



**FACULTAD DE INGENIERÍA CARRERA DE
INGENIERÍA INFORMÁTICA FACULTAD
COMUNITARIA DE CAACUPÉ**

PROGRAMACIÓN VI

PROFESOR: Ing. Ricardo Maidana

ALUMNA: Juana Espifania Otazo Benítez

Turno: Mañana

Caacupé – Paraguay

2023

CÓDIGO

```
import flet as ft
import random
import time

# Lista de elementos que se mostrarán en las tarjetas (pares)
elementos = ["🐱", "🐶", "🐭", "🐹", "🐻", "🐼", "🐹", "🐱"] * 2
random.shuffle(elementos) # Mezcla los elementos para que aparezcan en
orden aleatorio

def main(page: ft.Page):
    # Configuración inicial de la página
    page.title = "Inicio de Sesión - Juego de Memoria"
    page.window_width = 400
    page.window_height = 500
    page.padding = 10
    page.vertical_alignment = "center"
    page.horizontal_alignment = "center"

    # Variables para el juego
    tarjetas = [] # Lista para almacenar las tarjetas
    seleccionadas = [] # Lista para almacenar las tarjetas seleccionadas
    emparejadas = set() # Conjunto para almacenar las tarjetas
    emparejadas

    puntaje = ft.Text("Puntaje: 0", size=20, weight="bold") # Texto para
mostrar el puntaje
    intentos_errados = 0 # Contador de intentos errados
    intentos_permitidos = 5 # Número máximo de intentos permitidos

    # Función para mostrar el juego de memoria después del inicio de
sesión
    def mostrar_juego():
        nonlocal intentos_errados # Permite modificar la variable
intentos_errados dentro de la función

        page.clean() # Limpia la página actual
        page.title = "Juego de Memoria" # Cambia el título de la página

        # Actualizar el puntaje en la interfaz
        def actualizar_puntaje():
            puntaje.value = f"Puntaje: {len(emparejadas) // 2}" #
Actualiza el puntaje mostrando cuántas parejas se han encontrado
            page.update() # Actualiza la página

        # Función para finalizar el juego
        def finalizar_juego():
            page.clean() # Limpia la página actual
```

```

        # Muestra un mensaje de juego terminado y opciones para
reiniciar o salir
        page.add(ft.Column([
            ft.Text("Juego Terminado", size=30, weight="bold"),
            puntaje, # Muestra el puntaje final
            ft.Row([
                ft.ElevatedButton("Reiniciar", on_click=lambda e:
reiniciar_juego()),
                ft.ElevatedButton("Salir", on_click=lambda e:
page.window_close()),
            ], alignment=ft.MainAxisAlignment.CENTER)
        ], alignment=ft.MainAxisAlignment.CENTER))
        page.update() # Actualiza la página

# Reiniciar el juego
def reiniciar_juego():
    random.shuffle(elementos) # Mezcla de nuevo los elementos
    emparejadas.clear() # Limpia las tarjetas emparejadas
    intentos_errados = 0 # Reinicia el contador de intentos
errados

    puntaje.value = "Puntaje: 0" # Reinicia el puntaje
    mostrar_juego() # Vuelve a mostrar el juego

# Función para voltear una tarjeta
def voltear_tarjeta(e):
    nonlocal intentos_errados # Permite modificar la variable
intentos_errados dentro de la función
    idx = e.control.data # Obtiene el índice de la tarjeta
seleccionada
    if idx in emparejadas or idx in seleccionadas:
        return # Ignorar si la tarjeta ya está emparejada o
seleccionada

    # Mostrar el contenido de la tarjeta
    tarjetas[idx].content = ft.Text(elementos[idx], size=30)
    seleccionadas.append(idx) # Añadir índice a la lista de
seleccionadas
    page.update() # Actualiza la página

# Comprobar si se han seleccionado dos tarjetas
if len(seleccionadas) == 2:
    # Esperar un breve tiempo para que el usuario vea ambas
tarjetas
    time.sleep(0.5)
    # Comprobar si las tarjetas coinciden
    if elementos[seleccionadas[0]] ==
elementos[seleccionadas[1]]:
        emparejadas.update(seleccionadas) # Añadir al
conjunto de emparejadas

```

```

        actualizar_puntaje() # Actualiza el puntaje
    else:
        # Volver a ocultar las tarjetas si no coinciden
        for i in seleccionadas:
            tarjetas[i].content =
ft.Container(bgcolor="blue")
            intentos_errados += 1 # Incrementar el contador de
intentos errados
            # Si se alcanzan los intentos permitidos, finalizar
el juego
            if intentos_errados >= intentos_permitidos:
                finalizar_juego() # Llama a la función para
finalizar el juego
                seleccionadas.clear() # Reiniciar la selección
                page.update() # Actualiza la página

# Crear la cuadrícula de tarjetas
grid = ft.GridView(
    expand=True,
    runs_count=4,
    max_extent=100,
    child_aspect_ratio=1.0,
    spacing=10,
)

# Crear y añadir cada tarjeta a la cuadrícula
for i in range(len(elementos)):
    tarjeta = ft.Container(
        content=ft.Container(bgcolor="blue"), # Tarjeta oculta
        width=80,
        height=80,
        alignment=ft.alignment.center,
        on_click=voltear_tarjeta, # Asigna la función al evento
de clic
        data=i # Almacena el índice de la tarjeta
    )
    tarjetas.append(tarjeta) # Añadir la tarjeta a la lista
    grid.controls.append(tarjeta) # Añadir la tarjeta a la
cuadrícula

# Agregar la cuadrícula y el puntaje a la página
page.add(ft.Column([puntaje, grid], spacing=10,
alignment=ft.MainAxisAlignment.CENTER))
page.update() # Actualiza la página

# Función de inicio de sesión que lleva directamente al juego
def iniciar_sesion(e):
    mostrar_juego() # Llama a la función para mostrar el juego

```

```

# Componentes de la interfaz de inicio de sesión
email_input = ft.TextField(label="Correo electrónico", width=300) #
Campo para ingresar el correo electrónico
password_input = ft.TextField(label="Contraseña", password=True,
width=300) # Campo para ingresar la contraseña

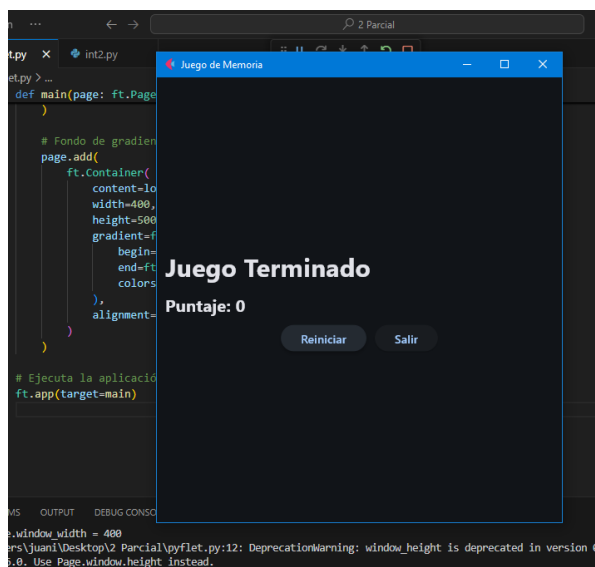
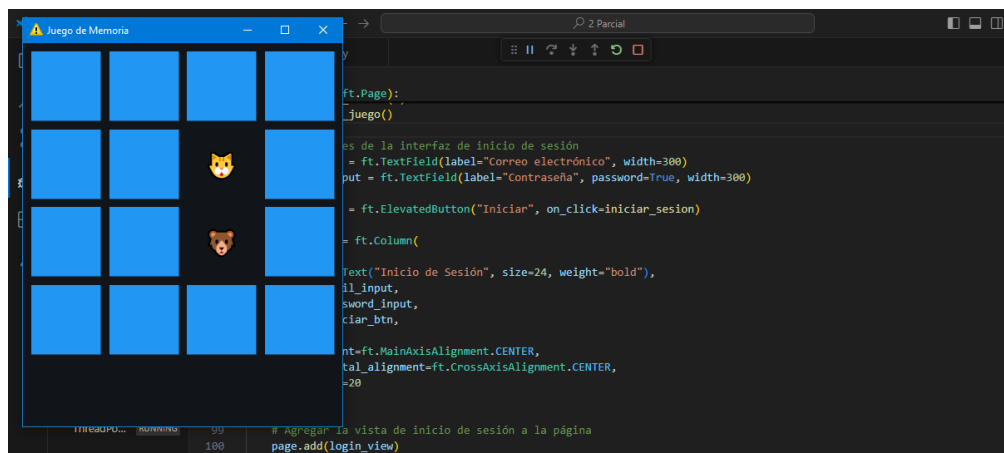
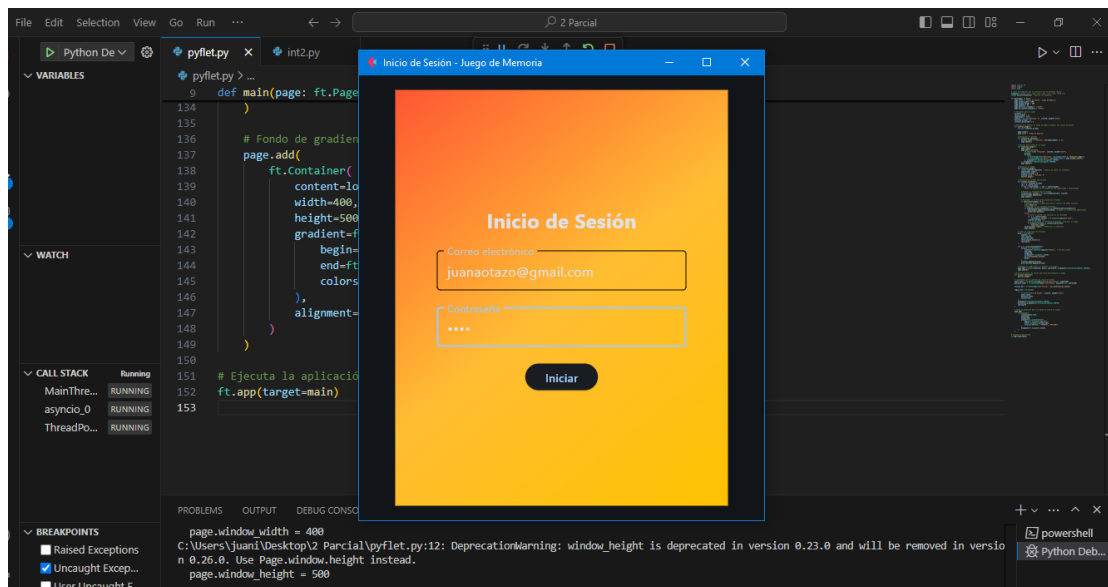
iniciar_btn = ft.ElevatedButton("Iniciar",
on_click=iniciar_sesion) # Botón para iniciar sesión

# Vista de inicio de sesión
login_view = ft.Column(
    [
        ft.Text("Inicio de Sesión", size=24, weight="bold"), #
Título
        email_input, # Campo de correo
        password_input, # Campo de contraseña
        iniciar_btn, # Botón de inicio
    ],
    alignment=ft.MainAxisAlignment.CENTER,
    horizontal_alignment=ft.CrossAxisAlignment.CENTER,
    spacing=20 # Espaciado entre componentes
)

# Fondo de gradiente para la página de inicio de sesión
page.add(
    ft.Container(
        content=login_view, # Añadir la vista de inicio de sesión
        width=400,
        height=500,
        gradient=ft.LinearGradient( # Crear un fondo de gradiente
            begin=ft.alignment.top_left,
            end=ft.alignment.bottom_right,
            colors=["#FF5733", "#FFBD33", "#FFC300"] # Colores del
gradiente
        ),
        alignment=ft.alignment.center, # Centrar el contenido
    )
)

# Ejecuta la aplicación
ft.app(target=main) # Inicia la aplicación

```



URL GitHub: https://github.com/JuanaOtazo/2_Parcial_J_O.git