Interfaz de usuario gráfica

Descripción generada automáticamente con confianza mediaDibujo animado de un personaje de caricatura

Descripción generada automáticamente con confianza media

**Mini-proyecto N°1**

**Nombres:**

Vicente Ignacio Ríos Adasme, 2020449783

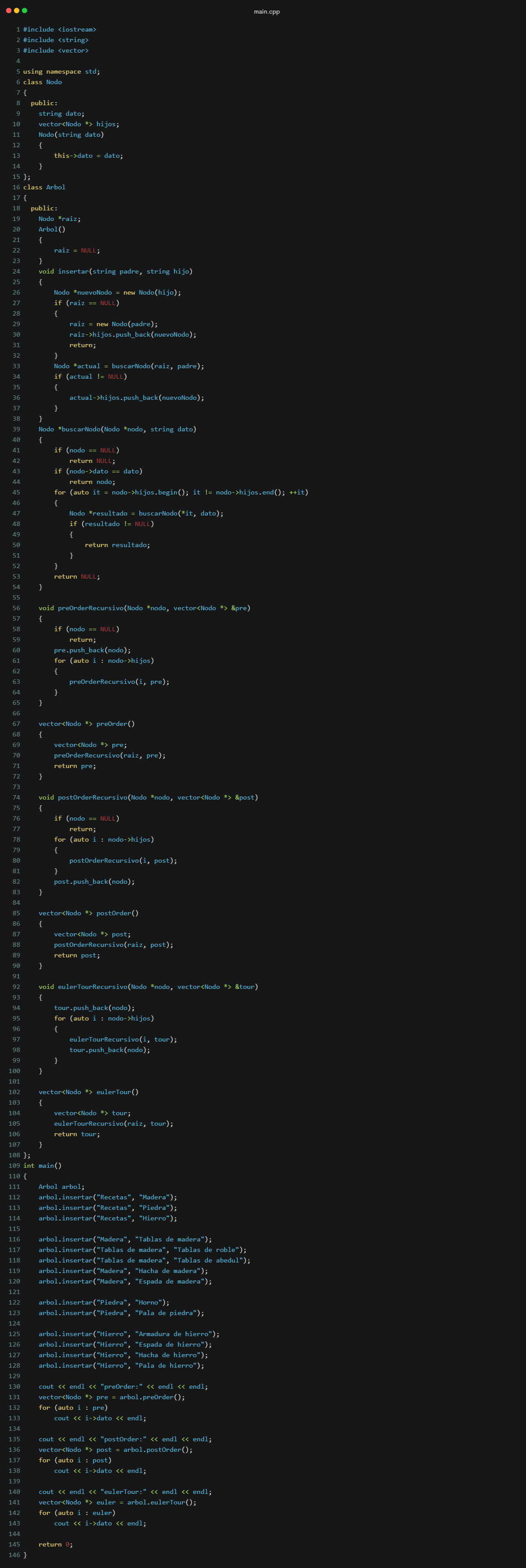
Pablo Alonso Sanhueza Yévenes, 2021439005

Lunes, 08 de mayo de 2023, Concepción

Introducción

Con el archivo main.cpp que se proporciona en las instrucciones del laboratorio, se pide implementar el recorrido *pre-order* para el árbol en cuestión para el método *preOrder().* Para esto, se crea un vector de punteros de objetos Nodo llamado “pre”, y luego se llama al método preOrderRecursivo(), este método vendría a ser el método usual de *pre-order*, donde recibe un nodo a visitar, se visita, y luego se visitan sus hijos de manera recursiva ingresando sus hijos como argumento de la misma función *preOrderRecursivo*. De esta forma, también se incluye el vector de punteros de objetos Nodo “pre”, de tal manera que para marcar un nodo como visitado, este se ingresa en el vector. Una vez se acaben todas las llamadas recursivas y se “regrese” a *preOrder(),* el método retorna el vector “pre”, luego estos valores se imprimen por pantalla. A continuación, se aprecia su implementación e impresión en pantalla.

**Script**: main.cpp



Texto

Descripción generada automáticamente

Implementación de ListArr

Se procede a implementar el recorrido *post-order* para el método *postOrder(),* de manera análoga a la anterior, se implementa un nuevo método *postOrderRecursivo()* con la implementación usual de *post-order*, marcando los nodos visitados en un vector.

**Script**: main.cpp

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Análisis teórico

Para el recorrido de Euler Tour se tiene el vector de punteros de nodos “tour” donde se marca la visita de los nodos. Se tiene, además, el método *eulerTourRecursivo()* para visitar de manera correspondiente estos nodos: Al inicio de dicho método se marca como visitado el nodo, luego, por cada hijo de aquel nodo se llama al recorrido de Euler. Cada vez que termine la llamada recursiva, nuevamente se tiene que marcar la visita del mismo nodo, de esta forma se tiene que al visitar todos los nodos descendientes de un nodo raíz de los subárboles del árbol original se vuelve a marcar la visita de dicha raíz. En palabras simples, se tiene que, dentro del vector, todos los elementos que están “encerrados” por un término diferente, entonces todos esos elementos encerrados son descendientes de aquel término diferente (entiéndanse “*término*” como variable/objeto almacenado dentro del vector).

**Script**: main.cpp

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Descripción relevante para análisis teórico

Finalmente, al compilar y ejecutar main.cpp se tiene la siguiente salida.

Análisis experimental

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Conclusiones