**3**

**Algoritmo Genético**

**3.1. Primeiro nível de subtítulos em Arial 11pt negrito minúsculas pequenas**

Um Algoritmo Genético é um método de procura de soluções ótimas inspirado nas formas de evolução presentes na natureza. Partindo de um conjunto de soluções geradas tipicamente de forma aleatória, este algoritmo, procura criar novas soluções inspiradas no conjunto inicial aplicando para isso operadores de seleção, mutação e *crossover* similares aos utilizados por organismos biológicos na sua evolução de geração em geração.

Este método de otimização é do tipo meta heurístico, ou seja, é uma busca do ótimo dentro de um espaço limitado de soluções, de forma a atingir uma solução aceitável para o problema, representa um ótimo local e não global. O espaço de pesquisa de soluções pode ser limitado de várias formas, pelos parâmetros e modo de funcionamento do algoritmo em si, por limitações computacionais ou pelas características do problema a resolver.

De forma simplificada o algoritmo funciona da seguinte forma:

É gerada a população inicial composta por indivíduos cujo cromossoma é gerado aleatoriamente. Cada indivíduo representa uma solução distinta para o problema. Cada elemento da população inicial é avaliado quanto à qualidade da solução por ele representada, essa avaliação é feita por uma função de fitness que pode ser fixa ou variar durante o *runtime* do algoritmo de otimização.

De seguida surge a criação da próxima geração iniciada pela seleção dos “pais”. O operador responsável pela seleção utiliza a classificação atribuída pela função de fitness a cada individuo da geração anterior para avaliar o mérito de cada solução, existindo assim uma maior probabilidade de os melhores indivíduos da geração anterior darem partes do seu cromossoma aos elementos da geração seguinte. Apos escolhidos os indivíduos a reproduzir para dar origem a um dado individuo na geração seguinte a função de *crossover* decide, usualmente de forma aleatória como “misturar” blocos de ADN dos “pais” para dar origem ao novo cromossoma. Por fim o novo cromossoma é sujeito a uma função de mutação dando origem a um novo individuo pronto para ser avaliado pela função de fitness e o processo repete-se até o algoritmo atingir um dado critério de paragem.

[fazer imagem bonita em UML do ciclo todo]

De seguida detalha-se cada um dos elementos fundamentais de um algoritmo genético.

**3.2. Elementos Fundamentais de um Algoritmo Genético**

3.2.1. Individuo

Na natureza uma população é constituída por vários organismos de uma determinada espécie, cada organismo é definido por um conjunto de instruções codificadas nos seus genes que constituem o cromossoma. Esse cromossoma define características especificas como por exemplo cor dos olhos, e cada característica pode ainda ter várias possibilidades: azul, castanho, verde.

O individuo no algoritmo genético não representa um organismo, mas sim uma possível solução para o problema, a forma como resolve o problema está codificada num *datatype* conveniente onde as varias características da solução são armazenadas. Em geral cada individuo partilha com todos os outros a estrutura geral do seu cromossoma variando o valor armazenado em cada gene. [imagem olhos cabelo etc etc]

A representação tradicional do cromossoma com uma serie de 0s e 1s armazenados numa *string,*[ver figura] apesar de ser a representação clássica, nem sempre a mais útil. Para problemas diferentes pode ser adequado utilizar outros *datatypes* para representar cada individuo do problema assim como outro sistema numérico que não o de base 2 (binário).

A complexidade do genoma e a subsequente escolha do *datatype* depende da quantidade de variáveis a otimizar, por exemplo para otimizar uma estrutura quanto ao tipo de secções pode-se utilizar um vector, mas para otimizar a topologia da mesma estrutura já é mais útil um *array* que armazena as coordenadas x,y e z de cada nó. [imagem]

[imagem de caixas binário] [imagem dos data types falados]

3.2.2. População inicial

A população inicial representa o inicio do algoritmo, é geralmente constituída por uma amostra de milhares de indivíduos com propriedades geradas aleatoriamente de forma a propagar um espaço de soluções suficientemente grande para assegurar uma busca suficientemente rigorosa das soluções possíveis – para não convergir prematuramente.

Para certos problemas de otimização onde já existe uma ideia da localização da solução ótima podem-se gerar indivíduos com um grau de aleatoriedade mais reduzido de forma a que grande parte da população propague a zona onde uma solução ótima é esperada.

A dimensão da população inicial tende a crescer com o numero de variáveis (genes) que o algoritmo genético manipula na sua busca pelo ótimo. [imagem de um solution space propagado] Por exemplo um programa que otimize as secções de um conjunto de barras terá um espaço de soluções menor que um programa que otimize as secções e também as coordenadas x,y,z dos nós que definem as barras, deste modo deve ser gerado uma população maior para o segundo caso.

3.2.3. Função de Fitness

A função de fitness do algoritmo é um ponto critico do programa, ela tem o objetivo avaliar o mérito de cada solução de forma a aumentar a probabilidade de os melhores elementos de uma dada geração serem selecionados para fornecerem os seus genes a indivíduos na geração seguinte.

Em casos simples a função de fitness pode ser bastante simples, quando existe um objetivo único e claro para o problema de otimização por exemplo:

-Dispor as varias peças de tecido a cortar na tela de forma a reduzir o desperdício. Função objetivo: avalia a área de tecido desperdiçada em cada individuo.

Para problemas mais complexo surge a primeira dificuldade que é otimizar mais do que um parâmetro.

A dificuldade de elaboração desta função advém do desafio que é em muitos casos traduzir vários parâmetros que influenciam a qualidade da solução num único valor, por exemplo: Quando otimizamos uma estrutura pode apenas interessar otimizar o uso de material, ou podemos também querer atingir uma defeção mínima em certo ponto, o que deixa a duvida, como traduzir o peso e a rigidez da estrutura num único valor? Que peso dar ao critério de massa e de rigidez? O deslocamento de cada ponto critico tem o mesmo peso ou há nós onde é mais importante baixa rigidez? De que forma penalizo uma solução com boa rigidez, mas mau uso de material? A forma de penalização é demasiado pesada, existe o risco de degenerar a solução?

A segunda dificuldade, muito comum em problemas de engenharia, é a impossibilidade de avaliação direta da solução na função de fitness, é necessário correr a solução por um programa de FEM ou CFD para depois analisar a solução: Por exemplo otimizar a forma de um carro para ter o menor coeficiente de arrasto, o coeficiente de arrasto tem de ser calculado em CFD para cada individuo o que atrasa o tempo de execução do programa.

A função de fitness pode variar ao longo da execução do programa, ou seja, nas primeiras gerações o modo de avaliar cada solução é diferente da função utilizada quando o algoritmo genetico esta mais avançado.

Quando é impossível chegar a uma medida objetiva da função de fitness pode recorrer-se a algoritmos genéticos interativos onde a função de fitness é o utilizador, isto é, cabe ao humano analisar cada solução e classifica-la, por exemplo: avaliar a estética de um produto gerado pelo algoritmo.

-------Falar das penalidades a dar e como ter cuidado !!!!!!!!!!!!!!!!!!

3.2.4. Função de Seleção

A função de seleção transmite o conceito presente na natureza de sobrevivência do mais forte para o algoritmo de otimização. Existem várias formas de implementação de uma função de seleção, as mais usadas são a roulette wheel e tournamet selection, que serão detalhadas à frente. Apesar de existirem diferenças claras de funcionamento dos vários métodos de implementação, o objectivo final é o mesmo, aumentar a probabilidade de os melhores indivíduos de uma geração emprestarem os seus genes à geração seguinte, transportando ao longo das gerações as melhores características de cada um de forma a chegar a uma solução ótima.

3.2.4.1. Roulette wheel

A seleção roulette wheel é a mais clássica, e funciona da seguinte forma:

- Somar valores de fitness de cada individuo e armazenar:

- Atribuir a cada elemento uma probabilidade de seleção:

- Gerar aleatoriamente um numero entre 0 e 1;

- Somar probabilidades de cada individuo até o número gerado ser menor que a soma de fitness;

- O ultimo elemento a ser somado corresponde ao individuo selecionado.

A sua implementação é bastante simples e intuitiva, para problemas simples em que não são esperados problemas de maior na convergência é um bom método a aplicar pela sua simplicidade, em casos mais complexos em que é necessário mais controlo no processo de seleção o método de tournament selection, detalhado de seguida, é mais indicado.

[imagem da roulette wheel]

3.2.4.2. Tournament selection

O método de *tournament selection* tal como o nome indica baseia-se selecionar aleatoriamente indivíduos da população inicial para preencher a *selection pool*. Os indivíduos presentes na *selection pool* são depois comparados entre si e o melhor individuo é o selecionado.

A vantagem deste método é o controlo que oferece, a facilidade com que se aumenta a *selection pressure* e o facto de se adequar sem ajustes a problemas de maximização e minimização.

Para aumentar a selection pressure – reduzir a probabilidade de selecção por parte de indivíduos com baixa performance – basta aumentar o tamanho da selection pool ao selecionar mais indivíduos aleatoriamente.

Dentro da selection pool pode-se ainda introduzir uma probabilidade de selecionar o melhor individuo ou o segundo / terceiro melhor o que pode ser útil para evitar uma convergência prematura num ótimo local.

[meter uma imagem.. fica bonito]

3.2.5. Operadores Genéticos

Os operadores genéticos são os responsáveis por receber os cromossomas selecionados ela função de seleção para reproduzir e criar um cromossoma novo inspirado no desses indivíduos. Apesar de em muitos casos o operador de crossover dê origem a um algoritmo genético aceitável o normal é aplicar os dois operadores tal como defendido por Fogel 2006 “Evolutionary Computation: Toward a New Philosophy of Machine Intelligence”

3.2.5.1. Crossover

O operador de crossover recebe os cromossomas da função de seleção e utiliza blocos de cada um de forma a criar o cromossoma do novo individuo, da geração seguinte.

Existem vários métodos de criação do novo cromossoma, os mais comuns são o single e double point crossover que serão detalhados de seguida, existe também o uniform crossover que permite alargar bastante o search space do algoritmo e que por essa mesma razão só é utilizado em problemas relativamente simples, em problemas mais complexos o alargamento do search space torna o tempo de execução da busca inaceitável.

Os métodos single e double point funcionam da mesma forma, cortar um bloco de um dos indivíduos e introduzi-lo no cromossoma do outro “pai” de forma a criar o cromossoma novo. A diferença entre os dois está no ponto de fim do bloco a introduzir, o single point crossover define o ponto a partir do qual o cromossoma é substituído e o double point para além do ponto inicial também identifica o ponto de fim da substituição tal como visto na figura [meter a figura]

3.2.5.1. Mutação

O operador de mutação introduz alguma informação nova no cromossoma recebido do operador de crossover, o objetivo preservar a variedade de soluções ao longo da busca para o algoritmo não convergir de forma prematura.

A operação de mutação é definida como uma probabilidade que cada gene do cromossoma tem de ser alterado. Essa probabilidade interessa ser baixa de forma a não degenerar a busca, o que transformava o algoritmo numa busca aleatória do espaço de soluções.

A forma como a mutação é feita depende muito do datatype em que o cromossoma esta armazenado, para o caso clássico de bit string uma alteração de cromossoma é feito como na figura, com a alteração de alguns 0 para 1 e alguns 1 para 0. [meter figura]

A mutação no caso de cromossomas onde os genes não são binários deve ser feita com cuidado utilizando por exemplo o operador de mutação gaussiano centrado no valor atual do gene de forma a não existir variação brusca do valor de um gene, este método foi utilizado na elaboração do programa de otimização de torres aqui apresentado e será detalhado mais à frente.

3.2.4. Critério de Terminação

O critério de terminação apesar de ter de existir para o programa não entrar num ciclo infinito é a parte menos importante do programa e onde mais liberdade é dada ao programador, alguns critérios de paragem são:

- Numero máximo de gerações atingido

- Solução com fitness aceitável foi encontrada

- Não existe melhoria da solução á X gerações

- Atingido os limites de computação

**3.3. Problemas de Algoritmos Geneticos**

Meto isto aqui ou no inicio/introdução????

**1.3. Elementos gráficos**

1.3.1. Figuras

As figuras deverão ser colocadas centradas, com a legenda **SOB** a mesma. Esta deverá utilizar tipo de letra Arial 9pt. A dimensão da figura deverá ter em conta a sua legibilidade (nem demasiado pequena, nem exageradamente grande, se tal não for necessário). As figuras não poderão exceder as margens pré-definidas para a impressão especificada em 1.2.2.

As figuras deverão ser numeradas sequencialmente (Fig.1 / Fig.2), eventualmente com a associação do capítulo em que surgem (Fig.1.1. / Fig.1.2. etc.). Esta alternativa apenas deverá ser utilizada quando, efetivamente, existam muitas figuras, aconselhando uma referenciação mais pormenorizada (o mesmo se aplicará a quadros/tabelas e a equações).

Poderão ser inseridas figuras em cor mas aconselha-se que sejam tratadas de forma a serem legíveis e corretamente percebidas a preto e branco ou tons de cinzento, uma vez que o controlo sobre o modo de reprodução perde-se a partir da altura em que o documento passar a estar disponível na base bibliográfica da FEUP (nomeadamente por via eletrónica) e existem muitas cores que, quando impressas a p&b ou fotocopiadas, pura e simplesmente desaparecem ou não são diferenciáveis de outras (caso dos tons de azul, amarelo e verde, principalmente).

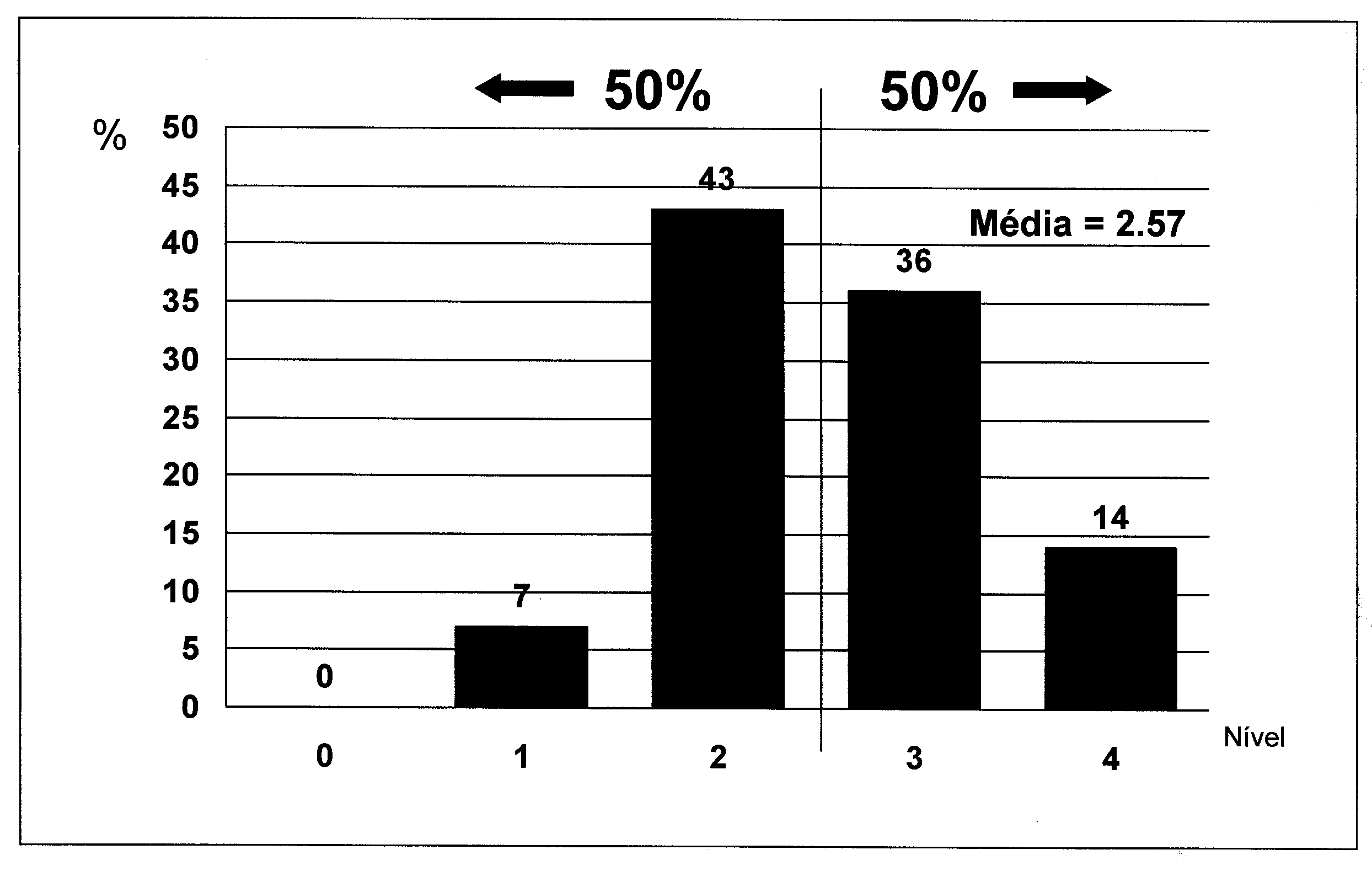


Fig.1 – Legenda Arial 9pt

1.3.2. Quadros ou Tabelas

Os quadros ou tabelas (designação deixada ao critério do autor) deverão igualmente ser inseridos centrados, **apenas com linhas horizontais**, com texto em tipo de letra Arial 10 pt e espaçamento de linhas como no restante texto. Deverão ser numerados sequencialmente (Quadro 1 ou 1.1., etc., conforme referido para as figuras), com o título **SOBRE** o mesmo, com tipo de letra Arial 9pt, tal como nas figuras. Entre o final de um quadro ou da legenda de uma figura e o texto seguinte deverá existir uma linha em branco.

Quadro 1 – Legenda Arial 9pt

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

1.3.3. Equações

As equações deverão ser inseridas na sequência normal do texto, centradas e numeradas consecutivamente ao longo deste. Esta numeração (Arial 9pt) deverá surgir dentro de parênteses, como no exemplo seguinte. Sugere-se a utilização de um editor de fórmulas matemáticas para a sua correta escrita.

 (1 ou 1.1.)

1.3.4. Listas

As listas com marcadores (*bullets)* deverão utilizar o mesmo tipo de letra que o texto corrente e espaçamento de linhas, exceto o acréscimo “Depois”, que deverá ser 0pt entre todas as linhas exceto a última, onde regressará a 6pt para separar a lista do texto subsequente, como no exemplo seguinte:

* Primeira linha da lista;
* Segunda linha da lista;
* Terceira linha da lista;
* Quarta linha da lista.

O tipo de marcadores a utilizar fica ao critério do autor. No entanto, recomenda-se alguma coerência na sua seleção, limitando os tipos utilizados a dois ou três, no máximo, e apenas no caso de fazer sentido uma distinção entre o contexto das várias listas. Se se tratar de listagens com o mesmo espírito, deverá utilizar-se apenas um tipo de marcador.

Caso se pretenda inserir listas numeradas, sugere-se a adoção de marcadores que não se confundam com a numeração de subcapítulos (por exemplo i. / ii. / … ou a) / b) etc.).

Recomenda-se ainda atenção aos alinhamentos verticais em que são inseridas as diversas listas, de modo a assegurar uma regularidade de imagem.

1.3.5. Efeitos

Apenas deverão ser utilizados os efeitos de **negrito**, *itálico*, ou sombreado em tons de cinzento (sugere-se 20% para melhor reprodução, como no exemplo de quadro apresentado). **Não deverá ser utilizado** o sublinhado, sendo o mesmo reservado para as hiperligações (por exemplo, endereços de páginas web), que deverão ser mantidas uma vez que são reconhecidas pelos tradutores para ficheiros em formato pdf.

1.3.6. Referências Bibliográficas

Existem diversos processos de inserção de referências bibliográficas, pelo que se remete essa escolha para o autor. As duas mais correntes consistem em:

* Referência numérica entre parênteses retos - [1], [2] - surgindo a descrição da fonte no capítulo de Bibliografia, ordenada pela ordem que surge no texto;
* Referência pelo nome de autor ou autores, seguida pelo ano da publicação entre parênteses curvos - Autor1, Autor2 (2000). Trabalhos com mais de dois autores são referenciados apenas pelo nome do primeiro e os restantes pela abreviatura *et al*; trabalhos diversos dos mesmos autores num mesmo ano são distinguidos pela nota *a, b,* etc - Autor1 *et al* (2002b). Na Bibliografia estas referências surgem, em primeiro lugar, por ordem alfabética e, de seguida, por ordem cronológica.

No ficheiro relativo ao formato da Bibliografia são indicados os campos a contemplar para cada tipo de referência bibliográfica.

Existe ainda a possibilidade de utilizar a funcionalidade *EndNote* disponibilizada pela Biblioteca da FEUP. Neste caso, deverá ser selecionado o formato da norma NP405 para a produção da lista bibliográfica.

**1.4. Notas Finais**

Pretende-se, com estas regras, estabelecer um conjunto de princípios que assegurem uma uniformidade adequada aos trabalhos a apresentar como Dissertações. Procurou-se que as mesmas não fossem demasiado rígidas e difíceis de entender mas, fundamentalmente, definir regras que deverão ser aplicadas às situações mais correntes e que possam, com sensatez, ser adaptadas para casos mais particulares.

Espera-se que os estudantes dediquem algum zelo à produção gráfica final dos seus documentos. Trata-se de algo que, ao contrário dos trabalhos produzidos no âmbito das restantes disciplinas cuja divulgação raramente ultrapassa o contexto da avaliação, ficará acessível de forma alargada ao meio científico, técnico e profissional, pelo que, depois do esforço na produção de conteúdos válidos, só fará sentido que estes sejam apresentados de forma profissional e graficamente atraente.