

# **Trabajo Práctico Integrador**

# **ÁRBOLES EN PYTHON**

Jennifer Franco ([jennyfranco31.jf@gmail.com](mailto:jennyfranco31.jf@gmail.com))  
Jonathan Franco ([nahuelfranco7@gmail.com](mailto:nahuelfranco7@gmail.com))

UTN-2025

**TEMA:**

# ÁRBOL DE DECISIÓN BINARIO

Implementación de un sistema de diagnóstico médico mediante árboles de decisión binarios en Python, aplicado a la detección de COVID-19.

Uso de programación orientada a objetos para modelar nodos y ramas del árbol, demostrando la utilidad de esta herramienta en la automatización de decisiones.

# ¿QUÉ ES UN ÁRBOL?

Un árbol es una estructura de datos formada por nodos organizados de manera jerárquica.

Cuenta con un nodo principal llamado raíz, del cual se ramifican otros nodos a través de conexiones llamadas ramas.

Estas ramas representan relaciones entre los nodos, permitiendo organizar la información de forma que cada elemento dependa de otro, similar a un árbol genealógico.

# ¿QUE ES UN ÁRBOL BINARIO?

Un árbol binario es un tipo de estructura de datos de árbol donde cada nodo puede tener un máximo de dos nodos secundarios, un nodo secundario izquierdo y un nodo secundario derecho

# ¿QUÉ ES UN ÁRBOL DECISIÓN?

Los árboles de decisión son una técnica de aprendizaje supervisado usada para predecir resultados. A partir de datos, crean diagramas lógicos que representan reglas para clasificar y resolver problemas de forma repetitiva, especialmente en inteligencia artificial.

# ¿POR QUÉ ELEGIMOS LOS ARBOLES DE DECISION?

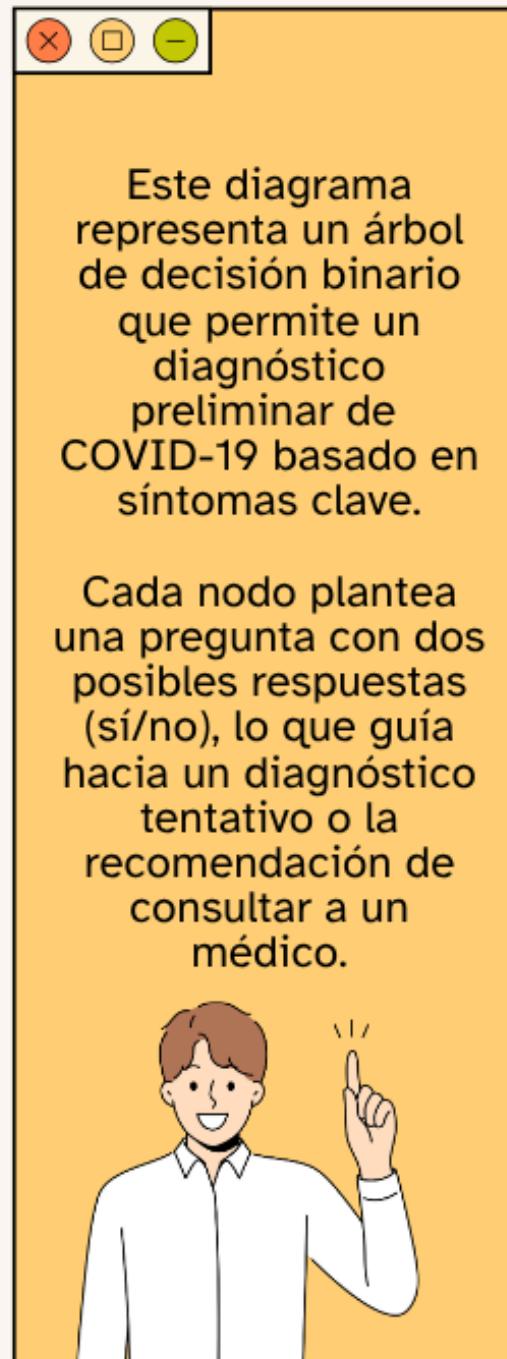
Los árboles de decisión organizan los datos de forma estructurada y eficiente, comenzando en un nodo raíz y descendiendo por ramas según las respuestas hasta llegar a nodos hoja que representan decisiones finales.

Cada nodo interno formula una pregunta sobre un atributo, y sus ramas reflejan las posibles respuestas, facilitando la clasificación del problema.

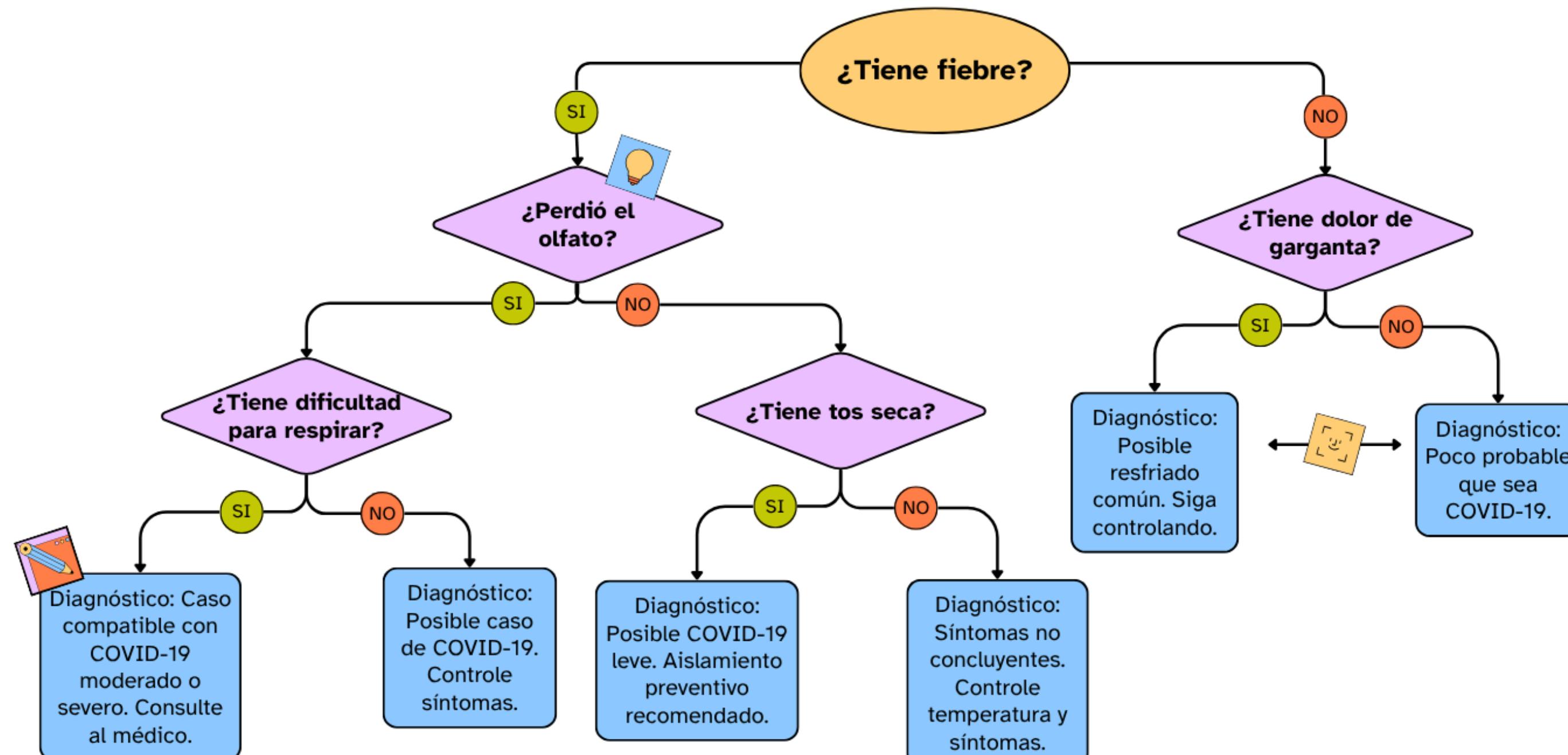


Para profundizar en los aspectos teóricos, se puede consultar el repositorio en GitHub:

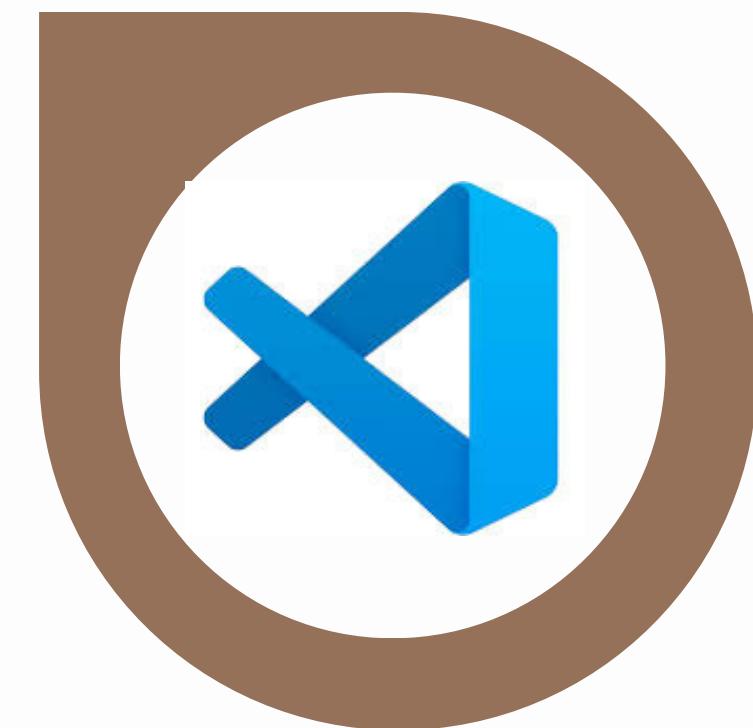
<https://github.com/JenniferFranco/tp-integrador-programacion.git>



## DIAGNÓSTICO DE COVID-19



Ahora pasemos a la construcción del código en Python para ver cómo se implementa este árbol de decisión.



# METODOLOGÍA

- ✓ Selección del tema: árbol de decisión binario aplicado a un diagnóstico médico.
- ✓ Investigación teórica.
- ✓ Diseño: estructura lógica de árbol binario con preguntas sí/no y diagnósticos tentativos.
- ✓ Implementación: desarrollo de clases Nodo y ArbolDeDecision en Python (POO).
- ✓ Pruebas: verificación del flujo de preguntas y coherencia de los diagnósticos.

# RESULTADOS

- ✓ Desarrollo de un sistema de diagnóstico médico simplificado y funcional.
- ✓ Flujo interactivo de preguntas (sí/no) sobre síntomas de COVID-19.
- ✓ Diagnóstico tentativo según combinaciones de síntomas.
- ✓ Validación de la lógica del árbol: decisiones encadenadas claras.
- ✓ Ejecución exitosa en consola con mensajes de colores (librería Colorama).

# CONCLUSIÓN

- ✓ Implementamos árboles de decisión binarios para guiar a los usuarios en un sistema de diagnóstico médico.
- ✓ Aplicamos programación orientada a objetos: clases, métodos y jerarquía de nodos.
- ✓ Validamos entradas de usuario y organizamos el código de forma clara y modular.
- ✓ Identificamos oportunidades de mejora:
  1. Incorporar datos personalizados como edad o antecedentes.
  2. Analizar múltiples síntomas en paralelo.
  3. Incluir niveles de gravedad o riesgo.
  4. Aplicar aprendizaje automático para mayor precisión.

# PARA MEJORAR Y ESCALAR EL SISTEMA, SE PUEDEN INCLUIR:

## 01. Pandas

Manejo y análisis eficiente de datos.

## 02. DecisionTreeClassifier (de scikit-learn)

Implementa fácilmente el árbol de decisión.

## 03. Confusion\_matrix (de scikit-learn):

Evalúa la precisión de la clasificación.

## 04. Pydotplus

Visualización y exportación de árboles.

## 05. Matplotlib y Seaborn

Para mostrar gráficos y visualizaciones básicas

## 06. NumPy

Operaciones rápidas con datos.



**Muchas  
GRACIAS**