

Grupo 3:

- Leandro Henrique Dantas Alves A82157
- André Martins Almeida A82211
- José Eduardo da Silva Santos A82350

# Aplicação de Conversação

---

PROJETO DE LABORATÓRIOS DE TELECOMUNICAÇÕES E  
INFORMÁTICA I

# Tópicos

---

- Fases do projeto
- Arquitetura e funcionalidades – modelo OSI
- Tecnologias e recursos humanos
- Competências necessárias
- Ligações físicas
- Interfaces de comunicação série
- Protocolos de controlo de acesso ao meio (MAC)
  - ALOHA
  - CSMA
  - *Polling*
- Controlo da ligação de dados
  - *Stop-and-Wait*
  - CRC
  - Estrutura das tramas
- Estrutura do *payload*
- Testes e resultados:
  - Fase 1
  - Fase 2
  - Fase 3
  - Fase 4
- Conclusões

# Fases do projeto

---

## FASE 1

1. Especificação do projeto
2. Comunicação via porta série PC-Arduino-Arduino-PC

## FASE 2

1. Comunicação por radiofrequência Arduino-TRF-TRF-Arduino
2. Controlo de acesso ao meio

## FASE 3

1. Controlo da ligação lógica

## FASE 4

1. Camada de aplicação e respetiva aplicação
2. Integração e testes finais

# Arquitetura e funcionalidades – modelo OSI

---

## CAMADA FÍSICA (NÍVEL 1):

Processamento dos dados recebidos/enviados entre componentes físicos.

---

## CAMADA DE LIGAÇÃO DE DADOS (NÍVEL 2):

Controlo da ligação lógica (LLC) e de acesso ao meio (MAC).

---

## CAMADA DE APLICAÇÃO (NÍVEL 7):

Construção e interpretação de mensagens e ficheiros recebidos/enviados.

## Tecnologias e recursos humanos

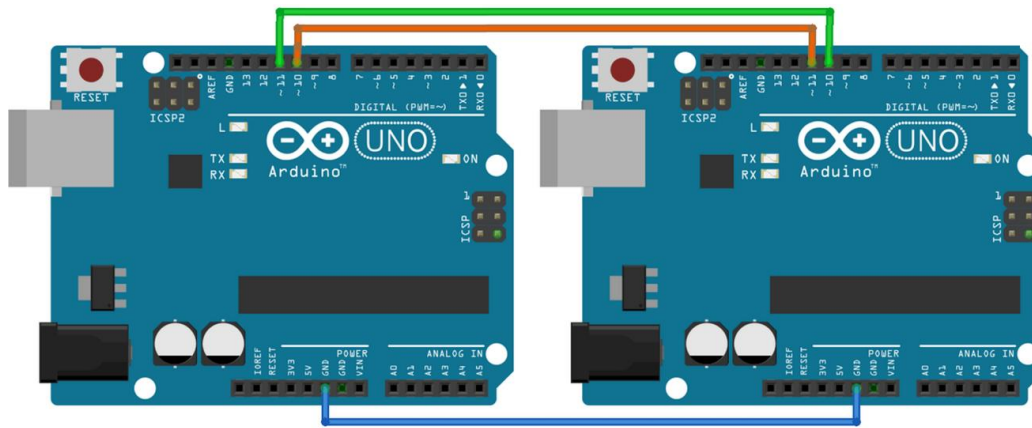
- Grupo de 3 membros com trabalho em equipa e autónomo dentro e fora das aulas da UC;
- 3 computadores (um por cada membro do grupo);
- 3 módulos Arduino Uno;
- 3 *transceivers* nRF24L01+;
- Arduino Software IDE, Eclipse, NetBeans e IntelliJ;
- Osciloscópio digital.

## Competências necessárias

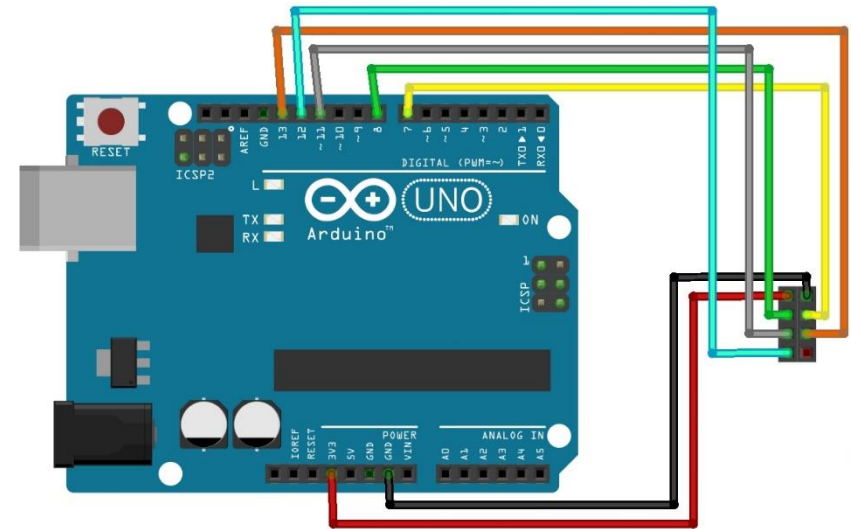
- Modelo OSI e funcionalidades associadas a cada camada;
- Funcionalidades oferecidas pelo módulo RF;
- Redes locais sem fios (WLAN);
- Protocolos de aplicação específicos;
- Programação estruturada em Java através de APIs;
- Interfaces de comunicação série (SPI e UART/RS-232);
- Trabalho em equipa e autónomo.

# Ligações físicas

## FASE 1



## REstantes FASES



# Interfaces de comunicação série

---

## PC-ARDUINO

### PARÂMETRO D-P-S:

- 8-N-1 – 8 bits de dados, sem bit de paridade e *stop bit* com valor 1.

### RS-232:

- bit 0 de 3 a 15 V e bit 1 de -15 a -3 V.

### UART:

- bit 0 equivale a 0 V e bit 1 equivale a 5 V.

## ARDUINO-TRF

### SPI:

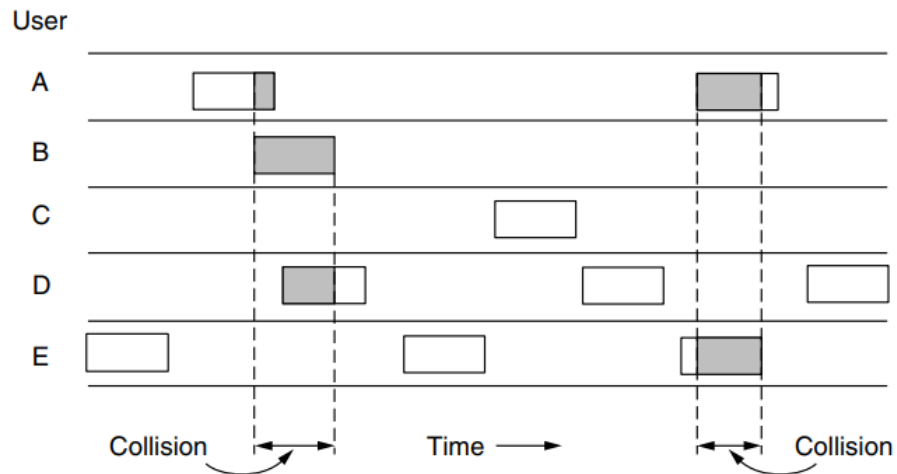
- Interface síncrona *full duplex*;
- *Master* – Arduino;
- *Slave* – TRF



# Protocolos de controlo de acesso ao meio (MAC)

## ALOHA

Quando o transmissor tem dados para enviar, estes são enviados imediatamente.



## CSMA

Quando o transmissor tem dados para enviar, este verifica se o canal está ocupado por outro transmissor.

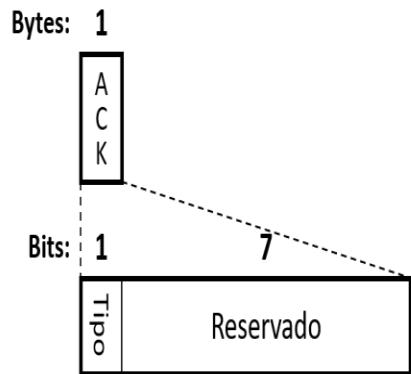
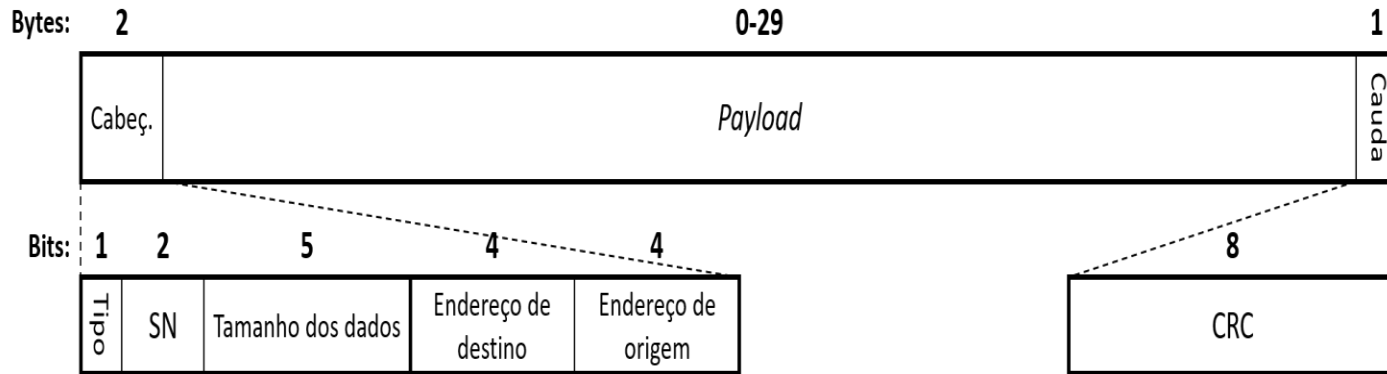
- Se estiver livre – os dados são enviados;
- Se estiver ocupado – o transmissor espera um tempo aleatório para voltar a verificar.

## POLLING

Quando o recetor acabar de receber os dados de um transmissor, é dada a vez ao próximo transmissor para enviar os seus dados.



# Estrutura das tramas



- **Cabeç.** – cabeçalho do pacote;
- ***Payload*** – dados do pacote;
- **Cauda** – cauda do pacote;
- **Tipo** – tipo do pacote  
(0 – mensagem, 1 – ACK);
- **SN** – número de sequência do pacote  
(0 - 3);
- **Tamanho dos dados** – tamanho dos dados  
a enviar (0 - 32 bytes);
- **Endereço de destino** – endereço do  
receptor;
- **Endereço de origem** – endereço do  
transmissor;
- **CRC** – FCS;
- **ACK** – trama de confirmação;
- **Reservado** – bits não usados.

# Estrutura do *payload*

---

## Mensagens:

- 2[]c  
tamanho[]texto
- Exemplo:
  - 2[]c  
15[]Olá, tudo bem?

## Ficheiros:

- 2[]i  
tamanho[nome]conteúdo
- Exemplo:
  - 2[]i  
19[ficheiro.txt]Eu sou um ficheiro.

- **2** – tamanho de “c\n” e “i\n”;
- **c** – identificador de mensagem;
- **tamanho** – tamanho do conteúdo da mensagem/ficheiro;
- **texto** – texto da mensagem;
- **i** – identificador de ficheiro;
- **nome** – nome do ficheiro;
- **conteúdo** – conteúdo do ficheiro.

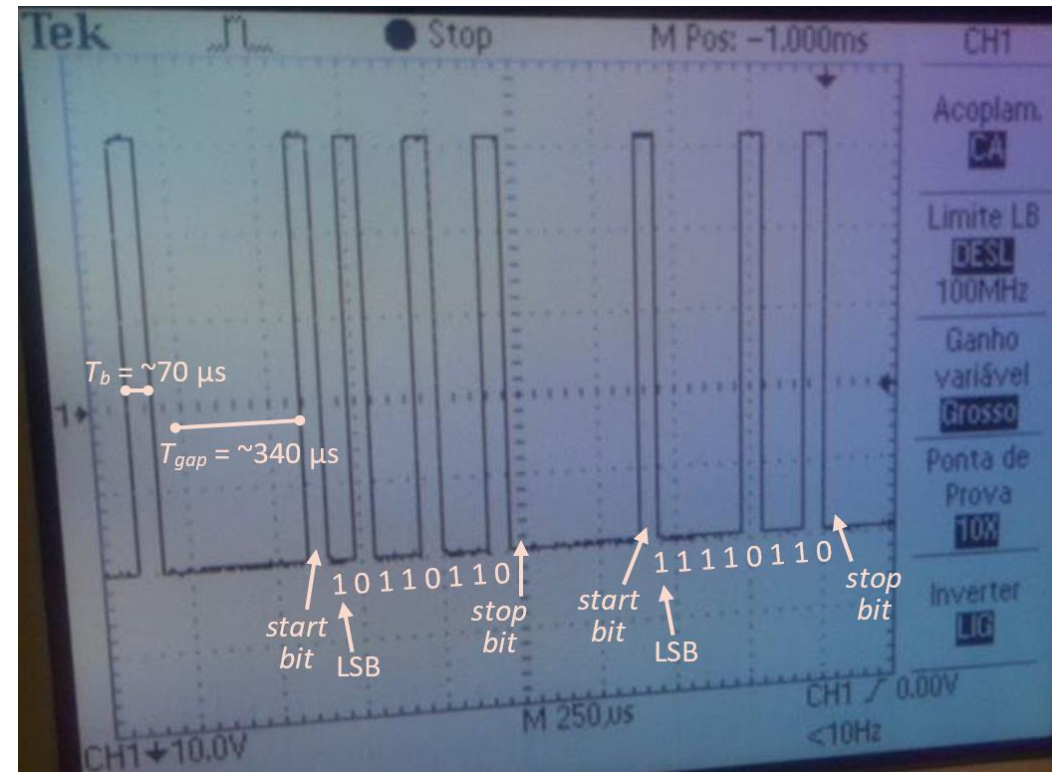
# Testes e resultados – Fase 1

Testes teóricos:

- $R = 14\,400\text{ bps}$
- $T_b \approx 69,444\text{ }\mu\text{s}$

Testes experimentais:

- Dados: “mo” (01101101 01101111)
- $T_b \approx 70\text{ }\mu\text{s}$
- $T_{\text{gap}} \approx 340\text{ }\mu\text{s}$



# Testes e resultados – Fase 2

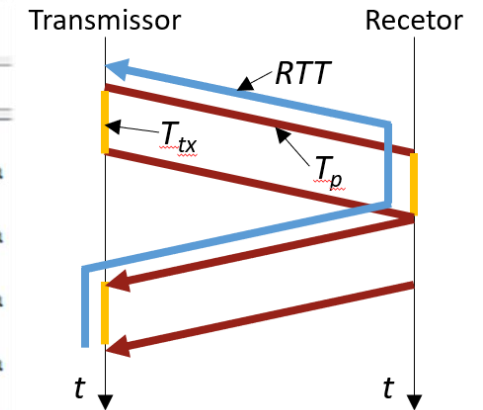
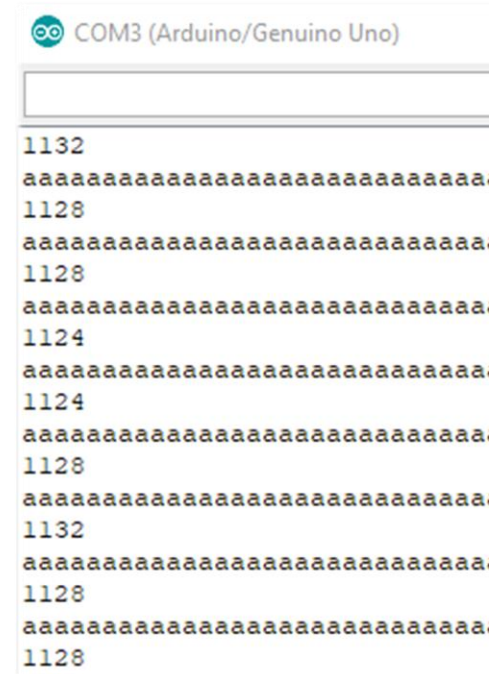
## TRANSMISSÃO E RECEÇÃO DE 32 BYTES ENTRE DOIS UTILIZADORES

### Testes teóricos:

- $L = 32$  bytes,  $R = 2$  Mbps,  $d = 1$  m
- $T_{tx} = 128 \mu s$
- $T_p \approx 3,336$  ns
- $RTT \approx 256,007 \mu s$

### Testes experimentais:

- $RTT \in \{1124, 1132\} \mu s$



# Testes e resultados – Fase 2

TRANSMISSÃO SIMULTÂNEA COM DOIS  
TRANSMISSORES E UM RECETOR COM ALOHA

CSMA

COM10 (Arduino/Genuino Uno)

```
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaa?????"??
User 1: aaaaaaaa`_???U*???W???J?□□
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
```

COM4 (Arduino/Genuino Uno)

```
Message sent!
Carrier: 1
Busy...
Carrier: 0
Message sent!
Carrier: 0
Message sent!
Carrier: 0
Message sent!
Carrier: 0
Message sent!
Carrier: 1
Busy...
Carrier: 0
Message sent!
Carrier: 0
Message sent!
Carrier: 0
Message sent!
```

COM10 (Arduino/Genuino Uno)

```
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
```

# Testes e resultados – Fase 2

## POLLING

```
COM3 (Arduino/Genuino Uno)
|
User 0: 2
User 0: 1
Message Sent!
User 0: 2
User 0: 1
Message Sent!
User 0: 2
User 0: 1
Message Sent!
User 0: 2
User 0: 1
Message Sent!
```

```
COM4 (Arduino/Genuino Uno)
|
User 0: 1
User 0: 2
Message sent!
User 0: 1
User 0: 2
Message sent!
User 0: 1
User 0: 2
Message sent!
User 0: 1
User 0: 2
Message sent!
```

```
COM10 (Arduino/Genuino Uno)
|
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
User 1: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
User 2: bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
```



# Testes e resultados – Fase 3

## Testes teóricos:

- $L_{\text{pacote}} = 32$  bytes,  $L_{\text{payload}} = 29$  bytes,  $L_{\text{ACK}} = 1$  byte
- $R = 250$  kbps,  $d = 1$  m
- $T_{tx \text{ pacote}} = 1$  ms,  $T_{tx \text{ ACK}} = 31,25$   $\mu$ s
- $T_p \approx 3,336$  ns
- $RTT = 1,031$  ms,  $U \approx 96,969\%$
- $S_t \approx 242,483$  Kbps,  $S_q \approx 219,750$  Kbps

## Testes experimentais:

- $RTT \approx 5,6$  ms
- $S_t = 45$  Kbps,  $S_q \approx 40,781$  Kbps

SN Utilizador

```
0-0: Somebody once told me the wor | Acknowledged (5 ms)
1-0: ld is gonna roll me | Resending | Acknowledged (6 ms).
0-1: I ain't the sharpest tool in
1-1: the shed
2-0: She was looking kind of dumb | Resending | Acknowledged (5 ms).
3-0: with her finger and her thumb | Resending | Acknowledged (6 ms).
0-0: | Acknowledged (8 ms).
2-1: In the shape of an "L" on her
3-1: forehead
1-0: Well the years start coming a | Resending | Acknowledged (5 ms).
2-0: nd they don't stop coming | Resending | Acknowledged (5 ms).
0-1: Fed to the rules and I hit th | Error!
0-1: Fed to the rules and I hit th
1-1: e ground running
3-0: Didn't make sense not to live | Acknowledged (5 ms).
0-0: for fun | Acknowledged (5 ms).
```

Confirmação do utilizador 1

RTT

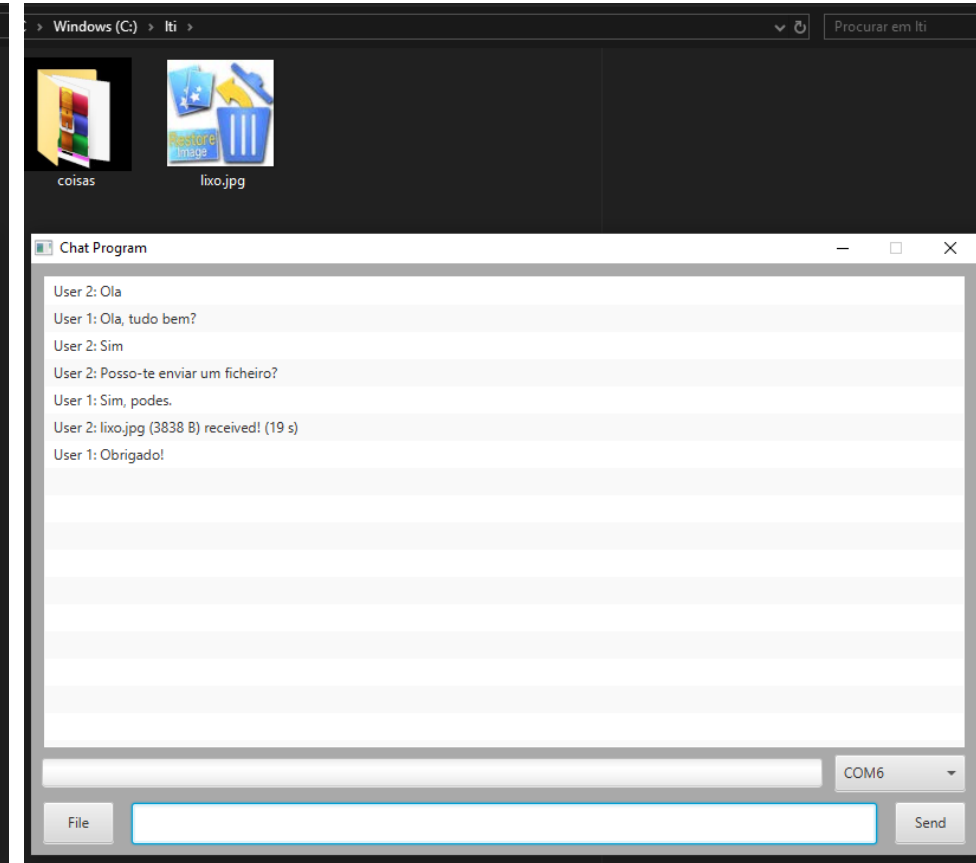
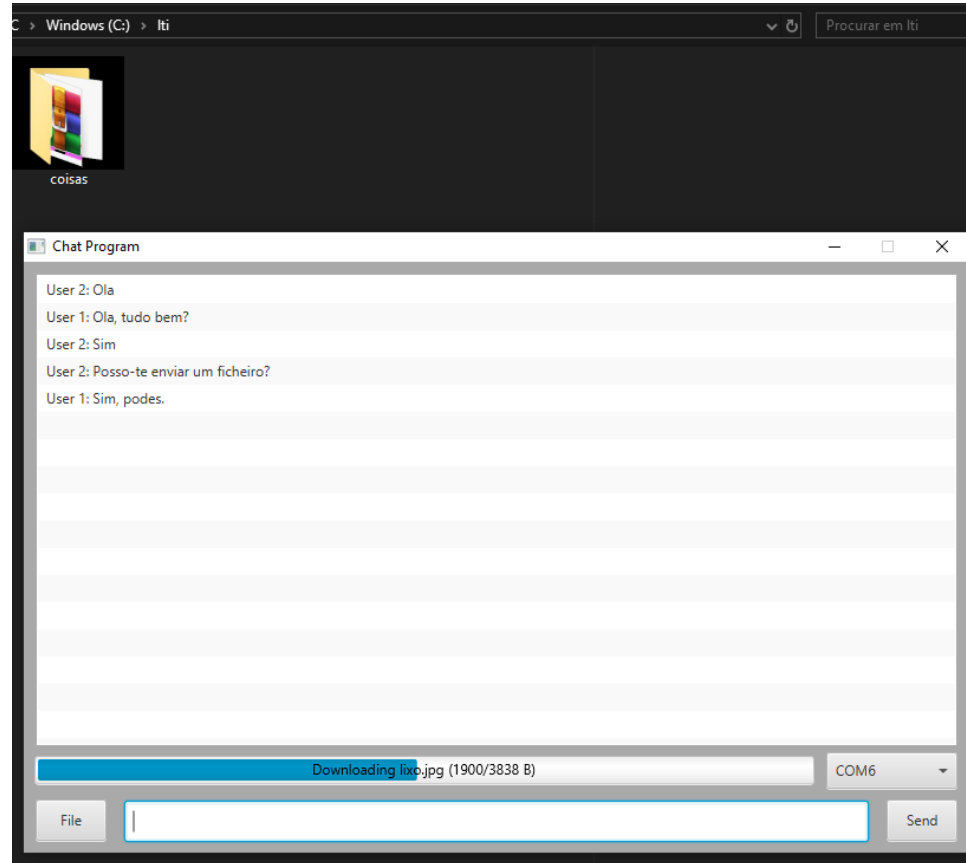
Reenvio da mensagem devido à falta de confirmação do utilizador 1 dentro do timeout

Erro recebido

Reenvio da mensagem corrigida pelo utilizador 1

# Testes e resultados – Fase 4

- $L_{\text{pacote}} = 29$  bytes
- $T_{\text{atraso/pacote}} = 150$  ms
- $L = 3\,838$  B
- $T_{\text{total teórico}} \approx 19,852$  s
- $T_{\text{total prático}} \approx 19,874$  s



# Conclusões

---

Este projeto ajudou-nos a perceber grande parte do objetivo principal deste curso, que é de desenvolver e perceber como é realizada e controlada a comunicação entre dispositivos e perceber a sua complexidade.

Também conseguimos aplicar e melhorar os nossos conhecimentos de outras UCs do curso, principalmente Redes de Computadores I.

Conseguimos desenvolver um espírito profissional pois o tema do projeto é algo útil e que foi desenvolvido em equipa e arduamente.

Ao longo destas fases tivemos bastantes dificuldades em alguns aspetos mas achamos que conseguimos superá-las com sucesso.