

**Instituto Politécnico do Cávado e do Ave**

**Escola Superior de Tecnologia**

**Mestrado em Engenharia Eletrónica e de Computadores**

Kustic: Braço robótico guitarrista

Robótica Avançada

Grupo I

André Moreira nº23914

Joaquin Dillen nº15463

Nuno Fernandes nº15464

julho 2022

Resumo

A robótica é uma área tecnológica que abrange diversas áreas tais como, a mecânica, a eletrónica e a computação. Na sociedade de hoje, o uso de robôs como máquinas industriais e também para desempenhar outras tarefas específicas, têm sido cada vez mais exponencial.

O robô utilizado para este trabalho foi a KUKA KR 6 R900

1. um parágrafo inicial de introdução do contexto geral do trabalho.
2. resumo dos aspetos mais importantes do trabalho descrito no presente relatório, que por sua vez documenta o trabalho mais importante realizado durante o estágio. Deve mencionar tudo aquilo que foi feito, por isso deve concentrar-se no que é realmente importante e que deve ajudar o leitor a decidir se deve ou não consultar o restante relatório.
3. um parágrafo final com as conclusões do trabalho realizado.

**Palavras-Chave (Tema):** Braço robótico, Guitarra

**Palavras-Chave (Tecnologias):** KUKA, OrangeEdit, Python, PyQt5

Índice

[Resumo iii](#_Toc26781319)

[Índice v](#_Toc26781320)

[Índice de Figuras vii](#_Toc26781321)

[Índice de Tabelas ix](#_Toc26781322)

[Notação e Glossário xi](#_Toc26781323)

[1.1 Enquadramento 1](#_Toc26781324)

[1.2 Tecnologias utilizadas 2](#_Toc26781325)

[1.3 Contributos deste trabalho 2](#_Toc26781326)

[1.4 Organização do relatório 2](#_Toc26781327)

[Bibliografia 7](#_Toc26781328)

[Anexo 1 - Conteúdo em anexos 9](#_Toc26781329)

[Anexo 2 - Regras de Conteúdo e Estrutura 10](#_Toc26781330)

[2.1 Linguagem 10](#_Toc26781331)

[2.2 Formatação 11](#_Toc26781332)

[2.3 Imagens e tabelas 13](#_Toc26781333)

[2.4 Bibliografia 14](#_Toc26781334)

[Anexo 3 - Projetos de desenvolvimento 17](#_Toc26781335)

[3.1 Análise 17](#_Toc26781336)

[3.2 Desenvolvimento 17](#_Toc26781337)

[3.3 Instalação/Experiências 18](#_Toc26781338)

Índice de Figuras

[Figura 1 - Exemplo de imagens a) difícil leitura; b) fácil leitura 17](#_Toc313398608)

[Figura 2- Exemplo de lista de bibliografia 20](#_Toc313398609)

Índice de Tabelas

[Tabela 1 - Estilos pré-definidos 16](#_Toc313398617)

[Tabela 2 - Exemplo de tabela 18](#_Toc313398618)

Notação e Glossário

|  |  |
| --- | --- |
| **CAD** | Computer Aided Design |
| **PTP** | Ponto a ponto |
|  |  |
|  |  |
|  |  |

# Introdução

## Enquadramento

O relatório final do Projeto/Estágio da Licenciatura em Engenharia Eletrotécnica e de Computadores é um documento que descreve todas as atividades desenvolvidas pelo aluno no decorrer do período de projeto/estágio. Como tal, deverá realçar a componente de integração profissional (no caso de estágio) ou desenvolvimento técnico (no caso de projeto).

Na elaboração do relatório o aluno deve ter presente que está a produzir um documento público e que, de certa forma, resume todo o seu processo de aprendizagem ao longo do curso.

A unidade curricular de projeto/estágio deve também desenvolver o espírito crítico do aluno, que se carateriza pelo apontar de defeitos e virtudes de um tema ou problema. Assim, espera-se que o relatório, mais do que uma mera descrição de atividades, contenha também uma análise autocrítica aos conhecimentos adquiridos e ao trabalho desenvolvido.

A introdução dá ao leitor a informação básica necessária por forma a facilitar o enquadramento dos objetivos e dos resultados. A introdução deve começar por dar uma perspetiva geral do problema em estudo (na secção de enquadramento) e à medida que vai progredindo, deve ir fornecendo informação mais específica, até se abordar a área em concreto tratada no relatório. Deve descrever, de forma sucinta, o problema em estudo e enunciar os principais métodos que são utilizados no trabalho, bem como a identificação clara dos aspetos inovadores da solução. A introdução deve terminar com a apresentação sucinta das secções que fazem parte do relatório.

## Tecnologias utilizadas

O robô utilizado para a realização deste trabalho, foi a KUKA KR 6 R900. Este robô tem seis graus de liberdade, consegue manusear uma carga máxima de 6 kg e um alcance máximo de 901.5 mm.

Todo o código, exceto o do robô, foi feito na linguagem de programação Python, no IDE PyCharm. Para fazer a comunicação entre o computador e o robô foi utilizada a livraria OpenShowVar, que funciona como um cliente TCP.

## Contributos deste trabalho

Com este trabalho tivemos a oportunidade de trabalhar com o nosso primeiro robô industrial, perceber o seu funcionamento e técnicas utilizadas para extrair mais funcionalidades dele.

## Organização do relatório

Apresentação sucinta dos capítulos que fazem parte do relatório, descrevendo em poucos parágrafos o que cada um deles irá tratar.

# Contexto teórico

A



# Descrição técnica

A

## Ferramenta

A

## Criação das músicas

Para criar as músicas foi necessário fazer um estudo prévio dos componentes que constituem uma tablatura de guitarra. As tablaturas de guitarra partilham semelhanças com a notação da pauta musical, mostrando quais as notas a tocar, quanto tempo as tocar, e quais as técnicas a utilizar. Mas quando comparada com a notação de música padrão, a tablatura da guitarra oferece a vantagem de mostrar onde tocar as notas na guitarra, especialmente, porque a guitarra tem muitas maneiras diferentes de tocar as mesmas notas.

A tablatura da guitarra é a representação visual das notas de uma canção e consiste em seis linhas horizontais, com cada linha a representar as seis cordas da guitarra. Ao olhar para a tablatura da guitarra de cima para baixo, a linha superior representa a corda E alto (a corda mais fina) seguida por linhas que representam as cordas B, G, D, A e E baixo (a corda mais grossa).



Figura 1 – Designação das cordas de uma guitarra

Em cada linha de tabulação existem também números. Estes números representam os trastes da guitarra, que são as faixas de metal encontradas na escala. Os trastes são numerados de 0 a 24, e começam na pestana (a peça mais próxima da cabeça), e correm ao longo de todo o comprimento da escala da guitarra.

A picture containing text, shoji, building

Description automatically generated

Figura 2 – Trastes de guitarra

Por exemplo, se a corda tiver um 0, isso significa que tem de se tocar essa corda "aberta", ou sem usar a mão no traste. Se a corda tiver um 1, então isso significa que deve ser tocada usando o primeiro traste.

Sendo que só tínhamos um robô disponível e este só podia tocar uma nota de cada vez, foram utilizadas tablaturas que fossem possíveis de ser tocados com um só dedo (Figura 3).

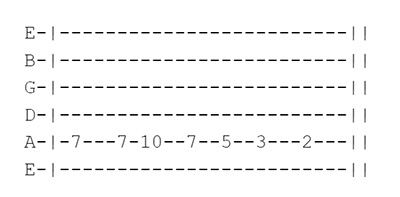


Figura 3 – Tablatura de guitarra

A última componente presente na tablatura é o tempo que nas notas devem ser tocadas, que é representado pelo espaço entre os números.

Todos estes componentes são facilmente convertidos para código Python através do uso de dicionários, sendo que todos eles têm um nome e um valor associado. No caso das cordas, a chave era a sua letra e a sua numeração, pois existem duas cordas co a mesma letra (E baixo e E alto), e o seu valor era a numeração (Figura 4).



Figura 4 – Dicionário Python de cordas de guitarra

Da mesma maneira, criamos também um dicionário para os trastes, sendo a sua chave o nome do traste e o seu valor o seu número correspondente (Figura 5).

A picture containing icon

Description automatically generated

Figura 5 – Dicionário Python de trastes de guitarra

Os valores da nota musical são determinados em referência ao comprimento de uma nota inteira (Semibreve). As outras notas são nomeadas em comparação: uma mínima que é metade do comprimento de uma nota inteira, uma semínima é um quarto do comprimento, etc. Todos as notas musicais podem ser vistas na Tabela 1.

Tabela 1 – Figuras de som

|  |  |  |
| --- | --- | --- |
| Nome | Símbolos | Tempos |
| Semibreve | A picture containing logo  Description automatically generated | 4 |
| Mínima | Icon  Description automatically generated | 2 |
| Semínima | A picture containing icon  Description automatically generated | 1 |
| Colcheia | Icon  Description automatically generated with medium confidence |  |
| Semicolcheia | A black and white logo  Description automatically generated with low confidence |  |
| Fusa | A picture containing text, clipart  Description automatically generated |  |
| Semifusa |  |  |

Tal como foi feito para as cordas e para os trastes, criamos um dicionário para as notas musicais e os seus tempos (Figura 6).

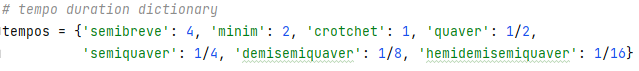


Figura 6 – Dicionário Python de trastes de guitarra

Para criar uma nota é dado o frete, a corda e o tempo que essa nota devia de ser tocada, que por sua vez é multiplicado pelo ritmo, para que a música possa ser tocada mais rápida ou mais lenta, mantendo a sua consistência.



Figura 7 – Nota individual

Sendo que uma música é um conjunto de notas, foi natural o uso de listas para a criação das mesmas (Figura 8).

A screenshot of a computer

Description automatically generated with medium confidence

Figura 8 – Exemplo de uma música

Estes dicionários tornam a inserção de músicas mais fácil, pois conseguimos ter a certeza de que o valor que estamos a inserir é igual em todas as músicas e se fosse necessário alterar o valor de um componente, apenas o teríamos de alterar num só sítio.

## Aquisição das coordenadas

Numa tablatura de guitarra, são indicados o traste e a corda a tocar para cada nota, sendo a combinação dessas informações o ponto (X, Y) onde temos de colocar o dedo.

Para fazer com que o robô toque uma nota, é necessário converter o número do traste e a corda para valores em milímetros. Nos seguintes subcapítulos é explicado os procedimentos utilizados para obter esses valores.

### Coordenadas do traste

Os luthiers utilizam uma fórmula matemática, intitulada de “Regra de 18”, para calcular a colocação dos trastes. Ao contrário de muitos predecessores históricos, a Regra de 18 baseia-se no conceito de temperamento igual, provavelmente antes de o termo existir no vernáculo. Mas, mais importante ainda é que, a Regra de 18 inclui o seu próprio fator de compensação no que diz respeito à posição da ponte.

Ao contrário do que o nome indica, 17,817, é o valor mais comummente utilizado atualmente, como o fator com o qual se divide o comprimento da corda. O resultado é considerado por muitos como sendo mais exato por colocar o 12º traste no ponto médio exato do comprimento da corda. Uma vez que o 12º traste é a oitava acima da corda aberta, faz todo o sentido que esteja no meio do comprimento da corda vibratória.

Para calcular o comprimento de um traste (CT) basta seguir a seguinte fórmula, onde primeiro se subtrai o comprimento acumulativo (CA), de todos os trastes até ao traste desejado, ao comprimento total da escala (CE), desde a pestana da guitarra até à ponte da guitarra e de seguida divide-se por 17,812.

Para exemplificar melhor, seguem-se 3 exemplos de como aplicar a fórmula, assumindo que o comprimento da escala é de 65 cm.

* Traste 1:
* Traste 2:
* Traste 3:

Utilizando esta fórmula e fazendo algumas alterações é possível calcular o valor da coordenada em X. Para isso é necessário somar os comprimentos de todos os trastes anteriores ao desejado e depois subtrair do comprimento acumulativo, metade do comprimento do traste pretendido, para o robô se posicionar no centro do traste, e metade do comprimento do primeiro traste, porque no nosso caso, o ponto (0, 0) era no centro do primeiro traste e na sexta corda (E baixo).

Na Figura 1 é apresentada a implementação em código Python do método explicado anteriormente.

Text

Description automatically generated

Figura 9 – Cálculo da coordenada X

### Coordenadas da corda

Table

Description automatically generated

Figura 10 – Cálculo da coordenada Y

## Comunicação

Depois de obtidas todas as coordenadas era necessário enviá-las para a KUKA. Para esse efeito foi utilizada a livraria desenvolvida em Python, chamada OpenShowVar. Esta livraria é um cliente TCP que serve de interface para os robôs KUKA. OpenShowVar implementa um protocolo fácil, para leitura e escrita de variáveis utilizadas dentro de um programa de movimento do robô.

Graphical user interface, text, application

Description automatically generated

Figura 11 – Exemplo de ligação, leitura e escrita de variáveis da KUKA através da livraria OpenShowVar

Uma vez estalecida a ligação entre o computador e a KUKA, e depois de selecionada a música e calculadas as coordenadas, estas eram enviadas para a estrutura definida na KUKA, sendo apenas alterados os valores de X, Y e do ritmo.

Text

Description automatically generated

Figura 12 – Envio das coordenadas referentes às notas que iriam ser tocadas

Para além de mandar as coordenadas para a KUKA, era também lido o estado atual do robô, para saber se este estava a tocar ou se estava à espera de receber a ordem de arranque. Com essa informação sabíamos qual a ordem a enviar (ordem de arranque ou ordem de paragem), quando o utilizador premia o botão de reprodução.

Foi também adicionado um botão para ativar a emergência através da interface gráfica, que quando premido alterava o valor de uma variável no robô sendo que no código do robô essa variável estava ligada a uma interrupção, que para o seu movimento.

## Interface

Com a lógica do lado do computador completa, foi necessário desenvolver uma interface gráfica de utilizador (GUI) para que um utilizador conseguisse escolher qual a música que o robô iria tocar e transmitir esses dados de uma forma simples e intuitiva. Para criar a GUI, foi utilizada a livraria PyQt5, que torna possível o uso das capacidades e funcionalidades do Qt em código Python.

Qt é um conjunto de bibliotecas C++ multiplataforma que implementam APIs de alto nível para aceder a muitos aspetos dos modernos sistemas desktop e móveis. Estes incluem serviços de localização e posicionamento, multimédia, conectividade NFC e Bluetooth, um navegador Web baseado em Chromium, bem como o desenvolvimento de interfaces de utilizador tradicionais.

PyQt5 é um conjunto abrangente de ligações Python para Qt v5. É implementado como mais de 35 módulos de extensão e permite que Python seja utilizado como uma linguagem de desenvolvimento de aplicações alternativa a C++ em todas as plataformas suportadas, incluindo iOS e Android.

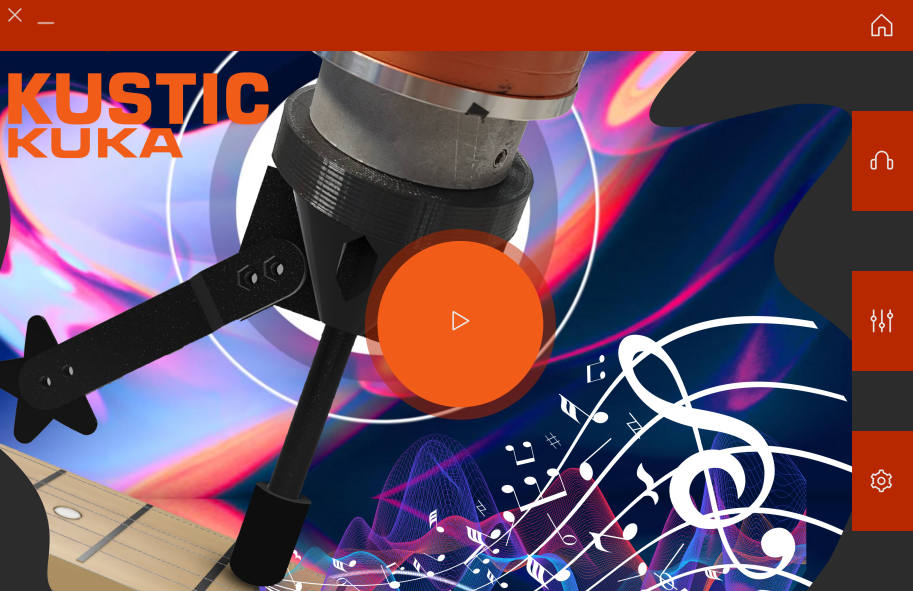


Figura 13 – Ecrã principal

A maior parte, dos elementos gráficos da GUI foram desenvolvidos no Qt Designer, que é a ferramenta Qt para conceber e construir GUIs com Qt Widgets. É possível compor e personalizar as suas janelas ou diálogos, e testá-los usando diferentes estilos e resoluções.

A GUI é composta por três ecrãs: o ecrã principal, que acolhe o utilizador (Figura 4); o ecrã de definições que tem informação sobre o programa e os seus criadores; e por último, e o mais relevante, o ecrã da seleção de músicas (Figura 5). O utilizador pode saltar diretamente para o ecrã que deseja, utilizando o menu lateral, presente em todos os ecrãs e composto por quatro botões: *“Home”*, “Músicas”, “Creativo” e “Definições”.

A picture containing graphical user interface

Description automatically generated

Figura 14 – Ecrã da seleção de músicas

A seleção de músicas é feita através de uma lista onde é possível ver a banda, se for o caso, e o título da música. Uma vez selecionada a música, o utilizador tem apenas de pressionar o botão de reprodução (com o ícone da nota musical) ou então, se tiver um teclado disponível, a tecla “ENTER”.

No momento que esse botão é pressionado, o programa faz todos os cálculos necessários para obter as coordenadas da música selecionada e envia esses dados para o robô. Caso o utilizador queira parar a música, apenas tem de premir o botão de reprodução novamente, e este dará a ordem de paragem ao robô.

## Braço robótico

A programação do braço robótico foi feita na linguagem KRL, tanto no IDE OrangeEdit como diretamente na consola do robô. O programa está divido e três partes sendo estas: Inicializações, deslocação para coordenadas e tocar as cordas.

A primeira tarefa a ser executada pelo robô era a atribuição da ferramenta e da base que criamos e calibramos às respetivas variáveis globais do robô, para que deste modo não houvesse dúvidas de qual ferramenta e/ou base estaríamos a utilizar.

Text

Description automatically generated

Figura 15 – Atribuição da ferramenta e da base a serem utilizadas pelo robô

As inicializações eram a segunda tarefa feita pelo robô e certificavam que todos os valores das variáveis estavam bem definidos e reinicializado, e que todos os pontos da estrutura tinham o valor de X e Y a zero para serem capazes de receber novas coordenadas e não serem influenciados pelas antigas. Era também atribuído um valor a Z, A, B e C, para que a altura da ferramenta e a sua posição fossem idênticas em todos os pontos da estrutura. Para além de alterar o valor dos pontos, era também alterado o valor do tempo que as notas deviam de ser tocadas.

Text

Description automatically generated

Figura 16 – Inicializações de variáveis, estruturas e interrupções

Depois de feitas as inicializações o robô executava a função HOME que o deslocava-se para o ponto *HOME\_KUSTIC*, através de um movimento ponto a ponto (PTP), onde aguardava a ordem de arranque. O ponto *HOME\_KUSTIC* estava definido localmente no ficheiro de declarações (.dat).

Neste movimento era limitada a velocidade do movimento PTP a 10% da velocidade de funcionamento do robô, visto que era um movimento longo e poderia causar a deterioração da calibração e porque só era realizado no início e no fim do código tornando o seu impacto, em termos de tempo e eficiência, negligenciável.

Text

Description automatically generated with medium confidence

Figura 17 – Função para se deslocar para o ponto HOME\_KUSTIC

Uma vez no ponto HOME o robô envia uma mensagem para a consola a informar o utilizador do seu estado e fica à espera de receber a ordem de arranque. Quando essa ordem é dada, é feito o *reset* às variáveis de início e paragem, e *set* à variável do estado, para que o computador possa ser informado, que a KUKA está a executar o código.

Text, letter

Description automatically generated

Figura 18 – Espera da ordem de arranque e reset/set de variáveis de estado

O código principal está dentro de um ciclo *for* que vai de 1 até ao número máximo definido na variável MUSIC\_SIZE, que no nosso caso é igual a 30.

# Conclusões

O capítulo de conclusões é um dos mais importantes do relatório, sendo aqui que devem ser apresentados os resultados do trabalho efetivamente desenvolvido.

As conclusões finais devem focar o sucesso/insucesso do trabalho, revendo as dificuldades encontradas. Devem resumir, de alguma forma, as vantagens do produto desenvolvido e a utilidade que possa ter para a instituição de estágio ou para os seus clientes/parceiros. Podem também referir a forma como o estágio decorreu, bem como a integração, a formação dada pela instituição, as facilidades e as dificuldades sentidas ao longo do estágio.

*As conclusões devem basear-se nos resultados realmente obtidos*. Devem enquadrar‑se os resultados obtidos com os objetivos enunciados e procurar extrair conclusões mais gerais, eventualmente sugeridas pelos resultados. Podem acompanhar as conclusões incluindo recomendações apropriadas, resultantes do trabalho, nomeadamente sugerindo e justificando eventuais extensões e modificações futuras.

Bibliografia

1. - Conteúdo em anexos

Esta parte do relatório deve conter informação adicional organizada por capítulos, que embora seja interessante, não faz parte do estritamente necessário ao relatório. Documentos importantes produzidos ou utilizados durante o estágio que, pela sua dimensão, não sejam colocáveis no corpo principal do relatório podem também ser incluídos em anexos.

Um exemplo possível é um capítulo com o “diário” de trabalho. Outro exemplo é um capítulo com experiências mais detalhadas e complexas realizadas. Eventualmente, nos anexos poderá também aparecer o manual de utilizador da aplicação ou módulo desenvolvido.

1. - Regras de Conteúdo e Estrutura

Dependendo de cada projeto específico, a proposta de estrutura apresentada neste documento pode ser alterada, acrescentando novos capítulos, subdividindo capítulos em dois ou juntando dois capítulos num só. *O aluno deve discutir com o orientador qual a melhor abordagem para o seu caso, seguindo as linhas orientadoras aqui apresentadas*.

A dimensão dos capítulos deve ser equilibrada de forma a não haver muita diferença no número de páginas entre capítulos. Exceção feita, obviamente aos capítulos de introdução e conclusões. Cada um destes dois capítulos deve ter uma dimensão à volta de 10% do total de páginas do relatório.

A estrutura dos capítulos deve ser tal que contenha secções e subsecções de forma equilibrada, cada uma contendo partes relativamente separadas do trabalho. A primeira secção deve começar no princípio do capítulo. Não incluir secções ou subsecções com menos de uma página e não criar apenas uma secção (subsecção) dentro de um capítulo (secção). Deve-se também evitar criar subsecções com demasiados níveis, devendo-se usar apenas, regra geral, até ao 3º nível, ex., 1.1.1.

As primeiras páginas, até à notação inclusive, identificam-se com numeração romana, em letras minúsculas. A numeração de capítulo/secção é efetuada em sequência. Cada novo capítulo deve iniciar-se no topo de página. A numeração de páginas dos anexos é feita continuando a numeração do texto principal. Para separar os capítulos no caso de estar a usar MS Word pode inserir secções de quebra do documento escolhendo a opção Insert 🡪 Break 🡪 Odd Page.

* 1. Linguagem

A linguagem de um relatório deve ser rigorosa, clara e com caráter técnico. Deve evitar-se escrever as frases na primeira pessoa; por exemplo a frase “desenvolvi em seguida o módulo de controlo” pode ser rescrita da seguinte forma: “foi desenvolvido em seguida o módulo de controlo” ou “em seguida desenvolveu-se o módulo de controlo”.

Também se deve evitar o uso de expressões “populares” e de opiniões pessoais. Excecionalmente poderão ser dadas opiniões pessoais nas conclusões, tendo sempre em atenção a polidez e a boa educação. As siglas devem ser sempre definidas da primeira vez que são usadas no texto.

É essencial não esquecer de rever ortograficamente o texto. Os processadores de texto têm normalmente facilidades de correção ortográfica, mas não são suficientes, pelo que devem sempre rever pessoalmente o texto.

* 1. Formatação

A formatação do relatório (tipo de fonte, tamanho, estilos utilizados) é da responsabilidade do autor. Devem seguir-se algumas regras de bom senso e boas práticas:

* Diminuir o número de fontes utilizado (duas ou três no máximo);
* Usar um tipo de fonte e tamanho de fácil leitura (por exemplo: Calibri 12pts);
* Ser consistente na utilização das fontes (usar sempre a mesma fonte para o texto, usar sempre a mesma fonte para os “headings”);
* Utilizar tamanhos de fonte razoáveis e lógicos (por exemplo, se o tamanho da fonte modificar de acordo com o nível de “heading” não usar um tamanho de fonte maior para um “heading” de nível inferior);
* Evitar “floreados” nas fontes (sombras, “borders”, etc.);
* Usar judiciosamente o negrito e o itálico nos parágrafos de texto, devendo a sua utilização ficar restrito a pequenos pedaços de texto que *realmente* importam realçar;
* Usar espaçamento de 1.5 entre as linhas facilita a leitura (não se deve, no entanto, usar esta técnica para aumentar o número de páginas!);
* Usar parágrafos justificados à esquerda e à direita;
* Usar numeração correta e lógica de páginas, de capítulos e sub-capítulos.

Devem utilizar as funcionalidades do vosso processador de texto para a definição de estilos por forma a facilitar e garantir um aspeto homogéneo no relatório. Caso utilizem o Microsoft Word utilizem a opção de menu Format 🡪 Styles and Formating.

Este modelo de relatório tem já definido uma série de estilos que devem utilizar, conforme definido na Tabela 1.

É necessário ter em atenção as margens das páginas e deixar espaço suficiente para a encadernação (evitar margens laterais inferiores a 2 cm). A impressão do relatório deve sempre que possível ser efetuada numa impressora laser de boa qualidade usando a frente e o verso das folhas. Ter em atenção que, caso se opte por impressão frente-e-verso, se deve ter o cuidado de iniciar os capítulos numa página ímpar. Adicionalmente, neste caso, pode-se definir um “header” e um “footer” diferente para as páginas pares e ímpares.

* 1. Imagens e tabelas

As imagens só devem ser colocadas no texto quando auxiliem a interpretação do assunto que se está a abordar. Na preparação das imagens deve haver cuidado para evitar má legibilidade prestando atenção ao número de elementos existentes na imagem, ao tamanho dos elementos e ao tamanho do texto. Adicionalmente deve evitar-se demasiadas cores e “floreados” nos diagramas técnicos a apresentar.

Cada imagem deve ser apresentada com um título curto que a identifique claramente, colocado por baixo da imagem. A figura seguinte mostra dois diagramas que traduzem o mesmo conteúdo, no entanto são bem diferentes em termos visuais e de facilidade de leitura.

As tabelas devem ser usadas para apresentar dados/informação que se queira cruzar em várias dimensões ou que se queira analisar segundo vários atributos. Cada tabela deve ser apresentada com um título curto que a identifique claramente, colocado por cima da tabela. A formatação de tabelas deve obedecer às mesmas regras apresentadas anteriormente de evitar demasiados “floreados” e devem garantir que a tabela não fica dividia entre duas páginas. Adicionalmente tem que se ter cuidado para facilitar a leitura e identificar corretamente a linha e/ou coluna de cabeçalho. A tabela seguinte é um exemplo possível de utilização e formatação de tabelas.

As figuras e tabelas devem sempre ter um título e um número. Verifiquem as funcionalidades do vosso processador de texto para criação automática de título de figuras e tabelas pois facilita a sua numeração e posterior criação de índices. Caso utilizem o Microsoft Word, procurem no menu Insert 🡪 Reference 🡪 Caption.

Os índices de tabelas e de figuras podem ser gerados automaticamente pelo MS Word, usando a opção de menu Insert 🡪 Reference 🡪 Indexes and Tables.

Quando imprimir o relatório, deve sempre atualizar as tabelas de índices existentes. No caso de usar o MS Word, defina nas opções Tools 🡪 Options 🡪 Print 🡪 update fields. Alternativamente pode efetuar a atualização manual, clicando na tabela em questão com o botão do lado direito do rato e escolhendo a opção Update Field.

* 1. Bibliografia

O capítulo de bibliografia apresenta a lista de bibliografia consultada para a execução dos trabalhos de projeto/estágio. A lista de bibliografia deve estar ordenada pela ordem que aparece no corpo do documento.

No texto, sempre que utilizem dados ou afirmações de outros, devem indicar a fonte desses mesmos dados ou afirmação, colocando entre parêntesis retos o número da referência, ex. [3].

A formatação de cada entrada bibliográfica é diferente consoante o tipo de documento em questão:

* *para um livro:* nome(s) do(s) autor(es), ano da edição entre parêntesis, título do livro em itálico, nome da editora, local da edição, país da edição;
* *para um artigo em revista:* nome(s) do(s) autor(es), ano da edição entre parêntesis, título do artigo em itálico, nome da revista, volume da edição a negrito, número da edição, páginas;
* *para uma comunicação em conferência:* nome(s) do(s) autor(es), ano da conferência entre parêntesis, título da comunicação entre aspas, nome da conferência em itálico, local da conferência, país da conferência, mês da conferência;
* *para uma tese:* nome do autor, ano da tese entre parêntesis, título da tese, tipo de tese, universidade da tese, local da universidade, país da universidade;
* *para um relatório interno:* nome(s) do(s) autor(es), ano do relatório entre parêntesis, título do relatório, origem do relatório, referência do relatório, instituição de acesso ao relatório, local da instituição, país da instituição, mês do relatório (abreviado com 3 letras, excepção aos meses com 4 letras);
* *para um documento extraído da Internet:* adicionar o endereço entre parênteses;
* *para um portal da Internet:* o endereço.

Na figura seguinte é apresentada uma lista bibliográfica com um exemplo de alguns tipos referidos.

1. - Projetos de desenvolvimento

Este anexo apresenta uma proposta de estrutura para a descrição técnica de relatórios de projeto/estágio que consistam no desenvolvimento de soluções. *O aluno deve, conjuntamente com o orientador, definir a estrutura mais adequada ao seu projeto.*

* 1. Análise

Neste capítulo é apresentado um modelo conceptual do problema a resolver. É aqui que normalmente se apresentam os modelos de dados, diagramas de estruturas (módulos e/ou classes), etc. correspondentes ao problema e à solução proposta.

Na análise de um problema e desenvolvimento da respetiva solução, é costume seguir-se os seguintes passos:

1. *requisitos/funcionalidades* – dialogando com o cliente, identificar as funcionalidades de alto nível (as “grandes” funções) pretendidas no sistema para cada perfil de utilizador.
2. *processos* – continuando o diálogo com o cliente, analisar e efetuar uma descrição de alto nível dos processos existentes no sistema e das interações entre os diferentes intervenientes nesses processos (“workflow”).
3. *estrutura lógica* – identificar e descrever detalhadamente as diferentes entidades existentes no sistema, bem como detalhar a interação com os sistemas anteriormente identificados, por forma a incluir os passos de implementação e respetivas operações.
4. *estrutura física* – identificar os diferentes elementos físicos do sistema (ex., bibliotecas de funções, executáveis), bem como identificar os recursos de “hardware” necessários à instalação do sistema.
   1. Desenvolvimento

Este capítulo descreve a implementação da solução proposta no capítulo anterior. Alguns dos diagramas referidos na secção anterior podem aparecer neste capítulo e não no capítulo de análise; é o caso dos diagramas de classes ou diagramas de funcionamento. Neste capítulo são detalhadas as operações/funções de cada módulo (usando pseudocódigo ou diagramas).

Neste capítulo são também descritas as especificidades de implementação de acordo com o ambiente de desenvolvimento, plataforma, linguagem e hardware escolhido para o desenvolvimento. Devem também ser reportados os problemas encontrados e a solução escolhida para os resolver. Os contratempos do estágio (por exemplo, o computador avariou e durante 3 dias não foi possível trabalhar) também podem ser referidos.

* 1. Instalação/Experiências

Este capítulo descreve a instalação da solução (não confundir com “*setup*”). O que se entende por instalação é a arquitetura física concreta onde a solução foi instalada e os componentes necessários.

São também aqui descritos os testes efetuados e apresentados os dados/modelos utilizados, bem como os resultados obtidos. Caso tenha havido lugar a melhorias, devido ao resultado dos testes ser insuficiente ou errado, tal também deve ser indicado.

Este capítulo pode não existir em todos os projetos/estágios, dependendo da sua natureza. Em alguns casos é também aceitável que se transforme numa secção do capítulo anterior.