

IE-0217 Estructuras abstractas de datos y algoritmos para ingeniería.

## Laboratorio 2: Compresión de Texto

Belinda Brown Ramírez - B61254

I-2019

---

### Tabla de contenidos

<b>1. Enunciado</b>	<b>2</b>
<b>2. Pruebas:</b>	<b>2</b>
2.1. C++: . . . . .	2
2.2. Python: . . . . .	4
<b>3. Conclusiones:</b>	<b>6</b>
<b>4. Diagramas de clase</b>	<b>6</b>
<b>5. Anexos</b>	<b>7</b>
5.1. Código C++: . . . . .	7
5.2. Código Python: . . . . .	13
<b>6. Consideraciones</b>	<b>14</b>

---

# 1. Enunciado

En este laboratorio su equipo de trabajo deberá implementar, en Python y C++, un algoritmo de compresión y descompresión de texto basado en reemplazos. Dicho algoritmo funciona de la siguiente manera:

- Tomando como entrada un archivo con texto, se establece un umbral de reemplazo.
- Para cada palabra cuya frecuencia sea igual o mayor a la del umbral de reemplazo, cambiar su hilera de caracteres por otra. Agregue en un archivo aparte una tabla con cada uno de los reemplazos, indicando la palabra original y su nueva hilera.

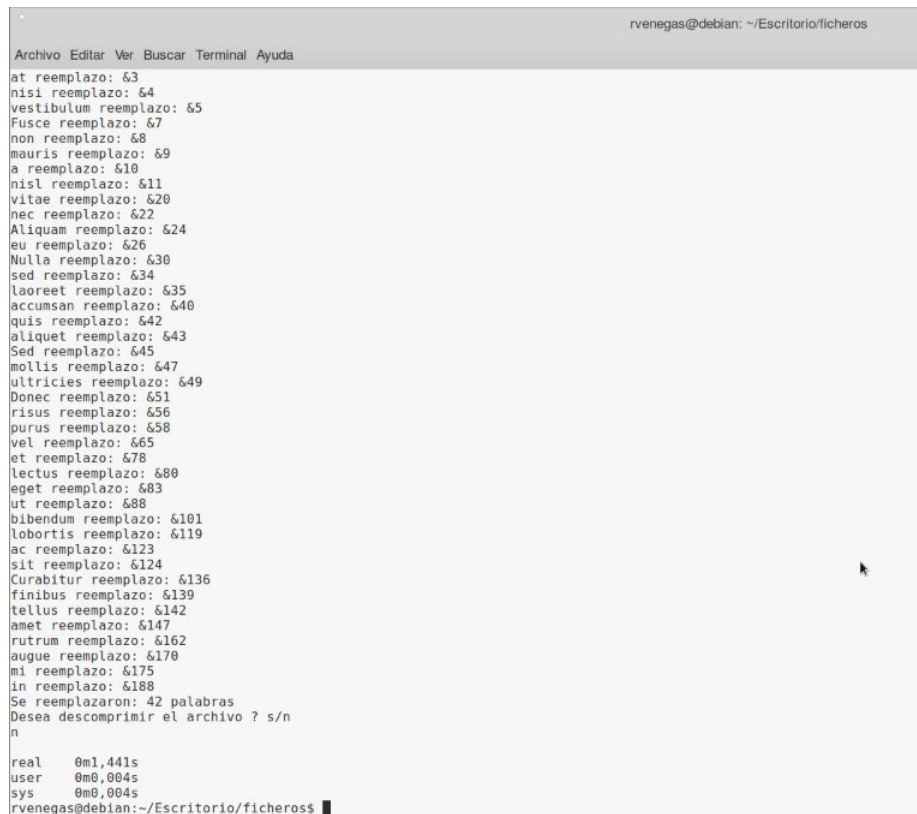
No modifique el archivo de entrada, cree uno nuevo con el mismo nombre que el original pero de extensión .rep y el archivo de la tabla, de la misma manera pero con extensión .tab.

Haga un programa de prueba para los ítemes anteriores que utilice tamaños de entrada diferentes, con archivos de 1MB, 10MB, 100MB y 1000MB (este de ser posible). Tome los tiempos de ejecución de ambas implementaciones con todos los tamaños de archivos y grafíquelos. Hipotetice sobre el por qué de la tendencia de los tiempos.

En la medida de lo posible utilice programación orientada a objetos (ej. una clase para reemplazar, otra para manejar la entrada/salida, etc.).

## 2. Pruebas:

### 2.1. C++:



```
rvenegas@debian: ~/Escritorio/ficheros
Archivo Editar Ver Buscar Terminal Ayuda
at reemplazo: 63
nisi reemplazo: 64
vestibulum reemplazo: 65
Fusce reemplazo: 67
non reemplazo: 68
mauris reemplazo: 69
a reemplazo: 610
nisl reemplazo: 611
vitae reemplazo: 620
nec reemplazo: 622
Aliquam reemplazo: 624
eu reemplazo: 626
Nulla reemplazo: 630
sed reemplazo: 634
laoreet reemplazo: 635
accumsan reemplazo: 640
quis reemplazo: 642
aliquet reemplazo: 643
Sed reemplazo: 645
mollis reemplazo: 647
ultrices reemplazo: 649
Donec reemplazo: 651
risus reemplazo: 656
purus reemplazo: 658
vel reemplazo: 665
et reemplazo: 678
lectus reemplazo: 680
eget reemplazo: 683
ut reemplazo: 688
bibendum reemplazo: 6101
lobortis reemplazo: 6119
ac reemplazo: 6123
sit reemplazo: 6124
Curabitur reemplazo: 6136
finibus reemplazo: 6139
tellus reemplazo: 6142
amet reemplazo: 6147
rutrum reemplazo: 6162
augue reemplazo: 6170
mi reemplazo: 6175
in reemplazo: 6188
Se reemplazaron: 42 palabras
Desear descomprimir el archivo ? s/n
n
real    0m1.441s
user    0m0.004s
sys     0m0.004s
rvenegas@debian:~/Escritorio/ficheros$
```

```

rvenegas@debian: ~/Escritorio/ficheros

Archivo Editar Ver Buscar Terminal Ayuda

gravida, reemplazo: 6421
eismod, reemplazo: 6422
nisl, reemplazo: 6423
pharetra, reemplazo: 6424
commodo, reemplazo: 6425
ante, reemplazo: 6426
semper, reemplazo: 6427
facillisis, reemplazo: 6428
dolor, reemplazo: 6429
laoreet, reemplazo: 6430
mattis, reemplazo: 6431
dolor, reemplazo: 6432
interdum, reemplazo: 6433
pellentesque, reemplazo: 6434
mollis, reemplazo: 6435
molestie, reemplazo: 6436
venenatis, reemplazo: 6437
ut, reemplazo: 6438
ullamcorper, reemplazo: 6439
eget, reemplazo: 6440
turpis, reemplazo: 6441
iaculis, reemplazo: 6442
faucibus, reemplazo: 6443
elementum, reemplazo: 6444
dignissim, reemplazo: 6445
mollis, reemplazo: 6446
vulputate, reemplazo: 6447
mattis, reemplazo: 6448
rutrum, reemplazo: 6449
feugiat, reemplazo: 6450
maximus, reemplazo: 6451
tempus, reemplazo: 6452
accumsan, reemplazo: 6453
vehicula, reemplazo: 6454
purus, reemplazo: 6455
porta, reemplazo: 6456
sodales, reemplazo: 6457
imperdiet, reemplazo: 6458
hendrerit, reemplazo: 6459
bibendum, reemplazo: 6460
odio, reemplazo: 6461
Se reemplazaron: 462 palabras
Desea descomprimir el archivo ? s/n
N

real    0m5,223s
user    0m1,764s
sys     0m0,016s
rvenegas@debian:~/Escritorio/ficheros$
```

```

rvenegas@debian: ~/Escritorio/ficheros

Archivo Editar Ver Buscar Terminal Ayuda

Esmirna reemplazo: 6165345
Silvano, reemplazo: 6165401
[Steiger]. reemplazo: 6165418
(eis reemplazo: 6165789
fingimientos reemplazo: 6165825
naos, reemplazo: 6165969
Cirilo reemplazo: 6167057
Primera, reemplazo: 6167068
—Griego, reemplazo: 6167507
[Agustín.] reemplazo: 6167803
luz." reemplazo: 6168153
Domiciano, reemplazo: 6169214
Sardis reemplazo: 6169830
Andreas reemplazo: 6169867
Revelación reemplazo: 6169922
Filadelfia reemplazo: 6170007
eclesio reemplazo: 6170105
Espíritus reemplazo: 6170175
Coptica, reemplazo: 6170204
jamás-Griego, reemplazo: 6170236
Coptica, reemplazo: 6170262
A, reemplazo: 6170334
Tiatira reemplazo: 6170360
B reemplazo: 6171109
Aleph, reemplazo: 6171110
B., reemplazo: 6171255
transfigurada reemplazo: 6171389
"santo," reemplazo: 6171519
Aleph reemplazo: 6171678
B, reemplazo: 6171691
C reemplazo: 6171814
Y-Así reemplazo: 6172228
Coptica; reemplazo: 6172397
Andreas, reemplazo: 6172589
Andreas; reemplazo: 6172598
bestia." reemplazo: 6172643
Andreas, reemplazo: 6172691
bestia; reemplazo: 6172986
ramera; reemplazo: 6173081
Ángel-Así reemplazo: 6173256
Novia, reemplazo: 6174108
Se reemplazaron: 25983 palabras
Desea descomprimir el archivo ? s/n
n

real    69m9,307s
user    47m27,968s
sys     0m2,132s
rvenegas@debian:~/Escritorio/ficheros$
```

## 2.2. Python:

```
caro@deathstar: ~/Music/Lab02/Lab02/Manejo_texto_python
File Edit View Search Terminal Help
caro@deathstar:~/Music/Lab02/Lab02/Manejo_texto_python$ make
python escribe_archivo.py
python Leer_archivo.py

Antoine de Saint Exupéry falleció en 1944 -un año
después de la publicación de El principito-,
cuando en una misión de reconocimiento se determinó
a su avión desaparecido, siendo encontrados restos del avión
e identificaciones del autor y su
esposa entre los años 1998 y 2000.

python reemplazo.py
Digite nombre del archivo (sin el .txt): 10MB
Digite nombre del archivo (con su .txt): 10MB.txt
python tabulacion_cambios.py
python leer_tabulacion_cambios.py

urna          ~,~^~
dolor         ~^~^~
quis          ~^~:~

cat *.rep
1
COMENTARIO
EXEGETICO Y EXPLICATIVO
DE LA BIBLIA
TOMO I: EL ANTIGUO TESTAMENTO
POR
Roberto Jamieson
A. R. Fausset
David Brown
2
TRADUCTORES:
Jaime C. Quarles
Lemuel C. Quarles
José M. Rodríguez
```

```
caro@deathstar: ~/Music/Lab02/Lab02/Manejo_texto_python
File Edit View Search Terminal Help
caro@deathstar:~/Music/Lab02/Lab02/Manejo_texto_python$ make
python escribe_archivo.py
python Leer_archivo.py

Antoine de Saint Exupéry falleció en 1944 -un año
después de la publicación de El principito-,
cuando en una misión de reconocimiento se determinó
a su avión desaparecido, siendo encontrados restos del avión
e identificaciones del autor y su
esposa entre los años 1998 y 2000.

python reemplazo.py
Digite nombre del archivo (sin el .txt): 1MB
Digite nombre del archivo (con su .txt): 1MB.txt
python tabulacion_cambios.py
python leer_tabulacion_cambios.py

urna          ~,~^~
dolor         ~^~^~
quis          ~^~:~

cat *.rep
Lorem ipsum ~^~,~ sit amet, consectetur adipiscing elit. Vivamus condimentum s
agittis lacus, laoreet luctus ligula laoreet ut. Vestibulum ullamcorper accumsan
Velit vel vehicula. Proin tempor lacus arcu. Nunc at elit condimentum, semper n
isi et, condimentum mi. In venenatis blandit nibh at sollicitudin. Vestibulum da
pibus mauris at orci maximus pellentesque. Nullam id elementum ipsum. Suspendiss
e cursus lobortis viverra. Proin et erat at mauris tincidunt porttitor vitae ac
dui.

Donec vulputate lorem tortor, nec fermentum nibh bibendum vel. Lorem ipsum ~^~,~
~,~ sit amet, consectetur adipiscing elit. Praesent dictum luctus massa, non euismod lacus. Pellentesque condimentum ~^~,~ est, ut dapibus lectus luctus ac. Ut
sagittis commodo arcu. Integer nisi nulla, facilisis sit amet nulla ~^~:~, ele
ifend suscipit purus. Class aptent taciti sociosqu ad litora torquent per conubi
a nostra, per inceptos himenaeos. Aliquam euismod ultrices lorem, sit amet imper
diet est tincidunt vel. Phasellus dictum justo sit amet ligula varius aliquet au
ctor et metus. Fusce vitae tortor et nisi pulvinar vestibulum eget in risus. Don
ec ante ex, placerat a lorem eget, ultricies bibendum purus. Nam sit amet neque
```

```
caro@deathstar: ~/Music/Lab02/Lab02/Manejo_texto_python
File Edit View Search Terminal Help

caro@deathstar:~/Music/Lab02/Lab02/Manejo_texto_python$ make
python escribe_archivo.py
python Leer_archivo.py

Antoine de Saint Exupéry falleció en 1944 —un año
después de la publicación de El principito—,
cuando en una misión de reconocimiento se determinó
a su avión desaparecido, siendo encontrados restos del avión
e identificaciones del autor y su
esposa entre los años 1998 y 2000.

python reemplazo.py
Digite nombre del archivo (sin el .txt): 10MB1
Digite nombre del archivo (con su .txt): 10MB1.txt
python tabulacion_cambios.py
python leer_tabulacion_cambios.py

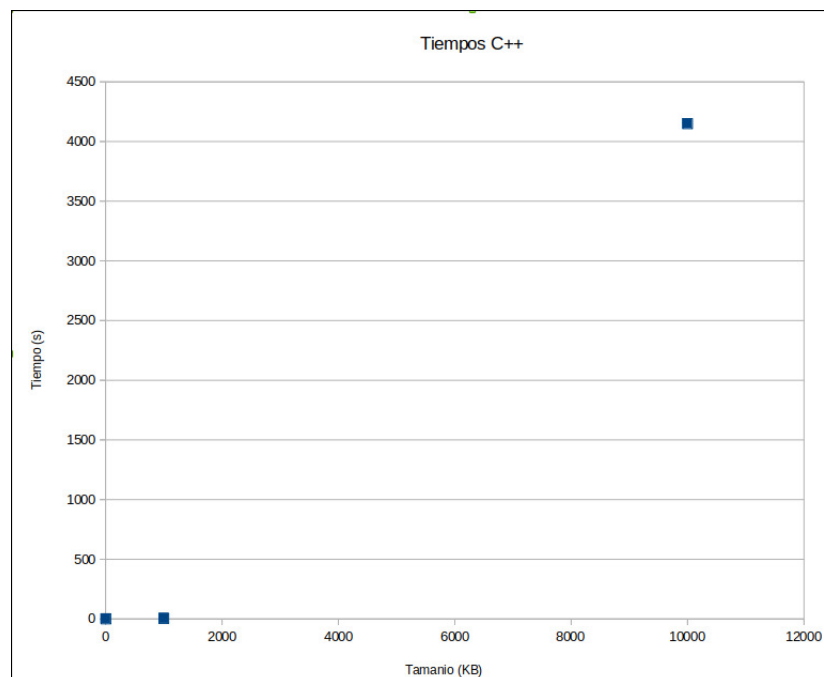
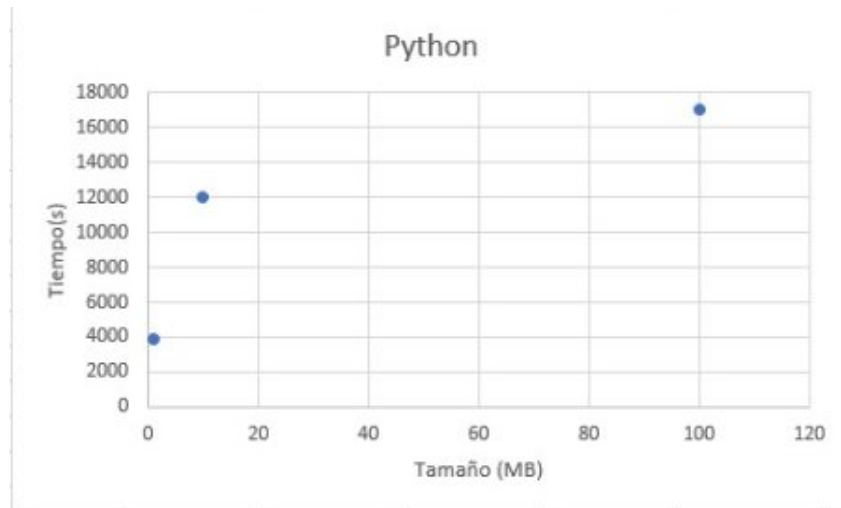
urna          ~,~^~
dolor         ~^~^,~
quis          ~^^;~~

cat *.rep
Lorem ipsum ~^^,~,~ sit amet, consectetur adipiscing elit. Vivamus condimentum s
agittis lacus, laoreet luctus ligula laoreet ut. Vestibulum ullamcorper accumsan
velit vel vehicula. Proin tempor lacus arcu. Nunc at elit condimentum, semper n
isi et, condimentum mi. In venenatis blandit nibh at sollicitudin. Vestibulum da
pibus mauris at orci maximus pellentesque. Nullam id elementum ipsum. Suspendiss
e cursus lobortis viverra. Proin et erat at mauris tincidunt porttitor vitae ac
dui.

Donec vulputate lorem tortor, nec fermentum nibh bibendum vel. Lorem ipsum ~^^,~,
~,~ sit amet, consectetur adipiscing elit. Praesent dictum luctus massa, non euismod
lacus. Pellentesque condimentum ~^^,~,~ est, ut dapibus lectus luctus ac. Ut
sagittis commodo arcu. Integer nisi nulla, facilisis sit amet nulla ~^^;~~, ele
ifend suscipit purus. Class aptent taciti sociosqu ad litora torquent per conubi
a nostra, per inceptos himenaeos. Aliquam euismod ultrices lorem, sit amet imper
diet est tincidunt vel. Phasellus dictum justo sit amet ligula varius aliquet au
ctor et metus. Fusce vitae tortor et nisi pulvinar vestibulum eget in risus. Don
ec ante ex, placerat a lorem eget, ultricies bibendum purus. Nam sit amet neque
```

### 3. Conclusiones:

Las instrucciones indican que se debe realizar una comparación entre ambos lenguajes, por tanto tenemos:



Con respecto a las gráficas podemos observar como al utilizar un lenguaje compilado como C++, el tiempo de espera es menor en comparación con el lenguaje interpretado Python.

### 4. Diagramas de clase

Se encontrarán adjuntos en refman.pdf. Creado por Doxygen, ubicado en la carpeta DOXY latex.

## 5. Anexos

### 5.1. Código C++:

main.cpp:

```
1  /*!  
2      *\author Ruben Venegas , Carolina Urrutia , Belinda Brown  
3      *\version 1.0  
4      *\date 11/05/2019  
5      *\mainpage Compresion de texto  
6      *\section intro_sec Introduccion  
7      * Este codigo reemplaza palabras segun la cantidad de recurrencias que presente  
8      * la palabra  
9      */  
10  
11 #include <iostream>  
12 #include "reemplazo.h"  
13 #include "creartab.h"  
14  
15 using namespace std;  
16  
17 int main(int argc , char* argv[]) {  
18  
19     CrearTab tab1(argv[2], atoi(argv[1]));  
20     int lineas = tab1.creacion_tab();  
21     Reemplazo rempl(argv[2], "in.tab", lineas);  
22     rempl.reemplazar();  
23     cout<<"Se reemplazaron: " <<lineas<<" palabras"<<endl;  
24  
25     cout<<"Desea descomprimir el archivo ? s/n" <<endl;  
26     string entrada;  
27     cin >> entrada;  
28     if (entrada == "s"){  
29         rempl.descomprimir();  
30         cout << "El archivo descomprimido se llama in-des.rep" <<endl;  
31     }  
32  
33  
34     return 0;  
35 }
```

```
1 #ifndef CREARTAB.H  
2 #define CREARTAB.H  
3  
4 #include <iostream>  
5 #include <fstream>  
6 #include <cstring>  
7 #include <sstream>  
8 #include <string>  
9  
10  
11 using namespace std;  
12  
13 class CrearTab {  
14     public:  
15         string filePath;  
16         int umbral;
```

```

17     CrearTab();
18     CrearTab(string filePath, int umbral);
19     int count_words(int longitud, string entrada);
20     int frecuencia_palabra(string* palabras, int* frecuencia, int num, string
21     word);
22     int creacion_tab();
23 };
24 #endif

1 #include "creartab.h"
2
3 using namespace std;
4
5
6
7 CrearTab::CrearTab() {};
8 CrearTab::CrearTab(string filePath, int umbral){
9
10     cout << "\t\t\t\t: " << filePath << " - " << umbral << endl;
11
12     this->filePath=filePath;
13     this->umbral=umbral;};
14 int CrearTab::count_words(int longitud, string entrada){ //cuenta las palabras en un
15     string
16     int i=0;
17     int cont=0;
18     while(i<longitud){
19         while (i<longitud && entrada[i]==' '){
20             i=i+1;
21         }
22         if(i<longitud){
23             cont=cont+1;
24         }
25         while(i<longitud && entrada[i]!=' '){
26             i=i+1;
27         }
28     }
29     return cont;
30 }
31 int CrearTab::frecuencia_palabra(string* palabras, int* frecuencia, int num, string
32 word){
33     bool condicion=false;
34     if (num==0){ //Llena el primer elemento del arreglo con la primer palabra
35         palabras[num]=word;
36         frecuencia[num]=frecuencia[num]+1;
37         num=num+1;
38     }
39     for (int i=0; i<=num; i++){
40         if (palabras[i]==word && num>0){
41             frecuencia[i]=frecuencia[i]+1; //Si la palabra ya estaba guardada,
42             aumenta su frecuencia
43             condicion=true;
44         }
45     }
46     if (condicion ==false && num>0) {
47         palabras[num]=word; //Si la palabra no es repetida la guarda
48         frecuencia[num]=frecuencia[num]+1;

```



```

47     num=num+1;
48
49 }
50 num=num;
51
52 return num;
53 }
54 int CrearTab::creacion_tab() {
55     int count =0;
56     ifstream archivo1;
57     archivo1.open(filePath , ios::in);
58     string texto1;
59     while(!archivo1.eof()){
60         getline(archivo1 , texto1); //Divide el texto en lineas
61         int k = count_words(texto1.length() , texto1); //por cada linea cuenta las
palabras
62         count=count+k; //contador de palabras totales del texto
63     }
64     cout<<"Palabras contadas: "<<count<<endl;
65     string palabras[count] {}; //crea un arreglo de strings con la cantidad de
palabras tot del texto
66     int frecuencias[count] {}; //crea un arreglo de ints con la cantidad de palabras
tot del texto que se
67                                     //asociara a cada string del arreglo anterior (
contara la frecuencia de cada palabra)
68     archivo1.close();
69     ifstream archivo2;
70     archivo2.open(filePath , ios::in);
71     string texto2;
72     int m =0;
73     int num=0;
74     int otra = 0;
75     while(!archivo2.eof()){
76         getline(archivo2 , texto2);
77         istream isstream(texto2);
78         bool condicion = false;
79         while(!isstream.eof()){ //divide cada linea en palabras
80             string tempStr;
81             isstream >> tempStr;
82             otra = otra +1;
83             if (otra >10000){
84                 cout<<"Creando in.tab, Voy revisando 10000 mas" <<endl;
85                 otra = 0;
86             }
87             if (num<count && tempStr != "." && tempStr != "," ) { //se guarda cada
palabra en el arreglo creado anteriormente
88                 int m = frecuencia_palabra(palabras , frecuencias , num, tempStr);
89                 num=m;
90             }
91         }
92     }
93     archivo2.close();
94     int cant_entradas = 0 ;
95     ofstream salida;
96     salida.open("in.tab" , ios::ate); //se usa ios::ate para escribir desde la ultima
parte del texto
97     for (int j=0; j<count; j++){
98         if(frecuencias[j]>=umbral && palabras[j] !="" ) { //solo escribe las palabras
que superen el rango de frecuencias

```

```

99         cout << palabras[j]<< " freq:" << frecuencias[j]<<" &"<<j<<endl;
100         salida<<palabras[j]<< " freq:" << frecuencias[j] <<" &"<<j<<endl;
101         cant_entradas = cant_entradas +1;
102     }
103 }
104
105 salida.close();
106 return cant_entradas; //cantidad de lineas escritas en in.tab
107 }

```

```

1 #ifndef REEMPLAZO.H
2 #define REEMPLAZO.H
3
4 #include <iostream>
5 #include <fstream>
6 #include <cstring>
7 #include <sstream>
8 #include <string>
9
10
11 using namespace std;
12
13 class Reemplazo {
14     public:
15         string filePath;
16         string tab_path;
17         int entradas;
18
19         Reemplazo(string filePath , string tab_path ,int entradas);
20         void reemplazar();
21         void descomprimir ();
22
23         Reemplazo();
24
25     };
26 #endif

```

```

1 #include "reemplazo.h"
2
3 Reemplazo::Reemplazo(string filePath , string tab_path ,int entradas){
4     this->filePath=filePath; //direccion de in.txt
5     this->entradas=entradas; //cantidad de lineas en in.tab
6     this->tab_path=tab_path; //direccion de in.tab
7 };
8
9 void Reemplazo::reemplazar(){
10     ifstream archivo1;
11     archivo1.open(tab_path , ios::in);
12     string texto1;
13     string palabras[entradas] {}; //arreglo donde se guardaran las palabras a
    reemplazar guardadas en in.tab
14     string reemplazos[entradas] {}; //arreglo donde se guardaran las palabras de
    reemplazo guardadas en in.tab
15     int iter = 0;
16     int otro =0;
17     while(!archivo1.eof()){
18         getline(archivo1 , texto1);
19         istringstream iss(texto1);
20         int k = 0;

```

```

21     while(!isstream.eof()){
22         string tempStr;
23         isstream >> tempStr;
24         if (k ==0 && iter<entradas && tempStr != "." && tempStr != "," ){
25             palabras[iter]=tempStr; //va guardando en un arreglo las palabras a
reemplazar guardadas en in.tab
26         }
27         if (k==2 && iter<entradas && tempStr != "." && tempStr != "," ){
28             reemplazos[iter]=tempStr; //va guardando en un arreglo las palabras
de reemplazo guardadas en in.tab
29         }
30         k=k+1;
31     }
32     iter=iter+1; //lleva el conteo de las palabras totales
33     int otro = otro +1;
34     if (iter == 10000){
35         cout<< "Llevo revisando 10000 palabras mil m s" <<endl;
36         otro =0;
37     }
38
39 }
40
41 archivo1.close();
42 cout<<"palabras a reemplazar: " <<endl;
43 for (int i=0; i<entradas; i++){
44     cout<<palabras[i]<< " reemplazo: " << reemplazos[i] << endl;
45 }
46 ifstream archivo2;
47 archivo2.open(filePath, ios::in);
48 string texto2;
49 ofstream salida;
50 salida.open("in.rep", ios::ate);
51 while(!archivo2.eof()){
52     getline(archivo2, texto2); //divide el texto en lineas
53     istringstream isstream(texto2);
54     string nuevo = {};
55     while(!isstream.eof()){ //divide las lineas en palabras
56         string tempStr;
57         isstream >> tempStr;
58         for (int j=0; j<entradas; j++){
59             if (tempStr==palabras[j] && tempStr != " ." && tempStr != " , " ){
60                 tempStr=reemplazos[j]; //va comparando si la palabra debe ser
guardada
61             }
62         }
63         nuevo.append(tempStr); //se agrega cada palabra para formar la linea
64         nuevo.append(" ");
65     }
66     salida<<nuevo<<endl; //se escribe cada linea del texto con las palabras
cambiadas
67 }
68 archivo2.close();
69 salida.close();
70 }
71
72 void Reemplazo::descomprimir (){
73     ifstream archivo1;
74     archivo1.open(tab_path, ios::in);
75     string texto1;

```

```

76     string palabras[entradas] {}; //arreglo donde se guardaran las palabras a
    reemplazar guardadas en in.tab
77     string reemplazos[entradas] {}; //arreglo donde se guardaran las palabras de
    reemplazo guardadas en in.tab
78     int iter = 0;
79     int otro =0;
80     while(!archivo1.eof()){
81         getline(archivo1, texto1);
82         istringstream isstream(texto1);
83         int k = 0;
84         while(!isstream.eof()){
85             string tempStr;
86             isstream >> tempStr;
87             if (k ==2 && iter<entradas && tempStr != "." && tempStr != "," ){
88                 palabras[iter]=tempStr; //va guardando en un arreglo las palabras a
    reemplazar guardadas en in.tab
89             }
90             if (k==0 && iter<entradas && tempStr != "." && tempStr != "," ){
91                 reemplazos[iter]=tempStr; //va guardando en un arreglo las palabras
    de reemplazo guardadas en in.tab
92             }
93             k=k+1;
94         }
95         iter=iter+1; //lleva el conteo de las palabras totales
96         int otro = otro +1;
97         if (iter == 10000){
98             cout<< "Llevo revisando 10000 palabras mil m s" <<endl;
99             otro =0;
100        }
101    }
102
103
104    archivo1.close();
105    cout<<"palabras a reemplazar: " <<endl;
106    for (int i=0; i<entradas; i++){
107        cout<<palabras[i]<< " reemplazo: " << reemplazos[i] << endl;
108    }
109    ifstream archivo2;
110    archivo2.open(filePath, ios::in);
111    string texto2;
112    ofstream salida;
113    salida.open("in-des.rep", ios::ate);
114    while(!archivo2.eof()){
115        getline(archivo2, texto2); //divide el texto en lineas
116        istringstream isstream(texto2);
117        string nuevo = {};
118        while(!isstream.eof()){ //divide las lineas en palabras
119            string tempStr;
120            isstream >> tempStr;
121            for (int j=0; j<entradas; j++){
122                if (tempStr==palabras[j] && tempStr != " ." && tempStr != " , " ){
123                    tempStr=reemplazos[j]; //va comparando si la palabra debe ser
    guardada
124                }
125            }
126            nuevo.append(tempStr); //se agrega cada palabra para formar la linea
127            nuevo.append(" ");
128        }
129        salida<<nuevo<<endl; //se escribe cada linea del texto con las palabras

```

```

    cambiadas
130     }
131     archivo2.close();
132     salida.close();
133 }
134
135 Reemplazo::Reemplazo(){};

```

## 5.2. Código Python:

```

1 # -*- coding: utf-8 -*-
2
3 #Writting first part
4 archivo = open("Historia_principito.txt", "w")
5
6 archivo.write("\n")
7 archivo.write("Antoine de Saint Exup ry falleci en 1944 un a o\n")
8 archivo.write("despu s de la publicaci n de El principito ,\n")
9 archivo.write("cuando en una misi n de reconocimiento se determin \n")
10 archivo.write("a su avi n desaparecido , siendo encontrados restos del avi n\n")
11 archivo.write("e identificaciones del autor y su\n")
12 archivo.write("esposa entre los a os 1998 y 2000.\n")
13 archivo.write("\n")
14 archivo.close()

```

```

1
2 archivo = open("Historia_principito.txt", "r")
3 for linea in archivo:
4     hilera = linea.replace("\n", "")
5     print(hilera)
6 archivo.close()

```

```

1 archivo = open("Tabla_Cambios.tab", "r")
2 for linea in archivo:
3     hilera = linea.replace("\n", "")
4     print(hilera)
5 archivo.close()

```

```

1 import re
2 import numpy
3 import os
4 import sys
5
6
7
8 replacements = {"urna": "~,~,^^-", "dolor": "-^^,~,~", "quis": "~^^:~~"}
9
10 file_a_leer = raw_input("Digite nombre del archivo (sin el .txt): ")
11 file_a_leer+="rep"
12
13 fname = raw_input("Digite nombre del archivo (con su .txt): ")
14
15 name = open(fname, 'r')
16
17 with name as infile, open(file_a_leer,"w") as outfile:
18     for line in infile:
19         for src, target in replacements.items():

```

```

20         line = line.replace(src , target)
21         outfile.write(line)
22 # ##archivo = name.read()
23 # ##divide = archivo.split()
24 #
25 # ##frecuenciaPalab = [divide.count(w) for w in divide]
26 #
27 # with name as infile , open(file_a_leer,"w") as outfile:
28 #     while line in infile:
29 #         while frecuenciaPalab > 3:
30 #             for src , target in replacements.items():
31 #                 line = line.replace(src , target)
32 #                 outfile.write(line)

1 # -*- coding: utf-8 -*-
2
3 #Writting first part
4 archivo = open("Tabla-Cambios.tab" , "w")
5
6 archivo.write("\n")
7 archivo.write("urna          ~,~,^^- \n")
8 archivo.write("dolor          -^^,~,~ \n")
9 archivo.write("quis          ~^^:~~ \n")
10
11 archivo.write("\n")
12 archivo.close()

```

## 6. Consideraciones

1. Equipos de 2 o 3 personas.
2. Genere un reporte en  $\text{\LaTeX}$  que incluya el enunciado, el diagrama de clases, sus conclusiones y, como anexo incluya su código fuente
3. Suba su código y documentación (doxygen, README, INSTALL) al GitLab respectivo de su grupo y el directorio del laboratorio.
4. Cada estudiante debe subir el reporte a Schoology.
5. Recuerde que por cada día tardío de entrega se le rebajaran puntos de acuerdo con la formula:  $4^d$  donde  $d > 1$  es la cantidad de días tardíos