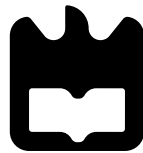


# Trabalho de aprofundamento AP2

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

# Conteúdo

|          |                                |          |
|----------|--------------------------------|----------|
| <b>1</b> | <b>Introdução</b>              | <b>1</b> |
| <b>2</b> | <b>Metodologia</b>             | <b>2</b> |
| <b>3</b> | <b>Aplicação de Speed Test</b> | <b>3</b> |
| 3.1      | Labi_02 . . . . .              | 3        |
| 3.1.1    | calc_download . . . . .        | 3        |
| 3.1.2    | calc_latency . . . . .         | 3        |
| 3.1.3    | country_test . . . . .         | 3        |
| <b>4</b> | <b>Resultados</b>              | <b>5</b> |
| <b>5</b> | <b>Análise</b>                 | <b>6</b> |
| <b>6</b> | <b>Conclusão</b>               | <b>7</b> |

## **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.

# 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 (??). Finalmente são tiradas as conclusões sobre os aspetos positivos e, potencialmente, negativos desta solução em concreto (??)

## Capítulo 2

# Metodologia

1. Criar o esqueleto do programa que é agora o inicializador da classe (labi02) se esta for chamada diretamente;
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.

## Capítulo 3

# Aplicação de Speed Test

### 3.1 Labi\_02

Esta é a aplicação que foi desenvolvida e que pode ser utilizada diretamente de acordo com o usage demonstrado ao correr a aplicação sem argumentos. Toda a descrição feita neste relatório remete na mesma para a documentação, esta criada a quando do desenvolvimento da aplicação.

#### 3.1.1 calc\_download

Cálculo da largura de banda:

Este método pede, inicialmente, para fazer um download de 100 megabytes ao target server dentro de 10 segundos. Depois verifica que não há mais data para ser recebida do *target\_server* e finalmente calcula o time download 1mb que a máquina demora a fazer download de 1 megabyte.

**Argumentos:** *target\_server*(dicionário com informação sobre o target server).

**Retorna:** float( $1/time\_download\_1mb$ )

#### 3.1.2 calc\_latency

Cálculo da latência:

Este método tenta trocar dez comandos PING-PONG com o targe-server e calcula o tempo médio em milisegundos entre estas trocas.

**Argumentos:** target server(dicionário com informação sobre o target server).

**Retorna:** int(tempo entre trocas em ms).

#### 3.1.3 country\_test

Este método serve para calcular o tempo de download e latency a um servidor random do país passado como argumento:

**Argumentos:** target country(str).

**Retorna:** objeto *SpeedTestResult* com as informações relativas aos resultados do teste.



## Capítulo 4

# Resultados

## Capítulo 5

# Análise

## Capítulo 6

## Conclusão