

Projeto Unicórnio

João Carlos Pinto, 20808

Orientação de
Alberto Simões

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

Alberto Simões

Resumo

Documento em construção...

Falta completar esta secção!

No final debes remover este comentário.

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.

Conteúdo

Lista de Figuras	vii
Lista de Tabelas	ix
Lista de Código	xi
Siglas & Acrónimos	xiii
Índice de Termos	xv
1 Introdução	1
1.1 Objetivos	1
1.2 Contexto	2
1.3 Estrutura do documento	2
2 Segundo capítulo	3
3 Diferentes exemplos num único capítulo	5
3.1 Preparar o ambiente LaTeX	5
3.2 Utilizar referências	5
3.3 Secção	6
3.3.1 Subsecção	6
3.4 Utilizar lista1	6
3.5 Utilizar lista2	6
3.6 Exemplo de colocação de figuras	6
3.7 Exemplo de tabela	7
3.8 Utilizar Verbatim	8
3.8.1 Lista dos caracteres a ignorar	8
3.9 Utilizar listagem de programa	8
3.9.1 Listagem de programa em C#	8

3.9.2	Listagem de programa em C	8
Bibliografia		11

Lista de Figuras

3.1	Figura para o exemplo	7
-----	---------------------------------	---

Lista de Tabelas

3.1	Tabela exemplo	7
-----	--------------------------	---

Listagens de Código

3.1	Método para contar o número de elementos numa lista iguais a uma determinada string.	8
3.2	Código do programa: apaga	8

Siglas & Acrónimos

HTTP HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto). 5

TCP Transmission Control Protocol. 5

Índice de Termos

lematizador Com semelhanças com o Stemmer, também reduz uma palavra ao seu lema, que corresponde ao verbo no infinitivo no caso dos verbos, e ao masculino singular, no caso de nomes ou adjetivos. . 5

stemmer Ferramenta capaz de reduzir uma palavra à sua raiz. Por exemplo, para a palavra “correria”, a sua raiz seria “corre”. . 5

1. Introdução

Documento em construção...

O texto neste capítulo explica a estrutura do documento. Depois de leres e compreenderes deves alterar o texto contextualizado com o tema do teu projeto. No final deves remover este comentário.

Se necessitares de ajuda para escreveres em L^AT_EX podes e deves consultar o capítulo com o código exemplo na página 5.

[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. Segundo capitulo

Blá, blá, blá...

Documento em construção...

Este capítulo está vazio e é intencional, debes remover ou substituir o conteúdo e apagar este comentário.

3. Diferentes exemplos num único capítulo

Esta é a segunda versão do documento utilizado como template para o relatório de trabalhos e/ou projetos desenvolvidos para cada uma das Unidades Curriculares da Licenciatura de Engenharia de Sistemas Informáticos na Escola Superior de Tecnologia¹.

NOTA: Documento em construção! Este capítulo 3 deve ser removido na versão final deste documento...

3.1 Preparar o ambiente LaTeX

Assumindo que estás a utilizar o sistema operativo Windows, para editares documentos LaTeX necessitas de instalar as seguintes ferramentas:

- **MiKTeX**², esta ferramenta funciona na linha de comandos (de um terminal) e disponibiliza um conjunto de comandos necessários para compilar o código e produzir o documento final. No repositório tens dois scripts³ para gestão do projeto:
 - **tex-win-make.bat** executa a sequência completa para compilar o projeto;
 - **tex-win-clear-temp-files.bat** limpa os ficheiros desnecessários criados durante a compilação do projeto.
- **TeXstudio**⁴, é um excelente IDE para LaTeX.

3.2 Utilizar referências

Este capítulo fala de stemmers. Mas não esquecer os Lematizadores

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

Citar o livro "The Art of Computer Programming"[1].

A introdução está colocada na página 1.

¹<https://est.ipca.pt/>

²<https://miktex.org/>

³no formato *batch processing file*, mas também foram criados para utilizar num terminal do MacOS

⁴<https://texstudio.org/>

3.3 Secção

Blá, blá, blá...

documento em construção...

3.3.1 Subsecção

Blá, blá, blá...

documento em construção...

Subsubsecção

Blá, blá, blá...

3.4 Utilizar lista1

Texto de suporte para um exemplo de utilização de lista.

- Item 1;
- Item 2;
- Item 3;
- Conclusão.

3.5 Utilizar lista2

Texto de suporte para outro exemplo de utilização de lista.

1. Item 1;
2. Item 2;
3. Item 3;
4. Conclusão.

3.6 Exemplo de colocação de figuras

Ao contrário do Word, o L^AT_EX 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 3.1” (a figura poderá estar numa página diferente, portanto se for muito importante indicar a página, necessitas apenas de colocar a referência para essa página 7 caso seja pertinente e necessário).

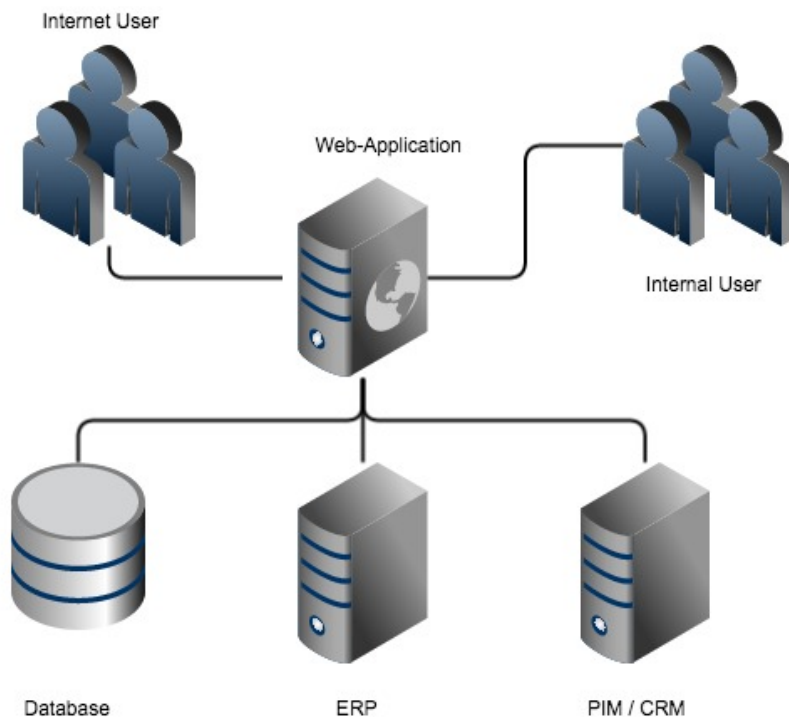


Figura 3.1: Figura para o exemplo

3.7 Exemplo de tabela

Texto de suporte para um exemplo de tabela.

O mesmo acontece com as tabelas, como se pode ver na Tabela 3.1 (a tabela poderá estar numa página diferente, portanto se for muito importante indicar a página, necessitas apenas de colocar a referência para essa página 7 caso seja pertinente e necessário).

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 3.1: Tabela exemplo

3.8 Utilizar Verbatim

Texto de suporte para o exemplo de utilização do bloco Verbatim.

3.8.1 Lista dos caracteres a ignorar

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

3.9 Utilizar listagem de programa

Texto de suporte para o exemplo de utilização de listagem.

3.9.1 Listagem de programa em C#

Para a inclusão de código, usa-se algo semelhante. Veja-se a Listagem 3.1.

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

Listagem 3.1: Método para contar o número de elementos numa lista iguais a uma determinada string.

3.9.2 Listagem de programa em C

Outro exemplo de código, usa-se algo semelhante. Veja-se a Listagem 3.2.

```
/**
 * 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 3.2: Código do programa: **apaga**.

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.