

Instituto Tecnológico de Monterrey

4.2 Ejercicio de programación 1

Maestría en inteligencia Artificial Aplicada (MNA)

Pruebas de software y aseguramiento de la calidad
Salvador D'Carlo Abad Guarro
A01017779
2-1-2025

Repositorio en GitHub:

https://github.com/dcarloa/A01017779_PruebasDeSoftware.git

Instalación

Se instaló la librería Pylint en nuestro editor de código. (En mi caso estoy usando Visual Studio)

```
Developer PowerShell
+ Developer PowerShell | [Icon] [Icon] [Icon]
*****
** Visual Studio 2022 Developer PowerShell v17.12.4
** Copyright (c) 2022 Microsoft Corporation
*****
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3>
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3>
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3>
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3> pip install pylint
Collecting pylint
  Downloading pylint-3.3.4-py3-none-any.whl.metadata (12 kB)
Collecting dill>=0.3.6 (from pylint)
  Downloading dill-0.3.9-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: platformdirs>=2.2.0 in c:\users\d_car\appdata\local\packages\pythonsoftwarefounda
.2.0)
Collecting astroid<=3.4.0-dev0,>=3.3.8 (from pylint)
  Downloading astroid-3.3.8-py3-none-any.whl.metadata (4.5 kB)
Collecting isort!=5.13.0,<7,>=4.2.5 (from pylint)
  Downloading isort-6.0.0-py3-none-any.whl.metadata (11 kB)
Collecting mccabe<0.8,>=0.6 (from pylint)
  Downloading mccabe-0.7.0-py2.py3-none-any.whl.metadata (5.0 kB)
Collecting tomlkit>=0.10.1 (from pylint)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: colorama>=0.4.5 in c:\users\d_car\appdata\local\packages\pythonsoftwarefoundation
```

• Problema 1: Compute statistics

Instrucciones:

Req1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a list of items (presumable numbers).

Req 2. The program shall compute all descriptive statistics from a file containing numbers. The results shall be print on a screen and on a file named StatisticsResults.txt. All computation MUST be calculated using the basic algorithms, not functions or libraries. The descriptive statistics are mean, median, mode, standard deviation, and variance.

Req 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue.

Req 4. The name of the program shall be `computeStatistics.py`

Req 5. The minimum format to invoke the program shall be as follows:
`python computeStatistics.py`
`fileWithData.txt`

Req 6. The program shall manage files having from hundreds of items to thousands of items.

Req 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen.

Req 8. Be compliant with PEP8.

Al ejecutarse el código, se logran apreciar los resultados:

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\PythonApplication_EjercicioDeProgramacionAct4_3> python .\computeStatics.py .\1_TC1.txt
Advertencia: Se ignora un dato invalido -> 405s

=== Resultados Estadísticos ===
Media: 241.91228070175438
Mediana: 239.0
Moda: [393.0, 170.0]
Varianza: 21086.30558853275
Rango: 499.0
Coef. de Variacion (%): 60.02640472940299
Tiempo de Ejecucion: 0.015957 segundos
```

Una vez finalizado el código, se corrió la instrucción `pylint` en el archivo. Se obtuvieron los siguientes resultados:

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\PythonApplication_EjercicioDeProgramacionAct4_3> python -m pylint computeStatics.py
```

***** Module computeStatics

`computeStatics.py:1:0: C0114: Missing module docstring (missing-module-docstring)`

`computeStatics.py:1:0: C0103: Module name "computeStatics" doesn't conform to snake_case naming style (invalid-name)`

 Your code has been rated at 9.72/10

Tras las correcciones, el resultado fue:

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\PythonApplication_EjercicioDeProgramacionAct4_3> python -m pylint compute_statics.py
```

 Your code has been rated at 10.00/10

Your code has been rated at 10.00/10

• Problema 2: Converter

Req1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a list of items (presumable numbers).

Req 2. The program shall convert the numbers to binary and hexadecimal base. The results shall be print on a screen and on a file named ConversionResults.txt. All computation **MUST** be calculated using the basic algorithms, not functions or libraries.

Req 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue.

Req 4. The name of the program shall be `convertNumbers.py`

Req 5. The minimum format to invoke the program shall be as follows:
`python convertNumbers.py fileWithData.txt`

Req 6. The program shall manage files having from hundreds of items to thousands of items.

Req 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen.

Req 8. Be compliant with PEP8.

Se obtienen los siguientes resultados al ejecutarse el código:

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\Converter_numbers> python .\converter_numbers.py .\2_TC2.txt
```

```
=== Resultados de Conversion ===
Decimal: 7116776, Binario: 11011001001011111101000, Hexadecimal: 6C97E8
Decimal: 1666340, Binario: 110010110110100100100, Hexadecimal: 196D24
Decimal: 8886983, Binario: 100001111001101011000111, Hexadecimal: 879AC7
Decimal: 839365, Binario: 11001100111011000101, Hexadecimal: CCEC5
Decimal: 924280, Binario: 11100001101001111000, Hexadecimal: E1A78
Decimal: 1026310, Binario: 11111010100100000110, Hexadecimal: FA906
Decimal: 1615293, Binario: 110001010010110111101, Hexadecimal: 18A5BD
Decimal: 1063875, Binario: 10000001101111000011, Hexadecimal: 103BC3
Decimal: 679035, Binario: 10100101110001111011, Hexadecimal: A5C7B
Decimal: 5201970, Binario: 10011110110000000110010, Hexadecimal: 4F6032
Decimal: 593979, Binario: 10010001000000111011, Hexadecimal: 9103B
Decimal: 801371, Binario: 11000011101001011011, Hexadecimal: C3A5B
Decimal: 3796878, Binario: 111001111011110001110, Hexadecimal: 39EF8E
Decimal: 7489201, Binario: 11100100100011010110001, Hexadecimal: 7246B1
Decimal: 9740020, Binario: 100101001001111011110100, Hexadecimal: 949EF4
Decimal: 9128737, Binario: 100010110100101100100001, Hexadecimal: 8B4B21
Decimal: 5473463, Binario: 10100111000010010110111, Hexadecimal: 5384B7
Decimal: 8701957, Binario: 100001001100100000000101, Hexadecimal: 84C805
Decimal: 8238050, Binario: 11111011011001111100010, Hexadecimal: 7DB3E2
Decimal: 8679038, Binario: 100001000110111001111110, Hexadecimal: 846E7E
Decimal: 385912, Binario: 1011110001101111000, Hexadecimal: 5E378
Decimal: 5867340, Binario: 10110011000011101001100, Hexadecimal: 59874C
Decimal: 4894542, Binario: 10010101010111101001110, Hexadecimal: 4AAF4E
Decimal: 8999451, Binario: 100010010101001000011011, Hexadecimal: 89521B
Decimal: 4392535, Binario: 10000110000011001010111, Hexadecimal: 430657
Decimal: 2078131, Binario: 111111011010110110011, Hexadecimal: 1FB5B3
Decimal: 3070124, Binario: 1011101101100010101100, Hexadecimal: 2ED8AC
Decimal: 7451998, Binario: 1110001011010101011110, Hexadecimal: 71B55E
Decimal: 5635510, Binario: 1010101111110110110110, Hexadecimal: 5FDB06
Decimal: 1233932, Binario: 100101101010000001100, Hexadecimal: 12D40C
Decimal: 6089867, Binario: 10111001110110010001011, Hexadecimal: 5CEC8B
Decimal: 1792316, Binario: 110110101100100111100, Hexadecimal: 1B593C
Decimal: 6298637, Binario: 11000000001110000001101, Hexadecimal: 601C0D
Decimal: 2408038, Binario: 1001001011111001100110, Hexadecimal: 24BE66
Decimal: 8510100, Binario: 100000011101101010010100, Hexadecimal: 81DA94
```

Al correr la instrucción pylint en el archivo, inicialmente se obtuvieron los siguientes resultados:

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\Converter_numbers> python -m pylint Converter_numbers.py
```

```
***** Module Converter_numbers
```

```
Converter_numbers.py:90:0: C0305: Trailing newlines (trailing-newlines)
```

```
Converter_numbers.py:1:0: C0103: Module name "Converter_numbers" doesn't conform to snake_case naming style (invalid-name)
```

```
-----
```

Your code has been rated at 9.67/10

Se entendieron las recomendaciones y se procedió a realizar los ajustes correspondientes:

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\Converter_numbers> python -m pylint converter_numbers.py
```

```
-----
Your code has been rated at 10.00/10 (previous run: 9.83/10, +0.17)
```

• **Problema 3: Count Words**

Req1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a words (presumable between spaces).

Req 2. The program shall identify all distinct words and the frequency of them (how many times the word "X" appears in the file). The results shall be print on a screen and on a file named WordCountResults.txt.

All computation MUST be calculated using the basic algorithms, not functions or libraries.

Req 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue.

Req 4. The name of the program shall be wordCount.py

Req 5. The minimum format to invoke the program shall be as follows:

python wordCount.py fileWithData.txt

Req 6. The program shall manage files having from hundreds of items to thousands of items.

Req 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen.

Req 8. Be compliant with PEP8.

Tras la ejecución del código, los resultados fueron los siguientes:

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\words_count> python .\words_count.py .\3_TC1.txt

=== Resultados del Conteo de Palabras ===
achievement: 1
adequate: 1
adventures: 1
anal: 1
andrews: 1
assessed: 1
bedding: 1
blues: 1
buying: 1
cartridge: 1
cgi: 1
championship: 1
clear: 1
club: 1
coastal: 1
collect: 1
comm: 1
confirm: 1
conservative: 2
consistent: 1
contamination: 1
could: 1
cove: 1
craps: 1
customized: 1
danish: 1
dial: 1
drum: 1
ebony: 1
enhanced: 1
exhaust: 1
explorer: 1
father: 1
```

Luego se aplicó pylint al código para validar el correcto estructuramiento:

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\words_count> python -m pylint .\words_count.py
```

***** Module words_count

words_count.py:87:0: C0305: Trailing newlines (trailing-newlines)

Your code has been rated at 9.82/10

Al no obtener el 100%, entonces se procedió a hacer los ajustes especificados.

```
PS C:\Users\d_car\source\repos\PythonApplication_EjercicioDeProgramacionAct4_3\words_count> python -m pylint .\words_count.py
```

Your code has been rated at 10.00/10 (previous run: 9.82/10, +0.18)

Obras Citadas

- Van Rossum, G., Warsaw, B. & Coghlan, Alyssa (2013). PEP 8 – Style Guide for Python Cod. Convención de codificación de Python - PEP8
<https://peps.python.org/pep-0008/>