

# Diseño de aplicativo en Python como soporte de enseñanza aprendizaje para la estabilidad de taludes

Application design in Python as a teaching learning support for slope stability

Roberto Martin Zamora Nevado<sup>1</sup>, Yovana Edith Medina Vásquez<sup>2</sup>

Recibido: 19/04/2023 - Aprobado: 26/06/2023 – Publicado: 06/10/2023

## ABSTRACT

Los procesos de enseñanza-aprendizaje, hoy en día requieren de soportes pedagógicos que le permitan al docente generar un espacio de interacción real con el estudiante desde el aula. La formación de profesionales en ciencias de la tierra o de ingeniería, debe estar acompañada de la enseñanza de programación de computadoras, utilizando lenguajes de alto nivel, de descarga gratuita y de código abierto, como recurso aplicado a la solución de problemas específicos, presentados en la estabilidad de taludes. Aquí se muestra el diseño de un aplicativo en Python para calcular y graficar el factor de seguridad de un talud, a través de una Interfaz Gráfica de Usuario (IGU), insertando: librerías, variables, funciones y widgets propias de este lenguaje. El factor de seguridad obtenido por el aplicativo es de 1.26, que coincide con el obtenido por el procedimiento manual, la coordenada que grafica el factor de seguridad ( $F_s$ ), para determinar su sensibilidad, en función de la altura recíproca del talud ( $H^{-1}$ ) sobre la recta será de (0.005, 1.26), y la coordenada de corte será de (0,0.8).

**Keywords:** Enseñanza-aprendizaje, estabilidad de taludes, factor de seguridad, funciones, interfaz, gráfica, usuario, librerías, Python, variables, widgets.

## RESUMEN

The teaching-learning processes nowadays require pedagogical supports that allow the teacher to generate a space of real interaction with the student from the classroom. The training of professionals in earth sciences or engineering should be accompanied by the teaching of computer programming, using high-level languages, free download and open source, as a resource applied to the solution of specific problems, presented in the stability of slopes. Here is shown the design of a Python application to calculate and plot the factor of safety of a slope, through a Graphical User Interface (GUI), inserting: libraries, variables, functions and widgets of this language. The safety factor obtained by the application is 1.26, which coincides with that obtained by the manual procedure, the coordinate that plots the safety factor ( $F_s$ ), to determine its sensitivity, as a function of the reciprocal height of the slope ( $H^{-1}$ ) on the straight line will be (0.005, 1.26), and the cut-off coordinate will be (0,0.8).

**Palabras claves:** Teaching-learning, stability of slopes, factor of safety, functions, graphical, user, interface, libraries, Python, variables, widgets.

<sup>1</sup> Universidad Tecnológica del Perú (UTP), Chiclayo, Perú.

Docente. Autor para correspondencia: [c26498@utp.edu.pe](mailto:c26498@utp.edu.pe) - ORCID: <https://orcid.org/0000-0001-9741-0708>

<sup>2</sup> Universidad Tecnológica del Perú (UTP), Chiclayo, Perú.

Docente. E-mail: [c18837@utp.edu.pe](mailto:c18837@utp.edu.pe) - ORCID: <https://orcid.org/0000-0002-3872-2047>

## I. INTRODUCCIÓN

La computación, en su avance tecnológico, ha introducido cambios en los modelos educativos en todos sus niveles; pero el universitario es el que más se ha beneficiado, pues le ha permitido al docente utilizarla como un soporte pedagógico, haciendo interactuar al estudiante con la realidad de su carrera desde el aula, potenciando sus competencias de abstracción, capacidad de análisis, creatividad e innovación; fortaleciendo el proceso de enseñanza-aprendizaje. De Puelles (1999) menciona que el docente hoy no solo se concibe como un transmisor de conocimientos sino también como el conductor de un complejo proceso de enseñanza aprendizaje, mucho más dinámico, en el que los estudiantes construyen su propio conocimiento y una propia interpretación del mundo. El aprendizaje de un lenguaje de programación de alto nivel como Python, aparecido ya hace algunos años, pero que aún mantiene su vigencia y popularidad debido a su constante dinámica y evolución a través de sus distintas versiones y actualizaciones, facilitará el diseño de aplicativos con fines de simulación en pruebas y ensayos de materiales, calculadoras sobre temas específicos y graficadoras, tanto en escenarios 2D como 3D. Peña (2016) menciona entre las ventajas de uso de Python: es un lenguaje muy expresivo, por lo que los programas escritos en este lenguaje suelen ser más cortos, su sintaxis es elegante y permite escribir programas muy legibles, propiciando el establecimiento de buenos hábitos de programación, existe una amplia variedad de recursos docentes, documentación, foros y herramientas de visualización, el traductor de Python proporciona información clara sobre los errores de sintaxis, facilitando al programador detectarlos y corregirlos, es de libre distribución y multiplataforma. La licencia de Python es de código abierto y no genera costo alguno por instalación, uso ni actualizaciones.

Los taludes que son pendientes construidas artificialmente para dar acceso a caminos, carreteras, y como elementos estructurales de presas o minas a cielo abierto, al igual que las laderas de los cerros, ven afectada su estabilidad por varios factores, generando a futuro problemas de situaciones complejas, y a veces difíciles de remediar, produciendo deslizamientos de grandes masas de tierra o bloques, causando daños significativos y pérdidas de vidas. De acuerdo a Tumialán (2004) “los deslizamientos son fenómenos muy comunes en las laderas de los valles desde los 800 a los 6000 m s. n. m., que se producen por exceso de lluvia que humedece el suelo cuaternario”. Pariseau (2012) establece que los taludes se pueden clasificar de varias maneras, por ejemplo, como taludes de roca o de suelo, si bien es cierto ambos comparten características fundamentales, como las leyes físicas, los métodos de análisis son generalmente distintos. Los taludes también se pueden clasificar de acuerdo con un modo esperado de falla, digamos, por traslación o rotación, asociando las fallas de taludes de roca de alta resistencia a los movimientos traslacionales, mientras que los taludes de suelo de baja resistencia comienzan a fallar en cuerpos rígidos como rotaciones. González de Vallejo et. al (2004) refiere que los métodos de análisis de estabilidad se pueden agrupar en métodos determinísticos que tienen en consideración los parámetros físicos y resistentes que

controlan el comportamiento del material, a partir de los cuales se define el factor de seguridad del talud, existen dos grupos: métodos de equilibrio límite y métodos tenso deformacionales. Los métodos probabilísticos, consideran la probabilidad de rotura de un talud conociendo las funciones de distribución de los diferentes valores considerados como variables aleatorias, y mediante procesos iterativos obtener las funciones de densidad de probabilidad y distribución de probabilidad del factor de seguridad. La presente publicación, tiene por objetivo diseñar un aplicativo en lenguaje Python para calcular el factor de seguridad de un talud, con mayor rapidez y confiabilidad que del modo manual tradicional, usando librerías, funciones y widgets propias de este lenguaje de programación. Una vez terminado y depurado, a través de una Interfaz Gráfica de Usuario (IGU), el estudiante realizará actividades de comparación entre los resultados del aplicativo y los obtenidos por un software comercial, explicando las diferencias si las hubiera, pudiendo modificar el escenario, variando los parámetros de entrada para que a través de la simulación se analice la influencia de los mismos en el comportamiento del factor de seguridad, reforzando con estas acciones el proceso de enseñanza aprendizaje sobre el tema.

## II. MÉTODOS

Para el presente estudio se ha considerado la estructura de un talud, con una altura de 182 m, un ángulo de cara de talud ( $\beta$ ) de  $50^\circ$  y de potencial superficie de falla ( $\alpha$ ) de  $40^\circ$ , con la cual se ha formado un bloque planar que puede llegar a deslizarse por efecto de la gravedad solamente, la cohesión de la superficie de falla ( $c$ ) es de 147 kPa, con un ángulo de fricción ( $\phi$ ) de  $35^\circ$ , y su peso específico de  $26 \text{ kN/m}^3$ ; se determinará el factor de seguridad ( $F_s$ ) del talud cuando el nivel freático discurre por debajo del pie del talud y su ancho es de 25m.

De acuerdo a Pariseau (2012) los deslizamientos de bloques planos son posibles cuando las “juntas” se introducen en una excavación superficial y la excavación es larga en relación con la altura. El criterio de resistencia de Mohr Coulomb (MC) es razonable para muchos taludes de roca y suelo. Con la suposición de MC, el factor de seguridad del bloque plano es:

$$F_s = \frac{N \tan(\phi) + C + T_b}{D}$$

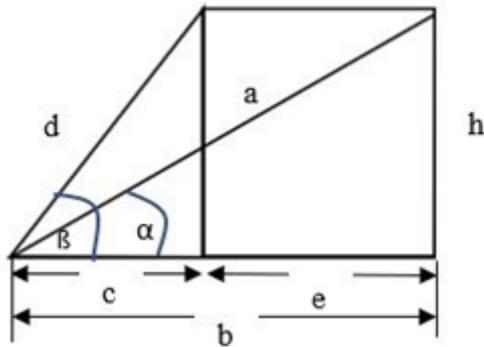
Donde ( $\phi$ ) es el ángulo de fricción interna del macizo rocoso.  $N$  es la fuerza normal,  $C$  fuerza cohesiva,  $T_b$  fuerza tangencial de refuerzo y  $D$  la fuerza de impulso. Para el presente caso no se ha considerado elementos de refuerzo para el talud ( $T_b$ ).

### 2.1 Desarrollo del procedimiento manual

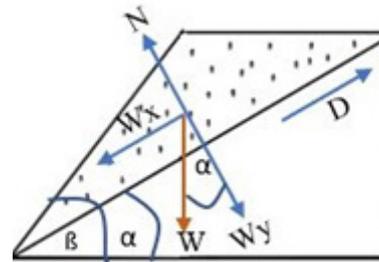
El talud es idealizado por tres figuras geométricas adyacentes, Figura 1; un rectángulo y dos triángulos rectángulos superpuestos, como se observa en la Figura 1.A, las fuerzas físicas y sus componentes que interactúan en un deslizamiento, para el caso presentado, se observa en la Figura 1.B.

**Figura 1**

A. Modelo idealizado del talud, mostrando sus elementos geométricos.



B. Modelo idealizado del talud, mostrando las fuerzas físicas y sus componentes que interactúan en un deslizamiento



A continuación, se describe el procedimiento de cálculo para la obtención del factor de seguridad del talud:

$$\text{sen}(40^\circ) = \frac{182\text{m}}{a} \rightarrow a = \frac{182\text{m}}{\text{sen}(40^\circ)} \rightarrow a = 283.14\text{m} \quad (1)$$

$$\text{tan}(40^\circ) = \frac{182\text{m}}{b} \rightarrow b = \frac{182\text{m}}{\text{tan}(40^\circ)} \rightarrow b = 216.90\text{m} \quad (2)$$

$$\text{tan}(50^\circ) = \frac{182\text{m}}{c} \rightarrow c = \frac{182\text{m}}{\text{tan}(50^\circ)} \rightarrow c = 152.72\text{m} \quad (3)$$

$$\text{sen}(50^\circ) = \frac{182\text{m}}{d} \rightarrow d = \frac{182\text{m}}{\text{sen}(50^\circ)} \rightarrow d = 237.58\text{m} \quad (4)$$

$$e = b - c = 216.90\text{m} - 152.72\text{m} = 64.18\text{m} \quad (5)$$

*Cálculo de las áreas*

$$A1(\text{rectángulo}) = (64.18\text{m})(182\text{m}) = 11680.76\text{m}^2 \quad (6)$$

$$A2(\text{triángulo } cdh) = \frac{(152.72\text{m})(182\text{m})}{2} = 13897.52\text{m}^2 \quad (7)$$

$$A3(\text{triángulo } abh) = \frac{(216.90\text{m})(182\text{m})}{2} = 19737.90\text{m}^2 \quad (8)$$

$$A4 = (A1 + A2) - A3 = (11680.76\text{m}^2 + 13897.52\text{m}^2) - (19737.90\text{m}^2) = 5840.38\text{m}^2 \quad (9)$$

*Cálculo del volumen*

$$V = (A4)(\text{Ancho de talud}) = (5840.38\text{m}^2)(25\text{m}) = 146009.5\text{m}^3 \quad (10)$$

*Cálculo del peso*

$$W = \gamma V = \left(\frac{26\text{kN}}{\text{m}^3}\right)(146009.5\text{m}^3) = 3796247\text{kN} \quad (11)$$

Cálculo de la fuerza normal

$$\sum Fy = 0 \quad (12)$$

$$N - Wy = 0 \rightarrow N = W \cos 40^\circ \rightarrow N = (3796247 \text{ kN}) (\cos 40^\circ) = 2908093.92 \text{ kN} \quad (13)$$

Cálculo de la fuerza de resistencia de oposición al deslizamiento (D)

$$\sum Fx = 0 \quad (14)$$

$$D - Wx = 0 \rightarrow D = W \text{sen} 40^\circ \rightarrow D = (3796247 \text{ kN}) \text{sen} 40^\circ = 2440180.53 \text{ kN} \quad (15)$$

Cálculo de la fuerza cohesiva (C)

$$C = (\text{cohesión})(a)(\text{ancho del talud}) = (147\text{kPa})(283.14\text{m})(25\text{m}) = 1040539.5\text{kN} \quad (16)$$

Cálculo del factor de seguridad (Fs)

$$F_s = \frac{N \tan(\varphi) + C}{D} = \frac{(2908093.92\text{kN}) \tan 35^\circ + 1040539.5\text{kN}}{2440180.53\text{kN}} = 1.26 \quad (17)$$

## 1.2 Desarrollo del aplicativo

El aplicativo se realizó con la versión Python 3.10, Python Software Foundation (2021), utilizando Visual Studio Code como editor de código fuente, Microsoft Visual Studio Code (s.f.). El aplicativo pretende solucionar un problema sobre estabilidad de taludes, en donde ingresando los parámetros manualmente desde el teclado, se calcularán las variables restantes internamente hasta llegar a obtener el factor de seguridad del talud, y graficar sobre una recta

la relación del factor de seguridad (Fs) para determinar su sensibilidad, en función de la altura recíproca del talud ( $H^{-1}$ ). El diagrama de flujo del aplicativo se observa en la Figura 2. En su desarrollo se hicieron uso de las siguientes librerías de Python:

*Tkinter*

Tkinter(Tk) es una herramienta para desarrollar aplicaciones de escritorio multiplataforma, con una interfaz gráfica para sistemas operativos Windows, Linux, Mac y otros. Tk es una librería de código abierto escrita en C y desarrollada en sus orígenes para el lenguaje de programación Tcl. Desde sus primeras versiones Python incluye en su biblioteca o librería estándar el módulo Tkinter, que permite interactuar con Tk para desarrollar aplicaciones de escritorio en Python. Es de fácil y rápido aprendizaje de modo que cualquier programador con una mínima base de Python puede comenzar rápidamente a crear aplicaciones gráficas profesionales, Python Assets (2021).

*Math*

Math es un módulo incorporado en la librería de Python 3 que proporciona constantes y funciones matemáticas estándar. Se puede usar el módulo Math para realizar varios cálculos matemáticos, como cálculos numéricos,

trigonométricos, logarítmicos y exponenciales, Lotfinejad (2022).

*Matplotlib*

Es una completa librería para crear visualizaciones estáticas, animadas e interactivas en Python. Matplotlib hace que las cosas fáciles sean fáciles y las difíciles sean posibles. La capacidad de esta librería radica en crear parcelas de calidad de publicación, diseñar figuras interactivas que puedan hacer zoom, desplazarse, actualizar; personalizar el estilo visual y el diseño, exportar los archivos a múltiples formatos, incrustar en JupyterLab e interfaces gráficas de usuario, utilizar una gran variedad de paquetes de terceros creados en Matplotlib, The Matplotlib development team (s.f.). El código de importación de estas librerías se puede observar en la Figura 3.

Una vez realizada la importación de las librerías, se procede a desarrollar y configurar el aplicativo incorporando todos los elementos necesarios como son las ventanas, variables, funciones y widgets correspondientes:

*Configuración de la ventana Tkinter*

Se define la configuración de la variable “ventana” igualando a la clase Tk, estableciendo sus dimensiones (geometry) y título (title). Ver Figura 4

**Figura 2**  
Diagrama de flujo del programa aplicativo

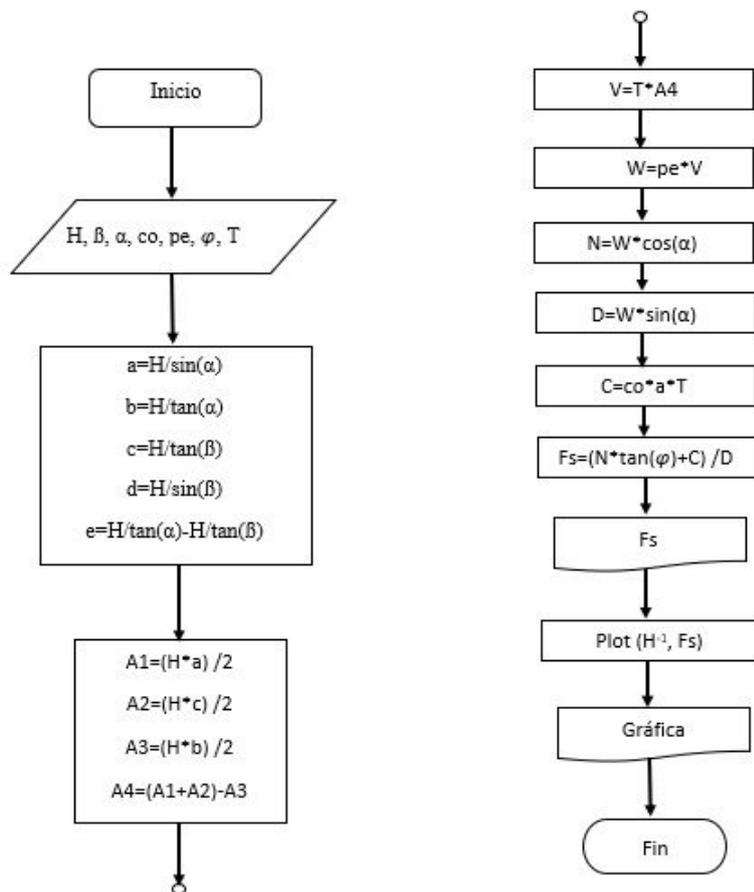


Figura 3

Código mostrando la importación de las librerías utilizadas en el presente programa aplicativo.

```

1  # Importamos las librerías a usar
2
3  from tkinter import*
4  import math
5  import matplotlib.pyplot as plt
6  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

```

Figura 4

Configuración de la ventana de Tkinter.

```

8  # Configuramos la ventana
9
10 ventana=Tk()
11 ventana.geometry("1100x650")
12 ventana.title("RZN 1.0")

```

#### *Configuración de los frame de resultados y gráfica*

El widget frame es muy importante para el proceso de agrupar y organizar otros widgets de una manera amigable. Funciona como un contenedor, que se encarga de organizar la posición de otros widgets. Utiliza áreas rectangulares en la pantalla para organizar el diseño y proporcionar relleno de estos widgets. Un frame también se puede utilizar como clase base para implementar widgets complejos, Tutorials Point India Private Limited (s.f.). Ver Figura 5.

#### *Declaración de variables*

Para declarar los parámetros que se ingresarán manualmente a través del teclado, se utilizarán las variables de control, objetos especiales asociados a los widgets, que nos permiten almacenar los valores, facilitándonos su disponibilidad en otras partes del programa, aquí se utilizó la variable de control DoubleVar, que declara una variable de tipo flotante, Suárez & Suárez (2016). Ver Figura 6.

#### *Introducción de los parámetros a través del teclado utilizando los widgets de Tkinter*

Para ello hacemos uso de las cajas de texto (Entry) widget que permite al usuario ingresar cualquier texto de una línea, Python Assets (2022); y para etiquetar utilizamos (Label), este widget implementa un cuadro de visualización donde se puede colocar texto o imágenes. El texto que muestra este widget se puede actualizar en cualquier momento. También es posible subrayar parte del texto (como para identificar un atajo de teclado) y dividir el texto en varias líneas, Tutorials Point India Private Limited (s.f.). Ver Figura 7.

#### *Cálculo de los resultados a través de la función def*

Esta quizás sea la parte más importante dentro de la estructura del programa, se emplea (def) que es la palabra clave para definir una función. El nombre de la función va seguido de los parámetros entre paréntesis. Los dos puntos

señalan el inicio del cuerpo de la función, que está marcado por una sangría, Robert Henderson Language Media Center (s.f.). Ver Figura 8.

#### *Configuración de la gráfica*

Para configurar nuestra gráfica hacemos uso de Matplotlib. pyplot, la cual es una colección de funciones que hacen que Matplotlib funcione como MATLAB. Cada función pyplot realiza algún cambio en una figura: por ejemplo, crea una figura, crea un área de trazado en una figura, traza algunas líneas en un área de trazado, decora el trazado con etiquetas, etc. En matplotlib. pyplot varios estados se conservan a través de las llamadas funciones, de modo que se realiza un seguimiento a la figura actual y al área de trazado, las funciones de trazado se dirigen a los ejes actuales, The Matplotlib development team (s.f.). Una vez configurada la gráfica en matplotlib hay que integrarla dentro de la Interfaz Gráfica de Usuario (IGU) de Tkinter y lo hacemos con la clase FigureCanvasTkAgg, un lienzo especial sobre el que se pueden trazar gráficos en Matplotlib y también se puede insertar en la (IGU) de Tkinter como un widget de Tkinter normal, Coders Legacy (2023). Ver Figura 9.

#### *Configuración de los botones de cálculo, de la gráfica y cierre del programa*

El widget Button se usa para agregar botones en una aplicación de Python. Estos botones pueden mostrar texto, como en este caso, o imágenes que transmiten el propósito de los botones; se puede enlazar una función o un método a un botón que se llama automáticamente haciendo clic sobre él, Tutorials Point India Private Limited (s.f.). Así mismo para el cierre del programa se utiliza el bucle infinito mainloop que se ejecuta para que siempre podamos ver la pantalla fija, y cuando cerramos la ventana, el ciclo termina o sale, este bucle se ubica en la última línea del código, Kumar (2021). Ver Figura 10.

**Figura 5.**

Configuración de los frame que alojarán los resultados y la gráfica

```

17 frame=Frame(ventana) #frame de resultados
18 frame.place(x=5,y=400)
19 frame.config(bg="white", width=560, height=210, relief="ridge",bd=5)
20
21 frame2=Frame(ventana) #frame de la gráfica
22 frame2.place(x=650,y=150)
23 frame2.config(bg="white", width=400, height=350, relief="ridge",bd=5)

```

**Figura 6**

Declaración de variables que serán utilizadas en el programa aplicativo

```

28 H=DoubleVar()
29 β=DoubleVar()
30 α=DoubleVar()
31 co=DoubleVar()
32 pe=DoubleVar()
33 φ=DoubleVar()
34 T=DoubleVar()

```

**Figura 7**

Código parcial del programa aplicativo mostrando el uso de los widgets Label y Entry en la introducción del parámetro "Altura de Talud"

```

57 label1=Label(ventana,text="Altura de Talud (m)",font=("Times_New_Roman",15))
58 label1.place(x=35,y=50)
59 entry1=Entry(ventana,textvariable=H,font=("Times_New_Roman",15),width=8)
60 entry1.place(x=470, y=50)

```

**Figura 8**

Código parcial del programa aplicativo mostrando el ingreso de fórmulas para el cálculo de los lados "a" y "b", dentro de la estructura de la función def

```

113 def talud (a,b,c,d,e,A1,A2,A3,A4,V,W,N,D,C):
114
115     w1=Label(ventana,text="",font=("Times_New_Roman",20))
116     a=round((H.get())/math.sin(math.radians(α.get()))),2)
117     w1.config(text="lado a "+str(a))
118
119     w2=Label(ventana,text="",font=("Times_New_Roman",20))
120     b=round((H.get())/math.tan(math.radians(β.get()))),2)
121     w2.config(text="lado b "+str(b))

```

Figura 9

Estructura de la función def para construir la gráfica del programa de aplicación.

```

205 def gráfica ():
206
207     fig=plt.Figure(figsize=(5,5),dpi=80)
208     ax=fig.add_subplot(111)
209     ax.set_xlim(0.000, 0.006)
210     ax.set_ylim(0.000, 4.000)
211
212     ax.plot([0.000,0.006],[m.get(),1.3],color='red')
213     ax.plot(pow(H.get(),-1),Fs.get(),color='black',marker='s')
214     ax.grid(True)
215     ax.set_box_aspect(1)
216
217     ax.set_xlabel('Altura recíproca del talud H^-1',color='black')
218     ax.set_ylabel('Factor de Seguridad',color='black')
219
220     ax.text(0.0002,0.55,'tan(φ)/tan(α)',fontsize=10,color='black')
221
222     canvas=FigureCanvasTkAgg(fig, master=frame2)
223     canvas.draw()
224     canvas.get_tk_widget().pack()

```

### III. RESULTADOS Y DISCUSIÓN

La Interfaz Gráfica de Usuario (IGU) del programa de aplicación desarrollado se puede apreciar en la Figura 11. Los resultados coinciden con los obtenidos en forma manual, el factor de seguridad es de 1.26, lo que indica que, si bien es cierto, es superior a la unidad, no está muy alejado de esta frontera; la gráfica muestra un sistema de ejes coordenados, siendo el eje de las abscisas el correspondiente a la altura recíproca del talud ( $H^{-1}$ ) y el de las ordenadas el factor de seguridad (Fs), el corte de la línea en su inicio, tiene por coordenadas para las abscisas el origen, y para las ordenadas la relación de las tangentes del ángulo de fricción interno del talud y del ángulo de potencial superficie de

deslizamiento del talud  $\left(\frac{\tan(\varphi)}{\tan(\alpha)}\right)$ , Pariseau (2012), en nuestro caso sería igual a  $\left(\frac{\tan(35^\circ)}{\tan(40^\circ)}\right)$ , equivalente a 0.8, por lo que este corte se origina en las coordenadas (0,0.8), la proyección de esta línea intercepta un punto generado por el factor de seguridad (Fs) en función de la altura recíproca del talud ( $H^{-1}$ ), siendo sus coordenadas (0.005, 1.26). Como se puede apreciar de la gráfica, se denota una sensibilidad del factor

de seguridad frente a la altura del talud, pues mientras mayor sea su altura, menor será este factor.

Dentro de la estructura interna de programación, hay que establecer rangos y delimitaciones que se ajusten al problema o a la situación específica, en nuestro caso para el eje de las ordenadas se estableció un rango de 0.0 a 4.0 y en el eje de las abscisas un rango de 0.0 a 0.006, lo que imposibilita graficar condiciones por encima de un factor de seguridad de 4.0 y alturas de talud por debajo de los 165 m.

El factor de seguridad (Fs) es obtenido automáticamente por el aplicativo y para no tener la necesidad de ingresarlo nuevamente como coordenada en la gráfica, lo definimos como una variable global `Fs.set(round((N*(math.tan(math.radians(φ.get())))+C)/(D),2))`. Lo mismo sucede en la relación de las tangentes del ángulo de fricción interna ( $\varphi$ ) y el ángulo de potencial superficie de deslizamiento ( $\alpha$ ), `m.set(round(math.tan(math.radians(φ.get()))/(math.tan(math.radians(α.get()))),2)`. La función `round` utilizada nos ha permitido redondear los valores decimales de los resultados, en este caso a dos cifras decimales. (Figura 12)

Figura 10

Código mostrando la configuración del widget Button para el cálculo y la gráfica, así como el cierre de programa con el bucle `mainloop`.

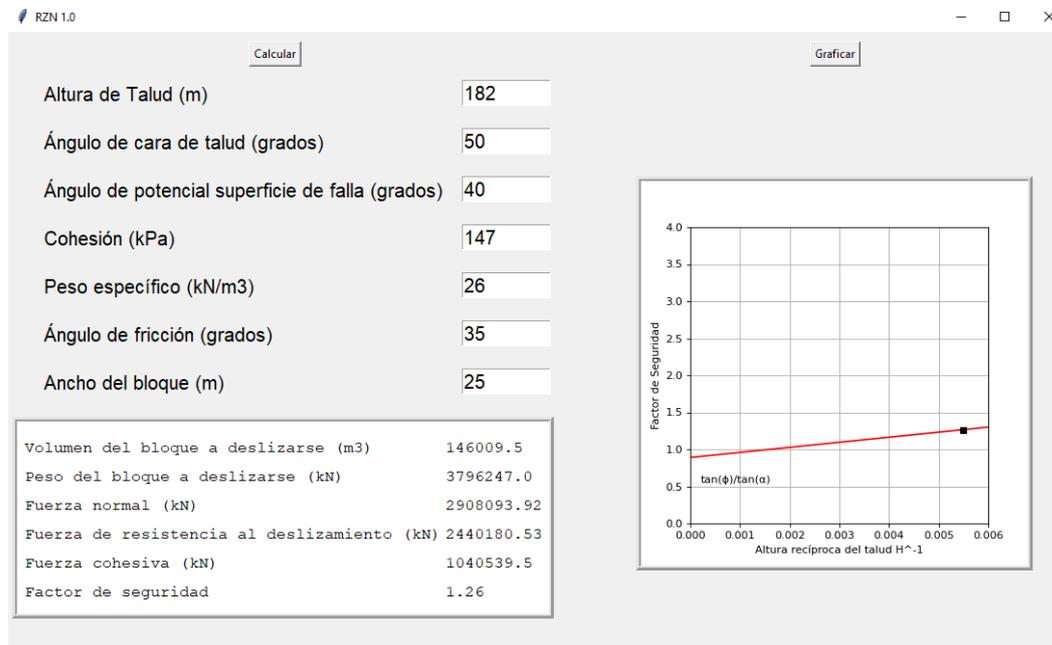
```

226 #Configuración del botón de cálculo
227 btn=Button(ventana, text="Calcular", command=lambda: (talud(a.get(), b.get(), c.get(), d.get(), e.get(), A1.get(), A2.get(),
228 | A3.get(), A4.get(), V.get(), W.get(), N.get(), D.get(), C.get()))
229 btn.place(x=250, y=10)
230
231 #Configuración del botón de la gráfica
232 plop_button=Button(master=ventana, command=gráfica, text="Graficar")
233 plop_button.place(x=830, y=10)
234
235 ventana.mainloop()

```

**Figura 11**

Interfaz Gráfica de Usuario (IGU) del programa aplicativo desarrollado, mostrando los resultados, así como la gráfica respectiva con su sistema de ejes coordenados



**Figura 12**

Código parcial que muestra la configuración de las variables globales Fs y m, a ser utilizadas también en la gráfica del aplicativo.

```

190 # Cálculo del factor de seguridad
191
192     W15=Label(ventana,text="",font=("Courier",12),fg="black")
193     Fs.set(round((N*(math.tan(math.radians(phi.get())))+C)/(D), 2))
194     W15.config(text="" +str(Fs.get()), bg="white")
195     W15.place(x=450,y=570)
196
197 # Relación de tangentes
198
199     W16=Label(ventana,text="",font=("Courier",12),fg="black")
200     m.set(round(math.tan(math.radians(phi.get()))/(math.tan(math.radians(alpha.get()))),2))
201     W16.config(text="" +str(m.get()),bg="white")
    
```

#### IV. CONCLUSIONES

El aprendizaje de un lenguaje de programación de alto nivel como Python es importante para la solución rápida y confiable del cálculo del factor de seguridad de un problema de estabilidad de taludes que, por métodos tradicionales, tediosos y laboriosos, demandaría mucho tiempo; permitiendo realizar comparaciones y simulaciones de diferentes escenarios, modificando los parámetros de entrada, determinando y explicando la influencia de los mismos en la estabilidad.

Con la práctica constante el estudiante, puede llegar a diseñar aplicativos de gran calidad, con alta capacidad de

procesamiento de información y mucha más funcionalidad, incluso superando las bondades y características de los softwares comerciales, sin verse en la necesidad de adquirir licencias de compra que aún son onerosas, o de instalarlas haciendo uso de métodos vedados como el cracking informático, u obteniendo versiones parciales que no entregan todas las características completas del software.

#### V. AGRADECIMIENTOS

Al Instituto de Investigación de la Facultad de Ingeniería Geológica, Minera, Metalúrgica y Geográfica de la

Universidad Nacional Mayor de San Marcos por permitir la publicación del presente artículo.

## VI. REFERENCIAS

Coders Legacy. (2023). FigureCanvasTkAgg with Matplotlib and Tkinter. <https://coderslegacy.com/figurecanvastkagg-matplotlib-tkinter/>

De Puelles, M. (1999). Formación Docente. Revista Iberoamericana de Educación 19, 9-11. <https://rieoei.org/RIE/article/view/1053/2006>

González de Vallejo, L. I., Ferrer, M., Ortuño, L., & Oteo, C. (2004). Ingeniería Geológica. Madrid: Pearson Educación S.A. <https://www.pearsoneducacion.net/espa%C3%B1a/TiendaOnline/ingenieria-geologica>

Kumar, B. (2021). PythonGuides.com. <https://pythonguides.com/python-tkinter-mainloop/>

Lotfinejad, M. (2022). Python Math Module Guide (22 Examples and 18 Functions). <https://www.dataquest.io/blog/python-math-module-and-functions/#:~:text=math%20is%20a%20built%20in,%2C%20logarithmic%2C%20and%20exponential%20calculations.>

Microsoft Visual Studio Code. (s.f.). Getting Started. <https://code.visualstudio.com/docs>

Pariseau, W. G. (2012). Design Analysis in Rock Mechanics. London, UK: CRC Press. <https://doi.org/10.1201/b11461>

Peña, R. (2016). Resolución de problemas para ingenieros con Python estructurado. Madrid: Ibergarceta Publicaciones, S.L. <https://www.iberlibro.com/9788416228713/Resoluci%C3%B3n-problemas-ingenieros-Python-estructurado-841622871X/plp>

Python Assets. (2021). Introducción a Tcl/Tk (tkinter). <https://recursospython.com/guias-y-manuales/introduccion-a-tkinter/>

Python Assets. (2022). Caja de texto (Entry) en Tcl/Tk (tkinter). <https://recursospython.com/guias-y-manuales/caja-de-texto-entry-tkinter/>

Python Software Foundation. (2021). Python 3.10.0. <https://www.python.org/downloads/release/python-3100/>

Robert Henderson Language Media Center. (s.f.). User- Defined Functions. [https://sites.pitt.edu/~naraehan/python3/user\\_defined\\_functions.html#:~:text=def%20is%20the%20keyword%20for,the%20value%20to%20be%20returned.](https://sites.pitt.edu/~naraehan/python3/user_defined_functions.html#:~:text=def%20is%20the%20keyword%20for,the%20value%20to%20be%20returned.)

Suárez, A., & Suárez, A. (2016). Variables de control en Tkinter. <https://python-para-impacientes.blogspot.com/2016/02/variables-de-control-en-tkinter.html>

The Matplotlib development team. (s.f.). Matplotlib: Visualization with Python. <https://matplotlib.org/>

The Matplotlib development team. (s.f.). Pyplot tutorial. <https://matplotlib.org/stable/tutorials/introductory/pyplot.html>

Tumialán De La Cruz, P. H. (2004). La geología en relación al sistema ecológico en el Perú. Revista Del Instituto De investigación De La Facultad De Minas, Metalurgia Y

Ciencias geográficas,7(13),9-15. <https://doi.org/10.15381/iigeo.v7i13.510>

Tutorials Point India Private Limited. (s.f.). Python-Tkinter Button. [https://www.tutorialspoint.com/python/tk\\_button.htm](https://www.tutorialspoint.com/python/tk_button.htm)

Tutorials Point India Private Limited. (s.f.). Python-Tkinter Frame. [https://www.tutorialspoint.com/python/tk\\_frame.htm](https://www.tutorialspoint.com/python/tk_frame.htm)

Tutorials Point India Private Limited. (s.f.). Python-Tkinter Label. [https://www.tutorialspoint.com/python/tk\\_label.htm](https://www.tutorialspoint.com/python/tk_label.htm)

### Contribución de autoría:

Conceptualización: R.Z, Curación de datos: R.Z, Análisis formal: R.Z, Adquisición de fondos: R.Z, Investigación: R.Z & Y.M; Metodología: R.Z & Y.M, Administración del proyecto: R.Z, Recursos: R.Z, Software: R.Z, Supervisión: R.Z & Y.M; Validación: R.Z & Y.M; Visualización: R.Z, Redacción-borrador original: R.Z & Y.M; Redacción-Revisión y edición: R.Z & Y.M.

### Conflicto de intereses:

Los autores declaran no tener conflictos de intereses