PONTIFICIA UNIVERSIDAD JAVERIANA

ESTRUCTURAS DE DATOS

ANDREA DEL PILAR RUEDA OLARTE

INTEGRANTES

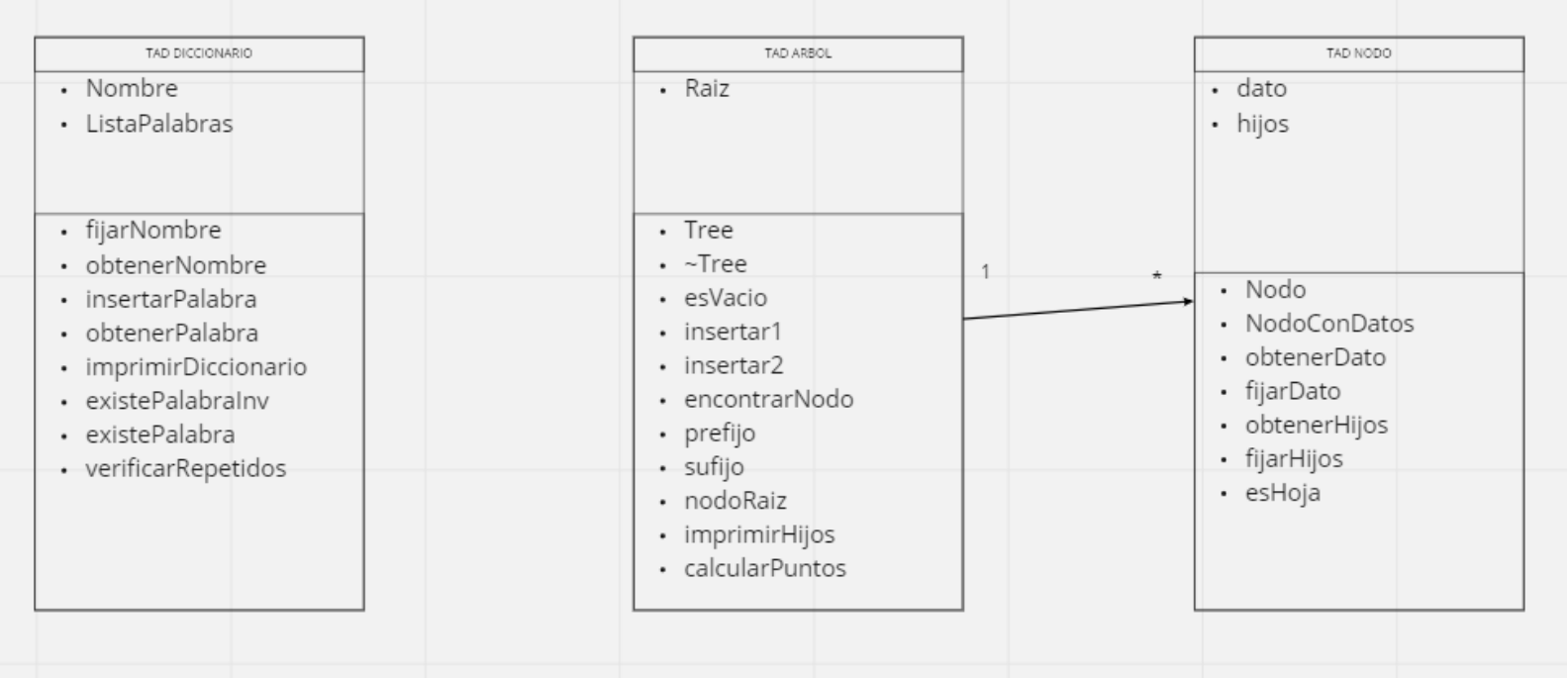
DAVID ALEJANDRO ANTOLÍNEZ SOCHA

DIEGO ALEJANDRO CARDOZO ROJAS

BRAYAN ESTIBEN GIRALDO LOPEZ

5 de marzo de 2021

DIAGRAMA DE RELACIÓN



DESCRIPCIÓN FUNCIONES

TAD diccionario

Datos mínimos:

- Nombre, cadena de caracteres, identifica el nombre del archivo diccionario que se abrió.
- Lista palabras, lista de una cadena de caracteres, almacena las palabras admitidas en el archivo diccionario.

Operaciones:

- FijarNombre(nombreP), reemplaza el nombre por la cadena nombreP.
- obtenerNombre(),retorna la cadena de caracteres almacenada en la variable nombre.
- insertarPalabra(palabraP), inserta una cadena de caracteres extraída del archivo diccionario en la lista palabras.
- obtenerPalabras(), retorna la variable lista palabras.
- imprimirDiccionario(), recorre la variable lista palabras e imprime los datos almacenados en esta.
- existePalabraInv(palabra), invierte la cadena de caracteres palabra, después de esto recorre la variable lista palabras comparando los datos almacenados en esta con la variable palabra si se encuentra un dato igual retorna un booleano.

- existePalabra(palabra), recorre la variable lista palabras comparando los datos almacenados en esta con la cadena de caracteres palabra si se encuentra un dato igual retorna un booleano.
- verificarRepetidos, función vacía que recorre el vector que almacena las palabras del archivo que se lee y las ordena de mayor a menor, una vez ordenado el arreglo se comprueba que solo exista una palabra y en caso contrario se eliminan las que están repetidas.

TAD árbol

Datos mínimos:

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

Operaciones:

- Tree, método constructor del tad árbol.
- ~Tree, método de eliminación del tad árbol.
- esVacio, función que retorna una variable booleana en caso de que la raíz sea nula.
- insertar1(lineaInsertar), funcion que retorna un booleano, esta función inicializa la raíz la cual será un carácter, este carácter inicial será un espacio, en caso de que la raíz ya fuera iniciada la función llama a la función insertar 2 y se le envía el nodo raíz junto con una cadena de caracteres que contiene la palabra que se desea agregar.
- insertar2(lineaInsertar,Nodo), función que es recurrente y asigna hijo por hijo, la función recibe una cadena de carácter con la palabra a ingresar, la cadena se recorre y se van ingresando sus datos, a medida que se ingresan los caracteres en la cadena se van eliminando hasta no quedar nada, en caso de que se vaya a ingresar un carácter ya existente este se borra directamente y no se agrega al arbol
- encontrarNodo(copiaPrefijo,nodo), función que es recurrente, recibe el prefijo y la raíz, la función busca en los hijos del nodo que se envía, un carácter que corresponda al carácter en la primera posición de la cadena una vez encontrado se elimina la primera letra de la cadena y se crea un nuevo nodo con los valores que se encontraron y se envía nuevamente a la función encontrar nodo, una vez que llega al final retorna el último nodo.
- imprimirHijos(nod,prefijo,nombreDeArchivo),la función recibe un nodo el prefijo y el nombre del archivo función que lee el archivo para verificar cuántos nodos son necesarios para una palabra una vez obtenido este dato se

calcula el puntaje de la palabra y se comienzan a recorrer los hijos del nodo para imprimir cada uno de los caracteres de la palabra.

- prefijo(prefijo,nombreArchivo), función que recibe dos cadenas de carácter,una con el prefijo y otra con el nombre del archivo, la función busca un nodo que contenga el carácter del prefijo para una vez hecho esto enviar los hijos e imprimir todas las palabras que se puedan hacer con este prefijo.
- sufijo,(sufijo, nombreArchivo), función que realiza el mismo proceso que el prefijo pero es aplicada en el árbol invertido pero usando el sufijo.
- nodoRaiz, función que retorna una variable de tipo nodo, la función retorna la raíz del árbol que se declaró
- calcularPuntos(letra),funcion que retorna un entero, recibe un carácter que al igual que en la función de puntos en el main compara este carácter y una vez lo encuentra lo retorna con su valor.

TAD nodo

Datos mínimos:

- Dato, variable tipo T, se almacena un carácter el cual corresponde a cada letra de la palabra que se ingrese del archivo diccionario.
- Hijos, variable de tipo vector, se almacena los nodos hijos <caracteres siguiente de una palabra>

Operaciones:

- Nodo, método constructor del tad nodo.
- NodoConDatos, método constructor del tad nodo que recibe un valor del dato que se va a ingresar en la estructura.
- ObtenerDato, funcion que retorna un dato tipo T que se almaceno en la variable dato del tad nodo.
- fijarDato(valor), función que recibe una variable de tipo T y la asigna a la variable dato.
- fijarHijos(hijos), función que recibe una variable de tipo vector la cual contiene todos los nodos que van a ser hijos y se asigna en la variable hijos del tad nodo
- esHoja, esta función retorna un booleano el cual indica si el nodo que se está iterando tiene hijos o no.

DESCRIPCIÓN OTRAS CLASES

Utilidades

Operaciones:

- impresionAyuda(), muestra en pantalla los comandos que puede usar el usuario.
- impresionAyudaEspecifico(valor), busca el comando enviado en la cadena de caracteres valor y cuando lo encuentra imprime la forma correcta en la que lo puede usar el usuario.

Main

Datos mínimos:

- Ciclo, entero usado como bandera para imprimir el menú.
- listaDiccionarios, lista que almacena datos de tipo TAD diccionario.

Operaciones:

- verificarPalabra(línea), comprueba que la cadena de caracteres línea está compuesta de caracteres admitidos por el programa.
- comandoInicializar(valor,listaDiccionario), comprueba que no exista ningún dato en lista diccionarios con el mismo nombre en caso de que no exista busca un archivo con el nombre enviado en la cadena de caracteres valor y comienza a insertar la información en los atributos de la variable listaDiccionario.
- comandoInicializarInverso(valor,listaDiccionario),comprueba que no exista ningún dato en lista diccionarios con el mismo nombre más la letra inv para definirlo como invertido en caso de que no exista busca un archivo con el nombre enviado en la cadena de caracteres valor, una vez encontrado le agrega las letras inv a su nombre y comienza a invertir los datos en el archvio y almacenarlos en la variable listaDiccionario.
- puntos(letra), recibe un caracter el cual se comienza a buscar en unas cadenas de caracteres que contienen las diferentes letras permitidas por el juego, una vez encontrada la letra se retorna un entero con el puntaje correspondiente a este.
- comandoPuntaje(valor,listaDiccionario), verifica que la cadena de caracteres valor se encuentre en el diccionario normal o inverso y que contenga caracteres válidos una vez que se comprueban estos dos requisitos se comienza a recorrer la cadena caracter por caracter y se envían a la función puntos y el resultado obtenido por esta se almacenan en un entero llamado puntaje el cual almacena el puntaje total de la variable valor.
- comandoIniciarArbol(valor,arbol,vector), función que recibe el nombre del archivo que se va a leer, un arbol de tipo carácter y un vector en el cual se guardan los árboles iniciados, lee los datos que se encuentran en el archivo y se le envía la cadena a la función insertar del tad árbol
- comandoIniciarArbolInverso(valor,arbolinv,vector),esta función tiene el mismo funcionamiento del árbol normal pero en este caso se le llama a la función arbolinv del tad árbol
- comandoPalabraPrefijo(valor,raiz), función que llama a la función prefijo que se encuentra en el tad árbol.
- comandoPalabraSufijo(valor,raíz), función que llama a la función sufijo que se encuentra en el tad árbol.
- comandoPosiblesPalabras(valor), **implementación de la segunda entrega.**
- comandoGrafoDePalabras(valor), **implementación de la segunda entrega.**

ACTA EVALUACIÓN ENTREGA ANTERIOR

En esta segunda entrega se perfeccionó la inserción de datos en los deque ya que había una función que cada vez que se agregaba una palabra se tenía que volverse a recorrer, motivo por el cual con archivos muy grandes tomaba mucho tiempo. Para su corrección se ingresan todos los datos y una vez insertados se llama a una función que elimina los datos repetidos.

También se agregó una descripción detallada de cada comando para nutrir la función de ayuda y que sea más fácil su entendimiento