2. Diseño de la interfaz gráfica de usuario con EasyGUI.

EasyGUI es una biblioteca externa que permite elaborar rápidamente interfaces gráficas de usuarios (*GUI*) con instrucciones muy sencillas, sin embargo, adolece de algunas desventajas contra otras bibliotecas para desarrollar *GUI*, como *TKinter*, *wxPython*.

Posee pocos elementos de comunicación gráfica que se usan comúnmente en entornos de programación orientada a objetos, sin embargo, su uso fácil permite desarrollar rápidamente aplicaciones que comunican eficazmente al usuario.

Una desventaja principal es que la información se muestra ventana a ventana, así que cuando se cierra una, la información desaparece a la vista del usuario, generando una desconexión con los datos que se deben ingresar y con los resultados. Sin embargo, puede invertirse poco tiempo de programación para lograr el efecto deseado.

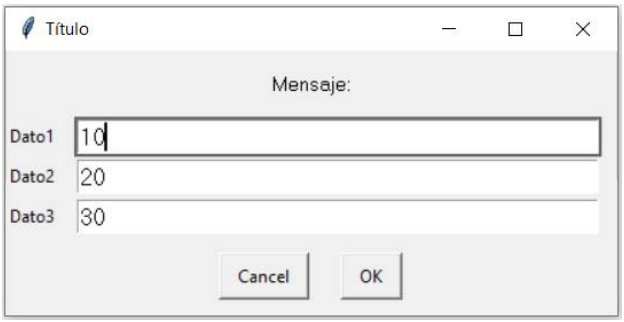
Las herramientas gráficas de *EasyGUI* para usarse en las ventanas son limitadas, pues solo permite el ingreso de datos mediante cajas de texto y el despliegue de los resultados en formato textual, siempre. Posee programados ciertos eventos asociados a botones y no permite el uso de elementos como botones de radio, casilleros de verificación, salidas de datos en renglones y columnas tipo Excel, listas desplegables, etc.; sin embargo, se logran resultados rápidos y útiles con los pocos objetos que posee.

La tabla 1 muestra las instrucciones y su efecto al usar los elementos de *EasyGUI*. Normalmente la sintaxis de estas instrucciones se ajusta a la forma *instruccion(mensaje,titulo,variables,acciones)*.

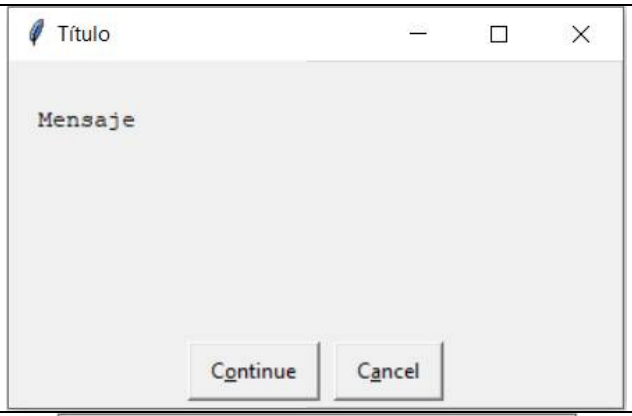
Tabla 1. Elementos de EasyGUI

INSTRUCCIÓN	EFECTO
easygui.enterbox(“Mensaje:”, ”Título”, 100)	
easygui.msgbox(“Mensaje:”, ”Título”, ”Aceptar”)	

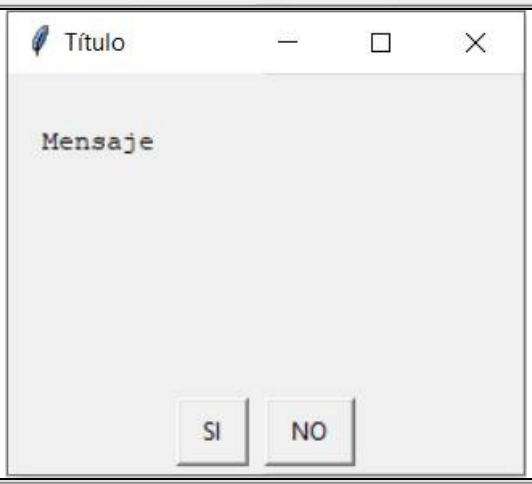
```
datos=[]
rotulos=["Dato1","Dato2","Dato
3"]
valores=[10,20,30]
datos=easygui.multenterbox(
                "Mensaje:",
                "Título",
                rotulos,
                values=valores)
```



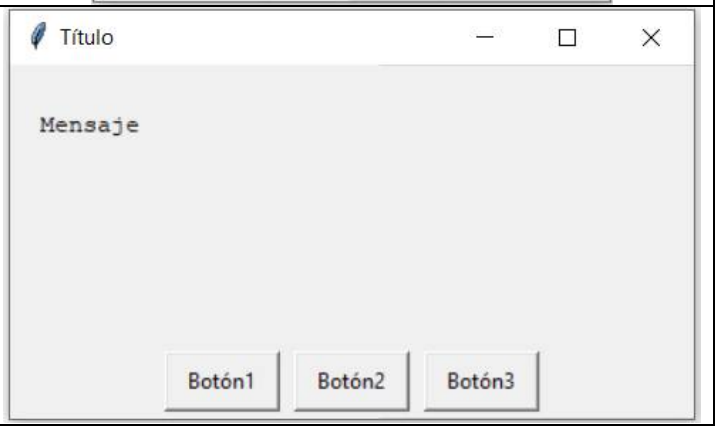

```
easygui.ccbox(“Título”,
                “Mensaje”)
```



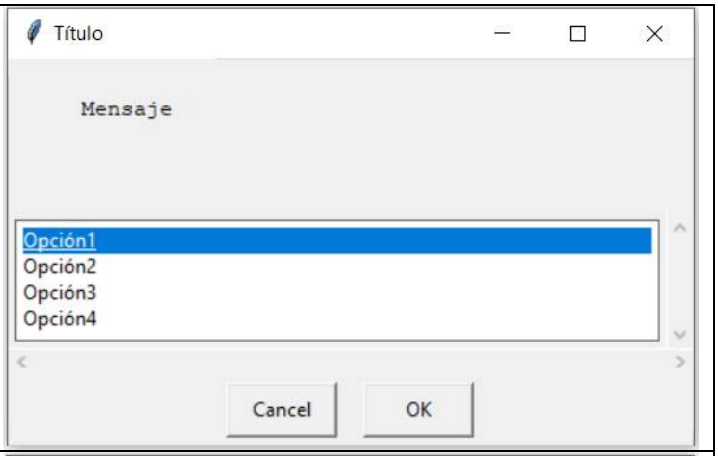

```
easygui.ynbox(“Título”,
                “Mensaje”,
          choices=[“SI”,”NO”)
```



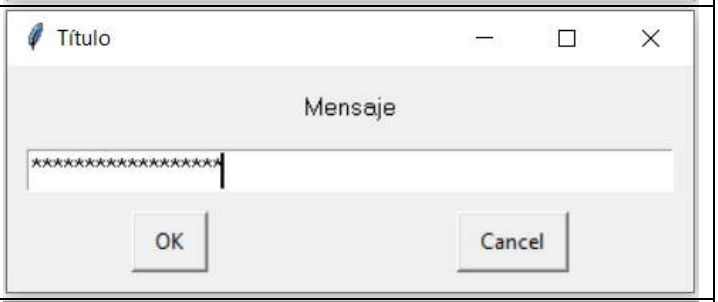

```
import easygui as eg
eg.buttonbox("Mensaje",
          "Título",
  choices=["Botón1",
            "Botón2",
           "Botón3"])
```



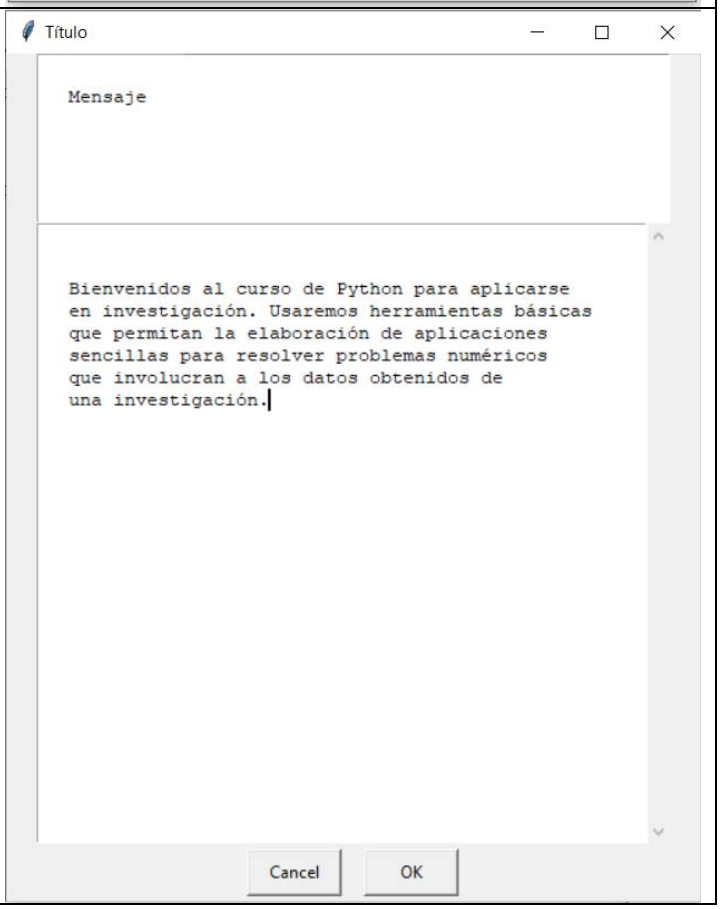
```
eg.choicebox("Mensaje",
             "Título",
        choices=["Opción1",
             "Opción2",
             "Opción3",
             "Opción4"])
```




```
eg.passwordbox("Mensaje",
               "Título")
```

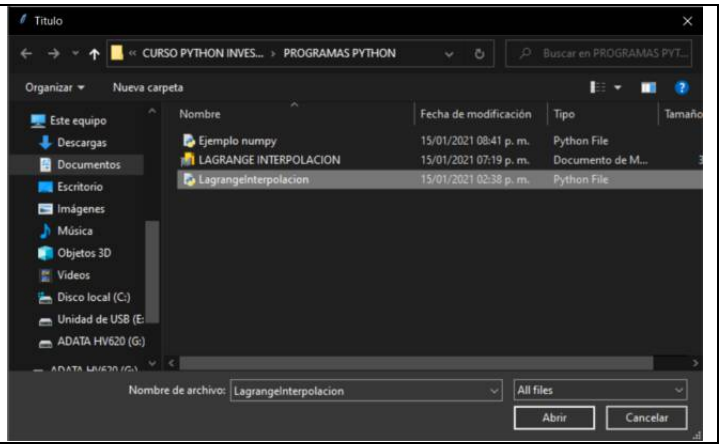



```
texto="""
Bienvenidos al curso de Python para aplicarse
en investigación. Usaremos herramientas básicas
que permitan la elaboración de aplicaciones
sencillas para resolver problemas numéricos
que involucran a los datos obtenidos de
una investigación."""
eg.textbox("Mensaje",
           "Título",
           texto)
```



```
archivo=eg.fileopenbox(
            "Titulo")
print(archivo)

C:\Users\Usuario\Documents\CURSO
PYTHON INVESTIGACION\PROGRAMAS
PYTHON\LagrangeInterpolacion.py

eg.filesavebox("Título")
```

Almacena el archivo.



Ejemplo 2.

Los datos experimentales del índice de compresión y los límites de *Atterberg* para varios tipos de arcillas minerales y suelos naturales se muestra en la tabla 2.

Tabla 2. Valores del índice de compresión Cc, LL y LP para arcillas minerales y suelos.

Tipo de arcilla	*Cc*	*LL (%)*	*LP (%)*
Arcilla azul de Boston	0.32	41	20
Arcilla de Chicago	0.42	58	21
Arcilla de Louisiana	0.33	74	26
Arcilla de New Orleans	0.29	79	26
Arcilla de Fort Union	0.26	89	20
Arcilla limosa orgánica de Delaware	0.95	84	46
Arcilla limosa de Indiana	0.21	36	20
Arcilla de Fore River	0.36	49	21
Arcilla de Beauharnois	0.55	56	22
Arcilla de Cincinnati	0.17	30	12
Arcilla de St Lawrence	0.84	55	22
Arcilla de Siburua	0.21	70	26
Arcilla CL suave	0.34	41	24
Arcilla CL firme	0.44	50	23
Limo arenoso ML	0.16	31	25
Arcilla CH suave	0.84	81	25
Arcilla CH con estratos limosos	0.52	71	28
Montmorillonita Na^{+}	2.60	710	54
Montmorillonita K^{+}	1.00	660	98
Montmorillonita Ca^{+2}	2.20	510	81
Montmorillonita H^{+}	1.90	440	55
Montmorillonita Mg^{+2}	1.90	410	60
Montmorillonita Fe^{+3}	1.60	290	75
Ilita Na^{+}	1.10	120	53
Ilita K^{+}	0.62	120	60
Ilita Ca^{+2}	0.86	100	45

Ilita H^{+}	0.61	100	51
Ilita Mg^{+2}	0.56	94	46
Caolinita Na^{+}	0.26	53	32
Caolinita Ca^{+2}	0.21	38	27
Caolinita H^{+}	0.23	53	25
Caolinita Mg^{+2}	0.24	54	31
Caolinita Fe^{+3}	0.24	59	37
Atapulgita Mg^{+2}	0.77	270	150
Arcilla teja	0.08	24	12

Desarrollar un programa que muestre la tabla de datos, calcule el $IP = LL - LP$, que realice las graficas Cc vs. IP, y que determine una ecuación experimental basada en la recta de mínimos cuadrados.

El modelo de la recta de mínimos cuadrados establece que para encontrar los coeficientes a y b de la ecuación $y(x) = a + bx$, se deberá resolver el sistema siguiente:

$$\begin{bmatrix} n & \sum X \\ \sum X & \sum X^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum Y \\ \sum XY \end{bmatrix}$$

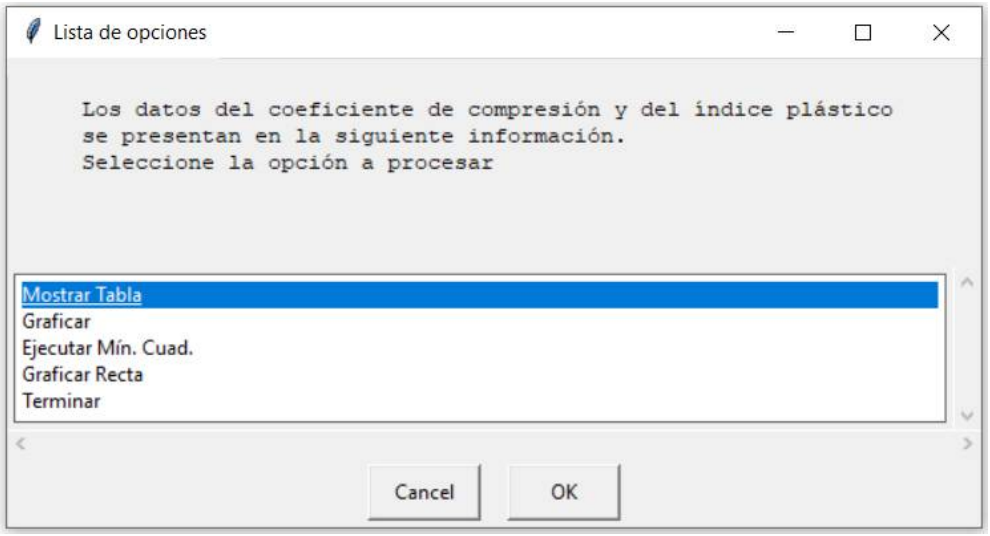
El programa se muestra a continuación:

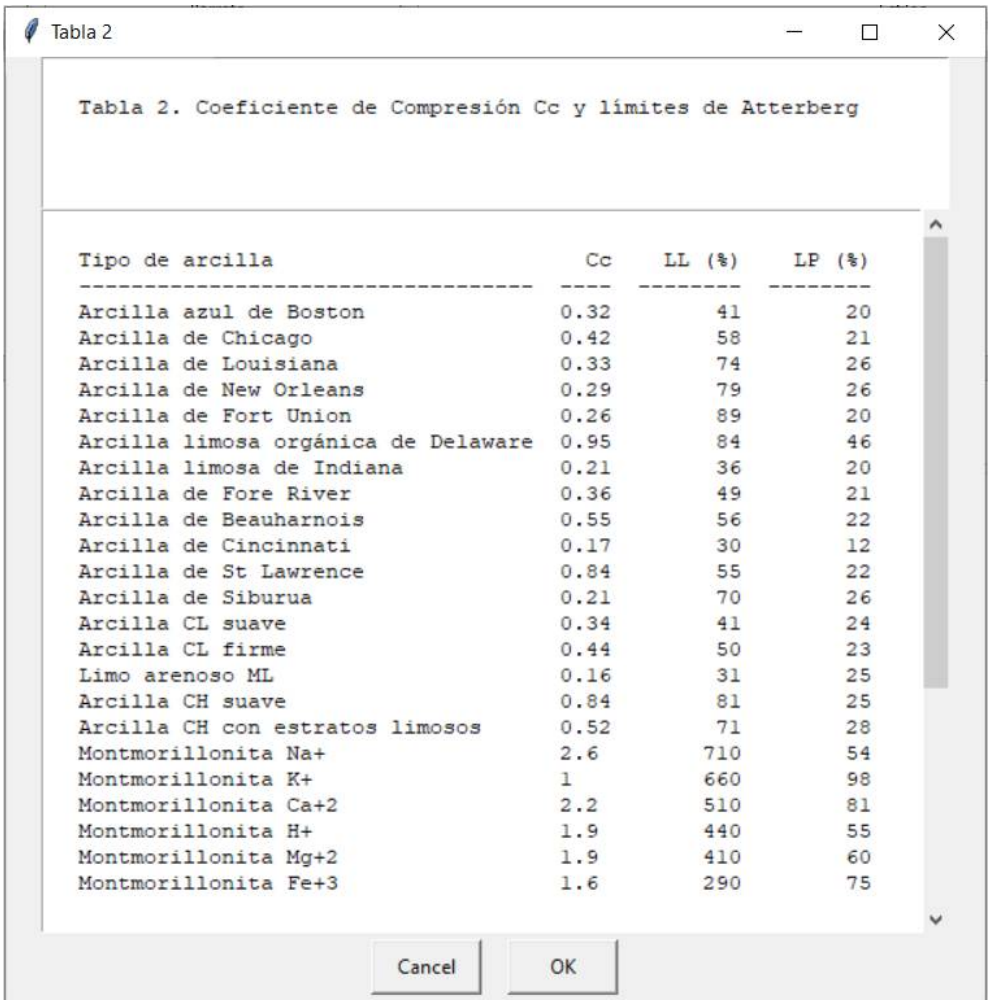
```
import math as m
import numpy as np
import numpy.linalg as al
from tabulate import tabulate as tab
import matplotlib.pyplot as py
import easygui as eg

def Tabla2():
    cabeza=["Tipo de arcilla","Cc","LL (%)","LP (%)"]
    tabla2=[["Arcilla azul de Boston",0.32,41,20],
            ["Arcilla de Chicago",0.42,58,21],
            ["Arcilla de Louisiana",0.33,74,26],
            ["Arcilla de New Orleans",0.29,79,26],
            ["Arcilla de Fort Union",0.26,89,20],
            ["Arcilla limosa orgánica de Delaware",0.95,84,46],
            ["Arcilla limosa de Indiana",0.21,36,20],
            ["Arcilla de Fore River",0.36,49,21],
            ["Arcilla de Beauharnois",0.55,56,22],
            ["Arcilla de Cincinnati",0.17,30,12],
            ["Arcilla de St Lawrence",0.84,55,22],
            ["Arcilla de Siburua",0.21,70,26],
            ["Arcilla CL suave",0.34,41,24],
            ["Arcilla CL firme",0.44,50,23],
            ["Limo arenoso ML",0.16,31,25],
            ["Arcilla CH suave",0.84,81,25],
            ["Arcilla CH con estratos limosos",0.52,71,28],
            ["Montmorillonita Na+",2.60,710,54],
```

```
            ["Montmorillonita K+",1.00,660,98],
            ["Montmorillonita Ca+2",2.20,510,81],
            ["Montmorillonita H+",1.90,440,55],
            ["Montmorillonita Mg+2",1.90,410,60],
            ["Montmorillonita Fe+3",1.60,290,75],
            ["Ilita Na+",1.10,120,53],
            ["Ilita K+",0.62,120,60],
            ["Ilita Ca+2",0.86,100,45],
            ["Ilita H+",0.61,100,51],
            ["Ilita Mg+2",0.56,94,46],
            ["Caolinita Na+",0.26,53,32],
            ["Caolinita Ca+2",0.21,38,27],
            ["Caolinita H+",0.23,53,25],
            ["Caolinita Mg+2",0.24,54,31],
            ["Caolinita Fe+3",0.24,59,37],
            ["Atapulgita Mg+2",0.77,270,150],
            ["Arcilla teja",0.08,24,12]]
    return tabla2,cabeza
def MostrarTabla():
    tabla2,cabeza=Tabla2()
    tabla2txt=tab(tabla2,cabeza)
    eg.textbox("Tabla 2. Coeficiente de Compresión Cc y límites de
Atterberg","Tabla 2",tabla2txt)
def MenuLista():
    texto="""Los datos del coeficiente de compresión y del índice
plástico se presentan en la siguiente información.\n"""
    opciones=["Mostrar Tabla","Graficar","Ejecutar Mín. Cuad.","Graficar
Recta","Terminar"]
    opcion=eg.choicebox(texto+"Seleccione la opción a procesar","Lista de
opciones",choices=opciones)
    return opcion
def CcIP(tabla2):
    Cc=[]
    IP=[]
    for i in range(0,len(tabla2)):
        Cc.append(tabla2[i][1])
        IP.append(tabla2[i][2]-tabla2[i][3])
    return Cc,IP
def MinCuad(x,y):
    n=len(x)
    sumax=sum(x)
    sumay=sum(y)
    x2=[]
    xy=[]
    for i in range(0,len(x)):
        x2.append(x[i]**2)
        xy.append(x[i]*y[i])
    sumax2=sum(x2)
    sumaxy=sum(xy)
    matrizA=[[n,sumax],[sumax,sumax2]]
    matrizB=[[sumay],[sumaxy]]
    ab=al.solve(matrizA,matrizB)
    a=ab[0][0]
    b=ab[1][0]
    return a,b
```

```
def Graficar(x,y):
    py.plot(x,y,"o")
    py.title("Gráfica Cc vs. IP")
    py.xlabel("Coeficiente de compresión Cc")
    py.ylabel("Índice Plástico IP (%)")
    py.show()
def Recta(a,b,x):
    y=[]
    for i in range(0,len(x)):
        y.append(a+b*x[i])
    return y
def GraficaRecta(x,y,recta):
    py.plot(x,y,"o",x,recta,"-")
    py.title("Recta de mínimos cuadrados")
    py.xlabel("Coeficiente de compresión Cc")
    py.ylabel("Índice Plástico IP (%)")
    py.grid(True)
    py.show()

while True:
    tabla2,cabeza=Tabla2()
    opcion=MenuLista()
    if opcion=="Mostrar Tabla":
        MostrarTabla()
    if opcion=="Ejecutar Mín. Cuad.":
        Cc,IP=CcIP(tabla2)
        a,b=MinCuad(Cc,IP)
        mensaje="Recta de Mínimos cuadrados: y(x) = "+str(a)+"+"+str(b)+"
x"
        eg.msgbox(mensaje,"RMC")
    if opcion=="Graficar":
        Cc,IP=CcIP(tabla2)
        Graficar(Cc,IP)
    if opcion=="Graficar Recta":
        Cc,IP=CcIP(tabla2)
        a,b=MinCuad(Cc,IP)
        recta=Recta(a,b,Cc)
        GraficaRecta(Cc,IP,recta)
    if opcion=="Terminar":
        break
```

La corrida del programa se muestra a continuación en las siguientes imágenes.





Tipo de arcilla	Cc	LL (%)	LP (%)
Arcilla azul de Boston	0.32	41	20
Arcilla de Chicago	0.42	58	21
Arcilla de Louisiana	0.33	74	26
Arcilla de New Orleans	0.29	79	26
Arcilla de Fort Union	0.26	89	20
Arcilla limosa orgánica de Delaware	0.95	84	46
Arcilla limosa de Indiana	0.21	36	20
Arcilla de Fore River	0.36	49	21
Arcilla de Beauharnois	0.55	56	22
Arcilla de Cincinnati	0.17	30	12
Arcilla de St Lawrence	0.84	55	22
Arcilla de Siburua	0.21	70	26
Arcilla CL suave	0.34	41	24
Arcilla CL firme	0.44	50	23
Limo arenoso ML	0.16	31	25
Arcilla CH suave	0.84	81	25
Arcilla CH con estratos limosos	0.52	71	28
Montmorillonita Na+	2.6	710	54
Montmorillonita K+	1	660	98
Montmorillonita Ca+2	2.2	510	81
Montmorillonita H+	1.9	440	55
Montmorillonita Mg+2	1.9	410	60
Montmorillonita Fe+3	1.6	290	75

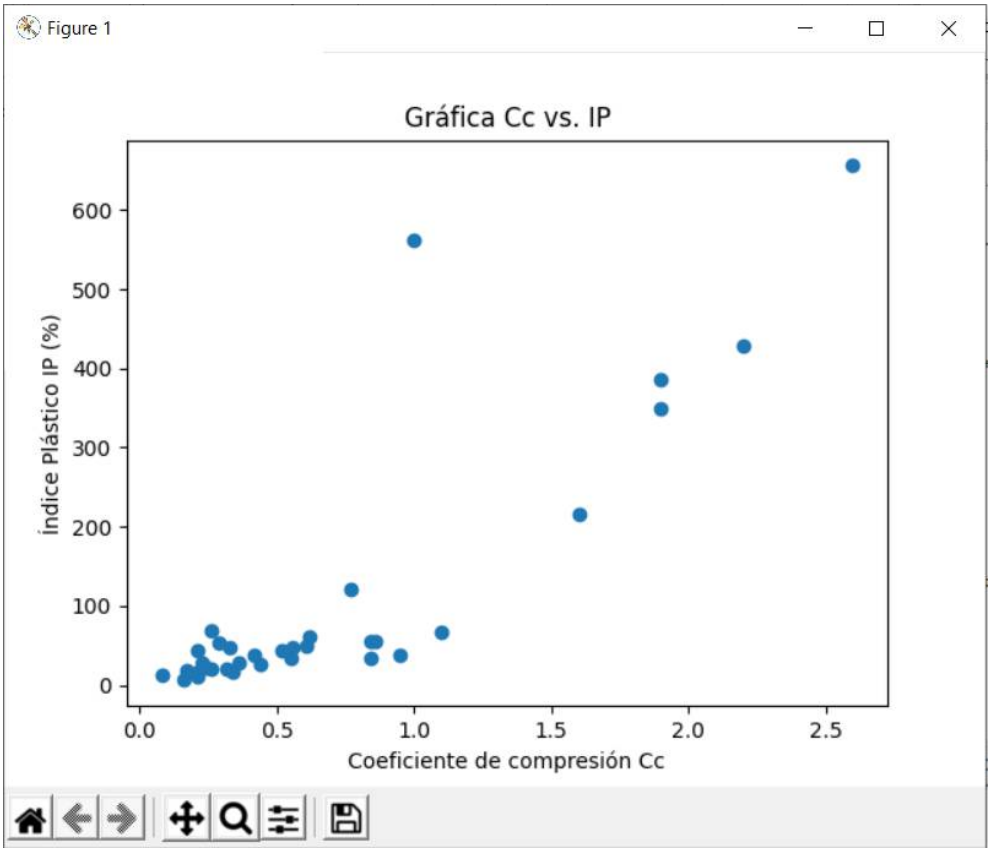


Figure 1
Gráfica Cc vs. IP
Índice Plástico IP (%)
Coeficiente de compresión Cc
0
100
200
300
400
500
600
0.0
0.5
1.0
1.5
2.0
2.5


RMC
Recta de Minimos cuadrados: y(x) = -46.825536721640965 + 220.74798616194434 x
OK

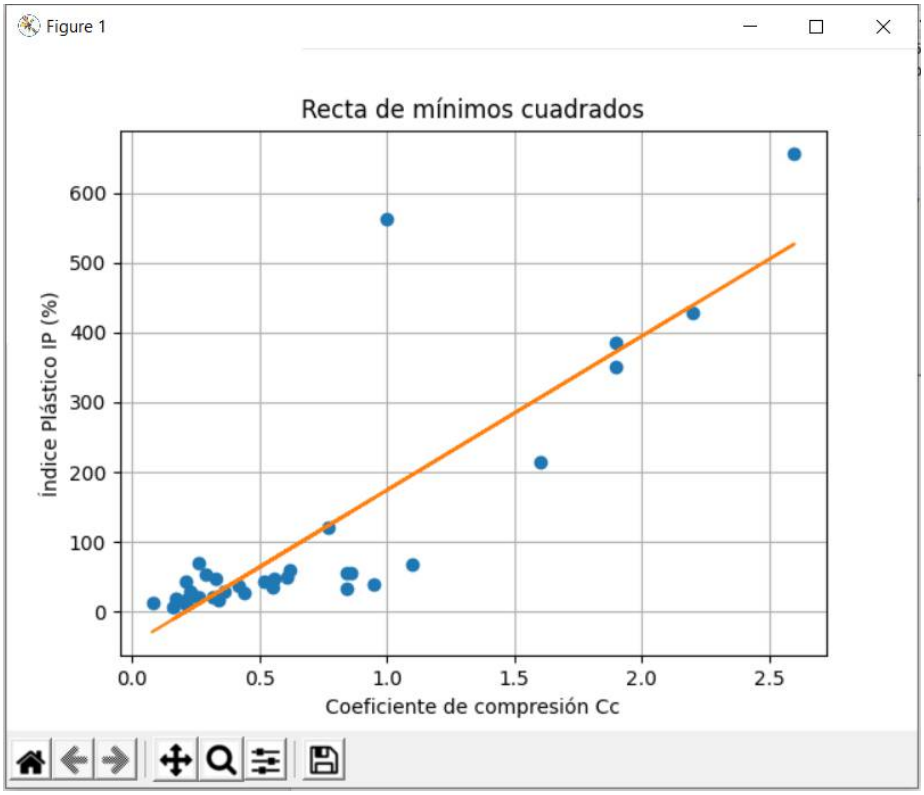


Figure 1
Recta de mínimos cuadrados
Índice Plástico IP (%)
Coeficiente de compresión Cc
0
100
200
300
400
500
600
0.0
0.5
1.0
1.5
2.0
2.5