



Buenas prácticas de programación en Python

Lección 1: Control de errores, pruebas y validación de datos

Control de errores, pruebas y validación de datos

Actividad relacionada con la lección 1:

Considere el fichero de datos en formato csv *finanzas2020* en el que ha ido anotando sus ingresos y gastos mensuales durante el año 2020.

APARTADO 1:

Implemente un programa que lea el contenido del fichero y realice los siguientes cálculos:

- ¿Qué mes se ha gastado más?
- ¿Qué mes se ha ahorrado más?
- ¿Cuál es la media de gastos al año?
- ¿Cuál ha sido el gasto total a lo largo del año?
- ¿Cuáles han sido los ingresos totales a lo largo del año?
- *Opcional:* Realice una gráfica de la evolución de ingresos a lo largo del año .

APARTADO 2:

Haciendo uso de excepciones, haga las siguientes comprobaciones:

- Compruebe que el fichero existe y que tiene 12 columnas, una para cada mes del año.
- Para cada mes compruebe que hay contenido.
- Compruebe que todos los datos son correctos. De no haber un dato correcto, el programa debe saber actuar en consecuencia y continuar con su ejecución.

Aplique los conocimientos adquiridos en esta lección sobre programación defensiva para realizar las comprobaciones que crea oportunas.

Condiciones de entrega:

- Entregue un documento en formato PDF en el que responda a cada una de las preguntas planteadas en el primer ejercicio. Además, describa cómo ha abordado el problema, tanto para la funcionalidad del programa como el uso de excepciones y validaciones.
- Entregue el código fuente.
- Si ha realizado el ejercicio opcional, exporte el gráfico a una imagen y añádela a la entrega.

CODIGO:

```

main.py > ...
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # define Python user-defined exceptions
5  class Error(Exception):
6      """Base class exceptions"""
7      pass
8  class DatoError(Error):
9      "Los datos tienen formato erroneo"
10     pass
11  class ColumnasMesError(Error):
12      "Las columnas no son correctas"
13     pass
14  class NumColumnasMesError(Error):
15      "El numero de columnas es erroneo"
16     pass
17  class ColumnasVaciasError(Error):
18      "Las columnas estan vacías"
19     pass
20
21  try:
22      df = pd.read_csv('finanzas2020.csv', sep = '\t')
23
24      if(len(df.columns) != 12):
25          raise NumColumnasMesError
26
27      if(list(df.columns) != ['Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio',
28                             'Agosto','Septiembre','Octubre','Noviembre','Diciembre']):
29          raise ColumnasMesError
30
31
32      for columna in df.columns:
33          if(df[columna].empty):
34              raise ColumnasVaciasError(columna)
35          else:
36              df[columna] = df[columna].astype(str)
37              for i in df[columna].index:
38                  if(type(df[columna][i]) is int):
39                      break
40                  else:
41                      df[columna][i] = df[columna][i].replace("\'",'')
42                      try:
43                          a = int(df[columna][i])
44                          #print("tipo de dato:",type(df[columna][i]),"columna: ", columna, " indice: ", i)
45                      except ValueError as de:
46                          df[columna][i] = 0
47                          # si el dato es erroneo lo ponemos a 0 ni suma ni resta
48                          # print("Error en dato:",df[columna][i],"columna: ", columna, "indice:",i, "columna: ", columna)
49                          continue
50              df[columna].astype(int)
51  except IOError as e:
52      print("Error - archivo no encontrado",e)
53  except ColumnasMesError as cme:
54      print("Headers incorrectos")
55  except NumColumnasMesError as ncme:
56      print("Numero Columnas incorrectos")
57  except ColumnasVaciasError as cve:
58      print("Hay una o mas Vacías ")
59  else:
60      #cambiamos valores a int

```

```

58 else:
59     #cambiamos valores a int
60     df = pd.DataFrame(df).astype(int)
61     df_sum_column = df.sum(axis=0)
62     # Mes mayor gasto (min)
63     print("# Mes mayor gasto (min)")
64     mes_mayor_gasto = df_sum_column.min()
65     print(mes_mayor_gasto)
66
67     # Mes mayor ahorro(max)
68     print("# Mes mayor ahorro(max)")
69     mes_mayor_ahorro = df_sum_column.max()
70     print(mes_mayor_ahorro)
71
72     # Media de gasto del año (avg)
73     print("# Media de gasto del año (avg)")
74     media_gasto_anno = df_sum_column.mean()
75     print(media_gasto_anno)
76
77     # ingresos totales (Mes a mes y total)
78     print("# ingresos totales (Mes a mes)")
79     print (df_sum_column)
80     df_total = df_sum_column.sum(axis=0)
81     print("# ingresos totales(Total)")
82     print (df_total)
83
84     # Opcional: Realice una gráfica de la evolución de ingresos a lo largo del año
85     df2 = pd.DataFrame({"Mes":list(df.columns),
86                         "Ahorro":[
87                             df['Enero'].sum(axis=0),
88                             df['Febrero'].sum(axis=0),
89                             df['Marzo'].sum(axis=0),
90                             df['Abril'].sum(axis=0),
91                             df['Mayo'].sum(axis=0),
92                             df['Junio'].sum(axis=0),
93                             df['Julio'].sum(axis=0),
94                             df['Agosto'].sum(axis=0),
95                             df['Septiembre'].sum(axis=0),
96                             df['Octubre'].sum(axis=0),
97                             df['Noviembre'].sum(axis=0),
98                             df['Diciembre'].sum(axis=0)]})
99     print(df2)
100
101     df2.plot.bar(x='Mes', y='Ahorro', rot=0)
102     plt.show()
103
104

```

```

# Mes mayor gasto (min)
# Mes mayor gasto (min) -18933 Abril
# Mes mayor ahorro(max)
# Mes mayor ahorro(max) Enero 11523
# Media de gasto del año (avg)
# Media de gasto del año (avg) -1414.4166666666667
# ingresos totales (Mes a mes y total)
# ingresos totales (Mes a mes)
Enero      11523
Febrero     39
Marzo      -7969
Abril      -18933
Mayo       10304
Junio      -1477
Julio       7698
Agosto     -8735
Septiembre -10948
Octubre     3412
Noviembre  1157
Diciembre  -3044

# ingresos totales(Total)
-16973
# Opcional: Realice una gráfica de la evolución de ingresos a lo largo de
l año

```

