

3. Elementos de programación estructurada para desarrollo de aplicaciones.

En los programas anteriores hemos usado la instrucción *def* para generar funciones y procedimientos como una forma de crear aplicaciones estructuradas. Este paradigma de programación permite organizar mejor la escritura del programa, reutilizar el código para evitar la repetición de las instrucciones, definir funciones y servir de punta de lanza para la programación orientada en objetos.

Un procedimiento es un conjunto de instrucciones definidas al inicio de un programa y que pueden ser llamadas más de una vez durante la corrida del mismo. En la programación estructurada, los procedimientos son los elementos básicos para desarrollar programas más funcionales. Al ser llamadas o invocadas varias veces, minimizan el código, aumentan la legibilidad y fomentan la reutilización de código. A diferencia de otros lenguajes, en *Python*, un procedimiento no existe hasta que es invocado. Se usa la palabra reservada *def* para crear un procedimiento o definir una función, con la diferencia que la segunda devuelve un valor.

Es importante recordar que un procedimiento, aunque esté escrito dentro de un programa, no existe en el mismo hasta que se llama o invoca. Para invocar un procedimiento, deberá escribirse su nombre dentro de un programa o de otro procedimiento, respetando los parámetros o variables que se requieren para pasar los valores. A continuación, desarrollaremos un programa para obtener los parámetros estadísticos descriptivos de un conjunto de datos utilizando procedimientos, compararemos los resultados con las funciones directas que existen en la biblioteca interna *statistics*, se utilizarán algunos elementos de la biblioteca *Numpy* y generaremos tres gráficas: histograma, puntos y líneas con la biblioteca *Matplotlib*.

```
import math as m
import easygui as eg
import numpy as np
from tabulate import tabulate as tab
import matplotlib.pyplot as plt

def Datos():
    txt="Desea ingresar los datos (Yes) \n"
    tx2="o cargar el vector de datos aleatorio (No):"
    yn=eg.ynbox(txt+tx2, "Estadística/Opción inicial")
    n=eval(eg.enterbox("Número de datos:", "Estadística/Ingresar dato"))
    if yn==True:
        datos=[]
        for i in range(0,n):
            dato=eval(eg.enterbox("Dato No. "+str(i), "Estadística/Ingresar dato"))
            datos.append(dato)
        datos=np.array(datos)
    else:
        vi=eval(eg.enterbox("Valor inicial:", "Estadística/Ingresar dato"))
        vf=eval(eg.enterbox("Valor final:", "Estadística/Ingresar dato"))
        datos=np.random.randint(vi, vf+1, n)
    numero=[]
    num=len(datos)
    for i in range(0, num):
        numero.append(i+1)
    vector=[numero, datos]
    encabe=["Número", "Dato"]
    txt=tab(np.transpose(vector), encabe)+"\n\n"+"Suma = "+str(sum(datos))
    eg.msgbox(txt, "Estadística/Tabla resultados")
    mostrar=eg.ynbox("Se muestran los parámetros uno a uno (Yes)\n"+
                    "o se muestran todos los parámetros (No)",
                    "Estadística/Opción de parámetros")
    return datos,mostrar
```

```

def Media(datos,mostrar=True):
    n=len(datos)
    suma=0
    for i in range(0,n):
        suma=suma+datos[i]
    media=suma/n
    if mostrar:
        eg.msgbox("Media aritmética (programa): "+str(round(media,4))+"\n"+
            "Media aritmética (statistics): "+str(round(datos.mean(),4)),
            "Estadística/Resultados")
    return media
def Varianza(datos,media,mostrar=True):
    n=len(datos)
    suma=0
    for i in range(0,n):
        suma=suma+(datos[i]-media)**2
    varianza=suma/n
    if mostrar:
        eg.msgbox("Varianza (programa): "+str(round(varianza,4))+"\n"+
            "Varianza (statistics): "+str(round(datos.var(),4)),
            "Estadística/Resultados")
    return varianza
def DesviacionEstandar(varianza,datos,mostrar=True):
    desvstd=m.sqrt(varianza)
    if mostrar:
        eg.msgbox("Desviación estándar (programa): "+str(round(desvstd,4))+"\n"+
            "Desviación estándar (statistics): "+str(round(datos.std(),4)),
            "Estadística/Resultados")
    return desvstd
def DesviacionMedia(datos,media,mostrar=True):
    n=len(datos)
    suma=0
    for i in range(0,n):
        suma=suma+abs(datos[i]-media)
    desvmedia=suma/n
    if mostrar:
        eg.msgbox("Desviación media: "+str(round(desvmedia,4)),
            "Estadística/Resultados")
    return desvmedia
def Rango(datos,mostrar=True):
    maximo=datos.max()
    minimo=datos.min()
    rango=maximo-minimo
    if mostrar:
        eg.msgbox("Máximo de los datos: "+str(maximo)+"\n"+
            "Mínimo de los datos: "+str(minimo)+"\n"+
            "Rango de los datos : "+str(rango),
            "Estadística/Resultados")
    return rango
def CoefVar(media,desvstd,mostrar=True):
    coefvar=desvstd/media*100
    if mostrar:
        eg.msgbox("Coeficiente de variación: "+str(coefvar)+" %",
            "Estadística/Resultados")
    return coefvar
def TablaFrec(datos):
    ordenados=np.sort(datos)
    n=len(datos)
    if n<=20:
        k=int(m.sqrt(n))+1
    else:
        k=int(1+3.22*m.log10(n))+1
    c=(datos.max()-datos.min())/k
    liminf=[]

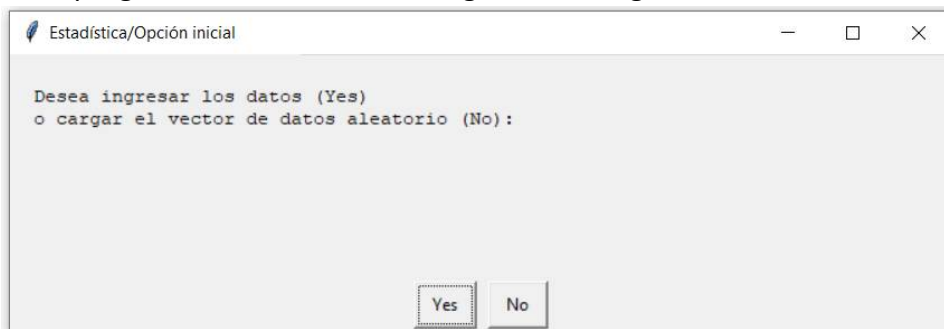
```

```

limsup=[]
liminf.append(datos.min())
limsup.append(liminf[0]+c)
marcaclase=[]
marcaclase.append((limsup[0]+liminf[0])/2)
for i in range(1,k):
    liminf.append(liminf[i-1]+c)
    limsup.append(limsup[i-1]+c)
    marcaclase.append(marcaclase[i-1]+c)
frec=[]
for i in range(0,k):
    conteo=0
    for j in range(0,n):
        if liminf[i]<=ordenados[j]<=limsup[i]:
            conteo=conteo+1
    frec.append(conteo)
tabla=[liminf,limsup,frec,marcaclase]
cabeza=["Lim.Inf.", "Lim.Sup", "Frec.", "Marca Clase"]
tx1="Ancho de clase: "+str(c)+"\n\n"
tx2=tab(np.transpose(tabla),cabeza)+"\n\nSuma de frecuencias: "+str(sum(frec))
eg.msgbox(tx1+tx2,
          "Estadística/Tabla")
return marcaclase,frec
def Graficos(marcaclase,frec):
fig, axs = plt.subplots(1, 3, figsize=(9, 3), sharey=True)
axs[0].bar(marcaclase, frec)
axs[1].scatter(marcaclase, frec)
axs[2].plot(marcaclase, frec,"+-")
fig.suptitle('Gráficos')
plt.show()
def Resultados(datos,mostrar):
media=Media(datos,mostrar)
varianza=Varianza(datos,media,mostrar)
devstd=DesviacionEstandar(varianza,datos,mostrar)
devmedia=DesviacionMedia(datos,media,mostrar)
rango=Rango(datos,mostrar)
coefvar=CoefVar(media,devstd,mostrar)
txt="Rango de datos: "+str(rango)+"\n"+"Media: "+str(media)+"\n"
tx2="Varianza: "+str(varianza)+"\n"+"Desviación Estándar: "+str(devstd)+"\n"
tx3="Coeficiente de variación: "+str(coefvar)+" %\n"
tx4="Desviación media: "+str(devmedia)+"\n"
if not(mostrar):
    eg.msgbox(txt+tx2+tx3+tx4,
              "Estadística/Resultados")
def Principal():
otro=True
while otro:
    datos,mostrar=Datos()
    Resultados(datos,mostrar)
    marcaclase,frec=TablaFrec(datos)
    Graficos(marcaclase,frec)
    otro=eg.ynbox("Otro (Yes/No): ",
                  "Estadística/Opción")
    if not(otro):
        break
Principal()

```

La corrida del programa se muestra en las siguientes imágenes:



Estadística/Ingresar dato

Número de datos:

OK Cancel

Estadística/Ingresar dato

Dato No. 0

OK Cancel

Estadística/Tabla resultados

Número	Dato
1	20
2	25
3	27
4	33
5	35
6	43
7	44
8	56

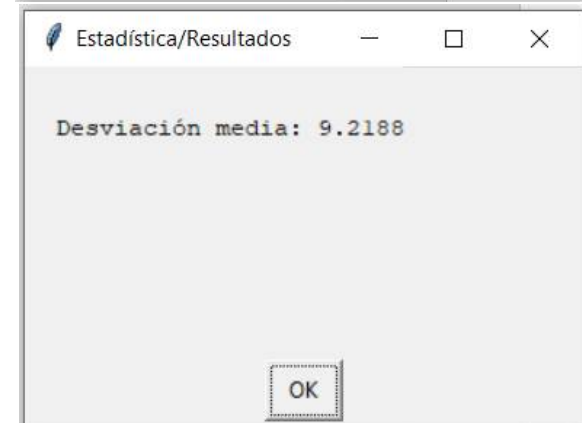
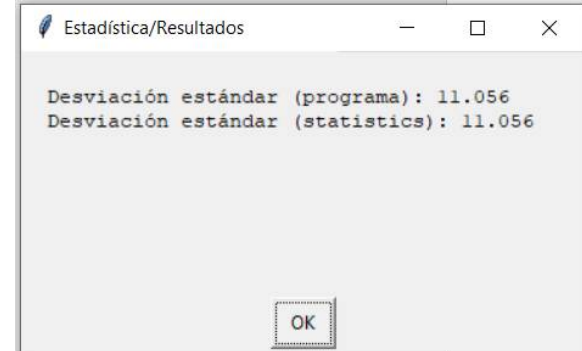
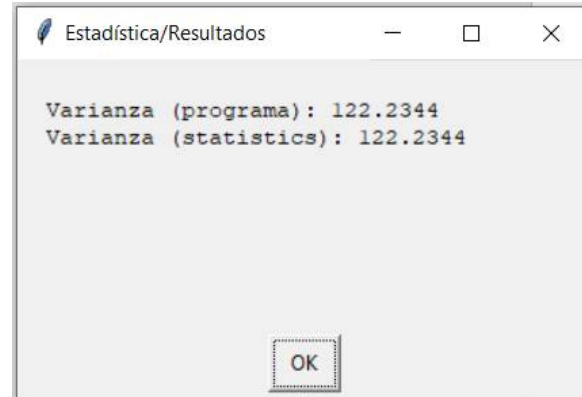
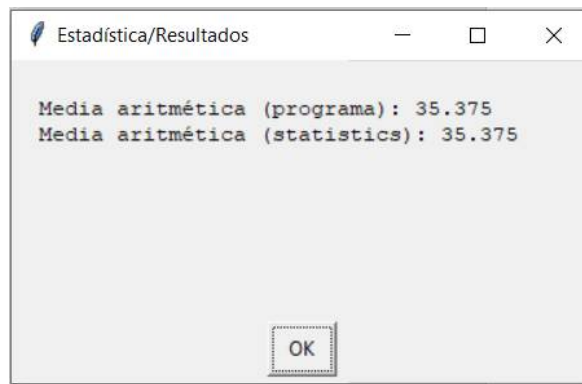
Suma = 283

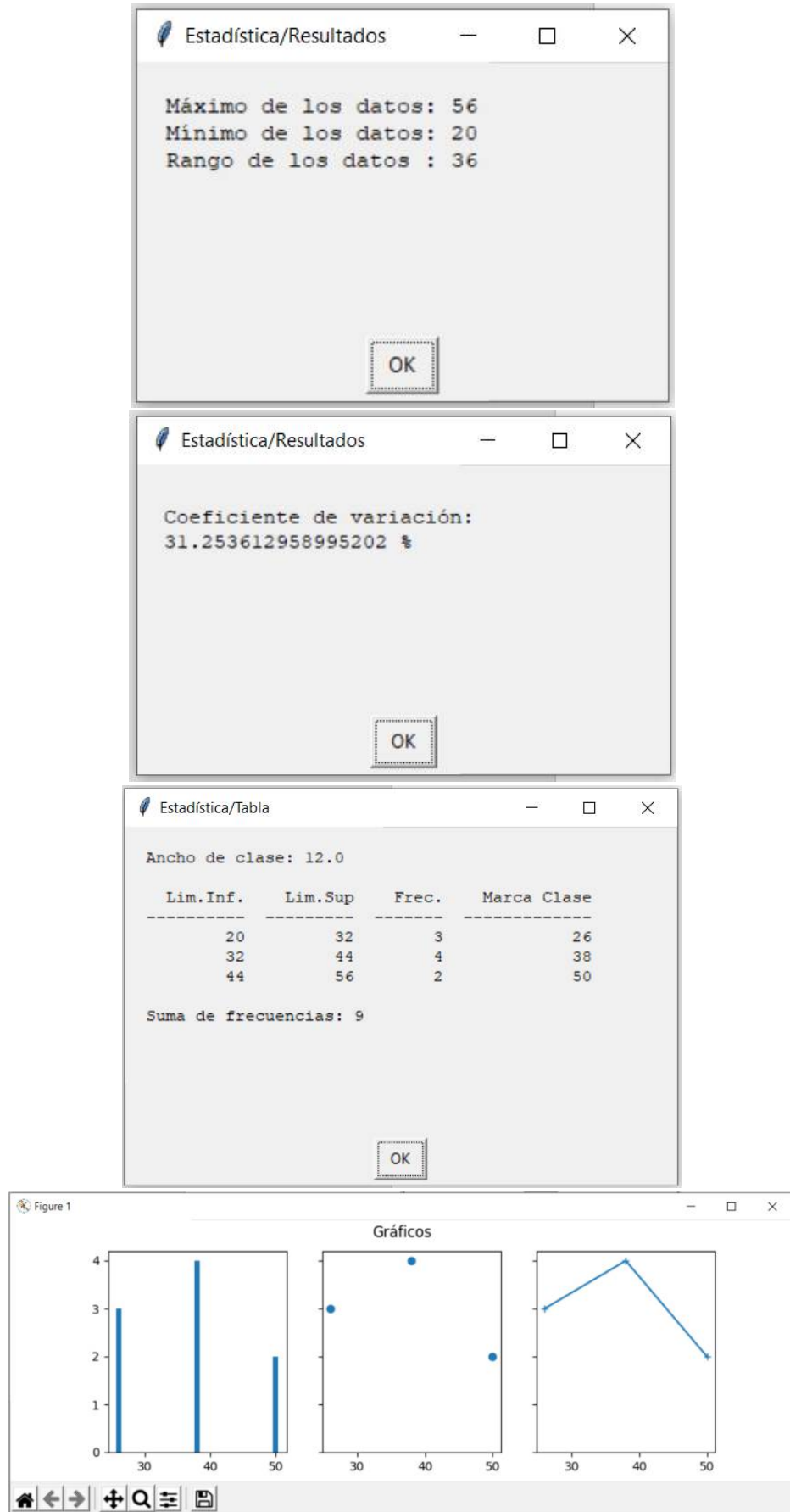
OK

Estadística/Opción de parámetros

Se muestran los parámetros uno a uno (Yes)
o se muestran todos los parámetros (No)

Yes No





Estadística/Opción

Otro (Yes/No) :

Yes No

Estadística/Opción inicial

Desea ingresar los datos (Yes)
o cargar el vector de datos aleatorio (No) :

Yes No

Estadística/Ingresar dato

Número de datos:

25

OK Cancel

Estadística/Ingresar dato

Valor inicial:

10

OK Cancel

Estadística/Ingresar dato

Valor final:

80

OK Cancel

Estadística/Tabla resultados

Número	Dato
1	52
2	42
3	58
4	80
5	58
6	63
7	27
8	54
9	59
10	39
11	20
12	60
13	25
14	58
15	58
16	60
17	39
18	16
19	33
20	51
21	72
22	14
23	75
24	20
25	43

Suma = 1176

OK

Estadística/Opción de parámetros

Se muestran los parámetros uno a uno (Yes)
o se muestran todos los parámetros (No)

Yes No

Estadística/Resultados

Rango de datos: 66
Media: 47.04
Varianza: 342.67839999999995
Desviación Estándar: 18.51157475743217
Coeficiente de variación: 39.352837494541184 %
Desviación media: 15.955200000000001

OK

Estadística/Tabla

Ancho de clase: 11.0

Lim.Inf.	Lim.Sup	Frec.	Marca Clase
14	25	5	19.5
25	36	3	30.5
36	47	4	41.5
47	58	7	52.5
58	69	8	63.5
69	80	3	74.5

Suma de frecuencias: 30

OK

