

Código (Juego del gato en Python)

```
from tkinter import *
```

```
from tkinter import messagebox
```

```
//Considero que la lectura de bibliotecas es una llamada al sistema ya que estas se encuentran en archivos fuera del programa de usuario
```

```
jugador = 1
```

```
buttons = []
```

```
puntuaciones = [{"Jugador uno", 0}, {"Jugador dos", 0}]
```

```
selec_uno = []
```

```
selec_dos = []
```

```
class Aplicacion(Frame):
```

```
    def __init__(self, master=None):
```

```
        super().__init__(master)
```

```
        self.master = master
```

```
        self.pack()
```

```
        self.generar_botones()
```

```
        self.generar_marcador()
```

```
    def generar_botones(self):
```

```
        for _ in range(9):
```

```
            newbutton = Button(self, bg="white", height=10, width=20,  
command = lambda name=_: self.seleccion_casilla(name))
```

```
            if _ <= 2:
```

```
                newbutton.grid(row=0, column=_)
```

```
            elif 2 < _ <= 5:
```

```
                newbutton.grid(row=1, column=_%3)
```

```
            elif 5 < _ <= 8:
```

```
        newbutton.grid(row=2, column=_%6)
    buttons.append(newbutton)
```

```
def generar_marcador(self):
```

```
    texto_uno = "Jugador 1: {}".format(puntuaciones[0][1])
```

```
    texto_dos = "Jugador 2: {}".format(puntuaciones[1][1])
```

```
    marcador_uno = Label(self, text = texto_uno)
```

```
    marcador_dos = Label(self, text = texto_dos)
```

```
    marcador_uno.grid(row=3, column=0)
```

```
    marcador_dos.grid(row=3, column=2)
```

```
def seleccion_casilla(self, indc):
```

```
    global jugador
```

```
    if jugador == 1:
```

```
        if casilla_libre(indc):
```

```
            selec_uno.append(indc)
```

```
            buttons[indc]["bg"] = ["red"]
```

```
            jugador -= 1
```

```
            verificar_tablero(selec_uno, puntuaciones[0])
```

```
    else:
```

```
        if casilla_libre(indc):
```

```
            selec_dos.append(indc)
```

```
            buttons[indc]["bg"] = ["blue"]
```

```
            jugador += 1
```

```
            verificar_tablero(selec_dos, puntuaciones[1])
```

```
def verificar_tablero(casillas, jugador):
```

```
    global selec_uno
```

```
    global selec_dos
```

```
    combinaciones = [
```

```
        (0,1,2),
```

```
        (3,4,5),
```

```
        (6,7,8),
```

```
        (0,3,6),
```

```
        (1,4,7),
```

```
        (2,5,8),
```

```
        (0,4,8),
```

```
        (2,4,6)
```

```
    ]
```

```
    if len(casillas) >= 3:
```

```
        for tupla in combinaciones:
```

```
            victoria = 0
```

```
            for num in casillas:
```

```
                if num in tupla:
```

```
                    victoria += 1
```

```
                    if victoria == 3:
```

```
                        messagebox.showinfo("Victoria", "Ronda  
finalizada: el {} ha ganado la ronda".format(jugador[0]))
```

```
//Considero que los mensajes son llamadas al sistema ya que estos requieren acceso a  
nuevos procesos
```

```
        limpiar_tablero()
```

```
        jugador[1] += 1
```

```
        selec_uno = []
```

```
        selec_dos = []
        app.generar_marcador()
        break

    if len(selec_uno+selec_dos) == 9:
        messagebox.showinfo("Empate", "Ronda finalizada: se ha tenido un
empate")
        limpiar_tablero()
        selec_uno = []
        selec_dos = []

def limpiar_tablero():
    for i in range(9):
        buttons[i]["bg"] = ["white"]

def casilla_libre(indc):
    if indc in (selec_uno+selec_dos):
        return False
    else:
        return True

root = Tk()
app = Aplicacion(master=root)
root.iconbitmap(r'Imagenes/Tic-Tac-Toe-Game.ico')
//El acceso a las imágenes también son llamadas al sistema ya que requieren de la
lectura de archivos y de igual manera, la escritura de estos para mostrarlos en
pantalla
root.resizable(0,0)
```

//Re-dimensionar la ventana gráfica y mantenerla fija también es una llamada al sistema debido a que prohíbe al mismo usuario la modificación de esta

```
root.title("Tic Tac Toe")
```

//Modificar el título de la pestaña requiere de llamadas al sistema para comunicar al sistema operativo el texto que deberá colocar en el programa al ejecutarlo

```
app.mainloop()
```

//El mainloop de una ventana gráfica es una llamada al sistema ya que requiere que se mantenga en ejecución el proceso y estos vayan cambiando conforme se modifique la ejecución del programa.