



Maestría en Inteligencia Artificial

Pruebas de software y aseguramiento de la calidad

Actividad 4.2

Ejercicio de programación 1

Profesor Titular: Dr. Gerardo Padilla Zárate

Profesor Asistente: Mtra. Yetnalezi Quintas Ruiz

Alumno:

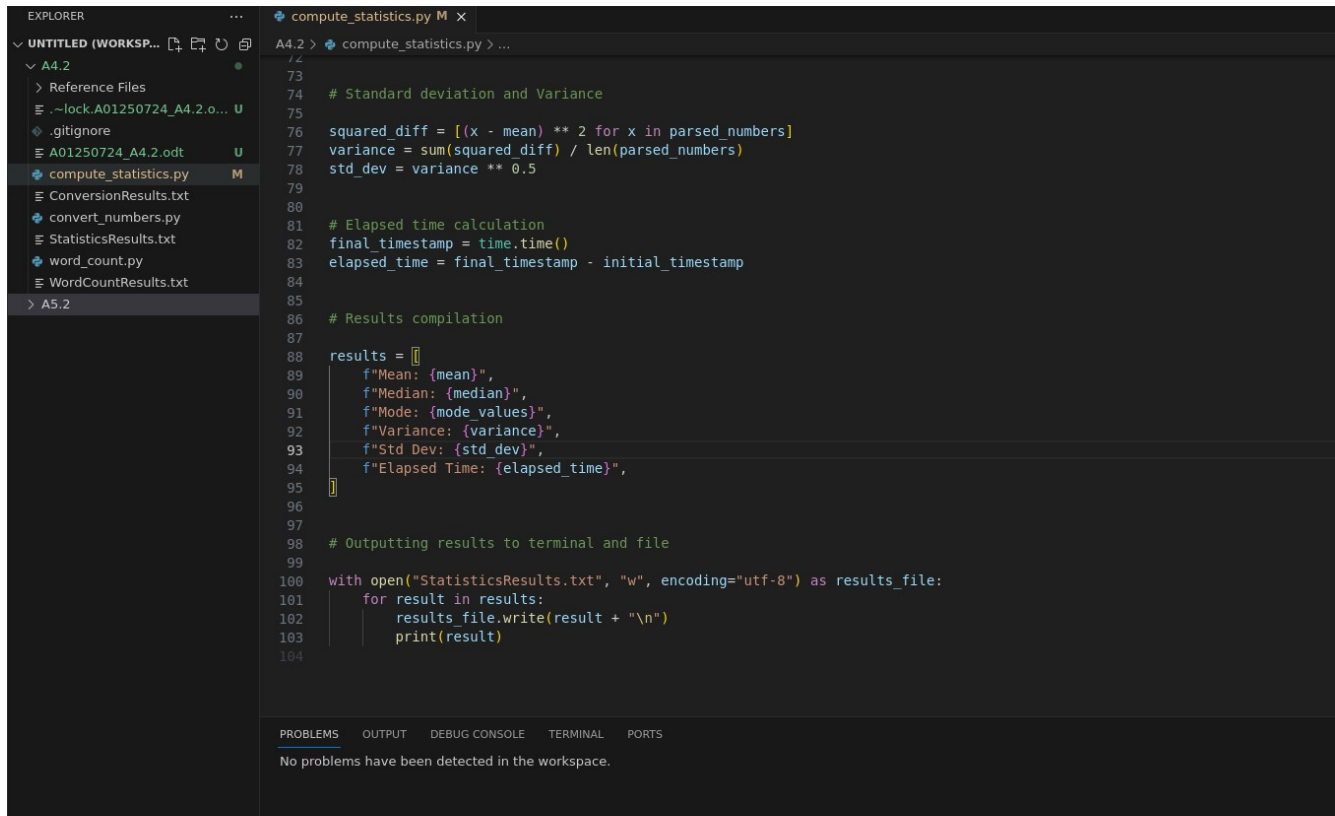
Jorge Arturo Federico Rivera (A01250724)

04 de febrero de 2024

## Ejercicio 1

### Pylint y flake8


No se muestra ningún problema dentro de VSCode:



The screenshot shows the VS Code interface with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows a project structure with a folder 'A4.2' containing several files, including 'compute\_statistics.py'. The Editor window displays the code for 'compute\_statistics.py', which includes calculations for mean, median, mode, variance, standard deviation, and elapsed time. The bottom status bar indicates 'No problems have been detected in the workspace.'

```
14
73
74 # Standard deviation and Variance
75
76 squared_diff = [(x - mean) ** 2 for x in parsed_numbers]
77 variance = sum(squared_diff) / len(parsed_numbers)
78 std_dev = variance ** 0.5
79
80
81 # Elapsed time calculation
82 final_timestamp = time.time()
83 elapsed_time = final_timestamp - initial_timestamp
84
85
86 # Results compilation
87
88 results = []
89     f"Mean: {mean}",
90     f"Median: {median}",
91     f"Mode: {mode values}",
92     f"Variance: {variance}",
93     f"Std Dev: {std_dev}",
94     f"Elapsed Time: {elapsed_time}",
95 ]
96
97
98 # Outputting results to terminal and file
99
100 with open("StatisticsResults.txt", "w", encoding="utf-8") as results_file:
101     for result in results:
102         results_file.write(result + "\n")
103         print(result)
104
```

Resultados de los Test Cases:



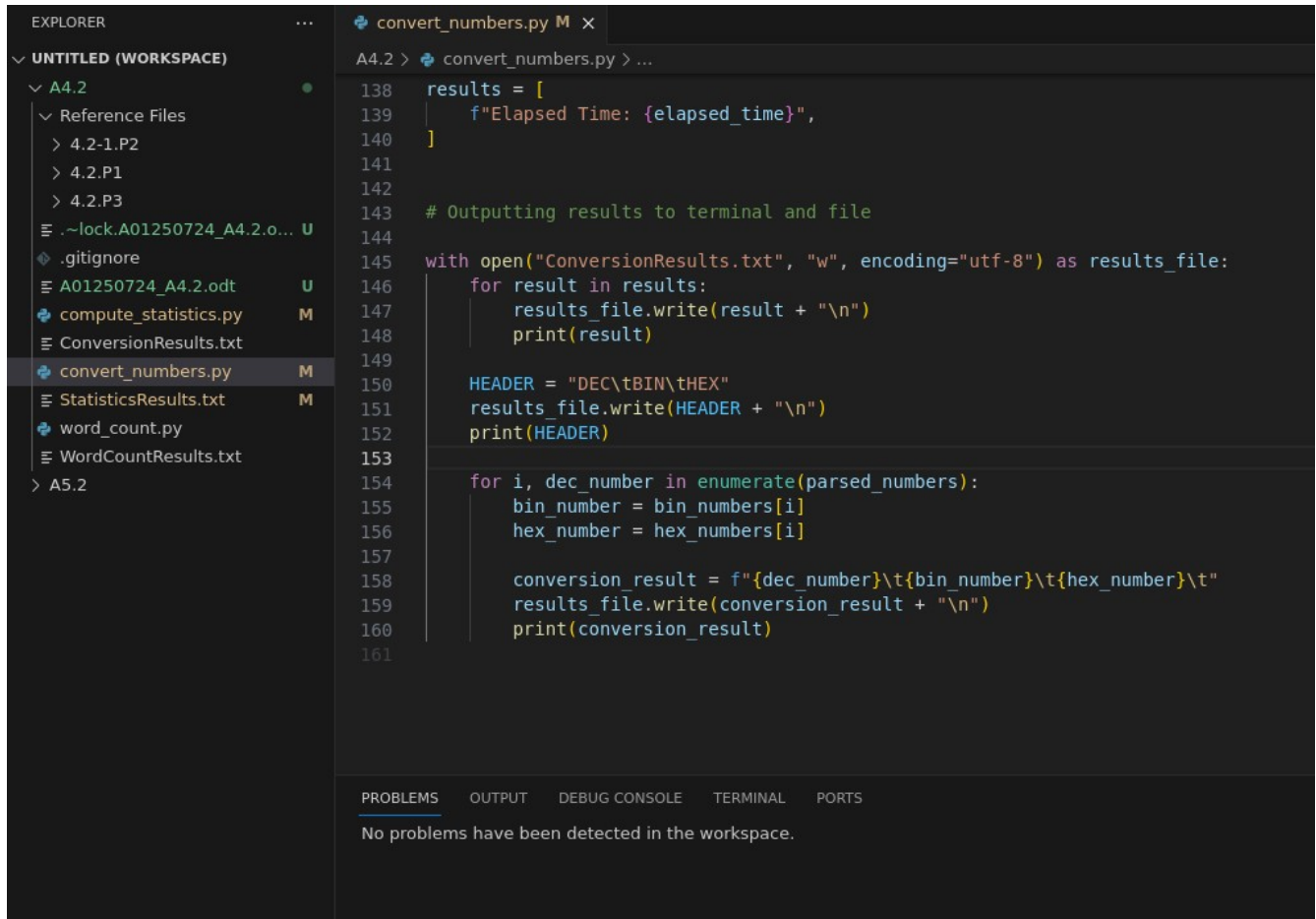
The screenshot shows the terminal output of the test cases. Each test case (TC1, TC2, TC3) is executed using the command 'sh-5.2\$ /bin/python /home/deck/Documents/School/Pruebas/A4.2/compute\_statistics.py Reference\ Files/4.2.P1/P1/TC1.txt'. The output for each test case includes the mean, median, mode, variance, standard deviation, and elapsed time. The results for TC1, TC2, and TC3 are as follows:

```
sh-5.2$ /bin/python /home/deck/Documents/School/Pruebas/A4.2/compute_statistics.py Reference\ Files/4.2.P1/P1/TC1.txt
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
Mean: 242.32
Median: 239.5
Mode: [170.0, 393.0]
Variance: 21099.917599999993
Std Dev: 145.25810683056554
Elapsed Time: 0.009261608123779297
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
sh-5.2$ /bin/python /home/deck/Documents/School/Pruebas/A4.2/compute_statistics.py Reference\ Files/4.2.P1/P1/TC2.txt
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
Mean: 250.7840161861406
Median: 247.0
Mode: [230.0]
Variance: 20785.36913247923
Std Dev: 144.17131868884059
Elapsed Time: 0.011495113372802734
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
sh-5.2$ /bin/python /home/deck/Documents/School/Pruebas/A4.2/compute_statistics.py Reference\ Files/4.2.P1/P1/TC3.txt
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
Mean: 249.77621989860583
Median: 249.0
Mode: [94.0]
Variance: 21117.27747316333
Std Dev: 145.31784980917976
Elapsed Time: 0.030217647552490234
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
```

## Ejercicio 2

### Pylint y flake8

No se muestra ningún problema dentro de VSCode:



The screenshot shows the VS Code interface with the Explorer sidebar on the left and the editor window on the right. The Explorer sidebar shows a workspace with a folder 'A4.2' containing several files, including 'convert\_numbers.py'. The editor window displays the code for 'convert\_numbers.py', which includes a list of results, a comment about outputting to terminal and file, and a function that writes results to a file. The bottom status bar indicates that no problems have been detected in the workspace.

```
138 results = [  
139     f"Elapsed Time: {elapsed_time}",  
140 ]  
141  
142  
143 # Outputting results to terminal and file  
144  
145 with open("ConversionResults.txt", "w", encoding="utf-8") as results_file:  
146     for result in results:  
147         results_file.write(result + "\n")  
148         print(result)  
149  
150     HEADER = "DEC\tBIN\tHEX"  
151     results_file.write(HEADER + "\n")  
152     print(HEADER)  
153  
154     for i, dec_number in enumerate(parsed_numbers):  
155         bin_number = bin_numbers[i]  
156         hex_number = hex_numbers[i]  
157  
158         conversion_result = f"{dec_number}\t{bin_number}\t{hex_number}\t"  
159         results_file.write(conversion_result + "\n")  
160         print(conversion_result)  
161
```

Resultados de los Test Cases:

```

sh-5.2$ /bin/python /home/deck/Documents/School/Pruebas/A4.2/convert_numbers.py Reference\ Files/4.2-1.P2/P2/TC1.txt
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
Elapsed Time: 0.0210263729095459
DEC      BIN      HEX
6980368 11010101000001100010000 6A8310
5517055 10101000010111011111111 542EFF
1336159 101000110001101011111 14635F
6750185 11001101111111111101001 66FFE9
1771937 110110000100110100001 1B09A1
360952 1011000000111111000 581F8
5672561 10101101000111001110001 568E71
916583 11011111110001100111 DFC67
2700138 1010010011001101101010 29336A
9645053 10010011001010111111101 932BFD
1181110 100100000010110110110 1205B6
1492185 101101100010011011001 16C4D9
4018595 1111010101000110100011 3D51A3
7654888 11101001100110111101000 74CDE8
7062453 11010111100001110110101 6BC3B5
2478010 1001011100111110111010 25CFBA
6134768 10111011001101111110000 5D9BF0
8420417 100000000111110001000001 807C41
2917489 1011001000010001110001 2C8471
3340773 1100101111100111100101 32F9E5
1115956 100010000011100110100 110734
9172192 100010111111010011100000 8BF4E0
6271996 10111111011001111111100 5FB3FC
8686939 100001001000110101011011 848D5B
50986 1100011100101010 C72A
9376410 100011110001001010011010 8F129A
5962327 10110101111101001010111 5AFA57
7686891 11101010100101011101011 754AEB
6615183 11001001111000010001111 64F08F
1864844 111000111010010001100 1C748C
3329962 1100101100111110101010 32CFAA
3942794 1111000010100110001010 3C298A
2614836 1001111110011000110100 27E634
7406772 11100010000010010110100 7104B4
2384190 1001000110000100111110 24613E
398347 1100001010000001011 6140B
8698503 100001001011101010000111 84BA87
9551696 100100011011111101010000 91BF50
1019556 11111000111010100100 F8EA4
1677430 110011001100001110110 199876
3479629 1101010001100001001101 35184D
9309008 100011100000101101010000 8E0B50
5266170 10100000101101011111010 505AFA
4094340 1111100111100110000100 3E7984

```

```

sh-5.2$ /bin/python /home/deck/Documents/School/Pruebas/A4.2/convert_numbers.py Reference\ Files/4.2-1.P2/P2/TC2.txt
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
Elapsed Time: 0.02134871482849121
DEC      BIN      HEX
7116776 11011001001011111101000 6C97E8
1666340 110010110110100100100 196D24
8886983 100001111001101011000111 879AC7
839365 11001100111011000101 CCEC5
924280 11100001101001111000 E1A78
1026310 11111010100100000110 FA906
1615293 110001010010110111101 18A5BD
1063875 100000011101111000011 103BC3
679035 10100101110001111011 A5C7B
5201970 10011110110000000110010 4F6032
593979 10010001000000111011 9103B
801371 11000011101001011011 C3A5B
3796878 1110011110111110001110 39EF8E
7489201 11100100100011010110001 7246B1
9740020 100101001001111011110100 949EF4
9128737 100010110100101100100001 8B4B21
5473463 10100111000010010110111 5384B7
8701957 100001001100100000000101 84C805
8238050 111110110110011111100010 7DB3E2
8679038 100001000110111001111110 846E7E
385912 1011110001101111000 5E378
5867340 10110011000011101001100 59874C
4894542 10010101010111101001110 4AAF4E
8999451 100010010101001000011011 89521B
4392535 10000110000011001010111 430657
2078131 111111011010110110011 1FB5B3
3070124 1011101101100010101100 2ED8AC
7451998 11100011011010101011110 71B55E
5635510 10101011111110110110110 55FDB6
1233932 100101101010000001100 12D40C
6089867 10111001110110010001011 5CEC8B
1792316 110110101100100111100 1B593C
6298637 11000000001110000001101 601C0D
2408038 1001001011111001100110 24BE66
8510100 100000011101101010010100 81DA94
991581 11110010000101011101 F215D
6455739 11000101000000110111011 6281BB
7829175 111011101110110110111 7776B7
6328931 11000001001001001100011 609263
9982305 100110000101000101100001 985161
2292022 1000101111100100110110 22F936
1070496 100000101010110100000 1055A0
8518360 100000011111101011011000 81FAD8

```

```

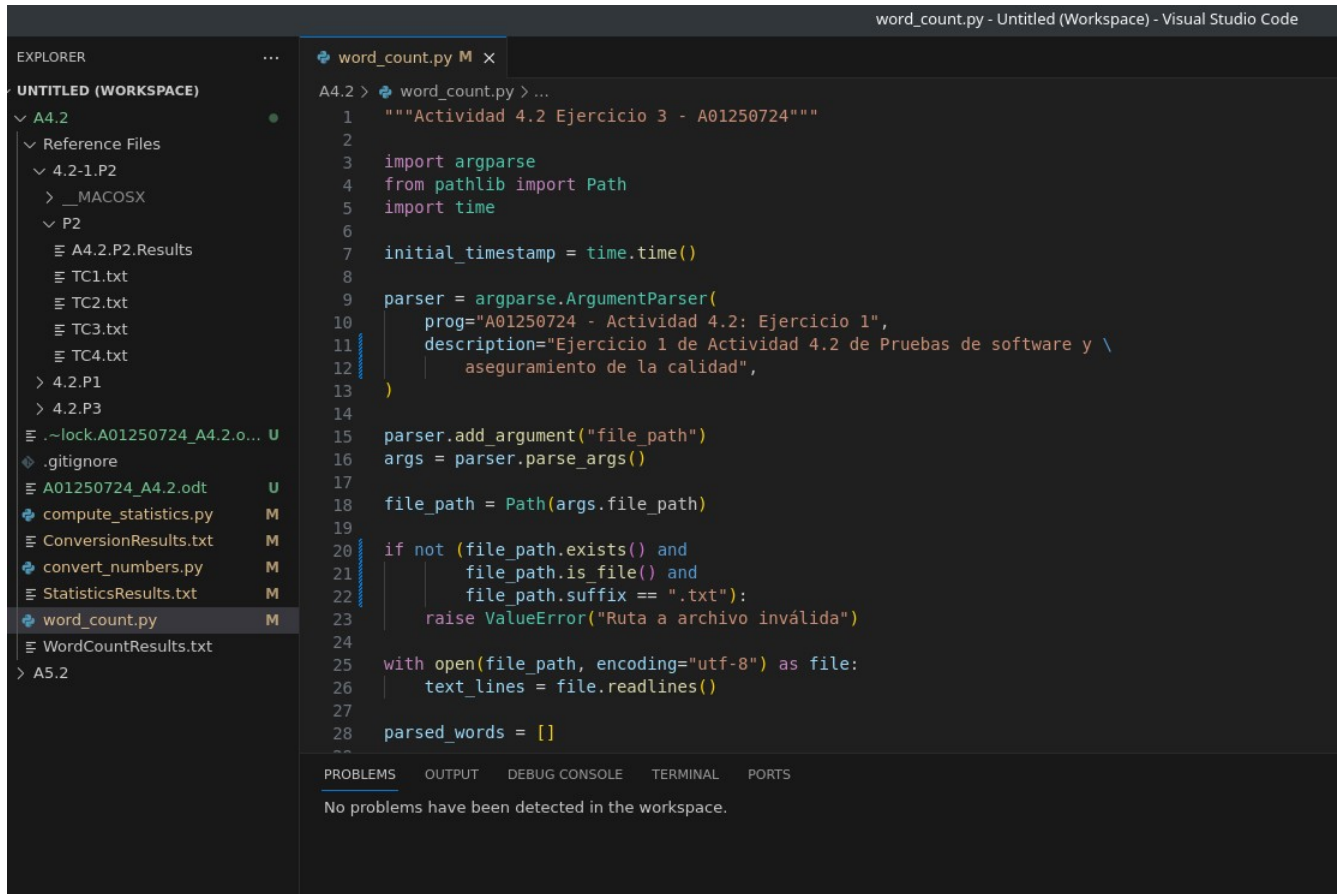
sh-5.2$ /bin/python /home/deck/Documents/School/Pruebas/A4.2/convert_numbers.py Reference\ Files/4.2-1.P2/P2/TC3.txt
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
Elapsed Time: 0.011340856552124023
DEC    BIN    HEX
-39    1111011001    FFFFFFFFD9
-36    1111011100    FFFFFFFFDC
8       1000    8
34      100010    22
17      10001    11
49      110001    31
5       101     5
39      100111    27
0       0       0
33      100001    21
12      1100     C
-6      1111111010    FFFFFFFFFA
27      11011    1B
-4      1111111100    FFFFFFFFFC
-38     1111011010    FFFFFFFFDA
26      11010    1A
49      110001    31
29      11101    1D
42      101010    2A
-16     1111110000    FFFFFFFF0
-28     1111100100    FFFFFFFFE4
34      100010    22
20      10100    14
0       0       0
25      11001    19
45      101101    2D
3       11      3
-46     1111010010    FFFFFFFFD2
-46     1111010010    FFFFFFFFD2
29      11101    1D
33      100001    21
29      11101    1D
26      11010    1A
-5      1111111011    FFFFFFFFBB
-36     1111011100    FFFFFFFFDC
12      1100     C
45      101101    2D
-50     1111001110    FFFFFFFFCE
0       0       0
-6      1111111010    FFFFFFFFFA
-39     1111011001    FFFFFFFFD9
35      100011    23
26      11010    1A
-35     1111011101    FFFFFFFFDD
-42     1111010110    FFFFFFFFD6

```

## Ejercicio 3

### Pylint y flake8

No se muestra ningún problema dentro de VSCode:



The screenshot shows the Visual Studio Code interface with a workspace named "word\_count.py - Untitled (Workspace)". The Explorer panel on the left shows a file tree with a folder "A4.2" containing several files and subfolders. The file "word\_count.py" is selected and open in the editor. The code in the editor is a Python script that uses argparse to parse command-line arguments, checks if a file exists, and reads its contents. The script is as follows:

```
A4.2 > word_count.py > ...
1  """Actividad 4.2 Ejercicio 3 - A01250724"""
2
3  import argparse
4  from pathlib import Path
5  import time
6
7  initial_timestamp = time.time()
8
9  parser = argparse.ArgumentParser(
10     prog="A01250724 - Actividad 4.2: Ejercicio 1",
11     description="Ejercicio 1 de Actividad 4.2 de Pruebas de software y \
12         |   aseguramiento de la calidad",
13 )
14
15 parser.add_argument("file_path")
16 args = parser.parse_args()
17
18 file_path = Path(args.file_path)
19
20 if not (file_path.exists() and
21         file_path.is_file() and
22         file_path.suffix == ".txt"):
23     raise ValueError("Ruta a archivo inválida")
24
25 with open(file_path, encoding="utf-8") as file:
26     text_lines = file.readlines()
27
28 parsed_words = []
```

The bottom panel of the editor shows the "PROBLEMS" tab, which displays the message: "No problems have been detected in the workspace."

Resultados de los Test Cases:

```
sh-5.2$ /bin/python /home/deck/Documents/School/Pruebas/A4.2/word_count.py Reference\ Files/4.2.P3/P3/TC3.txt
[468 preload-host-spawn-strategy] Warning: waitpid override ignores groups
Elapsed Time: 0.01584911346435547
Word      Count
neighbors      1
manual 1
political      1
mozambique     1
old            1
holding 1
fc            1
ford          1
comparable     1
industries     1
antiques       1
waste          1
voice          1
blond          1
z             1
flood          2
enables 1
feels          1
validity       1
midnight       1
matters 1
mortgage       1
daughters      1
diana          1
notice         3
postal         1
reproduced     1
mpegs          1
persistent     1
butts          1
ban            1
underground    1
'''
correct 1
```