

PONTIFICIA UNIVERSIDAD JAVERIANA



ESTRUCTURAS DE DATOS

ANDREA DEL PILAR RUEDA OLARTE

INTEGRANTES

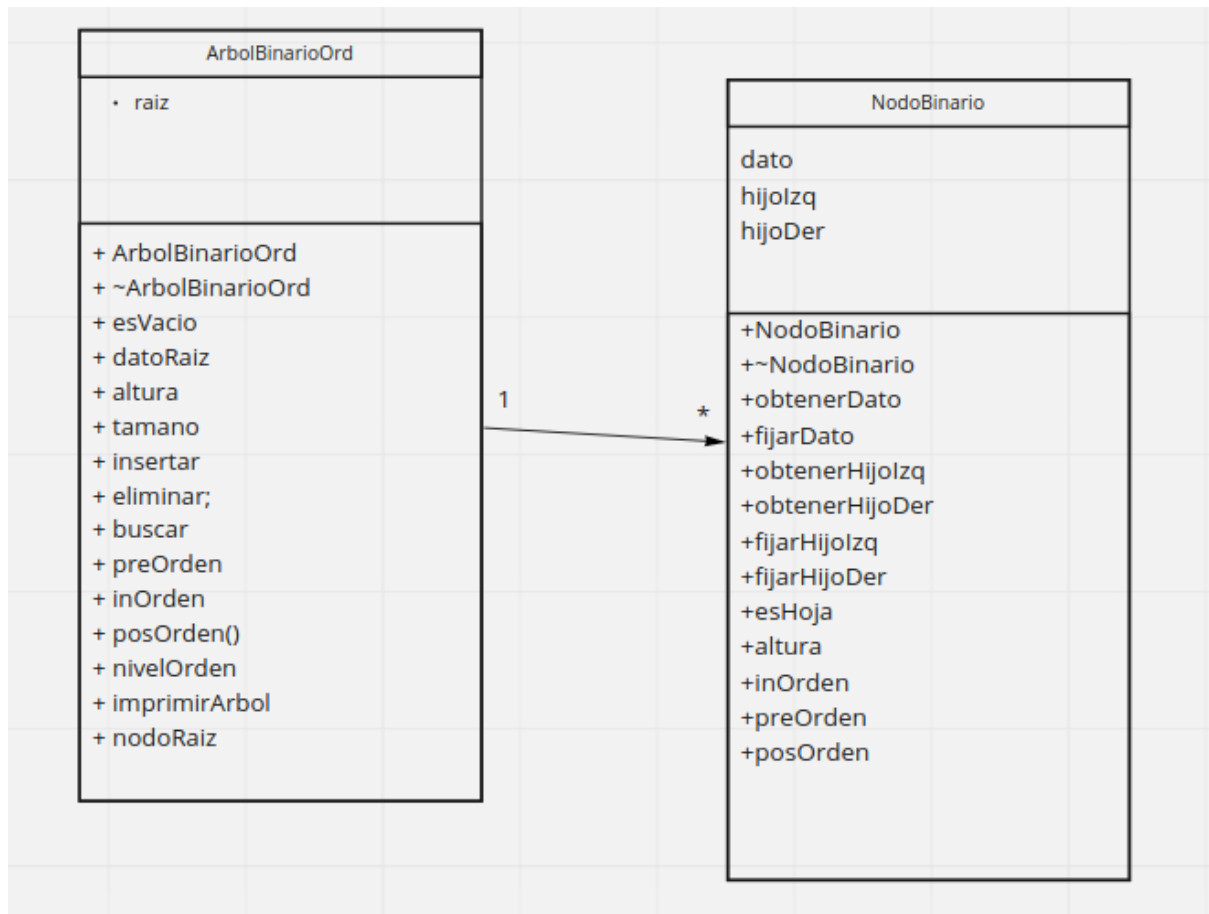
DAVID ALEJANDRO ANTOLÍNEZ SOCHA

DIEGO ALEJANDRO CARDOZO ROJAS

BRAYAN ESTIBEN GIRALDO LOPEZ

26 de marzo de 2021

DIAGRAMA DE TAD



DESCRIPCIÓN FUNCIONES

TAD Nodo binario

Datos mínimos:

- Dato, cadena de caracteres, se almacenan el código de los productos.
- HijoIzq, variable de tipo nodo, se guardaran los hijos más específicamente el hijo izquierdo cumpliendo con las reglas de un árbol binario, en caso de no tener hijos serán nulas.
- HijoDer, variable de tipo nodo, se guardaran los hijos más específicamente el hijo derecho cumpliendo con las reglas de un árbol binario, si el nodo no tiene hijos estas variables serán nulas.

Operaciones:

- NodoBinario, método constructor del tad nodo binario.
- ~NodoBinario, método de eliminación del tad nodo binario.
- ObtenerDato, función que retorna el dato tipo cadena de caracteres que se almaceno en la variable dato del tad nodo binario.
- fijarDato(valor), función que recibe una variable de tipo cadena de caracteres y la asigna a la variable dato.
- obtenerHijoIzq, función que retorna una variable tipo nodo binario y la cual corresponde a el hijo izquierdo del nodo, en caso de no tener nada retorna null.
- obtenerHijoDer, función que retorna una variable tipo nodo binario y la cual corresponde a el hijo derecho del nodo, en caso de no tener nada retorna null.
- fijarHijoIzq(hijoIzquierdo), función que recibe un dato tipo nodo binario. El dato que se envía a esta función será asignado como el hijo izquierdo del nodo.
- fijarHijoDer(hijoDerecho), función que recibe un dato tipo nodo binario. El dato que se envía a esta función será asignado como el hijo derecho del nodo.
- esHoja, esta función retorna un booleano el cual indica si el nodo que se está iterando tiene hijos o no.
- altura, función que retorna un dato de tipo integer, y calcula el camino más largo del nodo actual.
- inOrden, función que realiza una impresión siguiendo el recorrido inOrden de cada uno de los nodos.
- preOrden, función que realiza una impresión siguiendo el recorrido preOrden de cada uno de los nodos.
- posOrden, función que realiza una impresión siguiendo el recorrido posOrden de cada uno de los nodos.

TAD Árbol Binario Ordinario

Datos mínimos:

- Raiz, variable tipo nodo donde se guarda la raíz del árbol

Operaciones:

- ArbolBinarioOrd, método constructor del tad árbol binario.
- ~ArbolBinarioOrd, método de eliminación del tad árbol binario.
- esVacio, función que retorna una variable booleana en caso de que la raíz sea nula.
- datoRaiz, función que retorna el dato que se encuentre almacenado en el nodo, en este caso la variable es de tipo cadena de carácter.
- altura, función que retorna una variable de tipo entero, calcula el camino más largo del árbol y este es el valor que retorna
- tamano, función que retorna un entero el cual cuenta la cantidad de nodos que se encuentran en todo el árbol.
- insertar(valor), función que recibe una variable de tipo cadena de carácter la cual será insertada en un nuevo nodo del árbol, la función comprueba que el dato que se va a ingresar no se encuentre repetido y lo acomoda de tal forma que cumpla con un árbol ordenado.
- eliminar(valor) función que recibe una variable de tipo cadena carácter con el dato que se desea borrar, la función recorre el árbol en busca del dato una vez que encuentra el dato modifica las relaciones entre los nodos para no perder la integridad del árbol y finalmente borra el nodo.
- buscar(valor), función que recibe una variable de tipo cadena de carácter, la función recorre todos los nodos comparando sus valores hasta encontrar el dato una vez encontrado retorna una bandera positiva y si no lo encuentra retorna una bandera negativa.
- preOrden, función dentro del tad árbol que realiza una llamada a la función que se encuentra en el tad nodo y realiza impresión siguiendo el recorrido preOrden de cada uno de los nodos del árbol.
- inOrden, función dentro del tad árbol que realiza una llamada a la función que se encuentra en el tad nodo y realiza impresión siguiendo el recorrido inOrden de cada uno de los nodos del árbol.
- posOrden, función dentro del tad árbol que realiza una llamada a la función que se encuentra en el tad nodo y realiza impresión siguiendo el recorrido posOrden de cada uno de los nodos del árbol.
- nivelOrden, función dentro del tad árbol que realiza una impresión de los datos siguiendo el recorrido nivelOrden por cada uno de los nodos del árbol.

- imprimirArbol, función que recibe una variable de tipo nodo binario y un entero, la variable nodo corresponde a la raíz del árbol la cual es desde la cual se va a realizar la impresión, la variable entero solo es usada para para que la impresión se vea de una forma agradable
- nodoRaiz, función que retorna una variable de tipo nodo, la función retorna la raíz del árbol que se declaró.

Conclusión: A la hora de ingresar todos los datos a los dos árboles y obtener los resultados de tiempo, resulta evidente ver que el árbol RN es más rápido a la hora de gestionar el almacenamiento de datos, tanto en el guardado como en la eliminación de estos. Esta debe ser la principal razón por la cual la STL decidió implementar este tipo de árbol como el estándar para las funciones de Set y Map.

```
El tiempo del arbol avl fue de: 0.00075 seconds.
El tiempo del rojo negro fue de: 0.000597 seconds.

impresion arbol AVL
004763 009188 012154 056467 057354 108824 115429 116108 147885 165621 167652 168865 195060 219258 238750 242215 253748 256228 266644 281485 293439 298211
impresion arbol RN
004763 009188 012154 056467 057354 108824 115429 116108 147885 165621 167652 168865 195060 219258 238750 242215 253748 256228 266644 281485 293439 298211
```

NOTA: Profe no se pudo realizar la implementación del árbol AVL, ya que al hacer las rotaciones se nos perdían datos o teníamos errores de acceso a memoria. Perdon :C