**Contagem de frequência de Bytes em diversos tipos de arquivo**

**CEFET/RJ**– Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

**Disciplina de Organização e Estrutura de Arquivos**

Natália Nunes Vieira

Resumo: Este estudo tem como objetivo realizar uma análise sobre métodos de compressão de arquivos, onde serão analisadas as frequências de bytes ocorrentes em um arquivo antes e depois dos mesmos serem compactados como *.zip* ou *.rar*.

**Introdução**

O trabalho consiste em elaborar um código que seja capaz de contar a frequência de Bytes de um arquivo, depois compactar estes arquivos e realizar uma nova contagem de frequência. Identificar o grau de compressão de cada um deles e redigir uma documentação com o que foi analisado. Para isso, o desenvolvimento foi divido em três fases:

**Fase 1 (um):**

Desenvolver um algoritmo que seja capaz de ler cada Byte de um arquivo e contar a frequência em que eles ocorrem.

O algoritmo precisa:

* Ler uma pasta com diversos tipos de arquivo;
* Ler cada arquivo da pasta individualmente;
* Ler cada Byte do arquivo;
* Identificar qual Byte foi lido;
* Incrementar uma lista com a frequência que cada Byte aparece no arquivo;
* Escrever a lista na tela.

Linguagem: Python

**Fase 2 (dois):**

Compactar os arquivos utilizando formatos já conhecidos.

Será preciso:

* Escolher pelo menos dois formatos de compactação;
* Compactar cada um dos arquivos;
* Identificar o tamanho anterior e posterior à compactação.

Formato dos arquivos: .png, .jpeg, .txt, .pdf, .mp3, .mp4.

Formatos para compactação: *.rar* e *.zip*

**Fase 3 (três):**

Comparar a frequência de Bytes dos arquivos antes e depois de serem compactados.

Será preciso:

* Realizar a contagem da frequência de Bytes do arquivo antes da compactação;
* Realizar a contagem da frequência de Bytes do arquivo após a compactação;
* Identificar qual a diferença da contagem entre os dois testes;
* Redigir um documento com o que for identificado entre os arquivos.

**Desenvolvimento**

**Fase 1 (um) - Implementação:**

O algoritmo foi desenvolvido na linguagem de programação Python 2, utilizando a IDE do *Visual Studio Code*. O código encontra-se disponível no *Github* descrito no rodapé do relatório.

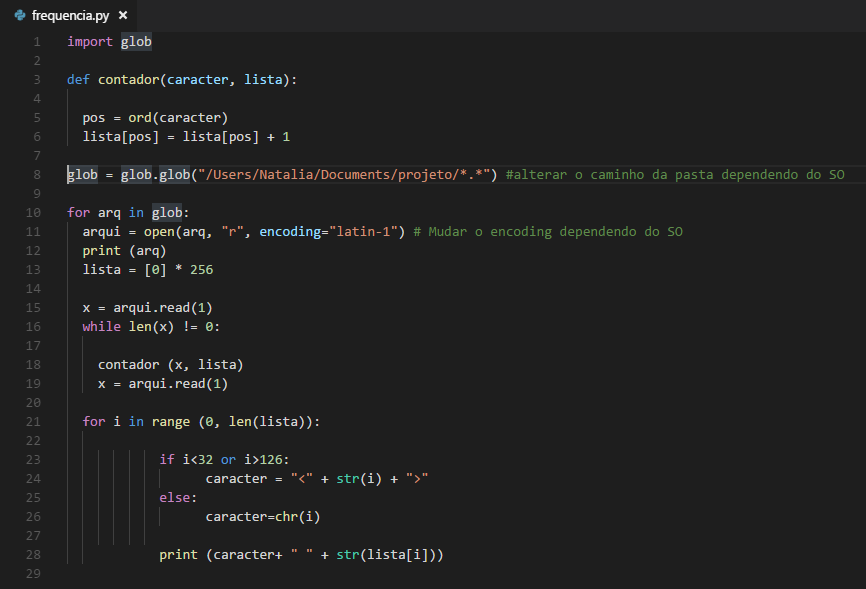


Figura 1: Contador de frequência de Bytes

Para o código da Figura 1 é importante ressaltar que, para que os testes ocorram com precisão, é necessário alterar o caminho da pasta onde estarão os arquivos para teste, e também verificar o *encoding* utilizado por cada sistema operacional.

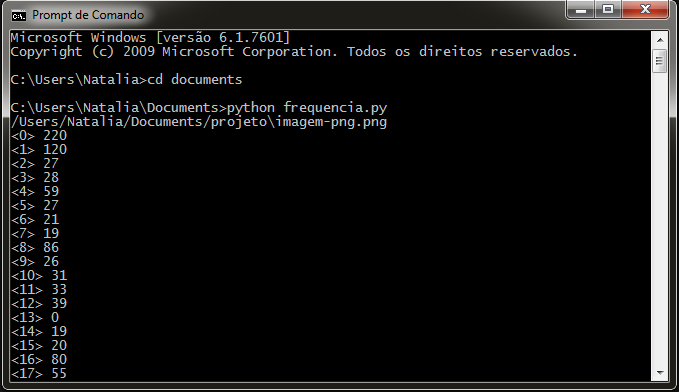


Figura 2: Exemplo de saída no console

**Fase 2 (dois) - Formatos:**

Para a realização dos testes, foram escolhidos os formatos de arquivo *.txt*, .*pdf*, *.jpeg*, *.png*, *.mp3*, *.mp4*. E os formatos de compactação .zip e .rar.

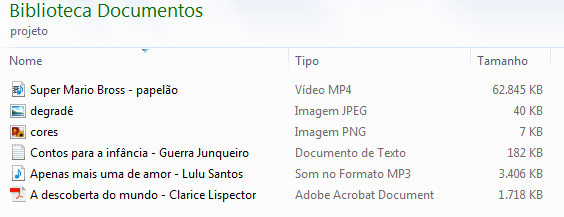


Figura 3: Arquivos nos formatos selecionados

Para cada arquivo foram utilizados os padrões de compactação abaixo descritos nas Figuras 4 e 5.

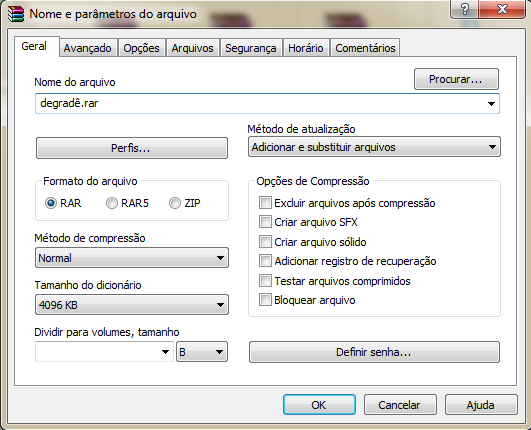


Figura 4: Especificações de compactação *.rar*

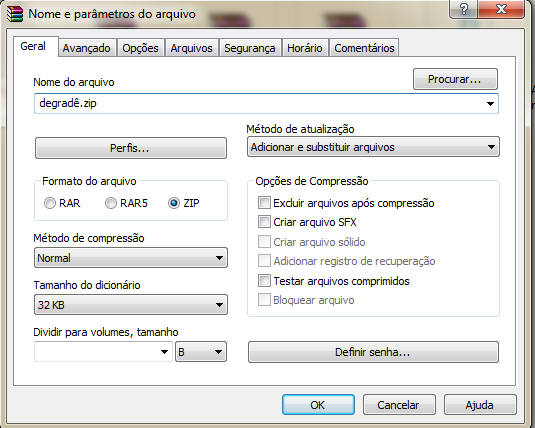


Figura 5: Especificações de compactação *.zip*

Todos os arquivos foram compactados, conforme a Figura 6.



Figura 6: Arquivos originais e compactados

**Fase 3 (três) – Testes:**

Pare realização dos testes foi utilizada uma máquina com a seguinte configuração: processador Core i5, 8GB de memória RAM e um SSD de 120 GB. Os arquivos utilizados para execução dos testes encontram-se disponíveis na pasta do drive informada no rodapé deste relatório.

Abaixo, os resultados identificados em cada teste.

* .txt:

O arquivo utilizado para testar o formato .txt é o livro *Contos para Infância*, do autor Guerra Junqueiro e foi obtido através do Projeto Gutemberg.

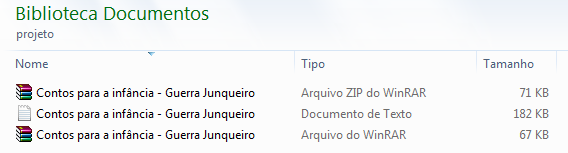


Figura 7: Arquivo *.txt* original e compactados

Os testes foram realizados para os três formatos de compactação do arquivo, conforme Figura 8.

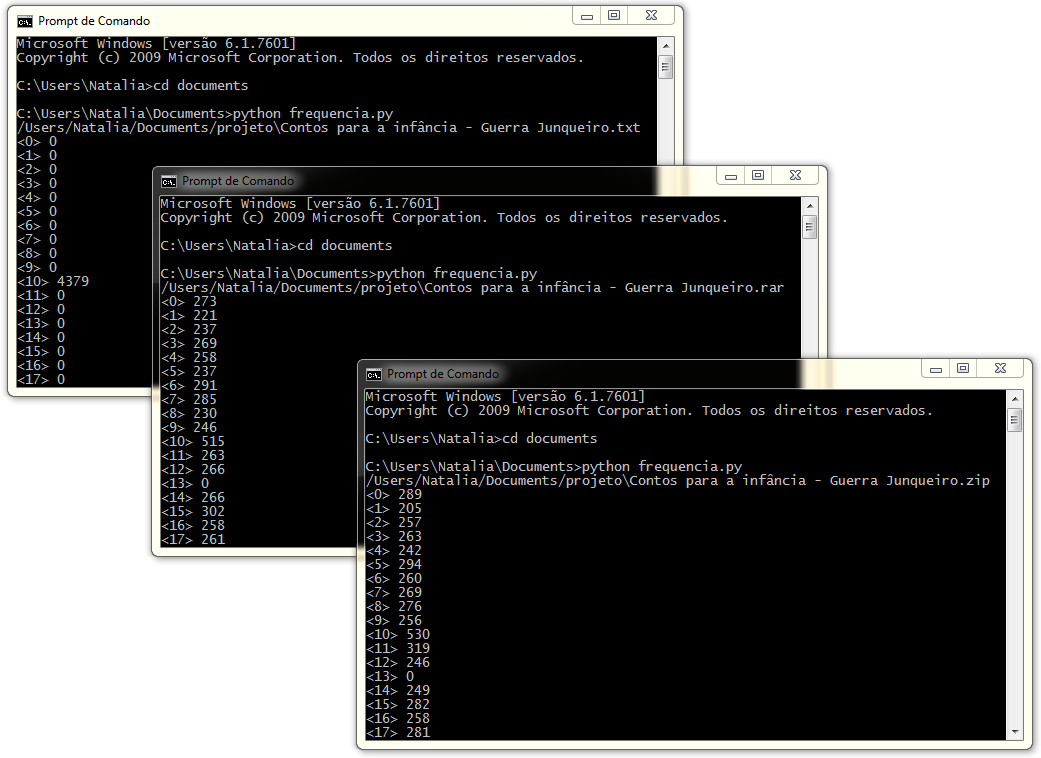


Figura 8: Compactação de arquivo.*txt*

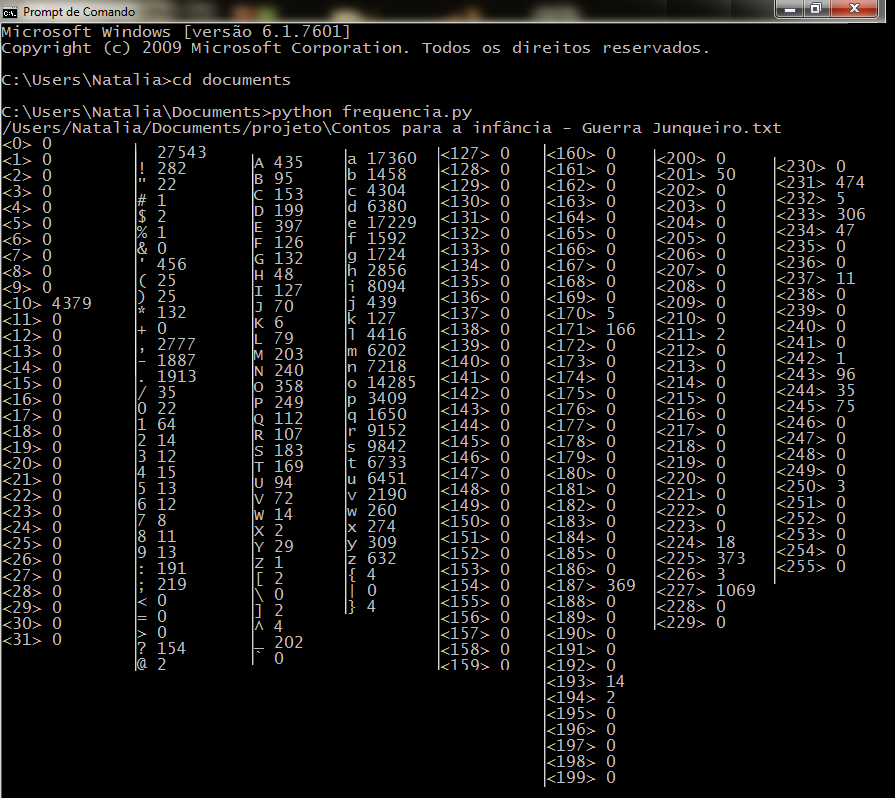


Figura 7: Resultado da contagem de Bytes do arquivo *.txt*

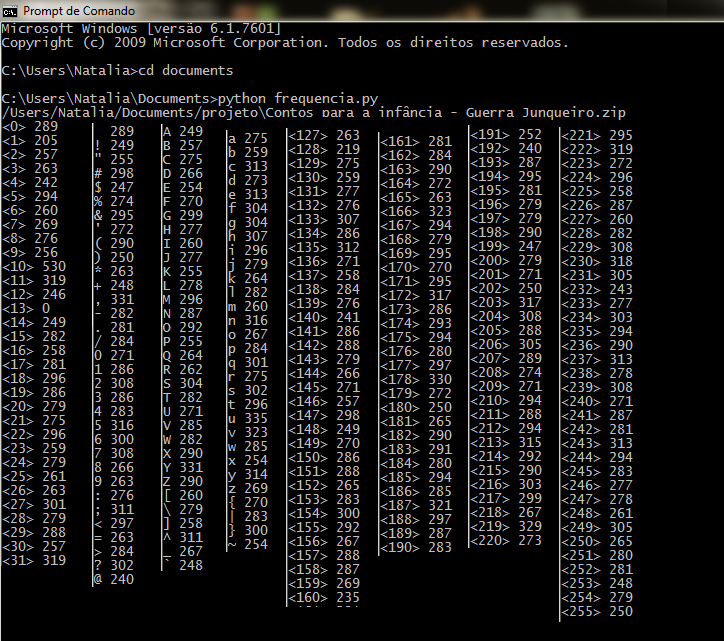


Figura 8: Resultado da contagem de Bytes para o arquivo *.txt* compactado em formato *.zip*

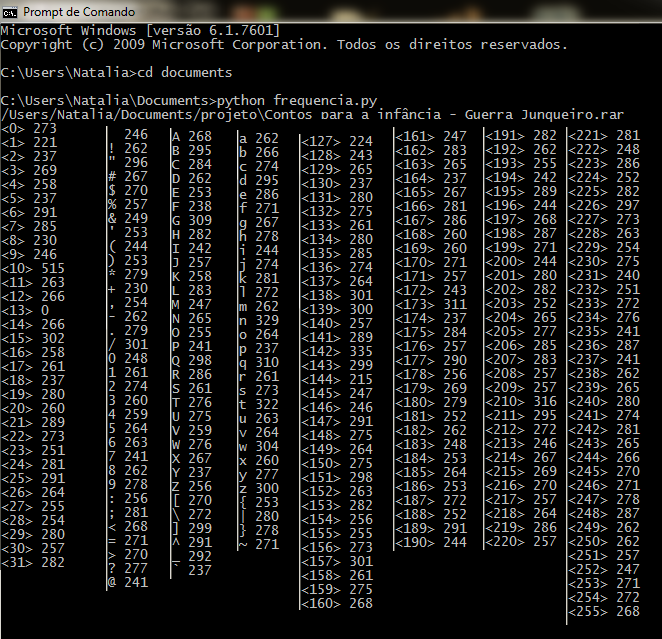


Figura 9: Resultado da contagem de Bytes para o arquivo *.txt* compactado em formato *.rar*

Na Figura 8, podemos perceber a diferença de tamanho para os arquivos compactados, enquanto a compactação em .zip reduziu o arquivo de 182 KB para 71 KB, a compactação do .rar reduziu o mesmo arquivo para 67 KB. Para a análise da frequência de Bytes, do arquivo .txt utilizamos, após identificação dos caracteres que apareceram com maior frequência, os caracteres de *espaço* e ‘*a*’ para comparação entre os três tipos de arquivo.

Percebemos que, para o arquivo não compactado, o caractere de *espaço* apareceu 27543 vezes, para 289 vezes no formato .zip e 246 vezes no formato .rar; para o caractere ‘*a*’, a ocorrência identificada no arquivo original foi de 17360 vezes, para 275 vezes no .zip e 262 vezes no .rar.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | .txt | .zip | .rar | Redução | |
| .zip | .rar |
| Espaço | 27543 | 289 | 246 | 95,3045x menor | 111,96x menor |
| a | 17360 | 275 | 262 | 63,1273x menor | 66,2595x menor |

Sendo assim, e possível identificar que após a compactação do arquivo, tanto para a frequência de Bytes, quanto pelo tamanho dos arquivos, para um formato .txt, o formato de compactação .rar compacta melhor que o .zip.

* .pdf:

Para os testes com arquivo pdf foi selecionado o livro *A Descoberta do Mundo*, de autora Clarice Lispector. O arquivo foi disponibilizado pela equipe do *LeLivros*, para uso parcial em pesquisas e estudos acadêmicos.

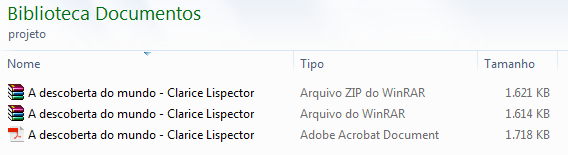


Figura 10: Arquivo *.pdf* original e compactados

Os testes foram realizados para os três formatos do arquivo, conforme Figura 12.

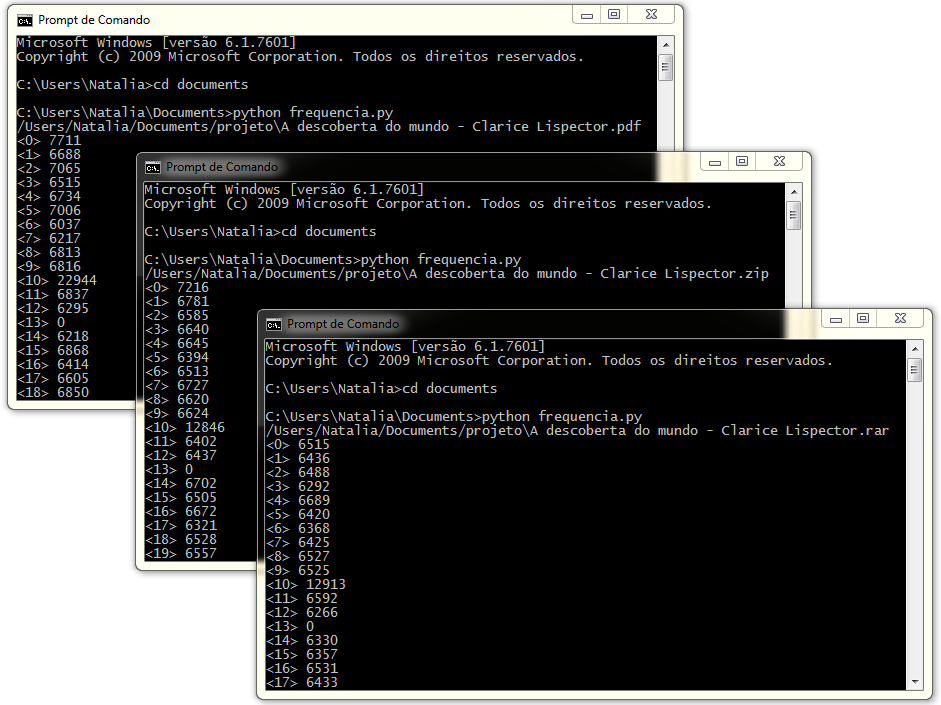
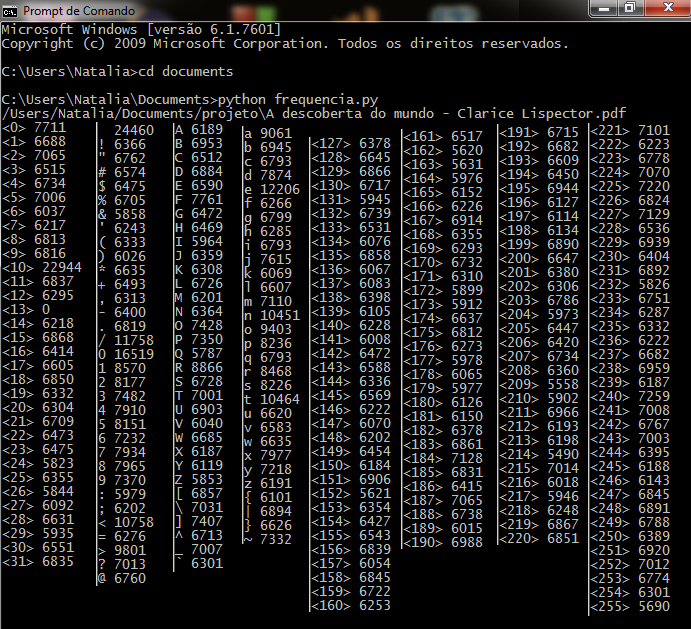
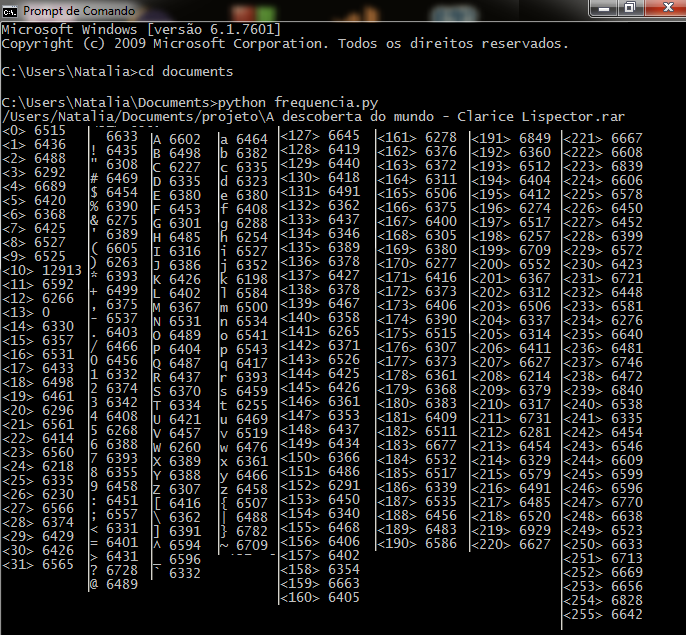
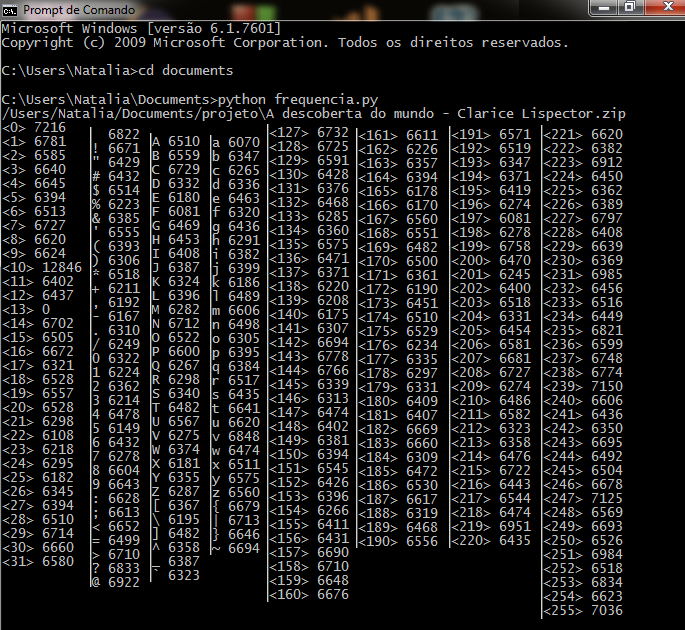


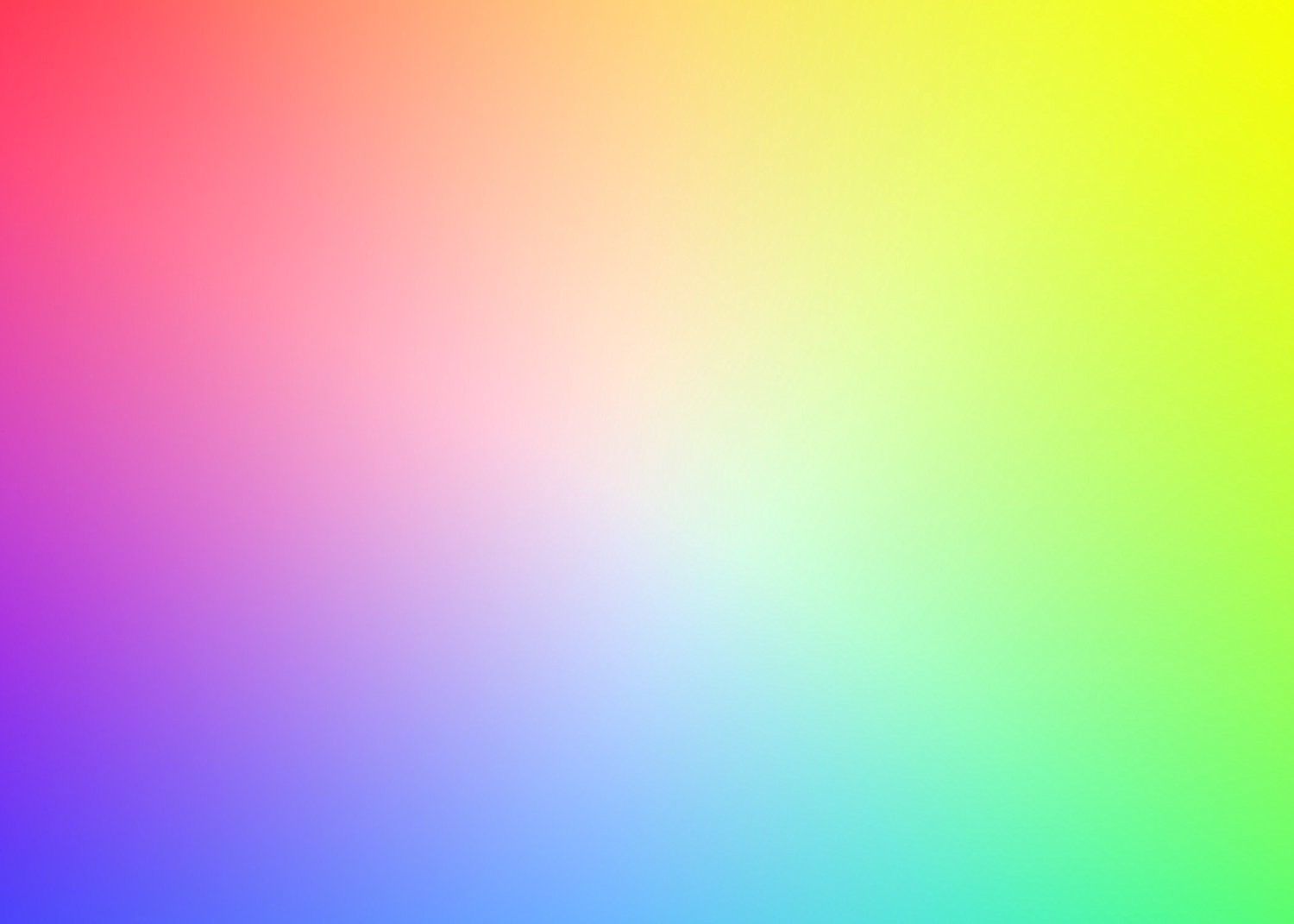
Figura 11: Compactação de arquivo .pdf







* .jpeg:

A imagem abaixo utilizada para os testes em .jpeg foi obtida através do Google Imagens e está disponível para download no Pinterest.

