



UNIVERSIDAD
DE PIURA

UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERÍA



Análisis de datos con Python Nivel 1

Trabajo Final

Grupo: 6

Integrantes:

- Gambini Gamboa, Ricardo Jesús
- Ipanaqué Hau Yon, Álvaro Daniel
- Piedra Almaster, Carla Franchesca
- Ramírez Vera, Ariana Elizabeth
- Saldarriaga Valencia, Hugo Franco
- Tipe Verástegui, Christian Andrés

INDICE

Introducción	3
Análisis de Sistema:	3
Ejemplo:.....	8
• Para añadir una fila:	8
• Para añadir una columna:	8
• Para filtrar:	9
• Para ordenar:	9
• Para Datos Estadísticos:	9
• Para realizar gráficos:	9
1. Histograma:	9
2. Dispersión:	10
• Para guardar:	11
Conclusiones:	12

Introducción

Las empresas comercializadoras de bienes o servicios están en busca de obtener utilidades positivas. Suponiendo que las ventas permanezcan a un nivel constante, este tipo de empresas solo podrán aumentar su utilidad si aumentan los precios de sus productos, reducen sus costos, o realizan ambas cosas. Es por ello que es importante gestionar el flujo de las ventas realizadas y tenerlas digitalizadas para poder realizar un análisis de la situación de la empresa.

Teniendo en cuenta todo lo anterior, este programa se ha realizado con el fin de solucionar las necesidades administrativas de una empresa genérica dedicada a la compra y venta de bienes o servicios y que desea registrar las transacciones u operaciones realizadas en el día para así poder realizar un análisis de ello.

Análisis de Sistema:

La empresa a través del sistema le preguntara al usuario que actividad es la que desea realizar, como añadir un nuevo cliente (S), crear una nueva característica(C), buscar datos de acuerdo a lo que el usuario ha ingresado (F), ordenar los datos de una columna específica (O), mostrar datos estadísticos (E) y finalmente dejar que cierre el programa y guardarlo en un archivo Excel (Presionando cualquier otra tecla).

```
import pandas as pd
import matplotlib.pyplot as plt

dicc={"RUC":[], "Cliente":[], "Operador":[], "Descripción":[], "Monto":[], "Fecha":[]}
df=pd.DataFrame(dicc)
def usuario_decision():
    decision=input("Escriba S para añadir Clientes, C para añadir nuevas Columna
    if decision=="S":
        estado=1
    elif decision=="C":
        estado=2
    elif decision=="F":
        estado=3
    elif decision=="O":
        estado=4
    elif decision=="D":
        estado=5
    elif decision=="H":
        estado=6
    else:
        estado=7
    return estado
```

Para ello se importó la librería **Pandas**, y **Matplotlib**, además se creó un diccionario que posteriormente pasará a convertirse en DataFrame, el cual se utilizará en todo el código, además se hizo uso del comando **def**, para crear la función **usuario_decision** y el comando **if**, **elif** y **else** para condicionar.

También se creó una función **def** para cada opción mencionada:

- Para la función **Añadir_fila**, se usó el siguiente código:

```
def añadir_fila():
    global df
    dicc_fila={}
    columnas_df=df.columns
    for columna in columnas_df:
        dicc_fila[columna]=input("Ingrese valor de la columna {} ".format(columna))
    df=df.append(dicc_fila,ignore_index=True)
```

Como bien sabemos el sistema va a tener que ir agregando nuevas filas, por ello se crea un diccionario **dicc_fila** y se utiliza el comando **columns**, para guardar el nombre de cada columna en la variable **columnas**, asimismo para guardar el dato en cada columna se utilizó el bucle **for** y una vez ingresada el nombre de la columna se guardará en el diccionario creado con el comando **append**.

- Para la función **Añadir_columna**, se usó el siguiente código:

```
def añadir_columna():
    global df
    name=input("Ingrese el nombre de la columna nueva: ")
    a=[]
    for i in range (int(len(df.index))):
        s=input("Escriba el valor del cliente NÂ°{}".format(i))
        a.append(s)
    df=df.assign(n=a)
    df.rename(columns = {"n":name},inplace=True)
```

El usuario ingresa el nombre de la nueva columna, y se almacenará en la variable **name**, además se crea una lista vacía llamada **a**, luego el usuario ingresa el valor del cliente dependiendo del índice de la fila que está representado por la letra **i** en el bucle **for**, cada valor ingresado por el usuario para cada cliente se guardará en la lista previamente creada, finalmente usaremos el comando **assign** para indicar que cree esa nueva columna y el comando **rename**, para nombrar la columna creada, como el usuario lo ha indicado.

- Para la función **filtrar**, se usó el siguiente código:

```
def filtrar():
    global df
    print("Las columnas son: ", df.columns)
    filtro=input("Ingrese la columna que desee filtrar")
    caract=input("Ingrese el valor que desea hallar")
    f=df.loc[df[filtro]==caract]
    return f
```

Como se puede observar el sistema mostrará al usuario todas las columnas existentes para que a este se le sea más fácil seleccionar el que desea, por ello usamos el comando **print** y **df.columns**, después se le pedirá al usuario ingresar el nombre de la columna que desea filtrar, la cual estará almacenada en la variable **filtro**, asimismo también le pedirá que ingrese el valor que desea hallar, almacenándolo en la variable **caract** y finalmente se hará uso del comando **df.loc** para localizar y buscar la información que se desea siguiendo la sintaxis mostrada.

- Para la función ordenar, se usó el siguiente código:

```
def ordenar():
    global df
    print("Las columnas son: ", df.columns)
    col=input("Ingrese la columna que desee ordenar")
    decision=input("Ingrese A para ordenar de manera ascendente o D para descend")
    if decision=="A":
        df.sort_values(col,ascending=True)
    elif decision=="D":
        df.sort_values(col,ascending=False)
```

El sistema muestra las columnas ya existentes para que el usuario pueda seleccionar la que desee fácilmente, para ello se usa el comando **print** y **df.columns**, luego el usuario ingresará el nombre de la columna que desea ordenar y se guardará en la variable **col**, también el sistema le preguntará si desea ordenarlo de manera ascendente (A) o descendente (D), y la decision tomada se guardará en la variable **decision**, además se hace uso del comando **if** para condicionar que si es = A, se usará el comando **df.sort_values(col,ascending=True)**, de lo contrario se usará el comando **df.sort_values(col,ascending=False)**

- Para la función histograma, se usó el siguiente código:

```
def histograma():
    global df
    print("Las columnas son: ", df.columns)
    Variable_Hist = list(df[ input("¿Con qué variable quiere realizar el histogr")
    plt.hist(Variable_Hist , bins = 6, alpha = 0.5, histtype='bar', color='steel
    plt.show()
```

Como se visualiza, el sistema imprime las columnas ya existentes para que el usuario pueda seleccionar la que desee fácilmente, para ello se usa el comando **print** y **df.columns**, luego el sistema le preguntara al usuario con que variables trabajará su histograma, los valores de la columna elegida, se almacenarán en una lista llamada **Variable_Hist**, después usaremos el comando **plt.hist**, para crear el histograma y **plt.show()** para mostrar el grafico al usuario.

- Para la función dispersión, se usó el siguiente código:

```
def dispersion():
    global df
    print("Las columnas son: ", df.columns)
    Var_Disp = int(input("¿Cuántas variables desea para el gráfico de dispersión"))
    if Var_Disp == 2:
        Variable_Disp1 = list(df[ input("Ingrese la variable numérica 1: ")])
        Variable_Disp2 = list(df[ input("Ingrese la variable numérica 2: ")])
        plt.scatter(Variable_Disp1 ,Variable_Disp2, alpha=0.5, cmap='viridis')
        plt.colorbar()
        plt.show()
    elif Var_Disp == 3:
        Variable_Disp1 = list(df[ input("Ingrese la variable numérica 1: ")])
        Variable_Disp2 = list(df[ input("Ingrese la variable numérica 2: ")])
        Variable_Disp3 = list(df[ input("Ingrese la variable categórica para ser p")])
        plt.scatter(Variable_Disp1 ,Variable_Disp2,c = Variable_Disp3, alpha=0.5)
        plt.colorbar()
        plt.show()
    elif Var_Disp == 4:
        Variable_Disp1 = list(df[ input("Ingrese la variable numérica 1: ")])
        Variable_Disp2 = list(df[ input("Ingrese la variable numérica 2: ")])
        Variable_Disp3 = list(df[ input("Ingrese la variable categórica para ser p")])
        Variable_Disp4 = list(df[ input("Ingrese la variable numérica para ser p")])
        plt.scatter(Variable_Disp1 ,Variable_Disp2,c = Variable_Disp3,s = Variable_Disp4)
        plt.colorbar()
        plt.show()
```

El sistema mostrará las columnas ya existentes para que el usuario pueda seleccionar la que desee fácilmente, para ello se usa el comando **print** y **df.columns**, además de preguntarle al usuario la cantidad de variables que va a utilizar para el gráfico, dicha cantidad se almacenará en la variable **Var_Disp**, usaremos el comando **if** y **elif** para condicionar de acuerdo a la cantidad de variables escritas, en esta situación si el usuario desea 2 variables, el usuario ingresará cada uno de los nombres de las variables numéricas, las cuales se guardarán en sus respectivas listas llamadas **Variable_Disp1** y **Variable_Disp2**, también se hace uso del comando **plt.scatter** para la creación del gráfico de dispersión, **plt.colorbar** para añadir una barra de colores al gráfico y **plt.show** para mostrar al usuario este gráfico; de esa misma manera se hará cuando el usuario elija de 3 a 4 variables, pero teniendo en cuenta que ingresará una variable categórica.

Luego de definir cada función, se crea la variable **state**, el cual almacenará el valor que me arroja la función **usuario_decision ()**.

```
state=usuario_decision()
while True:
    if state==1:
        añadir_fila()
        print("Cliente(s) añadidos con éxito")
        state=usuario_decision()

    if state==2:
        añadir_columna()
        print("Columna agregada con éxito")
        state=usuario_decision()

    if state==3:
        f=filtrar()
        print("Filtro Aplicado: ")
        print(f)
        state=usuario_decision()

    if state==4:
        ordenar()
        print("Orden Aplicado")
        print(df)
        state=usuario_decision()

    if state==5:
        print(df.describe())
        state=usuario_decision()

    if state==6:
        resp_graf="s"
        while resp_graf == "s":
            Tipo_Graf = str(input("¿Qué tipo de gráfico desea?, escriba hist pa
            if Tipo_Graf == "hist":
                histograma()
                break
            elif Tipo_Graf == "disp":
                dispersion()
                break
            else:
                break
        state=usuario_decision()

    if state==7:
        break
```

Esta variable se usará en el bucle **while**, lo que nos permitirá ejecutar las funciones creadas anteriormente de acuerdo a la decision que tome el usuario, al ser varias funciones usamos el condicional **if** y **print** para que el sistema muestre que la función fue ejecutada exitosamente, y posteriormente volverá a ejecutar la función **usuario_decision ()**, para preguntarle de nuevo al usuario que actividad desea hacer, el bucle termina cuando el usuario ingresa cualquier otra letra y se cortará el flujo, debido al comando **break**.

Luego de salir del bucle, el sistema le preguntará al usuario el nombre con el que desea guardar el archivo, y este se guardará en la variable excel, luego se hará uso del comando **df.to_excel**, que permite convertir el DataFrame a Excel y guardarlo.

Ejemplo:

- **Para añadir una fila:**

```
Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar da
tos,O para ordenar segun alguna columna, D para Datos estadísticos, H para gráfi
cos y cualquier otra tecla para Salir y guardar el Excel: S
Ingrese valor de la columnaRUC: 78549685425
Ingrese valor de la columnaCliente: Carla Piedra Almestar
Ingrese valor de la columnaOperador: Visa
Ingrese valor de la columnaDescripción: muebles blancos
Ingrese valor de la columnaMonto: 1500
Ingrese valor de la columnaFecha: 21/01/2022
Cliente(s) añadidos con éxito
Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar da
tos,O para ordenar segun alguna columna, D para Datos estadísticos, H para gráfi
cos y cualquier otra tecla para Salir y guardar el Excel: S
Ingrese valor de la columnaRUC: 85475236879
Ingrese valor de la columnaCliente: Ariana Ramirez Vera
Ingrese valor de la columnaOperador: Mastercard
Ingrese valor de la columnaDescripción: comedor de madera 8 sillas
Ingrese valor de la columnaMonto: 1320
Ingrese valor de la columnaFecha: 14/09/2021
Cliente(s) añadidos con éxito
Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar da
tos,O para ordenar segun alguna columna, D para Datos estadísticos, H para gráfi
cos y cualquier otra tecla para Salir y guardar el Excel: S
Ingrese valor de la columnaRUC: 69582314569
Ingrese valor de la columnaCliente: Ricardo Gambini Gamboa
Ingrese valor de la columnaOperador: Visa
Ingrese valor de la columnaDescripción: cama queen
Ingrese valor de la columnaMonto: 1850
Ingrese valor de la columnaFecha: 5/12/2019
Cliente(s) añadidos con éxito
Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar da
tos,O para ordenar segun alguna columna, D para Datos estadísticos, H para gráfi
cos y cualquier otra tecla para Salir y guardar el Excel: S
Ingrese valor de la columnaRUC: 14785245693
Ingrese valor de la columnaCliente: Alvaro Ipanaque Hau Yon
Ingrese valor de la columnaOperador: Visa
Ingrese valor de la columnaDescripción: juegos de ollas (10 piezas)
Ingrese valor de la columnaMonto: 2000
Ingrese valor de la columnaFecha: 13/05/2021
```

- **Para añadir una columna:**

```
Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar datos,
O para ordenar segun alguna columna, D para Datos estadísticos, H para gráficos y cu
alquier otra tecla para Salir y guardar el Excel: C
Ingrese el nombre de la columna nueva: Sexo
Escriba el valor del cliente N°0F
Escriba el valor del cliente N°1F
Escriba el valor del cliente N°2M
Escriba el valor del cliente N°3M
Escriba el valor del cliente N°4M
Escriba el valor del cliente N°5M
Columna agregada con éxito
```


- **Para filtrar:**

```

Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar datos,
O para ordenar segun alguna columna, D para Datos estadísticos, H para gráficos y cu
alquier otra tecla para Salir y guardar el Excel: F
Las columnas son: Index(['RUC', 'Cliente', 'Operador', 'Descripción', 'Monto', 'Fec
ha', 'Sexo'], dtype='object')
Ingrese la columna que desee filtrar: Operador
Ingrese el valor que desea hallar: Visa
Filtro Aplicado:
      RUC      Cliente Operador  ... Monto      Fecha Sexo
0  78549685425  Carla Piedra Almestar  Visa  ...  1500  21/01/2022  F
2  69582314569  Ricardo Gambini Gamboa  Visa  ...  1850  5/12/2019  M
3  14785245693  Alvaro Ipanaque Hau Yon  Visa  ...  2000  13/05/2021  M
5  78529852145  Hugo Saldarriaga Valencia  Visa  ...  1099  4/10/2022  M

[4 rows x 7 columns]

```

- **Para ordenar:**

```

Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar datos,
O para ordenar segun alguna columna, D para Datos estadísticos, H para gráficos y cu
alquier otra tecla para Salir y guardar el Excel: O
Las columnas son: Index(['RUC', 'Cliente', 'Operador', 'Descripción', 'Monto', 'Fec
ha', 'Sexo'], dtype='object')
Ingrese la columna que desee ordenar: Monto
Ingrese A para ordenar de manera ascendente o D para descendente: D
Orden Aplicado
      RUC      Cliente  ...      Fecha Sexo
3  14785245693  Alvaro Ipanaque Hau Yon  ...  13/05/2021  M
2  69582314569  Ricardo Gambini Gamboa  ...  5/12/2019  M
0  78549685425  Carla Piedra Almestar  ...  21/01/2022  F
1  85475236879  Ariana Ramirez Vera  ...  14/09/2021  F
4  74123658963  Christian Tipe Verástegui  ...  28/10/2022  M
5  78529852145  Hugo Saldarriaga Valencia  ...  4/10/2022  M

[6 rows x 7 columns]

```

- **Para Datos Estadísticos:**

```

Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar datos,
O para ordenar segun alguna columna, D para Datos estadísticos, H para gráficos y cu
alquier otra tecla para Salir y guardar el Excel: D
      RUC      Cliente Operador  ... Monto      Fecha Sexo
count      6      6      6  ...      6      6      6
unique      6      6      3  ...      6      6      2
top  14785245693  Alvaro Ipanaque Hau Yon  Visa  ...  2000  13/05/2021  M
freq      1      1      4  ...      1      1      4

[4 rows x 7 columns]

```

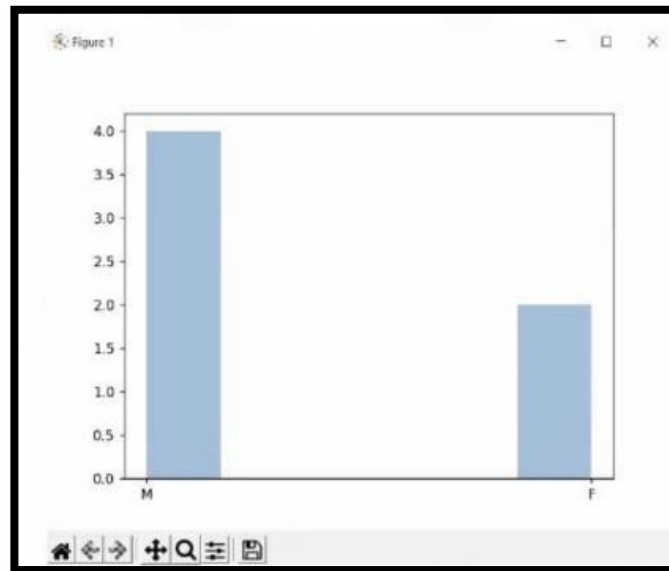
- **Para realizar gráficos:**

1. **Histograma:**

```

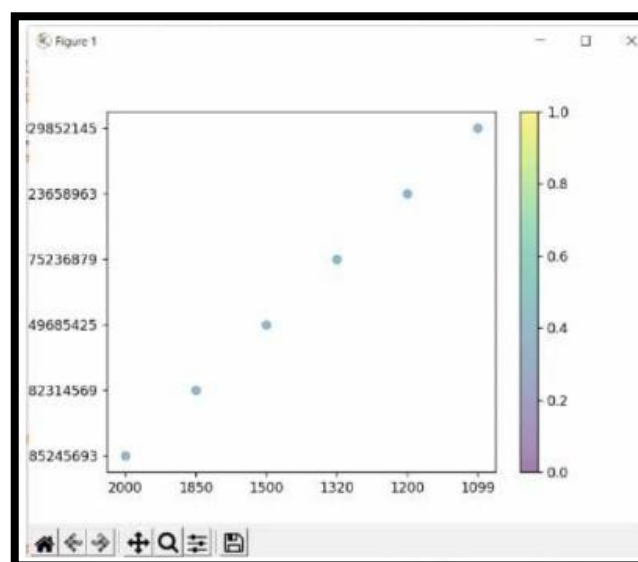
Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar datos,
O para ordenar segun alguna columna, D para Datos estadísticos, H para gráficos y cu
alquier otra tecla para Salir y guardar el Excel: H
¿Qué tipo de gráfico desea?, escriba hist para histograma, o disp para dispersión:
hist
Las columnas son: Index(['RUC', 'Cliente', 'Operador', 'Descripción', 'Monto', 'Fec
ha', 'Sexo'], dtype='object')
¿Con qué variable quiere realizar el histograma?Sexo

```



2. Dispersión:

Escriba S para añadir Clientes, C para añadir nuevas Columnas, F para filtrar datos, O para ordenar según alguna columna, D para Datos estadísticos, H para gráficos y cualquier otra tecla para Salir y guardar el Excel: H
 ¿Qué tipo de gráfico desea?, escriba hist para histograma, o disp para dispersión:
 disp
 Las columnas son: Index(['RUC', 'Cliente', 'Operador', 'Descripción', 'Monto', 'Fecha', 'Sexo'], dtype='object')
 ¿Cuántas variables desea para el gráfico de dispersión?2
 Ingrese la variable numérica 1: Monto
 Ingrese la variable numérica 2: RUC



- Para guardar:

```

Ingrese Nombre del Archivo excel(Se guardará en el escritorio)EJEMPLO01
RUC Cliente ... Fecha Sexo
3 14785245693 Alvaro Ipanaque Hau Yon ... 13/05/2021 M
2 69582314569 Ricardo Gambini Gamboa ... 5/12/2019 M
0 78549685425 Carla Piedra Almestar ... 21/01/2022 F
1 85475236879 Ariana Ramirez Vera ... 14/09/2021 F
4 74123658963 Christian Tipe Verástegui ... 28/10/2022 M
5 78529852145 Hugo Saldarriaga Valencia ... 4/10/2022 M

[6 rows x 7 columns]

```

The screenshot shows the Microsoft Excel interface with the 'Inicio' (Home) tab selected. The table data is as follows:

	A	B	C	D	E	F	G	H
		RUC	Cliente	Operador	Descripción	Monto	Fecha	Sexo
1								
2	3	14785245693	Alvaro Ipanaque Hau Yon	Visa	juegos de ollas (10 piezas)	2000	13/05/2021	M
3	2	69582314569	Ricardo Gambini Gamboa	Visa	cama queen	1850	5/12/2019	M
4	0	78549685425	Carla Piedra Almestar	Visa	muebles blancos	1500	21/01/2022	F
5	1	85475236879	Ariana Ramirez Vera	Mastercard	comedor de madera 8 sillas	1320	14/09/2021	F
6	4	74123658963	Christian Tipe Verástegui	American Express	Ropero	1200	28/10/2022	M
7	5	78529852145	Hugo Saldarriaga Valencia	Visa	juego de cortinas	1099	4/10/2022	M
8								
9								
10								

Conclusiones:

- Los diferentes conceptos aprendidos a lo largo del curso de Análisis de Datos con Python (Nivel 1), se complementan para dar como resultado un conjunto de argumentos que cumplen una función determinada dentro del programa, por lo que la base teórica es importante para hacer realidad el objetivo planeado, que en este caso es hacer correr el programa de flujo de transacciones de manera correcta.
- Gracias a este curso se ha podido aprender a solucionar problemáticas de la vida cotidiana mediante el uso del lenguaje de programación Python y así poder automatizar ciertos procesos que muchas veces realizamos de manera manual y secuencial.
- El uso de librerías ha sido indispensable en este trabajo, ya que con ellas hemos podido realizar acciones de manera sencilla en vez de utilizar códigos que son mucho más largos y que pueden tomar mucho más tiempo digitarlos.
- El uso de la creación de nuevas funciones mediante el argumento: “def ()” ayuda a generar un código más limpio y fácil de entender para el programador o para otras personas que lean el código, ya que permite tener un mejor orden y visualización de este.
- La implementación de los estados en el código fue parte principal para poder permitir al usuario decidir entre diferentes acciones que desee realizar y todo ello se complementó gracias al bucle “while True:”.
- Durante la elaboración del programa hubo algunas confusiones al momento de definir variables locales y globales, pero finalmente dichas dudas fueron solucionadas.