



**UnB**

Departamento de  
Ciência da Computação

Marcelo Antonio Marotta  
Teleinformática e Redes 1  
Trabalho FINAL

# 1 Enunciado

Simular o funcionamento da camada de enlace e camada física por meio da implementação dos protocolos de enquadramento, modulação banda-base e modulação por portadora. Os slides que estão disponíveis no moodle possuem os diagramas e exemplos de como cada etapa deve ser desenvolvida. Os grupos formados no trabalho devem ser mantidos.

Com o intuito de facilitar o entendimento dos protocolos da camada de enlace de dados e camada física, o trabalho foi dividido em etapas e subetapas.

## 1.1 Camada Física

### 1.1.1 [1/2]Modulação digital

Simular o funcionamento da camada física por meio da implementação das seguintes modulações digitais:

- Non-return to Zero Polar (NRZ-Polar);
- Manchester; e
- Bipolar.

Os slides que estão disponíveis no moodle no tópico Modulação possuem o diagrama e exemplos de como cada modulação deve ser desenvolvida.

### 1.1.2 [2/2]Modulação por portadora

Simular o funcionamento da camada física por meio da implementação das seguintes modulações por portadora:

- Amplitude Shift Keying (ASK);
- Frequency Shift Keying (FSK); e
- Phase Shift Keying (PSK - QPSK); e
- 16-Quadrature Amplitude Modulation (16-QAM).

Os slides que estão disponíveis no moodle no tópico Modulação possuem o diagrama e exemplos de como cada modulação deve ser desenvolvida.

## 1.2 Camada de Enlace

### 1.3 [1/3] Camada de Enlace

Acrescentar ao código do Trabalho Prático I, os protocolos de enquadramento de dados:

- Contagem de caracteres;
- Enquadramento com FLAGS e inserção de bytes ou caracteres; e
- Enquadramento com FLAGS Inserção de bits;

### 1.4 [2/3] Camada de Enlace

Acrescentar ao código da subetapa anterior, os protocolos de detecção de erros:

- Bit de paridade par;
- Checksum (Como o apresentado em sala de aula);
- CRC (polinômio CRC-32, IEEE 802).

### 1.5 [3/3] Camada de Enlace

Acrescentar o protocolo de correção de erros:

- Hamming.

## 2 Material a ser entregue

Relatório com no mínimo 3 páginas, contendo:

- Capa: Deve conter possuir as seguintes informações:(i)nome do simulador, e (ii) nome dos membros do Grupo.
- Introdução: Descrição do problema a ser resolvido e visão geral sobre o funcionamento do simulador.
- Implementação: Descrição detalhada do desenvolvimento com diagramas ilustrativos, o funcionamento dos protocolos, procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.
- Membros: Descrição das atividades desenvolvidas por cada membro do grupo.
- Conclusão: Comentários gerais sobre o trabalho e as principais dificuldades encontradas no desenvolvimento do simulador.

Código fonte com os seguintes arquivos:

- CamadaFisica: Implementações das funções declaradas para camada física
- CamadaEnlace: Implementações das funções declaradas para camada de enlace

- InterfaceGUI: Implementações das funções necessárias para entrada de dados e resultados gráficos obtidos. (NÃO É UMA TELA DE TERMINAL - Utilize bibliotecas linux para GUI. Preferência: GTK<sup>1 2</sup>)
- Simulador: Código com a rotina principal chamadora das demais para simular as camadas de redes.

Obs.: Diferenciar quando for transmissor ou receptor.

O relatório e o código fonte devem ser submetidos compactados (.zip) no Moodle. Serão aceitos relatórios em Jupyter.

### 3 Critérios de Avaliação

O trabalho será pontuado de acordo com a implementação e os critérios da Tabela 1. Código com falta de legibilidade e modularização pode perder ponto conforme informado na Tabela 1. Erros gerais de funcionamento, lógica ou outros serão descontados como um todo.

| Ítem                         | Quesitos  | Max Pontos |
|------------------------------|---|------------|
| Relatório                    | Documento PDF contendo todas as informações sobre o trabalho  | +2         |
| Código e execução            | O projeto compilou e executou corretamente  | +2         |
| Resultado                    | Saídas corretas de acordo com os protocolos implementados   | +3         |
| Conceitos de TR 1            | Código fonte implementados adequadamente  | +3         |
| Legibilidade e Modularização | Pode perder pontos caso não faça:<br>-Uso de comentários<br>-Indentação do código<br>-Uso de funções inadequadas (duplicada/redundante/não atingível<br>-Uso das declarações e implementações dos arquivos(.py, .hpp e .cpp) para o Simulador | -10        |
| Atraso                       | Perde 1 ponto para cada dia de atraso da data estabelecida (Max 5)  | -1         |
| Plágio                       | Cópia de qualquer forma   | -10        |

Tabela 1: Critérios de avaliação

### 4 Ferramentas

A implementação do trabalho será na linguagem C++ ou Python. Pode-se utilizar qualquer IDE/compilador para o desenvolvimento contanto que execute sem problemas no Linux. Pode ser desenvolvido em GNU-RADIO. **NÃO SERÃO ACEITOS DESENVOLVIMENTOS COM IMPORTAÇÃO DIRETA DE BIBLIOTECAS EXTERNAS, POR EXEMPLO, UTILIZAR A ZLIB PARA IMPLEMENTAR CRC.**

Siga o diagrama de desenvolvimento na Figura 1 para guiar o desenvolvimento do trabalho.

### 5 Informações Importantes

Cada grupo deverá desenvolver o trabalho e cada membro do grupo deverá conhecer e dominar todos os trechos de código gerados. Os grupos deverão desenvolver o projeto de maneira independente para não haver cópia ou compartilhamento de código. O projeto irá passar por um verificador automático de plágio. Os projetos detectados como plágio receberão nota zero, independente do grupo. Dessa forma, fica a cargo do grupo proteger o projeto contra cópias

<sup>1</sup>Python GTK: <https://python-gtk-3-tutorial.readthedocs.io/en/latest/>

<sup>2</sup>C++ GTK+ - <https://www.gtk.org/docs/language-bindings/cpp>

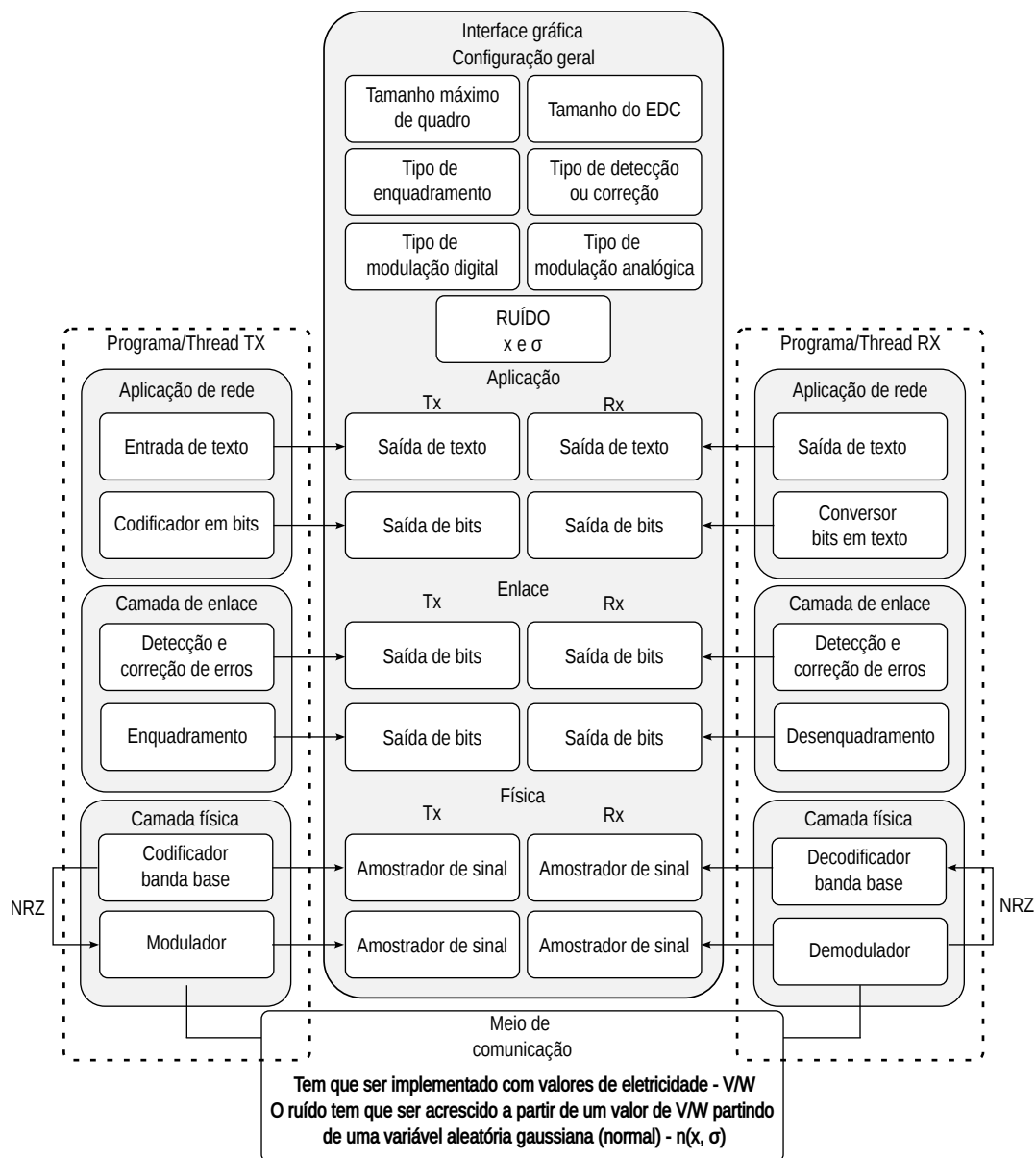


Figura 1: Diagrama de desenvolvimento

## 6 Datas

As datas e local da entrega do trabalho estarão disponíveis a partir do Moodle da Disciplina.