

Trabalho de aprofundamento AP2

André Patacas, Gil Teixeira



Aplicação para o cálculo de Largura de Banda e de latência

DETI

André Patacas, Gil Teixeira
(93357) andrepatacas@ua.pt, (88194) gilteixeira@ua.pt

9 de Abril de 2019

Resumo

Este relatório serve para descrever uma ferramenta desenvolvida para calcular a largura de banda e a latência da máquina, onde a aplicação se encontra a correr, a um determinado servidor ou a um conjunto, de cardinalidade especificável, de servidores de um país, sendo este também especificável.

Agradecimentos

Eventuais agradecimentos. Comentar bloco caso não existam agradecimentos a fazer.

Conteúdo

1	Introdução	1
2	Metodologia	2
2.1	Exemplos	2
2.1.1	Utilização de acrónimos	2
2.1.2	Referências bibliográficas	2
3	Resultados	3
4	Análise	4
5	Conclusões	5

Capítulo 1

Introdução

A aplicação foi desenvolvida em python3 no âmbito da disciplina de Laboratórios de Informática, no ano letivo 2018/2019. A adicionar às especificações básicas pedidas, segundo o guião sobre regras do segundo trabalho de aprofundamento, construi-se ainda suporte para pydocs para haver uma explicação mais detalhada sobre cada método e classe no nosso projeto. O programa foi escrito com base em test driven development (Capítulo 2) e como tal os testes unitários e funcionais foram criados primeiro, seguidos por um esqueleto do programa e finalmente por vários updates a ambos (chap.resultados) para chegar ao estado em que a aplicação se encontra de momento (Capítulo 4). Finalmente são tiradas as conclusões sobre os aspetos positivos e, potencialmente, negativos desta solução em concreto (Capítulo 5)

Capítulo 2

Metodologia

1. Criar o esqueleto do programa que é agora o inicializador da classe se esta for chamada como main. (if `__name__ == '__main__':`);
2. Criar o ficheiro `test_labi_02` como um teste que, apenas se a construção da aplicação for robusta e exatamente como especificada, passa.
3. Criar o programa `labi_02` e definir as funções com os argumentos de entrada e cada uma com uma descrição detalhada, disponível nos pydocs, dos aspetos funcionais de cada função.
4. Ajustar os métodos de forma a que a aplicação passa todos os testes impostos no teste criado.
5. Testar o programa manualmente e/ou com testes funcionais.
6. Corrigir eventuais erros.
7. Iterar o processo de debugging e correção de erros.

2.1 Exemplos

2.1.1 Utilização de acrónimos

Esta é a primeira invocação do acrónimo Universidade de Aveiro (UA). E esta é a segunda: UA.

Outras duas referências a Mestrado Integrado em Engenharia de Computadores e Telemática (MIECT) e MIECT.

2.1.2 Referências bibliográficas

Informação relativa à estrutura formal de um relatório pode ser obtida na página do Grey Literature International Steering Committee (GLISC)[glisc](#).

Capítulo 3

Resultados

Descreve os resultados obtidos.

Capítulo 4

Análise

Analisa os resultados.

Capítulo 5

Conclusões

Apresenta conclusões.

Contribuições dos autores

Resumir aqui o que cada autor fez no trabalho. Usar abreviaturas para identificar os autores, por exemplo AS para António Silva. No fim indicar a percentagem de contribuição de cada autor.

Acrónimos

UA Universidade de Aveiro

MIECT Mestrado Integrado em Engenharia de Computadores e Telemática

GLISC Grey Literature International Steering Committee