

---

# TÉCNICAS DE COMPRESSÃO **NÃO DESTRUTIVA**

Teoria da informação

---



UNIVERSIDADE D  
COIMBRA



- 01. POSSÍVEL SOLUÇÃO PARA COMPRESSÃO NÃO DESTRUTIVA**
- 02. RESULTADOS DA INVESTIGAÇÃO**
- 03. CONCLUSÃO E CONSIDERAÇÕES FINAIS**



UNIVERSIDADE DE  
COIMBRA

## NOME DO DOCENTE

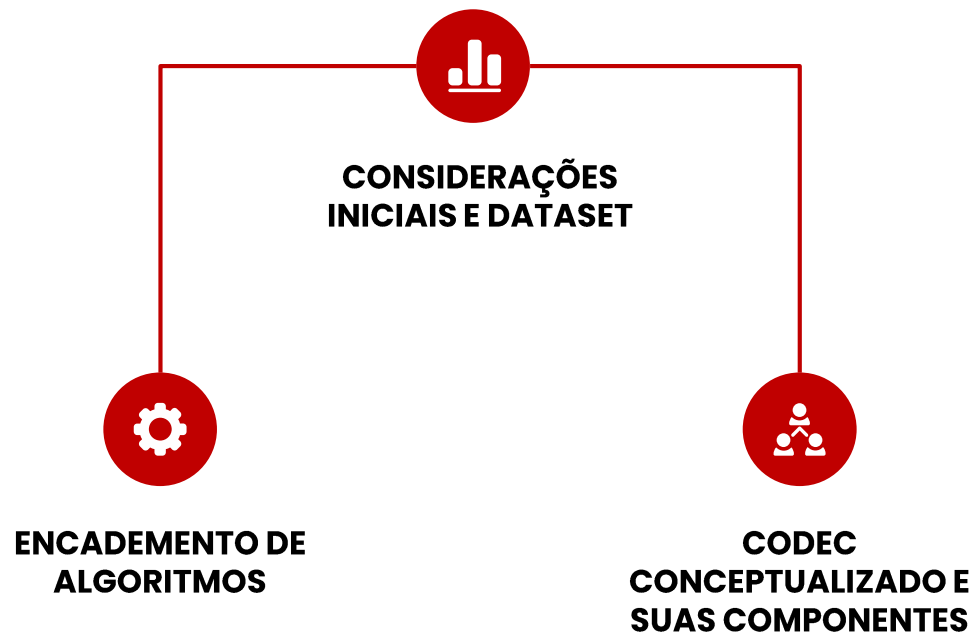
Paulo de Carvalho

Este trabalho foi-nos proposto pelo docente Paulo de Carvalho no âmbito da disciplina Teoria de Informação com o intuito de colocarmos os conhecimentos anteriormente adquiridos em prática com a finalidade de criação de um novo algoritmo de compressão.



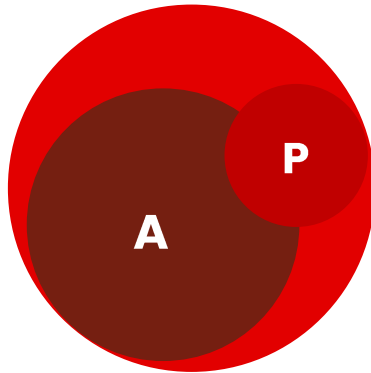
01

**POSSÍVEL SOLUÇÃO PARA COMPRESSÃO NÃO DESTRUTIVA**



# ENCADEAMENTO de algoritmos

Possível solução para compressão não destrutiva



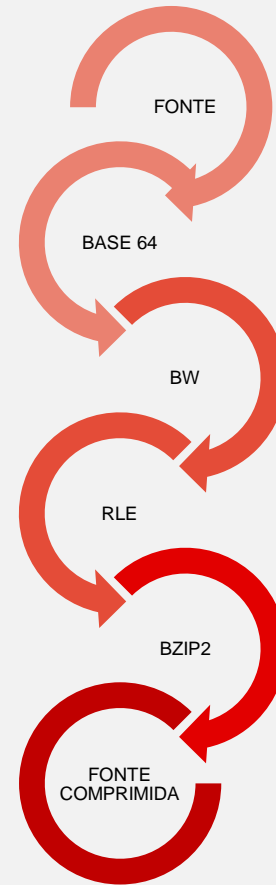
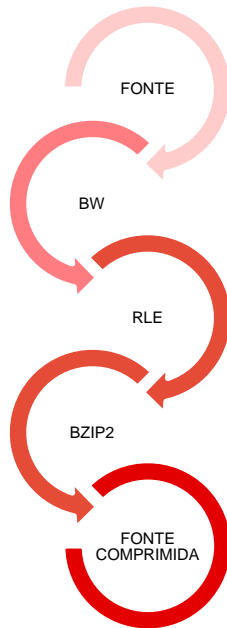
- Codificação entrópico
- Preditores
- Algoritmos das etapas de dependência estatística

## DATASET

File Name	Data type	Size (Bytes)	Description
war_and_peace	Text in english (.txt) format	3 359 549	The Project Gutenberg EBook of War and Peace, by Leo Tolstoy
cromenco_c10	Image File (.bmp) format	33 987 318	Monochromatic image of a chromenco computer and other devices

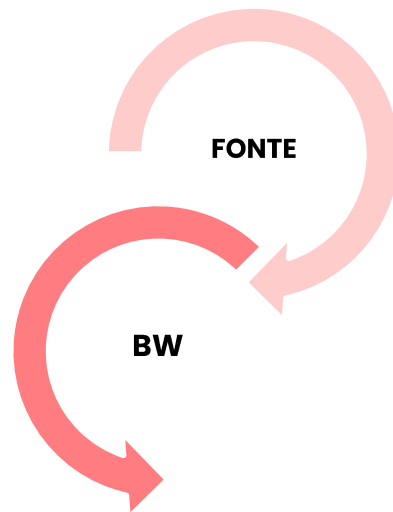
# CODEC CONCEITUALIZADO

E as suas componentes



## 1ª etapa (BW)

- Facilita a compressão
- Organiza a fonte (permite explorar a redundância)
- **bastante eficiente em fontes de texto**



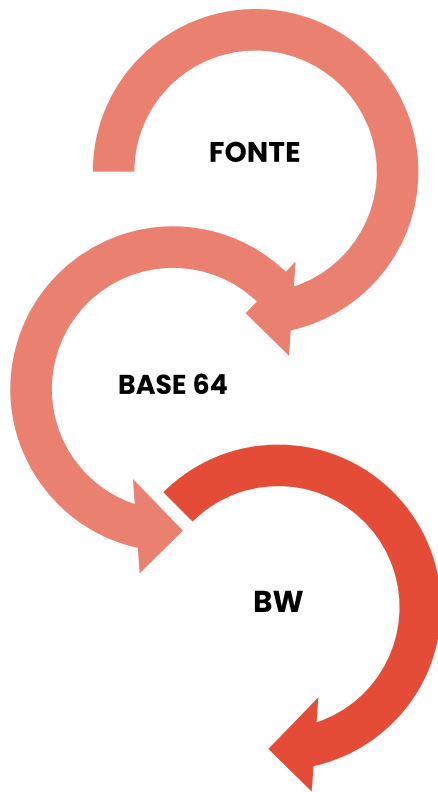
## PROBLEMA:

- Transformada, pensada e construída para fontes textuais



## Base 64

- Permite representar uma imagem como sequência de caracteres;
- Cada caractere é codificado em 6 bits;
- A fonte de informação é vista como bits (esta está agrupada em conjuntos de 8 bits, ou seja, bytes);
- Cria-se agrupamentos de 6 bits a começar pela esquerda;
- Verifica-se a correspondência desses 6 bits com um caractere da tabela;
- Escreve-se esse carácter na mensagem a ser transmitida;

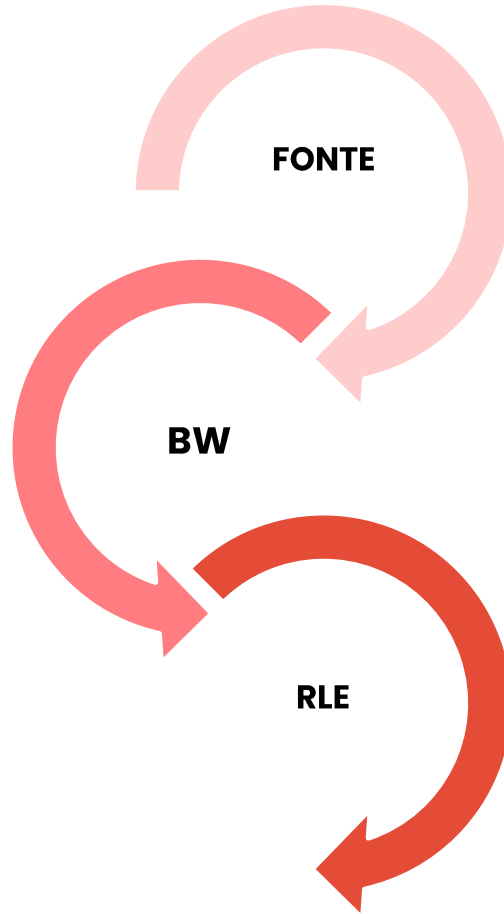


## SUMA:

- Aumenta a redundância da fonte;
- É introduzido algum overhead no ficheiro

## 2ª etapa (RLE)

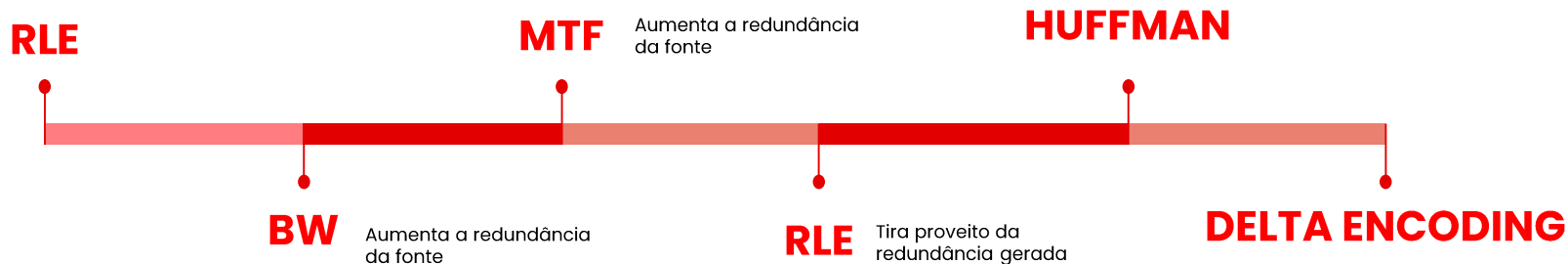
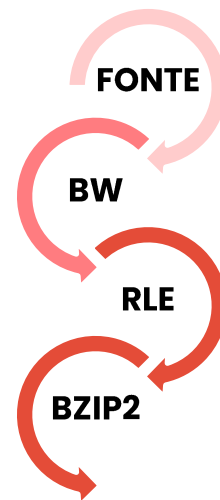
- Ao aplicar este algoritmo a fontes de texto com conjuntos de símbolos sequenciais iguais conseguimos eliminar o overhead produzido e comprimir a fonte de dados



- Dados redundantes são colmatados com o uso deste algoritmo

## 3ª etapa (BZIP2)

- Algoritmo poderoso e eficiente na compressão de texto e imagem





02

**RESULTADOS DA INVESTIGAÇÃO**

**Linguagem de programação usada**

**Python 3.7**

**Fontes**

**Wikipédia**

**Github**

**Sites**

**Livraria standard**

**Python**

**Encadeamento**

**Funções feitas por nós  
que permitem o uso  
dos diversos  
algoritmos e o seu  
encadeamento.**

**Especificações do computador usado  
para obtenção de resultados**

**Intel i7-8750H (12 cores) processador  
(clock speed de 2.200GHz), 16GB de  
RAM e correndo o sistema operativo  
Linux (Ubuntu 19.10 x86\_64)**

Algorithm	Text Original Size (bytes)	Text Final Size (bytes)	Ratio (txt)	Image Original Size (bytes)	Image Final Size (bytes)	Ratio (img)
Bzip2	3 359 549	8 884 54	3,78	33 987 318	7 717 164	4,40
PPM	3 359 549	14 398 61	2,33	33 987 318	8 205 563	4,14
PNG	- - - - -	- - - - -	- - -	33 987 318	12 772 549	2,66
BW	3 359 549	3 473 091	0,96	33 987 318	45 729 625	0,74
BW-RLE	3 359 549	3 456 008	0,97	33 987 318	32 302 300	1,05

# Tab3.

Resultados obtidos após aplicação direta destes algoritmos no Dataset fornecido

Algorithm	Text Original Size (bytes)	Text Final Size (bytes)	Ratio (txt)	Image Original Size (bytes)	Image Final Size (bytes)	Ratio (img)
BW-RLE-PPM	3 359 549	1 965 284	1.70	33 987 318	18 893 541	1.80
PPM-Bzip2	3 359 549	1 446 760	2.32	33 987 318	10 939 387	3.11
PNG-Bzip2	- - - - -	- - - - -	- - -	33 987 318	12 272 731	2.77
PNG-PPM	- - - - -	- - - - -	- - -	33 987 318	12 234 415	2.78
BW-Bzip2	3 359 549	1 954 283	1.72	33 987 318	17 342 202	1.96
Bzip2-PPM	3 359 549	944 779	4.37	33 987 318	7 785 234	3.55

# Tab4.

Resultados obtidos após aplicação de algumas combinações dos algoritmos anteriores no Dataset fornecido

Algorithm Steps	Text Start Size (bytes)	Text Final Size (bytes)	Ratio (txt)	Image Start Size (bytes)	Image Final Size (bytes)	Ratio (img)
BW	<b>3 359 549</b>	3 473 091	0.96	<b>33 987 318</b>	45 729 625	0.74
BW→ RLE	3 473 091	3 456 008	0.97	45 729 625	32 302 300	1.05
BW→RLE→ Bzip2	3 456 008	<b>1 952 669</b>	<b>1.72</b>	32 302 300	<b>17 502 354</b>	<b>1.94</b>

# Tab5.

Resultados parciais e finais obtidos após aplicação do nosso algoritmo no Dataset fornecido





# 03

## CONCLUSÃO E CONSIDERAÇÕES FINAIS



## AUMENTO DA REDUNDÂNCIA

Diminuição da taxa de  
compressão

0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

## BASE 64

Agrupa sequências de 8 bits  
em sequências de 6 bits



# FIM

**Universidade de Coimbra**

Gabriel Fernandes

Maria Dias

Pedro Rodrigues

[www.uc.pt](http://www.uc.pt)