

The background is a dark navy blue. In the corners, there are abstract geometric shapes made of overlapping translucent polygons. The top-left corner features a green shape, a blue shape, and a pink shape. The top-right corner features a green shape, a blue shape, and a pink shape. The bottom-left corner features a green shape, a blue shape, and a pink shape. The bottom-right corner features a green shape, a blue shape, and a pink shape.

Motor de Inferencia

Adrian Ureña M. B57340

Motor de Inferencia

Funciona al procesar información y aplicar reglas lógicas para llegar a conclusiones o recomendaciones. Por ejemplo, un motor de inferencia en un sistema de recomendación de películas puede tomar información sobre las preferencias del usuario y aplicar reglas para sugerir nuevas películas que podrían interesarle.

```
1 import pandas as pd
2 from tkinter import filedialog
3 import tkinter as tk
4 from tkinter import ttk
```

- ❑ **numpy:** Es una biblioteca de Python que proporciona soporte para trabajar con arreglos multidimensionales y matrices.
- ❑ **sklearn.model_selection:** Es un módulo de scikit-learn (también conocido como sklearn) que proporciona herramientas para dividir conjuntos de datos en subconjuntos de entrenamiento y prueba de forma aleatoria.
- ❑ **sklearn.metrics:** Es un módulo de scikit-learn que contiene funciones para evaluar la precisión y el rendimiento de los modelos de aprendizaje automático.
- ❑ **mean_squared_error:** Es una función dentro de sklearn.metrics que calcula el error cuadrático medio entre las predicciones de un modelo y los valores reales.
- ❑ **matplotlib.pyplot:** Es un módulo de la biblioteca matplotlib que proporciona una interfaz para crear gráficos y visualizaciones en Python.

```
reglas = [  
  {  
    "nombre": "Antracnosis",  
    "síntomas_presentes": ["Manchas oscuras en frutos", "Manchas oscuras  
    "síntomas_ausentes": ["Polvo blanco y gris en hojas"],  
    "diagnostico": "La planta sufre de Antracnosis.",  
    "explicacion": "Detectadas manchas oscuras sin presencia de polvo bl  
  },  
  {  
    "nombre": "Sigatoka Negra (en banano)",
```

Se establece un arreglo "reglas" en el cual se colocan las reglas que va usar el motor de inferencia, a partir de esta estructura.

```
def evaluar_reglas(sintomas, reglas):  
    diagnosticos = []  
    explicaciones = []  
    for regla in reglas:  
        if all(s in sintomas for s in regla["sintomas_presentes"]) and \  
            not any(s in sintomas for s in regla["sintomas_ausentes"]):  
            diagnosticos.append(regla["diagnostico"])  
            explicaciones.append(regla["explicacion"])  
    return diagnosticos, explicaciones
```

- › `all(s in sintomas for s in regla["sintomas_presentes"])`: Verifica si todos los elementos en la lista `regla["sintomas_presentes"]` están presentes en la lista `sintomas`.
- › `not any(s in sintomas for s in regla["sintomas_ausentes"])`: Verifica si ningún elemento en la lista `regla["sintomas_ausentes"]` está presente en la lista `sintomas`.
- › Si ambas condiciones son verdaderas para una regla específica, significa que se cumplen todas las condiciones de esa regla y, por lo tanto, se agrega el diagnóstico y la explicación asociados con esa regla a las listas `diagnosticos` y `explicaciones`.

```
filename = filedialog.askopenfilename(title="Seleccionar archivo Excel",  
# Verificar si se seleccionó un archivo  
if filename:  
    # Cargar datos desde el archivo Excel  
    df = pd.read_excel(filename)  
  
    # Obtener síntomas y valores de las plantas desde el DataFrame  
    sintomas = df.columns.tolist()  
    valores = df.values.tolist()
```

`filename = filedialog.askopenfilename(...)`: Muestra un cuadro de diálogo que permite al usuario seleccionar un archivo Excel.

`df = pd.read_excel(filename)`: Si se ha seleccionado un archivo, este código utiliza la biblioteca pandas (pd) para cargar los datos del archivo Excel


```
for i, planta in enumerate(valores):  
    sintomas_planta = [sintoma for j, sintoma in enumerate(sintomas) if planta[j]]  
    print(f"Síntomas de la Planta {i+1}: {sintomas_planta}")
```

Evaluar las reglas de diagnóstico para cada planta

```
for i, planta in enumerate(valores):  
    sintomas_planta = [sintoma for j, sintoma in enumerate(sintomas) if planta[j]]  
    diagnosticos = evaluar_reglas(sintomas_planta, reglas)  
    resultados[f"Planta {i+1}"] = diagnosticos
```

for i, planta in enumerate(valores):: Este bucle for itera sobre cada elemento (planta) en la lista valores.

sintomas_planta = [sintoma for j, sintoma in enumerate(sintomas) if planta[j]]: Para cada planta, este código crea una lista llamada sintomas_planta que contiene solo los síntomas asociados con esa planta.

for i, planta in enumerate(valores):: Este es otro bucle for similar al primero, que itera sobre cada planta en la lista valores.

diagnosticos = evaluar_reglas(sintomas_planta, reglas): Para cada planta, este código llama a una función llamada evaluar_reglas() y le pasa los síntomas asociados con esa planta (sintomas_planta) y las reglas de diagnóstico (reglas).

```
def crear_tabla(root, datos):  
    tabla = ttk.Treeview(root)  
    tabla["columns"] = ("Nombre de Planta", "Enfermedad", "Descripción")  
    tabla.heading("#0", text="Índice")  
    tabla.heading("Nombre de Planta", text="Nombre de Planta")  
    tabla.heading("Enfermedad", text="Enfermedad")  
    tabla.heading("Descripción", text="Descripción")
```

`tabla = ttk.Treeview(root)`: Crea un objeto `ttk.Treeview` llamado `tabla` y lo asocia con el widget principal `root`.

`tabla["columns"] = ("Nombre de Planta", "Enfermedad", "Descripción")`: Define las columnas de la tabla. La tabla tendrá tres columnas con los encabezados "Nombre de Planta", "Enfermedad" y "Descripción".

`tabla.heading("Nombre", text="Nombre")`: Define el encabezado de la columna.

Nombre de Planta	Enfermedad	Descripción
Planta 1	La planta sufre de Podredumbre Negra	Presencia de manchas oscuras en la
Planta 2	La planta sufre de Antracnosis.	Detectadas manchas oscuras sin pre:
Planta 3	La planta sufre de Mancha Negra.	Presencia de manchas negras en las
Planta 4	La planta sufre de Sigatoka Negra (e	Presencia de manchas negras alarg
Planta 5	No tiene enfermedad	N/A
Planta 6	La planta sufre de Fusariosis (Fusariu	Presencia de marchitez y decoloraci
Planta 7	No tiene enfermedad	N/A
Planta 8	No tiene enfermedad	N/A
Planta 9	No tiene enfermedad	N/A
Planta 10	No tiene enfermedad	N/A

Interpretación:

En este caso, el motor de inferencia efectivamente, con las reglas y síntomas establecidas, da las enfermedades de la planta correctamente.