



## **PRACTICO – MINI PROYECTOS - EJERCICIO**

### **“TARANTINO”**

#### **INTEGRANTES**

Alexander Yanarico Quispe

José Andres Salame Lijeron

Jeremy Said Pardo Camacho

Erwin David Pardo Hernandez

Pablo Jesús Fernández Jaime

Cristhian David Puña Salto

Hugo Clementelli Castedo

#### **DOCENTE**

Ing. Jimmy Nataniel Requena Llorentty

#### **ASIGNATURA**

Programación II

## INDICE

1. Tres en rayas.....	3
2. Batalla naval.....	4
3. Laberinto.....	6
4. Sala cine v1.....	7
5. Sala cine por categorías de edad.....	8

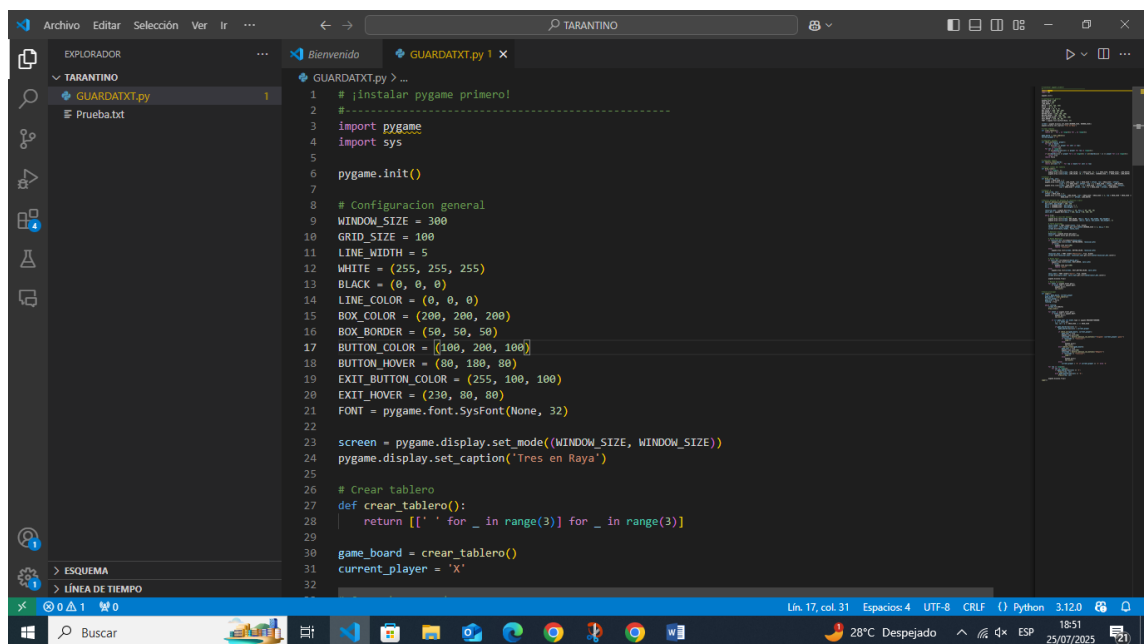
## 1. TRES EN RAYAS

### DESCRIPCIÓN GENERAL

#### Requisitos

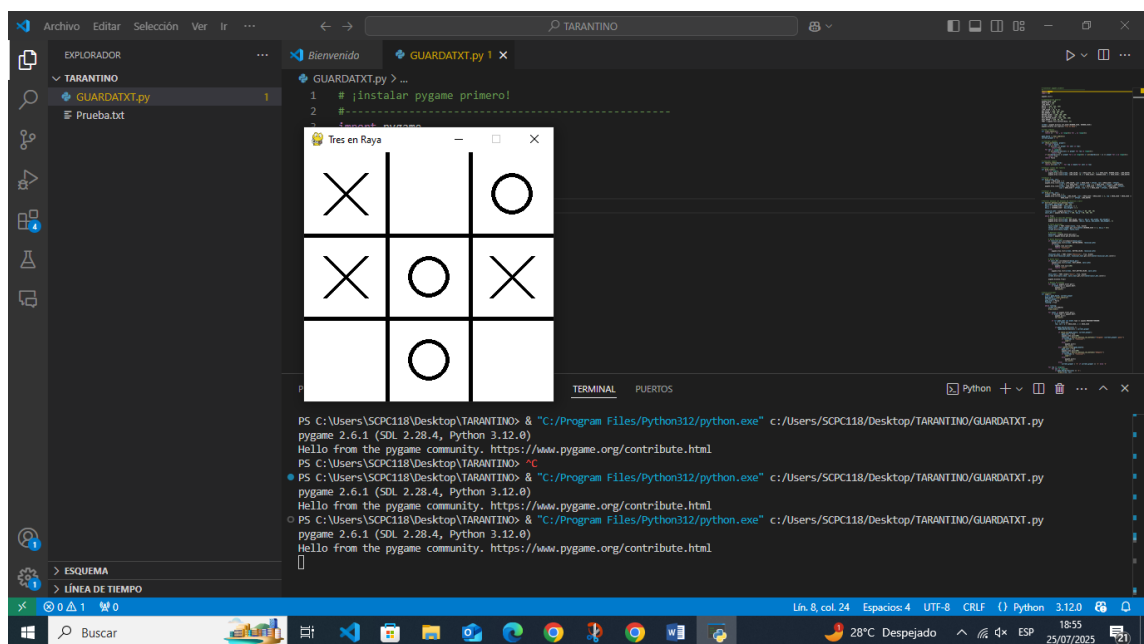
- Tener pygame instalado

Este programa implementa el clásico juego **Tres en Raya** (Tic-Tac-Toe) utilizando la biblioteca pygame. Dos jugadores (X y O) se turnan para marcar las casillas de una cuadrícula de 3x3. El juego detecta automáticamente si hay un ganador o un empate y ofrece botones para reiniciar o salir.



```
1 # instalar pygame primero!
2 #-----
3 import pygame
4 import sys
5
6 pygame.init()
7
8 # Configuración general
9 WINDOW_SIZE = 300
10 GRID_SIZE = 100
11 LINE_WIDTH = 5
12 WHITE = (255, 255, 255)
13 BLACK = (0, 0, 0)
14 LINE_COLOR = (0, 0, 0)
15 BOX_COLOR = (200, 200, 200)
16 BOX_BORDER = (50, 50, 50)
17 BUTTON_COLOR = (100, 200, 100)
18 BUTTON_HOVER = (80, 180, 80)
19 EXIT_BUTTON_COLOR = (255, 100, 100)
20 EXIT_HOVER = (230, 80, 80)
21 FONT = pygame.font.SysFont(None, 32)
22
23 screen = pygame.display.set_mode(WINDOW_SIZE, WINDOW_SIZE)
24 pygame.display.set_caption("Tres en Raya")
25
26 # Crear tablero
27 def crear_tablero():
28     return [[' ' for _ in range(3)] for _ in range(3)]
29
30 game_board = crear_tablero()
31 current_player = 'X'
32
```

Visualmente así funciona el juego:



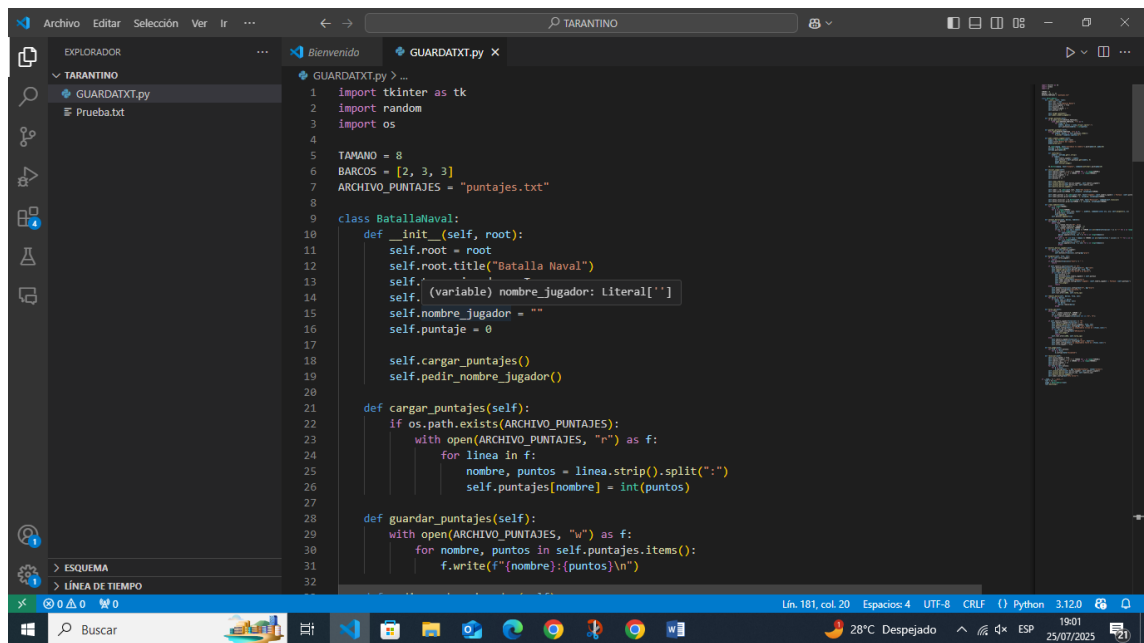
## LINK DEL CODE

<https://github.com/joseandres2003/TARANTINO/blob/main/PROYECTOS/3enraya.py>

## 2. BATALLA NAVAL

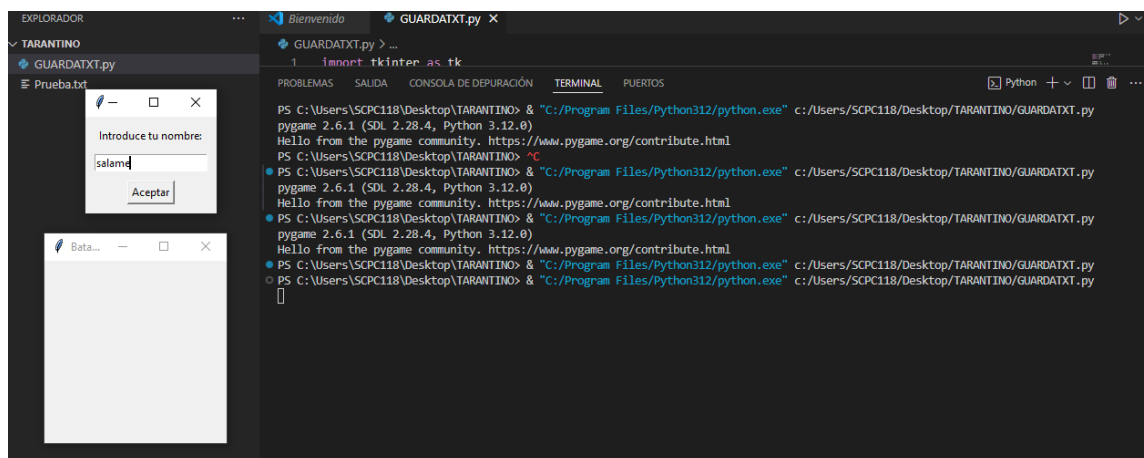
### DESCRIPCIÓN GENERAL

Este programa implementa una versión simplificada del juego **Batalla Naval** (Battleship) usando la biblioteca gráfica tkinter. El jugador se enfrenta contra la computadora en un tablero de 8x8. Ambos colocan barcos al azar y se turnan para disparar hasta que uno de los dos destruya todos los barcos enemigos.

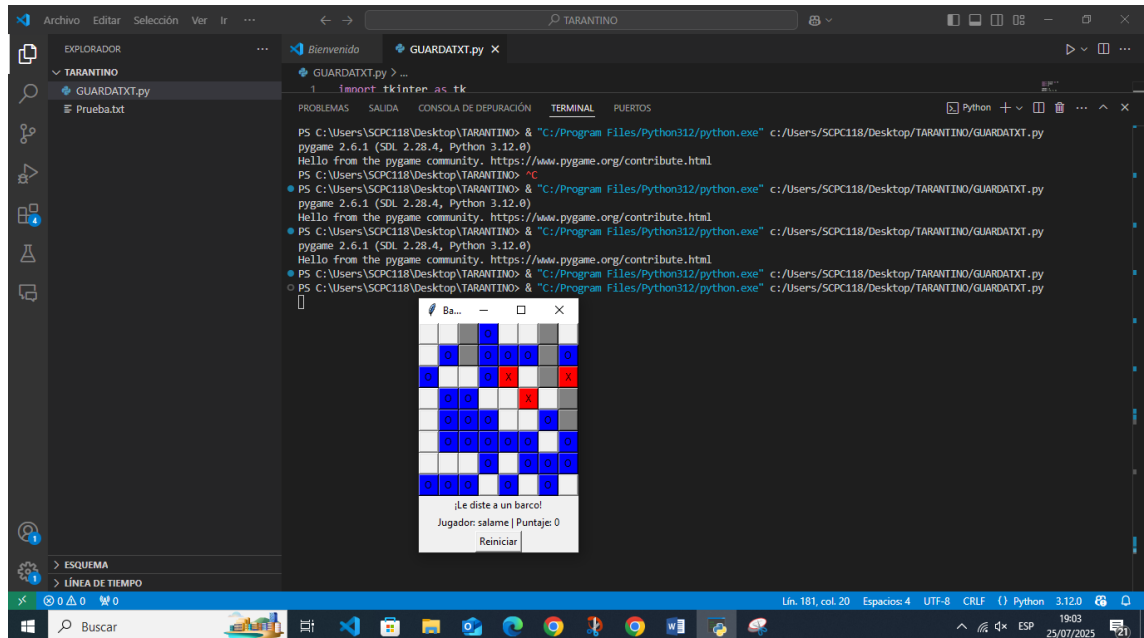


Visualmente así funciona el juego:

Donde primero pide nombre de usuario



Luego se ejecuta el juego



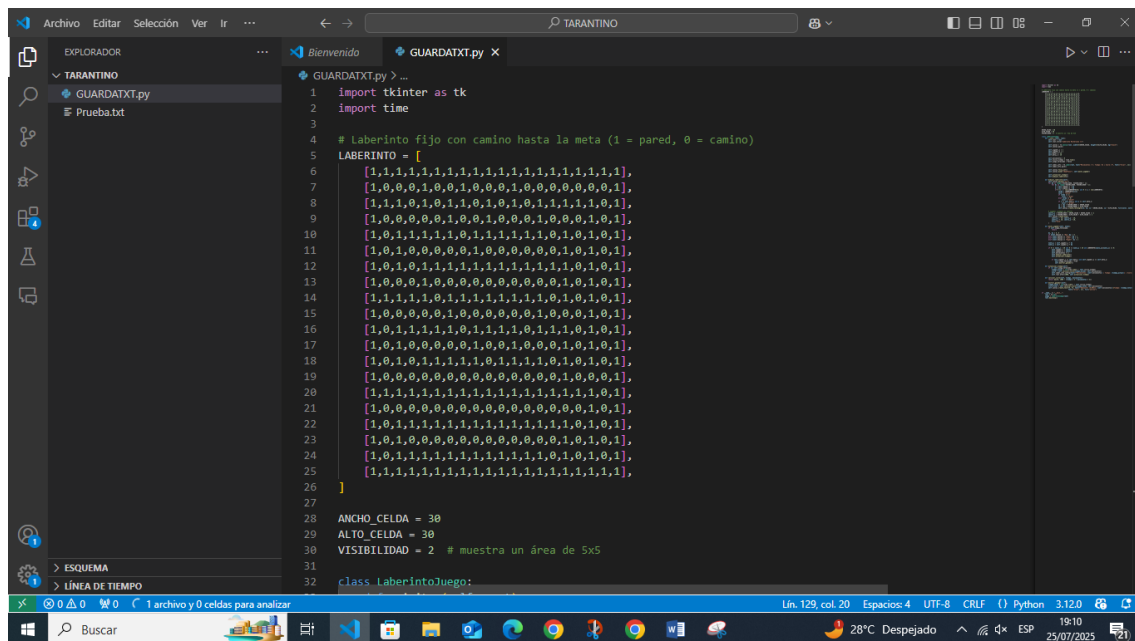
LINK DE CODE

<https://github.com/joseandres2003/TARANTINO/blob/main/PROYECTOS/batalla-naval.py>

### 3. LABERTINTO

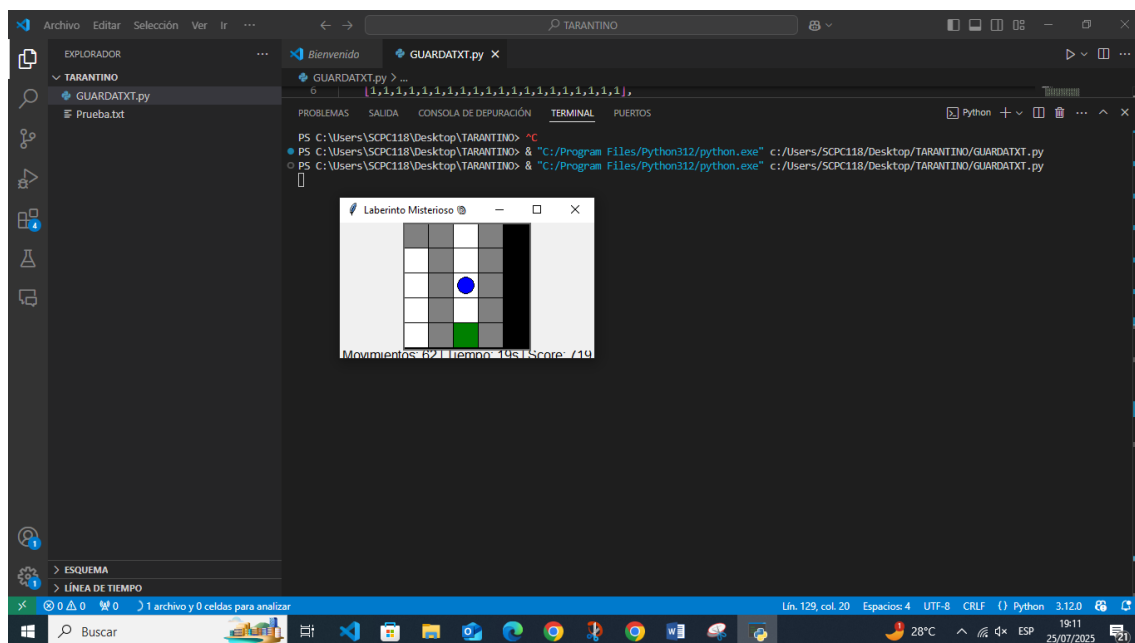
#### DESCRIPCIÓN GENERAL

Este proyecto implementa un juego de exploración en un laberinto utilizando tkinter. El jugador se mueve en un mapa limitado por la **visión**, buscando llegar a la **meta** lo más rápido posible y con el menor número de movimientos. El puntaje se calcula en función del tiempo y los pasos realizados.



```
1 import tkinter as tk
2 import time
3
4 # Laberinto fijo con camino hasta la meta (1 = pared, 0 = camino)
5 LABERINTO = [
6     [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
7     [1,0,0,0,1,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1],
8     [1,1,1,0,1,0,1,1,0,1,0,1,0,1,1,1,1,1,0,1],
9     [1,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,1,0,0,1],
10    [1,0,1,1,1,1,0,1,1,1,1,1,1,0,1,0,1,0,1,1],
11    [1,0,1,0,0,0,0,1,0,0,0,0,0,0,1,0,1,0,1,1],
12    [1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,0,1,1],
13    [1,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1],
14    [1,1,1,1,0,1,1,1,1,1,1,1,1,1,0,1,0,1,0,1],
15    [1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,1,1],
16    [1,0,1,1,1,1,0,1,1,1,1,1,1,0,1,1,1,0,1,1],
17    [1,0,1,0,0,0,0,1,0,0,1,0,0,0,1,0,1,0,1,1],
18    [1,0,1,0,1,1,1,1,1,0,1,1,1,1,0,1,0,1,0,1],
19    [1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1],
20    [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1],
21    [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1],
22    [1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1],
23    [1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1],
24    [1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1,0,1,0,1],
25    [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
26 ]
27
28 ANCHO_CELDA = 30
29 ALTO_CELDA = 30
30 VISIBILIDAD = 2 # muestra un área de 5x5
31
32 class LaberintoJuego:
```

El juego se ejecuta de la siguiente manera:



Donde te marca los movimientos y el tiempo.

LINK DE CODE

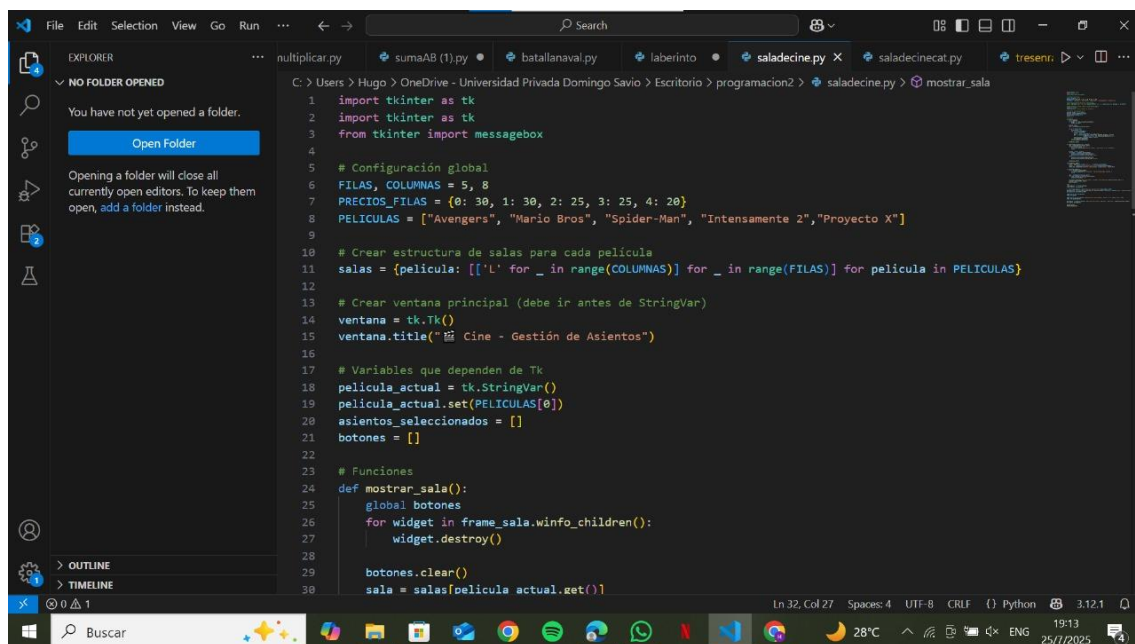
<https://github.com/joseandres2003/TARANTINO/blob/main/PROYECTOS/laberinto.py>

## 4. SALA DE CINE

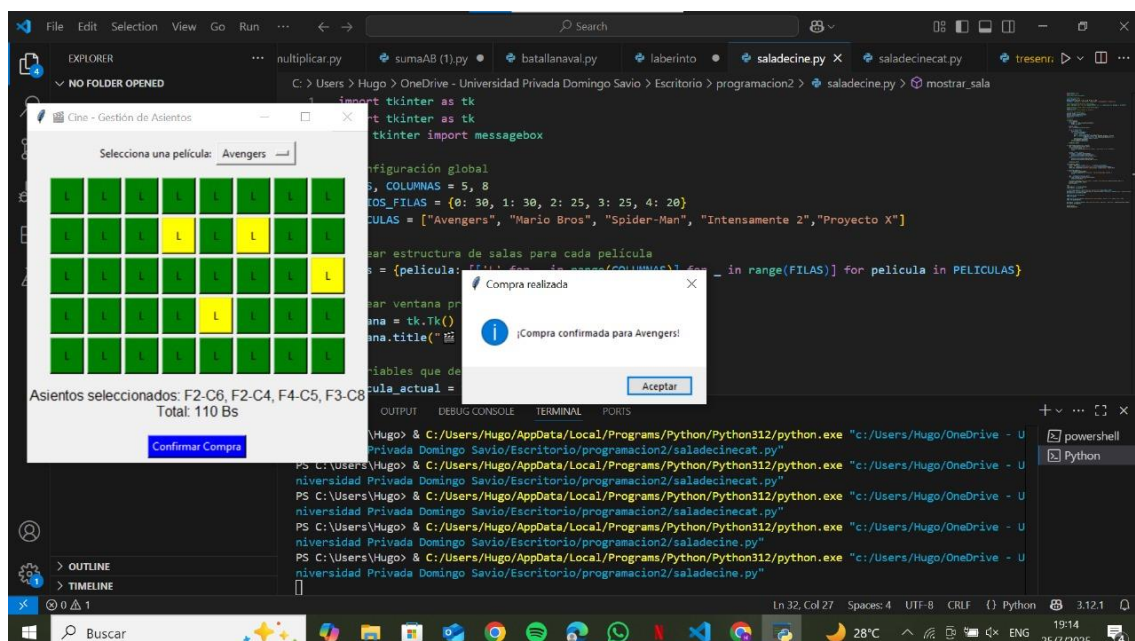
### DESCRIPCIÓN GENERAL

Este programa es una aplicación GUI (interfaz gráfica) creada con **Tkinter** que simula la compra de entradas en una sala de cine. Permite:

- Seleccionar una película.
- Visualizar los asientos disponibles.
- Elegir múltiples asientos.
- Ver el total a pagar según la fila.
- Confirmar la compra y marcar los asientos como ocupados.



El juego se ejecuta de la siguiente manera:



LINK DE CODE

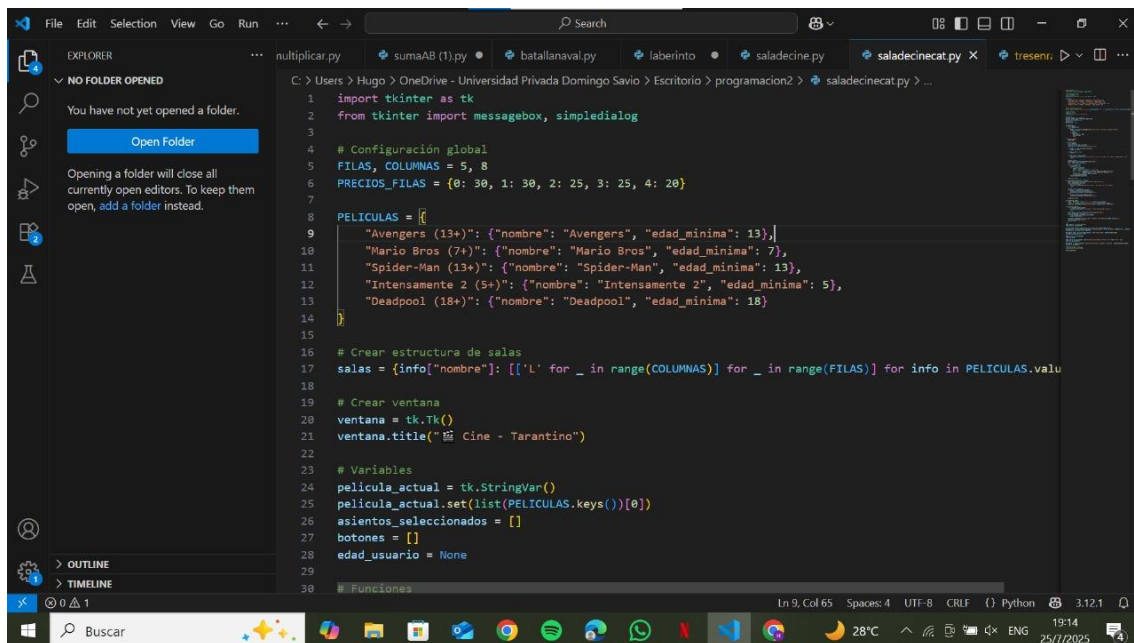
<https://github.com/joseandres2003/TARANTINO/blob/main/PROYECTOS/sala-cine.py>

## 5. SALA CINE POR CATEGORIA DE EDAD

### DESCRIPCIÓN GENERAL

Este proyecto es una aplicación gráfica desarrollada con **Tkinter** que simula la compra de entradas para un cine. El sistema permite:

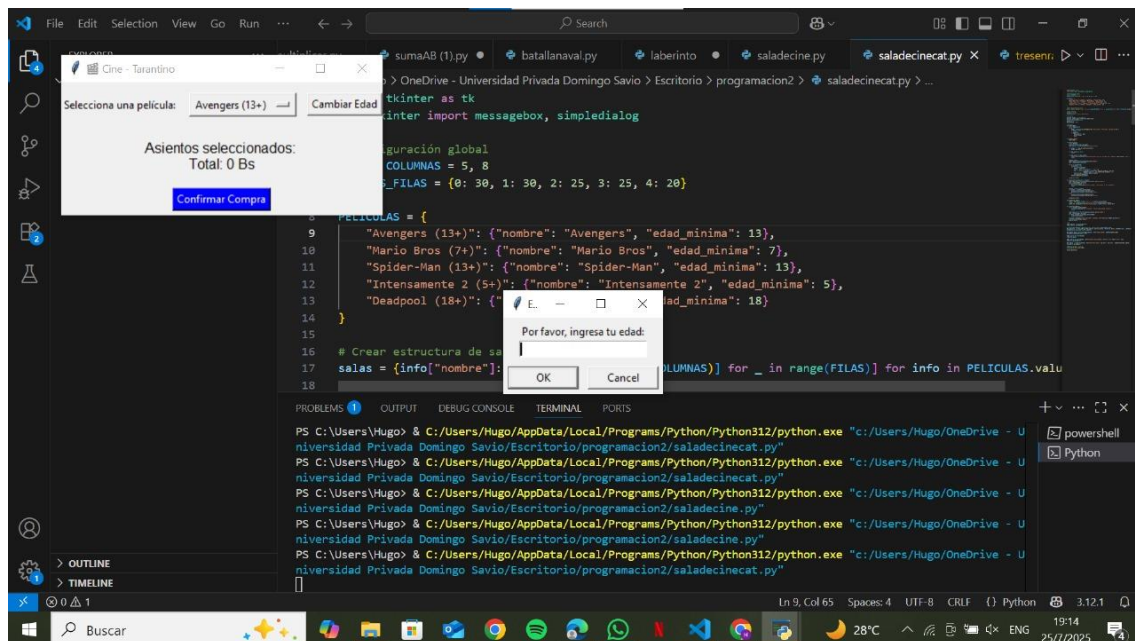
- Seleccionar películas con restricción de edad.
- Ver y seleccionar asientos disponibles.
- Calcular el costo total en base a la fila del asiento.
- Confirmar la compra y marcar asientos como ocupados.
- Cambiar la edad del usuario en cualquier momento.



```
1 import tkinter as tk
2 from tkinter import messagebox, simpledialog
3
4 # Configuración global
5 FILAS, COLUMNAS = 5, 8
6 PRECIOS_FILAS = {0: 30, 1: 30, 2: 25, 3: 25, 4: 20}
7
8 PELICULAS = [
9     "Avengers (13+): {"nombre": "Avengers", "edad_minima": 13},
10    "Mario Bros (7+): {"nombre": "Mario Bros", "edad_minima": 7},
11    "Spider-Man (13+): {"nombre": "Spider-Man", "edad_minima": 13},
12    "Intensamente 2 (5+): {"nombre": "Intensamente 2", "edad_minima": 5},
13    "Deadpool (18+): {"nombre": "Deadpool", "edad_minima": 18}
14 ]
15
16 # Crear estructura de salas
17 salas = {info["nombre"]: [{"L" for _ in range(COLUMNAS)] for _ in range(FILAS)] for info in PELICULAS.values()}
18
19 # Crear ventana
20 ventana = tk.Tk()
21 ventana.title("Cine - Tarantino")
22
23 # Variables
24 pelicula_actual = tk.StringVar()
25 pelicula_actual.set(list(PELICULAS.keys())[0])
26 asientos_seleccionados = []
27 botones = []
28 edad_usuario = None
29
30 # Funciones
```

El juego ejecutado funciona así





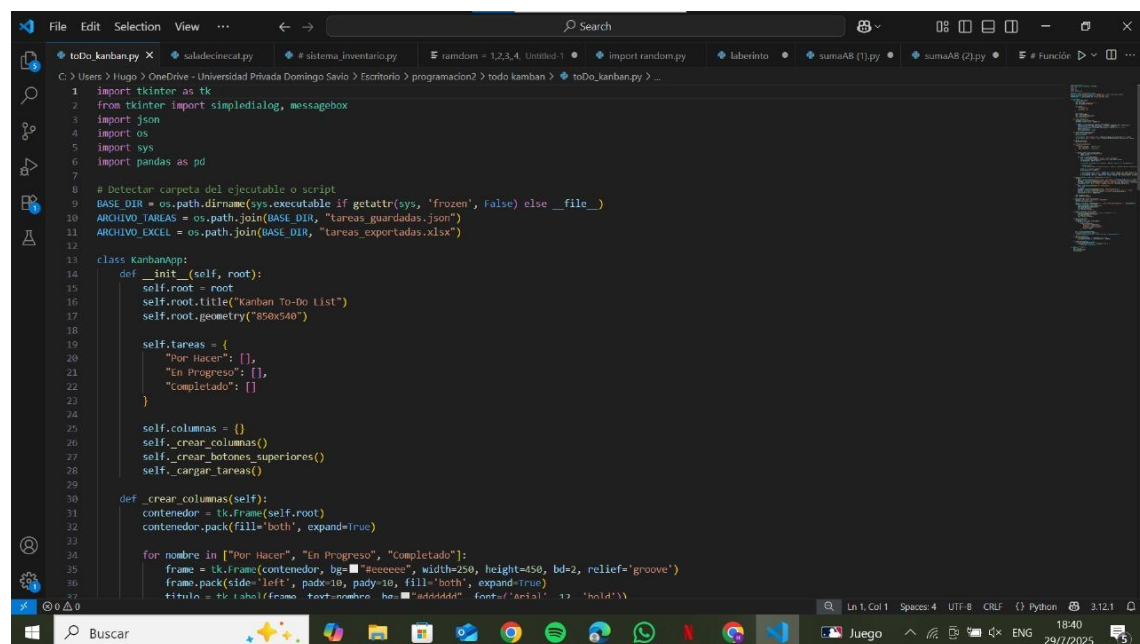
LINK DE CODE

<https://github.com/joseandres2003/TARANTINO/blob/main/PROYECTOS/salacine-cat.py>

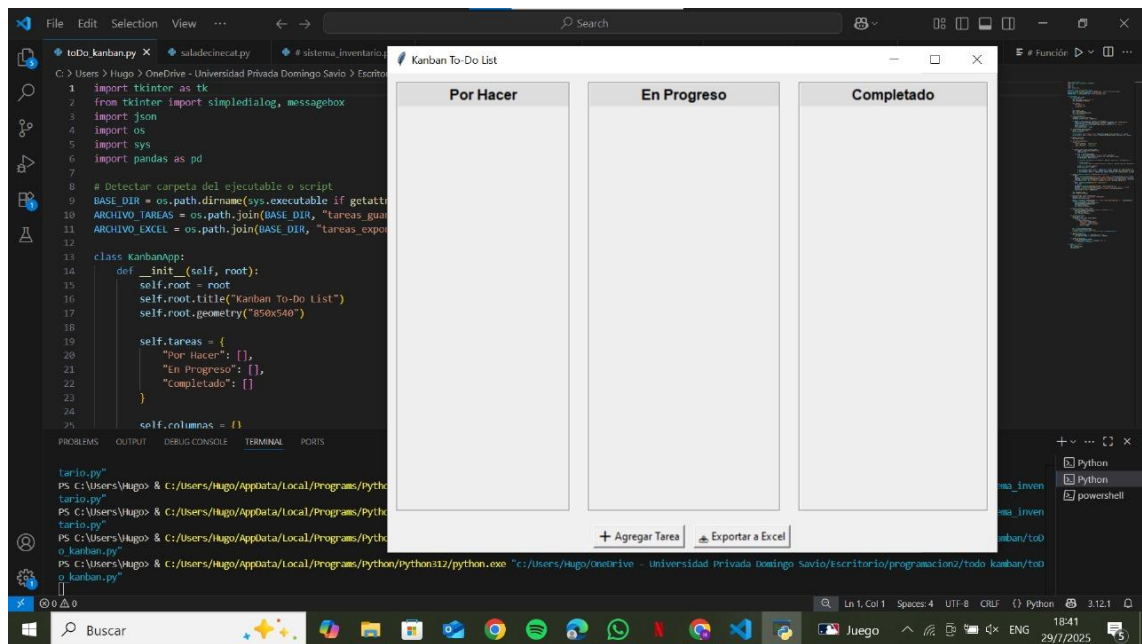
## 6. TODO LIST

### DESCRIPCÓN GENERAL

Este sistema está desarrollado en Python con Tkinter y permite gestionar productos, realizar ventas, trasladar stock entre sucursales y consultar un historial de actividades. Toda la información se guarda en archivos .json para persistencia.



El juego ejecutable se genera de la siguiente manera:



LINK DEL CODE:

<https://github.com/joseandres2003/TARANTINO/blob/main/PROYECTOS/toDolist.py>