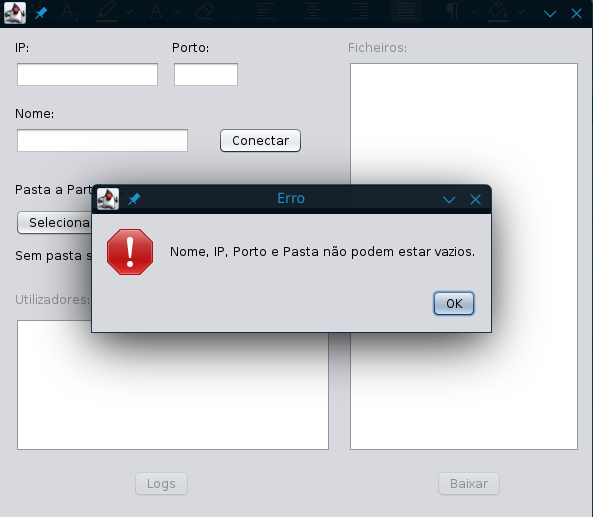


Figura 2: Falta de dados.

A seleção da pasta é feita através da própria aplicação.

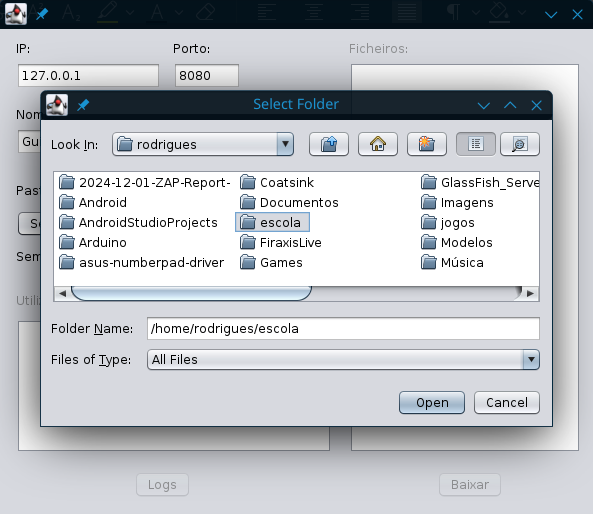


Figura 3: Seleção da pasta de partilha de ficheiros

Caso contrário e uma conexão seja bem sucedida é indicado o sucesso da conexão.

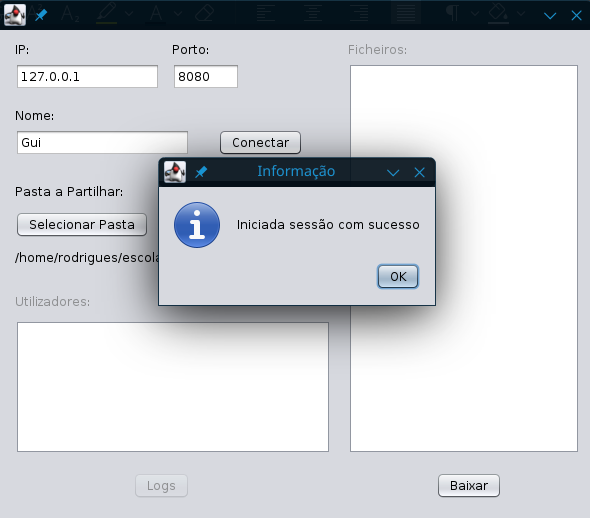


Figura 4: Inicio de sessão na aplicação

Ao iniciar sessão a nossa interface principal desbloqueia as funcionalidades de visualização dos logs e download de ficheiros, e bloqueado tudo o resto, impedindo assim que o utilizador altere qualquer tipo de informação sem antes proceder á desconexão do servidor.

O botão Baixar permite o download do ficheiro selecionado, no entanto ao clicar no mesmo sem ter selecionado um ficheiro um aviso é mostrado ao utilizador.

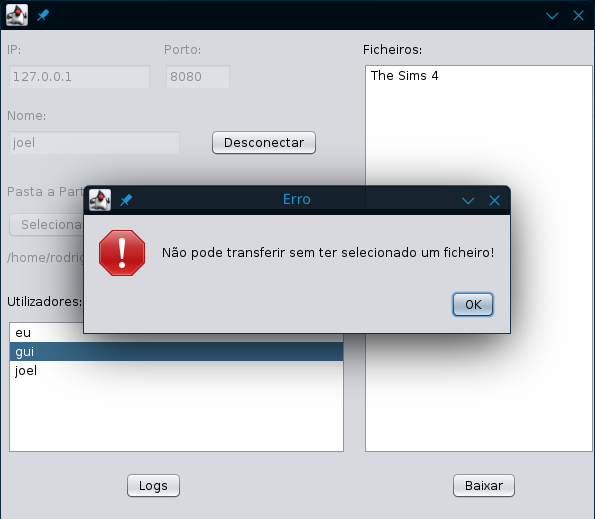


Figura 5: Tentativa de transferência sem seleção de um ficheiro

Por sua vez o botão Logs abre uma nova janela que permite a visualização de todos os logs do sistema

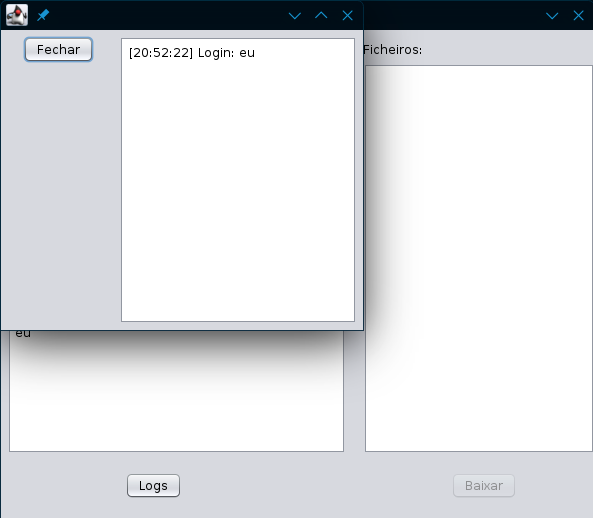


Figura 6: Página dos logs

Em seguida é mostrado a conexão de um segundo cliente. É possível observar que os logs são atualizados em tempo real assim como a lista de utilizadores.

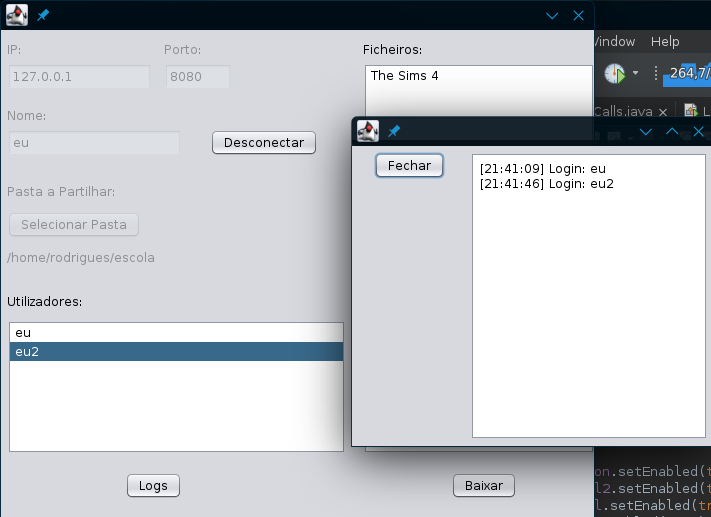


Figura 7: Conexão de um segundo utilizador

* Download de ficheiros

Primeiro verificamos a pasta de partilha e os seus conteúdos.

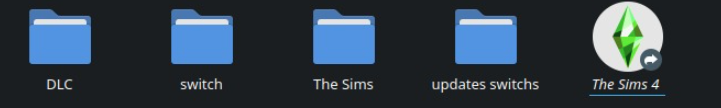


Figura 8: Conteudo da pasta partilhada

Em seguida escolhemos o ficheiro que queremos descarregar e pressionamos no botão baixar. O cliente que contem o ficheiro realiza o upload para o servidor e notifica o cliente que quer receber o ficheiro.

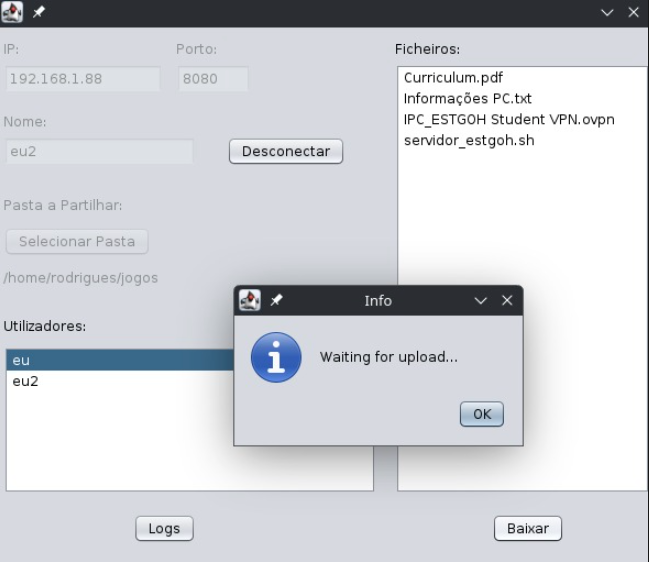


Figura 9: Mensagem de upload

Depois de alguns segundos o upload termina e é feito o download, quando este último estiver terminado o cliente é notificado.

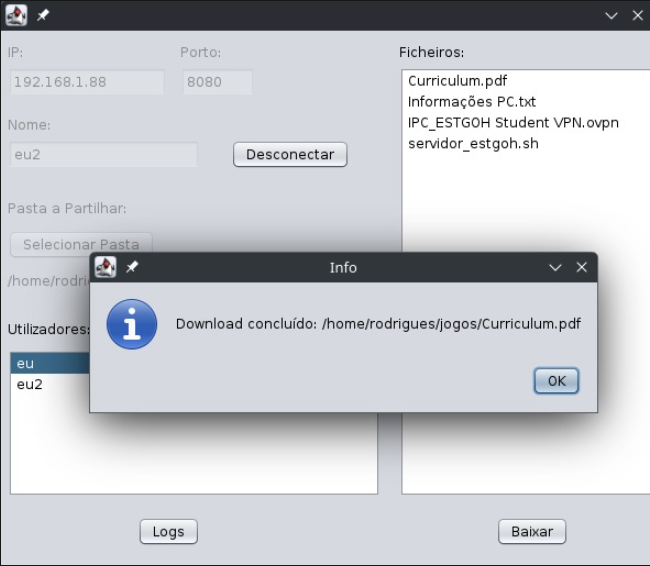


Figura 10: Download do ficheiro concluído

Por fim podemos verificar se o ficheiro realmente foi transferido.

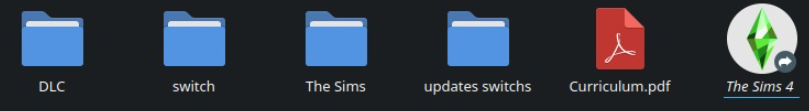


Figura 11: Verificação do ficheiro

# Análise crítica

|  |  |
| --- | --- |
| Requisitos | Consluído |
| Permitir definir o caminho para a “pasta de partilha” do programa cliente. | Concluído |
| Permitir definir o IP (ou nom e) e porto do program a servidor. | Concluído |
| Permitir definir um nome do utilizador ao fazer login na aplicação. Não é necessário prévio ou autenticação de utilizadores, porém não podem existir dois utilizadores com o mesmo nome | Concluído |

|  |  |
| --- | --- |
| A presentar o nom e de todos os utilizadores, com atualização autom ática; | Concluído |
| Apresentar os ficheiros existentes na “pasta de partilha” de cada utilizador, inclusive os do próprio utilizador. Esta apresentação pode ter atualização autom ática ou a pedido do utilizador (refresh). | Não Concluído |
| Permitir selecionar um ficheiro de um outro utilizador e transferi-lo ao próprio com putador, para a pasta de partilha local | Concluído |
| A presentar a todos os utilizadores m ensagens indicativas de login/logout de utilizadores e das transferências concluídas com sucesso (origem , destino e nom edo ficheiro). A s m ensagens devem ser precedidas da hora do servidor, no formato hh:m m :ss. | A origem não é mostrada |
| Perm itir fazer logout da aplicação sem sair desta, devendo ser possível fazer novo login. A pós o logout todas as inform ações anteriores devem ser apagadas. | Concluído |
| O program a cliente deve possuir interface gráfica. | Concluído |
| A transferência de ficheiros entre os utilizadores pode ocorrer de form a direta (entre com putadores clientes), ou via servidor. Contudo, na transferência via servidor, o ficheiro deve ser rem ovido do servidor após a operação | Concluído via servidor |
| A seleção do ficheiro a transferir deve ser feita graficam ente (ex. clicar ou selecionar um elem ento gráfico). | Concluído |
| A pós o utilizador selecionar o ficheiro a obter, deve ser apresentada um a dialog para confirm ação da transferência ou para cancelar a operação. | Concluído |

|  |  |
| --- | --- |
| Caso o ficheiro para transferência já exista na pasta de partilha local, deve ser pedido ao utilizador para confirm ar a substituição do ficheiro atual, ou cancelar a operação | Não Concluído |
| As mensagens de erro ou de sucesso devem ser apresentadas em dialogs (ex. porto inválido, transferência concluída com sucesso, etc.) | Concluído |
| A transferência de ficheiros pode utilizar um a tecnologia diferente da utilizada no restante do programa. | Concluído (Mesma tecnologia) |
| Não devem ser apresentadas ao utilizador as m ensagens em itidas anteriorm ente ao seu login. | Concluído |
| O program a servidor deve estar preparado para acesso concorrente dos clientes sem a possibilidade de ocorrerem “race-conditions” | Não Concluído |

Ao decorrer do projeto conseguimos tirar algumas conclusões.

-> outra tecnologia (sockets)

A tecnologia REST é muito útil em projetos web e outros afins no entanto a sua utilização em transferência de ficheiros mostrou-se confuso e trabalhoso. Poderíamos ter recorrido a sockets apenas para a transferência, no entanto a decisão de se manter a tecnologia foi tomada a pensar em obter mais conhecimentos sobre a mesma e como desafio de execução do trabalho.

-> melhor gestão de tempo

A gestão do tempo provou-se mais uma vez uma nuance a ser melhorada. Por ser um projeto relativamente pequeno o grau de dificuldade não se mostrou tão alto desde o início pelo que ao perceber os desafios ao longo da execução do mesmo provou-se que com uma melhor preparação e mais tempo seria possível concluir o trabalho a 100%.

# Conclusão

Com este trabalho pode-se concluir que, apesar de alguns problemas com o programa, este revela ser utilizável de maneira geral. Achamos que com mais algum tempo conseguíramos ter um programa perfeitamente executado, apesar das dificuldades com a transferência de ficheiros em REST. De qualquer forma, conseguimos aprender e adquirir competências relativamente à planificação e criação de sistemas distribuídos, que nos serão úteis no futuro.

# Referências