Instituto Superior de Engenharia de Lisboa Licenciatura em Engenharia Informática e de Computadores 2022 / 2023 Unidade Curricular de Comunicação Digital Docente Vítor Fialho

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores**

**Módulo 1**

Autores: 48337 Daniel Antunes

48292 Tiago Neves

46080 Sérgio Capela

Relatório para a Unidade Curricular de Comunicação Digital de Informação da Licenciatura em Engenharia Informática e de Computadores

09-05-2023

Conteúdo

[Introdução 3](#_Toc134565153)

[Desenvolvimento 4](#_Toc134565154)

[Resultados experimentais 6](#_Toc134565155)

[Conclusão 10](#_Toc134565156)

# Introdução

Neste trabalho prático tínhamos como objetivo o desenvolvimento de programas e aplicações em “Python”.

O trabalho encontra-se dividido em 5 exercícios com um total de 12 alíneas onde foi necessário aplicar diferentes conceitos lecionados em aula.

Para melhor compreensão do nosso trabalho, o código enviado em anexo apresenta-se comentado.

# Desenvolvimento

* 1. **4**
     1. Neste exercício tivemos de fazer duas funções, ambas recebem um inteiro, a primeira retorna a quantidade de bits a 0 no mesmo numero em binário e a segunda o numero de bits a 1.
     2. Neste exercício tivemos de fazer uma função que recebe um valor inteiro, a função retorna como caracteres os valores dos bits de todos os valores inteiros desse inteiro.
     3. Neste exercício tivemos de fazer uma função que recebe um ficheiro, a função retorna o símbolo mais frequente presente nesse ficheiro imprimindo na consola o número de vezes que o mesmo aparece.
     4. Neste exercício tivemos de fazer uma função que recebe dois ficheiros, um de input outro de output, a função transforma o primeiro ficheiro recebido no segundo negando cada bit do primeiro.

**5**

1. Neste exercício tivemos de fazer uma função onde retornamos os primeiros *N* termos da progressão geométrica de primeiro termo *u* e razão *r*. *N*, *u* e *r* são passados como parâmetro.
2. Neste exercício tivemos de fazer uma função onde retornamos o máximo divisor comum entre dois números inteiros através do algoritmo de Euclides.
3. Neste exercício tivemos de fazer uma função onde retornamos o símbolo mais e menos frequente de um ficheiro passado como parâmetro, a frequência de ambos os símbolos é passada na consola.
4. Neste exercício tivemos de fazer uma função onde retornamos o histograma de um ficheiro, o valor da informação própria de cada símbolo e a entropia do ficheiro.
5. **a)** Neste exercício tivemos de fazer uma função onde retornamos o histograma de um ficheiro, o valor da informação própria de cada símbolo e a entropia do ficheiro.
6. **a)** Neste exercício tivemos de fazer uma função onde retornamos o histograma de um ficheiro, o valor da informação própria de cada símbolo e a entropia do ficheiro.
7. **a)** Neste exercício tivemos de fazer uma função que retorna uma cifre de Vernam e recebe como parâmetros o texto em claro (plainText) e a chave, tanto os parâmetros de entrada como o valor de retorno têm tamanho do texto em claro.
8. **a)** Neste exercício tivemos de fazer uma função que implementa o modelo *Binary Symmetric Channel* (BSC), a função recebe: uma sequência binária, o valor pretendido *p* para a BER passado como parâmetro com o valor real obtido.

# Resultados experimentais

**2. a)** **A picture containing text, screenshot, diagram, plot

Description automatically generated**

Figura 1 - Histograma do ficheiro a.txt

Entropia do ficheiro a.txt: 4.86863218432445

Número de símbolos: 156

A picture containing text, screenshot, diagram, line

Description automatically generated

Figura 2 - Histograma do ficheiro alice29.txt

Número de símbolos: 148482

Entropia do ficheiro alice29.txt: 4.512862347015978

**3.** Para a resolução deste exercício criámos a função symbCreation que recebe com parâmetros o tamanho da sequência pretendido (N), a função massa probabilidade (M) e uma flag(word) que determina se é pretendido o retorno de palavras ou apenas de uma sequência de símbolos. Em caso da flag estar com o valor “True” então são realizadas as palavras com o tamanho variável entre 8 e 12.

**b)** Para z = {‘a’: 0.3, ‘b’: 0.5, ‘c’: 0,1, ‘d’: 0.1} e n = 10 é escrito no ficheiro o exemplo: dcaabbbbbb; H(x) = 3.8

Para o mesmo z e n = 5 é escrito no ficheiro o exemplo: abaab; H(x) = 1.9

Podemos assim observar que a entropia é menor para um n menor.

**c)** Para as dimensões pretendidas no exercício obtemos os seguintes exemplos: ['adbdabbbbbb', 'ccbbbcab', 'cbdbbbbbb', 'baabbaabbb', 'babbbbbb']

**4.** Para a realizar este exercício fizemos as funções makeVernamCypher e cypherBigText.

A função makeVernamCypher é responsável pela cifra e decifra das sequências de símbolos recebidos, fazendo um XOR bit a bit de todos os caracteres, entre a sequência e a chave passadas como parâmetros.

A função cypherBigText implementa a função makeVernamCypher para grandes textos e escreve esses textos num novo ficheiro.

**a)**

**b) **

Figura 3 - Texto cifrado e decifrado com chave constante

****

Figura 4 - Entropia do ficheiro alice29.txt cifrado com chave constante

****

Figura 5 - Entropia do ficheiro alice29.txt cifrado com chave aleatória

**** A partir destas demonstrações podemos concluir que a cifra com chave constante realiza ficheiros com entropias próximas ou iguais aos ficheiros em claro.

Figura 6 - Entropia do ficheiro alice29.txt em claro

**5.** Para a resolução deste exercício foram realizadas duas 4 funções principais e 2 funções auxiliares.

As funções “string\_para\_binario” e “binário\_para\_string” são “auto-esclarecedoras” pois fazem aquilo que o seu nome indica. Estas são as funções auxiliares.

As funções “binarySymmetricChannel”, “interleaving”, “deInterleaving” e “interleavingBSC” são as funções principais do ficheiro.

binarySymmetricChannel simula o modelo BSC apresentado no enunciado do trabalho. Este realiza a operação de XOR bit a bit tendo em conta a BER passada como parâmetro. Devido a dificuldades ocorridas na realização do trabalho, em código, não usamos a operação XOR, porém o resultado é o pretendido. Esta decisão acontece, pois, com a utilização da operação XOR bit a bit estávamos a obter resultados errados.

interleaving simula a técnica de entrelaçamento das sequências de símbolos passadas como parâmetros, a matriz realizada é configurável tendo em conta o tamanho da sequência.

deInterleaving reverte a função interleaving pelos mesmos métodos da passada função.

interleavingBSC junta todas as funções num processo completo de leitura, conversão, armazenamento e escrita.

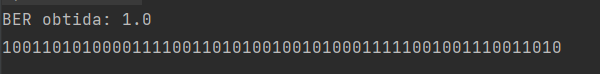
**a)  **

Figura 7- Chamada da função binarySymetricChannel com BER=1 e sequencia binaria

Figura 8 - Resultados obtidos da função e BER real obtida

**b)**

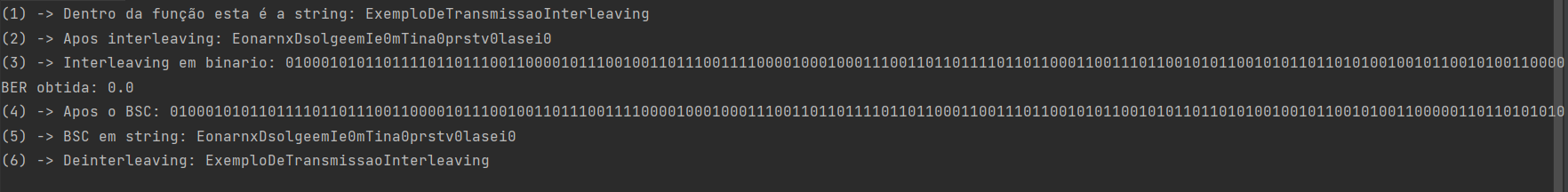
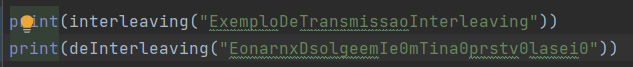
** **

Figura 9 - Funcionamento da função interleavingBSC

Figura 10 - Resultados obtidos das chamadas das funções interleaving e deInterleaving

# Conclusão

Com a realização deste trabalho, o grupo foi capaz de consolidar os conhecimentos obtidos em aula sobre as matérias lecionadas assim como realizar exercícios práticos que forneceram suporte no estudo para o primeiro teste da disciplina.