Actividad 4.

Profesor: Adalberto Emmanuel Rojas Perea

Materia: Estructuras de datos

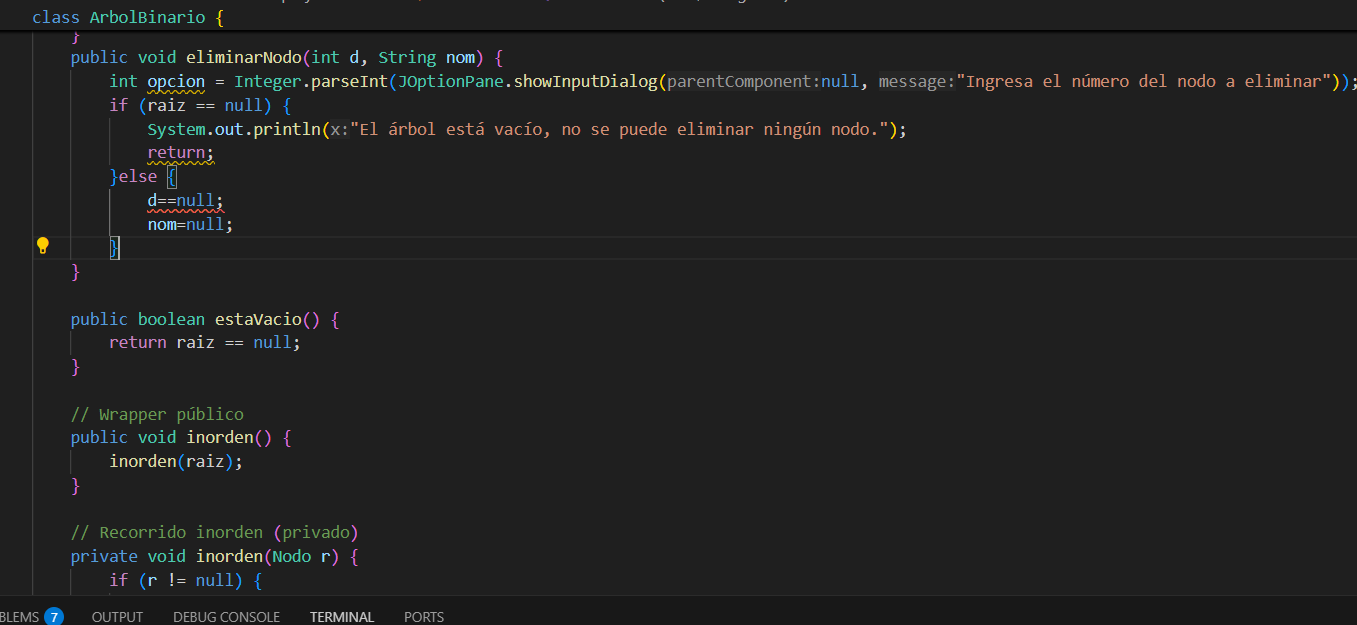
Alumno: Ricardo Baranda Cisneros



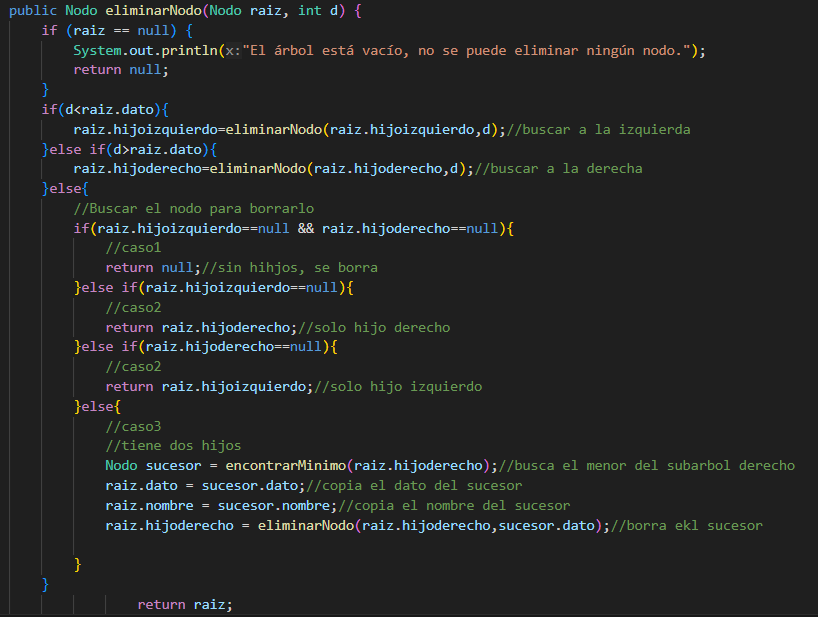
El objetivo de esta actividad es implementar un árbol binario en Java, realizando operaciones básicas (**inserción**, **eliminación**, **búsqueda**), y demostrar diferentes tipos de recorridos.

Adicionalmente, gestionar un sistema de empleados utilizando árboles binarios para optimizar la búsqueda y gestión de información de los empleados.

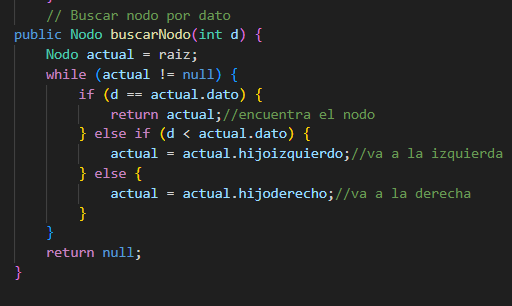
1. **Implementación del árbol binario:**



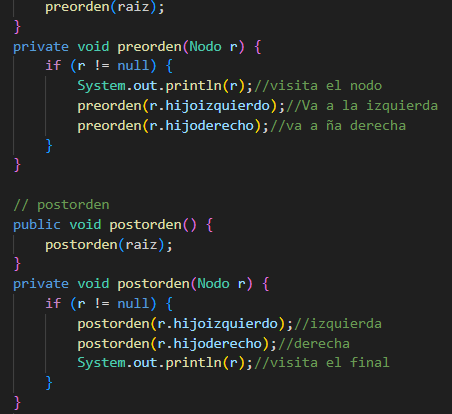
La parte principal del código venía otorgada por parte de Canvas, así que para adaptar el código se buscó una manera de eliminar los nodos en el árbol.



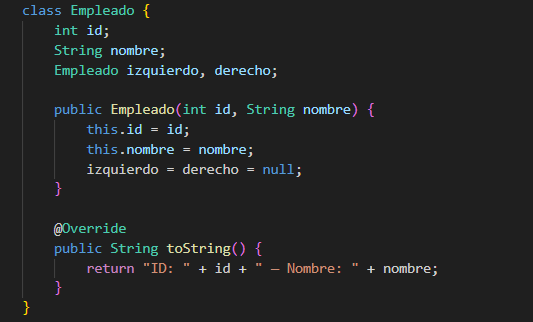
Aquí se muestra la función o método para borrar un nodo, en el cual puede haber varias situaciones como que un nodo no tenga hijos o sea una “hoja”, ese simplemente se elimina devolviendo o retornando “null”, y dependiendo si tiene solo un hijo izquierdo o derecho el nodo eliminado es reemplazado por su único hijo y el otro caso es cuando el nodo tiene dos hijos, en ese se busca un “sucesor” utilizando el “inorden”, que sería el nodo más pequeño del subárbol, entonces se copia el dato y nombre del “sucesor” al nodo actual, para después borrar al sucesor, que llevaría un caso de hoja o con un hijo, como tal método es recursivo ya que va bajando por el árbol comparando el valor que se quiere eliminar “d” con el valor del nodo actual o “raíz.dato”.



También se incluyeron los métodos para buscar un nodo, “preorden” y “postorden”.

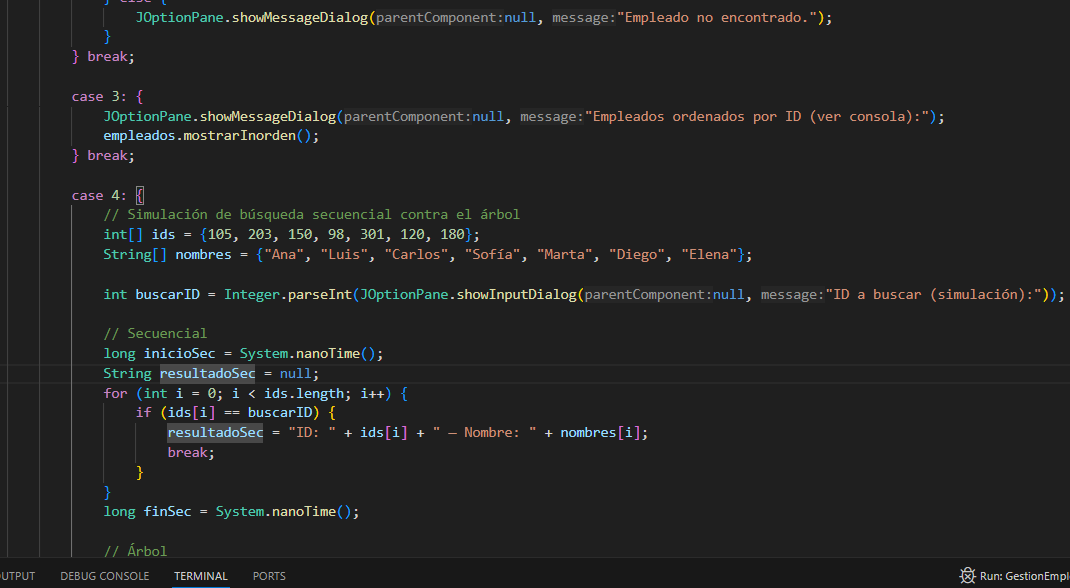


Después se empezó con el ejemplo de la gestión de empleados por medio de un “ID”.

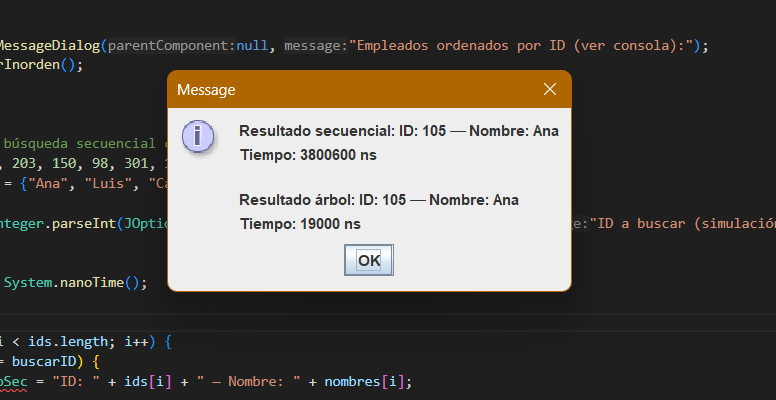


En este caso la clase empleado sería el equivalente al nodo, ya que lleva un “ID” que podría decirse que es su valor y un nombre de empleado que sería como tal el nombre del nodo, haciendo la comparación con el ejemplo del árbol binario.

Del resto el código se queda prácticamente igual, pero lo interesante de aquí es la comparación entre la búsqueda secuencial y la del árbol.



En este caso para hacer la comparación se usa un arreglo de empleados y “IDs” para comparar, pero para que la comparación pueda ser realizada se necesita agregar a uno de los empleados que están en el arreglo para comparar el tiempo de consulta. Para explicar un poco más de ambos, la búsqueda secuencial recorre cada elemento uno por uno desde el inicio hasta encontrar el buscado o llegar al final sin éxito, esto es bueno para listas desordenadas, pero es lenta si hay muchos datos, por otro lado, el árbol binario empieza en la raíz y decide si ir a la izquierda o derecha según el valor, es mucho más rápido si el árbol está balanceado, pero si requiere que los datos se organicen correctamente.



Y para recorrer y mostrar a los empleados se usa el método de “inorden” para mostrarlos de una forma de menor a mayor por sus “IDS”.

Y con esto concluye la actividad 4.