

Projeto Unicórnio

João Carlos Pinto, 20808

Orientação de

Nome professor

LICENCIATURA EM ENGENHARIA EM SISTEMAS INFORMÁTICOS
(REGIME PÓS-LABORAL)

ESCOLA SUPERIOR DE TECNOLOGIA
INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

Identificação do aluno

João Carlos Pinto, 20808

Licenciatura em Engenharia em Sistemas Informáticos (regime pós-laboral)

Orientação

Nome professor

Resumo

Resumo do trabalho realizado. Deve ser sucinto, e cobrir todo o relatório: uma introdução ao problema que se pretendeu resolver, um pequeno resumo da abordagem realizada, e algumas conclusões do trabalho atingido.

Poderão ser criados vários parágrafos, até para que cada um corresponda às três fases de introdução, desenvolvimento e conclusão.

Não é relevante colocar no resumo o local de estágio ou a referência ao curso. Essa informação já consta da capa.

Abstract

This is the translation of the previous text. It should say the exact same thing. Please do not use directly Google Translator.

Agradecimentos

[A secção de agradecimentos é a parte pessoal do documento, e o único sítio onde o aluno pode escrever de forma menos formal, usando o tipo de linguagem que lhe parecer adequado para as pessoas a quem agradece.]

Conteúdo

Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	1
1.1 Objetivos	1
1.2 Contexto	1
1.3 Estrutura do documento	2
2 Capítulo 2	3
2.1 Sub-capítulo 1	3
2.2 Figuras	3
3 Capítulo 3	5
3.1 Sub-capítulo 1	5
3.1.1 Lista dos caracteres a ignorar	5
3.2 Sub-capítulo 2	5
3.2.1 Listagem de programa	5
4 Capítulo 4	7
4.1 Sub-capítulo 1	7
4.2 Sub-capítulo 2	7
Lista de Código	9
Bibliografia	11

Lista de Figuras

2.1	Esta é a legenda da figura	3
-----	--------------------------------------	---

Lista de Tabelas

2.1	Legenda da tabela.	4
-----	----------------------------	---

1. Introdução

[A introdução deve, tal como o próprio nome indica, introduzir o tema do trabalho. Não deve haver pressa em falar da empresa onde foi realizado o estágio ou o curso a que se refere o trabalho. Deve fazer-se uma introdução à área, Os Sistemas Informáticos ou as Ciências da Computação são áreas bastante grandes, pelo que não se deve supor que o leitor está a par das necessidades ou das tecnologias usadas em determinada área. No entanto, não devem ser explicados conceitos básicos, que qualquer licenciado numa engenharia de sistemas informáticos ou em ciências da computação tenham obrigação de conhecer.

Na formatação do texto tente-se que não existam demasiadas zonas em branco. Não é pelo número de páginas que se mede a qualidade de um relatório. E, uma vez que os documentos são impressos, poupar algumas folhas é económico e ecológico.

Relembra-se que todo o conteúdo do documento deve ser original. Quaisquer citações retiradas de algum livro ou sítio da Internet devem ser devidamente formatadas, e a referência bibliográfica adicionada [1]:

By understanding a machine-oriented language, the programmer will tend to use a much more efficient method; it is much closer to reality.

Do mesmo modo, se algum texto, embora usando palavras do autor do documento, refira alguma ideia defendida por um outro autor, num outro documento, então também deverá aparecer a respetiva referência bibliográfica (PennState University Libraries, 2017).

O uso de citações é especialmente útil para defender ideias que outros autores também defendem, e que o autor do documento não tem com provar.]

1.1 Objetivos

[Numa pequena secção da introdução liste, cuidadosamente, os objetivos do trabalho. Não confundir com os requisitos do software. Apenas o que se pretendia atingir originalmente.]

1.2 Contexto

[No caso de um estágio, é nesta secção que se deverá falar da empresa em que o estágio foi realizado. Se o projeto desenvolvido faz parte de um projeto mais amplo, faz sentido que se documente os objetivos do projeto com um todo, de modo que o leitor consiga

perceber onde o trabalho realizado encaixa.]

1.3 Estrutura do documento

[A última secção da introdução deve explicar a estrutura do documento: quais são só capítulos existentes (para além do primeiro) e o que será discutido em cada um desses capítulos. A estrutura típica de um relatório de desenvolvimento de software é:

Introdução, com um breve resumo do que se pretende atingir, e uma descrição clara dos objetivos;

1. Análise ao problema, que poderá incluir uma análise ao estado da arte ou ao modelo de negócio onde se pretende intervir;
2. Análise e modelação do sistema, em que sejam levantados sistematicamente os requisitos, descritos diagramas de caso de uso e de atividade (que descrevam/formalizem o modelo de negócio).
3. Implementação, em que se descrevam as tecnologias escolhidas (e se justifiquem), e se refira detalhes sobre a implementação.
4. Análise de resultados e testes, seja uma análise/avaliação aos resultados obtidos, sejam testes de usabilidade ou unitários ao trabalho desenvolvido.
5. Conclusão.]

2. Capítulo 2

Este capítulo fala de stemmers. Mas não esquecer os Lematizadores

2.1 Sub-capítulo 1

O HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto) (HTTP) é um protocolo baseado em TCP.

2.2 Figuras

Ao contrário do Word, o \LaTeX usa um mecanismo de colocação de figuras e tabelas em que estas flutuam ao longo das páginas de acordo com a necessidade/disponibilidade em termo de espaço vertical. Assim, não devem usar frases como “na figura acima”, ou “na figura abaixo”, mas fazer referências: “tal como se pode observar na Figura 2.1.”

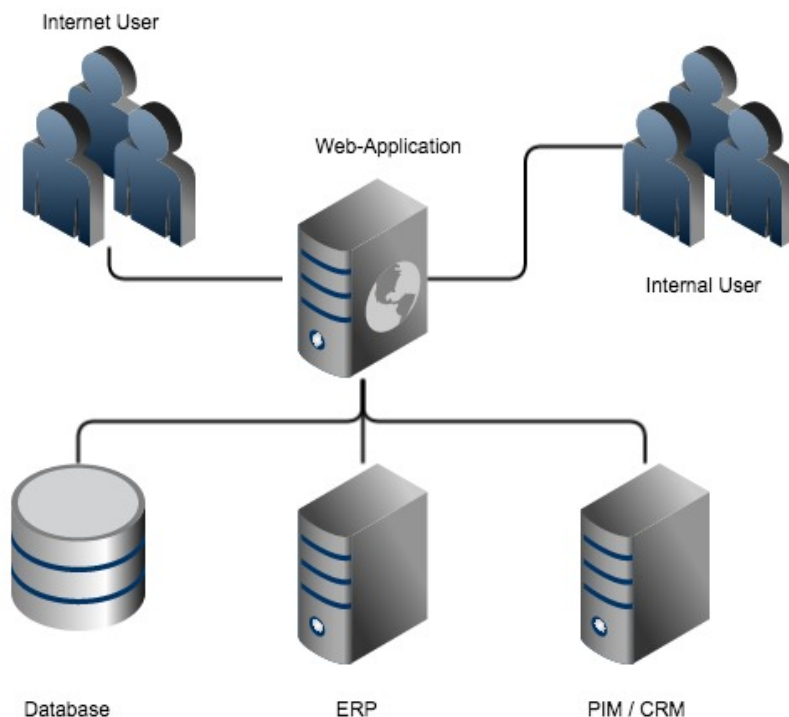


Figura 2.1: Esta é a legenda da figura

O mesmo acontece com as tabelas, como se pode ver na Tabela 2.1.

A	B	C	D	Total
1	2	3	4	10
2	3	4	5	14
3	4	5	6	18
4	5	6	7	22

Tabela 2.1: Legenda da tabela.

Para a inclusão de código, usa-se algo semelhante. Veja-se a Listagem 2.1.

```
public int count(string x) {
    return items.Select( y => y == x ).Count();
}
```

Listagem 2.1: Método para contar o número de elementos numa lista iguais a uma determinada string.

3. Capitulo 3

3.1 Sub-capitulo 1

Texto do sub-capitulo 1 que inclui um pouco de código...

3.1.1 Lista dos caracteres a ignorar

```
t_ignore = " \t\n"
```

3.2 Sub-capitulo 2

Texto do sub-capitulo 2

3.2.1 Listagem de programa

```
/**
 * programa apaga.c
 * @author #20808 Joao Carlos Pinto
 */
#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include "mytools.h"

int main(int argc, char *argv[]) {
    char buffer[MAXBUFFERSIZE];
    if (argc < 2) {
        buffer[0] = '\0';
        sprintf(buffer, "Falta: _ficheiro\n_Deve_ "+
            "utilizar -se_desta_forma:\n%s_ficheiro(s)\n",
            argv[0]);
        escrevErro(buffer);
        return 1;
    }
}
```

```

    }
    int resultado, bkErrno;
    for(int i=1; i<argc; i++){
        if ((resultado = unlink(argv[i])) == -1) {
            buffer[0] = '\0';
            bkErrno = errno;
            sprintf(buffer, "erro_(%d,_%s)_ao_apagar_o_"+
                "ficheiro:_%s\n", bkErrno, strerror(bkErrno),
                argv[i]);
            escrevErro(buffer);
            return 1;
        }
        if (resultado == 0) {
            buffer[0] = '\0';
            sprintf(buffer, "o_ficheiro_\"%s\"_foi_apagado!\n",
                argv[i]);
            escrever(buffer);
        } else {
            buffer[0] = '\0';
            bkErrno = errno;
            sprintf(buffer, "resultado_(%d,_%s)_inesperado_ao_"+
                "apagar_o_ficheiro_\"%s\"!\n", bkErrno,
                strerror(bkErrno), argv[i]);
            escrevErro(buffer);
            return 1;
        }
    }
    return 0;
}

```

Listagem 3.1: Código do programa: **apaga**.

4. Capitulo 4

4.1 Sub-capitulo 1

Texto do sub-capitulo 1

4.2 Sub-capitulo 2

Texto do sub-capitulo 2

Listagens de Código

2.1	Método para contar o número de elementos numa lista iguais a uma determinada string.	4
3.1	Código do programa: apaga	5

Bibliografia

- [1] Donald .E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*. Number v. 1–2 in Series in computer science and information processing. Addison-Wesley, 1973.