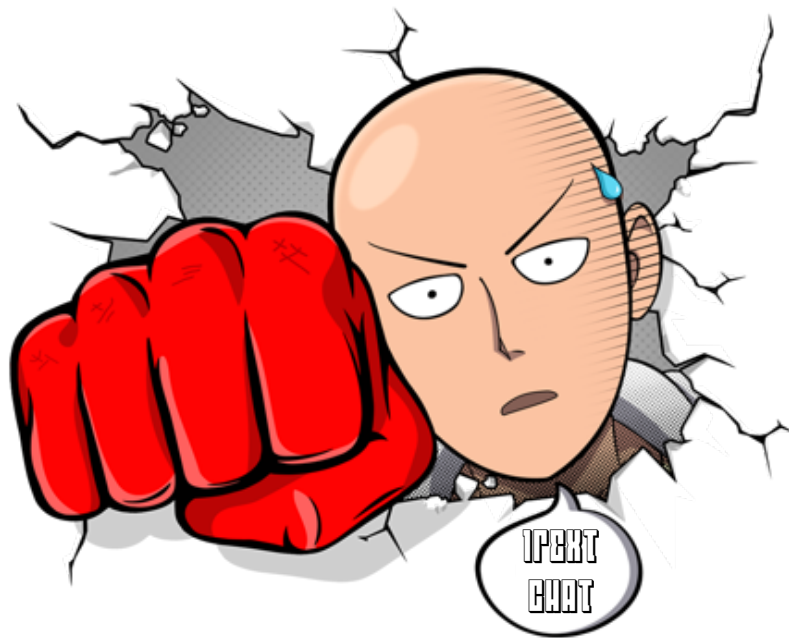


# Sistemas de Comunicação com Equipas

Relatório do Trabalho Prático



Curso: Engenharia Informática

Disciplina: Redes de Computadores

Docentes: Pedro Patinho, Pedro Salgueiro

Entregue Junho 2022

Alunos:

Joana Carrasqueira nº48566

João Condeço nº48976

# Índice

1.	Objetivos do Programa .....	2
2.	Protocolo de Comunicação .....	3
a.	Identificação .....	3
b.	Comunicação escrita (“chats” e “posts”) .....	4
c.	Transmissão de ficheiros .....	5
d.	Mensagens do servidor .....	5
3.	Tags de Equipa .....	6
4.	Posts Multilinha .....	7
5.	Utilizadores Online .....	7
6.	Exemplos Concretos .....	8
7.	Dificuldades e Complicações Encontradas .....	13

# 1. Objetivos do Programa

Este projeto foi elaborado no âmbito da cadeira de Redes de Computadores da licenciatura Engenharia Informática da Universidade de Évora. O objetivo passa por elaborar um sistema de comunicação para equipas de trabalho, com possibilidade de “chat”, “posts” e transmissão de ficheiros.

O sistema assenta num protocolo de comunicação bem definido, de forma a possibilitar a interação entre clientes e servidores, independentemente da linguagem de programação em que são implementados. Foi ainda acrescentado ao trabalho diversas extensões, nomeadamente Tags de equipa (Cargo admin nas Tags), Posts multilinha, Utilizadores online, Adicionar e remover os utilizadores às tags.

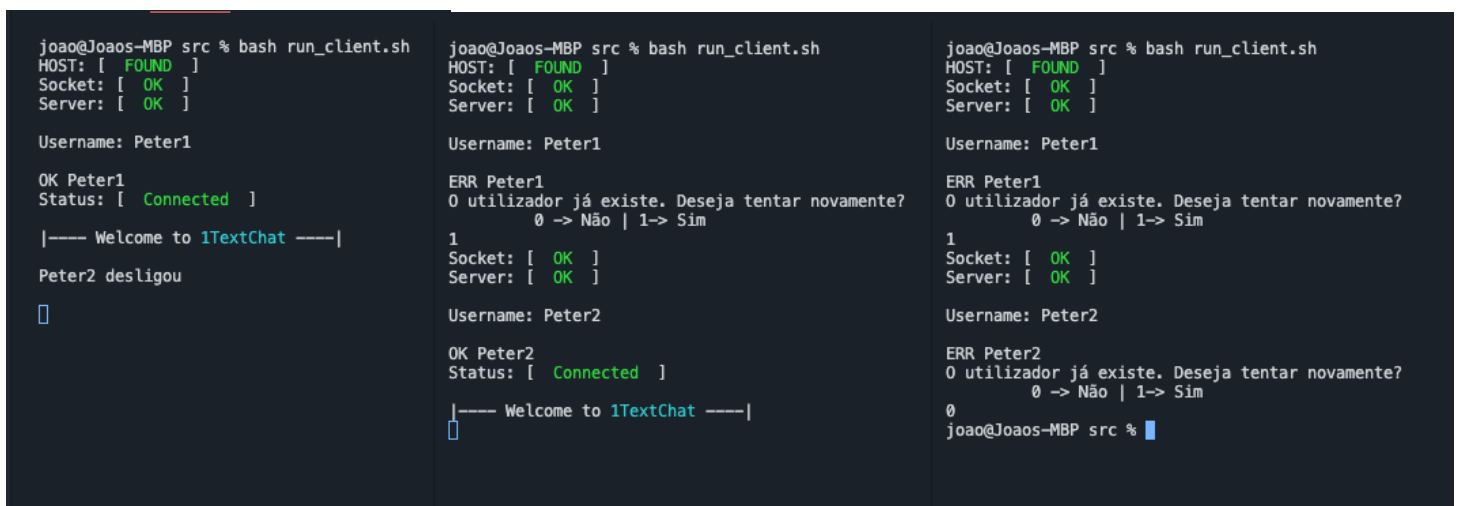
## 2. Protocolo de Comunicação

### a. Identificação

Após a ligação, o cliente identifica-se perante o servidor, enviando um “username” pelo qual será conhecido durante a sua ligação. O servidor deverá enviar para o cliente:

- OK <username> \n – caso o username seja aceite pelo servidor
- ERR <username> \n – caso o username não seja aceite pelo servidor

Um pormenor que foi acrescentado da nossa parte, foi o facto de o cliente, em vez de terminar o programa quando o username já existe, avisa o utilizador desta ocorrência e pergunta ao mesmo se deseja continuar. Aqui, o cliente deve de responder 0 caso não queira continuar ou 1 caso contrário.



The figure consists of three terminal screenshots side-by-side, showing the execution of a client program (run\_client.sh) on a Mac (joao@Joaos-MBP src %). Each screenshot shows a different scenario of user identification and connection status.

```
joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Peter1

OK Peter1
Status: [ Connected ]

|---- Welcome to 1TextChat ----|
Peter2 desligou
█
```

```
joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Peter1

ERR Peter1
0 utilizador já existe. Deseja tentar novamente?
0 -> Não | 1-> Sim
1
Socket: [ OK ]
Server: [ OK ]

Username: Peter2

OK Peter2
Status: [ Connected ]

|---- Welcome to 1TextChat ----|
█
```

```
joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Peter1

ERR Peter1
0 utilizador já existe. Deseja tentar novamente?
0 -> Não | 1-> Sim
1
Socket: [ OK ]
Server: [ OK ]

Username: Peter2

ERR Peter2
0 utilizador já existe. Deseja tentar novamente?
0 -> Não | 1-> Sim
0
joao@Joaos-MBP src % █
```

Fig 1 – Exemplos da Identificação do cliente

b. Comunicação escrita (“chats” e “posts”)

A comunicação entre clientes pode ser feita em forma de conversa de um-para-um ou de um-para-todos (“chat”) ou na forma de “posts” permanentes, que podem ser consultados mais tarde. Cada mensagem usa também uma TAG, que poderá ser GLOBAL (ou seja, todos os utilizadores pertencem à mesma), USER <username> (dirigida apenas para o utilizador escolhido) ou TAG <tagName> (que será abordado em mais detalhe no ponto Tags de Equipa).

- MSG <tagName> <message>\n - Envia a mensagem <msg> para a tag <tagName>
- POST <tagName> <message>\n - Envia um post com a mensagem <msg> para a tag <tagName> (não é possível enviar “posts” para utilizadores, já que as mensagens privadas são temporárias)
- READ <tagName>\n - Pede ao servidor a lista de “posts” já enviados para a tag <tagName>
- OK MSG <tagName>\n - O servidor aceitou a mensagem enviada
- OK POST <tagName>\n - O servidor aceitou a mensagem
- ERR POST <tagName>\n - O servidor não aceitou a mensagem (post para uma tag inválida, etc)
- OK ALLPOSTS <tagName> N \n  
POST 1  
conteúdo do primeiro post  
-autor  
  
POST 2  
conteúdo do segundo post  
-autor  
.  
.  
.  
POST N  
conteúdo do terceiro post

- ERR ALLPOSTS <tagName>\n - Não é possível mostrar a lista de "posts" (tag inválida, etc.)

Foi tomada a decisão de implementar os posts multilinha. Como tal a primeira linha da mensagem serve apenas para entrar no modo escrita, e as seguintes contem o conteúdo, na última linha consta o nome do autor.

### c. Transmissão de ficheiros

A transmissão de ficheiros pode ser feita de um-para-um, assumindo que o destinatário está "online", sendo possível um utilizador ver quem está online com a extensão abordada no ponto 5. Para transmitir um ficheiro, é necessário que os clientes passem a usar um modo binário, onde as mensagens podem conter qualquer byte (incluindo o '\0') e não são terminadas com '\n', sem ter limite de tamanho de ficheiros em conta.

- FILE <tagName> <fileName> <bytes> \n - Inicia o envio do ficheiro <fileName>, para a tag <tagName>. Após esta mensagem, inicia a transmissão do conteúdo do ficheiro. O tamanho do ficheiro não é dado pelo utilizador, mas sim calculado pelo programa e acrescentado à mensagem enviada para o servidor
- ERR FILE <username> <fileName>\n - O servidor não aceitou o ficheiro, ou não foi entregue ao destinatário

### d. Mensagens do servidor

O servidor envia ainda algumas mensagens, sem a solicitação do cliente, a fim de o notificar de certos acontecimentos. Podendo estes acontecimentos ser:

- Outro utilizador acrescentar este a uma Tag;
- Outro utilizador enviar um ficheiro a este;

- Quando um utilizador sai do servidor;
- Se o username foi aceite/recusado;
- Quando um utilizador faz um post numa Tag à qual este pertence.

### 3. Tags de Equipa

Para além da Tag GLOBAL (obrigatória) foi também implementada a criação de Tags por parte dos utilizadores. Desta forma os utilizadores podem criar uma Tag onde adicionam quem quiserem e trocam mensagens e posts com estes.

A criação de uma Tag ocorre na tentativa de enviar uma mensagem ou de submeter um post para uma Tag não existente, assumindo o cliente responsável por este ato como administrador da mesma. Sendo administrador pode utilizar as funções ADD e RMV para gerir os elementos com acesso à Tag. De forma a não perder acesso ao conteúdo das Tags quando o servidor vai abaixo é registado no ficheiro afeto a esta o nome do admin e armazenado numa pasta do servidor. Assim quando o servidor retoma o seu funcionamento basta o administrador registrar-se e o acesso à Tag ser-lhe-á retornado tendo apenas de readicionar os participantes. Numa implementação futura poderia ainda ser adicionado uma password de forma a garantir a identidade do Administrador.

Relativamente aos posts cada tag guarda estes num ficheiro .txt com o título do nome da mesma de forma a não se perder o conteúdo numa falha do servidor.

Relativamente aos participantes numa tag, cada utilizador tem associado uma estrutura com o número do socket, username e uma lista onde estão guardadas as tags nas quais o mesmo está

envolvido. Sempre que este quiser interagir com uma das tags basta aceder à sua lista e verificar se a tag pretendida reside nesta.

Os comandos ADD e RMV, como dito anteriormente, permitem gerir os participantes de uma tag, atuando no seguinte formato:

- ADD <tagName> <username>
- RMV <tagName> <username>

## **4. Posts Multilinha**

Foi implementado também a possibilidade de escrever posts com mais de uma linha. De forma a garantir isto, quando recebido uma frase com o formato POST <tagName> <username> o cliente entra em modo escrita começando apenas a partir daqui a contar o conteúdo para o post. Ao longo da escrita cada linha é armazenada numa lista para mais tarde ser removido o conteúdo. Quando inserida uma frase começada por "EXIT" é então recolhido o conteúdo da lista, armazenado num array e enviado para o servidor juntamente com as informações da tag à qual este se destina. Neste momento, o servidor escreve no ficheiro afeto à tag o conteúdo do post e os participantes desta tag serão informados da ocorrência de um novo post.

## **5. Utilizadores Online**

Foi proposta e implementada uma extensão capaz de mostrar, caso solicitada, a lista de elementos conectados ao servidor. Assim facilita ao utilizador a escrita de mensagens privadas, adicionar às tags ou envio de ficheiros para os mesmos.



## 6. Exemplos Concretos

```
joao@Joao-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Anya

OK Anya
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

MSG GLOBAL A spy? An assassin? This is so exciting Wako Wako :>
-You

OK MSG GLOBAL

MSG GLOBAL Twilight reporting for duty
-Loyd

MSG GLOBAL [Thorn Princess] May I have the honor of taking your life tonight?
-Yor

[]

joao@Joao-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Yor

OK Yor
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

MSG GLOBAL A spy? An assassin? This is so exciting Wako Wako :>
-Anya

MSG GLOBAL Twilight reporting for duty
-Loyd

MSG GLOBAL [Thorn Princess] May I have the honor of taking your life tonight?
-You

OK MSG GLOBAL

[]

joao@Joao-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Loyd

OK Loyd
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

MSG GLOBAL A spy? An assassin? This is so exciting Wako Wako :>
-Anya

MSG GLOBAL Twilight reporting for duty
-You

OK MSG GLOBAL

MSG GLOBAL [Thorn Princess] May I have the honor of taking your life tonight?
-Yor

[]

joao@Joao-MBP src % bash run_server.sh
Socket: [ OK ]
Nova conexão, o FD do socket é 4 , o ip é : 127.0.0.1 e a porta : 57847
Nova conexão, o FD do socket é 5 , o ip é : 127.0.0.1 e a porta : 57848
Nova conexão, o FD do socket é 11 , o ip é : 127.0.0.1 e a porta : 57849
Anya enviou 'MSG GLOBAL A spy? An assassin? This is so exciting Wako Wako :>' para todos
Loyd enviou 'MSG GLOBAL Twilight reporting for duty' para todos
Yor enviou 'MSG GLOBAL [Thorn Princess] May I have the honor of taking your life tonight?' para todos
[]
```

Fig 2 – Exemplo que demonstra o funcionamento de MSG na Tag Global e as devidas mensagens do servidor

```
joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]
```

Username: Steven

OK Steven  
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

POST GLOBAL  
My name is Steven with a V,  
Sometimes Mr. Knight  
Sometimes an idiot  
EXIT

OK POST

Recebeu um novo post na tag: GLOBAL

MSG GLOBAL \*kills you in Spanish\*  
~Jake

MSG Khonshu leave us alone  
~You

█

```
joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]
```

Username: Khonshu

OK Khonshu  
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

Recebeu um novo post na tag: GLOBAL

READ GLOBAL  
OK ALLPOSTS GLOBAL

POST 1

My name is Steven with a V,  
Sometimes Mr. Knight  
Sometimes an idiot  
~Steven

POST GLOBAL  
He's right  
He's an idiot  
That should be killed  
EXIT

OK POST

MSG GLOBAL \*kills you in Spanish\*  
~Jake

MSG Khonshu leave us alone  
~Steven

█

```
joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]
```

Username: Jake

OK Jake  
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

Recebeu um novo post na tag: GLOBAL

Recebeu um novo post na tag: GLOBAL

READ GLOBAL  
OK ALLPOSTS GLOBAL

POST 1

My name is Steven with a V,  
Sometimes Mr. Knight  
Sometimes an idiot  
~Steven

POST 2

He's right  
He's an idiot  
That should be killed  
~Khonshu

MSG GLOBAL \*kills you in Spanish\*  
~You

OK MSG GLOBAL

█

```
joao@Joaos-MBP src % bash run_server.sh
Socket: [ OK ]
Nova conexão, o FD do socket é 4 , o ip é : 127.0.0.1 e a
porta : 57864
Nova conexão, o FD do socket é 5 , o ip é : 127.0.0.1 e a
porta : 57865
Nova conexão, o FD do socket é 11 , o ip é : 127.0.0.1 e
a porta : 57866
tags/GLOBAL.txt
tags/GLOBAL.txt
Jake enviou 'MSG GLOBAL *kills you in Spanish*' para todos
```

Steven enviou 'MSG Khonshu leave us alone' para Khonshu

█

Fig 3 – Exemplo que demonstra o funcionamento de MSG na Tag Global e as devidas mensagens do servidor

```

joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Piplup

OK Piplup
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

MSG GLOBAL Who's a water type?
-You

OK MSG GLOBAL

MSG GLOBAL Me
-Vapoleon

MSG GLOBAL :(
-Charizard

POST WaterTypes
Welcome my bros from the Ocean
This is a group to talk about water
because we are water dudes
its what we do
EXIT

OK POST

ADD WaterTypes Vapoleon
ADD WaterTypes Charizard
MSG WaterTypes :D
-Charizard

MSG WaterTypes oops
-You

RMV WaterTypes Charizard
MSG GLOBAL (T-T)
-Charizard

MSG WaterTypes auch...
-Vapoleon

Charizard desligou

joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Charizard

OK Charizard
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

MSG GLOBAL Who's a water type?
-Piplup

MSG GLOBAL Me
-Vapoleon

MSG GLOBAL :(
-You

MSG WaterTypes oops
-Piplup

Foi removido da tag WaterTypes

MSG GLOBAL (T-T)
-You

OK MSG GLOBAL

QUIT
joao@Joaos-MBP src %

joao@Joaos-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Vapoleon

OK Vapoleon
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

MSG GLOBAL Who's a water type?
-Piplup

MSG GLOBAL Me
-You

OK MSG GLOBAL

MSG GLOBAL :(
-Charizard

Foi adicionado à tag WaterTypes

MSG WaterTypes :D
-Charizard

MSG WaterTypes oops
-Piplup

MSG GLOBAL (T-T)
-Charizard

MSG WaterTypes auch...
-You

Charizard desligou

ONLINE
Utilizadores Online:
Piplup

joao@Joaos-MBP src % bash run_server.sh
Socket: [ OK ]
Nova conexão, o FD do socket é 4 , o ip é : 127.0.0.1 e a
porta : 57975
Nova conexão, o FD do socket é 5 , o ip é : 127.0.0.1 e a
porta : 57976
Nova conexão, o FD do socket é 11 , o ip é : 127.0.0.1 e
a porta : 57978
Piplup enviou 'MSG GLOBAL Who's a water type?' para todos
Vapoleon enviou 'MSG GLOBAL Me' para todos
Charizard enviou 'MSG GLOBAL :( ' para todos
tags/WaterTypes.txt
Tag WaterTypes criada por Piplup
Charizard enviou 'MSG GLOBAL (T-T)' para todos
Charizard desconnectou-se do servidor

```

Fig 4 – Exemplo que demonstra o funcionamento de POST, ADD, RMV

<pre> jrcondeco@AsusJC:~/Desktop/48566_48976/src\$ ./client t localhost 1234 HOST: [ FOUND ] Socket: [ OK ] Server: [ OK ]  Username: Piplup  OK Piplup Status: [ Connected ]   ---- Welcome to 1TextChat ----   ADD WaterTypes Vaporeon █ </pre>	<pre> jrcondeco@AsusJC:~/Desktop/48566_48976/src\$ ./client localhost 1234 HOST: [ FOUND ] Socket: [ OK ] Server: [ OK ]  Username: Vaporeon  OK Vaporeon Status: [ Connected ]   ---- Welcome to 1TextChat ----   Foi adicionado à tag WaterTypes  READ WaterTypes OK ALLPOSTS WaterTypes  POST 1  Welcome my bros from the Ocean This is a group to talk about water because we are water dudes its what we do -Piplup  █ </pre>
<pre> jrcondeco@AsusJC:~/Desktop/48566_48976/src\$ ./client localhost 1234 HOST: [ FOUND ] Socket: [ OK ] Server: [ OK ]  Username: Charizard  OK Charizard Status: [ Connected ]   ---- Welcome to 1TextChat ----  READ WaterTypes ERR ALLPOSTS WaterTypes █ </pre>	<pre> jrcondeco@AsusJC:~/Desktop/48566_48976/src\$ ./serv er Socket: [ OK ] Nova conexão, o FD do socket é 4 , o ip é : 127.0. 0.1 e a porta : 59478 Nova conexão, o FD do socket é 5 , o ip é : 127.0. 0.1 e a porta : 59480 Nova conexão, o FD do socket é 20 , o ip é : 127.0 .0.1 e a porta : 59482 █ </pre>

Fig 5 – Exemplo que demonstra o funcionamento de POST, READ, respostas devidas do servidor

```

joao@Joao's-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Thor

OK Thor
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

MSG GLOBAL Let's do get help
-You

OK MSG GLOBAL

MSG GLOBAL I hate get help
-Loki

MSG GLOBAL Come on you love it
-You

OK MSG GLOBAL

MSG GLOBAL we are not doing get help
-Loki

FILE Loki anya.jpg
ERR FILE
FILE Loki anya.jpg
[]

joao@Joao's-MBP src % bash run_client.sh
HOST: [ FOUND ]
Socket: [ OK ]
Server: [ OK ]

Username: Loki

OK Loki
Status: [ Connected ]

|---- Welcome to 1TextChat ----|

MSG GLOBAL Let's do get help
-Thor

MSG GLOBAL I hate get help
-You

OK MSG GLOBAL

MSG GLOBAL Come on you love it
-Thor

MSG GLOBAL we are not doing get help
-You

OK MSG GLOBAL

FILE Loki anya.jpg 5900
-Thor

joao@Joao's-MBP src % bash run_server.sh
Socket: [ OK ]
Nova conexão, o FD do socket é 4 , o ip é : 127.0.0.1 e a porta : 58251
Nova conexão, o FD do socket é 5 , o ip é : 127.0.0.1 e a porta : 58252
Thor enviou 'MSG GLOBAL Let's do get help' para todos
Loki enviou 'MSG GLOBAL I hate get help' para todos
Thor enviou 'MSG GLOBAL Come on you love it' para todos
Loki enviou 'MSG GLOBAL we are not doing get help' para todos
Thor enviou um ficheiro de 5900 bytes para Loki
[]

```

Fig 6 – Exemplo que demonstra o funcionamento de FILE e respostas devidas do servidor

## 7. Dificuldades e Complicações Encontradas

Durante a elaboração deste trabalho deparamo-nos com alguns erros. Nomeadamente, um problema associado com o envio de ficheiros para utilizadores. O problema residia no facto do servidor não aguardar pela chegada de todos os dados, o que fazia com que o ficheiro recebido pelo utilizador aparecesse apenas parte do mesmo, estando este corrompido (tal pode ser observado nas figuras abaixo).

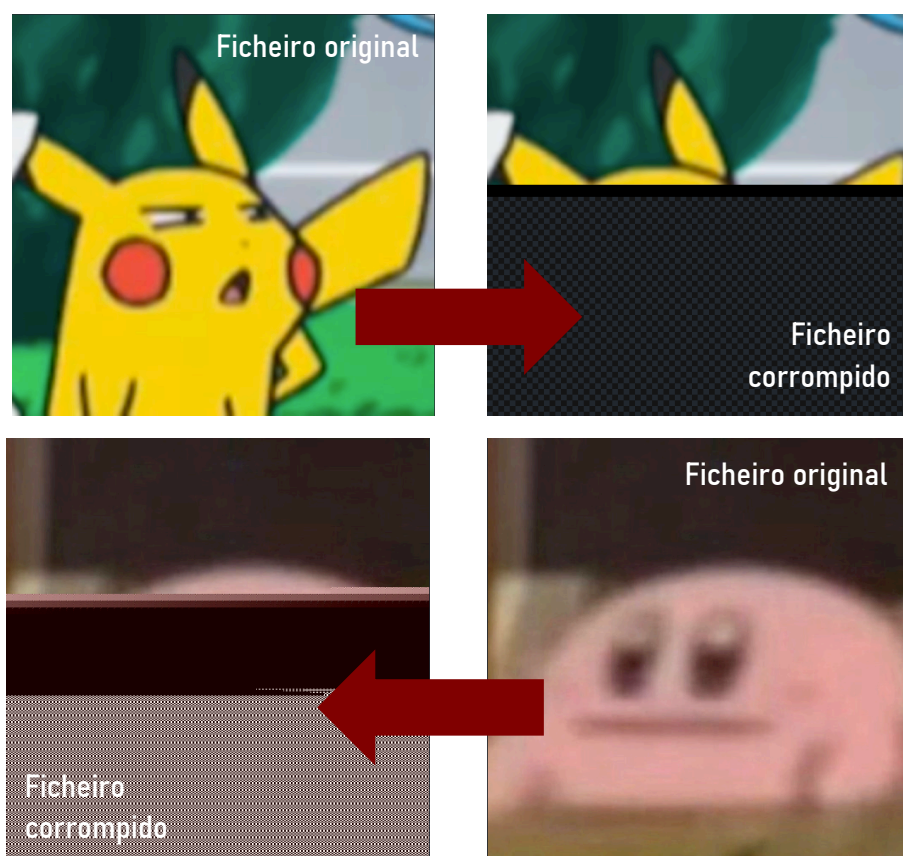


Fig 7 – Demonstração dos erros nos ficheiros

Este problema foi resolvido aplicando um ciclo for que garante o tempo suficiente para os dados chegarem devidamente ao servidor, e consequentemente, ao utilizador pretendido.

Outro problema que surgiu foi na inserção de dados na lista, ocorriam vários erros de *segmentation fault*, devido a uma má alocação da memória, uma vez identificado os mesmos, foram feitas as devidas alocações e o programa passou a funcionar como pretendido.