

# Trabajo Práctico Integrador

## Programación I - Datos complejos:

### Arboles

---

Alumno: Facundo Milano, Emmanuel Gonzalez

Materia: Programación I

Tema: Datos complejos: Arboles

Carrera: Tecnicatura Universitaria en Programación

Universidad: Universidad Tecnológica Nacional

Profesor/a: Cinthia Rigoni

Fecha de Entrega: 9/6/2025

#### Índice

1. Introducción
2. Marco Teórico
3. Caso Práctico
4. Metodología Utilizada
5. Resultados Obtenidos
6. Conclusiones
7. Bibliografía
8. Anexo

## 1-Introduccion

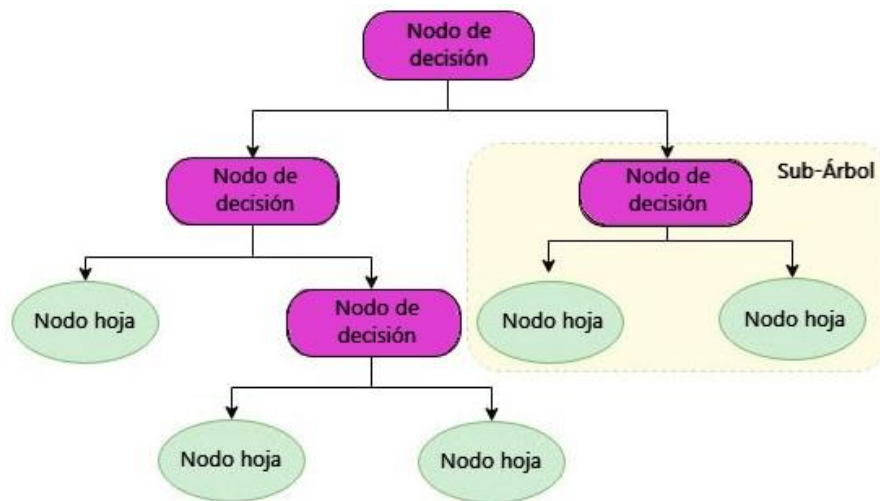
En el mundo de la programación, el buen manejo de los datos es fundamental para el diseño de algoritmos y sistemas complejos. Una de las estructuras de datos más importantes y versátiles para lograr esto son los árboles. Estos mismos permiten representar información de forma jerárquica, facilitando el acceso, la organización y el procesamiento de datos. Esta estructura, esta compuesta de nodos interconectados que establecen relaciones padre-hijo, lo que permite representar situaciones como jerarquías organizativas, sistemas de archivos, o estructuras de bases de datos.

En este trabajo implementaremos la estructura de árbol binario para la toma de decisiones, este mismo empieza con una pregunta (en la raíz), y según las respuestas, se sigue por diferentes caminos (ramas) hasta llegar a una conclusión (hojas).

## 2. Marco Teórico

Los árboles binarios son una estructura de datos jerárquica en la que cada nodo tiene, como máximo, dos hijos: uno izquierdo y uno derecho.

Su organización se basa en una secuencia de decisiones binarias, es decir, en cada punto del árbol se evalúa una condición que puede tener solo dos posibles salidas: verdadero o falso, sí o no.



Esta estructura se organiza con un nodo raíz y luego los hijos se van posicionando a la izquierda o la derecha dependiendo de si se cumple o no una condición:

- Rama izquierda: si la condición no se cumple.
- Rama derecha: si la condición se cumple.

El árbol se compone de los siguientes elementos:

**Nodo raíz:** Es el punto de inicial del árbol. Contiene la primera condición que se evalúa sobre los datos.

**Nodos internos:** Son nodos de decisión que aparecen después de la raíz. Cada uno contiene una condición basada en una característica específica de los datos. A partir de cada nodo interno, emergen exactamente dos ramas que representan las posibles respuestas a la condición (por ejemplo, sí/no o mayor/menor).

**Ramas:** Representan los caminos que se pueden seguir dependiendo del resultado de la condición en el nodo padre. En un árbol binario, cada nodo interno siempre tendrá dos ramas.

**Nodos hoja:** Son los nodos terminales del árbol. No contienen condiciones, sino que representan el resultado final de las decisiones tomadas en el camino desde la raíz.

#### Este tipo de árbol presenta como ventajas:

Fácil interpretación: Los árboles de decisión son fáciles de entender y visualizar.

Poca preparación de datos: No requieren una extensa preparación de los datos.

Manejo de datos numéricos y texto: Pueden manejar ambos tipos de datos.

#### Este tipo de árbol presenta como desventaja:

-Si el árbol es muy grande y complejo, puede aprender detalles específicos de los datos usados para entrenarlo y no funcionar bien cuando se enfrenta a datos nuevos.

-Cambios pequeños en los datos pueden dar árboles muy distintos.

Los árboles de decisión se utilizan para tomar decisiones basadas en datos de forma ordenada y fácil de entender. Uno de sus usos principales es la clasificación, como cuando se quiere saber si un correo es spam o si una persona tiene una enfermedad. También se usan para predecir valores numéricos, por ejemplo, calcular el precio estimado de una casa según su tamaño y ubicación.

Además, estos árboles ayudan en la toma de decisiones complejas, ya que muestran paso a paso las condiciones y posibles resultados. Su estructura clara permite identificar qué características influyen más en cada decisión, lo cual es muy útil para analizar los datos.

Estas características que poseen los árboles de decisión nos han llevado a querer utilizarlos para nuestro proyecto, ya que el mismo se va a centrar en la elección que tome el usuario para llegar a una decisión final.

### 3. Caso Práctico

Elegimos para la utilización de esta estructura la implementación de un algoritmo de toma de decisiones utilizado para elegir un modelo de auto, de la marca Volkswagen, dependiendo de las preferencias del usuario. Mediante el programa se vaya ejecutando el usuario deberá contestar las preguntas que el mismo le realice para poder elegir un modelo que sea acorde a lo que necesite.

Comenzamos definiendo la función para recorrer el árbol creado a base de listas. En el mismo el usuario va a tener que ingresar por consola la respuesta si o no, dependiendo de su preferencia a la pregunta que le realiza el programa.

```
#Definicion de funciones

def recorrer_arbol(nodo):
    # Muestra el resultado final de la eleccion
    if nodo[3] is not None:
        print(f"\nRecomendacion: {nodo[3]}")
        return

    # Toma la respuesta de la primer pregunta (nodo raiz)
    respuesta = input(nodo[0] + " (sí/no): ").strip().lower()

    # Dependiendo de la respuesta sigue por el nodo que corresponda
    if respuesta == "si":
        recorrer_arbol(nodo[1])
    elif respuesta == "no":
        recorrer_arbol(nodo[2])
    else:
        print("Respuesta incorrecta. Escriba 'si' o 'no'.")
        recorrer_arbol(nodo)
```

Luego definimos la función main(), que va a contener el árbol de decisiones creado a base de listas, para luego ser utilizado en la función recorrer\_arbol().

```
def main():
    # Arbol de decision creado a base de listas
    arbol = [
        "¿Querés un vehículo eléctrico?",

        # Si elije opcion SI, consulta si quiere un vehiculo ligero.
        [
            "¿Querés un vehículo ligero?",
            [None, None, None, "Te convendria una bicicleta eléctrica Volkswagen Cargo e-bike."],
            [None, None, None, "Te convendria un auto eléctrico Volkswagen ID.4."],
            None
        ],

        # Si elije opcion NO, consulta si quiere un vehiculo grande..
        [
            "¿Querés un vehículo grande?",
            [None, None, None, "Te convendria una camioneta, tenemos disponible una Volkswagen Amarok."],

            # Si elije opcion NO, consulta si quiere un vehiculo no tan grande y familiar.
            [
                "¿Querés un vehículo no tan grande y familiar?",
                [None, None, None, "Te convendria una camioneta para ciudad, tenemos disponible Volkswagen Suran."],
                [None, None, None, "Te convendria un auto para ciudad, tenemos disponible el Volkswagen Polo."],
                None
            ],
            None
        ],
        None
    ]
]
```

Por último, dentro de la función main() y luego de que este definido el árbol, se muestra por pantalla el mensaje del inicio del programa, y a continuación esta la función recorrer\_arbol() para iniciar con las elecciones del usuario.

```
# Mensaje de inicio del programa siguiendo con el recorrido del arbol.
print("Bienvenido al sistema de recomendacion de vehiculos Volkswagen, a continuacion debera contestar las siguientes preguntas:")
recorrer_arbol(arbol)
```

Después de definir las funciones se muestra el código para ser ejecutado, utilizando la función main() y a continuación de la misma el mensaje que saldrá por pantalla una vez finalizado el programa.

```
#Codigo de ejecucion.

main()

print("Gracias por usar nuestro sistema de recomendacion.")
```

#### 4. Metodología Utilizada

Para la realización del trabajo se buscó información relacionada al tema elegido, para poder completar el código con material real en referencia a la marca Volkswagen.

Para comenzar realizamos la creación del árbol con listas para después poder desarrollar la función para recorrerlo. Una vez creado el árbol, seguimos con la función `recorrer_arbol()` la cual implementamos que sea recursiva.

Se realizaron pruebas, dependiendo de las elecciones que podría llegar dar el usuario, para corroborar que se realice un recorrido correcto por el árbol, que las preguntas sean claras y que el usuario al ingresar entradas no validas este sea informado para que ingrese la correcta.

Para el entorno de desarrollo utilizamos Visual Studio Code, aplicando el lenguaje de programación Python.

Adicionalmente se realizo consultas a la inteligencia artificial a problemas surgidos en las pruebas del código.

#### 5. Resultados Obtenidos

Con el desarrollo del caso práctico se logró implementar un sistema de recomendación de vehículos Volkswagen basado en un arbol de decisiones. El programa guía al usuario mediante una serie de preguntas estructuradas, y ofrece una recomendación final según las preferencias del usuario (vehículo eléctrico, ligero, familiar, etc.).

Se realizaron pruebas de seguimiento en cada pregunta para ver si el programa finalizaba acorde a las preguntas que se hacían. También se realizo la prueba si el usuario ingresaba un dato distinto del que se pedía para corrobora que el programa se ejecute hasta que el usuario ingrese una entrada valida.

#### 6. Conclusiones

Al realizar este trabajo se pudo aplicar los conocimientos que se vienen aprendiendo en la cursada y poder utilizarlos en un proyecto que podría ser real.

El aprendizaje sobre arboles nos da una vista sobre como se pueden estructurar datos para ordenarlos y luego utilizarlos. Y nos muestra las variedades de implementaciones que existen para ello.

Para nosotros presento un desafío poder armar este tipo de estructura con las herramientas dadas, en el cual nuestro primer inconveniente era poder armar una lista que represente un árbol de decisiones.

Si bien al código se lo puede modificar para agregar más ramas para la toma de decisiones, creemos que con el armado que tuvimos, el mismo se logra entender fácilmente y cumple su función para que el usuario lo pueda utilizar sin que le resulte complejo.

## 7. Bibliografía

- <https://sitiobigdata.com/2019/12/14/arbol-de-decision-en-machine-learning-parte-1/>
- <https://www.datasciencebase.com/supervised-ml/algorithms/decision-trees/theory/>
- <https://medium.com/%40hpumah/arboles-de-desici%C3%B3n-98eb432777a7>
- <https://www.ibm.com/es-es/think/topics/decision-trees>
- Material de cátedra UTN - Arquitectura y Sistemas Operativos

## 8. Anexo

A continuación, se adjuntan imágenes del programa funcionando dependiendo de la elección del usuario.

Primer ejemplo:

```
Cuatrimestre/01-Programacion_I/Trabajo Práctico Integrador/Consulta_auto.py"
Bienvenido al sistema de recomendación de vehículos Volkswagen, a continuación deberá contestar las siguientes preguntas:
¿Querés un vehículo eléctrico? (si/no): si
¿Querés un vehículo ligero? (si/no): no

Recomendación: Te convendría un auto eléctrico Volkswagen ID.4.
Gracias por usar nuestro sistema de recomendación.
```

En este ejemplo el usuario elige que si quiere un auto eléctrico, y cuando el programa consulta si quiere un vehículo ligero, y el usuario indica que no, el programa le da la recomendación en base a lo que decidió.

Segundo ejemplo:

```
Bienvenido al sistema de recomendación de vehículos Volkswagen, a continuación deberá contestar las siguientes preguntas:
¿Querés un vehículo eléctrico? (si/no): no
¿Querés un vehículo grande? (si/no): no
¿Querés un vehículo no tan grande y familiar? (si/no): si

Recomendación: Te convendría una camioneta para ciudad, tenemos disponible Volkswagen Suran.
Gracias por usar nuestro sistema de recomendación.
```

En este ejemplo el usuario no quiere un vehículo eléctrico, y el programa le consulta si busca un vehículo grande, el usuario al contestar que no, el programa le consulta si busca un vehículo familiar no tan grande y al contestar que si el programa le da la recomendación que mas se ajusta a lo que decidió.

Tercer ejemplo:

```
Cuatrimestre/01-Programacion_I/Trabajo Practivo Integrador/Consulta_auto.py"
Bienvenido al sistema de recomendacion de vehiculos Volkswagen, a continuacion debera contestar las siguientes preguntas:
¿Quieres un vehículo eléctrico? (si/no): no
¿Quieres un vehículo grande? (si/no): uno chico
Respuesta incorrecta. Escriba 'si' o 'no', para poder continuar.
¿Quieres un vehículo grande? (si/no): un vehiculo mediano
Respuesta incorrecta. Escriba 'si' o 'no', para poder continuar.
¿Quieres un vehículo grande? (si/no): si

Recomendacion: Te convendría una camioneta, tenemos disponible una Volkswagen Amarok.
Gracias por usar nuestro sistema de recomendacion.
```

En este ejemplo el usuario indica que no quiere un vehículo eléctrico, pero cuando el programa le pregunta si quiere un vehículo grande, el usuario escribe que tipo de vehículo quiere en referencia al tamaño. El programa vuelve a formular la pregunta, pero antes le aclara que debe responder si o no para poder continuar.

Link Video:

[https://www.youtube.com/watch?v=V5UJGDty9tk&ab\\_channel=FacundoMilano](https://www.youtube.com/watch?v=V5UJGDty9tk&ab_channel=FacundoMilano)

Link Repositorio: [https://github.com/EmmanuelJG/TP-Integrador\\_Arboles.git](https://github.com/EmmanuelJG/TP-Integrador_Arboles.git)