



**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)

29 de outubro de 2020

**Relatório**  
**FASE 1**

Inês Barreira Marques - [a84923@alunos.uminho.pt](mailto:a84923@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

<b>Índice de figuras.....</b>	<b>4</b>
<b>Índice de tabelas.....</b>	<b>5</b>
<b>Lista de abreviaturas.....</b>	<b>6</b>
<b>Introdução.....</b>	<b>7</b>
<b>1. Especificação do projeto.....</b>	<b>8</b>
1.1. Modelo OSI .....	8
1.2. Descrição das camadas.....	9
1.3. Modos de transmissão .....	10
1.4. Arquitetura do sistema.....	11
1.4.1. Arquitetura de <i>hardware</i> .....	11
1.4.2. Arquitetura de <i>software</i> .....	12
1.5. Identificação das tecnologias e recursos .....	13
1.5.1. Recursos ao nível do <i>hardware</i> .....	13
1.5.2. Recursos ao nível do <i>software</i> .....	14
1.6. Discussão sobre as competências necessárias .....	15
1.7. Planeamento de Tarefas .....	16
<b>2. Comunicação via porta série PC-Arduino-Arduino-PC.....</b>	<b>17</b>
2.1. Fundamentos.....	17
2.2. Desenvolvimento.....	18
2.2.1. Descrição do código em JAVA.....	18
2.2.2. Fluxograma do programa em JAVA.....	19
2.3. Testes.....	20
2.4. Discussão de resultados .....	21
<b>Conclusão .....</b>	<b>22</b>
<b>Referências.....</b>	<b>23</b>

## Índice de figuras

Figura 1 - Camadas do Modelo OSI. ....	8
Figura 2 - Transmissão Simplex. ....	10
Figura 3 - Transmissão Half-Duplex. ....	10
Figura 4 - Transmissão Full-Duplex. ....	10
Figura 5 - Arquitetura do Sistema. ....	11
Figura 6 - Diagrama de Gantt. ....	16
Figura 7 - Comunicação UART. ....	17
Figura 8 - Constituição de uma trama. ....	17
Figura 9 - Configuração dos parâmetros da porta série. ....	17
Figura 10 - Tempo de transmissão. ....	18
Figura 11 - Menu da aplicação. ....	18
Figura 12 - Fluxograma do programa em JAVA. ....	19

## Índice de tabelas

Tabela 1 - Recursos ao nível do hardware.....	13
Tabela 2 - Recursos ao nível do software.....	14
Tabela 3 - Comparação entre tempo teórico e experimental de um ficheiro texto.....	20
Tabela 4 - Comparação entre tempo teórico e experimental de um ficheiro imagem.	20

## Lista de abreviaturas

**OSI** - Open System Interconnection

**LTI I** - Laboratórios de Telecomunicações e Informática I

**PC** - Personal Computer

**LLC** - Logic Link Control

**MAC** - Media Access Control

**RF** - Radiofrequência

**USB** - Universal Serial Bus

**IDE** - Integrated Development Environment

**SPI** - Serial Peripheral Interface

**UART** - Universal Asynchronous Receiver/Transmitter

# Fase 1

## Introdução

No âmbito da Unidade Curricular de LTI I (Laboratórios de Telecomunicações e Informática 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 será devidamente explicado posteriormente. 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.

Este projeto está dividido em 5 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 corresponde à fase 1 em que os objetivos principais passam pela elaboração da arquitetura do sistema, identificação dos recursos e tecnologias necessárias para a realização do projeto, o planeamento através de um diagrama de *Gantt* de modo a haver uma melhor organização por parte do grupo e implementação de um programa no PC para a transferência de um ficheiro de texto e de um ficheiro de imagem por cabo via porta série entre dois computadores.

De modo a 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. Especificação do projeto

## 1.1. Modelo OSI

O modelo OSI (Open System Interconnection), foi criado em 1984 pela ISO (International Standards Organization), pois surgiu a necessidade de criar uma arquitetura padrão para que pudesse ser feita uma comunicação entre sistemas abertos, ou seja, aparelhos de diferentes fabricantes. Este modelo é desde então usado como base para a construção de qualquer tipo de rede e foi desenvolvido com o intuito de ser o padrão internacional dos protocolos utilizados nas várias camadas.

Este modelo está dividido em 7 camadas em que cada uma define as suas funções, fornecendo serviços às camadas superiores e recebendo de camadas inferiores.

Como já foi referido na introdução deste relatório apenas iremos utilizar três camadas para a elaboração deste projeto. Elas são a camada física, a camada de ligação de dados e a camada da aplicação.



*Figura 1 - Camadas do Modelo OSI.*



## 1.2. Descrição das camadas

O modelo OSI está dividido em camadas hierárquicas em que cada camada tanto fornece serviços às camadas superiores como recebe de camadas inferiores de modo a simplificar as operações para o utilizador. Em seguida iremos descrever as camadas que utilizaremos neste projeto.

### ➤ 1ª Camada - Física

A camada física é a camada mais baixa do modelo OSI, em que o principal objetivo é transmitir e receber bytes de informação através de um canal de comunicação. Esta camada constitui basicamente o *hardware* necessário para a comunicação.

### ➤ 2ª Camada - Ligação de dados

Esta camada permite o controlo de erros da camada física, controla o fluxo de dados e sincronização de tramas/pacotes. Esta camada está dividida em 2 subcamadas:

- **LLC (Logic Link Control)** - esta subcamada controlo o fluxo de dados, faz a gestão de erros e comunica com camadas superiores, e tem como objetivo ajudar na entrega da trama/pacote ao destino.
- **MAC (Media Access Control)** - a subcamada MAC é implementada pelo *hardware*. Acede aos dados, controlando as permissões dos mesmos para que não haja colisão de tramas/pacotes havendo assim um funcionamento correto e fiável.

### ➤ 7ª Camada - Aplicação

Esta camada determina um protocolo de aplicação de um domínio específico de aplicação (HTTP, SMTP, FTP, etc.). Também é responsável pela conversão, interface gráfica, pelo envio e pelo cliente e servidor.

### Vantagens do Modelo OSI:

- Decompõe as redes de comunicações em partes mais pequenas, facilitando a deteção de erros, fazendo com que a correção dos mesmos seja também mais fácil;
- Permite que a mudança de uma camada não afete as restantes;
- Permite que haja uma maior especialização, promovendo a evolução da indústria tecnológica;

### 1.3. Modos de transmissão

Existem 3 modos de transmissão entre 2 dispositivos, simplex, half-duplex e full-duplex que definem a direção em que o sinal flui.

#### - Transmissão Simplex:

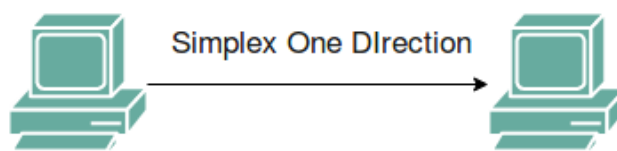


Figura 2 - Transmissão Simplex.

Na transmissão Simplex a transferência de informação é feita unidireccionalmente, ou seja, a informação enviada por um aparelho emissor é recebida por um aparelho recetor.

#### - Transmissão Half-Duplex:

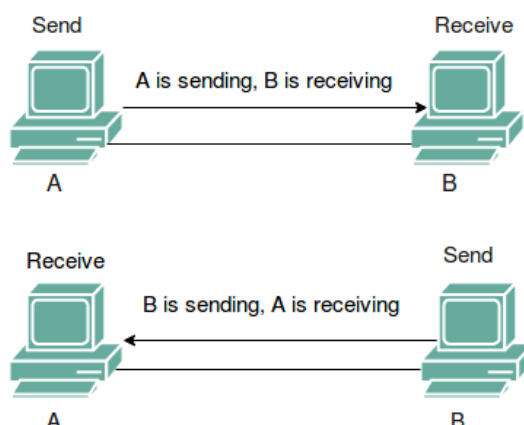


Figura 3 - Transmissão Half-Duplex.

Na transmissão Half-Duplex a comunicação é bidireccional podendo um aparelho ser tanto emissor como transmissor, no entanto a transferência de informação apenas é possível ser feita por um dispositivo de cada vez. Este modo é o que será utilizado na comunicação do presente projeto.

#### - Transmissão Full-Duplex:

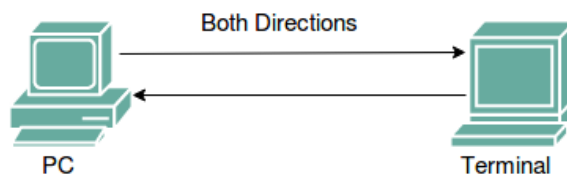


Figura 4 - Transmissão Full-Duplex.

Na transmissão Full-Duplex a comunicação é também bidireccional, mas neste caso é possível a transferência entre 2 dispositivos ao mesmo tempo.

## 1.4. Arquitetura do sistema

### 1.4.1. Arquitetura de hardware

A arquitetura do sistema representa a estrutura de componentes e as suas ligações, os modos de funcionamento e a evolução ao longo da implementação.

O sistema que iremos utilizar no projeto consiste numa comunicação bidirecional entre computadores no qual cada computador tanto funciona como recetor como emissor.

Com a utilização do *transceiver* de RF é nos permitida a transmissão de dados de modo *wireless*, neste caso através de radiofrequências. Para tal será necessária a utilização de dois dispositivos da mesma espécie (*transceivers*, RF nRF24L01+), qualquer um deles terá de realizar o papel de emissor ou recetor.

As ligações feitas entre os componentes serão feitas de forma física entre os PCs e os microcontroladores e de forma *wireless* entre os *transceivers*.

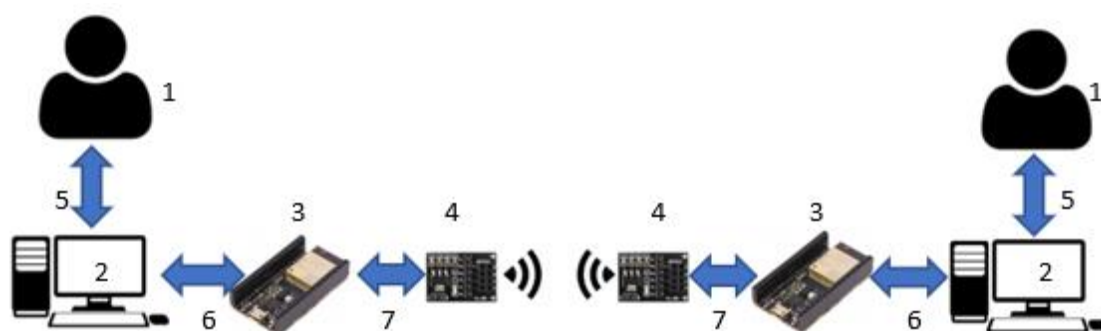


Figura 5 - Arquitetura do Sistema.

Legenda da figura 2:

1. Utilizador
2. PCs
3. Placa ESP32
4. *transceivers* RF (nRF24L01+)
5. Interface
6. Micro USB
7. Ligação microcontrolador-*transceiver*

#### 1.4.2. Arquitetura de *software*

Como vemos na imagem acima as ligações e conexões entre componentes são bidirecionais que só serão possíveis de alcançar através de uma aplicação.

A aplicação para além de enviar texto entre os computadores poderá enviar qualquer tipo de ficheiro (PDF, JPEG, etc...). Para que seja de uso agradável também deverá ter uma interface gráfica de fácil compreensão.

Teremos também de realizar um módulo que terá a função de converter os ficheiros para que posteriormente possam ser enviados através de um novo módulo de envio.

De reforçar que a aplicação também tem o papel de fazer que cada computador funcione como servidor e como cliente, baseado na arquitetura de redes *peer-to-peer*. Esta arquitetura permite partilha de dados sem necessidade de um servidor central e pode ser usada para a transferência de ficheiros em vários formatos.

## 1.5. Identificação das tecnologias e recursos

Em qualquer projeto são necessárias tecnologias que nos facilitem o trabalho e ao mesmo tempo aumentem a produtividade. Para este trabalho teremos de recorrer a diversos recursos e tecnologias, tanto ao nível *hardware* e *software*, de modo a cumprir todos os objetivos.

### 1.5.1. Recursos ao nível do *hardware*

Em relação ao *hardware*, utilizaremos as seguintes ferramentas:

*Tabela 1 - Recursos ao nível do hardware.*

FERRAMENTA	QUANTIDADE	UTILIZAÇÃO
Computador Portátil	2	Desenvolvimento de código, elaboração do relatório, pesquisa e testes
Placa ESP32	2	Comunicação com os módulos <i>transceivers</i> de RF
<i>Transceiver</i> de RF	2	Comunicação <i>wireless</i> entre ambos os módulos e transmissão e receção de dados
Cabos Micro-USB	2	Ligações físicas entre os PCs e as placas ESP32
Fios de ligação	Incerto	Efetuar as ligações entre as placas ESP32

### 1.5.2. Recursos ao nível do *software*

No que toca ao *software* iremos utilizar um vasto leque de aplicações que nos vão ajudar no desenvolvimento do projeto. Na tabela 2 iremos realçar as mais importantes, sendo que esta poderá sofrer alterações no futuro, uma vez que ainda nos encontramos numa fase inicial do trabalho.

Tabela 2 - Recursos ao nível do *software*.

LOGO	DESCRIÇÃO	UTILIZAÇÃO
	Microsoft Word	Realização e sincronização do relatório do projeto
	GanttProject	Planeamento do projeto através de diagramas de Gantt
	Arduino IDE	Software para a conexão com a placa ESP32 e comunicação com o mesmo
	IntelliJ IDEA	Desenvolvimento da aplicação em linguagem JAVA
	GitHub	Sincronização do código da aplicação
	Visual Paradigm	Implementação de fluxogramas para a elaboração da aplicação e do projeto
	Google Drive	Sincronização de ficheiros essenciais à realização do trabalho
	Discord	Comunicação entre o grupo

## 1.6. Discussão sobre as competências necessárias

No final de uma leitura e análise do enunciado do projeto concluímos que precisamos de alguns conhecimentos já adquiridos em algumas unidades curriculares, com especial relevância em Eletrónica I e II, Métodos de programação I e Paradigmas de Programação I e II. Iremos também precisar de competências que serão lecionadas durante este semestre nas unidades curriculares de Redes de Computadores I e Sistemas Operativos.

Durante a realização do projeto teremos de conhecer e entender:

- Modelo OSI, em específico a primeira camada (Camada física), segunda camada (Camada de ligação de dados) e a sétima camada (Camada de aplicação);
- As interfaces SPI e UART/RS-232;
- Redes locais sem fios por radiofrequência (RF);
- Programação e configuração do Arduino;
- Módulos *transceivers* de RF NRF24L01+;

A organização e distribuição das tarefas pelos elementos do grupo, bem como o trabalho de grupo são competências que não podem faltar para um bom resultado na elaboração do projeto de forma eficaz e positiva.

## 1.7. Planeamento de Tarefas

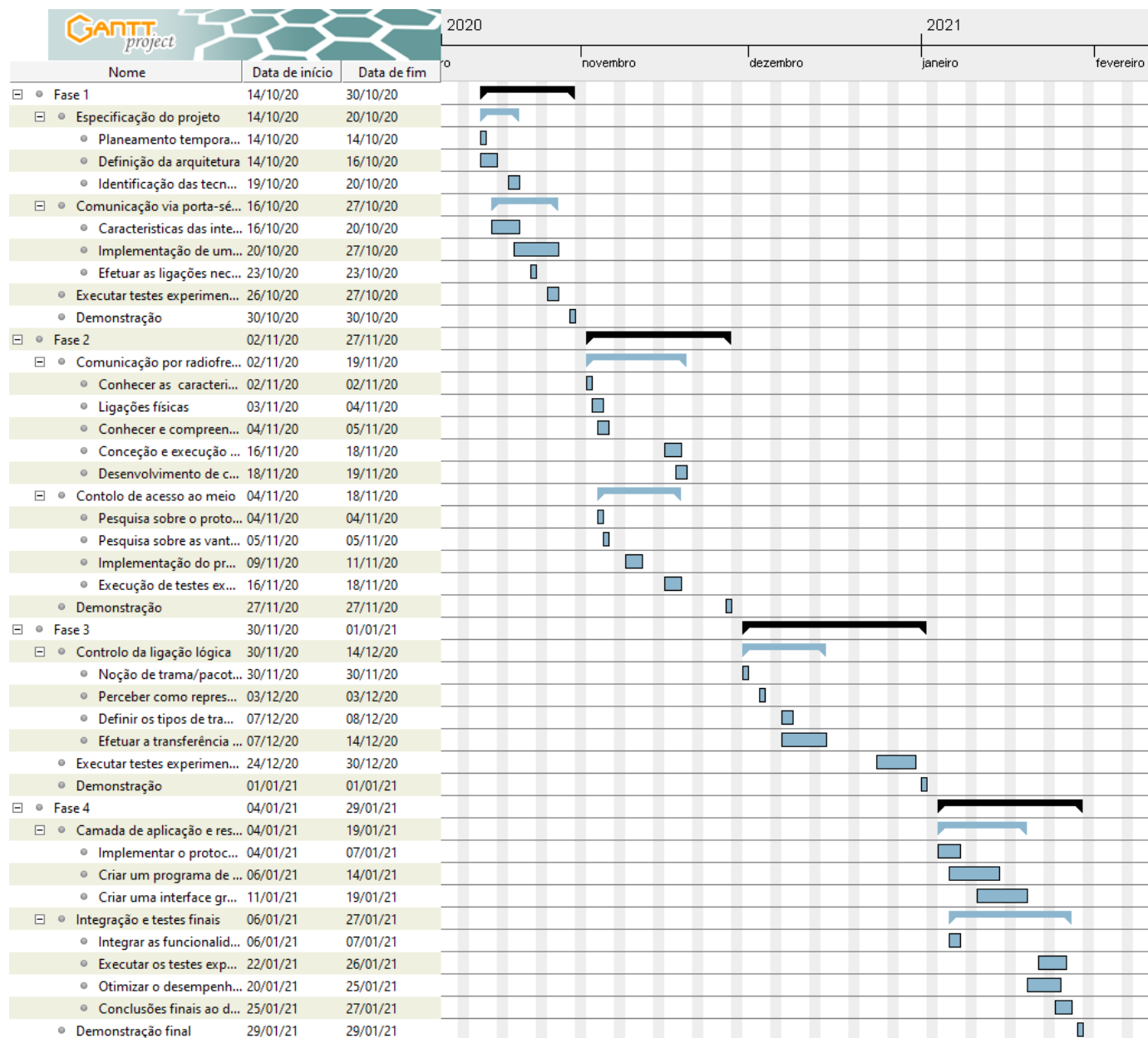


Figura 6 - Diagrama de Gantt.



## 2. Comunicação via porta série PC-Arduino-Arduino-PC

### 2.1. Fundamentos

Este projeto utiliza a interface UART, uma vez que permite a comunicação entre as duas placas ESP32 através da ligação do pino Rx de uma ao pino Tx da outra e o pino Tx de uma ao Rx da outra (Figura 7). Com a interface UART, ao enviarmos dados através de um computador, este transforma esta mensagem numa sequência de bits que vai ser lido posteriormente pelo computador recetor.

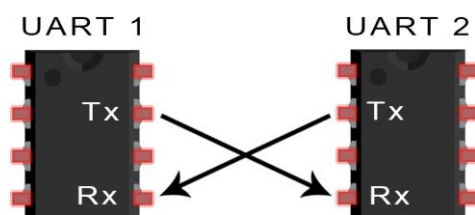


Figura 7 - Comunicação UART.

Como estamos a utilizar uma transmissão assíncrona, pois esta facilita a comunicação entre dois computadores visto que estes não necessitam de estar sincronizados, é necessário que existam bits de controlo na trama.

Neste projeto, a trama (Figura 8) é constituída por 8 bits, ou 1 byte, de dados, o start bit que é sempre 0, um stop bit e 0 bits de paridade previamente declarados no código em Java (Figura 9).

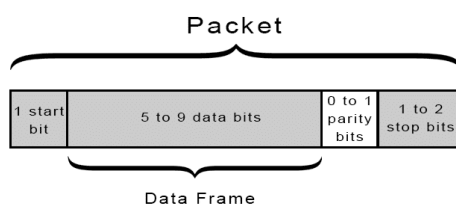


Figura 8 - Constituição de uma trama.

```
port.setComPortParameters( newBaudRate: 9600, newDataBits: 8, newStopBits: 1, newParity: 0 );
```

Figura 9 - Configuração dos parâmetros da porta série.

## 2.2. Desenvolvimento

### 2.2.1. Descrição do código em JAVA

Para desenvolver o programa que permitisse a comunicação série entre os dois computadores através da porta série decidimos programar o mesmo na linguagem java, uma vez que já estamos habituados a programar na mesma e esta ser de alto. Para desenvolver o programa que permitisse a comunicação série entre os dois computadores através da porta série decidimos programar o mesmo na linguagem java, uma vez que já estamos habituados a programar na mesma e esta ser de alto nível.

Para podermos aceder e configurar as portas série através do nosso programa decidimos utilizar a biblioteca jSerialComm, pois foi a melhor opção encontrada por todos os membros.

O algoritmo do programa foi relativamente simples de elaborar. Primeiro apresenta-se ao utilizador as portas série disponíveis e este seleciona a que pretende usar. Depois é apresentado ao utilizador um pequeno menu onde este decide o que deseja fazer (Figura 11).

O programa contém quatro métodos: enviarFicheiroTexto(), receberFicheiroTexto(), enviarImagem() e receberImagem(). Todos os métodos recebem como argumento a porta série escolhida pelo utilizador.

O método enviarFicheiroTexto() converte o ficheiro de texto que se pretende enviar num array de bytes e envia para a porta série.

O método receberFicheiroTexto() fica à espera de que o último byte seja transmitido, depois converte o array de bytes recebido no ficheiro texto.

Os métodos enviarImagem() e receberImagem() são iguais aos métodos enviarFicheiroTexto() e receberFicheiroTexto(), no entanto como o próprio nome refere estes enviam uma imagem e recebem uma imagem.

Para podermos realizar testes a velocidade de transmissão e comparar com o valor teórico da mesma, nos métodos enviarFicheiroTexto() e enviarImagem() no final é apresentado ao utilizador o tempo total de transmissão (Figura 10).

```
Choose Port:
1. COM3
2. COM4
3.
Port Opened

----- LTI I - FASE 1 -----
1. Send Text File
2. Receive Text File
3. Send Image File
4. Receive Image File
5. Exit

Option:
```

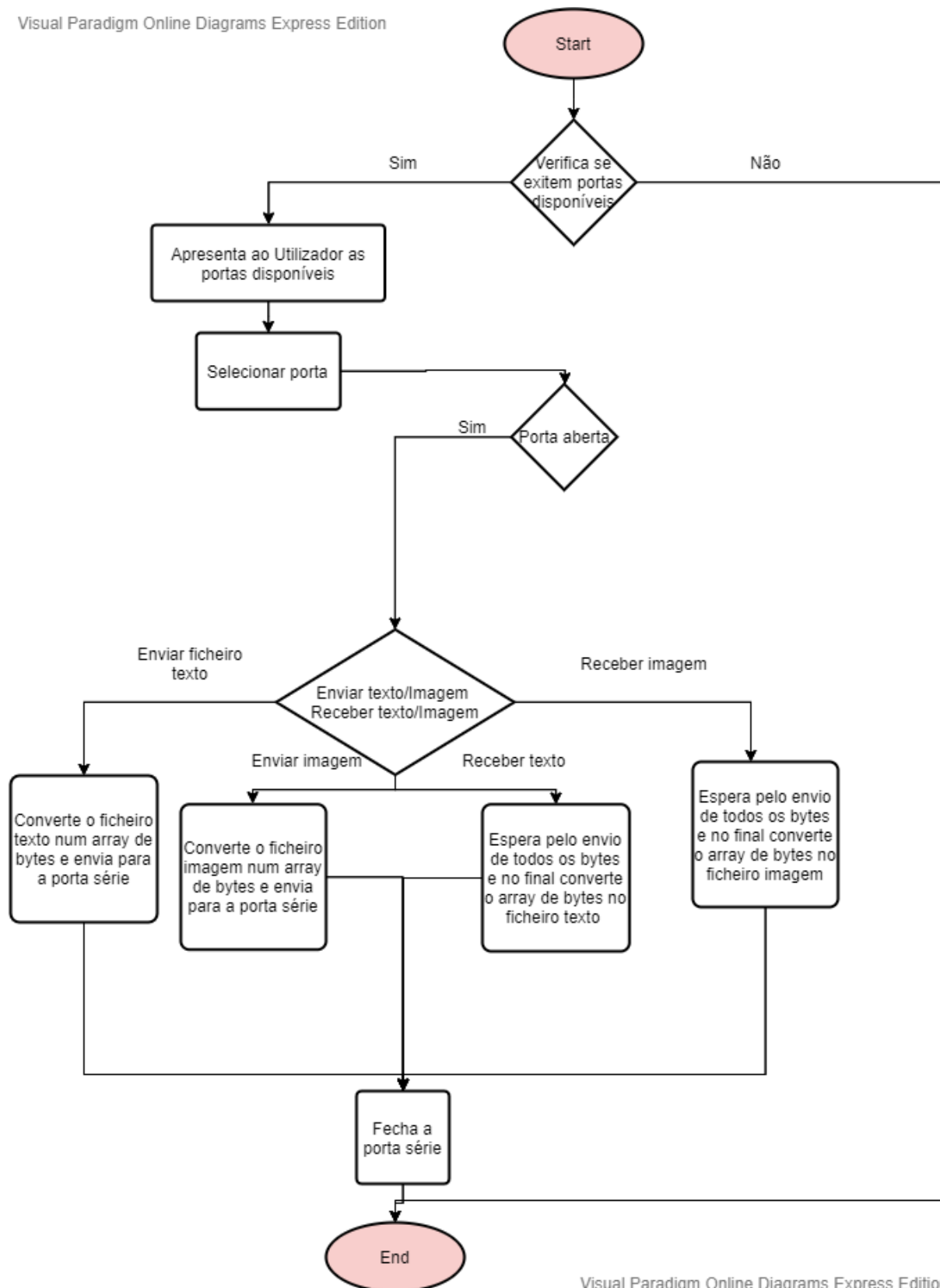
Figura 11 - Menu da aplicação.

18081ms to send file

Figura 10 - Tempo de transmissão.

### 2.2.2. Fluxograma do programa em JAVA

Visual Paradigm Online Diagrams Express Edition



Visual Paradigm Online Diagrams Express Edition

Figura 12 - Fluxograma do programa em JAVA.

## 2.3. Testes

### FICHEIRO TXT (7285 bytes)

*Tabela 3 - Comparação entre tempo teórico e experimental de um ficheiro texto.*

<i>BAUD-RATE</i> (bps)	RESULTADO TEÓRICO (s)	RESULTADO EXPERIMENTAL (s)
<b>9600</b>	6,07	6,96
<b>19200</b>	3,04	3,48
<b>57600</b>	1,012	1,165
<b>115200</b>	0,506	0,582

### FICHEIRO PNG (17894 bytes)

*Tabela 4 - Comparação entre tempo teórico e experimental de um ficheiro imagem.*

<i>BAUD-RATE</i> (bps)	RESULTADO TEÓRICO (s)	RESULTADO EXPERIMENTAL (s)
<b>9600</b>	14,91	18,08
<b>19200</b>	7,46	9,04
<b>57600</b>	2,49	3,02
<b>115200</b>	1,24	1,51

## 2.4. Discussão de resultados

Após a realização dos testes referidos no ponto anterior podemos retirar algumas conclusões.

O tempo teórico de transmissão de um ficheiro entre dois computadores através da porta série é calculado segundo a seguinte expressão:  $T_{teórico} = \frac{L}{R}$ , onde L é o tamanho do ficheiro em bits e o R é o baudrate(bits/s).

O tempo de transmissão experimental foi medido através do uso do método `currentTimeMillis()` no programa desenvolvido em JAVA, sendo este apresentado ao utilizador depois do ficheiro ser enviado.

Usamos diferentes baudrates para aumentar a velocidade de transmissão e verificar se o ficheiro (texto e imagem) mesmo com velocidades maiores de transmissão chegava completo ao computador e não corrompido.

Verificamos que em todos os baud-rates tabelados os ficheiros foram transferidos com sucesso e quanto maior o baud-rate menor o tempo de transmissão, no entanto quando aumentávamos significativamente o baud-rate (na ordem das centenas de milhares) ocorriam erros que eram expectáveis devido à elevada velocidade de transferência.

Depois de comparados os tempos teóricos e experimentais verificamos que os tempos experimentais eram superiores aos teóricos com era de esperar visto que o baud-rate era menor experimentalmente e teoricamente não eram levados em conta fatores como por exemplo o ruído, ou até a temperatura que são motivos de discrepância nos resultados.

Durante os testes deparamo-nos com um problema do qual não estávamos à espera, quando tentávamos enviar um ficheiro com tamanho superior a 65536 bytes o ficheiro chegava corrompido ao computador. Depois de alguns testes feitos e procura na internet chegamos à origem do problema, uma função em JAVA não pode exceder 65536 bytes. Nesta fase não procuramos ir à procura de uma solução para isto, tentaremos fazer isso numa próxima fase.

## Conclusão

Podemos concluir que a primeira fase do projeto serviu como planeamento e organização geral do grupo assim como implementação de um programa para a transferência de ficheiros. Foi importante a boa compreensão do projeto para que a sua planificação fosse eficaz e cuidada, de forma a que as tarefas fossem bem distribuídas. A nossa organização permitiu-nos um trabalho mais organizado e mais completo.

Para além disso, a elaboração do projeto fez nos entender a importância de cadeiras de anos anteriores e ano atual para a realização do projeto desta UC, destacando a importância de Redes de Computadores (3ºano), Paradigmas de Programação(2ºano) e Eletrónica (2ºano).

Conseguimos também entender o funcionamento da placa ESP32 que será utilizada para o restante projeto. Com o conhecimento do seu funcionamento a sua programação foi mais fácil e correta. Desta forma conseguiremos entrar na próxima fase com o conhecimento necessário para a sua realização.

Numa fase final, entendemos a importância do trabalho constante, pois sem esse trabalho constante seria impossível a realização do projeto. Apesar das dificuldades que nos foram aparecendo ao longo do tempo conseguimos sempre superar com o trabalho de equipa.

## Referências

- [1] Will Hedgecock, “What is jSerialComm?”, <https://fazecast.github.io/jSerialComm/> (22 de outubro de 2020).
- [2] Leonardo Oliveira, “Vantagens do modelo OSI e articulação entre camadas”, <https://tozehgamer.wordpress.com/vantagens-do-modelo-osi-e-articulacao-entre-camadas/> (23 de outubro de 2020).
- [3] “O que é o modelo OSI?” <https://canaltech.com.br/produtos/o-que-e-modelo-osi/> (23 de outubro de 2020).
- [4] Maria João Nicolau, Slides da unidade curricular de Redes de computadores I.
- [5] Pedro Pinto, “Redes”, <https://pplware.sapo.pt/tutoriais/networking/redes-sabe-o-que-e-o-modelo-osi/> (23 de outubro de 2020).
- [6] Will Hedgecock, “A platform-independent serial port access library for JAVA”, <https://github.com/Fazecast/jSerialComm> (24 de outubro de 2020).
- [7] Yuri Matheus, “O modelo OSI e as suas camadas”, <https://www.alura.com.br/artigos/conhecendo-o-modelo-osi> (24 de outubro de 2020).
- [8] Lou Frenzel, “What’s the difference between bit rate and baud rate?”, <https://www.electronicdesign.com/technologies/communications/article/21802272/whats-the-difference-between-bit-rate-and-baud-rate> (27 de outubro de 2020).
- [9] “Transmission Modes In Computer Networks” <https://www.geeksforgeeks.org/transmission-modes-computer-networks/> (28 de outubro de 2020).
- [10] Scott Campbell, “Basics of UART communication”, <https://www.circuitbasics.com/basics-uart-communication/> (28 de outubro de 2020).

# Autoavaliação

## **Inês Barreira Marques:**

O meu contributo nesta fase foi pesquisa de informação para a realização do relatório, na análise de camadas, orientação do projeto e contribui nas ligações do ESP32.

## **José Pedro Fernandes Peleja:**

Contribui na arquitetura do projeto, na pesquisa de informação, elaboração de relatório, na análise de camadas, distribuição de tarefas, organização e contribui nas ligações do ESP32.

## **Rui Filipe Ribeiro Freitas:**

Nesta primeira fase do projeto ajudei na elaboração do código JAVA e na realização dos testes necessários assim como na pesquisa e escrita do relatório.

## **Tiago João Pereira Ferreira:**

Contribui na elaboração do código JAVA e do relatório. Estive também presente na realização dos testes.