

Materia: Pruebas de software y aseguramiento de la calidad

Maestro Titular: Dr. Gerardo Padilla Zárate

Maestro Asistente: Yetnalezi Quintas Ruiz

Matricula: A0173101

Alumno: Guillermo Alfonso Muñiz Hermosillo

E-mail: A01793101@tec.mx

5.2 Ejercicio de programación 2

Repositorio de GitHub: https://github.com/A01793101-GMuniz/A01793101_PruebasDeSoftware/tree/main/A01793101_A5.2

Problema 1: Compute sales

Nota: Para poder cumplir con el estándar de PEP-8 se ha cambiado el nombre del archivo para cumplir con el camel case a `compute_sales.py`

En este ejercicio de programación se plantea la creación de un programa que sea capaz de calcular el costo total de las ventas incluidas en un par de archivos de JSON. El primero cuenta con información del catalogo de productos y la información de dichos productos tales como nombre, descripción, especificaciones y valoración. En el segundo archivo JSON se encontrará la información de ventas, en cada uno de los casos de prueba se cuenta con el identificador de la venta, la fecha de venta, el producto y la cantidad vendida.

El objetivo será imprimir en pantalla el total de la venta realizada en cada prueba case, relacionando los productos del catalogo y la cantidad adquirida. Es decir, del catálogo necesitamos obtener el precio del producto para relacionarlo con la venta realizada y sumar al total de las ventas considerando el precio y la cantidad vendida.

Codigo Fuente Final:

```
1  """
2      Program to compute the total cost for all sales
3      included in the JSON archives. The results shall
4      be print on a screen and on a file named SalesResults.txt.
5
6      The total cost should include all items in the sale considering
7      the cost for every item.
8  """
9  import sys
10 import os
11 import re
12 import time
13 import json
14 import pandas as pd
15 from tabulate import tabulate
16
17
18 def create_results_file(r_file_path, test_case_key):
19     """Funcion para crear un archivo de resultados
20     con cada ejecucion"""
21     if os.path.isfile(r_file_path):
22         with open(r_file_path, 'w', encoding="UTF-8") as result_file:
23             result_file.close()
24     with open(r_file_path, 'w', encoding="UTF-8") as result_file:
25         # Crear el archivo con headers
26         result_file.write(f"{test_case_key}\n")
27
28
29 def main():
30     """Main program function definition."""
31     start_time = time.time()
32     if not os.path.isfile(sys.argv[1]):
33         raise Exception(f"{sys.argv[1]} is not a file")
34     with open(sys.argv[1], 'r', encoding="UTF-8") as archivo:
35         cat_df = json.load(archivo)
36     cat_df = pd.DataFrame(cat_df)
37     cat_df = cat_df.rename(columns={"title": "product"})
38
39     if not os.path.isfile(sys.argv[2]):
40         raise Exception(f"{sys.argv[2]} is not a file")
41     with open(sys.argv[2], 'r', encoding="UTF-8") as archivo:
42         sales_df = json.load(archivo)
43     sales_df = pd.DataFrame(sales_df)
44     sales_df = sales_df.rename(columns={"Product": "product",
45                                         "SALE Date": "sale_date",
46                                         "Quantity": "quantity"})
```

El programa utiliza bibliotecas como `sys`, `os`, `re`, `time`, `json`, `tabulate` y `pandas` para el tratamiento de la información leída y la que se va a mostrar/escribir en los resultados finales.

La función `create_results_file()` se encarga de crear el archivo de resultados con un encabezado, validando que si existe algún archivo con el mismo nombre autogenerado este sea sobrescrito para tener los resultados más actualizados. Cabe mencionar que el nombre del archivo es tomado directamente del número de TC que se encuentra en el nombre de los directorios.

[illegible]

```
93 if __name__ == "__main__":
94     main()
```

```
Administrador: C:\Windows\system32\cmd.exe

(base) C:\Users\gmuni>pylint "C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py"
***** Module compute_sales
OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:17:50: C0303: Trailing whitespace (trailing-whitespace)

-----
Your code has been rated at 9.78/10 (previous run: 7.35/10, +2.43)

(base) C:\Users\gmuni>
```

Ejecución de pylint sin errores:

```
(base) C:\Users\gmuni>pylint "C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py"

-----
Your code has been rated at 10.00/10 (previous run: 9.78/10, +0.22)

(base) C:\Users\gmuni>
```

Primera ejecución con flake8

```
Administrador: C:\Windows\system32\cmd.exe

(base) C:\Users\gmuni>flake8 "C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py"
C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:16:1: E302 expected 2 blank lines, found 1
C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:26:1: E302 expected 2 blank lines, found 1
C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:38:45: E127 continuation line over-indented for visual indent
C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:43:80: E501 line too long (94 > 79 characters)
C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:48:80: E501 line too long (95 > 79 characters)
C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:49:38: E225 missing whitespace around operator
C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:69:80: E501 line too long (89 > 79 characters)
C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py:71:1: E305 expected 2 blank lines after class or function definition, found 1

(base) C:\Users\gmuni>
```

Ejecucion de flake8 sin errores

```
(base) C:\Users\gmuni>flake8 "C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py"

(base) C:\Users\gmuni>
```

Ejecucion de pylint y flake sin errores

```
(base) C:\Users\gmuni>flake8 "C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py"

(base) C:\Users\gmuni>pylint "C:\Users\gmuni\OneDrive\Documentos\TEC_Maestria\Pruebas de Software\A01793101\A01793101_PruebasDeSoftware\A01793101_A5.2\compute_sales.py"

-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

(base) C:\Users\gmuni>
```