

Proyectos Finales Estructura de datos.

1. Sistema de Monitoreo de Accesos

Objetivo: Simular un sistema que registre y analice los intentos de acceso de múltiples usuarios a distintos servidores, detectando posibles amenazas y generando reportes.

Componentes utilizados: - **Vectores:** usuarios, servidores - **Matrices:** intentos, IP, tipo, hora - **Condicionales y bucles:** validación, alertas, reportes - **Funciones:** RegistrarIntento, MostrarReporte, GenerarAlertas – **Bibliotecas (Investigar para este proyecto), incluye interfaz grafica y manual de uso.**

```
import tkinter as tk
from tkinter import ttk, messagebox
import datetime
import random

# -----
# Datos del sistema (vectores)
# -----
usuarios = ["Ana", "Luis", "Carlos", "María"]
servidores = ["Servidor1", "Servidor2", "Servidor3"]

# Matriz de intentos: [usuario, servidor, ip, tipo, hora]
intentos = []

# -----
# FUNCIONES PRINCIPALES
# -----


def RegistrarIntento():
    usuario = combo_usuario.get()
    servidor = combo_servidor.get()
    tipo = combo_tipo.get()
    ip = entry_ip.get()

    if usuario == "" or servidor == "" or tipo == "":
        messagebox.showwarning("Error", "Todos los campos deben estar llenos")
        return

    if ip == "":
        ip = f"192.168.{random.randint(0,255)}.{random.randint(0,255)}"

    hora = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    intentos.append([usuario, servidor, ip, tipo, hora])

    messagebox.showinfo("Éxito", "Intento registrado correctamente")
    entry_ip.delete(0, tk.END)
```

```
def MostrarReporte():
    text_reporte.delete(1.0, tk.END)

    if len(intentos) == 0:
        text_reporte.insert(tk.END, "No hay intentos registrados.\n")
        return

    for i, intento in enumerate(intentos):
        text_reporte.insert(tk.END,
            f'Intento #{i+1}\n'
            f"Usuario: {intento[0]}\n"
            f"Servidor: {intento[1]}\n"
            f"IP: {intento[2]}\n"
            f"Tipo: {intento[3]}\n"
            f"Hora: {intento[4]}\n\n"
        )
```

```
def GenerarAlertas():
    text_alertas.delete(1.0, tk.END)

    if len(intentos) == 0:
        text_alertas.insert(tk.END, "No hay intentos registrados.\n")
        return

    alertas = 0

    for intento in intentos:
        usuario, servidor, ip, tipo, hora = intento

        if tipo == "Fallido":
            alertas += 1
            text_alertas.insert(tk.END, f"[ALERTA] Intento fallido de {usuario} desde {ip} a las {hora}\n")

        if not ip.startswith("192.168"):
            alertas += 1
            text_alertas.insert(tk.END, f"[ALERTA] IP sospechosa detectada: {ip}\n")

    if alertas == 0:
        text_alertas.insert(tk.END, "No se detectaron amenazas.\n")
    else:
        text_alertas.insert(tk.END, f"\nTotal de alertas: {alertas}\n")
```

```
# -----
# INTERFAZ GRÁFICA
# -----
```

```
ventana = tk.Tk()
```

```
ventana.title("Sistema de Monitoreo de Accesos")
ventana.geometry("750x600")

titulo = tk.Label(ventana, text="Sistema de Monitoreo de Accesos", font=("Arial", 18, "bold"))
titulo.pack(pady=10)

frame_inputs = tk.Frame(ventana)
frame_inputs.pack(pady=10)

# Usuario
tk.Label(frame_inputs, text="Usuario:").grid(row=0, column=0)
combo_usuario = ttk.Combobox(frame_inputs, values=usuarios)
combo_usuario.grid(row=0, column=1)

# Servidor
tk.Label(frame_inputs, text="Servidor:").grid(row=1, column=0)
combo_servidor = ttk.Combobox(frame_inputs, values=servidores)
combo_servidor.grid(row=1, column=1)

# Tipo
tk.Label(frame_inputs, text="Tipo acceso:").grid(row=2, column=0)
combo_tipo = ttk.Combobox(frame_inputs, values=["Correcto", "Fallido"])
combo_tipo.grid(row=2, column=1)

# IP
tk.Label(frame_inputs, text="IP:").grid(row=3, column=0)
entry_ip = tk.Entry(frame_inputs)
entry_ip.grid(row=3, column=1)

# Botón registrar
btn_registrar = tk.Button(ventana, text="Registrar Intento", command=RegistrarIntento)
btn_registrar.pack(pady=10)

# Área de reporte
tk.Label(ventana, text="Reporte de Intentos", font=("Arial", 14)).pack()
text_reporte = tk.Text(ventana, height=10, width=80)
text_reporte.pack()

btn_reporte = tk.Button(ventana, text="Mostrar Reporte", command=MostrarReporte)
btn_reporte.pack(pady=5)

# Área de alertas
tk.Label(ventana, text="Alertas del Sistema", font=("Arial", 14)).pack()
text_alertas = tk.Text(ventana, height=10, width=80)
text_alertas.pack()

btn_alertas = tk.Button(ventana, text="Generar Alertas", command=GenerarAlertas)
btn_alertas.pack(pady=5)

ventana.mainloop()
```

