

**3ºano – MIEIC – Outubro 2017**

**Protocolo de Ligação de Dados**

**RCOM**

**Turma 2 - Grupo y**

**João Francisco Veríssimo Dias Esteves – up201505145**

**João Miguel Matos Monteiro – up201506130**

**Maria Eduarda Santos Cunha – up201506524**

Índice

[1. Sumário 4](#_Toc496979744)

[2. Introdução 5](#_Toc496979745)

[3. Arquitetura e Estrutura do Código 6](#_Toc496979746)

[3.1. Camada de Ligação de Dados 6](#_Toc496979747)

[3.2. Camada de Aplicação 6](#_Toc496979748)

[3.3. Interface 6](#_Toc496979749)

[4. Casos de Uso Principais 7](#_Toc496979750)

[5. Protocolo de Ligação Lógica 8](#_Toc496979751)

[5.1. llopen() e llclose() 8](#_Toc496979752)

[5.2. llwrite() e llread() 8](#_Toc496979753)

[6. Protocolo de Aplicação 9](#_Toc496979754)

[7. Validação 10](#_Toc496979755)

[8. Elementos de Valorização 11](#_Toc496979756)

[9. Conclusão 12](#_Toc496979757)

[10. Anexos 13](#_Toc496979758)

# Sumário

Este projeto consiste numa aplicação capaz de transmitir ficheiros entre 2 computadores pelo uso de uma porta de série assíncrona, resistente a certas falhas que possam surgir durante o processo de envio, nomeadamente a introdução de erros através do fio disponível na porta série e o fecho da mesma.

O problema proposto foi implementado com sucesso e absolutamente essencial para a consolidação dos conceitos lecionados nas aulas teóricas e laboratoriais.

# Introdução

Este relatório tem como objetivo complementar o primeiro projeto da Unidade Curricular Redes de Computadores, intitulado “Protocolo de Ligação de Dados”, com vista a permitir uma análise do código com o auxílio da perspetiva de quem o escreveu.

A funcionalidade principal desse projeto é permitir a comunicação de dados fiável entre 2 computadores ligados por um cabo de série.

O relatório encontra-se dividido nas seguintes secções:

**Arquitetura:** Blocos funcionais e interface;

**Estrutura do Código:** APIs, principais estruturas de dados, funções e respeitante relação com a arquitetura;

**Casos de Uso Principais:** Respetiva identificação e sequências de chamadas de funções;

**Protocolo de Ligação Lógica:** Descrição da estratégia aplicada e identificação dos aspetos funcionais principais;

**Protocolo de Aplicação:** Semelhante ao encontrado no Protocolo de Ligação Lógica, mas para a Aplicação;

**Validação:** Testes efetuados e resultados;

**Elementos de Valorização:** Componentes adicionais passíveis de serem implementadas.

# Arquitetura e Estrutura do Código

O projeto encontra-se dividido em 2 camadas: protocolo de ligação de dados e aplicação.

## Camada de Ligação de Dados

A solução pode ser dividida em três fases.

Através do algoritmo de Pesquisa Exata ou Aproximada, o utilizador insere o nome da rua que está à procura e verifica se existe.

## Camada de Aplicação

A solução pode ser dividida em três fases.

Através do algoritmo de Pesquisa Exata ou Aproximada, o utilizador insere o nome da rua que está à procura e verifica se existe.

## Interface

A solução pode ser dividida em três fases.

Através do algoritmo de Pesquisa Exata ou Aproximada, o utilizador insere o nome da rua que está à procura e verifica se existe.

# Casos de Uso Principais

Identificação

Sequência de chamada de funções

# Protocolo de Ligação Lógica

Identificação dos principais aspetos funcionais

Descrição da estratégia de implementação destes aspetos com a apresentação de extratos de codigo

## llopen() e llclose()

A solução pode ser dividida em três fases.

Através do algoritmo de Pesquisa Exata ou Aproximada, o utilizador insere o nome da rua que está à procura e verifica se existe.

## llwrite() e llread()

A solução pode ser dividida em três fases.

Através do algoritmo de Pesquisa Exata ou Aproximada, o utilizador insere o nome da rua que está à procura e verifica se existe.

# Protocolo de Aplicação

Identificação dos principais aspetos funcionais

Descrição da estratégia de implementação destes aspetos com a apresentação de extratos de codigo

# Validação

De forma a testar a eficiência do código implementado, procedemos à realização de vários testes, como a transferência do ficheiro sem qualquer tipo de “obstáculo” (fig.x), fechando e voltando a abrir a porta de série (fig.y), fechando-a até ao TIMEOUT (fig.z) e com a introdução de erros (fig.w).

Para os diferentes tipos de teste, utilizou-se vários ficheiros: pinguim.gif e alguns ficheiros extra, de entre os quais ficheiros do tipo .txt e imagens com diferentes tamanhos mais pesados que o sugerido.

Como resposta a estes testes, é imprimido na consola o progresso em percentagem do envio do ficheiro, se foi ou não enviado com sucesso, os TIMOUTS, se deu reject etc.

Eficiência do protocolo de ligação de dados

caraterização estatística da eficiência do protocolo, feita com recurso a medidas sobre o código desenvolvido. A caracterização teórica de um protocolo Stop&Wait, que deverá ser usada como termo de comparação, encontra-se descrita nos slides de Ligação Lógica das aulas teóricas

# Elementos de Valorização

**Seleção do parâmetro ficheiro pelo utilizador:** Aquando da chamada da função do emissor, o ficheiro a ser enviado é um dos parâmetros.

**Implementação de REJ:**

**Registo de ocorrências:**

# Conclusão

A realização deste projeto, ainda que muito trabalhosa, foi fundamental para a interiorização dos conceitos lecionados nas aulas teóricas e laboratoriais.

Consideramos que o nosso objetivo foi cumprido, dado que respondemos com sucesso a todas as especificações pedidas no enunciado do trabalho. Temos 2 camadas independentes, mas com uma ligação unidirecional, já que a aplicação exerce controlo sobre a ligação de dados.

Após alguma reflexão, concluímos que, se nos fosse concedido mais tempo, podíamos implementar geração aleatória de erros, a receção na linha de comandos de parâmetros como baud rate, o valor de *time out*, o número de tentativas, o tamanho máximo dos pacotes, etc.

# Anexos

Encontra-se em anexo o código fonte do projeto.