“AÑO DEL FORTALECIMIENTO DE LA SOBERANÍA NACIONAL”

INTEGRANTES:

- DOMINGUEZ BACA, ZULMA TATIANA
- JUÁREZ SILVA, MARYCIELO

PROFESOR:

- PEDRO ROTTA SAAVEDRA

CURSO:

- ANÁLISIS DE DATOS CON PYTHON (NIVEL I – FUNDAMENTOS)

NOMBRE DE LA ENTREGA:

- PROGRAMA DE SISTEMATIZACIÓN (TRABAJO 1)

2022

1. INTRODUCCIÓN

Python es un lenguaje de programación multiplataforma y de código abierto que puede utilizarse tanto para desarrollo web, creación de software y procesamiento de datos. La facilidad y sencillez en su aprendizaje lo han convertido en el lenguaje de programación más popular del mundo.

Particularmente, el presente informe busca detallar el desarrollo de un código enfocado al procesamiento de bases de datos con determinadas características y empleando los comandos estudiados durante las horas de clase. Además, precisar las librerías utilizadas para la optimización del código.

Por último, se presentarán en los párrafos posteriores las conclusiones a las que han llegado las integrantes tras haber expuesto sus conocimientos en la creación y desarrollo del código, como parte de la tarea asignada.

2. ANÁLISIS DEL SISTEMA

El trabajo se encuentra en el repositorio GitHub, enlace: https://github.com/marycielojs/trabajo-1.git

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
```

Para iniciar, se empleó el comando **"import"** seguido del nombre de las librerías que se deseaban importar sin la terminación .py. Esto permite hacer uso de todo el código que tal módulo o librería contenga.

El comando **"as"** para pedirle a los agentes en el mundo que realicen acciones de forma selectiva, ya sea por el tipo del agente o por propiedades intrínsecas a los mismos.

```
path="BASE DE DATOS.xlsx"
BD=pd.read_excel(path)
```

Ahora, **"path"** como una lista de nombres de directorios que constituye la ruta de búsqueda actual, seguido del nombre del archivo Excel que contenga la base de datos para que el código pueda extraer la información con la que trabajará**.**

```
col_val=[]
columnas=[]
for i in BD:
  columnas.append(i)
  col_val.append(list(BD[i]))
```

Como se trata de datos, se realizan actos repetitivos. Por tanto, se utiliza el comando o bucle **"for"**, cuyo propósito es ordenar que se apliquen los comandos para todos los elementos de la colección y se repita una o más instrucciones un determinado número de veces.

Por su parte, el comando **"list",** el cual es un tipo de secuencia o contenedor compuesto que se emplea para almacenar conjuntos de elementos relacionados ya sea del mismo tipo o de tipos diferentes.

```
while True:
  print("1 Mostrar base de datos")
  print("2 Insertar valor")
  print("3 Generar estadísticas")
  print("4 Generar gráfico")
  print("5 Insertar característica")
  print("6 Salir")
  print()
  opc=input("Selecciona una opción: ")
  print()
```

El bucle **"while"** es una instrucción de control de flujo que permite ejecutar una serie de comandos repetidamente sobre la base de una condición dada. En este caso, se propone "mientas sea cierto o se cumpla tal condición" se cumplirán los comandos dentro.

Además, se utiliza el comando **"print"** que sirve para introducir cadenas de texto que le aparecerán al usuario. Por su lado, el comando "input" permite que el usuario introduzca una información, por ejemplo, el número correspondiente a la opción que desea (2 = insertar valor).

```
if opc=="1":
  if len(BD)==0:
    print("No hay datos")
    print()
  else:
    print(BD)
    print()
```

La pseudocodificación es crucial en este punto, se prevén todas las posibilidades que tiene el usuario para la mayor optimalización del código.

Se muestra el comando **"if"** cuya función es comprobar si se cumple una condición lógica, para ello se coloca dos veces el signo igual **"=="**, se asigna una comparación de igualdad.

En este caso se evidencia la indexación, es decir, condiciones dentro de otra condición ("**if nested**"). Así, se coloca dentro el comando "else", el cual permite que un programa ejecute unas instrucciones cuando se cumple una condición y otras instrucciones cuando no se cumple tal condición.

```
#print(col_val[ind_col])
j=True
for i in range(len(col_val[ind_col])):
  if pd.isna(col_val[ind_col][i]):
    col_val[ind_col][i]=val[val.index(",")+1:len(val)]
    j=False
    break
```

Se utilizan los **numerales (##**) para introducir comentarios sobre el desarrollo del código. Asimismo, el comando **"break"** para detener la ejecución de un bucle y salirse de él; y el comando **"range"** que es una lista inmutable de números enteros en sucesión aritmética.

```
elif opc=="2":
  print("CARAPTERISTICAS")
  if len(columnas)==0:
    print("No hay caracteristicas")
  else:
    for i in range(1,len(columnas)):
      if columnas[i]!="Unnamed: 0":
        print("",i,columnas[i])
    print()
```

El método **"len"** que devuelve la longitud de la lista de elementos o valores. También el comando **"elif"**, el cual permite la ejecución de condiciones de forma jerárquica, es decir, si no se cumple la primera condición se evalúa la siguiente condición y así sucesivamente.

```
val=int(input("Ingresa n° de característica: "))
print()
print(columnas[val])
if isinstance(col_val[val][0],float) or isinstance(col_val[val][0],int):
  print(" ","Total:  ",round(sum(col_val[val]),2))
  print(" ","Máximo: ",max(col_val[val]))
  print(" ","Mínimo: ",min(col_val[val]))
  print(" ","Promedio:",round(sum(col_val[val])/len(col_val[val]),2))
```

Inclusive, el comando **"int"** que precede a input es para indicar que tal entrada es un número entero, puesto que input solo soporta cadenas de texto.

Ahora bien, el comando **“isinstence”** se utilizó para que el código determine si un valor ubicado en determinada columna contiene valores numéricos (ej. precio) o valores descriptivos (ej. operadores)

Y el comando **“float”** que interpreta la entrada como un numero real con parte decimal. Esto se relaciona con la función **“round”,** la cual redondea el decimal al número dado de dígitos (en este caso es 2) y devuelve el número de punto flotante.

3. EJEMPLOS AL CORRER EL SISTEMA

Este ejemplo es acerca de una base de datos que contiene información sobre una determinada cantidad de usuarios: sus nombres (columna 1), el operador que utilizan (columna 2), el precio que pagaron por el servicio del operador (columna 3).

Se debe subir el archivo y colocar el nombre del documento Excel sin modificaciones dentro del código para que el sistema tome la base de datos (con la terminación .xlsx), tal como se muestra:

```
path="BASE DE DATOS.xlsx"
BD=pd.read_excel(path)
```

Luego, se ejecuta el código y aparecen las siguientes opciones de acciones que puede realizar el programa con un número correspondiente, seguido del texto “Seleccione una opción”:

```
1 Mostrar base de datos
2 Insertar valor
3 Generar estadísticas
4 Generar gráfico
5 Insertar característica
6 Salir

Selecciona una opción: 1
```

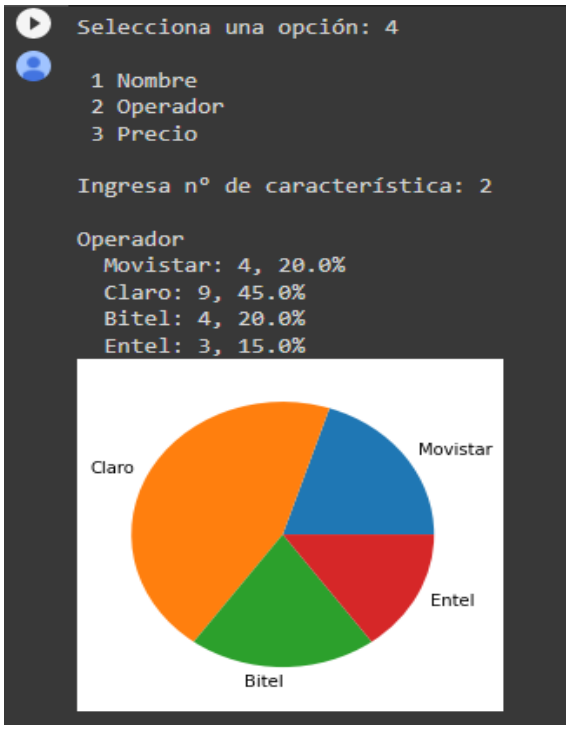
El usuario tiene la opción de ingresar el número que desee según la opción (1-6). En este caso se colocó el número 1 para mostrar la base de datos:

```
Selecciona una opción: 1

    Unnamed: 0     Nombre  Operador  Precio
0            0       Hugo  Movistar   150.2
1            1     Martín     Claro   151.2
2            2      Lucas     Claro   152.2
3            3      Mateo     Bitel   153.9
4            4        Leo     Bitel   154.2
5            5     Daniel     Bitel   255.2
6            6  Alejandro     Claro   156.2
7            7      Pablo     Claro   157.2
8            8     Manuel  Movistar   158.2
9            9     Álvaro  Movistar   259.2
10          10     Adrián     Claro   160.2
11          11      David     Claro   141.2
12          12      Mario     Entel   162.8
13          13       Enzo     Entel   163.2
14          14      Diego     Entel   164.2
15          15     Marcos     Claro   145.2
16          16       Izan     Claro   166.2
17          17     Javier     Bitel   167.2
```

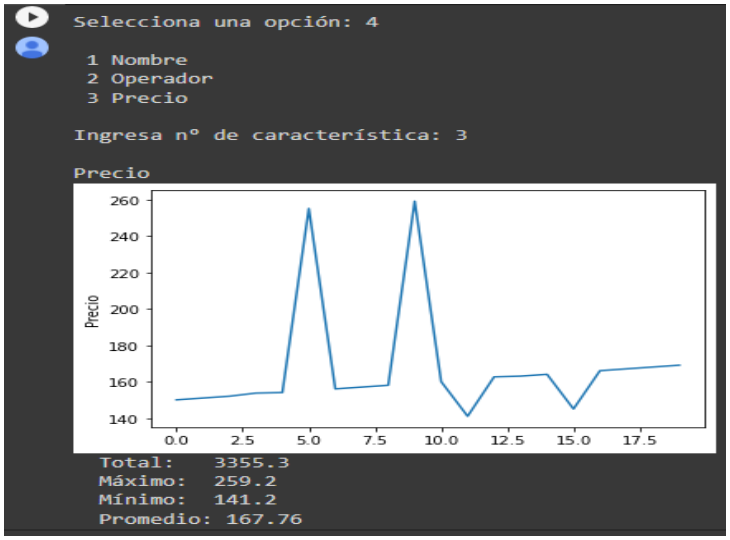
Después, se vuelve a mostrar el texto “Seleccione una opción”. Se colocó el número 3 que corresponde a “Generar estadísticas”. Por tanto, aparecen las 3 características u columnas en el archivo Excel (nombre 1, operador 2 o precio 3) y se debe escoger el número. En esta ocasión se introdujo el número 2 relativo al operador, el cual tiene valores descriptivos. El programa brinda estadísticas en porcentaje de cuantas personas son usuarias de estas operadoras respecto del total:

```
Selecciona una opción: 3

 1 Nombre
 2 Operador
 3 Precio

Ingresa nº de característica: 2

Operador
  Movistar: 4, 20.0%
  Claro: 9, 45.0%
  Bitel: 4, 20.0%
  Entel: 3, 15.0%
```

Seguido de ello, se vuelve a mostrar la cadena de texto “Seleccione una opción”, colocamos la número 4 “Generar gráfico”. Se escoge la característica número 2 operador, se vuelve a mostrar los porcentajes seguido del gráfico circular:

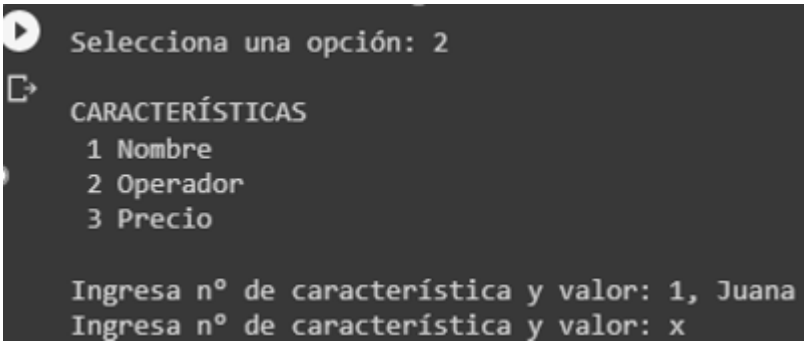


Se muestra la opción de introducir una nueva acción, colocamos la numero 4 “Generar un gráfico”, pero esta vez respecto de la característica número 3 (precio), la cual tiene valores numéricos. Por tanto, el grafico que se muestra es uno lineal, seguido del valor total (suma) de todos los datos numéricos, el valor máximo y mínimo encontrado, y el promedio de ellos (media):

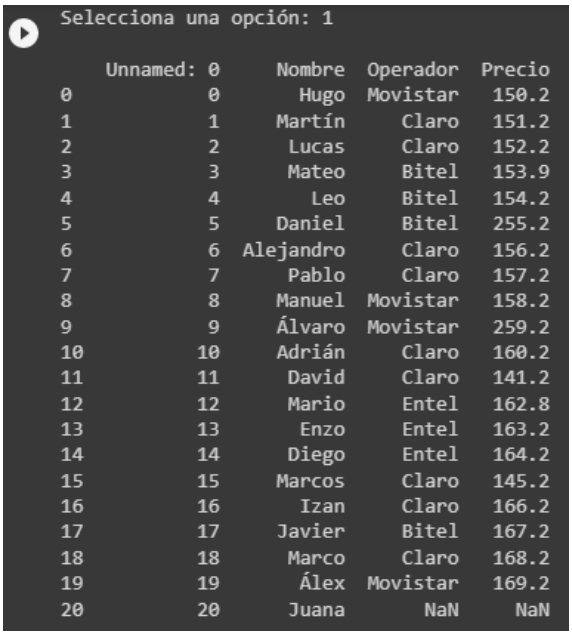


En la siguiente opción introducimos la acción número 2 correspondiente a “Insertar valor” para que se actualice la base de datos con lo que le agreguemos. Así, aparecen las características o columnas y el texto “Ingresa n° de característica y valor” con la opción de que el usuario coloque tal información. Aquí introducimos el número 1 “nombre” seguido de una coma con la palabra “Juana”.

Es un bucle por lo que continuará apareciendo el mismo texto, para hacer uso de la función break y ponerle fin a las repeticiones colocamos la letra “x”:

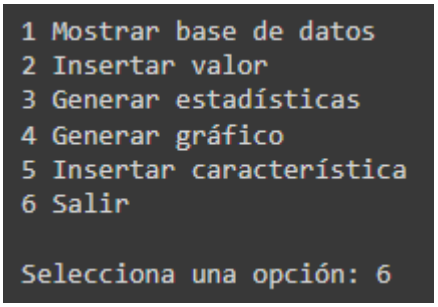


Para comprobar si es que se actualizó colocamos la opción 1 “Mostrar base de datos”. En efecto, aparece el nombre Juana al final de la columna “Nombre”, su información relativa a operador y precio se rellena con “NaN” por defecto ya que no se introdujo valores para ello:



	Unnamed: 0	Nombre	Operador	Precio
0	0	Hugo	Movistar	150.2
1	1	Martín	Claro	151.2
2	2	Lucas	Claro	152.2
3	3	Mateo	Bitel	153.9
4	4	Leo	Bitel	154.2
5	5	Daniel	Bitel	255.2
6	6	Alejandro	Claro	156.2
7	7	Pablo	Claro	157.2
8	8	Manuel	Movistar	158.2
9	9	Álvaro	Movistar	259.2
10	10	Adrián	Claro	160.2
11	11	David	Claro	141.2
12	12	Mario	Entel	162.8
13	13	Enzo	Entel	163.2
14	14	Diego	Entel	164.2
15	15	Marcos	Claro	145.2
16	16	Izan	Claro	166.2
17	17	Javier	Bitel	167.2
18	18	Marco	Claro	168.2
19	19	Álex	Movistar	169.2
20	20	Juana	NaN	NaN

Por último, colocamos la acción 6 “Salir” para dar por finalizado el programa. Se emplea la función break para finalizar con el bucle:



4. CONCLUSIONES

A través de la realización del presente trabajo en grupo hemos llegado a las siguientes conclusiones respecto de lo aprendido en lenguaje Python:

-Al ser un lenguaje de código abierto, Python es libre de usar y cualquiera puede modificar o crear extensiones para este lenguaje.

- Su versatilidad nos indica que, como desarrollador, se tiene una amplia gama de opciones de trabajo.

- Su automatización es otra área por la que vale la pena aprender este lenguaje Python. Su capacidad para escribir scripts de sistema origina que se puedan crear programas Python sencillos para automatizar tareas monótonas que disminuyen tu productividad.

Como conclusión general de este aprendizaje podemos decir que aprender el código Python lo que hace es prepararnos para el desarrollo de Internet, en especial para el futuro de los trabajos tecnológicos, porque se utiliza para algo más que el desarrollo tradicional. Los lenguajes de programación se adaptan a las necesidades de las personas y, en especial de las que quieren mantenerse en una carrera tecnológica sin precedentes.