



Catarina Pereira
Catarina Amorim
Inês Neves
Leonardo Martins

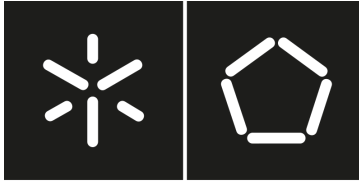
**Comunicação pela Luz: Li-Fi em Ação para
Conversação e Transferência de Ficheiros**



Universidade do Minho
Escola de Engenharia

Catarina da Cunha Malheiro da Silva Pereira
Catarina Marinho Amorim
Inês Cabral Neves
Leonardo Dias Martins

**Comunicação pela Luz: Li-Fi em Ação
para Conversação e Transferência de
Ficheiros**



Universidade do Minho

Escola de Engenharia

Catarina da Cunha Malheiro da Silva Pereira
Catarina Marinho Amorim
Inês Cabral Neves
Leonardo Dias Martins

Comunicação pela Luz: Li-Fi em Ação para Conversação e Transferência de Ficheiros

Relatório de Especificação da Fase C
Projeto Integrador em
Telecomunicações e Informática
Mestrado em Engenharia
Telecomunicações e Informática

Trabalho efetuado sob a orientação de:

Professor Doutor Joaquim Melo Henriques Macedo

Professor Doutor Adriano Jorge Cardoso Moreira





**Professor Doutor Bruno Daniel Mestre Viana
Ribeiro**

e de

Professor Doutor José Augusto Afonso

Identificação do Grupo

O Grupo 03 é constituído por quatro membros, sendo que três deles são do 1º ano do Mestrado em Engenharia de Telecomunicações e Informática (METI), sendo identificados por Pós-Graduação (PG) seguido dos seus números mecanográficos, enquanto o quarto membro pertence ao 4º ano de Mestrado Integrado em Engenharia de Telecomunicações e Informática (MIETI) e é identificado pelo código Aluno (A) seguido do seu número mecanográfico.

Imagem	Nome / Número Mecanográfico / E-mail institucional
	Catarina da Cunha Malheiro da Silva Pereira PG53733 pg537336@alunos.uminho.pt
	Catarina Marinho Amorim A93094 a93094@alunos.uminho.pt
	Inês Cabral Neves PG53864 pg53864@alunos.uminho.pt
	Leonardo Dias Martins PG53996 pg53996@alunos.uminho.pt

Índice

Identificação do Grupo	ii
Índice de Figuras	iv
Índice de Tabelas	v
Lista de Acrónimos	vi
Acrónimos	vi
1 Introdução	1
2 Revisão da Literatura	2
3 Recursos e Instrumentos Utilizados	3
3.1 Software	3
3.2 Hardware	3
4 Arquitetura do Sistema	4
5 Protocolo de Comunicação	5
5.1 PC-ESP32	5
5.2 Aplicação-Aplicação	5
6 Interface Gráfica	7
7 Melhorias Relativas á Fase Anterior	9
8 Requisitos Funcionais e Não Funcionais	10
9 Plano de Atividades	11
9.1 Atividades	11
9.2 Lista de Riscos	11
10 Conclusão	13
Referências Bibliográficas	14
Anexo I - Diagrama de Gantt	14

Índice de Figuras

1	Placa ESP32 [8].	3
2	Arquitetura Geral da fase C.	4
3	Pinout do Circuito Emissor.	4
4	Pinout do Circuito Emissor.	4
5	Mecanismo de Controlo de Fluxo <i>Stop-and-Wait</i>	5
6	Especificação das Tramas da Camada de Aplicação.	6
7	Aplicação do emissor.	7
8	Aplicação do recetor.	8

Índice de Tabelas

Tabela 1: Requisitos Funcionais do projeto.	10
Tabela 2: Requisitos Não Funcionais do projeto.	10
Tabela 3: Plano de Atividades.	11
Tabela 4: Riscos inerentes ao desenvolvimento do projeto.	12

Acrónimos

A	Aluno
GPIO	General Purpose Input/Output
IDE	Integrated Development Environment
LEDs	Díodos Emissores de Luz
Li-Fi	Light Fidelity
METI	Mestrado em Engenharia de Telecomunicações e Informática
MIETI	Mestrado Integrado em Engenharia de Telecomunicações e Informática
OSI	Open Systems Interconnection
PG	Pós-Graduação
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
TINA	Toolkit for Interactive Network Analysis
UC	Unidade Curricular

1 Introdução

Este relatório faz parte da Unidade Curricular (UC) Projeto Integrador em Telecomunicações e Informática, do 2º semestre do 1º ano do Mestrado (Integrado) em Engenharia de Telecomunicações e Informática. Este projeto foi desenvolvido como resposta a um problema apresentado pelos docentes.

A ascensão dos Díodos Emissores de Luz (LEDs) transformou significativamente o panorama das tecnologias de iluminação, proporcionando eficiência, durabilidade e versatilidade notáveis. Presentes em diversas aplicações, desde iluminação residencial até sinalização avançada e iluminação automóvel, os LEDs não apenas redefiniram a maneira como se ilumina a área circunvente de cada pessoa, mas também desbravaram novos caminhos para a comunicação e inovação.

Na Fase C do projeto, é atingido uma fase crucial onde espera-se que se concretize o culminar dos esforços no desenvolvimento do protótipo para transmissão ótica de dados entre dois PCs. Esta fase marca a integração de todos os componentes e funcionalidades num sistema coeso que incorpora a essência dos objetivos do projeto. O foco principal da Fase C é finalizar a implementação, realizar testes abrangentes e apresentar uma solução totalmente funcional que demonstre o uso inovador da tecnologia Light Fidelity (Li-Fi) para permitir a comunicação perfeita entre dispositivos. Este relatório descreve as especificações detalhadas, o progresso, os desafios superados e a reflexão abrangente sobre a jornada para alcançar um resultado bem-sucedido.

O projeto não só proporciona uma aplicação prática dos conceitos de comunicação ótica, mas também desafia os estudantes a integrarem eficazmente hardware e software, contribuindo para o seu desenvolvimento técnico e científico em áreas como eletrónica. O relatório oferecerá um relato detalhado do pensamento da criação do projeto para atual fase.

A lista de tarefas a executar na Fase C é:

- Implementar um protocolo de nível 7 (camada de aplicação);
- Criar o programa com as funcionalidades de chat e transferência de ficheiros entre PCs. Este programa deverá utilizar o protocolo da camada 7 deve ser o mesmo para os dois PCs;
- Criar uma interface gráfica para o programa;

2 Revisão da Literatura

O modelo Open Systems Interconnection (OSI), uma estrutura fundamental para comunicação em rede, consiste em sete camadas, cada uma com funções e interações específicas. Estas camadas incluem as camadas Física, Ligação de Dados, Rede, Transporte, Sessão, Apresentação e Aplicação, que gerem coletivamente a transferência de dados e garantem uma comunicação segura. É destacado o papel do modelo em facilitar a transferência de informações entre diferentes endereços lógicos, com os gateways desempenhando um papel crucial neste processo [1]. Também é enfatizada a importância do modelo OSI no estabelecimento de protocolos de comunicação para redes de computadores [2].

A camada 7 do modelo OSI, conhecida como Camada de Aplicação, é a camada superior do modelo e é responsável por fornecer serviços de rede diretamente às aplicações dos utilizadores [3]. Esta camada permite a comunicação entre diferentes aplicações e é onde os utilizadores interagem com a rede. Esta camada define os protocolos e métodos que as aplicações usam para trocar dados pela rede [4]. A camada de aplicação inclui protocolos como HTTP, FTP, SMTP e DNS, que são essenciais para vários tipos de comunicação de rede e troca de dados. Em essência, a Camada 7 concentra-se nos requisitos de comunicação das aplicações de software e garante que os dados sejam apresentados ao utilizador de forma significativa [5].

3 Recursos e Instrumentos Utilizados

Neste capítulo explora-se em detalhe os elementos que desempenham um papel fundamental na condução deste projeto, com um foco predominante em ferramentas conforme a sua categoria: *Software* e *Hardware*. O conjunto de ferramentas e recursos utilizados abrange uma ampla variedade de aplicações, cada uma desempenhando um papel específico e vital no desenvolvimento do projeto.

3.1 Software

As ferramentas e recursos utilizados são:

- **Smartsheet:** Emprega-se o programa Smartsheet para o planeamento temporal das tarefas do grupo, garantindo uma gestão eficaz do cronograma de trabalho.
- **Miro:** Utiliza-se a plataforma Miro para criar diagramas de blocos e fluxogramas, facilitando a visualização e a comunicação de conceitos complexos.
- **Arduino Integrated Development Environment (IDE):** Para a edição, compilação e envio do código para a placa ESP32.
- **PyCharm:** Para desenvolvimento e execução de código Python.
- **Simulador Toolkit for Interactive Network Analysis (TINA) (versão 9) e Circuit Diagram:** Permitem fazer simulações tanto de circuitos analógicos como digitais.
- **Plataformas de Comunicação:** Para facilitar a comunicação, colaboração e organização do código desenvolvido pelo grupo, utiliza-se as plataformas Discord e Whatsapp.
- **OverLeaf:** Para a elaboração de relatórios em formato \LaTeX , utiliza-se a plataforma OverLeaf, que simplifica a formatação e a colaboração em documentos técnicos.

Estas ferramentas e recursos desempenham um papel essencial na pesquisa, contribuindo para a eficiência na recolha e análise de dados, bem como na comunicação e documentação dos resultados.

3.2 Hardware

Para as ferramentas de *hardware* utilizar-se-à uma placa ESP32, Amplificador operacional UA741CN, Fotodetetor PL-51P3C, Fotodiodo PL-53F3BT e Transistor 2N2222A para o projeto.

A placa ESP32, Figura 1, é uma placa de desenvolvimento de *hardware* que utiliza o microcontrolador ESP32. O ESP32 é um microcontrolador de baixo custo e baixo consumo de energia que possui recursos avançados como conectividade Wi-Fi e Bluetooth, memória *flash* integrada, entrada e saída de dados digitais e analógicos, entre outros [6, 7].

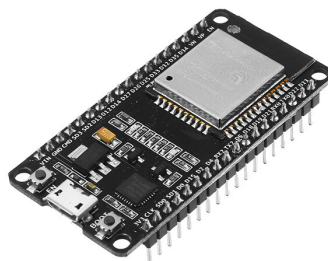


Figura 1: Placa ESP32 [8].

4 Arquitetura do Sistema

A arquitetura geral da Fase C é delineada na Figura 2, destacando os componentes essenciais e os protocolos envolvidos. Nesta figura, são apresentados o emissor, o circuito driver, o LED de infravermelhos, o fotodetector, o PC, a ESP32 e a conexão ótica, juntamente com o protocolo UART.

O circuito driver é um circuito eletrônico projetado para amplificar e controlar a corrente ou tensão num dispositivo, como um LED infravermelhos, neste caso. Este circuito será conectado ao Digital Lab, fornecendo os sinais ao driver, que os amplificará e controlará de acordo com as especificações do dispositivo.

Por outro lado, o circuito frontend é projetado para receber e processar sinais de entrada de uma fonte externa, que será o emissor. Este circuito é responsável por amplificar, filtrar e converter os sinais de entrada num formato adequado para processamento posterior pelo circuito eletrónico.

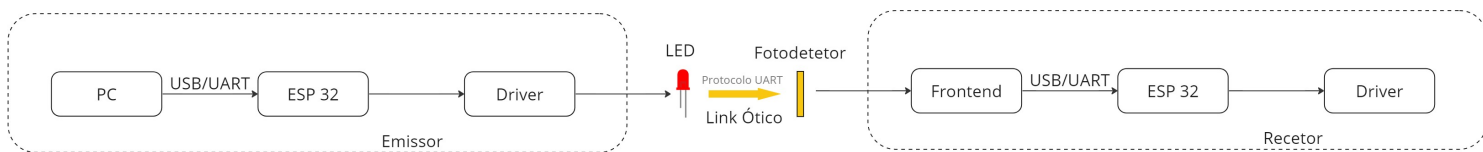


Figura 2: Arquitetura Geral da fase C.

A Figura 3 mostra a conexão do circuito Driver ao ESP32 através do pino General Purpose Input/Output (GPIO) 3 para o uso do protocolo UART, enquanto a Figura 4 mostra a conexão do circuito Frontend ao ESP32 através do pino GPIO16.

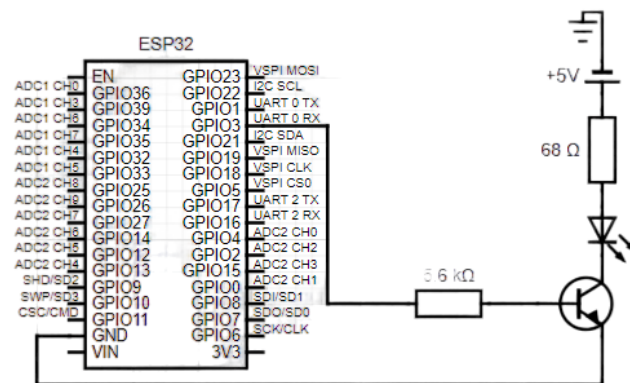


Figura 3: Pinout do Circuito Emissor.

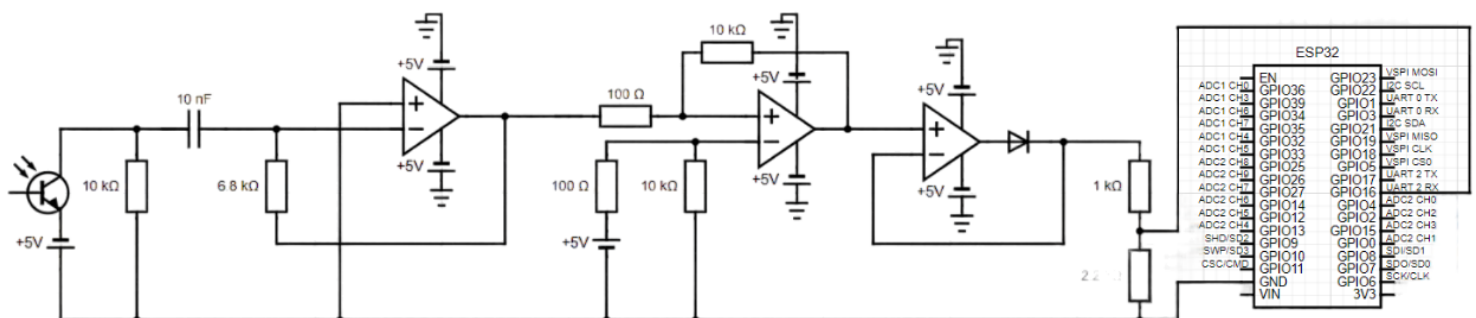


Figura 4: Pinout do Circuito Emissor.

5 Protocolo de Comunicação

Pretende-se desenvolver aplicações de interface com o utilizador por isso, será necessário desenvolver protocolos de comunicação entre PC (aplicação) e a ESP32 e, entre as aplicações.

5.1 PC-ESP32

Este protocolo de comunicação irá focar-se essencialmente no controlo de fluxo e será adotada a estratégia *Stop-and-Wait*, ou seja, a aplicação envia uma trama e aguarda a receção de um *Acknowledgement* (ACK) para enviar outra trama. Também é ativado um *timeout* de 1 segundo para caso não receba a confirmação, retransmitir a trama. Este controlo de fluxo apenas ocorrerá no lado do emissor. Do lado do recetor, caso a aplicação recetora esteja a receber um fluxo de dados e detete que não recebeu uma trama, por exemplo, recebeu as tramas número 1, 2 e 4, detetou que não recebeu a trama número 3, então envia uma mensagem de erro para a ESP32. Por sua vez, a ESP32 não envia mais dados para a aplicação, pois não vale a pena. A Figura 5 exemplifica o protocolo de comunicação usado entre o PC e ESP32.

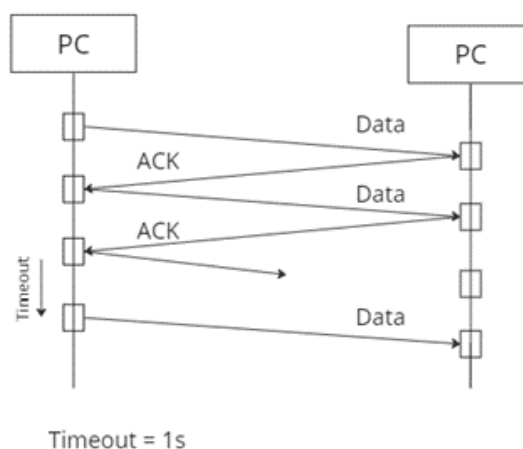


Figura 5: Mecanismo de Controlo de Fluxo *Stop-and-Wait*.

Ao receber os dados da aplicação, a placa ESP32 não se deve preocupar com o seu conteúdo ou formato, simplesmente encapsula esses dados com o seu cabeçalho e envia-os através do link ótico. O cabeçalho é simples, apenas contém um campo com o tamanho dos dados enviados pela aplicação.

5.2 Aplicação-Aplicação

Para uma eficiente comunicação e interpretação dos dados entre o remetente e o destinatário, deverá definir-se um conjunto de mensagens (que apenas serão enviadas do remetente para o destinatário, uma vez que a comunicação é unidirecional):

- Início de transmissão - antes do envio de um fluxo de dados, o emissor envia uma mensagem para indicar ao recetor que está prestes a iniciar o envio de um conjunto de dados. Também indica qual o tipo de dados do stream, se é texto simples, ou se for um ficheiro, indica também o nome do ficheiro.
- Fim de transmissão - indica ao recetor que a transmissão daquele fluxo de dados terminou.
- Dados - mensagem com os dados que o emissor quer fazer chegar ao recetor, pode ser texto, um ficheiro, etc.

A Figura 6 apresenta a especificação das tramas da camada de aplicação que se encontra encapsulada dentro do campo "Dados" da trama da camada de ligação.

5.2. APLICAÇÃO-APLICAÇÃO

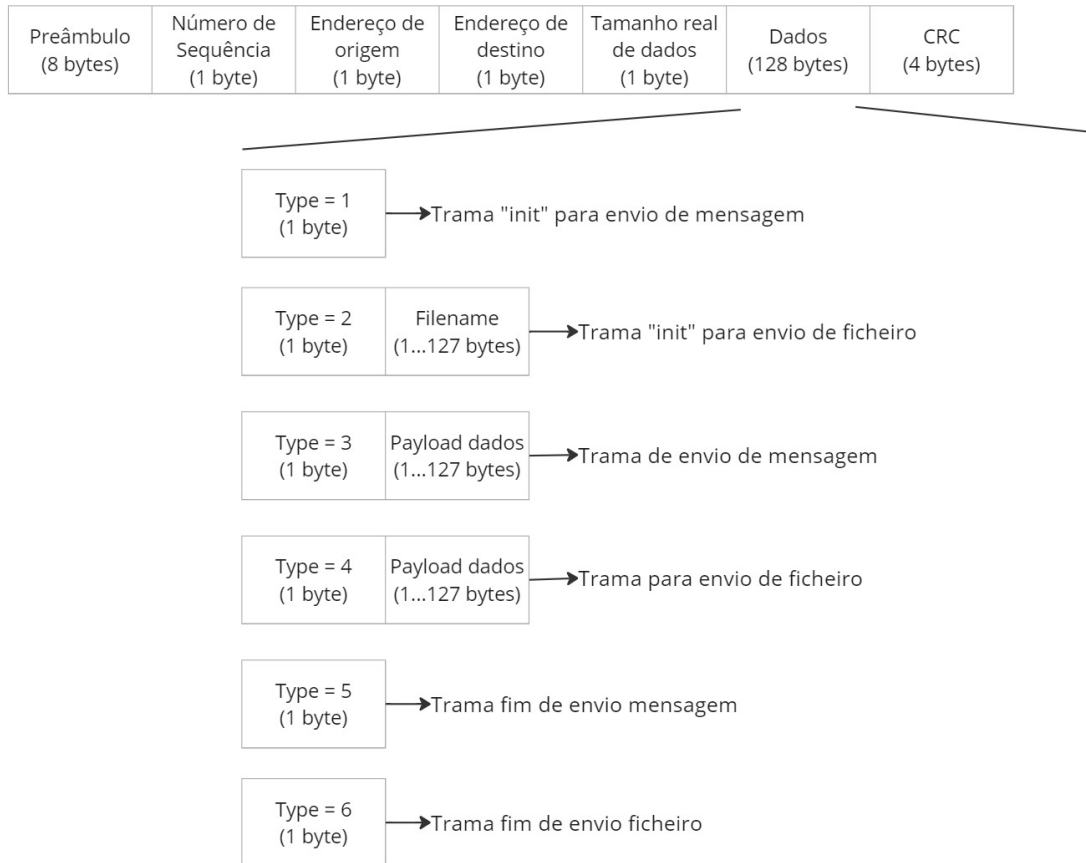


Figura 6: Especificação das Tramas da Camada de Aplicação.

6 Interface Gráfica

Nesta fase do projeto, serão desenvolvidas duas aplicações diferentes, ambas com uma interface gráfica, que permita ao utilizador efetuar a comunicação de uma forma mais intuitiva, sendo uma para o lado do transmissor e outra para o lado do recetor, tendo características a nível de funcionalidades muito diferentes entre elas. Nos pontos abaixo serão explicadas as funcionalidades e as diferenças entre cada programa.

A aplicação de transmissão, observada na Figura 7 irá permitir enviar dados, como por exemplo, mensagens de texto e até mesmo ficheiros do modulo emissor para o recetor. A aplicação permite também selecionar a porta COM associada à placa ESP32 que se encontra conectada ao PC. Para se ter uma perceção do estado do envio de pacotes, principalmente em ficheiros ou mensagens grandes, a aplicação irá conter uma barra de progresso para mostrar a evolução do transmissão dos dados ao longo do tempo.

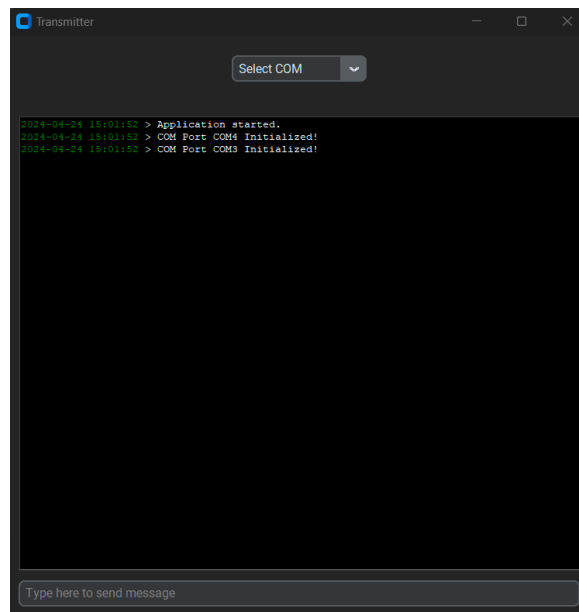


Figura 7: Aplicação do emissor.

A aplicação de receção, observada na Figura 8 apenas irá fazer a receção dos dados enviados a partir da aplicação de transmissão e fazer a correção dos erros dos blocos de dados que lhe chegam. No caso de ser uma simples mensagem, disponibilizar a mesma na caixa de texto da aplicação ou no caso de ser um ficheiro aguardar por todas as tramas correspondentes a ele e disponibilizar o ficheiro ao utilizador. Tal como na aplicação de transmissão, é também possível selecionar a porta COM associada à placa ESP32 que se encontra conectada ao PC.

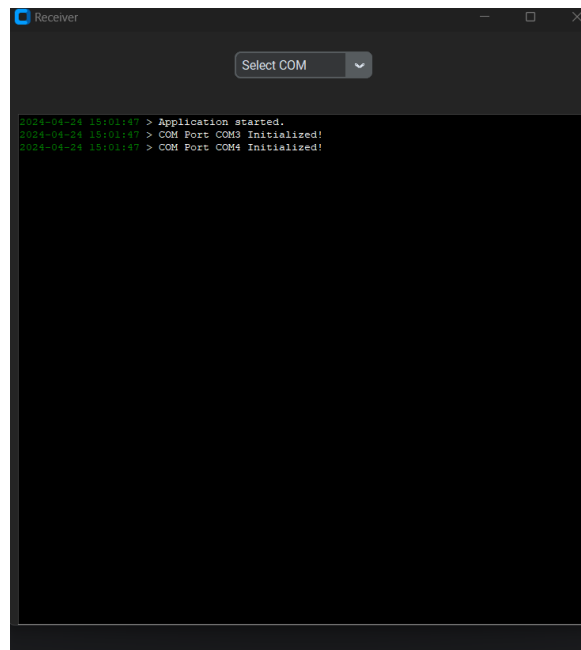


Figura 8: Aplicação do recetor.

7 Melhorias Relativas à Fase Anterior (API)

Durante o desenvolvimento da Fase B, foram implementadas melhorias significativas na API de Comunicação Aplicação-Ligação, conforme descrita no protocolo de especificação da fase anterior.

Funções disponíveis para a interação entre a camada de aplicação e a camada de ligação, do ponto de vista do emissor:

- Ler ACKs: Permite ao emissor monitorizar e interpretar os acknowledgments (ACKs) recebidos da camada de ligação, confirmando a receção bem-sucedida dos pacotes de dados enviados.
- Encapsular o pacote: Permite o emissor a encapsular os dados provenientes da camada de aplicação em pacotes apropriados para transmissão, preparando-os para serem enviados através da camada de ligação.
- Enviar o pacote: Oferece a funcionalidade de enviar os pacotes encapsulados pela camada de ligação, transmitindo-os para o destino final ou para a próxima etapa do processo de comunicação.

Funções disponíveis para a interação entre a camada de aplicação e a camada de ligação, do ponto de vista do recetor:

- Enviar ACKs: Permite ao recetor enviar acknowledgments (ACKs) à camada de ligação, confirmando a receção bem-sucedida dos pacotes de dados transmitidos.
- Ler o pacote: Permite o recetor a ler e interpretar os pacotes recebidos da camada de ligação, extraindo os dados encapsulados e encaminhando-os para a camada de aplicação para processamento posterior.

8 Requisitos Funcionais e Não Funcionais

Neste capítulo, são apresentados os Requisitos Funcionais (RF), Tabela 1, e Requisitos Não Funcionais (RNF), Tabela 2, duas categorias essenciais de especificações que direcionam o desenvolvimento do sistema de software e hardware em questão. Estes requisitos delineiam o que o sistema deve realizar (RF) e as condições nas quais deve operar (RNF).

Tabela 1: Requisitos Funcionais do projeto.

ID	Requisitos Funcionais	Justificação
RF1	Implementação um protocolo da Camada 7 (Camada de Aplicação).	Desenvolver um protocolo para a Camada de Aplicação para permitir a comunicação entre diferentes aplicações.
RF2	Criação um programa de conversa e transferência de arquivos	Desenvolver um programa com funcionalidades de conversa e transferência de arquivos entre dois PCs.
RF3	Projeção uma interface gráfica de utilizador (GUI)	Criar uma interface intuitiva e fácil de usar para o programa

Tabela 2: Requisitos Não Funcionais do projeto.

ID	Requisitos Não Funcionais	Justificação
RNF1	Desempenho.	O sistema deve ser capaz de transferir arquivos e mensagens de chat de forma rápida e eficiente.
RNF2	Confiabilidade	Garantir a transferência confiável de dados sobre a ligação sem fios, mesmo em caso de interrupção momentânea do feixe ótico.
RNF3	Segurança	Implementar medidas de segurança para proteger a privacidade e integridade dos dados transmitidos.

Para demonstração, o sistema deve conseguir transferir com sucesso um arquivo de imagem e um arquivo de texto entre dois computadores através da camada 2 e da porta série, cumprindo todos os requisitos funcionais e não funcionais estabelecidos.

9 Plano de Atividades

Neste capítulo, detalha-se as atividades planeadas, definimos prazos e recursos necessários. Além disso, abordamos a sequência lógica das tarefas e como elas se relacionam umas com as outras, garantindo uma execução eficiente e coordenada.

A compreensão deste Plano de Atividades é fundamental para a gestão eficaz do projeto, permitindo que todas as partes interessadas tenham uma visão clara das etapas a serem seguidas e dos marcos a serem alcançados. Isso assegura que o projeto seja concluído dentro do prazo e com sucesso.

9.1 Atividades

Na Tabela 3, são apresentadas as atividades planeadas, juntamente com datas de início e conclusão. Além disso, no Anexo I, encontra-se o Diagrama de Gantt que fornece uma representação visual do plano de trabalho.

Tabela 3: Plano de Atividades.

ID	Atividade	Início	Conclusão
1	Fase C - Camada de aplicação e respetivas aplicações	23/04/2024	19/05/2024
1.1	Estudo e Pesquisa	23/04/2024	30/04/2024
1.1.1	Conhecimento da Camada 7.	23/04/2024	30/04/2024
1.2	Relatório de Especificação	23/04/2024	30/04/2024
1.2.1	Preparar esquemas detalhados do emissor e do recetor.	23/04/2024	30/04/2024
1.2.2	Definir o formato das tramas.	23/04/2024	30/04/2024
1.2.3	Entregar o relatório de especificação.	23/04/2024	30/04/2024
1.3	Projeto e Implementação	23/04/2024	19/05/2024
1.3.1	Especificação das Funcionalidades da Aplicação.	23/04/2024	30/04/2024
1.3.2	Desenvolver o protocolo de comunicação PC-ESP32.	26/04/2024	01/05/2024
1.3.3	Desenvolver das aplicações nos PCs.	30/04/2024	08/05/2024
1.3.4	Integração do link ótico com ESP32.	30/04/2024	08/05/2024
1.3.5	Testes e Análise de Resultados.	08/05/2024	15/05/2024
1.3.6	Desenvolver do Relatório Final.	10/05/2024	19/05/2024
1.3.7	Desenvolver da Apresentação Final.	10/05/2024	19/05/2024
1.4	Entrega da Fase C	19/05/2024	19/05/2024

9.2 Lista de Riscos

Nesta secção identifica-se e descreve-se os riscos potenciais associados ao projeto, Tabela 4. Cada risco é avaliado quanto à probabilidade de ocorrência, impacto, seriedade e os seus impactos/efeitos. Também são fornecidas ações de mitigação para lidar com estes riscos e minimizar os seus impactos. A cada um dos itens, para a probabilidade e o impacto, é atribuída uma pontuação numa escala de 1 a 5, em que o 1 corresponde a baixo e 5 corresponde a alto. A seriedade de cada risco obtém-se multiplicando a probabilidade pelo impacto, permitindo enaltecer os riscos que mais impacto poderão causar no projeto caso ocorram, de forma a estar mais atentos a eles.

A gestão dos riscos é uma parte essencial do planeamento do projeto, pois ajuda a prevenir problemas e a manter o projeto no caminho certo. Portanto, a identificação e avaliação destes riscos são cruciais para o sucesso do projeto.

9.2. LISTA DE RISCOS

Tabela 4: Riscos inerentes ao desenvolvimento do projeto.

ID	Risco	Mitigação	P	I	S
R1	Problemas de integração entre hardware e software.	Realizar testes frequentes de integração.	4	5	20
R2	Falhas nos testes de comunicação.	Realizar testes em diferentes condições de rede e ambiente, implementar mecanismos de detecção e correção de erros.	3	4	12
R3	Problemas de compatibilidade entre as plataformas de comunicação.	Escolher e testar plataformas de comunicação adequadas desde o início do projeto, manter backups e documentação atualizada.	2	3	6
R4	Falta de conhecimento técnico adequado.	Promover a partilha de conhecimento entre membros da equipa.	3	3	9
R5	Compreensão inadequada das interações de API entre camadas.	Definir e implementar minuciosamente as funções da API.	2	3	6

10 Conclusão

A realização deste trabalho destaca a importância dos Diodos Emissores de Luz (LEDs) na transformação das tecnologias de iluminação, abrindo caminho para inovações em comunicação. A implementação da tecnologia Light Fidelity (Li-Fi) para a transmissão ótica de dados entre dois PCs representa um marco significativo no projeto, culminando esforços para criar uma solução funcional e inovadora. A Fase C concentra-se na finalização da implementação, realização de testes abrangentes e apresentação de uma solução totalmente operacional, demonstrando a aplicação prática dos conceitos de comunicação ótica. A arquitetura do sistema, os protocolos de comunicação entre PC-ESP32 e aplicações, e a interface gráfica desenvolvida refletem a integração de hardware e software, contribuindo para o desenvolvimento técnico dos estudantes.

Referências Bibliográficas

- [1] Syed V. Ahamed e Victor B. Lawrence. "The Role of the OSI Model". Em: Springer US, 1997, pp. 94–123. DOI: 10.1007/978-1-4615-6341-9_4.
- [2] Sushmita Biya e Renuka Uday Kotwal. "The OSI Model: Overview of All Seven Layers of Computer Networks". Em: *International Journal of Advanced Research in Science, Communication and Technology* (set. de 2023), pp. 427–432. ISSN: 2581-9429. DOI: 10.48175/IJARSCT-13064.
- [3] Peter Boait et al. "Data Link Layer — Layer 2". Em: Macmillan Education UK, 1988, pp. 62–89. DOI: 10.1007/978-1-349-10306-5_5.
- [4] Gerry Howser. "The OSI Seven Layer Model". Em: Springer International Publishing, 2020, pp. 7–32. DOI: 10.1007/978-3-030-34496-2_2.
- [5] Pradeep Kumar Srivastava e Sandhya Tiwari. "An Overview of Open System Interconnection (OSI): A Seven Layered Model". Em: 2017. URL: <https://api.semanticscholar.org/CorpusID:113631165>.
- [6] Espressif Systems. *ESP32 Series Datasheet*. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acedido em 13 de fevereiro de 2023. Jan. de 2023.
- [7] <https://www.botnroll.com/pt/arduino-controladores/3540-esp32-placa-de-desenvolvimento-wifi-bluetooth-esp-32s-ai-thinker.html>. Acedido em 15 de fevereiro de 2023.
- [8] BangGood. https://pt.banggood.com/ESP32-Development-Board-WiFi+bluetooth-Ultra-Low-Power-Consumption-Dual-Cores-ESP-32-ESP-32S-Board-Geekcreit-for-Arduino-products-that-work-with-official-Arduino-boards-p-1109512.html?cur_warehouse=CN. Acedido em 13 de fevereiro de 2023.

Anexo I - Diagrama de Gantt

