|  |  |  |
| --- | --- | --- |
| **CARRERA:**  Ingeniería en Software | **GUÍA**  No. 01 | **TIEMPO ESTIMADO:** |
| **ASIGNATURA:**  Estructura de datos  NRC: 2967 | **FECHA DE ELABORACION:** 27-01-2020  **SEMESTRE**: septiembre 2019 – enero 2020 | |
| **TÍTULO:**  Proyecto Tercer Parcial (Diccionario) | **DOCENTE:** Ing. Fernando Solís | |

**OBJETIVO**

Aplicar el conocimiento de las estructuras de datos “Árboles binarios de búsqueda” en el desarrollo de un Diccionario ESP-ING.

**INSTRUCCIONES**

1. Utilice como material principal, las principales herramientas para desarrollo en C++
2. Utilice información consultada en Internet y conocimiento adquirido en clase.

**ACTIVIDADES**

1. **Ubicación de recursos**
2. Formar grupos de máximo 2 personas por computador
3. Realizar el programa en cualquier IDE para C++
4. **Planteamiento del problema**

**Ejercicio No 1**

Desarrollar un programa que aplique los conocimientos aprendidos en el tercer parcial, principalmente “Árboles” y recursividad. Utilizar memoria dinámica. Se deben mostrar la traducción de la palabra introducida.

1. **Marco Teórico**

El diccionario bilingüe de lengua general está destinado a la búsqueda de equivalentes para una palabra en otra lengua. Este tipo de diccionario es frecuentemente bidireccional (ej. inglés-español, español-inglés) y generalmente se organiza de forma alfabética.

****

**¿Como resolverlo con árboles binarios de búsqueda?**

Se ha pedido que la palabra ingresada se muestre traducida al otro idioma para ello se guardara en un archivo plano las palabras, sin embargo, se pueden introducir más palabras a este archivo, cada palabra se ingresara a un árbol binario haciendo que la búsqueda sea más eficiente.

Se ha utilizado varias funciones recursivas que verifiquen la búsqueda de cada palabra haciendo este problema eficiente.

1. **Código del proyecto**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* UNIVERSIDAD DE LAS FUERZAS ARMADAS \*

\* ESPE \*

\*TRABAJO EN GRUPO: \*

\* NOMBRES:ANTONI TOAPANTA \*

\* JHONY NARANJO \*

\*MATERIA: ESTRUCTURA DE DATOS \*

\*NRC:2967 \*

\*Fecha de Creacion:26/01/2020 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <iostream>

#include <stdlib.h>

#include <conio.h>

#include <windows.h>

#include <ctype.h>

#include <string.h>

#include <fstream>

#include <string>

#include <time.h>

#include <pthread.h>

#include <fstream>

**using** **namespace** std**;**

#define TECLA\_ARRIBA 72

#define TECLA\_ABAJO 80

#define TECLA\_DERECHA 77

#define TECLA\_IZQUIERDA 75

#define TECLA\_ENTER 13

void gotoxy**(**int x**,** int y**)**

**{**

HANDLE hCon**;**

hCon**=**GetStdHandle**(**STD\_OUTPUT\_HANDLE**);**

COORD dwPos**;**

dwPos**.**X**=**x**;**

dwPos**.**Y**=**y**;**

SetConsoleCursorPosition**(**hCon**,**dwPos**);**

**}**

void color**(**int x**)**

**{**

SetConsoleTextAttribute**(**GetStdHandle **(**STD\_OUTPUT\_HANDLE**),**x**);**

**}**

int menu**(**const char **\***titulo**,**const char **\***opciones**[],** int numero**)**

**{**

bool repite**=true;**

int seleccionar**=**1**,**tecla**;**

int i**;**

system**(**"cls"**);**

**do**

**{**

gotoxy**(**10**,**2**);**

color**(**158**);**

printf**(**"================================================"**);**

color**(**11**);**

printf**(**"\n\t\t\t%s\t\t\t\n\n"**,**titulo**);**

color**(**15**);** //letras en blanco

**for(**i**=**0**;**i**<**numero**;**i**++)**

printf**(**"\t\t%s\n"**,**opciones**[**i**]);**

gotoxy**(**16**,**4**+**seleccionar**);** //17

color**(**249**);**

//printf("%c %s",175,opciones[seleccionar-1]); //impresion con asterisco

printf**(**"%s"**,**opciones**[**seleccionar**-**1**]);**

color**(**158**);**

gotoxy**(**10**,**15**);**

printf**(**"================================================"**);**

color**(**15**);**

**do**

**{**

tecla**=**getch**();**

**}while(**tecla**!=**TECLA\_ARRIBA **&&** tecla**!=**TECLA\_ABAJO **&&** tecla**!=** TECLA\_ENTER **&&** tecla **!=**00**);**

**switch(**tecla**)**

**{**

**case** TECLA\_ARRIBA**:**

**{**

seleccionar**--;**

**if(**seleccionar**==**0**)**

seleccionar**=**numero**;**

**}**

**break;**

**case** TECLA\_ABAJO**:**

**{**

seleccionar**++;**

**if(**seleccionar**==**numero**+**1**)**

seleccionar**=**1**;**

**}**

**break;**

**case** TECLA\_ENTER**:**

repite**=false;**

**break;**

**case** 0**:**

ShellExecute**(NULL,** TEXT**(**"open"**),**TEXT**(**"Extras\\Ayuda.chm"**),NULL,** **NULL,**SW\_SHOWNORMAL**);**

**break;**

**}**

**}while(**repite**);**

color**(**15**);**

**return** seleccionar**;**

**}**

void AltEnter**()**

**{**

keybd\_event**(**VK\_MENU**,**

0x38**,**

0**,**

0**);**

keybd\_event**(**VK\_RETURN**,**

0x1c**,**

0**,**

0**);**

keybd\_event**(**VK\_RETURN**,**

0x1c**,**

KEYEVENTF\_KEYUP**,**

0**);**

keybd\_event**(**VK\_MENU**,**

0x38**,**

KEYEVENTF\_KEYUP**,**

0**);**

**}**

void imprimirTXT**(**char txt**[])**

**{**

string frase**;**

ifstream ficheroEntrada**;**

color**(**10**);**

ficheroEntrada**.**open**(**txt**);**

**while** **(!**ficheroEntrada**.**eof**()** **)**

**{**

getline**(**ficheroEntrada**,** frase**);**

cout **<<** frase **<<** endl**;**

**}**

ficheroEntrada**.**close**();**

**}**

bool validacionCaracter**(**char cadena**[]){**

int i**;**

**for(**i**=**0**;**cadena**[**i**]!=**'\0'**;**i**++)**

**{**

**if((**cadena**[**i**]<**65**||**cadena**[**i**]>**90**)&&(**cadena**[**i**]<**97**||**cadena**[**i**]>**122**)&&**cadena**[**i**]!=**32**&&**cadena**[**i**]!=-**92**&&**cadena**[**i**]!=-**91**)**

**{**

printf**(**"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\nDato Incorrecto !!\nNo puede contener caracteres especiales, ni numeros\nVuelva a ingresar por favor\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n"**);**

**return** **true;**

**break;**

**}**

**}**

**return** **false;**

**}**

void guardarNombre**(**string nombre**)**

**{**

ofstream arch**;**

arch**.**open**(**"Nombre.txt"**,**ios**::**out**);**

arch**<<**nombre**;**

arch**.**close**();**

**}**

string lectura**()**

**{**

ifstream archivo**;**

string texto**;**

archivo**.**open**(**"Datos.txt"**,**ios**::**in**);**

**if(**archivo**.**fail**())**

**{**

exit**(**1**);**

**}**

**else{**

**while(!**archivo**.**eof**())**

**{**

getline**(**archivo**,**texto**);**

**}**

**}**

archivo**.**close**();**

**return** texto**;**

**}**

void codificar**(**string **&**nombre**,**int i**)**

**{**

**do**

**{**

**if(**i**%**2**==**0**)**

**{**

nombre**.**at**(**i**)=(**nombre**.**at**(**i**)+**6**)+**nombre**.**length**();**

**}**

**else{**

nombre**.**at**(**i**)=(**nombre**.**at**(**i**)+**3**)-**nombre**.**length**();**

**}**

i**++;**

**}while(**i**!=**nombre**.**length**());**

**}**

void decodificar**(**string **&**nombre**,**int i**)**

**{**

**do**

**{**

**if(**i**%**2**==**0**)**

**{**

nombre**.**at**(**i**)=(**nombre**.**at**(**i**)-**6**)-**nombre**.**length**();**

**}**

**else{**

nombre**.**at**(**i**)=(**nombre**.**at**(**i**)-**3**)+**nombre**.**length**();**

**}**

i**++;**

**}while(**i**!=**nombre**.**length**());**

**}**

void archivoC**(**string nombre**)**

**{**

ofstream archivo**;**

archivo**.**open**(**"codificar.txt"**,**ios**::**out**);**

**if(**archivo**.**fail**())**

**{**

cout**<<**"No se pudo Abrir"**<<**endl**;**

exit**(**1**);**

**}**

**else{**

archivo**<<**nombre**;**

**}**

archivo**.**close**();**

**}**

void archivoD**(**string nombre**)**

**{**

ofstream archivo**;**

archivo**.**open**(**"decodificar.txt"**,**ios**::**out**);**

**if(**archivo**.**fail**())**

**{**

cout**<<**"No se pudo Abrir"**<<**endl**;**

exit**(**1**);**

**}**

**else{**

archivo**<<**nombre**;**

**}**

archivo**.**close**();**

**}**

#ifndef TRAFORMACIONSTRING\_H\_INCLUDED

#define TRAFORMACIONSTRING\_H\_INCLUDED

#include<string>

#include<stdlib.h>

#include <sstream>

#include<iostream>

**using** **namespace** std**;**

char**\***trasformarStringChar**(**string str**)**

**{**

char **\***c**=** **new** char**(**str**.**length**());**

c**=(**char**\*)**str**.**c\_str**();**

**return** c**;**

**}**

int**\***arregloEntero**(**string str**)**

**{**

int **\***e **=** **new** int**(**str**.**length**());**

**for(**int i**=**0**;** i**<**str**.**length**();**i**++)**

**{**

**\*(**e**+**i**)=(**int**)**str**.**at**(**i**)-**48**;**

**}**

**return** e**;**

**}**

int stringAint**(**string str**)**

**{**

**return** atoi**(**str**.**c\_str**());**

**}**

template **<**class Dato**>**

string arreglosAstring**(**Dato **\***arrayDato**,**int tam**)**

**{**

ostringstream aux**;**

**for(**int i**=**0**;**i**<**tam**;**i**++)**

**{**

aux**<<\*(**arrayDato**+**i**);**

**}**

**return** aux**.**str**();**

**}**

#endif // TRAFORMACIONSTRING\_H\_INCLUDED

#include <stdio.h>

#include <string.h>

#include <windows.h> // Win32Api Header File

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <iostream>

#include <string>

#include<fstream>

#include <iostream>

#include <conio.h>

#include <string>

#include <locale.h>

#include <stdio.h>

#include <stdlib.h>

#include <windows.h>

#include <time.h>

#include <ctype.h>

#define KEY\_ESC 27

**using** **namespace** std**;**

static HWND hConWnd**;**

HWND BCX\_Bitmap**(**char**\*,** HWND **=** 0**,** int **=** 0**,** int **=** 0**,** int **=** 0**,** int **=** 0**,** int **=** 0**,** int **=** 0**,** int **=** 0**,** int **=** 0**);**

HWND GetConsoleWndHandle**(**void**);**

int main**()**

**{**

hConWnd **=** GetConsoleWndHandle**();**

**if** **(**hConWnd**)**

**{**

// select a bitmap file you have or use one of the files in the Windows folder

// filename, handle, ID, ulcX, ulcY, width, height 0,0 auto-adjusts

BCX\_Bitmap**(**"LOGO.bmp"**,** hConWnd**,** 123**,**1**,**1**,**0**,**0**);**

Sleep**(**8000**)** **;**

**}**

//system("pause");

**return** 0**;**

**}**

// draw the bitmap

HWND BCX\_Bitmap**(**char**\*** Text**,** HWND hWnd**,** int id**,** int X**,** int Y**,** int W**,** int H**,** int Res**,** int Style**,** int Exstyle**)**

**{**

HWND A**;**

HBITMAP hBitmap**;**

// set default style

**if** **(!**Style**)** Style **=** WS\_CLIPSIBLINGS **|** WS\_CHILD **|** WS\_VISIBLE **|** SS\_BITMAP **|** WS\_TABSTOP**;**

// form for the image

A **=** CreateWindowEx**(**Exstyle**,** "static"**,** **NULL,** Style**,** X**,** Y**,** 0**,** 0**,** hWnd**,** **(**HMENU**)**id**,** GetModuleHandle**(**0**),** **NULL);**

// Text contains filename

hBitmap **=** **(**HBITMAP**)**LoadImage**(**0**,** Text**,** IMAGE\_BITMAP**,** 0**,** 0**,** LR\_LOADFROMFILE **|** LR\_CREATEDIBSECTION**);**

// auto-adjust width and height

**if** **(**W **||** H**)** hBitmap **=** **(**HBITMAP**)**CopyImage**(**hBitmap**,** IMAGE\_BITMAP**,** W**,** H**,** LR\_COPYRETURNORG**);**

SendMessage**(**A**,** **(**UINT**)**STM\_SETIMAGE**,** **(**WPARAM**)**IMAGE\_BITMAP**,** **(**LPARAM**)**hBitmap**);**

**if** **(**W **||** H**)** SetWindowPos**(**A**,** HWND\_TOP**,** X**,** Y**,** W**,** H**,** SWP\_DRAWFRAME**);**

**return** A**;**

**}**

// tricking Windows just a little ...

HWND GetConsoleWndHandle**(**void**)**

**{**

HWND hConWnd**;**

OSVERSIONINFO os**;**

char szTempTitle**[**64**],** szClassName**[**128**],** szOriginalTitle**[**1024**];**

os**.**dwOSVersionInfoSize **=** **sizeof(**OSVERSIONINFO**);**

GetVersionEx**(&**os**);**

// may not work on WIN9x

**if** **(**os**.**dwPlatformId **==** VER\_PLATFORM\_WIN32s**)** **return** 0**;**

GetConsoleTitle**(**szOriginalTitle**,** **sizeof(**szOriginalTitle**));**

sprintf**(**szTempTitle**,** "%u - %u"**,** GetTickCount**(),** GetCurrentProcessId**());**

SetConsoleTitle**(**szTempTitle**);**

Sleep**(**60**);**

// handle for NT and XP

hConWnd **=** FindWindow**(NULL,** szTempTitle**);**

SetConsoleTitle**(**szOriginalTitle**);**

// may not work on WIN9x

**if** **(**os**.**dwPlatformId **==** VER\_PLATFORM\_WIN32\_WINDOWS**)**

**{**

hConWnd **=** GetWindow**(**hConWnd**,** GW\_CHILD**);**

**if** **(**hConWnd **==** **NULL)** **return** 0**;**

GetClassName**(**hConWnd**,** szClassName**,** **sizeof(**szClassName**));**

// while ( \_stricmp( szClassName, "ttyGrab" ) != 0 )

**while** **(**strcmp**(**szClassName**,** "ttyGrab"**)** **!=** 0**)**

**{**

hConWnd **=** GetNextWindow**(**hConWnd**,** GW\_HWNDNEXT**);**

**if** **(**hConWnd **==** **NULL)** **return** 0**;**

GetClassName**(**hConWnd**,** szClassName**,** **sizeof(**szClassName**));**

**}**

**}**

**return** hConWnd**;**

**}**

#pragma once

#include "Palabra.h"

string Palabra**::**getEspanol**(**void**)**

**{**

**return** **this->**espanol**;**

**}**

void Palabra**::**setEspanol**(**string newEspanol**)**

**{**

**this->**espanol **=** newEspanol**;**

**}**

string Palabra**::**getIngles**(**void**)**

**{**

**return** **this->**ingles**;**

**}**

void Palabra**::**setIngles**(**string newIngles**)**

**{**

**this->**ingles **=** newIngles**;**

**}**

int Palabra**::**getClave**(**void**)**

**{**

**return** **this->**clave**;**

**}**

void Palabra**::**setClave**(**int newClave**)**

**{**

**this->**clave **=** newClave**;**

**}**

void Palabra**::**setClaveE**(**int newClave**)**

**{**

**this->**claveE **=** newClave**;**

**}**

int Palabra**::**getClaveE**(**void**)**

**{**

**return** **this->**claveE**;**

**}**

Palabra**::**Palabra**()**

**{**

**}**

Palabra**::~**Palabra**()**

**{**

// TODO : implement

**}**

#pragma once

#if !defined(\_\_Class\_Diagram\_1\_Palabra\_h)

#define \_\_Class\_Diagram\_1\_Palabra\_h

#include<string>

**using** **namespace** std**;**

class Palabra

**{**

public**:**

string getEspanol**(**void**);**

void setEspanol**(**string newEspanol**);**

string getIngles**(**void**);**

void setIngles**(**string newIngles**);**

int getClave**(**void**);**

void setClave**(**int newClave**);**

int getClaveE**(**void**);**

void setClaveE**(**int newClave**);**

Palabra**();**

**~**Palabra**();**

protected**:**

private**:**

string espanol**;**

string ingles**;**

int clave**;**

int claveE**;**

**};**

#endif

#include <iostream>

#include <string>

#include <stdlib.h>

**using** **namespace** std**;**

class Ingreso**{**

public**:**

float ingresarFloat**(**char**\*);**

bool validarFloat**(**string**);**

int ingresarInt**(**char**\*);**

bool validarInt**(**string**);**

string ingresarString**(**char**\*);**

bool validarString**(**string**);**

string ingresar10Digitos**(**char**\*);**

bool validar10Digitos**(**string**);**

**};**

float Ingreso**::**ingresarFloat**(**char**\*** msg**){**

string flotante**;**

bool flag**;**

**do{**

**try{**

cout**<<**msg**;**

getline**(**cin**,**flotante**);**

flag **=** validarFloat**(**flotante**);**

**if(**flag**){**

**throw** flotante**;**

**}**

**}catch(**string e**){**

cout**<<**"Dato invalido "**<<**e**<<**endl**;**

**}**

**}while(**flag**);**

**return** atof**(**flotante**.**c\_str**());**

**}**

bool Ingreso**::**validarFloat**(**string valor**){**

bool flag**;**

**for(**int i **=** 0**;** i**<**valor**.**length**();** i**++){**

**if(**isdigit**(**valor**[**i**])){**

flag **=** **false;**

**}else{**

**if(**valor**[**i**]==** '.'**){**

flag **=** **false;**

**}else{**

flag **=** **true;**

**break;**

**}**

**}**

**}**

**return** flag**;**

**}**

int Ingreso**::**ingresarInt**(**char**\*** msg**){**

string numero**;**

bool flag**;**

**do{**

**try{**

cout**<<**msg**;**

getline**(**cin**,**numero**);**

flag **=** validarInt**(**numero**);**

**if(**flag**){**

**throw** numero**;**

**}**

**}catch(**string e**){**

cout**<<**"Dato invalido "**<<**e**<<**endl**;**

**}**

**}while(**flag**);**

**return** atoi**(**numero**.**c\_str**());**

**}**

bool Ingreso**::**validarInt**(**string valor**){**

bool flag**;**

**for(**int i **=** 0**;** i**<**valor**.**length**();** i**++){**

**if(**isdigit**(**valor**[**i**])){**

flag **=** **false;**

**}else{**

**return** **true;**

**}**

**}**

**return** flag**;**

**}**

string Ingreso**::**ingresarString**(**char**\*** msg**){**

string dato**;**

bool flag**;**

**do{**

**try{**

cout**<<**msg**;**

getline**(**cin**,**dato**);**

flag **=** validarString**(**dato**);**

**if(**flag**){**

**throw** dato**;**

**}**

**}catch(**string e**){**

cout**<<**"Dato invalido "**<<**e**<<**endl**;**

**}**

**}while(**flag**);**

**return** dato**;**

**}**

bool Ingreso**::**validarString**(**string valor**){**

bool flag**;**

**for(**int i **=** 0**;** i**<**valor**.**length**();** i**++){**

**if(**isalpha**(**valor**[**i**])** **||** valor**[**i**]** **==** ' '**){**

flag **=** **false;**

**}else{**

**return** **true;**

**}**

**}**

**return** flag**;**

**}**

string Ingreso**::**ingresar10Digitos**(**char**\*** msg**){**

string dato**;**

bool flag**;**

**do{**

**try{**

cout**<<**msg**;**

getline**(**cin**,**dato**);**

flag **=** validar10Digitos**(**dato**);**

**if(**flag**){**

**throw** dato**;**

**}**

**}catch(**string e**){**

cout**<<**"Dato invalido "**<<**e**<<**endl**;**

**}**

**}while(**flag**);**

**return** dato**;**

**}**

bool Ingreso**::**validar10Digitos**(**string valor**){**

bool flag**;**

int cont **=** 0**;**

**for(**int i **=** 0**;** i**<**valor**.**length**();** i**++){**

**if(**isdigit**(**valor**[**i**])){**

cont**++;**

**if(**cont **==** 10**){**

flag **=** **false;**

**}else{**

flag **=** **true;**

**}**

**}else{**

**return** **true;**

**}**

**}**

**return** flag**;**

**}**

#ifndef TRAFORMACIONSTRING\_H\_INCLUDED

#define TRAFORMACIONSTRING\_H\_INCLUDED

#include<string>

#include<stdlib.h>

#include <sstream>

#include<iostream>

**using** **namespace** std**;**

char**\***trasformarStringChar**(**string str**)**

**{**

char **\***c**=** **new** char**(**str**.**length**());**

c**=(**char**\*)**str**.**c\_str**();**

**return** c**;**

**}**

int**\***arregloEntero**(**string str**)**

**{**

int **\***e **=** **new** int**(**str**.**length**());**

**for(**int i**=**0**;** i**<**str**.**length**();**i**++)**

**{**

**\*(**e**+**i**)=(**int**)**str**.**at**(**i**)-**48**;**

**}**

**return** e**;**

**}**

int stringAint**(**string str**)**

**{**

**return** atoi**(**str**.**c\_str**());**

**}**

template **<**class Dato**>**

string arreglosAstring**(**Dato **\***arrayDato**,**int tam**)**

**{**

ostringstream aux**;**

**for(**int i**=**0**;**i**<**tam**;**i**++)**

**{**

aux**<<\*(**arrayDato**+**i**);**

**}**

**return** aux**.**str**();**

**}**

#endif // TRAFORMACIONSTRING\_H\_INCLUDED

#ifndef TRANSFORMACIONSTRING\_H

#define TRANSFORMACIONSTRING\_H

#pragma once

#include<string>

#include<stdlib.h>

#include <sstream>

#include<iostream>

**using** **namespace** std**;**

class Transformacion**{**

public**:**

char**\*** trasformarStringChar**(**string str**);**

int**\*** arregloEntero**(**string str**);**

int stringAint**(**string str**);**

**};**

#endif // TRANSFORMACIONSTRING\_H

#pragma once

#include<string>

#include<stdlib.h>

#include <sstream>

#include<iostream>

#include "TransformacionString.h"

**using** **namespace** std**;**

char**\*** Transformacion**::**trasformarStringChar**(**string str**)**

**{**

char **\***c**=** **new** char**(**str**.**length**());**

c**=(**char**\*)**str**.**c\_str**();**

**return** c**;**

**}**

int**\*** Transformacion**::**arregloEntero**(**string str**)**

**{**

int **\***e **=** **new** int**(**str**.**length**());**

**for(**int i**=**0**;** i**<**str**.**length**();**i**++)**

**{**

**\*(**e**+**i**)=(**int**)**str**.**at**(**i**)-**48**;**

**}**

**return** e**;**

**}**

int Transformacion**::**stringAint**(**string str**)**

**{**

**return** atoi**(**str**.**c\_str**());**

**}**

template **<**class Dato**>**

string arreglosAstring**(**Dato **\***arrayDato**,**int tam**)**

**{**

ostringstream aux**;**

**for(**int i**=**0**;**i**<**tam**;**i**++)**

**{**

aux**<<\*(**arrayDato**+**i**);**

**}**

**return** aux**.**str**();**

**}**

#ifndef TIE\_H

#define TIE\_H

#include <QDialog>

**namespace** Ui **{**

class TIE**;**

**}**

class TIE **:** public QDialog

**{**

Q\_OBJECT

public**:**

explicit TIE**(**QWidget **\***parent **=** **nullptr);**

**~**TIE**();**

private slots**:**

void on\_pushButton\_clicked**();**

private**:**

Ui**::**TIE **\***ui**;**

**};**

#endif // TIE\_H

#pragma once

#ifndef TSI\_H

#define TSI\_H

#include <QDialog>

**namespace** Ui **{**

class Tsi**;**

**}**

class Tsi **:** public QDialog

**{**

Q\_OBJECT

public**:**

explicit Tsi**(**QWidget **\***parent **=** **nullptr);**

**~**Tsi**();**

private slots**:**

void on\_pushButton\_clicked**();**

private**:**

Ui**::**Tsi **\***ui**;**

**};**

#endif // TSI\_H

#include "agregar.h"

#include "ui\_agregar.h"

#include "ManejoArchivos.h"

#include <QRegExpValidator>

Agregar**::**Agregar**(**QWidget **\***parent**)** **:**

QDialog**(**parent**),**

ui**(new** Ui**::**Agregar**)**

**{**

ui**->**setupUi**(this);**

QRegExp rx**(**"^[a-zA-Záéíóúñ]{3,16}$"**);**

QRegExpValidator **\***validar **=** **new** QRegExpValidator**(**rx**,this);**

ui**->**lineEdit**->**setValidator**(**validar**);**

ui**->**lineEdit\_2**->**setValidator**(**validar**);**

**}**

Agregar**::~**Agregar**()**

**{**

**delete** ui**;**

**}**

void Agregar**::**on\_pushButton\_clicked**()**

**{**

ManejoArchivo controlador**;**

controlador**.**agregar\_Diccionario**(**ui**->**lineEdit**->**text**().**toStdString**(),**ui**->**lineEdit\_2**->**text**().**toStdString**());**

Correcto **\***ventana **=** **new** Correcto**;**

ventana**->**show**();**

**}**

void Agregar**::**on\_pushButton\_2\_clicked**()**

**{**

MainWindow **\***ventana **=** **new** MainWindow**;**

**this->**hide**();**

ventana**->**show**();**

**}**

#pragma once

#include "Arbol.h"

Palabra Arbol**::**getPalabra**(**void**)**

**{**

**return** palabra**;**

**}**

void Arbol**::**setPalabra**(**Palabra newPalabra**)**

**{**

palabra **=** newPalabra**;**

**}**

Arbol**\*** Arbol**::**getDer**(**void**)**

**{**

**return** der**;**

**}**

void Arbol**::**setDer**(**Arbol**\*** newDer**)**

**{**

der **=** newDer**;**

**}**

Arbol**\*** Arbol**::**getIzq**(**void**)**

**{**

**return** izq**;**

**}**

void Arbol**::**setIzq**(**Arbol**\*** newIzq**)**

**{**

izq **=** newIzq**;**

**}**

Arbol**::**Arbol**()**

**{**

**}**

Arbol**::~**Arbol**()**

**{**

// TODO : implement

**}**

#pragma once

#if !defined(\_\_Arbol\_Arbol\_h)

#define \_\_Arbol\_Arbol\_h

#include "Palabra.h"

class Arbol

**{**

public**:**

Palabra getPalabra**(**void**);**

void setPalabra**(**Palabra newPalabra**);**

Arbol**\*** getDer**(**void**);**

void setDer**(**Arbol**\*** newDer**);**

Arbol**\*** getIzq**(**void**);**

void setIzq**(**Arbol**\*** newIzq**);**

Arbol**();**

**~**Arbol**();**

Arbol**\*** der**;**

Arbol**\*** izq**;**

protected**:**

private**:**

Palabra palabra**;**

**};**

#endif

#include "tie.h"

#include "ui\_tie.h"

#include <string>

#include<iostream>

#include<ManejoArchivo.h>

**using** **namespace** std**;**

TIE**::**TIE**(**QWidget **\***parent**)** **:**

QDialog**(**parent**),**

ui**(new** Ui**::**TIE**)**

**{**

ui**->**setupUi**(this);**

**}**

TIE**::~**TIE**()**

**{**

**delete** ui**;**

**}**

void TIE**::**on\_pushButton\_clicked**()**

**{**

Arbol **\***arbol**=NULL;**

OperacionArbol opA**;**

Palabra palabraT**;**

string result**;**

QString palabra **=** ui**->**txting**->**text**();**

open\_file**(**arbol**);**

string var**=**palabra**.**toStdString**();**

convertirMayusculas**(**var**);**

palabraT**.**setClave**(**clave**(**var**));**

opA**.**busqueda**(**arbol**,** palabraT**.**getClave**(),**var**,**result**);**

ui**->**txtespa**->**setText**(**QString**::**fromStdString**(**result**));**

**}**

#include "tsi.h"

#include "ui\_tsi.h"

#include <iostream>

#include <string>

#include "ManejoArchivos.h"

#include "OperacionArbol.h"

**using** **namespace** std**;**

Tsi**::**Tsi**(**QWidget **\***parent**)** **:**

QDialog**(**parent**),**

ui**(new** Ui**::**Tsi**)**

**{**

ui**->**setupUi**(this);**

**}**

Tsi**::~**Tsi**()**

**{**

**delete** ui**;**

**}**

void Tsi**::**on\_pushButton\_clicked**()**

**{**

ManejoArchivo controlador**;**

Arbol **\***arbolE **=** **NULL;**

Palabra palabraT**;**

controlador**.**open\_file\_espanol**(**arbolE**);**

OperacionArbol opAr**;**

string result**;**

QString palabra**=**ui**->**txtes**->**text**();**

string vari **=** palabra**.**toStdString**();**

controlador**.**convertirMayusculas**(**vari**);**

palabraT**.**setClaveE**(**controlador**.**clave**(**vari**));**

opAr**.**busquedaEspanol**(**arbolE**,** palabraT**.**getClaveE**(),**vari**,**result**);**

ui**->**txting**->**setText**(**QString**::**fromStdString**(**result**));**

**}**

#pragma once

#include "OperacionArbol.h"

#include <iostream>

#include "Palabra.h"

**using** **namespace** std**;**

Arbol **\***crearNodo**(**Palabra newpalabra**){**

Arbol **\***nuevo\_nodo**=new** Arbol**();**

nuevo\_nodo**->**setPalabra**(**newpalabra**);**

nuevo\_nodo**->**setDer**(NULL);**

nuevo\_nodo**->**setIzq**(NULL);**

**return** nuevo\_nodo**;**

**}**

void OperacionArbol**::**insertar**(**Arbol **\*&**arbol**,** Palabra palabra**)**

**{**

**if(**arbol**==NULL){**

Arbol **\***nuevo\_nodo**=**crearNodo**(**palabra**);**

arbol**=**nuevo\_nodo**;**

**}**

**else{**

int valorR **=** arbol**->**getPalabra**().**getClave**();**

**if(**palabra**.**getClave**()<**valorR**){**

insertar**(**arbol**->**izq**,**palabra**);**

**}**

**else{**

insertar**(**arbol**->**der**,**palabra**);**

**}**

**}**

**}**

void OperacionArbol**::**insertarE**(**Arbol **\*&**arbol**,** Palabra palabra**)**

**{**

**if(**arbol**==NULL){**

Arbol **\***nuevo\_nodo**=**crearNodo**(**palabra**);**

arbol **=** nuevo\_nodo**;**

**}**

**else{**

int valorR **=** arbol**->**getPalabra**().**getClaveE**();**

**if(**palabra**.**getClaveE**()** **<** valorR**){**

insertarE**(**arbol**->**izq**,**palabra**);**

**}**

**else{**

insertarE**(**arbol**->**der**,**palabra**);**

**}**

**}**

**}**

void space**(**int tam**)**

**{**

**for(**int i**=**0**;**i**<**tam**;**i**++)**

**{**

cout**<<**" "**;**

**}**

**}**

void OperacionArbol**::**mostrar**(**Arbol **\***arbol**,**int cont**)**

**{**

**if(**arbol**==NULL)**

**{**

**return;**

**}**

**else**

**{**

mostrar**(**arbol**->**getDer**(),**cont**+**1**);**

space**(**cont**);**

// cout<<arbol->getPalabra().getEspanol()<<" "<<arbol->getPalabra().getIngles()<<endl;

cout**<<**arbol**->**getPalabra**().**getClaveE**()<<**endl**;**

mostrar**(**arbol**->**getIzq**(),**cont**+**1**);**

**}**

**}**

void OperacionArbol**::**eliminar**(**Arbol **\*&**arbol**,**Palabra palabra**)**

**{**

// TODO : implement

**}**

void OperacionArbol**::**busqueda**(**Arbol **\***arbol**,** int n**,** string plbra**,**string **&**result**)**

**{**

**if(**arbol **==** **NULL)**

**{**

cout**<<** " No registrado en el diccionario "**<<**endl**;**

**}**

**else**

**{**

**if(**arbol**->**getPalabra**().**getClave**()** **==** n**)**

**{**

cout**<<**plbra**<<**" - "**<<**arbol**->**getPalabra**().**getEspanol**()<<**endl**;**

result**=**arbol**->**getPalabra**().**getEspanol**();**

**return;**

**}**

**else**

**{**

**if(** n **<** arbol**->**getPalabra**().**getClave**())**

**{**

busqueda**(**arbol**->**getIzq**(),**n**,**plbra**,**result**);**

**}**

**else**

**{**

busqueda**(**arbol**->**getDer**(),**n**,**plbra**,**result**);**

**}**

**}**

**}**

**}**

void OperacionArbol**::**busquedaEspanol**(**Arbol **\***arbol**,** int n**,** string plbra**,**string **&**result**)**

**{**

**if(**arbol **==** **NULL)**

**{**

cout**<<** " No registrado en el diccionario "**<<**endl**;**

**}**

**else**

**{**

**if(**arbol**->**getPalabra**().**getClaveE**()** **==** n**)**

**{**

cout**<<**plbra**<<**" - "**<<**arbol**->**getPalabra**().**getIngles**()<<**endl**;**

result**=**arbol**->**getPalabra**().**getIngles**();**

**return;**

**}**

**else**

**{**

**if(** n **<** arbol**->**getPalabra**().**getClaveE**())**

**{**

busquedaEspanol**(**arbol**->**izq**,**n**,**plbra**,**result**);**

**}**

**else**

**{**

busquedaEspanol**(**arbol**->**der**,**n**,**plbra**,**result**);**

**}**

**}**

**}**

**}**

void OperacionArbol**::**inorden**(**Arbol **\***arbol**)** **{**

**if** **(**arbol **==** **NULL)** **{**

**return;**

**}**

**else** **{**

inorden**(**arbol**->**getIzq**());**

cout **<<** arbol**->**getPalabra**().**getClave**()** **<<** " "**;**

inorden**(**arbol**->**getDer**());**

**}**

**}**

#pragma once

#if !defined(\_\_Arbol\_OperacionArbol\_h)

#define \_\_Arbol\_OperacionArbol\_h

#include "Arbol.h"

class OperacionArbol

**{**

public**:**

//Arbol \*crearNodo(Palabra palabra)

void insertar**(**Arbol **\*&**arbol**,** Palabra palabra**);**

void insertarE**(**Arbol **\*&**arbol**,** Palabra palabra**);**

void mostrar**(**Arbol **\***arbol**,**int cont**);**

void eliminar**(**Arbol **\*&**arbol**,**Palabra palabra**);**

void busqueda**(**Arbol **\***arbol**,** int n**,**string plbra**,**string **&**result**);**

void busquedaEspanol**(**Arbol **\***arbol**,** int n**,**string plbra**,**string **&**result**);**

void inorden**(**Arbol **\***arbol**);**

//void space(int tam);

protected**:**

private**:**

**};**

#endif

#pragma once

#ifndef MANEJOARCHIVOS\_H

#define MANEJOARCHIVOS\_H

#include<iostream>

#include<stdlib.h>

#include<string>

#include "Palabra.h"

#include "Arbol.h"

**using** **namespace** std**;**

class ManejoArchivo**{**

public**:**

void write\_file**(**Palabra p**);**

void open\_file**(**Arbol **\*&**arbol**);**

void open\_file\_espanol**(**Arbol **\*&**arbol**);**

int clave**(**string palabra**);**

void convertirMayusculas**(**string **&**palabra**);**

void agregar\_Diccionario**(**string**,**string**);**

**};**

#endif // MANEJOARCHIVOS\_H

#ifndef MANEJOARCHIVO\_H\_INCLUDED

#define MANEJOARCHIVO\_H\_INCLUDED

#include<iostream>

#include<stdlib.h>

#include<string>

#include "Palabra.h"

#include "OperacionArbol.h"

#include "IngresoBloqueTeclas.h"

#include "TransformacionString.h"

#include "Arbol.h"

#include<fstream>

**using** **namespace** std**;**

void write\_file**(**Palabra p**)**

**{**

ofstream archivo**;**

archivo**.**open**(**"Dic.txt"**,**ios**::**out**|**ios**::**app**);**

**if(**archivo**.**fail**())**

**{**

cout**<<**"No se pudo Abrir"**<<**endl**;**

exit**(**1**);**

**}**

**else{**

archivo**<<**p**.**getEspanol**()<<**";"**<<**p**.**getIngles**()<<**";"**<<**p**.**getClave**()<<**";"**<<**p**.**getClaveE**()<<**endl**;**

**}**

archivo**.**close**();**

**}**

void open\_file**(**Arbol **\*&**arbol**)**

**{**

ifstream archivo**;**

Transformacion transformacion**;**

OperacionArbol opA**;**

string espanol**,**ingles**,**clave**,**claveE**;**

archivo**.**open**(**"Dic.txt"**,**ios**::**in**);**

**if** **(**archivo**.**fail**())**

**{**

cout**<<**"No se puedo acceder al archivo"**<<**endl**;**

**}**

**else**

**{**

**while(!**archivo**.**eof**()){**

Palabra p**;**

getline**(**archivo**,**espanol**,**';'**);**

getline**(**archivo**,**ingles**,**';'**);**

getline**(**archivo**,**clave**,**';'**);**

getline**(**archivo**,**claveE**,**'\n'**);**

p**.**setEspanol**(**espanol**);**

p**.**setIngles**(**ingles**);**

p**.**setClave**(**transformacion**.**stringAint**(**clave**));**

p**.**setClaveE**(**transformacion**.**stringAint**(**claveE**));**

opA**.**insertar**(**arbol**,**p**);**

**}**

**}**

archivo**.**close**();**

**}**

void open\_file\_espanol**(**Arbol **\*&**arbol**)**

**{**

Transformacion transformacion**;**

ifstream archivo**;**

OperacionArbol opA**;**

string espanol**,**ingles**,**clave**,**claveE**;**

archivo**.**open**(**"Dic.txt"**,**ios**::**in**);**

**if** **(**archivo**.**fail**())**

**{**

cout**<<**"No se puedo acceder al archivo"**<<**endl**;**

**}**

**else**

**{**

**while(!**archivo**.**eof**()){**

Palabra p**;**

getline**(**archivo**,**espanol**,**';'**);**

getline**(**archivo**,**ingles**,**';'**);**

getline**(**archivo**,**clave**,**';'**);**

getline**(**archivo**,**claveE**,**'\n'**);**

p**.**setEspanol**(**espanol**);**

p**.**setIngles**(**ingles**);**

p**.**setClave**(**transformacion**.**stringAint**(**clave**));**

p**.**setClaveE**(**transformacion**.**stringAint**(**claveE**));**

opA**.**insertarE**(**arbol**,**p**);**

**}**

**}**

archivo**.**close**();**

**}**

int clave**(**string palabra**)**

**{**

int j**=**0**;**

**for(**int i**=**0**;**i**<**palabra**.**length**();**i**++)**

**{**

j**=**palabra**.**at**(**i**)+**j**;**

**}**

**if(**j**>=**100 **||** j**<**300**)**

**{**

j**=**j**-**100**;**

**}**

**else** **if(**j**>=**300**||** j**<**650**)**

**{**

j**=**j**-**200**;**

**}**

**else{**

j**=**j**+**500**;**

**}**

**return** j**;**

**}**

void convertirMayusculas**(**string **&**palabra**)**

**{**

**for(**int i**=**0**;**i**<**palabra**.**length**();**i**++)**

**{**

palabra**.**at**(**i**)=**toupper**(**palabra**.**at**(**i**));**

**}**

**}**

void agregar\_Diccionario**()**

**{**

char c**;**

string a**,**b**;**

Ingreso ig**;**

cout**<<**"Espanol"**<<**endl**;**

a**=**ig**.**ingresarLetras**(&**c**);**

cout**<<**"Ingles"**<<**endl**;**

b**=**ig**.**ingresarLetras**(&**c**);**

Palabra p**;**

convertirMayusculas**(**a**);**

convertirMayusculas**(**b**);**

p**.**setEspanol**(**a**);**

p**.**setIngles**(**b**);**

p**.**setClave**(**clave**(**b**));**

p**.**setClaveE**(**clave**(**a**));**

write\_file**(**p**);**

**}**

#endif // MANEJOARCHIVO\_H\_INCLUDED

#ifndef AGREGAR\_H

#define AGREGAR\_H

#include <QDialog>

#include <correcto.h>

#include <mainwindow.h>

**namespace** Ui **{**

class Agregar**;**

**}**

class Agregar **:** public QDialog

**{**

Q\_OBJECT

public**:**

explicit Agregar**(**QWidget **\***parent **=** **nullptr);**

**~**Agregar**();**

private slots**:**

void on\_pushButton\_clicked**();**

void on\_pushButton\_2\_clicked**();**

private**:**

Ui**::**Agregar **\***ui**;**

**};**

#endif // AGREGAR\_H

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow**::**MainWindow**(**QWidget **\***parent**)**

**:** QMainWindow**(**parent**)**

**,** ui**(new** Ui**::**MainWindow**)**

**{**

ui**->**setupUi**(this);**

**}**

MainWindow**::~**MainWindow**()**

**{**

**delete** ui**;**

**}**

void MainWindow**::**on\_pushButton\_clicked**()**

**{**

Tsi **\***ventana **=** **new** Tsi**;**

ventana**->**show**();**

**}**

void MainWindow**::**on\_pushButton\_2\_clicked**()**

**{**

TIE **\***ventana **=** **new** TIE**;**

ventana**->**show**();**

**}**

void MainWindow**::**on\_pushButton\_3\_clicked**()**

**{**

Agregar **\***ventana **=** **new** Agregar**;**

**this->**hide**();**

ventana**->**show**();**

**}**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include<tie.h>

#include<tsi.h>

#include<agregar.h>

QT\_BEGIN\_NAMESPACE

**namespace** Ui **{** class MainWindow**;** **}**

QT\_END\_NAMESPACE

class MainWindow **:** public QMainWindow

**{**

Q\_OBJECT

public**:**

MainWindow**(**QWidget **\***parent **=** **nullptr);**

**~**MainWindow**();**

private slots**:**

void on\_pushButton\_clicked**();**

void on\_pushButton\_2\_clicked**();**

void on\_pushButton\_3\_clicked**();**

private**:**

Ui**::**MainWindow **\***ui**;**

**};**

#endif // MAINWINDOW\_H

#pragma once

#include <iostream>

#include<stdlib.h>

#include<conio.h>

#include<stdio.h>

#include<string>

**using** **namespace** std**;**

class Ingreso **{**

public**:**

double ingresarDouble**(**char**\*,**char**\*);**

int ingresarInt**(**char**\*);**

char ingresCaracter**(**char**\*);**

string ingresarLetras**(**char**\*);**

string ingresarNumeros**(**char**\*);**

**};**

#pragma once

#include "IngresoBloqueTeclas.h"

#include <iostream>

#include<stdlib.h>

#include<conio.h>

#include<stdio.h>

#include<string>

double Ingreso**::**ingresarDouble**(**char **\***msg**,**char**\***p**)** **{**

char tecla**;**

int i**=**0**,**cont**=**0**;**

p**[**0**]=**'\0'**;**

cout**<<**msg**<<**endl**;**

**do**

**{**

tecla**=**getch**();**

**if(**i**>** 0 **&&** tecla **==** 8**)**

**{**

cout**<<**"\b \b"**;**

cont**--;**

p**[--**i**]=**'\0'**;**

**}**

**else{**

**if((**tecla**>=**48 **&&** tecla**<=**57**)||(**tecla**==**46 **&&** cont**<=**0**)** **)**

**{**

cout**<<**tecla**;**

p**[**i**++]=**tecla**;**

**if(**tecla**==**46**)**

**{**

cont**++;**

**}**

**}**

**}**

**}while(**tecla**!=**13 **||** p**[**0**]==**'\0'**);**

p**[**i**]=**'\0'**;**

cout**<<**endl**;**

system**(**"cls"**);**

**return** atof**(**p**)** **;**

**}**

int Ingreso**::**ingresarInt**(**char **\***p**)**

**{**

char tecla**;**

int i**=**0**;**

p**[**0**]=**'\0'**;**

**do**

**{**

tecla**=**getch**();**

**if(**i**>** 0 **&&** tecla **==** 8**)**

**{**

cout**<<**"\b \b"**;**

p**[--**i**]=**'\0'**;**

**}**

**else{**

**if((**tecla**>=**48 **&&** tecla**<=**57**))**

**{**

cout**<<**tecla**;**

p**[**i**++]=**tecla**;**

**}**

**}**

**}while(**tecla**!=**13 **||** p**[**0**]==**'\0'**);**

p**[**i**]=**'\0'**;**

cout**<<**endl**;**

system**(**"cls"**);**

**return** atoi**(**p**);**

**}**

string Ingreso**::**ingresarLetras**(**char **\***p**)**

**{**

string s**;**

char tecla**;**

int i**=**0**;**

p**[**0**]=**'\0'**;**

**do**

**{**

tecla**=**getch**();**

**if(**i**>** 0 **&&** tecla **==** 8**)**

**{**

cout**<<**"\b \b"**;**

p**[--**i**]=**'\0'**;**

**}**

**else{**

**if((**tecla**>=**97 **&&** tecla**<=**122**)||** **(**tecla**>=**65 **&&** tecla**<=**90**))**

**{**

cout**<<**tecla**;**

p**[**i**++]=**tecla**;**

**}**

**}**

**}while(**tecla**!=**13 **||** p**[**0**]==**'\0'**);**

p**[**i**]=**'\0'**;**

cout**<<**endl**;**

s**=**string**(**p**);**

system**(**"cls"**);**

**return** s**.**c\_str**();**

**}**

string Ingreso**::**ingresarNumeros**(**char **\***p**)**

**{**

string s**;**

char tecla**;**

int i**=**0**;**

p**[**0**]=**'\0'**;**

**do**

**{**

tecla**=**getch**();**

**if(**i**>** 0 **&&** tecla **==** 8**)**

**{**

cout**<<**"\b \b"**;**

p**[--**i**]=**'\0'**;**

**}**

**else{**

**if((**tecla**>=**48 **&&** tecla**<=**57**))**

**{**

cout**<<**tecla**;**

p**[**i**++]=**tecla**;**

**}**

**}**

**}while(**tecla**!=**13 **||** p**[**0**]==**'\0'**);**

p**[**i**]=**'\0'**;**

cout**<<**endl**;**

s**=**string**(**p**);**

system**(**"cls"**);**

**return** s**.**c\_str**();**

**}**

char Ingreso**::**ingresCaracter**(**char **\***p**)**

**{**

char tecla**;**

char c**;**

int i**=**0**;**

p**[**0**]=**'\0'**;**

**do**

**{**

tecla**=**getch**();**

**if(**i**>** 0 **&&** tecla **==** 8**)**

**{**

cout**<<**"\b \b"**;**

p**[--**i**]=**'\0'**;**

**}**

**else{**

**if((**tecla**==**104 **||** tecla**==**109 **||** tecla**==**72 **||** tecla**==**77 **||** tecla**==**83 **||** tecla**==**67 **||** tecla**==**68**))**

**{**

cout**<<**tecla**;**

p**[**i**++]=**tecla**;**

**}**

**}**

**}while(**tecla**!=**13 **||** p**[**0**]==**'\0'**);**

p**[**i**]=**'\0'**;**

cout**<<**endl**;**

system**(**"cls"**);**

c**=\*(**p**+**0**);**

**return** c**;**

**}**

#ifndef CORRECTO\_H

#define CORRECTO\_H

#include <QDialog>

**namespace** Ui **{**

class Correcto**;**

**}**

class Correcto **:** public QDialog

**{**

Q\_OBJECT

public**:**

explicit Correcto**(**QWidget **\***parent **=** **nullptr);**

**~**Correcto**();**

private**:**

Ui**::**Correcto **\***ui**;**

**};**

#endif // CORRECTO\_H

#include "correcto.h"

#include "ui\_correcto.h"

Correcto**::**Correcto**(**QWidget **\***parent**)** **:**

QDialog**(**parent**),**

ui**(new** Ui**::**Correcto**)**

**{**

ui**->**setupUi**(this);**

**}**

Correcto**::~**Correcto**()**

**{**

**delete** ui**;**

**}**

#include "mainwindow.h"

#include <QApplication>

int main**(**int argc**,** char **\***argv**[])**

**{**

QApplication a**(**argc**,** argv**);**

MainWindow w**;**

w**.**show**();**

**return** a**.**exec**();**

**}**

#include <iostream>

#include "PersonalLibrary.h"

#include "ingreso.h"

#include <pthread.h>

**using** **namespace** std**;**

void**\*** moverMarquesina**(**void **\***marq**){**

int a**,** b**,** c**,** n**=**0**,**letra**=**150**,**pos**=**1**,**cont**=**39**,**aux**,**cont1**=**39**,**auxg**;**

char t**[**50**]** **=**"GRACIAS POR USAR NUESTRO PROGRAMA"**,**auxt**[**39**]=**" "**;**

**do{**

**for** **(**a**=**0**;**a**<**42**;**a**++){**

aux**=**pos**;**

**for(**b**=**39**;**b**>**cont**;**b**--){**

gotoxy**(**pos**,**1**);**

cout**<<**t**[**b**];**

pos**--;**

**}**

aux**++;**

pos**=**aux**;**

cont**--;**

Sleep **(**75**);**

**if(**a**==**40**){**

**break;**

**}**

**}**

**for(**a**=**3**;**a**<**70**;**a**++){**

gotoxy**(**a**-**1**,**1**);**

cout**<<**" "**;**

gotoxy**(**a**,**1**);**

cout**<<**t**;**

Sleep **(**75**);**

**}**

pos**=**70**;**

auxg**=**69**;**

**for** **(**a**=**0**;**a**<**41**;**a**++){**

gotoxy**(**auxg**,**1**);**

cout**<<**" "**;**

aux**=**pos**;**

**for(**b**=**0**;**b**<=**cont1**;**b**++){**

gotoxy**(**pos**,**1**);**

cout**<<**t**[**b**];**

pos**++;**

**}**

cont1**--;**

aux**++;**

pos**=**aux**;**

auxg**++;**

Sleep **(**75**);**

**}**

cont1**=**39**;**

letra**=**39**;**

pos**=**1**;**

cont**=**39**;**

c**=**0**;**

**}while** **(**c**=**1**);**

**}**

int main**()**

**{**

AltEnter**();**

const char **\***opciones**[]={**"1) Traductor Palabras"**,**"2) Sello ESPE"**,**"3) Mostrar Ayuda"**,**"4) Codificar"**,**"5) Decodificar"**,**"6) Generar PDF"**,**"7) Codigo de Barras"**,**"8) Salir"**};**

bool bandera **=** **true;**

string nombre**,**datos **=** "Naranjo Jhony 1750184010\nToapanta Antoni 1720273794"**;**

pthread\_t thread1**;**

Ingreso leer**;**

**do{**

int opcion **=** menu**(**"Bienvenido"**,**opciones**,**8**);**

**switch(**opcion**){**

**case** 1**:**

system**(**"cls"**);**

system**(**"Proyecto20.exe"**);**

system**(**"pause"**);**

**break;**

**case** 2**:**

system**(**"cls"**);**

system**(**"imagenExtras.exe"**);**

system**(**"pause"**);**

**break;**

**case** 3**:**

system**(**"Traductor.chm"**);**

**break;**

**case** 4**:**

system**(**"cls"**);**

codificar**(**datos**,**0**);**

archivoC**(**datos**);**

system**(**"codificar.txt"**);**

system**(**"pause"**);**

**break;**

**case** 5**:**

system**(**"cls"**);**

decodificar**(**datos**,**0**);**

archivoD**(**datos**);**

system**(**"decodificar.txt"**);**

system**(**"pause"**);**

**break;**

**case** 6**:**

system**(**"cls"**);**

system**(**"Autores.pdf"**);**

system**(**"pause"**);**

**break;**

**case** 7**:**

system**(**"cls"**);**

system**(**"barras.png"**);**

system**(**"pause"**);**

**break;**

**case** 8**:**

system**(**"cls"**);**

pthread\_create**(&**thread1**,NULL,**moverMarquesina**,NULL);**

system**(**"pause"**);**

bandera **=** **false;**

**break;**

**}**

**}while(**bandera**);**

**return** 0**;**

**}**

1. **Conclusiones**

Los árboles binarios de búsqueda han sido de gran ayuda para el proyecto pues es más fácil comparar tanto para el hijo derecho como al hijo izquierdo. Así mismo la recursividad sigue siendo una forma de resolver este problema de una forma más sencilla.