



**Universidade do Minho**  
Escola de Engenharia

**Laboratórios de Telecomunicações e Informática I**  
ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA  
**2020/2021**

(Docentes: José Augusto Afonso, Nuno Vasco Moreira Lopes)

9 de fevereiro de 2020

**Relatório**  
**FASE 4**

Inês Barreira Marques – [a84913@alunos.uminho.pt](mailto:a84913@alunos.uminho.pt)

José Pedro Fernandes Peleja - [a84436@alunos.uminho.pt](mailto:a84436@alunos.uminho.pt)

Rui Filipe Ribeiro Freitas - [a84121@alunos.uminho.pt](mailto:a84121@alunos.uminho.pt)

Tiago João Pereira Ferreira - [a85392@alunos.uminho.pt](mailto:a85392@alunos.uminho.pt)



# Índice

Índice de figuras .....	4
Índice de tabelas.....	5
Introdução .....	6
1. Fundamentos.....	7
2. Desenvolvimento.....	8
2.1. Desenvolvimento de código .....	9
2.1.1. Código Arduino .....	10
2.1.2. Código JAVA.....	11
2.2. Interface Gráfica .....	12
3. Testes.....	13
3.1. Transferência de um ficheiro .....	13
3.2. Aplicação de conversação.....	14
4. Discussão de resultados .....	14
Conclusão.....	15
Referências .....	16
Autoavaliação .....	17

## Índice de figuras

Figura 1 - Fluxograma geral. ....	9
Figura 2 - Fluxograma Arduino. ....	10
Figura 3 - Fluxograma JAVA. ....	11
Figura 4 - Interface gráfica (CHAT). ....	12
Figura 5 - Interface gráfica (Primeiro Ecrã). ....	12
Figura 6 - Demonstração da transferência de um ficheiro.....	13
Figura 7 - Transferência ficheiro MENU. ....	13
Figura 8 - Transferência de um ficheiro imagem.....	13

## Índice de tabelas

Tabela 1 - Decisões tomadas ao longo do projeto. ....	8
---	---

# Introdução

No âmbito da Unidade Curricular de LTI I foi-nos proposto o desenvolvimento de uma aplicação que permita a conversação em modo texto, em tempo real, entre dois computadores pessoais. Pretende-se que para além da possibilidade de ambos os utilizadores conversarem em modo texto possam enviar os mais variados ficheiros incluindo imagens.

Para a realização deste projeto iremos utilizar o modelo OSI que foi devidamente explicado na fase 1. As camadas que utilizaremos neste projeto serão as camadas 1, 2 e 7 que correspondem à camada física, à camada de ligação de dados e à camada de aplicação onde nesta segunda fase foca-se particularmente na sétima.

Este projeto está dividido em 4 fases em que cada uma terá uma entrega de um relatório e demonstração prática da aplicação em funcionamento, com exceção da última fase que para além disso terá também uma apresentação oral.

O presente relatório é da fase 4 onde os objetivos são entender a noção da camada de aplicação (protocolo de nível 7), entender como desenvolver a conversação e transferência de ficheiros. Para tal será necessário realizar uma interligação entre as fases anteriores com a fase atual, perceber como interligar as camadas seguindo o princípio da independência de camadas. Por exemplo, a camada 2 não deve conhecer se a informação que está a transportar pertence (ou não) a um ficheiro, enquanto que a camada 7 não deve ter conhecimento das tramas de controlo de erros trocadas no interior da camada 2.

Desta forma, para realizar um projeto com maior desempenho e com todas as funcionalidades, temos de integrar as fases anteriores e otimizar, reduzindo assim o tempo de transferências dos ficheiros ao mínimo possível.

Em suma, para sermos capazes de cumprir com os objetivos deste projeto semestral devemos pôr em prática conhecimentos adquiridos noutras unidades curriculares, nomeadamente Redes de Computadores I, Sistemas Operativos, Paradigmas de Programação, entre outras.

# 1. Fundamentos

Como referido em fases anteriores utilizamos o modelo OSI para definir as camadas que necessitávamos desenvolver. Desta forma, com a camada 1 e camada 2 (camada física e camada de ligação) já realizadas esta fase consiste em desenvolver a sétima camada (aplicação). Esta fase é constituída exclusivamente por software, e corresponde às aplicações que serão utilizadas para que consigamos garantir uma interação eficiente entre o computador e o utilizador.

Esta camada determina um protocolo de aplicação de um domínio específico de aplicação. Para tal, é necessário realizar a integração das funcionalidades anteriores, tais como o controlo de erros.

Nesta camada também foi nos recomendado a elaboração de uma interface gráfica que infelizmente não chegamos a uma solução eficiente que permita a transferência eficaz de ficheiros. Explicaremos este assunto na discussão de resultados visto que apenas conseguimos fazer uma aplicação de conversação com interface gráfica.

## 2. Desenvolvimento

Na solução final do sistema elaborado pelo nosso grupo tivemos de tomar várias decisões de modo a tornar o produto final o mais eficiente possível. Em seguida apresentamos essas mesmas opções tomadas por nós através de uma tabela.

*Tabela 1 - Decisões tomadas ao longo do projeto.*

Assunto	Opção escolhida
Técnica de controlo de fluxo	<i>Stop-and-wait</i>
Técnica de correção de erros	<i>Stop-and-wait</i>
Técnica de deteção de erros	CRC-8
Endereço Emissor	$00_2$
Endereço Recetor	$01_2$
Trama de dados	$001_2$
Trama ACK	$010_2$
Trama NACK	$011_2$
Trama Final	$111_2$
Número de sequência	$0_2$ ou $1_2$
Linguagem interface gráfica	JAVA



## 2.1. Desenvolvimento de código

Nesta secção apresentamos vários fluxogramas que foram auxiliares ao desenvolvimento da nossa aplicação. Na figura 1 está presente o fluxograma geral onde inicialmente é definido os canais para a comunicação entre os *transceivers*, assim como declarar as variáveis necessárias e a configuração dos módulos RF. De seguida desabilitamos a deteção de erros e de controlo de fluxo no *transceiver* e configuramos o canal que irá ser utilizado para a conexão, que no nosso caso é o 60 correspondente a uma frequência de 2460 MHz.

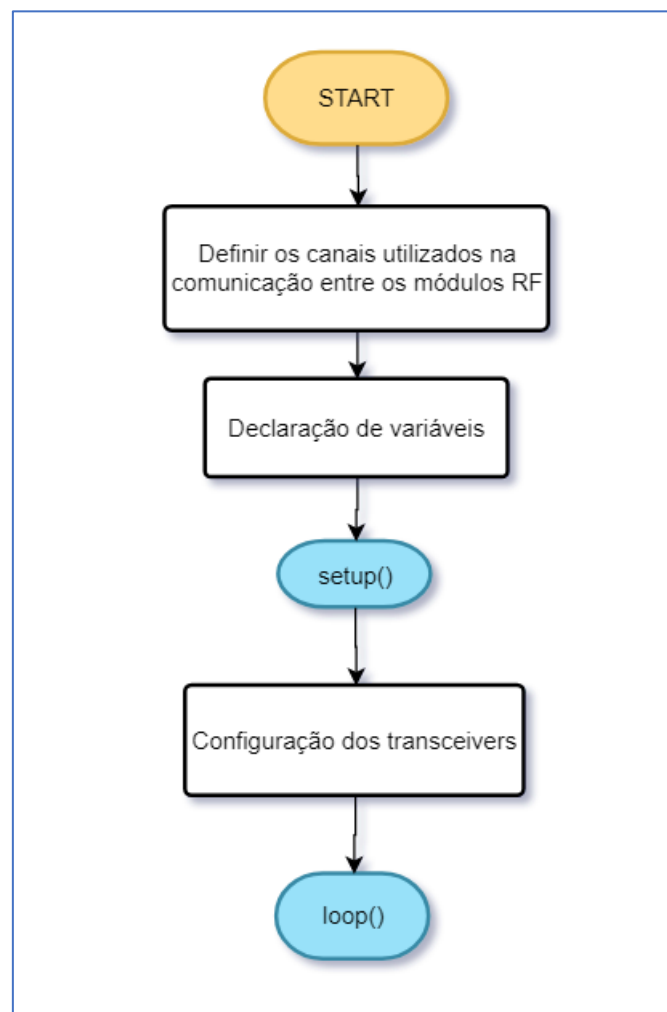


Figura 1 - Fluxograma geral.

### 2.1.1. Código Arduino

Para esta fase, as alterações efetuadas no código Arduino não foram muito significativas em relação à fase anterior, apenas tivemos de corrigir certos erros que foram ocorrendo aquando da execução do código em JAVA e alterar a forma como era iniciada a transferência entre ambas as máquinas. Na figura seguinte está presente um pequeno fluxograma que explica o processo realizado no Arduino. Dentro da função *setup()* é realizada a configuração dos *transceivers* e dentro da função *loop()*, no caso do emissor, este fica à espera de receber algo da porta série proveniente do código JAVA e reenvia o conteúdo com a adição do *header* e o CRC para o recetor. No caso do recetor este espera pelo envio de bytes por parte do emissor e depois envia para a porta série os bytes recebidos para que o programa em JAVA os possa processar quer seja através de um chat quer seja através de ficheiros.

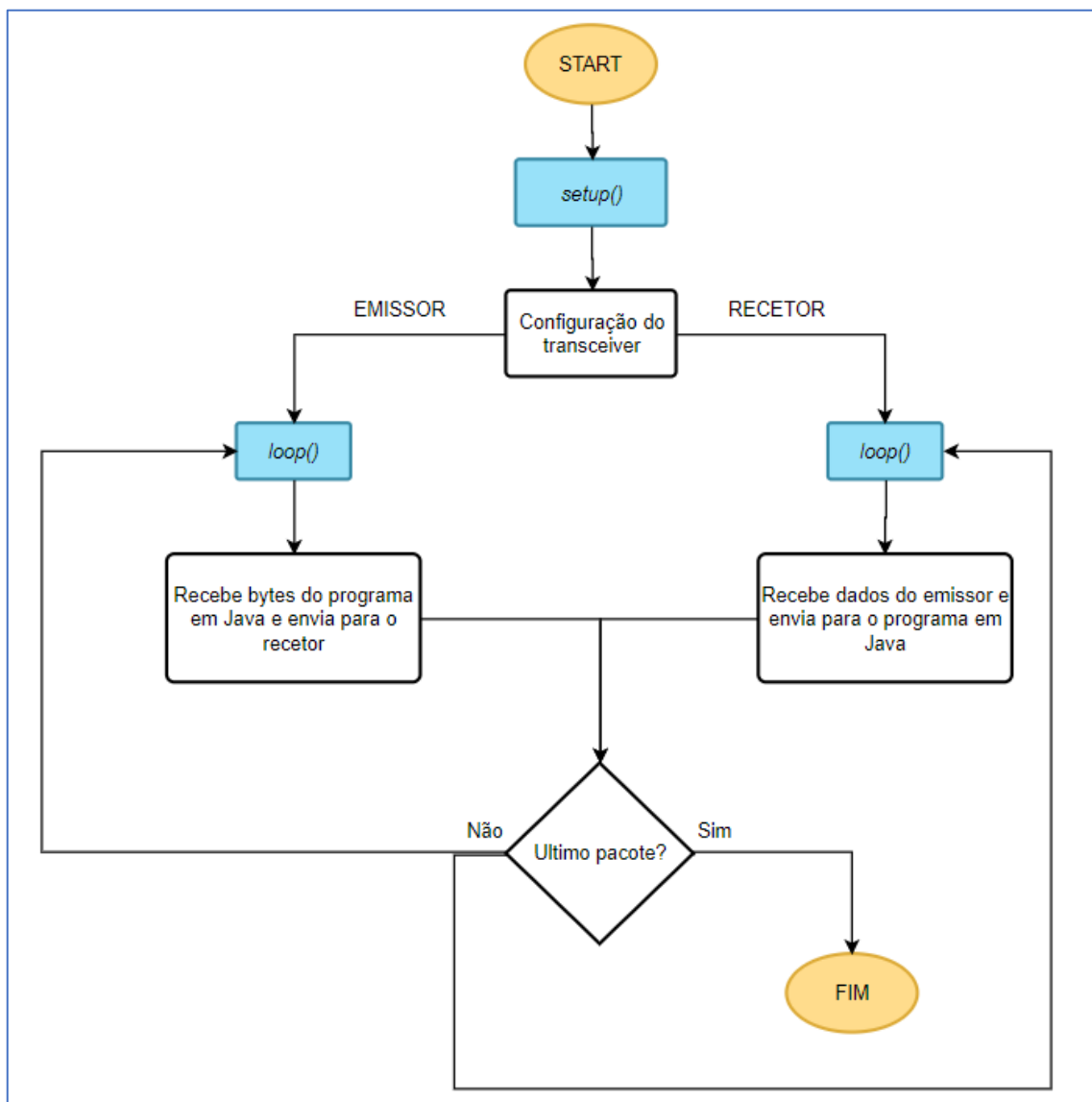


Figura 2 - Fluxograma Arduino.

### 2.1.2. Código JAVA

Em relação ao código efetuado em JAVA este foi feito de modo a cumprir 2 objetivos. O primeiro que passa pela interligação entre o computador e o Arduino e o segundo que corresponde à elaboração da interface gráfica que permite uma forma mais fácil e eficiente de o utilizador controlar o que pretende fazer.

Em relação à comunicação computador-*transceiver*, na fase 1 foi realizado um código mais simples que permitia a transferência de ficheiros entre 2 computadores com o auxílio das portas séries. Para esta fase tínhamos de pegar nesse código já realizado e adaptá-lo para que em vez de utilizarmos as portas séries para comunicar um com o outro utilizássemos os *transceivers* como comunicador entre estes, ou seja, o emissor envia bytes para a porta série através do programa em JAVA, no Arduino este pega nesses bytes e encapsula adicionando o *header* e o CRC e envia para o recetor, que recebe os bytes encapsulados, observa se ocorreram erros e caso tudo tenha corrido bem envia o *payload* dos bytes recebidos para o programa em JAVA do recetor.

Para enviar ficheiros com tamanho superior a 30 bytes foi necessário partir o ficheiro em pacotes de 30 bytes e escrevê-los sequencialmente na porta série. Após cada pacote ser escrito na porta série e enviado com sucesso para o recetor, o Arduino envia uma confirmação de volta para o pc através de 2 bytes. Depois de receber a confirmação o programa envia o pacote seguinte, permitindo assim controlar o fluxo de bytes pela porta série e prevenir a escrita do pacote seguinte antes do anterior ser recebido com sucesso.

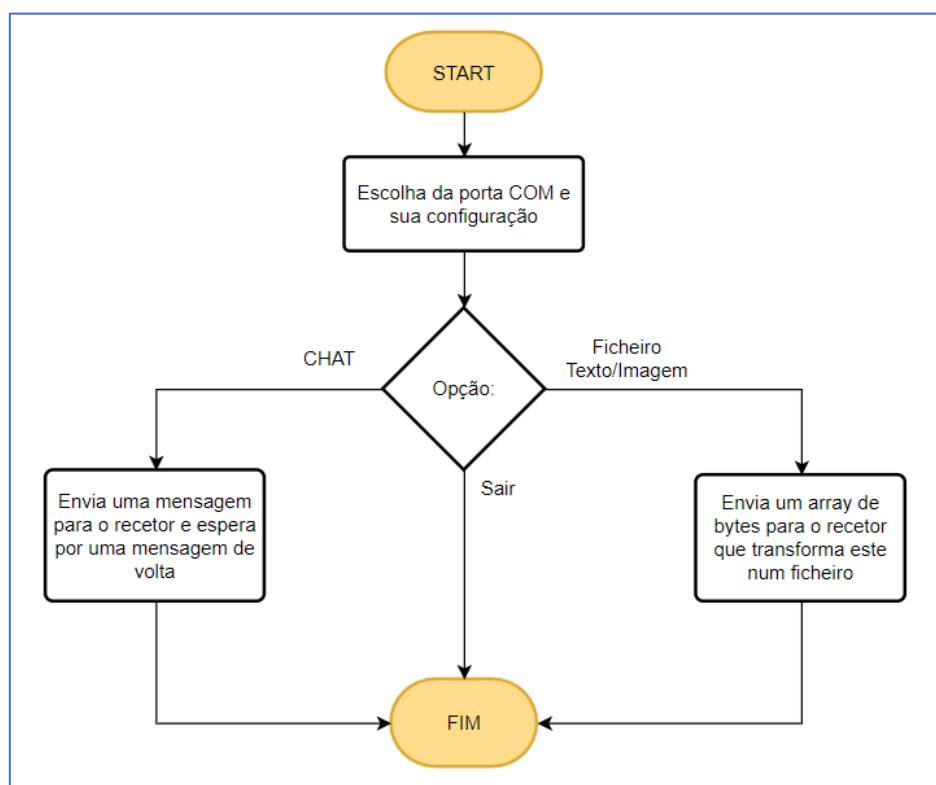


Figura 3 - Fluxograma JAVA.

## 2.2. Interface Gráfica

Em relação à parte opcional desta fase 4 era nos proposto a realização de uma interface gráfica [1] de modo a tornar mais fácil a conexão entre o utilizador e o computador. A figura 5 corresponde ao primeiro ecrã devolvido ao utilizador quando começa a aplicação, onde pede ao utilizador para colocar qual a porta série que quer utilizar de uma lista de portas disponíveis. Na figura 4 apresentamos a conexão entre 2 computadores através de um chat realizado por nós.

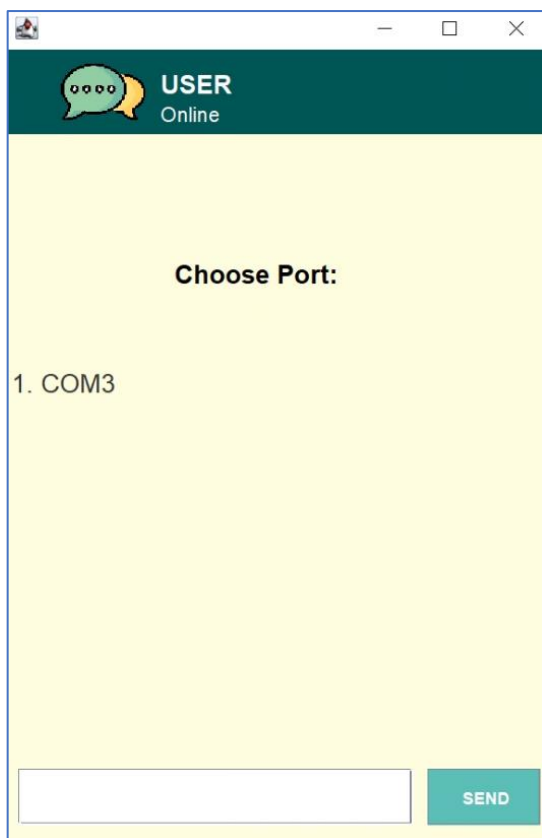


Figura 5 - Interface gráfica (Primeiro Ecrã).

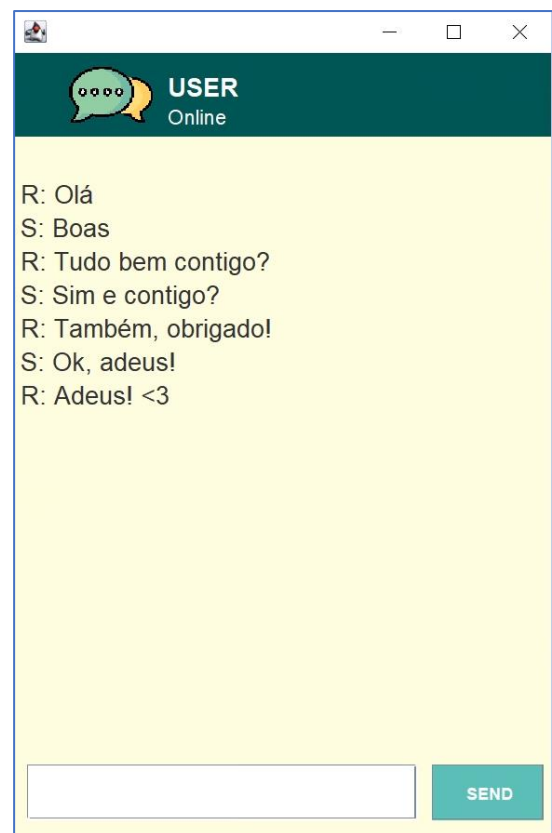


Figura 4 - Interface gráfica (CHAT).

### 3. Testes

#### 3.1. Transferência de um ficheiro

Na fase de testes para esta fase enviamos ficheiros de texto e de imagem entre ambos os PC's com e sem erros para ver se tudo estava a correr de acordo com o esperado. Ficheiros de texto conseguimos efetuar com sucesso como demonstrado no vídeo de demonstração e demonstramos nas imagens 6 e 7 o processo a ser desenvolvido aos olhos do utilizador. Na primeira observamos o menu fornecido ao utilizador quando inicia a aplicação e na segunda a observação do ficheiro enviado e do recebido em que são iguais. Relativamente aos ficheiros de imagem não obtivemos tanto sucesso visto que apenas conseguimos enviar ficheiros de imagem com pouco tamanho (à volta dos 3500 bytes). Na transferência eram obtidas imagens com bastantes erros que não eram nada semelhantes às originais, outras vezes eram enviadas imagens parecidas como demonstrado na figura 8. Infelizmente não chegamos a tempo à origem do erro pelo que não colocamos a transferência de um ficheiro de imagem no vídeo de demonstração.

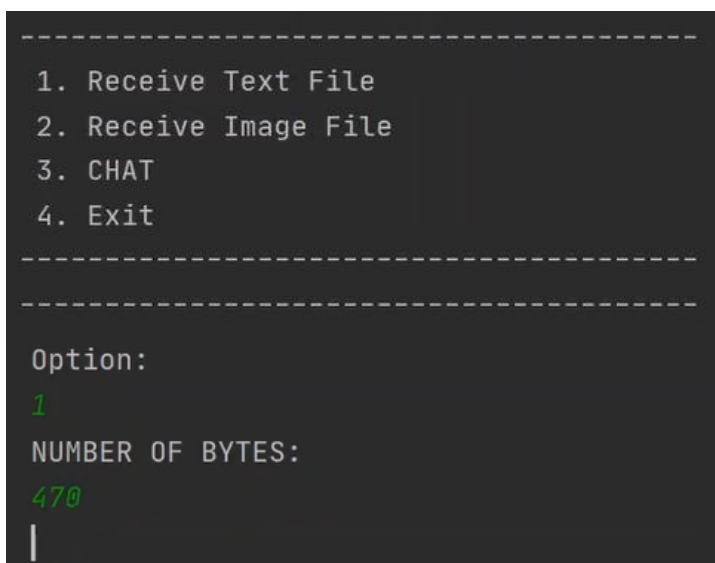


Figura 7 - Transferência ficheiro MENU.

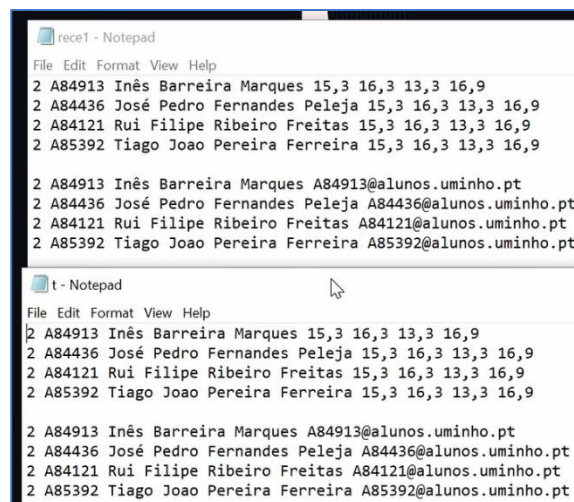


Figura 6 - Demonstração da transferência de um ficheiro.

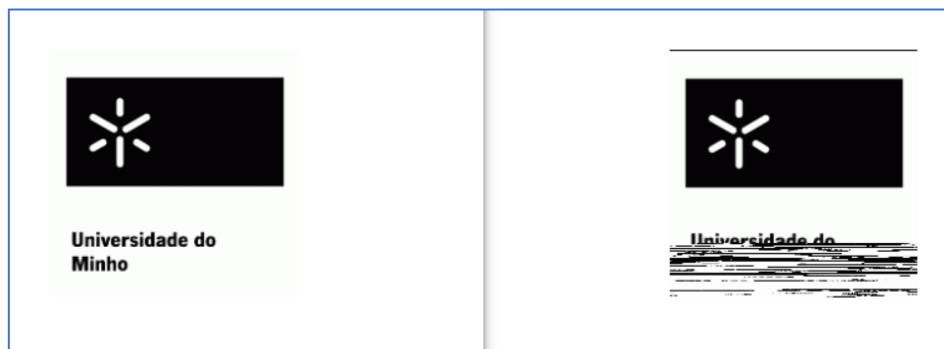


Figura 8 - Transferência de um ficheiro imagem.

### 3.2. *Aplicação de conversação*

Em relação à aplicação de conversação decidimos realizar uma interface gráfica de modo a melhorar a experiência do utilizador. Os testes realizados para o funcionamento do chat foram demonstrados no vídeo de demonstração anexado aos ficheiros enviados. A demonstração destes estão na secção interface gráfica onde estão as imagens demonstrativas da aplicação fornecida ao utilizador.

## 4. Discussão de resultados

Nesta fase os testes feitos pelo nosso grupo tiveram mais destaque na transferência com sucesso e com erros de ficheiros. Em relação aos ficheiros de texto estes tinham uma taxa de sucesso perto dos 100% o que torna a transferência de ficheiros de texto eficientes. No entanto, os ficheiros de imagem apenas tinham uma taxa de sucesso a rondar os 40% quando o tamanho era inferior a 20 mil bytes. Quando era superior raramente era obtido sucesso.

## Conclusão

Como foi pedido no enunciado do projeto todas as camadas implementadas dependem apenas de si para funcionar, ou seja, o Arduino não tem ideia do que está a enviar, apenas sabe que recebe bytes e envia bytes. Graças a isto é possível enviar qualquer tipo de ficheiro visto que o Arduino apenas recebe e envia bytes, o programa em JAVA é que tem de se preocupar em transformar o ficheiro em bytes de modo a reencaminhar estes.

Para desenvolvimento de código e do relatório foi necessária contribuição de todo o grupo e junção de conhecimentos e entreajuda. Assim conseguimos alcançar os objetivos necessários para este projeto. Acabamos com a convicção que realizamos um bom trabalho ao longo do semestre e sabendo que ganhamos sabedoria para o futuro através da realização deste. De salientar que ultrapassamos os obstáculos encontrados ao longo do semestre com o esforço e empenho de todos tendo em conta o ano atípico em que realizamos este projeto face à pandemia mundial. Isto trouxe-nos dificuldades na comunicação e no trabalho em grupo.

## Referências

- [1] <https://docs.oracle.com/javase/tutorial/uiswing/> , JAVA Documentation.



# Autoavaliação

**Inês Barreira Marques:** O meu contributo nesta fase foi na realização do relatório, bem como na pesquisa para a realização do mesmo, ajudei também na elaboração do *PowerPoint*.

**José Pedro Fernandes Peleja:** Nesta fase contribui com a elaboração do *PowerPoint* também contribui na pesquisa e escrita do relatório. Ajudei também na divisão de tarefas.

**Rui Filipe Ribeiro Freitas:** Para esta fase contribui na elaboração do código assim como na realização dos testes. Ajudei também na pesquisa e escrita do relatório bem como na sua revisão.

**Tiago João Pereira Ferreira:** Contribui no desenvolvimento dos vários códigos e estive presente na realização dos testes. Para além disso ajudei na elaboração do relatório.