



Catarina Pereira
Catarina Amorim
Inês Neves
Leonardo Martins

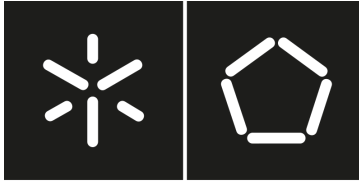
**Comunicação pela Luz: Li-Fi em Ação para
Conversação e Transferência de Ficheiros**



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 B
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	5
5 Protocolo de Comunicação	6
5.1 Noção de trama/pacote	6
5.2 Data Link Layer	6
6 Análise Quantitativa	8
7 Requisitos Funcionais e Não Funcionais	9
8 Plano de Atividades	10
8.1 Atividades	10
8.2 Lista de Riscos	10
9 Conclusão	12
Referências Bibliográficas	13
Anexo I - Diagrama de Gantt	13

Índice de Figuras

1	Placa ESP32 [11].	4
2	Arquitetura Geral da fase B.	5
3	Pinout do Circuito Emissor.	5
4	Pinout do Circuito Recetor.	5
5	Formato geral de uma trama [13].	6
6	Enquadramento do Protocolo.	7
7	Mecanismo de controlo de fluxo Stop-and-Wait.	7

Índice de Tabelas

Tabela 1: Caraterísticas da Placa ESP-32S - AI-THINKER [10].	4
Tabela 2: Requisitos Funcionais do projeto.	9
Tabela 3: Requisitos Não Funcionais do projeto.	9
Tabela 4: Plano de Atividades.	10
Tabela 5: Riscos inerentes ao desenvolvimento do projeto.	11

Acrónimos

A	Aluno
API	Application Programming Interfaces
ARQ	Automatic Repeat Request
ESP32	Espressif System's Programmable 32-bit microcontroller
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
PCs	Computadores Pessoais
PG	Pós-Graduação
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
TINA	Toolkit for Interactive Network Analysis
UART	Universal Asynchronous Receiver-Transmitter
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.

O presente relatório aborda o desenvolvimento de um projeto centrado na implementação de um sistema de comunicação baseado na tecnologia Light Fidelity (Li-Fi). Este projeto tem como objetivo principal criar um protótipo que permita a transmissão de informações entre dois Computadores Pessoais (PCs) por meio de um *link* ótico. O sistema é composto por componentes essenciais: um **emissor**, responsável por transmitir informações através de um LED de infravermelhos, um **recetor** capaz de receber e apresentar as informações a um utilizador, e um canal de retorno estabelecido por uma ligação elétrica.

A fase atual (Fase B - Camada de ligação de dados) foca-se no seguinte:

- O foco está na transferência confiável de dados entre as placas Espressif System's Programmable 32-bit microcontroller (ESP32) por meio do *link* ótico, concentrando-se na camada de ligação de dados. Os objetivos incluem compreender os modos de funcionamento da porta série, configurar parâmetros, entender a estrutura de tramas/pacotes e definir tipos de trama com campos específicos para delimitação, deteção de erros e controlo de fluxo. Os alunos devem especificar primitivas de serviço, aplicar o conceito de API e calcular teoricamente o tempo de transmissão de um arquivo. Na demonstração, o sistema deve transferir com sucesso um arquivo de imagem e um arquivo de texto entre dois computadores, garantindo comunicação *half-duplex*.

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 B é:

- **Estudo da Porta série:** Estudar os modos de funcionamento da porta série (normal e transparente (*raw*)) e configurar os parâmetros e o modo de funcionamento da porta série;
- **Compreensão da Estrutura do Pacote:** Compreender a estrutura de uma trama/pacote, incluindo cabeçalho, *payload* e cauda, e representar corretamente a estrutura de um pacote, identificando nome, tamanho e conteúdo de cada campo.
- **Definição das Tramas e Campos:** Definir os tipos de trama necessários para a camada 2 do projeto e identificar os campos necessários em cada tipo de trama para garantir funcionalidades como delimitação das tramas, deteção e correção de erros, e controlo de fluxo.
- **Projetar Primitivas de Serviço:** Especificar e conceber as primitivas de serviço a serem oferecidas à camada superior e especificar as interações entre os dois componentes para garantir uma comunicação eficiente.
- **Aplicação da API:** Aplicar o conceito de API para facilitar a comunicação entre os componentes.
- **Cálculo do Tempo de Transmissão:** Perceber o cálculo teórico do tempo de transmissão de um ficheiro com base nos parâmetros de transmissão, como tamanho do ficheiro, *baud rate* e tamanho máximo da trama.

2 Revisão da Literatura

Os modos de funcionamento da porta série, como normal e transparente (*raw*), são fundamentais para entender a comunicação de série de forma detalhada:

- **Modo Normal:** Este modo é utilizado para operações padrão de comunicação de série. Na configuração normal, os dados são transmitidos e recebidos de forma convencional, seguindo os protocolos estabelecidos para a porta série. É o modo mais comum e adequado para a maioria das aplicações que requerem uma comunicação de série tradicional.
- **Modo Transparente (Raw):** No modo transparente, os dados são transmitidos sem qualquer processamento adicional ou formatação na porta série. Isto significa que os dados são enviados exatamente como são, sem adição de cabeçalhos ou outros elementos que possam modificar a sua estrutura original. Este modo é útil em situações em que é necessário uma comunicação direta e sem interferências na transmissão de dados.

Ou seja, o modo normal é utilizado para operações padrão e protocolizadas na porta série, enquanto o modo transparente (*raw*) permite uma comunicação direta e sem processamento adicional dos dados, sendo útil em cenários específicos que demandam essa abordagem direta na transmissão de informações.

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 de ligação de dados, também conhecida como camada 2 do modelo OSI, é um componente crucial da comunicação de rede. Facilita a transferência de dados entre entidades da rede, garantindo a troca ordenada de informações, detecção e correção de erros e controlo de fluxo [3, 4]. Esta camada atua como ponte entre o meio físico de transmissão e a camada de rede, ditando formatos de dados, temporização e sequenciamento de bits [5]. É responsável pela criação do ambiente de rede e é essencial para o funcionamento das conexões de rede. A camada de ligação de dados também é a base para diversas tecnologias LAN, como Ethernet e token ring [5].

Application Programming Interfaces (API) são um conjunto de regras e protocolos que permitem que diferentes aplicativos de software se comuniquem entre si. Eles definem os métodos e formatos de dados que as aplicações podem usar para solicitar e trocar informações. Existem diferentes tipos de APIs, incluindo APIs *web*, APIs de sistema operacional e APIs de biblioteca [6]. As APIs da Web, em particular, são amplamente utilizadas para permitir a comunicação entre diferentes serviços da *web*. Estes podem ser criados utilizando diversas linguagens e ferramentas de programação, e sua documentação é crucial para que os programadores entendam como utilizá-los de forma eficaz [7]. Testar APIs também é importante para garantir a sua funcionalidade e desempenho [7]. APIs são comumente utilizadas no desenvolvimento de *chatbots*, onde fornecem uma forma conveniente e eficiente de acessar e integrar serviços externos [8]. No geral, as APIs desempenham um papel crucial no desenvolvimento de software moderno, permitindo a criação de aplicações inovadoras e interligadas.

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.
- **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-á uma placa ESP32, Amplificador operacional UA741CN, Fotodetetor PL-51P3C, Fotodiodo PL-53F3BT e Transístor 2N2222A para o projeto.

A protocolo Universal Asynchronous Receiver-Transmitter (UART), que será o protocolo a ser utilizado neste projeto, é uma unidade de hardware integrada em muitos microcontroladores e *chips* de comunicação. Esta unidade lida diretamente com a conversão de dados entre o formato paralelo usado internamente no dispositivo e o formato série usado para comunicação externa. O hardware UART inclui *buffers* de entrada e saída, circuitos de temporização para sincronização de bits, e circuitos para gerar e interpretar os sinais de início, dados e parada.

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 [9, 10].

3.2. HARDWARE

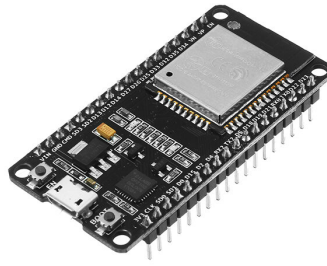


Figura 1: Placa ESP32 [11].

A Tabela 1 apresenta as características da placa em questão.

Tabela 1: Características da Placa ESP-32S - AI-THINKER [10].

Módulos	ESP - 32S - Ai-Thinker
SPI flash	32 <i>Mb (bits)</i> , 3.3 <i>V</i>
Core	ESP32-D0WD
Crystal	40 <i>MHz (Hertz)</i> (apenas para a funcionalidade do Wi-Fi e do Bluetooth)
Antena	Onboard PCB - Antenna
Dimensões (Unidade: mm)	$(48.26 \pm 0.01) \times (25.40 \pm 0.10)$

4 Arquitetura do Sistema

A arquitetura geral da Fase B 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 eletrônico.

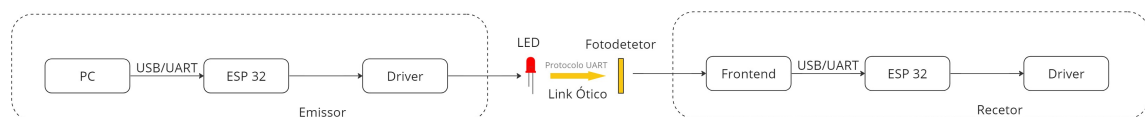


Figura 2: Arquitetura Geral da fase B.

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.

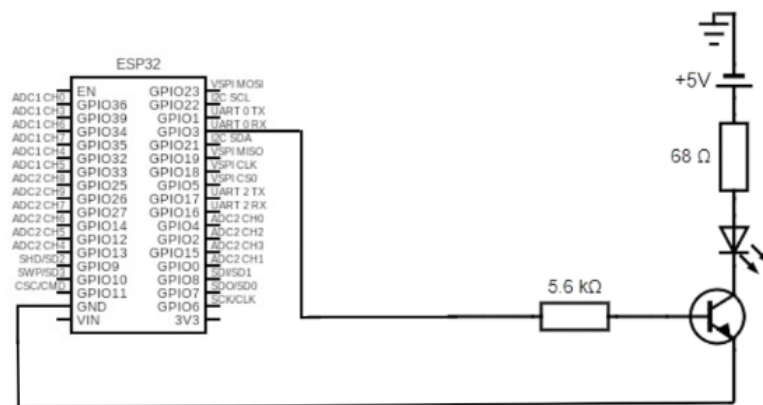


Figura 3: Pinout do Circuito Emissor.

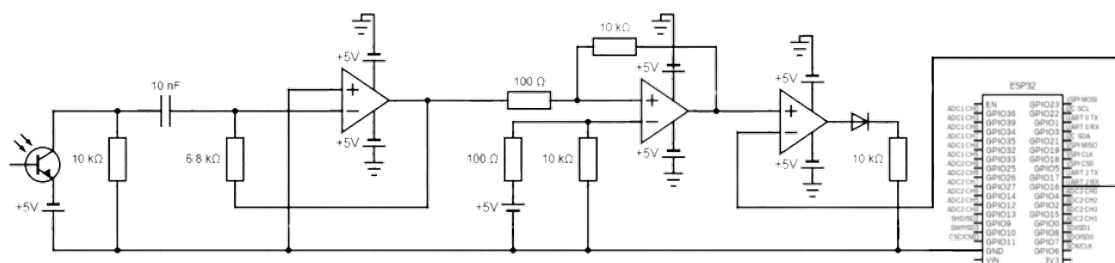


Figura 4: Pinout do Circuito Recetor.

5 Protocolo de Comunicação

Nesta secção especifica-se o tipo de mensagens e os protocolos de comunicação que é preciso utilizar nesta fase e nas seguintes.

5.1 Noção de trama/pacote

Em redes de computadores, pacote ou trama é uma estrutura unitária de transmissão de dados. A informação total a transmitir geralmente é dividida em vários pacotes e então transmitida [12].

A informação útil (*payload*) a ser transmitida é encapsulada por um cabeçalho e por uma cauda. Cada campo contém essencialmente as seguintes informações, representado na Figura 5:

- **Cabeçalho** - tamanho do *payload*, endereço do destinatário e da origem, número de sequência, etc.
- **Payload** - dados úteis a serem transmitidos.
- **Cauda** – pode conter *bits* que delimitam o fim do pacote e também *bits* para se efetuar a deteção de erros.

A vantagem de dividir um ficheiro em pacotes é que se um pacote for descartado ou perdido apenas tem que se reenviar esse pacote e não o ficheiro todo novamente, tornando assim a transmissão mais eficiente.

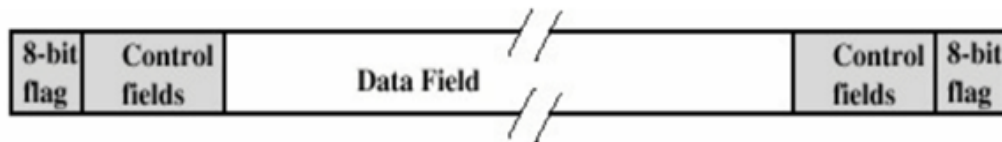


Figura 5: Formato geral de uma trama [13].

5.2 Data Link Layer

A camada de ligação de dados é fundamental para o estabelecimento e controlo da comunicação entre dispositivos de uma rede. Nesta secção, aborda-se os principais aspetos que compõem essa camada, desde o enquadramento dos protocolos de comunicação até o controlo de erros e fluxo [14].

• Enquadramento

Para o enquadramento do protocolo de comunicação, usa-se os seguintes campos, representados na Figura 6:

- **Preâmbulo**: Responsável por marcar o início da trama, assegurando assim a sincronização entre o emissor e o recetor;
- **Número de Sequência**: Valor que indica o pacote atual a ser recebido, usado para a identificação do pacote na sua receção, sendo assim possível o envio de um pacote ACK para a confirmação da mesma. O número de sequência varia de 0 a 255 (1 byte), quando atinge o valor máximo este recomeça a contagem;
- **Endereço de origem/destino**: Identifica o emissor e o recetor, como a comunicação é feita de ponto a ponto não seria necessário endereços, no entanto serão implementados por uma questão de escalabilidade. Serão usados valores inteiros entre 0 e 255 (1 byte) para identificação;
- **Tamanho real de dados**: Como opta-se por um enquadramento de tamanho fixo (para evitar erros que poderiam acontecer em enquadramentos de tamanho dinâmico), foi adicionado um campo que indica o tamanho real do *payload* sem contar com o *padding*, para que este seja mais facilmente excluído na receção do pacote;
- **Dados**
- **CRC**: Bits usados para a deteção de erros através da Verificação cíclica de redundância;

5.2. DATA LINK LAYER

Preâmbulo (8 bytes)	Número de Sequência (1 byte)	Endereço de origem (1 byte)	Endereço de destino (1 byte)	Tamanho real de dados (1 byte)	Dados (128 bytes)	CRC (4 bytes)
------------------------	---------------------------------	--------------------------------	---------------------------------	-----------------------------------	----------------------	------------------

Figura 6: Enquadramento do Protocolo.

- **Endereçamento**

No endereçamento o emissor é o dispositivo responsável por enviar os dados, enquanto o recetor é o dispositivo que os recebe. Embora a comunicação ocorra diretamente entre estes dois dispositivos e, teoricamente, não seja necessário o uso de endereços, estes serão implementados por uma questão de escalabilidade.

- **Controlo de Erros e de Fluxo**

Para o controlo de erros será usado o mecanismo Stop-And-Wait Automatic Repeat Request (ARQ), Figura 7.

Este protocolo deteta erros nos dados transmitidos através de *checksums* ou verificações de redundância cíclica. Caso seja detetado um erro, o recetor não envia um ACK ao emissor, indicando que os dados necessitam de ser retransmitidos. Além disso, o Stop-and-Wait ARQ pode ser utilizado para controlo de fluxo, permitindo que o recetor regule a taxa de transmissão de dados pelo emissor. Esta funcionalidade é útil em situações onde o recetor possui um espaço de *buffer* limitado ou recursos de processamento limitados.

O protocolo Stop-and-Wait é utilizado para garantir uma transmissão segura e eficiente de dados entre as partes envolvidas, minimizando a perda de informações e garantindo a integridade do fluxo de dados. Destaca-se pela sua abordagem simplificada de controlo de fluxo, facilidade na deteção de erros e capacidade de adaptação a diferentes velocidades de transmissão, sendo uma escolha preferencial para comunicações em curtas distâncias, onde a simplicidade e a eficiência são prioritárias.

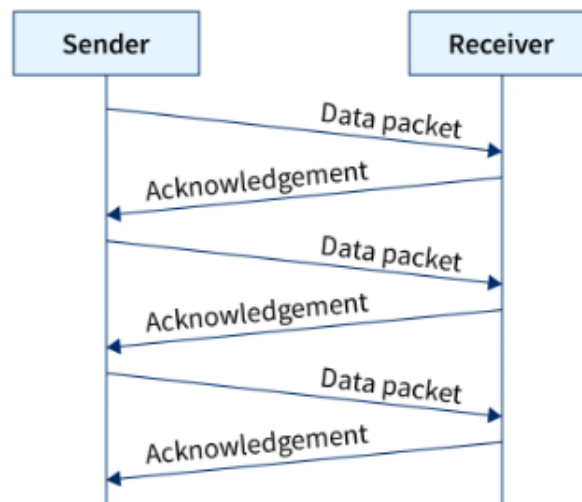


Figura 7: Mecanismo de controlo de fluxo Stop-and-Wait.

6 Análise Quantitativa

Para calcular teoricamente o tempo de transmissão de um arquivo com base nos parâmetros de transmissão, como o tamanho do arquivo, a taxa de transmissão (baud rate) e o tamanho máximo da trama, é necessário seguir a seguinte equação para determinar o tempo de transmissão:

$$\text{Tempo de transmissão} = \frac{\left(\frac{\text{Tamanho ficheiro}}{\text{Tamanho payload trama}} \right) \times \text{Tamanho total trama}}{\text{Baud rate}} + D \quad (6.1)$$

Onde:

- `Tamanho ficheiro` é o tamanho do ficheiro em bits;
- `Tamanho payload trama` é o tamanho do payload da trama em bits;
- A divisão entre o `Tamanho ficheiro` e o `Tamanho payload trama` é arredondado para o próximo número inteiro;
- `Tamanho total trama` é o tamanho total da trama em bits;
- `Baud Rate` é a taxa de transmissão em bits por segundo (bps);
- D é o atraso adicional devido ao protocolo Stop-And-Wait em segundos.

Estes cálculos básicos ajudarão a estimar o tempo teórico necessário para transmitir um arquivo com base nos parâmetros de transmissão fornecidos.

7 Requisitos Funcionais e Não Funcionais

Neste capítulo, são apresentados os Requisitos Funcionais (RF), Tabela 2, e Requisitos Não Funcionais (RNF), Tabela 3, 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 2: Requisitos Funcionais do projeto.

ID	Requisitos Funcionais	Justificação
RF1	Conhecimento dos Modos de Funcionamento da Porta Série.	Essencial para compreender como a comunicação serial ocorre entre as placas ESP32.
RF2	Configuração dos Parâmetros da Porta Série.	Permite ajustar os parâmetros de comunicação para garantir a transferência adequada de dados.
RF3	Compreensão da Estrutura de um Pacote.	Fundamental para manipular corretamente os dados transmitidos, incluindo cabeçalho, payload e cauda.
RF4	Definição dos Tipos de Trama e Campos Necessários.	Para assegurar a delimitação das tramas, detecção e correção de erros, e o controle de fluxo na camada 2 do projeto.
RF5	Especificação das Primitivas de Serviço e Interações entre Componentes.	Define como a camada superior interage com a camada 2, estabelecendo as funcionalidades oferecidas.
RF6	Implementação da API e Cálculo do Tempo de Transmissão Teórico	Facilita a comunicação entre os componentes e permite estimar o tempo de transmissão com base nos parâmetros.

Tabela 3: Requisitos Não Funcionais do projeto.

ID	Requisitos Não Funcionais	Justificação
RNF1	Implementação do Protocolo de Nivel 2 em Modo Half-Duplex.	Garante que a comunicação ocorra em ambas as direções, mas não simultaneamente.
RNF2	Utilização de Funções Conhecidas para Comunicação entre Camadas	Facilita a integração entre a camada 2 e o restante código através de uma API bem definida.
RNF3	Garantia de Transmissão Fiável em Caso de Erros	A camada 2 deve lidar com erros de transmissão, garantindo a integridade dos dados mesmo em situações adversas.
RNF4	Capacidade de Timeout e Recuperação da Transmissão em Caso de Interrupção Breve	Permite que o sistema retome a transmissão no ponto exato após uma interrupção temporária na comunicação.

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.

8 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.

8.1 Atividades

Na Tabela 4, 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 4: Plano de Atividades.

ID	Atividade	Início	Conclusão
1	Fase B - Camada de ligação de dados	13/03/2024	16/04/2024
1.1	Estudo e Pesquisa	13/03/2024	20/03/2024
1.1.1	Conhecimento dos Modos de Funcionamento da Porta Série.	13/03/2024	20/03/2024
1.1.2	Compreensão da Estrutura de um Pacote.	13/03/2024	20/03/2024
1.1.3	Configuração dos Parâmetros da Porta Série.	15/03/2024	20/03/2024
1.1.4	Definição dos Tipos de Trama e Campos Necessários.	15/03/2024	20/03/2024
1.1.5	Cálculo Teórico do Tempo de Transmissão.	15/03/2024	20/03/2024
1.2	Relatório de Especificação	19/03/2024	20/03/2024
1.2.1	Preparar esquemas detalhados do emissor e do recetor, indicando os componentes utilizados e os seus valores.	19/03/2024	19/03/2024
1.2.2	Entregar o relatório de especificação.	20/03/2024	20/03/2024
1.3	Projeto e Implementação	20/03/2024	16/04/2024
1.3.1	Implementação da Transferência de Arquivos.	20/03/2024	03/04/2024
1.3.2	Implementação da Camada 2 em Comunicação Half-Duplex.	02/04/2024	10/04/2024
1.3.3	Garantia da Transmissão Fiável dos Dados.	02/04/2024	10/04/2024
1.3.4	Gestão de Timeout e Recuperação da Transmissão.	10/04/2024	16/04/2024
1.4	Entrega da Fase B	16/04/2024	16/04/2024

8.2 Lista de Riscos

Nesta secção identifica-se e descreve-se os riscos potenciais associados ao projeto, Tabela 5. 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.

8.2. LISTA DE RISCOS

Tabela 5: Riscos inerentes ao desenvolvimento do projeto.

ID	Risco	Mitigação	P	I	S
R1	Enquadramento incorreto de pacotes de dados, levando à corrupção de dados.	Garantir a estruturação adequada dos pacotes e a verificação de erros.	4	5	20
R2	Falha ao estabelecer comunicação confiável devido a problemas de tempo limite.	Implementar mecanismos de tempo limite para retransmissão.	3	4	12
R3	Perda de pacotes de informação durante a transmissão.	Implementação de detecção de erros e mecanismos de correção.	3	4	12
R4	Configuração inadequada dos parâmetros da porta série.	Compreender e definir completamente as configurações da porta série.	2	3	6
R5	Compreensão inadequada das interações de API entre camadas.	Definir e implementar minuciosamente as funções da API.	2	3	6

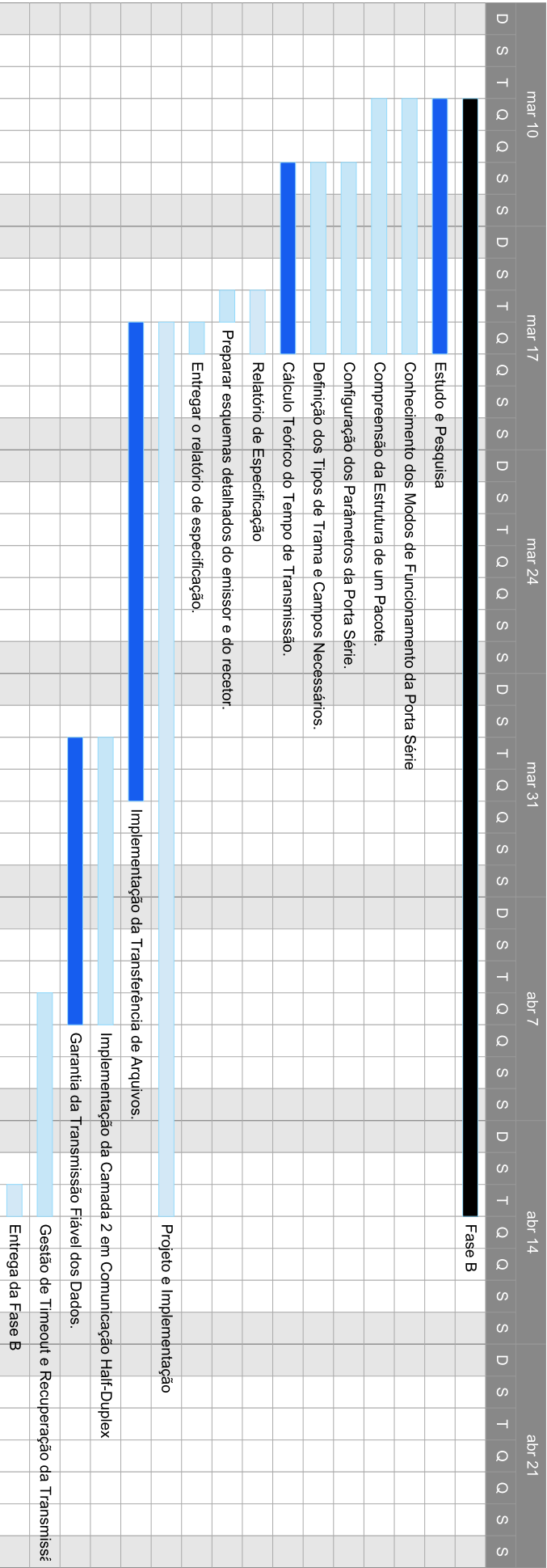
9 Conclusão

Ao longo deste relatório, foram abordados diversos aspetos essenciais, desde a arquitetura do sistema até a análise quantitativa dos circuitos, passando pelos requisitos funcionais e não funcionais, o plano de atividades e a identificação dos riscos associados ao projeto.

O foco está em adquirir e aprofundar diversas competências, tanto académicas quanto pessoais. Para esta etapa específica, é crucial consolidar conceitos previamente aprendidos em disciplinas anteriores e aplicá-los de forma prática. São essenciais habilidades em protocolos de comunicação para implementar o controlo de fluxo e desenvolver tramas para diferentes tipos de mensagens. Além disso, é fundamental possuir conhecimentos em desenvolvimento de aplicações para facilitar a troca de dados entre emissor e recetor. Na área da eletrónica, será necessário ajustar o circuito frontend para alinhar os valores de tensão com os da placa ESP32. Em termos pessoais, a capacidade de identificar e resolver problemas, juntamente com o pensamento crítico, são competências-chave para o sucesso do projeto. Coletivamente, como equipa, procura-se aprimorar a comunicação e a criatividade ao longo da execução do projeto, visando um desenvolvimento notável e uma entrega eficaz.

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/IJARST-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] W. Buchanan. "High-level Data Link Control (HDLC)". Em: Springer US, 1996, pp. 111–126. DOI: 10.1007/978-1-4613-1207-9_6.
- [5] Michael A. Gallo e William M. Hancock. "Data Link Layer Concepts and IEEE Standards". Em: Elsevier, 2002, pp. 181–212. DOI: 10.1016/B978-155558252-4/50031-6.
- [6] Erik Tollerud. *Wayback Machine*. Mar. de 2016. URL: https://web.archive.org/web/20200213224625/https://zenodo.org/record/48447/files/API_lightning.pdf.
- [7] Ana Karoliny Silva De Araujo e Jose Ferdinandy Silva Chagas. "Um estudo sobre boas práticas para documentação de APIs". Em: *Brazilian Journal of Development* 9 (05 mai. de 2023), pp. 14632–14660. ISSN: 2525-8761. DOI: 10.34117/bjdv9n5-011.
- [8] REJANE FROZZA e YAN PABLO RECKZIEGEL RODRIGUES. "Utilização de APIs no desenvolvimento de chatbots: Análise e comparação". Em: nov. de 2020. DOI: 10.14488/ENECEP2020_TN_STO_351_1805_40098.
- [9] 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.
- [10] <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.
- [11] 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.
- [12] V. Cerf e R. Kahn. "A Protocol for Packet Network Intercommunication". Em: *IEEE Transactions on Communications* 22.5 (1974), pp. 637–648. DOI: 10.1109/TCOM.1974.1092259.
- [13] Maria João Nicolau. *Conceitos básicos sobre transmissão de dados*. Blackboard. Acedido em 27 de março de 2023. 2022.
- [14] José Augusto Afonso. "Data Link Layer". Em: *Blackboard* (2023). Acedido em 20 de março de 2024.



Anexo I - Diagrama de Gantt