Taller 1

Por V. Castiglia, G. Giusiano, S. Genes

Prof. José Clavijo

Lunes 17 de febrero de 2025

Índice

Introducción al problema

Se nos plantea para este proyecto de Taller 1 desarrollar un sistema para manejar ecuaciones de primer y segundo, este sistema debe permitir al usuario crear y operar una serie de funcionalidades sobre ellas.

El usuario interactúa con el programa mediante una línea de comandos, permitiendo ejecutar, luego de pasar por un control de sintaxis, los siguientes comandos:

crear: Le permite al usuario crear una nueva ecuación, con un identificador, un grado y coeficientes determinados por él, siempre y cuando, no haya otra ecuación de igual identificador en memoria.

mostrar: Lista por pantalla la totalidad de las ecuaciones que se encuentran alojadas en memoria principal.

resolver: Le devuelve al usuario la o las soluciones a una ecuación específica.

sumar: Suma dos ecuaciones de igual o distinto grado y aloja en memoria el resultado como una nueva ecuación.

guardar: Guarda una ecuación en disco en un archivo con el identificador como nombre y la extensión “.dat”.

recuperar: Dado un identificador, si existe una archivo asociado este, lee la ecuación del disco y la vuelve a cargar en memoria principal.

salir: Libera toda la memoria dinámica solicitada durante la ejecución del programa y termina el bucle de ejecución.

Análisis

Cada módulo tiene sus propias funciones y procedimientos que surgieron de la letra del problema planteado y del desarrollo del pseudocódigo, cada una está pensada para cumplir eficazmente la resolución de cada comando.

Las funciones auxiliares no están completamente definidas, ya que podrían surgir más durante el desarrollo de los requerimientos. Sin embargo, se procura minimizar su cantidad, con el fin de cumplir con los tiempos estimados.

Está previsto el testing de cada módulo del proyecto, de cada módulo se probarán funciones, procedimientos y se harán casos de prueba para asegurar el correcto funcionamiento de los mismos.

En base a la letra del problema y al análisis hecho, se eligieron las siguientes estructuras de datos principales para la resolución del proyecto:

Se utilizará un árbol binario de búsqueda para almacenar las ecuaciones, cada nodo del árbol es una ecuación, se usará el componente identificador de cada ecuación como identificador único para ordenar alfabéticamente los nodos.

Se utilizará una lista Comando en la cual se almacenarán los parámetros de cada comando, luego se haber sometido el string ingresado por el usuario a un procedimiento de parsing, cada nodo de la lista es de tipo string.

Tipos de datos

string:

```
const int MAX = 80;
typedef char * string;
```

boolean:

```
typedef enum { FALSE, TRUE }boolean;
```

Grado:

```
typedef enum { PRIMER, SEGUNDO } Grado;
```

PrimerGrado:

```
typedef struct {
        int coefA, coefB;
} PrimerGrado;
```

SegundoGrado:

```
typedef struct {
        int coefA, coefB, coefC;
} SegundoGrado;
```

Ecuación:

```
typedef struct {
        string ident;
        Grado discriminante;
        union {
                PrimerGrado primer;
                SegundoGrado segundo;
        } coeficientes;
}  Ecuacion;
```

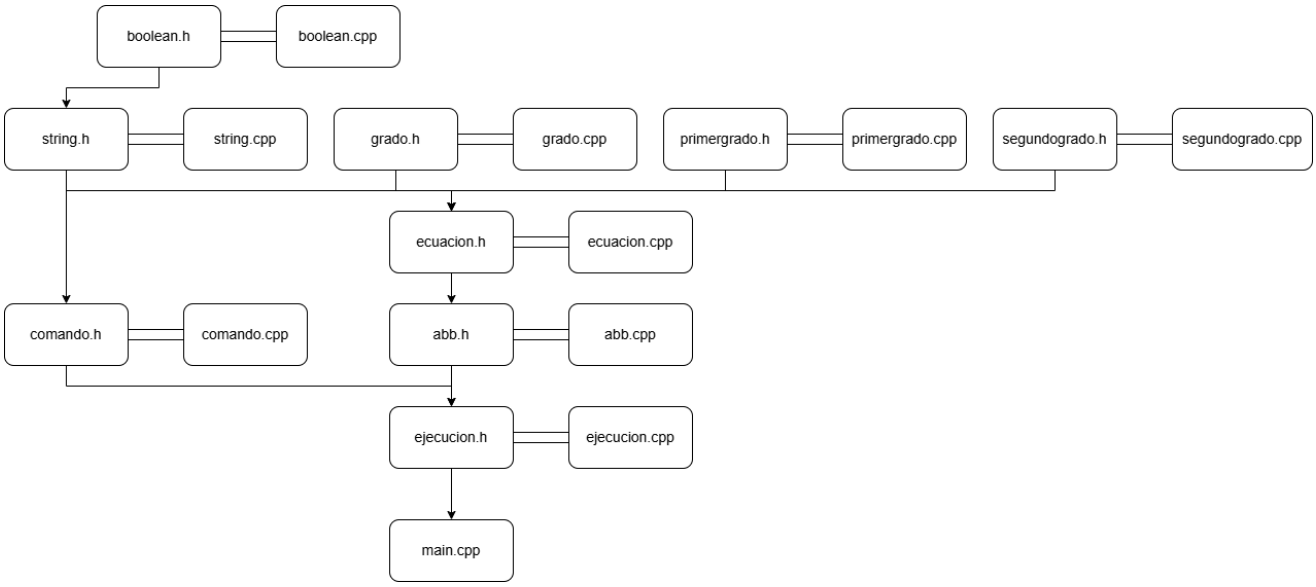
Árbol Binario de Búsqueda

```
typedef struct NodoA {
        Ecuación info;
        NodoA * hizq;
        NodoA * hder;
} NodoArbol;
typedef NodoArbol * ABBEcuaciones;
```

Comando:

```
typedef struct NodoL {
        string info;
        NodoL * sig;
} Nodo;
typedef Nodo * Comando;
```

Esquemas de módulos



Pseudocódigo

Crear: crea una nueva ecuación en memoria, a partir de sus coeficientes

Leer el comando por teclado y cargarlo en un string
Partir dicho string en una lista de strings

SI el 1er string de la lista dice “crear” ENTONCES
    Verificar si el string tiene entre 4 y 5 palabras en total
    SI no tiene entre 4 y 5 palabras ENTONCES
        error: Cantidad de parámetros incorrecta
    SINO
        Verificar si el segundo string es completamente alfabético
        SI el segundo string no es alfabético ENTONCES
            error: El nombre de la ecuación solo debe contener letras
        SINO
            Verificar que el resto de strings solo sean números enteros
            SI no son completamente números enteros ENTONCES
                error: Solo se admiten coeficientes enteros
            SINO
                Verificar que el tercer string no sea igual que cero
                SI el 3er string es igual que cero
                    error: El coeficiente principal no puede ser cero
                SINO
                    Verificar que el segundo string no sea el identificador de otra ecuación en memoria
                    SI existe otra ecuación en el ABB con el mismo identificador ENTONCES
                        error: Ya existe una ecuación con dicho identificador
                    SINO
                        Crear la ecuación y asignar el segundo string como identificador

Verificar la cantidad de strings que siguen el 2do
Convertir los strings en datos de tipo entero
SI siguen dos ENTONCES
    Asignar PRIMER al discriminante
    Asignar 3er y 4to string a la componente
    coefA y coefB de la componente “primer”
    de la union discriminada
SINO
    Asignar SEGUNDO al discriminante
    Asignar 3er, 4to y 5to string a las
    componentes coefA, coefB y coefC de la
    componente “segundo” de la union discriminada
Insertar en el ABB
Mostrar la ecuación
FIN SI
Liberar memoria de la lista

Mostrar: muestra por pantalla todas las ecuaciones existentes en memoria hasta el momento

Leer el comando por teclado y cargarlo en un string
Partir dicho string en una lista de strings

SI el 1er string de la lista dice “mostrar” ENTONCES
    Verificar si existen ecuaciones
    SI no existen ecuaciones entonces
        mensaje: No existen ecuaciones en el sistema
    SINO
        PARA CADA ecuación en memoria HACER
            (Recorro el árbol en orden)
            Desplegar su identificador
            SI el coeficiente principal es negativo ENTONCES
                Mostrar el coeficiente con un signo negativo delante de él
            SINO
                Mostrar sólo el coeficiente
            FIN
                Mostrar el valor del segundo coeficiente.
            SI el coeficiente vale 0 ENTONCES
                Avanzar al siguiente término
            FIN
            SI el coeficiente vale 1 o -1 y está delante de una variable ENTONCES
                Mostrar sólo la variable
            FIN
            SI el término no tiene variable ENTONCES
                Mostrar valor del coeficiente
            FIN
        FIN
    FIN
FIN
Liberar memoria de la lista

Resolver: resuelve una ecuación existente en memoria, mostrando su(s) solución(es)

Leer el comando por teclado y cargar en un string
Partir dicho string en una lista de strings

```
SI el 1er string de la lista dice resolver ENTONCES
        SI la lista no tiene dos palabras
                mensaje: tiene que tener exactamente dos palabras
        SINO
                obtener segundo identificador de la lista
                SI el identificador no es alfabético ENTONCES
                        mensaje: error: el parámetro ingresado debe contener solo letras
                SINO
                        SI no existe ecuación en memoria con ese identificador ENTONCES
                                mensaje: La ecuación no existe en memoria
                        SINO
                                SI la ecuación es de primer grado y coincide con el identificador
                                ingresado ENTONCES
                                        calcular y mostrar una solución o ninguna
                                SINO
                                        calcular y mostrar dos soluciones o una o ninguna
                                FIN
                        FIN
                FIN
        FIN
FIN
Liberar memoria de la lista
```

Sumar: suma dos ecuaciones existentes en memoria, creando una nueva ecuación como resultado

Leer el comando por teclado y cargarlo en un string
Partir dicho string en una lista de strings

SI el 1er string de la lista dice “sumar” ENTONCES
    Verificar si el string tiene 3 palabras más
    SI tiene más de 3 o menos de 3 ENTONCES
        Mensaje: error, la cantidad de identificadores no es la necesaria
    SINO
        Chequear que los tres identificadores sean puramente alfabéticos
        SI al menos uno no lo es ENTONCES
            Mensaje: error, hay al menos un identificador que no es alfabético
        SINO
            Verificar que las 2 primeras ecuaciones existan en memoria
            SI al menos una no existe ENTONCES
                Mensaje: error, al menos una ecuación no existe en memoria
            SINO
                Verificar que no exista una ecuación en memoria con el 3º identificador
                SI existe una ecuación llamada igual que el 3º identificador
            ENTONCES
                    Mensaje: error, ya existe una ecuación en memoria con el
                    nombre del 3º identificador
                SINO
                    Verificar que las ecuaciones a sumar sean del mismo grado
                    SI lo son ENTONCES
                        Sumar los coeficientes principales de las ecuaciones
                        SI el resultado de la suma dio como valor 0
                    ENTONCES
                            Mensaje: error, la suma de los coeficientes
                            principales es 0
                        SINO
                            Sumar los coeficientes de los términos de igual
                    grado

FIN

SINO

Sumar los coeficientes de los términos de igual grado

FIN

Almacenar la ecuación en memoria con el nombre del 3º identificador

FIN

FIN

FIN

FIN

FIN

Liberar memoria de la lista

Guardar: guarda en un archivo en disco una ecuación existente en memoria

Leer el comando por teclado y cargarlo en un string
Partir dicho string en una lista de strings

SI el 1er string de la lista dice “guardar” ENTONCES
    Verificar si el string tiene 2 palabras
    SI no tiene 2 palabras ENTONCES
        error: Cantidad de parámetros incorrecta
    SINO
        Verificar que el 2do string sea sólo alfabético
        SI no es sólo alfabético ENTONCES
            error: Los identificadores son puramente alfabéticos, verifiqué que el identificador de la ecuación sólo contiene letras
        SINO
            Verificar que en el ABB exista una ecuación con dicho identificador
            SI no existe ENTONCES
                error: No existe una ecuación con dicho identificador en memoria
            SINO
                Agregar la extensión “.dat” al 2do string
                Verificar que no exista ya en disco un archivo con ese nombre
                SI existe ENTONCES
                    error: Ya existe una ecuación en disco con este identificador.
                SINO
                    Respaldo en disco la ecuación
                    mensaje: Ecuación guardada correctamente en “identificador”.dat

FIN
Liberar memoria de la lista

Recuperar: recupera a memoria una ecuación previamente guardada en un archivo en disco

Leer el comando por teclado y cargarlo en un string
Partir dicho string en una lista de strings

SI el 1er string de la lista dice “guardar” ENTONCES
    Verificar si el string tiene 2 palabras
    SI no tiene 2 palabras ENTONCES
        error: Cantidad de parámetros incorrecta
    SINO
        Verificar que el 2do string sea sólo alfabético
        SI no es sólo alfabético ENTONCES
            error: Los identificadores son puramente alfabéticos, verifiqué que el identificador de la ecuación sólo contiene letras
        SINO
            Agregar la extensión “.dat” el 2do string
            Verificar si existe un archivo que coincida con el 2do string
            SI no existe ENTONCES
                error: No existe dicha ecuación en disco
            SINO
                Verificar que no exista una ecuación con el mismo identificador en el ABB
                SI existe una ecuación con el mismo identificador ENTONCES
                    error: Ya existe una ecuación con el mismo identificador en memoria
                SINO
                    Leer los datos del archivo y asignarlos a una ecuación auxiliar
                    Insertar dicha ecuación en el ABB
                    Mostrar ecuación
FIN
Liberar memoria de la lista

Salir: sale de la aplicación

Leer el comando por teclado y cargarlo en un string

Partir dicho string en una lista de strings

SI el 1er string de la lista dice “salir” ENTONCES

    Verificar la cantidad de palabras de la lista

    SI es mayor a 1 ENTONCES

        error: Cantidad de parámetros incorrecta. Para cerrar el programa digite simplemente “salir”

    SINO

        Liberar toda la memoria dinámica asignada al ABB y a la lista del comando

        mensaje: Hasta la próxima!

FIN

Cabezales sintácticos:

String.h

- void strcrear (string &s);
- void strdestruir (string &str);
- void strscan (string &str);
- void print (string s);
- boolean streq (string str1, string str2)
- boolean strmen (string str1, string str2);
- void strcop (string &str1, string str2);
- boolean esAlfabetico (string str)
- boolean esNumerico (string str)
- boolean distintoCero (string str)
- int convertirStringNumerico (string str)
- void agregarExtencion (string &str)
- void respaldarString (string str, FILE * file)
- void recuperarString (string &str, FILE * file)
- void AgregarCaracter (string &str, char car)

Boolean.h

Grado.h

PrimerGrado.h

- void DevolverCoeficientesP (PrimerGrado primer, int &coefa, int &coefb)
- int DevolverCoefPrimerA (PrimerGrado primer)
- int DevolverCoefPrimerB (PrimerGrado primer)
- float ResolverPrimerGrado(PrimerGrado primer)
- void MostrarPrimerGrado (PrimerGrado primer)
- PrimerGrado SumarPrimerGrado (PrimerGrado coef1, PrimerGrado coef2)

SegundoGrado.h

- void DevolverCoeficientesS (SegundoGrado segundo, int &coefa, int &coefb, int &coefc)
- int DevolverCoefSegundoA (SegundoGrado segundo)
- int DevolverCoefSegundoB (SegundoGrado segundo)
- int DevolverCoefSegundoC (SegundoGrado segundo)
- int NmroDeSoluciones (SegundoGrado segundo)
- void ResolverSegundoGrado2Soluciones (SegundoGrado segundo, float &x1, float &x2)
- float ResloverSegundoGrado1Solucion (SegundoGrado segundo)
- void MostrarSegundoGrado (SegundoGrado seg)
- SegundoGrado SumarSegundoGrado (SegundoGrado seg1, SegundoGrado seg2)

Ecuacion.h

- void mostrarEcuacion (Ecuacion ecu)
- Grado DevolverGrado (Ecuacion ecu)
- void DevolverIdentificador (Ecuacion ecu, string &str)
- void DevolverCoeficientesPrimer (Ecuacion ecu, int &coefa, int &coefb)
- void DevolverCoeficientesSegundo (Ecuacion ecu, int &coefa, int &coefb, int &coefc)
- boolean mismoGrado (Ecuacion ecu1, Ecuacion ecu2)
- void RespaldarEcuacion (Ecuacion ecu , string nombreArchivo)
- Ecuacion RecuperarEcuacion (string nomArch)
- Ecuacion CrearEcuacionPrimerGrado (PrimerGrado coef, string iden)
- Ecuacion CrearEcuacionSegundoGrado (SegundoGrado coef, string iden)
- Ecuacion SumarEcucionesDiferenteGrado (SegundoGrado ecu1, PrimerGrado ecu2, string newident)

ArbolBinarioDeBusqueda.h

- boolean existeIdentificador (ABBEcuaciones abb, string str)
- void insertarEcuacion (ABBEcuaciones &abb, Ecuacion ecu)
- void mostrarABBEcuaciones (ABBEcuaciones abb)
- boolean esVacio (ABBEcuaciones abb)

- void crearAbb (ABBEcuaciones &abb)
- void vaciarAbb (ABBEcuaciones &abb)
- Ecuacion obtenerEcuacion (ABBEcuaciones abb, string id)

Comando.h

- void partirString (Comando &com, string str)
- boolean restoListaesNumerica (Comando com)
- int cantidadStrings (Comando com)
- string obtenerStringComando (Comando com, int pos)
- void crearLista (Comando &com)
- void vaciarLista (Comando &com)
- void InsertarString (Comando &com, string str)
- boolean ComadoVacio (Comando com)

Ejecucion.h

- boolean existeArchivo (const int char nombreArchivo)
- void EjecutarCrear (Comando parametros, ABBEcuaciones &abb)
- void EjecutarMostrar (Comando parametros, ABBEcuaciones abb)
- void EjecutarResolver (Comando parametros, ABBEcuaciones abb)
- void EjecutarSumar (Comando parametros, ABBEcuaciones &abb)
- void EjecutarGuardar (Comando parametros, ABBEcuaciones abb)
- void EjecutarRecuperar (Comando parametros, ABBEcuaciones &abb)
- void EjecutarSalir (Comando &parametros, ABBEcuaciones &abb, boolean &salir)

Diagrama de Gantt