

Kliver Daniel Giron
Cristian Andrés Cobo
Joe Hernández

Informe laboratorio

Contexto problemático:

1. Identificación del problema:

Una empresa desarrolladora desea construir un software que consiste en el manejo de información masiva de basquetbolistas de todo el mundo de manera eficiente. Para darle solución a ésta problemática se requiere realizar la implementación de una herramienta para el manejo de información de gran tamaño que permita ingresar datos, ya sea de gran volumen o a través de una interfaz. Es de gran relevancia que la manipulación de estos datos mediante las diferentes funciones como eliminar, modificar, realizar consultas de los jugadores, etc debe ser eficiente, es decir, el tiempo debe ser mínimo utilizando como criterios de búsqueda las categorías estadísticas incluidas por ejemplo, encontrar aquellos jugadores que han anotado una cantidad de puntos por partido, o más de 20 rebotes por partido, etc.

2. Recopilación de información:

- **Baloncesto:**

El baloncesto es un deporte de equipo, jugado entre dos conjuntos de cinco jugadores cada uno durante cuatro períodos o cuartos de diez o doce minutos cada uno. El objetivo del equipo es anotar puntos introduciendo un balón por la canasta, un aro a 3,05 metros sobre la superficie de la pista de juego del que cuelga una red. La puntuación por cada canasta o cesta es de dos o tres puntos, dependiendo de la posición desde la que se efectúa el tiro a canasta, o de uno, si se trata de un tiro libre por una falta de un jugador

contrario. El equipo ganador es el que obtiene el mayor número de puntos.

El contacto con la pelota se realiza con las manos. Los jugadores, también llamados baloncestistas, no pueden trasladarse sujetando la pelota, sino botándola contra el suelo. El equipo en posesión del balón o atacante, intenta anotar puntos mediante tiros, entradas a canasta o mates, mientras que el equipo defensor busca impedirlo robando la pelota o efectuando tapones. Cuando un tiro hacia la canasta fracasa, los jugadores de ambos equipos intentan atrapar el rebote.

- **¿Cómo es el formato de puntaje en baloncesto?**

Tiro libre	Tiro de campo	Tiro triple
1 Punto	2 Puntos	3 Puntos

- **Rebote (baloncesto)** :un rebote en baloncesto es el acto de conseguir la posesión del balón después de un lanzamiento de campo o de un tiro libre fallado. Son uno de los objetivos fundamentales de los hombres altos de un equipo, aunque cualquier jugador en pista puede intentar conseguirlos.
- **Asistencia (baloncesto)** :una **asistencia** es un pase a un jugador que se encuentra en una posición de ventaja o que le ayuda a conseguir una canasta. Las asistencias se cuentan entre las estadísticas individuales y colectivas de los equipos de **baloncesto**.
- **Robo (baloncesto)**: los robos se acreditan al jugador defensivo que causa primero la pérdida de balón, incluso si no termina con la posesión de la pelota viva. Para ganar un robo, el jugador defensivo debe ser el iniciador de la acción que causó la pérdida de balón, no

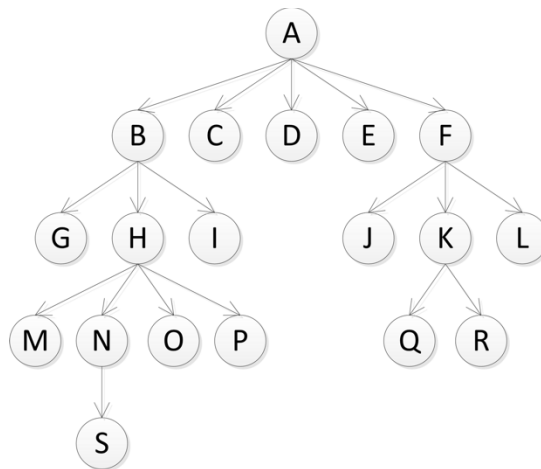
sólo el benefactor. Cada vez que un robo se registra por un jugador defensivo, un jugador ofensivo debe ser acreditado con un balón perdido.

3. Búsqueda de soluciones creativas:

- **Alternativa No. 001 [Árbol n-ario]:**

Es una estructura de datos donde cada nodo posee un número indeterminado de hijos. es una estructura recursiva, y corresponde a la generalización de un árbol binario de cuyos nodos pueden desprenderse múltiples árboles binarios. las reglas que aplican a los árboles binarios pueden ser fácilmente transpoladas a los árboles n-arios así como los consejos base.

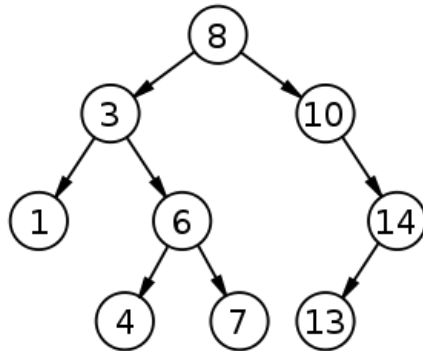
Un árbol n-ario puede tomarse como un árbol de n elementos asociados a cada uno de sus componentes.



- **Alternativa No. 002 [Árbol binario]:**

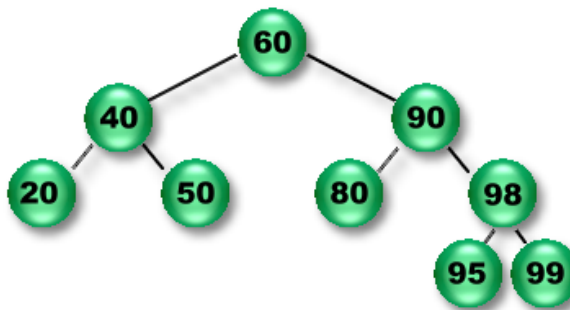
En ciencias de la computación, un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos (de ahí el nombre "binario"). Si algún hijo tiene como referencia a null, es decir que no

almacena ningún dato, entonces este es llamado un nodo externo.
En el caso contrario el hijo es llamado un nodo interno.



- **Alternativa No. 003 [Arbol AVL]:**

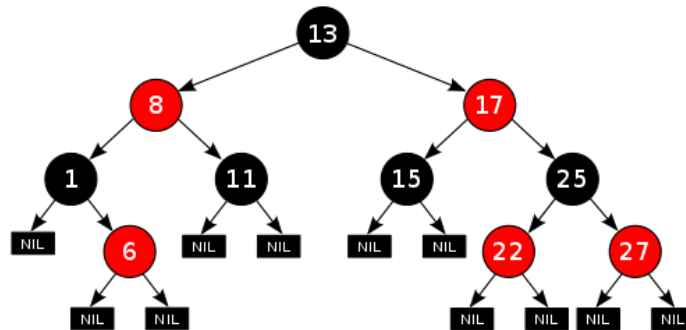
Un árbol AVL es un tipo especial de árbol binario auto balanceable. Los árboles AVL están siempre equilibrados de tal modo que para todos los nodos, la altura de la rama izquierda no difiere en más de una unidad de la altura de la rama derecha o viceversa. Gracias a esta forma de equilibrio (o balanceo), la complejidad de una búsqueda en uno de estos árboles se mantiene siempre en orden de complejidad $O(\log n)$. El factor de equilibrio puede ser almacenado directamente en cada nodo o ser computado a partir de las alturas de los subárboles.



- **Alternativa No. 004 [Arbol Rojo - Negro]:**

Un árbol rojo-negro es un tipo abstracto de datos. Concretamente, es un árbol binario de búsqueda equilibrado, una estructura de datos utilizada en informática y ciencias de la computación. Un árbol rojo-negro es un tipo especial de árbol binario usado en informática para organizar información compuesta por datos comparables (por ejemplo, números). En los árboles rojo-negro las hojas no son relevantes y no contienen datos.

En los árboles rojo-negro, como en todos los árboles binarios de búsqueda, es posible moverse ordenadamente a través de los elementos de forma eficiente si hay forma de localizar el padre de cualquier nodo. El tiempo de desplazarse desde la raíz hasta una

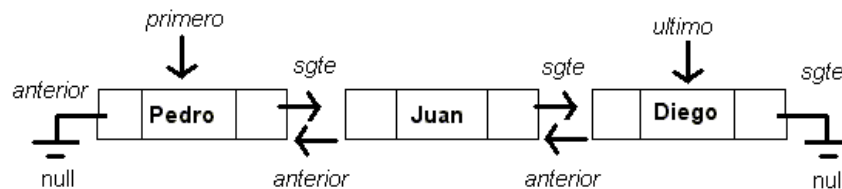


hoja a través de un árbol equilibrado que tiene la mínima altura posible es de $O(\log n)$.

Al implementar esta estructura es posible utilizar un único nodo centinela. Este cumple la función de hoja para todas las ramas del árbol. Así, todos los nodos internos que finalicen en una hoja tienen referencia a este único nodo centinela. Esto no es necesario, ya que puede hacerse una referencia nula (NIL) en el final de cada rama.

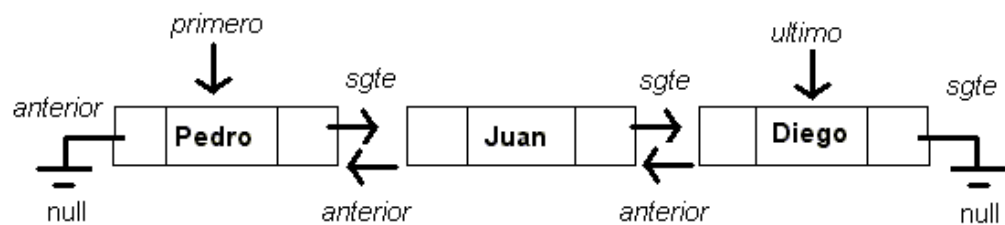
- **Alternativa No. 005 [Lista simplemente enlazada]:**

Una lista enlazada simple es una estructura de datos en la que cada elemento apunta al siguiente. De este modo, teniendo la referencia del principio de la lista podemos acceder a todos los elementos de la misma



- **Alternativa No. 006 [Lista doblemente enlazada]:**

En ciencias de la computación, una lista doblemente enlazada es una estructura de datos que consiste en un conjunto de nudos enlazados secuencialmente. Cada nudo contiene dos campos, llamados enlaces, que son referencias al nudo siguiente y al anterior en la secuencia de nudos. El enlace al nudo anterior del primer nudo y el enlace al nudo siguiente del último nudo, apuntan a un tipo de nudo que marca el final de la lista, normalmente un nudo centinela o puntero null, para facilitar el recorrido de la lista. Si existe un único nudo centinela, entonces la lista es circular a través del nudo centinela.



El doble enlace de los nodos permite recorrer la lista en cualquier dirección. Mientras que agregar o eliminar un nodo en una lista doblemente enlazada requiere cambiar más enlaces que en estas mismas operaciones en una lista enlazada simple, las operaciones

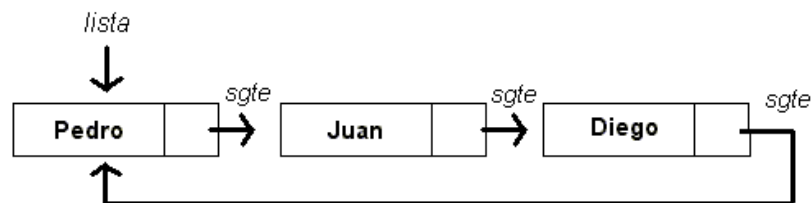
son más simples porque no hay necesidad de mantener guardado el nodo anterior durante el recorrido, ni necesidad de recorrer la lista para hallar el nodo anterior, la referencia al nodo que se quiere eliminar o insertar es lo único necesario.

- **Alternativa No. 007 [Lista circular]:**

La lista circular es una especie de lista enlazada simple o doblemente enlazada, pero que posee una característica adicional para el desplazamiento dentro de la lista: esta no tiene fin.

Para que la lista sea sin fin, el puntero siguiente del último elemento apuntará hacia el primer elemento de la lista en lugar de apuntar al valor NULL, como hemos visto en el caso de listas enlazadas simples o doblemente enlazadas.

En las listas circulares, nunca se llega a una posición en la que ya no sea posible desplazarse. Cuando se llegue al último elemento, el desplazamiento volverá a comenzar desde el primer elemento.

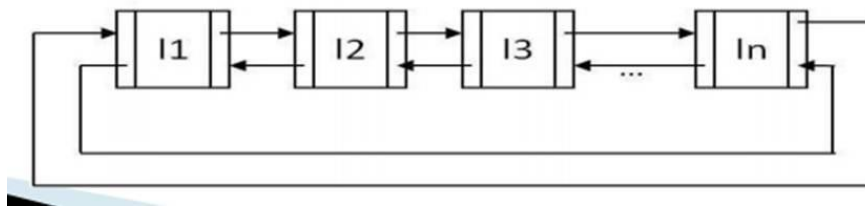


- **Alternativa No. 008 [Lista doblemente circular]:**

La lista circular doble es una especie de lista enlazada “doblemente enlazada”, pero que posee una característica adicional para el desplazamiento dentro de la lista, “ésta no tiene fin” y tiene 2 apuntadores a si misma.

Para que la lista sea sin fin, el puntero siguiente del último elemento apuntará hacia el 1er elemento y el puntero anterior del primer elemento apunta hacia el último elemento de la lista en lugar de apuntar al valor NULL, como hemos visto en el caso de listas enlazadas simples o doblemente enlazadas.

En las listas circulares dobles, nunca se llega a una posición en la que ya no sea posible desplazarse. Cuando se llegue al último elemento, el desplazamiento volverá a comenzar desde el primer elemento.



4. Transición de las ideas a diseños preliminares

Tras haberse hecho búsquedas exhaustivas de diferentes posibles estructuras de datos para darle solución los problemas anteriormente presentados, es momento de analizar cuáles serían quizás las más eficientes.

Al hacer diferentes revisiones diferentes se conduce a lo siguiente:

Alternativa 1 (Listas enlazadas)

- Acceso a posiciones intermedias con coste dependiente del tamaño de la lista
- Necesidad de memoria adicional para almacenar los objetos Node con sus atributos Entre una de sus desventajas podemos mencionar que al tener que mantener dos referencias el código se vuelve más complejo, y también que ocupa más espacio en memoria.
- Para la posible solución del problema planteado es muy importante que el acceso y eliminación de items almacenados en una estructura sea lo suficientemente rápido. Esto permite decir que una lista enlazada no es una buena opción para ser considerada más adelante

Alternativa 2 (Listas doblemente enlazadas)

- Entre una de sus desventajas podemos mencionar que al tener que mantener dos referencias el código se vuelve más complejo, y también que ocupa más espacio en memoria.
- Para la posible solución del problema planteado es muy importante que el acceso y eliminación de items almacenados en una estructura sea lo suficientemente rápido. Esto permite decir que una lista enlazada no es una buena opción para ser considerada más adelante

Alternativa 2 (Listas circulares)

- No permite el acceso directo a un elemento arbitrario de la lista Para acceder al i ésimo elemento debemos recorrer la lista comenzando por el primer nodo hasta llegar al elemento deseado

5. Evaluación y selección de la mejor solución

Después de analizar los diferentes problemas y determinar aspectos que son relevantes para la elección de las posibles estructuras a implementar en la solución se hace las siguientes evaluaciones.

A continuación se evidencia los criterios que llevan a la elección de la solución.

Criterio A. Acceso rápido, se requiere un algoritmo el cual se eficiente en el acceso.

- [5] Constante
- [4] Mayor a constante
- [3] Logaritmica
- [2] Mayor a logaritmica
- [1] Lineal

Criterio B. Inserción. El problema exige insercion rapida

- [5] Constante
- [4] Mayor a constante
- [3] Logaritmica
- [2] Mayor a logaritmica
- [1] Lineal

Criterio C Eliminación . Un algoritmo que sea eficiente en el momento de eliminar información contenida en una estructura.

- [5] Constante
- [4] Mayor a constante
- [3] Logaritmica
- [2] Mayor a logaritmica
- [1] Lineal

Criterio D Estructura autobalanceada. Las estructuras se autobalancean

- [2] si
- [1] no

Criterio E facil implementación . Las estructuras se implementan fácilmente en lenguaje modernos.

- [2] Fácil
- [1] La Implementación no es sencilla

Estructura	Criterio A	Criterio B	Criterio C	Criterio D	Criterio E	Total
Arbol binario de busqueda	2	2	2	2	2	10
Arbol n-ario	2	2	2	1	1	8
Lista doblemente ciruclar	1	1	1	1	2	6
Arbol rojo - negro	3	3	3	3	1	13
Arbol AVL	3	3	3	3	1	13

De la anterior tabla de puede inferir que las estructuras que mas se ajustan al problema planteado son los arboles rojo y negros y AVLs debido a que estas son estructuras que se autobalancean cuando se realizan operaciones como insertar o eliminar, esto permite que al momento de hacer consultas se hagan de manera eficiente es decir, no toma mucho tiempo en acceder a informacion especifica. La eleccion de estas estructuras se reflejan mediante los criterios anteriormente estipulados y vemos que en metodos basicos como son consultar lo hace en tiempo logaritmico a diferencia de las otras estructuras en disputa.

Es importante destacar que los arboles binarios de busqueda (ABB) se tendran a consideración en el programa con el fin de hacer algunas comparaciones para llegar a la demostración que utilizar estructuras autobalaceables son más eficientes de las que no son.

6. Documentación:

