

<b>Data de publicação do enunciado:</b>	2015-10-03 (v1.04: 2015-10-16)
<b>Data de entrega da etapa 1:</b>	2015-11-07
<b>Data de entrega da etapa 2:</b>	2015-12-19
<b>Informações:</b>	Moodle
<b>Data estimada da publicação de resultados:</b>	1ª etapa: 2015-12-07; 2ª etapa: 2016-01-17

## Programação Avançada – Projeto

### *paCodec*

## 1 - Introdução

O objetivo do projeto *paCodec* é o de implementar, com recurso à linguagem C, a aplicação de modo consola *pacodec*, um utilitário que permita efetuar a compressão com perda de imagens do tipo *Portable Gray Map (PGM)* e respetiva descompressão recorrendo a um algoritmo de substituição por padrões. Este projeto está dividido em duas fases de entrega e apresentação. As duas entregas **são obrigatórias**.

## 2 - Algoritmo

Atualmente, uma parte significativa de dados em formato digital corresponde a conteúdo multimédia, seja áudio, imagens ou vídeo. O elevado volume dos formatos digitais torna pois imperativo o uso de compressão por forma a reduzir as necessidades de armazenamento e de largura de banda. Para o efeito, existem duas classes de algoritmos de compressão: algoritmos sem perdas e algoritmos com perdas. Como o nome sugere, os algoritmos sem perdas visam reduzir a entropia dos dados, sem perda do conteúdo original dos dados. Pelo contrário, na compressão com perda, há lugar à perda de dados, aceitando-se uma perda de qualidade em troca de um maior rácio de compressão.

O *pacodec* visa ser um codificador/descodificador (CODEC) de imagem com perdas baseado na aproximação de padrões onde cada bloco da imagem a comprimir é substituído por um padrão (bloco) existente num conjunto de padrões designado por *dicionário de padrões*. O bloco substituto corresponde ao bloco do dicionário de padrões que menos perdas introduz, isto é, o bloco que maior semelhança apresenta com o bloco a substituir. O *pacodec* apenas processa imagens do tipo *Portable Gray Map (PGM<sup>1</sup>)*, isto é, imagens com tons de cinzento. Por sua vez, o dicionário de padrões é fornecido na forma de ficheiro sendo empregue nas operações de codificação e de descodificação. Um dicionário de padrões é

<sup>1</sup> <http://netpbm.sourceforge.net/doc/pgm.html>

composto por um número de blocos de tamanho uniforme. Por exemplo, um dicionário  $1024/2 \times 2$  apresenta 1024 blocos, cada bloco com largura de 2 píxeis e altura de 2 píxeis, ou seja 4 píxeis. O conteúdo de cada bloco do dicionário corresponde à intensidade, em tons de cinzento, dos píxeis que o compõe. Assim, num bloco  $2 \times 2$ , o primeiro valor corresponde ao píxel de coordenada (0,0) (canto superior esquerdo), o segundo valor ao píxel da coordenada (0,1) (canto superior direito), o terceiro valor ao píxel (1,0) (canto inferior esquerdo) e, finalmente, o quarto valor corresponde ao píxel (1,1) (canto inferior direito). Os blocos de um dicionário são identificados pelo respetivo índice, desde o índice 0 (zero) correspondente ao primeiro bloco até ao bloco com índice  $n-1$  (isto é, 1023 no exemplo considerado).

O algoritmo subjacente ao *pacodec* é composto por duas operações distintas: codificação e descodificação. As secções seguintes descrevem detalhadamente cada uma das operações.

## 2.1 - Operação de codificação

A fase de compressão requer como elementos de entrada, 1) o ficheiro de imagem a codificar e 2) o ficheiro com o dicionário de padrões, produzindo como resultado o ficheiro codificado (extensão *.cod*).

A operação de compressão processa-se da seguinte forma. A imagem a codificar é processada em blocos com as mesmas dimensões do bloco do dicionário de padrões (e.g,  $2 \times 2$  píxeis). Assim, para cada bloco da imagem de entrada é efetuada uma pesquisa no dicionário de padrões por forma a ser encontrado o bloco do dicionário que mais se assemelha ao bloco da imagem de entrada em processamento. O *grau de semelhança* entre um bloco da imagem e um bloco do dicionário é avaliado através do *erro quadrático* que corresponde à soma do quadrado das diferenças entre as intensidades dos píxeis do bloco de imagem em processamento e as intensidades dos píxeis correspondentes do bloco de dicionário em avaliação. Um píxel  $Pi_j$  do bloco de imagem é dito correspondente a um píxel  $Pd_j$  do bloco de dicionário, se tiverem o mesmo  $j$ . Por exemplo, o píxel  $Pi_{(0,0)}$  do bloco em processamento da imagem de entrada corresponde ao píxel  $Pd_{(0,0)}$  do bloco do dicionário em apreço. Considerando um bloco  $2 \times 2$ , o erro quadrático é calculado da seguinte forma:

$$erro_{quadrático} = (Pi_{(0,0)} - Pd_{(0,0)})^2 + (Pi_{(0,1)} - Pd_{(0,1)})^2 + (Pi_{(1,0)} - Pd_{(1,0)})^2 + (Pi_{(1,1)} - Pd_{(1,1)})^2$$

Calculado o erro quadrático do bloco da imagem de entrada em processamento para todos os blocos do dicionário, é selecionado como bloco substituto, o bloco do dicionário que apresenta o menor erro quadrático, correspondendo a codificação ao índice do bloco do dicionário que foi selecionado. Caso o valor mínimo do erro quadrático se verifique para mais do que um bloco do dicionário, é selecionado o bloco do dicionário que tenha menor índice. O processo de codificação é repetido para cada bloco da imagem de entrada. Terminada a codificação, os índices dos blocos do dicionário que foram selecionados

como blocos substitutos são escritos para o ficheiro de saída, que corresponde ao ficheiro codificado. O formato do ficheiro de imagem PGM, do ficheiro do dicionário DIC e do ficheiro codificado são apresentados na Secção Formato dos ficheiros.

Por uma questão de simplificação, o *pacodec* apenas deve processar ficheiros (por codificados/codificados) cujas dimensões sejam múltiplas das dimensões do bloco individual do dicionário de padrões indicado para a operação. Por exemplo, o *pacodec* pode codificar um ficheiro de imagem com dimensões  $32 \times 16$  com um dicionário de padrões com dimensionamento de padrão  $2 \times 4$ , pois 32 é múltiplo de 2 e 16 múltiplo de 4. Contudo, o mesmo já não sucede com um dicionário de padrão  $3 \times 3$ .

## 2.2 - Operação de descodificação

No *pacodec*, e como convém num CODEC, a operação de descodificação é computacionalmente muito mais simples e consequentemente menos onerosa do que a operação de codificação. Concretamente, o processo de descodificação do *pacodec* consiste na leitura dos valores inteiros que estão sequencialmente guardados no ficheiro codificado (**.cod**). Cada valor inteiro corresponde a um índice do dicionário de padrões. Assim, para cada um dos valores inteiros do ficheiro codificado, efetua-se a escrita dos píxeis do respetivo bloco do dicionário para o ficheiro de saída, isto é, o ficheiro descodificado. Assim, para efeitos da operação de descodificação são necessários 1) o ficheiro codificado (extensão **.cod**) e 2) o ficheiro do dicionário de padrões que foi empregue para a codificação.

Como é óbvio, aplica-se o anteriormente indicado à respeito da necessidade do ficheiro a descodificar apresentar dimensões que são múltiplas das dimensões do bloco individual do dicionário de padrões. Assim, não é possível descodificar um ficheiro codificado com dimensões indicadas de  $32 \times 16$  utilizando-se um dicionário com blocos individuais de padrões de dimensão  $3 \times 3$ .

## 3 - Peak Signal-to-Noise Ratio (PSNR)

Obviamente, dado tratar-se de codificação assente no uso de padrões e, consequentemente, com perdas, só raramente é possível reconstituir plenamente o ficheiro original. Assim, na maioria dos casos, o ficheiro descodificado constitui uma aproximação ao ficheiro original. Uma das métricas empregues para medir a qualidade de codificação/descodificação de um CODEC é o PSNR (*Peak Signal-To-Noise Ratio*).

O PSNR é calculado através da seguinte fórmula<sup>2</sup>:

$$PSNR = 20 \log_{10} \left( \frac{MAX_f}{\sqrt{MSE}} \right)$$

<sup>2</sup> <http://www.ni.com/white-paper/13306/en/>

em que  $MAX_f$  corresponde à intensidade máxima da imagem original e  $MSE$  corresponde ao *Mean Squared Error* e é dado pela seguinte fórmula:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|f(i,j) - g(i,j)\|^2$$

em que  $n$  representa o número de colunas das imagens (original e reconstruída),  $j$  o índice das colunas,  $m$  corresponde ao número de linhas e  $i$  refere o índice da linha. Uma das funcionalidades do *paCodec* é o cálculo do PSNR entre duas imagens.

## 4 - Formato dos ficheiros

O *pacodec* manipula ficheiros PGM, ficheiros com dicionários de padrões e ficheiros codificados (.cod).

### 4.1 - Formato PGM

O formato Portable Gray Map (PGM) é um dos formatos mais simples para a representação de imagens com tons de cinzento. Na prática existem dois formatos PGM: o formato P2 e o formato P5. O formato P2 é também designado por formato *PGM-ascii* dado que todo o conteúdo do ficheiro está em caracteres *ascii*, sendo portanto texto. O formato P5 é também identificado por formato *PGM-raw*, dado que, apesar de apresentar o cabeçalho em formato de texto, a parte referente à intensidade dos píxeis se encontra em formato binário (*raw*). Cada ficheiro do tipo PGM começa com um identificador mágico (*magic number* na literatura Anglo-Saxónica) codificado com dois caracteres ASCII (dois bytes). Assim, o formato P2 inicia-se pela cadeia de caracteres “P2” e o formato P5 por “P5”.

Magic Number	Tipo/Formato	Codificação
P2	Escala de cinzas	ASCII
P5	Escala de cinzas	Binária

Tabela 1 – Formatos P2 e P5 e respetiva codificação.

A codificação ASCII (P2) possibilita a leitura dos ficheiros por parte do utilizador e facilita o envio entre várias plataformas, desde que estas compreendam o ASCII. Por outro lado, os formatos codificados em binário são mais eficientes em tamanho e no tratamento (*parsing*), visto não possuírem espaços em branco. Na codificação binária o PGM utiliza 8 bits por pixel. A listagem seguinte mostra, na totalidade o ficheiro 16x32\_p2.pgm (formato P2, i.e, ASCII) que corresponde à imagem (ampliada) mostrada junto à listagem. Note-se que o formato P2 permite outras representações, pelo que deve consultar a descrição formal do formato disponível no seguinte endereço: <http://netpbm.sourceforge.net/doc/pgm.html>

```

16 32
# width=16, height=32
255
# pixels (left to right, top to bottom)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80

```



Ficheiros do tipo P2 e P5 podem ser criados a partir de outros formatos recorrendo-se a ferramentas como o imagemagick<sup>3</sup> e o editor gráfico GIMP<sup>4</sup>.

## 4.2 - Formato DIC

O formato DIC é um formato de ficheiro de texto empregue para a representação dos ficheiros do dicionário de padrões. Um ficheiro DIC é iniciado por um cabeçalho de três linhas. Assim, a 1ª linha indica o número de blocos de padrões existente no ficheiro, ao passo que a 2ª linha contém a largura do bloco individual do dicionário e a 3ª linha a altura do bloco individual.

Cada linha a seguir ao cabeçalho corresponde a um bloco do dicionário, com o primeiro número da linha a indicar a intensidade do píxel do canto superior esquerdo (coordenada (0,0)), o segundo número a indicar a intensidade do píxel da coluna seguinte e assim sucessivamente. As linhas de comentários não correspondem a nenhum padrão, devendo ser ignoradas aquando do processamento de um ficheiro DIC.

Um exemplo (parcial) de ficheiro de formato DIC é mostrado na listagem seguinte, correspondendo a um dicionário com 128 padrões, sendo que cada padrão tem formato 2 x 4 (i.e., largura x altura). O primeiro padrão (linha 4) tem os valores 90, 89, 91, 89, 92, 90, 92 e 91 significando que o píxel (0,0) do padrão tem intensidade 90, o píxel (0,1) tem intensidade 89, o píxel (1,0) tem intensidade 91, o píxel (1,1) tem

<sup>3</sup> <http://www.imagemagick.org/>

<sup>4</sup> <http://www.gimp.org>

intensidade 89, o píxel (2,0) tem intensidade 92, o píxel (2,1) tem intensidade 90, o píxel (3,0) tem intensidade 92 e, finalmente, o píxel (3,1) tem intensidade 91. Esse primeiro padrão corresponde ao padrão de índice 0 do dicionário. Por sua vez, o segundo padrão (linha 5) corresponde ao padrão de índice 1 do dicionário e tem os valores 30, 51, 51, 44, 44, 51, 44, 52 com significado similar ao anteriormente descrito para o primeiro padrão. Por fim a última linha do ficheiro (118, 84, 87, 66, 70, 58, 63, 58) corresponde ao padrão de índice 127.

É importante notar que o **pacodec** deve estar preparado para lidar com dicionários de padrões com blocos individuais de quaisquer dimensões.

```
128
2
4
90 89 91 89 92 90 92 91
30 51 51 44 44 51 44 52
(...)
118 84 87 66 70 58 63 58
```

### 4.3 - **Formato COD**

O formato COD é empregue para a representação de um ficheiro codificado. As linhas iniciadas pelo símbolo # correspondem a linhas de comentário e deve ser ignoradas. As quatro primeiras linhas efetivas (as linhas de comentário não são obviamente consideradas nessa contagem) do ficheiro correspondem ao cabeçalho do ficheiro. Concretamente, a 1ª linha do ficheiro deve ter Z2 ou Z5, consoante o ficheiro de imagem origem esteja no formato P2 ou P5, respetivamente. A 2ª linha do ficheiro contém as dimensões da imagem codificada sobre a forma de um par de números: o primeiro corresponde à largura da imagem e o segundo à altura. A 3ª linha do cabeçalho contém o índice de maior valor do padrão que foi empregue na codificação da imagem de origem. Por sua vez a 4ª linha contém as dimensões de bloco de dicionário (largura x altura) que foi empregue para obtenção do ficheiro codificado. Finalmente, após a 4ª linha ocorrem os índices dos blocos do dicionário de padrões empregues na substituição dos blocos da imagem original. O formato dos índices é texto caso o ficheiro codificado seja formato Z2 e binário (*raw*) caso o ficheiro codificado seja do tipo Z5.

A listagem seguinte apresenta um ficheiro codificado com o formato Z2. Da leitura do cabeçalho deprende-se que se trata de um ficheiro do tipo Z2, representando a codificação de uma imagem de 512 (largura) por 512 (altura). O maior índice do dicionário de padrões empregue é o 1022 (3ª linha), e o dicionário empregue na codificação tinha bloco unitários de dimensões 2 por 2 píxeis. Finalmente, a codificação levou a que o 1º e 2º, 4º e 5º blocos (sublinhados na figura) da imagem original foram substituídos pelo bloco do dicionário de padrões com índice 47.

Z2

```
# coded file for 'example.p2.ascii.pgm' (dict:'1024_2_2+.dic')
512 512
1022
# width x height of individual block of dict
2 2
47 47 15 47 47 15 527 15 15 15 271 15 15 783 15 15 15 15 527 303 815
(...)
```

## 5 - Funcionalidades do *paCodec*

O utilitário *paCodec* possui as seguintes funcionalidades:

- **--decode** <file>: descodifica o ficheiro do tipo COD (Z2 ou Z5) passado por parâmetro;
- **--decode-dir**<dir>: descodifica todos os ficheiros com extensão `.cod` que se encontram na pasta **dir** (e respetivas subpastas);
- **--dict** <file>: especifica *file* como ficheiro de dicionário de padrões. Caso esta opção surja isoladamente, o *pacodec* deve somente validar o dicionário de padrões, verificando que o número e as dimensões de blocos anunciadas no cabeçalho correspondem efetivamente ao conteúdo do dicionário. Esta opção é obrigatória sempre que se utilizam as opções que requerem um dicionário de padrões, isto é: **--decode**, **--decode-dir**, **--encode** e **--parallel-encode** <file> **--threads** <nthreads>;
- **--encode** <file>: codifica o ficheiro PGM passado por parâmetro.
- **--parallel-encode** <file> **--threads** <nthreads>: codifica o ficheiro PGM passado por parâmetro usando um máximo de *nthreads*. Caso a opção `--threads` não seja especificada, o *pacodec* deverá usar tantas *threads* como o número de *cores* da máquina onde se processa a execução;
- **--PSNR** <original file>,<decoded file><sup>5</sup>: calcula o PSNR (*peak signal-to-noise ratio*) entre uma imagem dita original e uma imagem reconstituída, isto é, a versão descodificada. O PSNR deve ser mostrado com 5 casas decimais;
- **--help**: exhibe ajuda sucinta sobre as opções da linha de comando disponibilizadas pelo *pacodec*;
- **--about**: exhibe o nome e número de estudante dos autores do trabalho.

Tenha em atenção que o processamento dos parâmetros de entrada da aplicação deve ser feito com recurso à ferramenta *gengetopt*<sup>6</sup>.

Durante o desenvolvimento do projeto recomenda-se a instalação do utilitário *hexer* para visualizar, e opcionalmente editar, o conteúdo de um ficheiro binário.

<sup>5</sup> O nome do ficheiro original e o nome do ficheiro descodificado estão separados somente por um vírgula (sem espaços)

<sup>6</sup> <http://www.gnu.org/software/gengetopt/gengetopt.html>

## 5.1 - Saída

NOTA: deve consultar a última secção deste documento para uma completa descrição do formato das mensagens de erro do **pacodec** por forma a que seja possível a utilização dos testes disponíveis através do script *test-pacodec-etapa1.pl*.

### 5.1.1 - Operação de encode e de decode

No final de uma operação de encode ou decode, o **pacodec** deve apresentar na saída de erro (stderr), os seguintes elementos:

- nome do ficheiro inicial (ficheiro de imagem original se for operação de *encode*, ficheiro codificado no caso de operação de *decode*), seguido do nome da operação efetuada (*ENCODE* ou *DECODE*) e do nome do ficheiro de dicionário de padrões;
- tempo total (em segundos, com três casas decimais) que a operação demorou a efetuar.

O formato da saída é:

```
operation:OK:<filename> <ENCODE ou DECODE> <dictionary>  
execution time: <tempo de execução>s
```

Exemplo:

```
operation:OK:exemplo_p2.pgm ENCODE 1024_8_2+.dic  
execution time: 9.234s
```

No caso em que ocorra um erro (e.g, o ficheiro a codificar / decodificar apresenta dimensões incompatíveis com as dimensões dos blocos individuais do dicionário de padrões, ou um dos ficheiros indicados não existe, etc.), a saída deve ser a seguinte:



```
operation:FAILURE:exemplo_p2.pgm ENCODE 1024_3_3+.dic
execution time: 0.002s
FAILURE:incompatible dimensions of file 'exemplo_p2.pgm' with dict '1024_3_3+.dic'
```

### 5.1.2 - Cálculo do PSNR

No final do cálculo do PSNR (opção **--PSNR file1,file2**), o *paCodec* deve apresentar na primeira linha da saída de erro, cinco campos separados por dois pontos (:). Os campos são 1) a palavra PSNR; 2) OK ou FAILURE, 3) o nome do ficheiro dito original, 4) o nome do ficheiro reconstituído e finalmente, o valor do PSNR expresso com cinco casas decimais. Na 2ª linha é apresentado o tempo de execução. Caso ocorra uma falha, a 3ª linha incia-se por FAILURE: e contém uma string identificadora do erro.

Exemplo:

```
PSNR:OK:ficheiro_p2.pgm:ficheiro_p2.reconstruido.pgm:21.91293
Execution time: 0.234s
```

### 5.2 - Tratamento de sinais

O *paCodec* deve estar preparado para receber o sinal SIGINT. Assim, ao receber o sinal SIGINT (enviado via *shell* através do Control+C ou através do comando *kill*), o programa deve interromper a operação que estava a fazer, libertar explicitamente todos os recursos (e.g., memória alocada) e terminar o processo. Deve ainda acrescentar à saída de erro a seguinte linha (data e hora a corresponder ao momento da interrupção):

```
[SIGINT] - Operation interrupted by user @2015-10-01 18h48
```

### 5.3 - Etapa 1

Na etapa 1, o *paCodec*, deverá suportar as seguintes funcionalidades: **--decode**, **--decode-dir**, **--dict** e **--PSNR**. Para além disto, o *paCodec* deverá já estar apto a receber todos os parâmetros da linha de comando, embora tenha apenas implementadas as funções anteriormente descritas. Nas demais opções deverá apresentar a mensagem “[TO BE DONE] option not implemented yet”.

### 5.4 - Etapa 2

Na etapa 2, o *paCodec*, deverá suportar as restantes funcionalidades, nomeadamente: **--encode**, **--parallel-encode** e **--threads**.

## 6 - Descrição das funcionalidades

Esta secção descreve detalhadamente cada uma das funcionalidades apresentadas anteriormente.

### 6.1 - Funcionalidade **--decode**

Esta funcionalidade permite descodificar o ficheiro do tipo COD passado por parâmetro. O nome do ficheiro descodificado é composto pelo nome original e retirada a extensão **.cod** (ignorando maiúsculas ou minúsculas), caso esta exista. Caso a extensão **.cod** não exista, o conteúdo do ficheiro a descodificar é substituído pelo ficheiro descodificado. Por exemplo:

Ficheiro a descodificar	Ficheiro descodificado
file.pgm.cod	file.pgm
exemplo.pgm.cod	exemplo.pgm
file_Z2	file_Z2

É necessário ter em atenção que o uso da opção **--decode <file>** requer a opção **--dict <dictionary>**.

**Notas adicionais:** recorra a um ficheiro temporário para guardar o resultado da descompressão e no final renomeie o ficheiro temporário para o nome desejado. Recomenda-se a utilização da função `tmpfile`.

```
$ ./pacodec --decode file.pgm.cod --dict 512_4_2+.dic  
operation: file.pgm.cod DECODE 512_4_2+.dic  
Execution time: 0.002s
```

### 6.2 - Funcionalidade **--dir-decode**

Ao ser invocada esta funcionalidade, o **pacodec** deve percorrer o diretório indicado (e respetivos subdiretórios) e descodificar todos os ficheiros encontrados que contenham a extensão **.cod** (ignorando maiúsculas ou minúsculas). Os ficheiros descomprimidos devem ser guardados na mesma pasta que o ficheiro comprimido.

O utilitário deve processar recursivamente uma pasta, ignorando *links* simbólicos e apenas contemplar ficheiros com a extensão **.cod** (ignorando maiúsculas ou minúsculas).

É necessário ter em atenção que o uso da opção **--decode-dir <file>** requer a opção **--dict <dictionary>**.

```
$ ./pacodec --dir-decode DIR1 --dict 512_8_2+.dic
operation: DIR1/file1.pgm.cod DECODE 512_8_2+.dic
operation: DIR1/PA.pgm.cod DECODE 512_8_2+.dic
operation: DIR1/EI.pgm.cod DECODE 512_8_2+.dic
Execution time: 1.453s
$
```

### 6.3 - Funcionalidade **--encode**

Esta funcionalidade permite codificar o ficheiro passado por parâmetro de acordo com o algoritmo apresentado na Secção 2. O nome do ficheiro codificado é composto pelo nome original acrescido do sufixo `.cod`. Por exemplo:

Ficheiro a codificar	Ficheiro codificado
image1_p2.pgm	image_p2.pgm.cod
image2.p5.pgm	image2.p5.pgm.cod

Caso o ficheiro já exista, o seu conteúdo é apagado e substituído pelo ficheiro comprimido.

```
$ ./pacodec --encode image1.pgm --dict 1024_2_2+.dic
operation: image1.pgm ENCODE 1024_2_2+.dic
Execution time: 0.053s
$
```

Relembra-se que o uso da opção **--encode <file>** requer a opção **--dict <dictionary>**.

### 6.4 - Funcionalidade **--parallel-encode**

Na funcionalidade **--parallel-encode**, o **pacodec** deve recorrer a uma versão *multithreaded* do algoritmo de codificação, distribuindo o trabalho de codificação pelas várias *threads* da aplicação. Valorizam-se soluções que promovam i) o reaproveitamento de *threads* e ii) um correto balanceamento de carga entre as várias *threads*.

A opção **--parallel-encode** deve ser acompanhada da opção **--dict** que especifica o nome do ficheiro de dicionário de padrões a ser empregue. Conforme anteriormente indicado, caso também seja especificada a opção **--threads <nthreads>**, o **pacodec** deve usar no máximo **nthreads threads**. Caso essa opção não seja especificada, o **pacodec** deve usar um máximo de uma *thread* por *core*.

**NOTA:** valorizam-se soluções que minimizem as operações de criação de *threads* assim como as situações de contenção.

## 7 - Relatório

Para cada uma das etapas deve existir um sucinto relatório (máximo 10 páginas) com a descrição dos algoritmos implementados e a justificação das opções tomadas. O relatório deve citar todas as referências bibliográficas (livros, sítio da Internet, etc.) no qual se baseou o código dos algoritmos. No relatório deve também **constar explicitamente** a indicação dos **objetivos atingidos e não atingidos**. O código a ser entregue deve conter documentação no formato *doxygen*, por forma a que possa ser gerada documentação automática, documentação que não deve integrar o relatório da etapa.

## 8 - Critérios de avaliação

A avaliação de cada etapa será feita de acordo com os seguintes tópicos:

### 8.1 - Etapa 1

Peso	Descrição
5,00%	Relatório com justificação das opções tomadas
5,00%	Estrutura/Qualidade do código e documentação
5,00%	Programa <b>pacodec</b> - processamento dos parâmetros da linha de comando ( <code>gengetopt</code> )
12,50%	Programa <b>pacodec</b> - funcionalidade <code>--decode</code>
10,00%	Programa <b>pacodec</b> - funcionalidade <code>--decode-dir</code>
2,50%	Programa <b>pacodec</b> – funcionalidade <code>--PSNR</code>
<b>40,00%</b>	<b>Total</b>

Acresce-se uma apresentação cujo resultado percentual (0 a 100%) é multiplicado pela nota acima obtida.

### 8.2 - Etapa 2

Peso	Descrição
5,00%	Relatório com justificação das opções tomadas
5,00%	Estrutura/Qualidade do código e documentação
25,00%	Programa <b>pacodec</b> –funcionalidade <code>--encode</code>
25,00%	Programa <b>pacodec</b> –funcionalidade <code>--parallel-encode</code>
<b>60,00%</b>	<b>Total</b>

Acresce-se a possibilidade de uma apresentação cujo resultado percentual (0 a 100%) é multiplicado pela nota acima obtida.

## 9 - Regras

- 1 - O trabalho será realizado **obrigatoriamente** em grupos de 2 estudantes (não serão aceites trabalhos realizados por grupos com mais de 2 elementos). Os estudantes que pretendam realizar trabalho individual deverão efetuar o pedido até 2015.10.13, por escrito, ao docente do respetivo turno prático, bem como ao docente das aulas teóricas. Tenha em atenção, que apenas em casos bem fundamentados serão autorizados trabalhos individuais<sup>7</sup>.
- 2 - O trabalho deve estar claramente identificado, com o **nome completo e** respetivo **número de estudante**. Para além dessa identificação, cada ficheiro fonte deverá conter em comentários um cabeçalho com o nome e número dos autores.
- 3 - O trabalho deve compilar com recurso ao ficheiro *makefile* entregue. Trabalhos que não compilem não poderão ser avaliados.
- 4 - Os comentários e os variados identificadores presentes no código fonte (nome de variáveis, funções, ficheiros fontes, etc.) devem estar em inglês (excluem-se os ficheiros fornecidos pelos docentes).
- 5 - O código fonte deve encontrar-se comentado num formato apropriado ao processamento via *doxygen*.
- 6 - O relatório, denominado '**relatorio\_n1\_n2\_etapaN.pdf**' (entregue no **formato PDF**, sendo n1 e n2, o número ESTG de cada elemento do grupo e *etapaN* **etapa1** para a 1ª etapa e **etapa2** para a 2ª etapa) terá que ser compacto (máx. 10 páginas) e incluir os elementos mencionados na apropriada secção do presente enunciado. Na 1ª página, o relatório deverá identificar os estudantes do grupo através do número ESTG, do nome completo e de uma foto de rosto atualizada de cada elemento do grupo. Cada foto deverá ser colocada junto do nome do respetivo estudante.
- 7 - Todos os ficheiros do projeto (relatório, ficheiros de código fonte, ficheiro *makefile*) devem ser reunidos, através de um utilitário de arquivo e compressão (zip, 7Z, tar.gz, ou tar.bz2), num único ficheiro denominado "**PA.turno\_T\_etapaX\_2015-16.n1-n2** " em que **T** representa o turno prático frequentado pelo estudante com menor número ESTG. T deve corresponder a **D + o número do turno** se se tratar de um turno do regime diurno (exemplo: D1 para o turno 1 do regime diurno) e **N + o número do turno** se se tratar de um turno do regime pós-laboral (exemplo: N2 para o turno prático 2 do regime noturno). Por sua vez, **X** representa o número da etapa (1 ou 2), **n1** representa o número de estudante do 1º elemento do grupo e **n2** o número de estudante do 2º elemento do grupo.
- 8 - O relatório deverá incluir a seguinte declaração na capa: "**Nome\_Estudante\_1 (numero\_estudante\_1) e por Nome\_Estudante\_2 (numero\_estudante\_2) declaram sob compromisso de honra que o presente trabalho (código, relatórios e afins) foi integralmente realizado por nós, sendo que as contribuições externas se encontram claramente e inequivocamente identificadas no relatório e no próprio código fonte. Mais se declara que os estudantes acima identificados não disponibilizaram o código ou partes dele a terceiros**".

<sup>7</sup> Esta medida visa combater o insucesso escolar que se tem verificado em anos anteriores quando os projetos são desenvolvidos apenas por um estudante.

9 - Cada etapa do projeto deve ser entregue através do mecanismo de entrega disponibilizado no sítio da unidade curricular. Em caso de dúvidas deve consultar os docentes.

10 - Fraudes ou tentativas de fraudes originam uma classificação **nula** no presente trabalho para os prevaricadores, bem como o relato do sucedido às instâncias superiores.

11 - Deverá recorrer ao horário de gabinete do docente para o acompanhamento e esclarecimento de dúvidas relativas ao projeto.

12 - Caso faça uso do correio eletrónico para o esclarecimento de dúvidas, deve sempre iniciar o assunto da mensagem por **[EI\_PA]** (caso contrário, a mensagem corre o risco de não ser corretamente identificada pelo filtro *anti-spam*). Para além disso, deve identificar-se com o nome, número, regime e turno prático que frequenta.

13 - Após a entrega de cada etapa do projeto, poderá ser necessária uma apresentação oral do mesmo, a marcar pelo docente. A apresentação é individual, sendo que a nota percentual na apresentação (de 0% a 100%) é multiplicada pela nota resultante da correção para efeitos de cálculo da nota final da etapa.

## 10 - Avaliação do código

Para evitar problemas com bibliotecas, todos os projetos serão avaliados com a máquina virtual lubuntu 14.04 que é empregue nas aulas práticas e que está disponível no endereço indicado no moodle da UC.

# Testes

## 11 - Testes

Na plataforma Moodle poderá efetuar o download do utilitário `test-pacodec-etapa1.pl` que permite testar as funcionalidades da etapa 1 do projeto **pacodec**. Estes testes assumem que o utilitário **pacodec** foi compilado e que as saídas obedecem plenamente ao formato descrito neste documento. A passagem com sucesso em todos os testes tem um peso significativo na valorização do projeto.

A tabela seguinte ilustra as opções da linha de comando que poderá utilizar.

<code>./test-pacodec-etapa1.pl -v</code>	Executa todos os testes em modo <i>verbose</i>
<code>./test-pacodec-etapa1.pl -v dict</code>	Executa apenas os testes relacionados apenas com a opção <code>--dict</code>
<code>./test-pacodec-etapa1.pl -v cod</code>	Executa apenas os testes relacionados com a correta leitura do formato <code>.cod</code>
<code>./test-pacodec-etapa1.pl -v decode</code>	Executa apenas os testes relacionados com a opção <code>--decode</code>

<code>./test-pacodec-etapa1.pl -v decode-dir</code>	Executa apenas os testes relacionados com a opção <code>--decode-dir</code>
<code>./test-pacodec-etapa1.pl -v pgm</code>	Executa apenas os testes relacionados com a correta leitura do format .pgm
<code>./test-pacodec-etapa1.pl -v psnr</code>	Executa apenas os testes relacionados com a opção <code>--PSNR</code>
<code>./test-pacodec-etapa1.pl -h</code>	Exibe outras opções do utilitário não especificadas neste documento

## 12 - Situações de erro

De forma a uniformizar as saídas de erro a fim de poder correr os testes com sucesso, a tabela seguinte contém as mensagens de erro a exibir na saída de erro (*stderr*) para cada uma das situações assinaladas. O projeto entregue deverá respeitar plenamente o formato indicado. **Note que o símbolo \$ deve ser substituído pelo nome do ficheiro relativo ao erro reportado.**

As mensagens de sucesso devem obedecer ao formato especificado no enunciado.

Situação	Estrutura da mensagem de erro
O formato do dicionário não é válido	FAILURE:file '\$' is not a valid DIC file
Dicionário possui blocos com pixeis a menos	FAILURE:missing pixels in DIC file '\$'
Dicionário possui blocos a menos	FAILURE:missing blocks in DIC file '\$'
Ficheiro .COD não é válido	FAILURE:file '\$' is not a valid COD file
Ficheiro .COD contém um número de formato inválido (<> Z2 e <> Z5)	FAILURE:file '\$' contains an invalid COD format
Ficheiro .COD contém um tamanho de bloco que não é compatível com a dimensão da imagem	FAILURE:file '\$' contains an invalid block size
Na opção <code>--decode</code> , o ficheiro DIC e .COD possuem dimensões que não são compatíveis	FAILURE:incompatible dimensions of file '\$' with dict '\$'
Na opção <code>--decode</code> , o ficheiro .COD possui um índice que não está no dicionário	FAILURE:index out of range of file '\$' with dict '\$'
Ficheiro .PGM não é válido	FAILURE:file '\$' is not a valid PGM file

Ficheiro .PGM contém um número de formato inválido (<> P2 e <> P5)	FAILURE:file '\$' contains an invalid PGM format
Na opção --PSNR, os ficheiros a comparar possuem dimensões diferentes	FAILURE:incompatible dimensions for PSNR calculation between files '\$' and '\$'

## Exemplos:

FAILURE:file 't00-missing_size.dic' is not a valid DIC file
FAILURE:missing pixels in DIC file 't31-missing_block_pixels.dic'
FAILURE:missing blocks in DIC file 't41-missing_blocks.dic'
FAILURE:file 't00-missing_header.cod' is not a valid COD file
FAILURE:file 't02-invalid_header-2.cod' contains an invalid COD format
FAILURE:file 't53-block_height_mismatch.cod' contains an invalid block size
FAILURE:incompatible dimensions of file 'encoded.cod' with dict 't01-size-mismatch-1.dic';encoded.cod
FAILURE:index out of range of file 'encoded.cod' with dict 't02-index-incompatible-1.dic'
FAILURE:file 't00-missing_header.pgm' is not a valid PGM file
FAILURE:file 't02-invalid_header-2.pgm' contains an invalid PGM format
FAILURE:incompatible dimensions for PSNR calculation between files 't11-original_pacodec.pgm' and 't11-decoded_pacodec.pgm'