

Andy Asael Sánchez Jiménez

201904024

Laboratorio Introducción a la Programación y Computación 2 sección N

## Explicación de Actividad Final: Totito

Se utilizó la librería Tkinter la cual sirve para hacer aplicaciones con interfaz gráfica en Python.

```
from tkinter import *
from tkinter import messagebox

root = Tk()
root.title('Totito')
click = True
count = 0
```

En esta parte se importaron las librerías Tkinter, después se implemento la aplicación grafica para Tk y se le coloco de titulo a la ventana como "Totito".

```
def getganador():
    global ganador
    ganador = False

    if b1["text"] == "X" and b2["text"] == "X" and b3["text"] == "X":
        b1.config(bg="red")
        b2.config(bg="red")
        b3.config(bg="red")
        ganador = True
        messagebox.showinfo("Totito", "¡¡¡ Gana!!!")
        disable_all_buttons()
    elif b4["text"] == "X" and b5["text"] == "X" and b6["text"] == "X":
        b4.config(bg="red")
        b5.config(bg="red")
        b6.config(bg="red")
        ganador = True
        messagebox.showinfo("Totito", "¡¡¡ Gana!!!")
        disable_all_buttons()
    elif b7["text"] == "X" and b8["text"] == "X" and b9["text"] == "X":
        b7.config(bg="red")
        b8.config(bg="red")
        b9.config(bg="red")
        ganador = True
        messagebox.showinfo("Totito", "¡¡¡ Gana!!!")
        disable_all_buttons()
```

En la función siguiente, llamada getganador, se estableció una variable True o False llamada ganador, para establecer si el j1 ganaba o el j2.

Se estableció la variable ganadora por defecto en False.

Luego se hicieron las condiciones para ganar, se validaron los botones en horizontal, vertical e inclinado, para cada X y O de todas las formas posibles para ganar.

Si el jugador gana, cambia el color de la línea ganadora en azul con "bn.config" (n es el numero del botón, en los que en total hay 9). Luego manda un messageBox anunciando que el jugador ganó.

En total se hicieron 8 condiciones para cada jugador, siendo un total de 9 "elifs".

```
def b_click(b):
    global click, count

    if b["text"] == " " and click == True:
        b["text"] = "X"
        click = False
        count += 1
        getganador()
    elif b["text"] == " " and click == False:
        b["text"] = "O"
        click = True
        count += 1
        getganador()
```

La función llamada b\_click funciona para validar cada "click" que se le hacía al botón determinado, se determinaron 2 variables globales. La variable count funciona para los turnos del jugador, para que pueda alternarse el X y el O. La variable click es de tipo booleana para determinar si el botón estaba ocupado o no, para después llamar a la función getGanador.

```

def crear():
    global b1, b2, b3, b4, b5, b6, b7, b8, b9
    global click, count
    click = True
    count = 0

    b1 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b1))
    b2 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b2))
    b3 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b3))

    b4 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b4))
    b5 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b5))
    b6 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b6))

    b7 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b7))
    b8 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b8))
    b9 = Button(root, text=" ", height=3, width=6, command=lambda: b_click(b9))

    b1.grid(row=0, column=0)
    b2.grid(row=0, column=1)
    b3.grid(row=0, column=2)

    b4.grid(row=1, column=0)
    b5.grid(row=1, column=1)
    b6.grid(row=1, column=2)

    b7.grid(row=2, column=0)
    b8.grid(row=2, column=1)
    b9.grid(row=2, column=2)

crear()
root.mainloop()

```

Y finalmente la última función llamada crear, sirve para crear los botones, y establecerlos en orden gracias al comando “grid” que los ordena como una matriz con filas y columnas. Igualmente se establecieron las variables globales click y count para contar los turnos y validar los turnos.

Después simplemente se llama la función crear() para que se ejecute al principio y root.mainloop() para que no finalice la aplicación.