

## Trabajo Práctico Integrador Árboles con listas enlazadas


 **Materia:**  
**Programación I**

 **Profesor:** Trapé, Julieta.-

 **Tutor:** López, Angel David.-

 **Alumnos:**

- Martínez, Agustín Ezequiel.-
- Mele Gavazza, Bruno.-

 **Fecha de Entrega:** 09 de Junio de 2025.-



## Índice

<b>1. Introducción</b>	<b>Pág. 3</b>
<b>2. Objetivos</b>	<b>Pág. 3</b>
<b>3. Marco teórico</b>	<b>Pág. 4</b>
<b>4. Metodología utilizada</b>	<b>Pág. 5</b>
<b>5. Desarrollo / Implementación</b>	<b>Pág. 6</b>
<b>6. Resultados</b>	<b>Pág. 7</b>
<b>7. Conclusiones</b>	<b>Pág. 8</b>
<b>8. Bibliografía</b>	<b>Pág. 8</b>

## 1. Introducción

Este trabajo práctico tiene como objetivo aplicar el conocimiento de estructuras de datos, particularmente árboles binarios implementados con listas en Python, en el contexto de un torneo de fútbol tipo Mundial. La idea principal es representar el desarrollo de un torneo desde su fase de grupos hasta la final utilizando una estructura de árbol.

En un Mundial de Fútbol participan 32 equipos que se agrupan inicialmente en 8 grupos de 4 integrantes. A medida que avanza el torneo, los equipos clasificados pasan a fases eliminatorias directas, lo que puede representarse de forma natural mediante un árbol binario, donde cada partido genera un ganador que avanza al siguiente nivel. Esta estructura jerárquica se ajusta perfectamente a un árbol, en el que cada nodo representa un enfrentamiento, y sus hijos, los equipos que compiten.

La implementación propuesta en este trabajo se centra en el uso de listas anidadas para representar árboles en Python, lo cual permite trabajar con estructuras complejas sin recurrir a clases u orientación a objetos. Esta elección metodológica busca simplificar la lógica subyacente para que el foco esté en la comprensión de la estructura del árbol y sus operaciones.

El desarrollo incluye la creación de una estructura inicial fija con los grupos, una etapa de interacción con el usuario para seleccionar los equipos clasificados, y la generación dinámica del árbol de eliminación directa. Asimismo, se ofrece la posibilidad de elegir entre una simulación automática (con resultados al azar) o una manual (interactiva). Finalmente, se implementan distintos tipos de recorrido del árbol (preorden, inorden, postorden) y funciones de búsqueda para identificar países específicos o conocer los puestos clave (campeón, subcampeón, etc.).

Este enfoque lúdico y práctico facilita la asimilación de conceptos fundamentales sobre árboles y su aplicación en un entorno conocido y motivador como lo es un torneo deportivo.

## 2. Objetivos

- Representar un torneo de fútbol mediante árboles binarios con listas anidadas.
- Utilizar estructuras de control para interactuar con el usuario y obtener resultados.
- Implementar distintos tipos de recorrido de árboles (preorden, inorden y postorden).
- Aplicar búsquedas en el árbol para conocer información como ganadores y posiciones.

### 3. Marco Teórico

Un árbol es una estructura de datos no lineal que se organiza jerárquicamente. A diferencia de las listas o arreglos tradicionales, los árboles permiten representar relaciones padre-hijo, lo que los hace especialmente útiles para modelar sistemas organizacionales, expresiones matemáticas, archivos en un sistema operativo y, en este caso, torneos deportivos.

En particular, un árbol binario es un tipo de árbol en el que cada nodo puede tener como máximo dos hijos: uno izquierdo y uno derecho. Esta restricción simplifica muchas operaciones y hace que los árboles binarios sean una estructura común en algoritmos de búsqueda, ordenamiento y toma de decisiones.

Cada nodo del árbol contiene un valor y referencias (o enlaces) a sus subárboles izquierdo y derecho. En el contexto de este trabajo, se adopta una representación sencilla y didáctica utilizando listas en Python. Cada nodo se estructura como una lista de tres elementos: [valor, subárbol\_izquierdo, subárbol\_derecho].

Esta representación evita el uso de clases y objetos, facilitando así la comprensión del funcionamiento de los árboles desde una perspectiva más funcional y directa. Es especialmente útil para estudiantes que están dando sus primeros pasos en estructuras de datos y programación en Python.

En el caso del Mundial de Fútbol, esta estructura permite representar cada partido de la fase eliminatoria como un nodo del árbol. Los dos hijos (subárboles) representan los equipos que se enfrentan, y el nodo padre contiene el nombre del equipo ganador. A medida que los resultados de los partidos se van resolviendo (de forma manual o automática), se completa el árbol desde las hojas (octavos de final) hasta la raíz (el campeón del torneo).

Además de la construcción del árbol, se exploran operaciones fundamentales como los recorridos (preorden, inorden y postorden), que permiten recorrer los equipos en distintos órdenes, y la búsqueda de nodos, lo cual facilita la localización de un país o posición específica (por ejemplo, campeón o semifinalista).

En resumen, el árbol binario se convierte en una herramienta poderosa y flexible para representar la lógica progresiva de un torneo, haciendo que este trabajo no solo sea funcional sino también pedagógicamente enriquecedor.

## 4. Metodología utilizada

- Se parte de una lista anidada fija que representa los 8 grupos del Mundial, con 4 países cada uno.
- Se solicita al usuario ingresar por consola los dos países que pasan de cada grupo.
- Con esa información, se va armando un árbol que representa la fase eliminatoria.
- El usuario puede elegir si quiere completar los resultados de forma manual o automática.
- En la modalidad automática, se eligen los ganadores al azar hasta obtener el campeón.
- El árbol puede recorrerse en preorden, inorden o postorden, mostrando los países en orden respectivo.
- Se puede realizar una búsqueda en el árbol para encontrar un país o alguna posición importante.

## 5. Desarrollo / Implementación

<https://github.com/ByJeJox/Trabajo-Integrador-Programacion/blob/main/Campeonato/Mundial.py>

Se adjuntó el link del código atento a lo extenso del código para exponerlo en el presente documento.

## 6. Resultados Obtenidos

La implementación realizada permite simular un torneo de fútbol tipo Mundial de manera interactiva utilizando estructuras de árboles binarios representadas con listas en Python. A través de esta propuesta, se logró construir una representación jerárquica clara de la evolución del torneo desde la fase de grupos hasta la gran final, lo cual refleja fielmente el formato tradicional de este tipo de competencias.

Durante el desarrollo de la aplicación, se observó que este enfoque facilita significativamente la comprensión de conceptos clave de estructuras de datos, tales como jerarquía entre nodos, recorridos del árbol y búsqueda de elementos. Al interactuar con el sistema, el usuario puede visualizar de forma concreta cómo los equipos avanzan ronda a ronda, y cómo el árbol se va construyendo dinámicamente a partir de las decisiones tomadas (manuales o automáticas).

Además, la implementación de los diferentes tipos de recorrido (preorden, inorden y postorden) permite ver el mismo árbol desde distintas perspectivas, lo cual resulta útil no solo para entender el comportamiento de cada tipo de recorrido, sino también para apreciar cómo varía la lógica de visita a los nodos según el método aplicado.

Por otro lado, el agregado de funcionalidades como la búsqueda de países o la consulta de posiciones finales (campeón, subcampeón, tercer y cuarto puesto) refuerza el valor educativo del trabajo, al conectar directamente la teoría de árboles con un caso práctico, visual y entretenido.

En conclusión, este enfoque no solo cumple con los objetivos académicos propuestos, sino que también ofrece una experiencia lúdica que contribuye al aprendizaje significativo de estructuras de datos en programación.

## 7. Conclusiones

El uso de árboles binarios representados con listas en Python resulta una estrategia efectiva para comprender cómo se organizan y recorren estructuras jerárquicas. Aplicado a la simulación de un torneo como el Mundial, permite visualizar de forma práctica cómo cada resultado determina el avance hacia la final. Esta implementación no solo refuerza conceptos clave como inserción, recorrido y búsqueda en árboles, sino que también promueve el aprendizaje activo y contextualizado. Por su simplicidad y utilidad, es una herramienta ideal para introducir estructuras de datos en niveles educativos iniciales.

## 8. Bibliografía

**Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C.**  
*Introduction to Algorithms. MIT Press, 2009.*

Obra fundamental en informática. Proporciona conceptos teóricos sólidos sobre estructuras de datos, incluyendo árboles binarios y sus recorridos.

**Miller, B. & Ranum, D.**  
*Problem Solving with Algorithms and Data Structures using Python.*

Texto educativo que explica de forma clara y aplicada cómo implementar estructuras de datos como árboles utilizando listas en Python. Disponible online de forma gratuita.

**Python Software Foundation**  
<https://docs.python.org>

Documentación oficial del lenguaje Python. Útil para consultar detalles de sintaxis, estructuras como listas y funciones utilizadas en la implementación.

### **Apuntes de clase / Material docente**

Información proporcionada en clases teóricas y prácticas sobre estructuras de datos y su implementación en Python, utilizada como base para el diseño del trabajo práctico.



