

# Relatório do Trabalho de Aprofundamento 2

Universidade de Aveiro

Adalberto Júnior da Trindade Vaz do Rosário,  
Bruno Araújo Acioli Lins



VERSÃO FINAL

# Relatório do Trabalho de Aprofundamento 2

Departamento de Electrónica, Telecomunicações e  
Informática

Universidade de Aveiro

Adalberto Júnior da Trindade Vaz do Rosário, Bruno Araújo Acioli Lins  
(105589) adalberto.rosario@ua.pt, (101077) baal@ua.pt

30 de Maio de 2021

### **Resumo**

Este relatório tem como intuito de apresentar o trabalho de aprofundamento 2 da cadeira de Laboratório de Informática, feito no segundo semestre do primeiro ano do curso de Mestrado Integrado em Engenharia de Computadores e Telemática. Em que o mesmo é feito em grupo de dois alunos. O referido trabalho teve o início no princípio do mês do maio e com a data de entrega prevista para o dia 30-05-2021.

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Apresentação</b>	<b>2</b>
<b>3</b>	<b>Implementação</b>	<b>3</b>
<b>4</b>	<b>Dificuldades</b>	<b>10</b>
<b>5</b>	<b>Conclusão</b>	<b>11</b>

# Lista de Figuras

3.1	Código do teste funcional do server.py . . . . .	3
3.2	Código do teste funcional do client.py . . . . .	4
3.3	Início do jogo . . . . .	4
3.4	Início do jogo com segundo cliente . . . . .	5
3.5	Funcionamento correcto do jogo . . . . .	5
3.6	Operação Quit . . . . .	6
3.7	Fazendo Quit antes do registo do cliente . . . . .	6
3.8	Fazendo operação guess antes do registo do cliente . . . . .	7
3.9	Esgotado as tentativas . . . . .	7
3.10	Fim dos testes . . . . .	8
3.11	Registo dos clientes . . . . .	8

# Capítulo 1

## Introdução

Com objetivo e muita satisfação, devoção de guiar e mostrar um pequeno resumo do nosso trabalho é feito este relatório, em que a nossa motivação resume-se em tornar o referido trabalho mais compreensível. Este relatório é composto por 4 capítulos, em que depois desta introdução, no Capítulo 2, é apresentado o grupo, bem como os autores que o compõe, no Capítulo 3 é apresentado um pouco de trabalho, onde será mostrado a implementação e outras informações sobre o mesmo. Em seguida, no Capítulo 4 é apresentado as principais dificuldades que os autores tiveram na implementação do trabalho e finalmente, no Capítulo 5 são apresentadas as conclusões do trabalho.

## Capítulo 2

# Apresentação

Como já referido, o trabalho é elaborado por grupo de dois alunos, e este é o do grupo 17. Sendo o mesmo composto por:

- Adalberto Júnior da Trindade Vaz do Rosário, número mecanográfico 105589, email: [adalberto.rosario@ua.pt](mailto:adalberto.rosario@ua.pt). Aluno da Turma P3.  
E por :
- Bruno Araújo Acioli Lins, número mecanográfico: 101077, email: [baal@ua.pt](mailto:baal@ua.pt). Aluno da Turma P5.

Somos Estudantes da Universidade de Aveiro, inscritos no Mestrado Integrado em Engenharia de Computadores e Telemática. Este trabalho é o segundo trabalho de aprofundamento referente a disciplina Laboratório de Informática (LABI) do curso em questão.



## Capítulo 3

# Implementação

Para confirmar o funcionamento do programa fizemos testes unitários e também testes funcionais, em que iremos disponibilizar abaixo de acordo com cada teste realizado.

- Testes funcionais.

Nestas imagens é mostrado um pouco do código dos teste funcionais implementados.

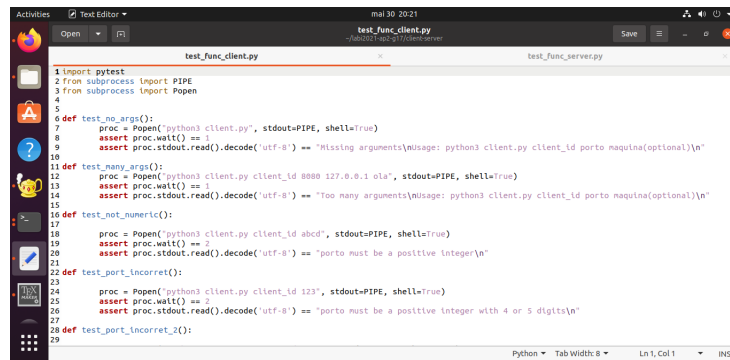
A screenshot of a text editor window titled 'Text Editor' with a dark theme. The window shows two tabs: 'test\_func\_client.py' and 'test\_func\_server.py'. The 'test\_func\_server.py' tab is active, displaying Python code for functional tests. The code includes imports for 'pytest', 'subprocess', and 'PIPE'. It defines four test functions: 'test\_no\_args()', 'test\_not\_numeric()', 'test\_port\_incorret\_server()', and 'test\_port\_incorret\_server\_2()'. Each function uses 'Popen' to run 'python3 server.py' with specific arguments and then asserts the output. The status bar at the bottom indicates 'Python', 'Tab Width: 8', 'Ln 1, Col 1', and 'INS'.

```
1 import pytest
2 from subprocess import PIPE
3 from subprocess import Popen
4
5
6 def test_no_args():
7     proc = Popen("python3 server.py", stdout=PIPE, shell=True)
8     assert proc.wait() == 1
9     assert proc.stdout.read().decode('utf-8') == "Usage: python3 server.py porto\n"
10
11 def test_not_numeric():
12     proc = Popen("python3 server.py abcd", stdout=PIPE, shell=True)
13     assert proc.wait() == 2
14     assert proc.stdout.read().decode('utf-8') == "porto must be a positive integer\n"
15
16 def test_port_incorret_server():
17
18     proc = Popen("python3 server.py 123", stdout=PIPE, shell=True)
19     assert proc.wait() == 2
20     assert proc.stdout.read().decode('utf-8') == "porto must be a positive integer with 4 or 5 digits\n"
21
22 def test_port_incorret_server_2():
23     proc = Popen("python3 server.py 123666", stdout=PIPE, shell=True)
24     assert proc.wait() == 2
25     assert proc.stdout.read().decode('utf-8') == "porto must be a positive integer with 4 or 5 digits\n"
26
27
```

Figura 3.1: Código do teste funcional do server.py

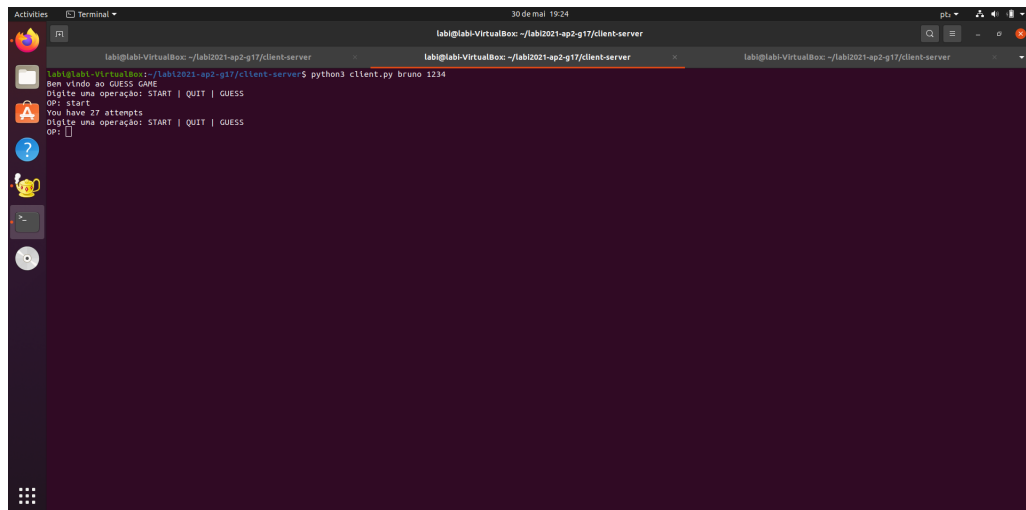
- Testes Unitários

As imagens disponibilizadas abaixo dizem respeito ao teste untário aplicado no programa.



```
1 import pytest
2 from subprocess import PIPE
3 from subprocess import Popen
4
5
6 def test_no_args():
7     proc = Popen('python3 client.py', stdout=PIPE, shell=True)
8     assert proc.wait() == 1
9     assert proc.stdout.read().decode('utf-8') == "Missing arguments\nusage: python3 client.py client_id porto maquina(optional)\n"
10
11 def test_many_args():
12     proc = Popen('python3 client.py client_id 8080 127.0.0.1 ola', stdout=PIPE, shell=True)
13     assert proc.wait() == 1
14     assert proc.stdout.read().decode('utf-8') == "Too many arguments\nusage: python3 client.py client_id porto maquina(optional)\n"
15
16 def test_not_numeric():
17     proc = Popen('python3 client.py client_id abcd', stdout=PIPE, shell=True)
18     assert proc.wait() == 2
19     assert proc.stdout.read().decode('utf-8') == "porto must be a positive integer\n"
20
21
22 def test_port_incorret():
23
24     proc = Popen('python3 client.py client_id 123', stdout=PIPE, shell=True)
25     assert proc.wait() == 2
26     assert proc.stdout.read().decode('utf-8') == "porto must be a positive integer with 4 or 5 digits\n"
27
28 def test_port_incorret_2():
29
```

Figura 3.2: Código do teste funcional do client.py



```
labig@labi-VirtualBox: ~/labi2021-ap2-g17/client-server
labig@labi-VirtualBox: ~/labi2021-ap2-g17/client-server$ python3 client.py bruno 1234
Ben vindo ao GUESS GAME
Digite uma operação: START | QUIT | GUESS
OP: start
You have 27 attempts
Digite uma operação: START | QUIT | GUESS
OP: 
```

Figura 3.3: Início do jogo

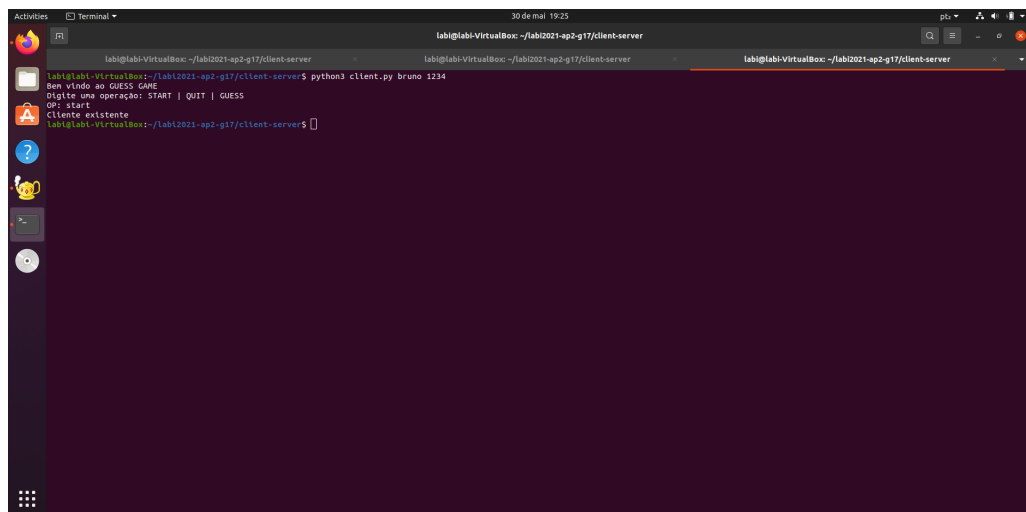


Figura 3.4: Início do jogo com segundo cliente

Na figura 3.3, é iniciado o jogo com o user chamado bruno. E com isso a operação Start foi realizada com sucesso. Na figura 3.4, é iniciado o jogo mais uma vez com o cliente bruno e deu erro, isto porque ele já está ativo.

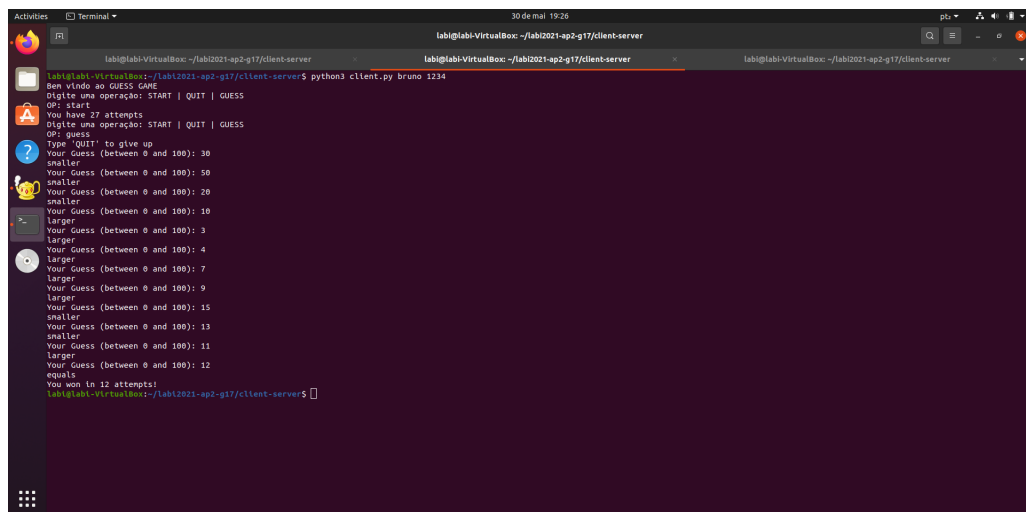
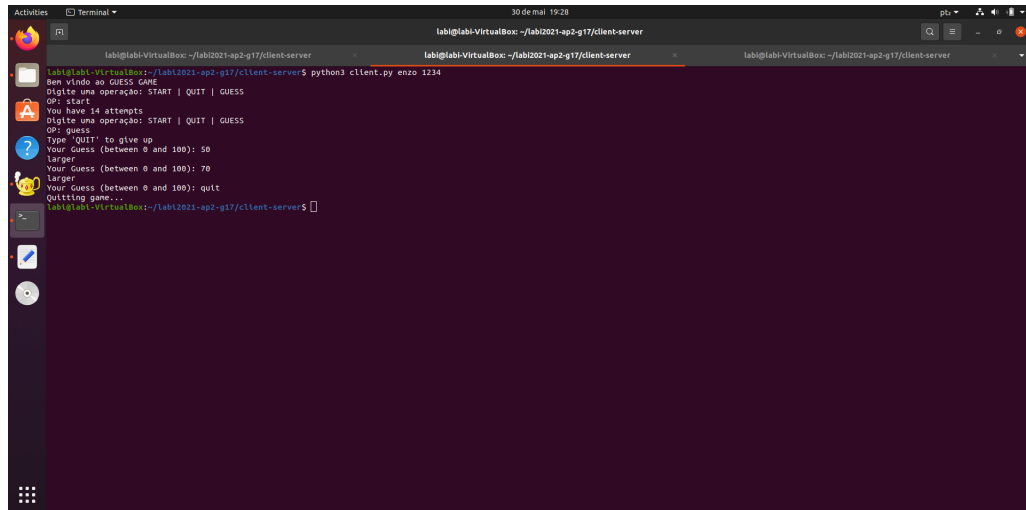


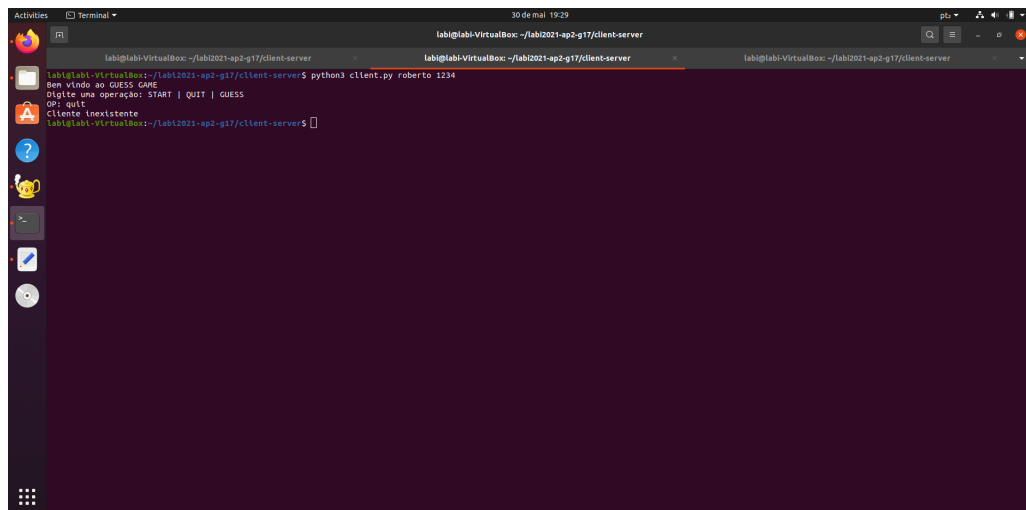
Figura 3.5: Funcionamento correcto do jogo

Na figura 3.5 é apresentado o funcionamento correcto do jogo, na figura 3.6, o jogo é iniciado com o cliente chamado enzo mas o mesmo desistiu no meio do caminho, testando assim a operação Quit. Já na figura 3.7 é feito a



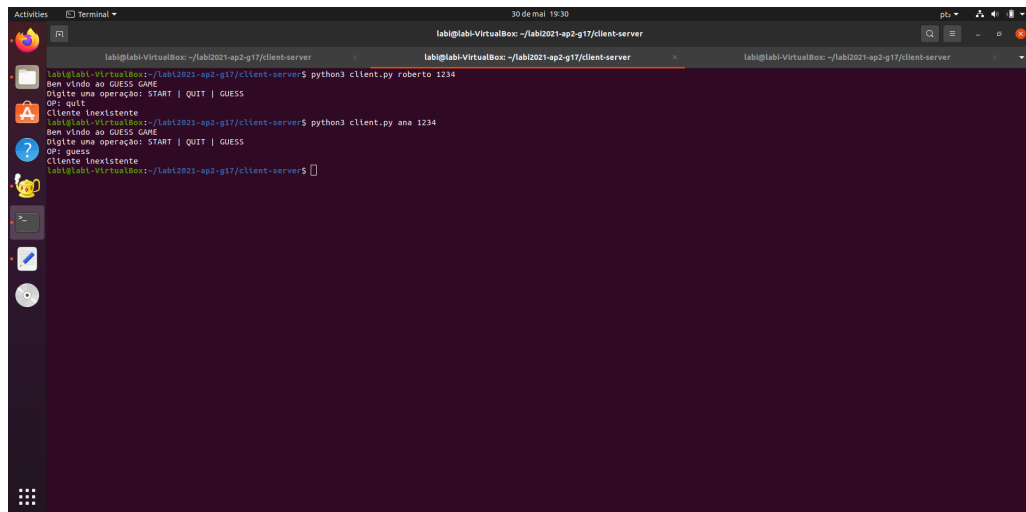
```
labiglabi-VirtualBox: ~/labi2021-ap2-g17/client-server
labiglabi-VirtualBox: ~/labi2021-ap2-g17/client-server$ python3 client.py enzo 1234
Ben vindo ao GUESS GAME
Digite uma operação: START | QUIT | GUESS
OP: start
You have 14 attempts
Digite uma operação: START | QUIT | GUESS
OP: guess
Type 'QUIT' to give up
Your Guess (between 0 and 100): 50
larger
Your Guess (between 0 and 100): 70
larger
Your Guess (between 0 and 100): quit
quitting game...
labiglabi-VirtualBox: ~/labi2021-ap2-g17/client-server$
```

Figura 3.6: Operação Quit



```
labiglabi-VirtualBox: ~/labi2021-ap2-g17/client-server
labiglabi-VirtualBox: ~/labi2021-ap2-g17/client-server$ python3 client.py roberto 1234
Ben vindo ao GUESS GAME
Digite uma operação: START | QUIT | GUESS
OP: quit
Cliente inexistente
labiglabi-VirtualBox: ~/labi2021-ap2-g17/client-server$
```

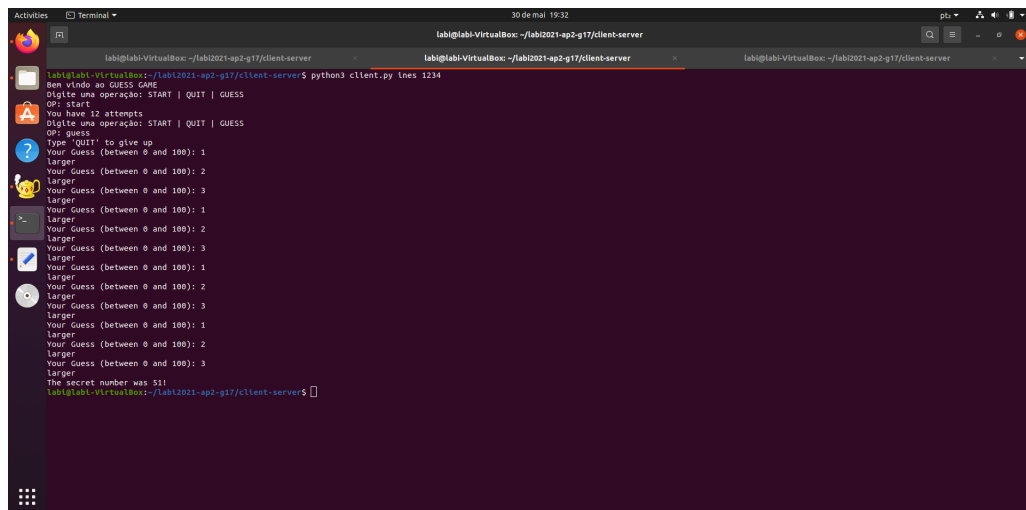
Figura 3.7: Fazendo Quit antes do registo do cliente



A terminal window titled 'lab1@lab1-VirtualBox: ~/lab12021-ap2-g17/client-server' showing the execution of a Python script. The script prompts for a client name and then a guess. The user enters 'roberto' and '1234'. The script outputs 'Ben vindo ao GUESS GAME', 'Digite uma operação: START | QUIT | GUESS', and 'OP: guess'. It then checks for the client and outputs 'Cliente Inexistente'.

```
lab1@lab1-VirtualBox:~/lab12021-ap2-g17/client-server$ python3 client.py roberto 1234
Ben vindo ao GUESS GAME
Digite uma operação: START | QUIT | GUESS
OP: guess
Cliente Inexistente
lab1@lab1-VirtualBox:~/lab12021-ap2-g17/client-server$
```

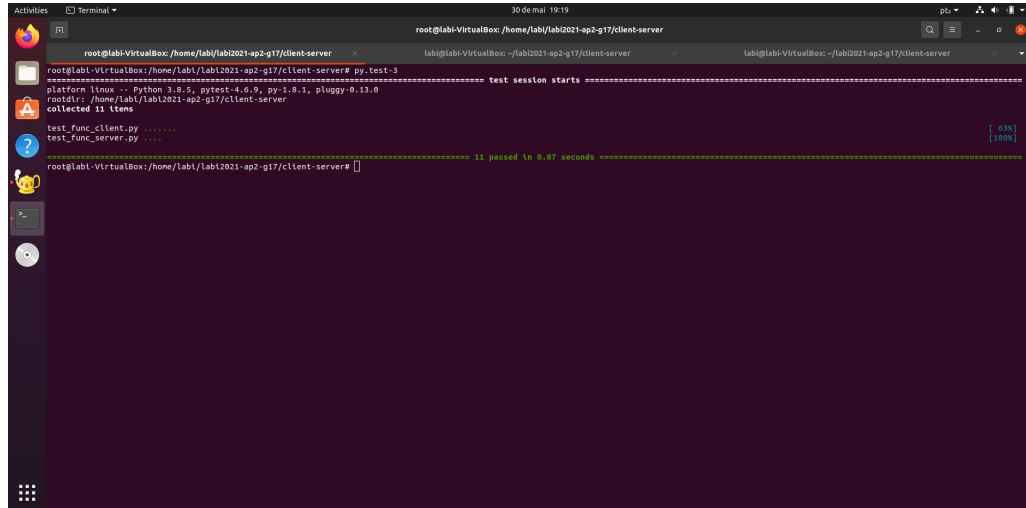
Figura 3.8: Fazendo operação guess antes do registo do cliente



A terminal window titled 'lab1@lab1-VirtualBox: ~/lab12021-ap2-g17/client-server' showing the execution of a Python script. The script prompts for a client name and then a guess. The user enters 'lmes' and '1234'. The script outputs 'Ben vindo ao GUESS GAME', 'Digite uma operação: START | QUIT | GUESS', and 'OP: start'. It then prompts for a guess and outputs 'You have 12 attempts'. The user enters '1', '2', '3', '1', '2', '3', '1', '2', '3', '1', '2', '3'. The script outputs 'Your Guess (between 0 and 100): 1', 'larger', 'Your Guess (between 0 and 100): 2', 'larger', 'Your Guess (between 0 and 100): 3', 'larger', 'Your Guess (between 0 and 100): 1', 'larger', 'Your Guess (between 0 and 100): 2', 'larger', 'Your Guess (between 0 and 100): 3', 'larger', 'Your Guess (between 0 and 100): 1', 'larger', 'Your Guess (between 0 and 100): 2', 'larger', 'Your Guess (between 0 and 100): 3', 'larger'. Finally, it outputs 'The secret number was 511'.

```
lab1@lab1-VirtualBox:~/lab12021-ap2-g17/client-server$ python3 client.py lmes 1234
Ben vindo ao GUESS GAME
Digite uma operação: START | QUIT | GUESS
OP: start
You have 12 attempts
Digite uma operação: START | QUIT | GUESS
OP: guess
Type 'QUIT' to give up
Your Guess (between 0 and 100): 1
larger
Your Guess (between 0 and 100): 2
larger
Your Guess (between 0 and 100): 3
larger
Your Guess (between 0 and 100): 1
larger
Your Guess (between 0 and 100): 2
larger
Your Guess (between 0 and 100): 3
larger
Your Guess (between 0 and 100): 1
larger
Your Guess (between 0 and 100): 2
larger
Your Guess (between 0 and 100): 3
larger
Your Guess (between 0 and 100): 1
larger
Your Guess (between 0 and 100): 2
larger
Your Guess (between 0 and 100): 3
larger
The secret number was 511
lab1@lab1-VirtualBox:~/lab12021-ap2-g17/client-server$
```

Figura 3.9: Esgotado as tentativas



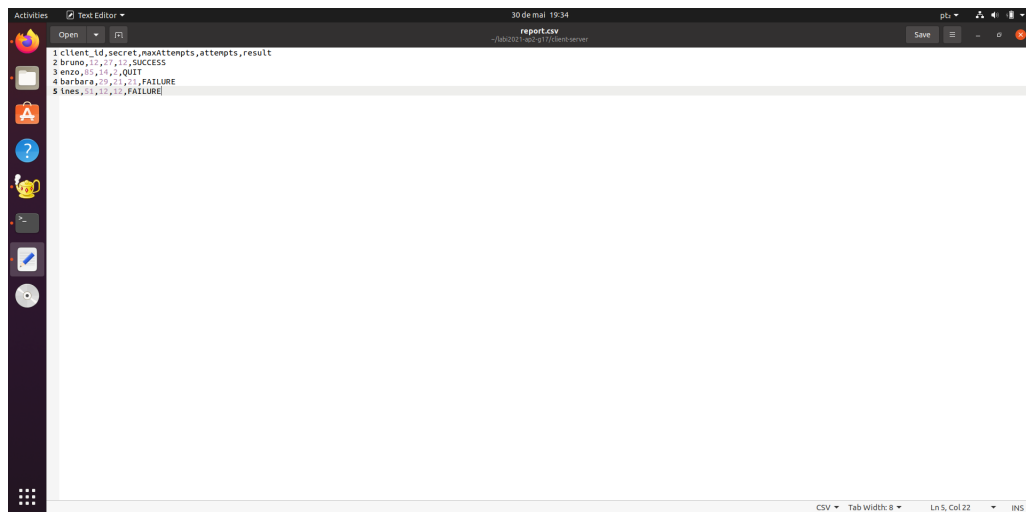
A terminal window titled 'root@lab1-VirtualBox: /home/lab1/lab12021-ap2-g17/client-server'. The terminal shows the execution of a test suite. It starts with 'platform linux -- Python 3.8.5, pytest-4.6.0, py-s1.0.1, pluggy-0.13.0' and 'collected 11 items'. It then shows 'test\_func\_client.py' and 'test\_func\_server.py' being executed. The output indicates that 11 tests passed in 0.87 seconds. The terminal also shows the path 'root@lab1-VirtualBox: /home/lab1/lab12021-ap2-g17/client-server#'.

```
root@lab1-VirtualBox: /home/lab1/lab12021-ap2-g17/client-server# py.test-3
platform linux -- Python 3.8.5, pytest-4.6.0, py-s1.0.1, pluggy-0.13.0
collected 11 items

test_func_client.py ..... [ 63%]
test_func_server.py ..... [100%]

===== 11 passed in 0.87 seconds =====
root@lab1-VirtualBox: /home/lab1/lab12021-ap2-g17/client-server#
```

Figura 3.10: Fim dos testes



A text editor window titled 'report.csv' showing a CSV file with client registration data. The data is as follows:

client_id	secret	maxAttempts	attempts	result
2	bruno	10	10	SUCCESS
3	enzo	10	10	QUIT
4	barbara	10	10	FAILURE
5	ines	10	10	FAILURE

Figura 3.11: Registo dos clientes

operação Quit antes de registrar o cliente, em que deve apresentar a mensagem "Cliente inexistente", fazendo a operação guess antes de registrar o cliente deve apresentar o mesmo erro apresentado na figura 3.7, ou seja, "cliente inexistente", e este resultado é mostrado na figura 3.8. Uma das recomendações propostas é que o cliente tenha o número de tentativas limitadas, e o teste do mesmo é mostrado na figura 3.9. Após o esgotamento do número de tentativas é dito o número secreto e o jogo termina. Na figura 3.10 é mostrado que todos os teste executado por nós foram concluídos com sucesso. E finalmente é mostrado na figura 3.11 o print do report.csv com o registo dos clientes.

## Capítulo 4

# Dificuldades

Ao longo do nosso percurso na elaboração do trabalho nós deparamos com imensas dificuldades, tais como: compreensão e análise do problema que condicionou o início do trabalho, entender o funcionamento das funções e implementar a função stop. Mas com muito empenho conseguimos superar essas dificuldades.



## Capítulo 5

# Conclusão

Quando iniciamos o trabalho deparamos com muitas dificuldades na compreensão do problema e isso condicionou um atraso na implementação do mesmo. Mas após muitas leituras e análises dos problemas e também ajuda dos nossos professores conseguimos superar essas dificuldades e finalizar com êxito o trabalho. Com o mesmo podemos compreender e aprender mais sobre python e conseguimos esclarecer muitas dúvidas que tínhamos sobre esta linguagem. Realizamos diversos testes e os resultados permitiram concluir que finalizamos corretamente as recomendações disponibilizadas no enunciado dado. Apesar de muito esforço para terminar o trabalho não conseguimos concluir a parte de encriptamento dos dados.

Por fim, como uma das nossas principais dificuldades foi a compreensão do problema, gostaríamos de deixar como sugestão para os próximos trabalhos a disponibilização de um mini projeto exemplo de forma que possamos estar familiarizados com os projetos do gênero.

# Contribuições dos autores

O trabalho foi dividido de forma que cada um podesse ajudar da melhor forma focando no que dominamos individualmente, sendo:

- Bruno Araújo Acioli Lins: Por compreender bem mais o trabalho proposto e por dominar um pouco da linguagem Python mais do que o meu companheiro, foquei mais na criação e implementação do projeto. Porém, estive sempre em sintonia com o meu companheiro do grupo a partilhar ideias e a esclarecer as eventuais dúvidas surgidas.
- Adalberto Júnior Vaz do Rosário: Tive muita mais dificuldades na compreensão e implementação do trabalho em relação ao meu companheiro, então foquei mais no relatório e na implementação de algumas funções que pude compreender. No entanto, estive sempre em sintonia com o meu companheiro do grupo de forma a esclarecer as eventuais dúvidas surgidas e também a partilhar as ideias.

Cada um concluiu com êxito as suas tarefas, estando sempre em sintonia com commits frequentes e também trocando ideias e a esclarecer as dúvidas. E por assim dizer concluímos que a nota deve ser distribuída igualmente a cada um dos autores.