

Índice

1. Introducción.....	2
2. Desarrollo de la práctica y resultados.....	3
Apéndice A: Código Fuente	6

1. Introducción

En la segunda práctica estudiaremos la estructura del lenguaje. Para ello primero generaremos una aproximación cero al idioma español considerando todos los símbolos equiprobables (incluido el espacio). Después una primera aproximación usando el mismo proceso de creación de una tabla de frecuencias de la práctica 1, pero añadiendo un espacio. A continuación generaremos una segunda aproximación al idioma español considerando una fuente de Markov de orden uno y utilizando el método para obtener las probabilidades condicionales de orden uno. Por último haremos una cuarta aproximación al idioma español usando una fuente de memoria nula que emite palabras en lugar de letras.

2. Desarrollo de la práctica y resultados

Para el desarrollo de esta práctica hemos decidido crear un programa que nos calcule y genere todos los resultados que queremos. El código fuente de dicho programa se encuentra en el apéndice A. A continuación describiremos el funcionamiento del programa y los resultados obtenidos.

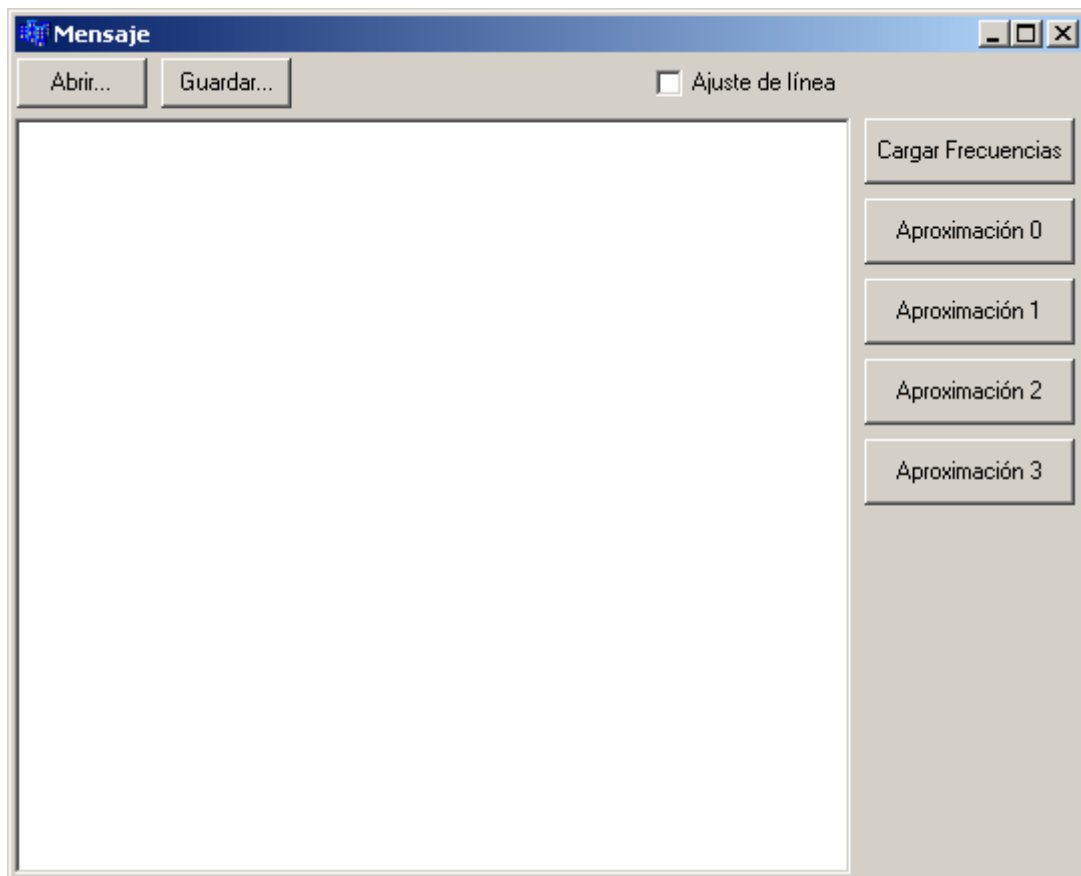


Figura 1: Programa

En la figura 1 tenemos una captura de pantalla de nuestro programa. Como se ve es un programa sencillo. Dado que la aproximación 0 al idioma implica que todos los símbolos son equiprobables, directamente pulsando en el botón “Aproximación 0” nos muestra un texto aleatorio con dichas consideraciones. El resultado es el mostrado en la figura 2.

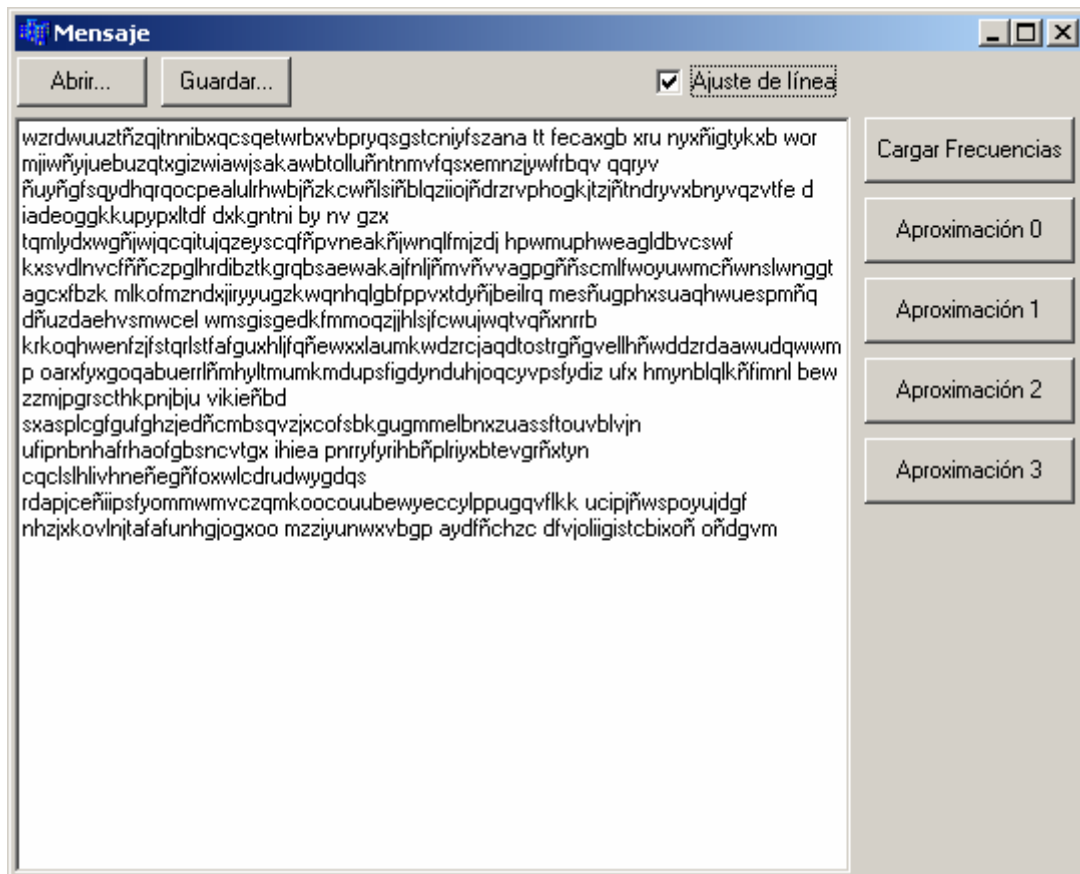


Figura 2: Aproximación 0

Para la Aproximación 1 usaremos un fichero de frecuencias, para ello pulsamos el botón “Abrir...” y abrimos el fichero frecs.txt. A continuación pulsamos el botón “Cargar Frecuencias” y después “Aproximación 1” y el resultado es el obtenido en la Figura 3.

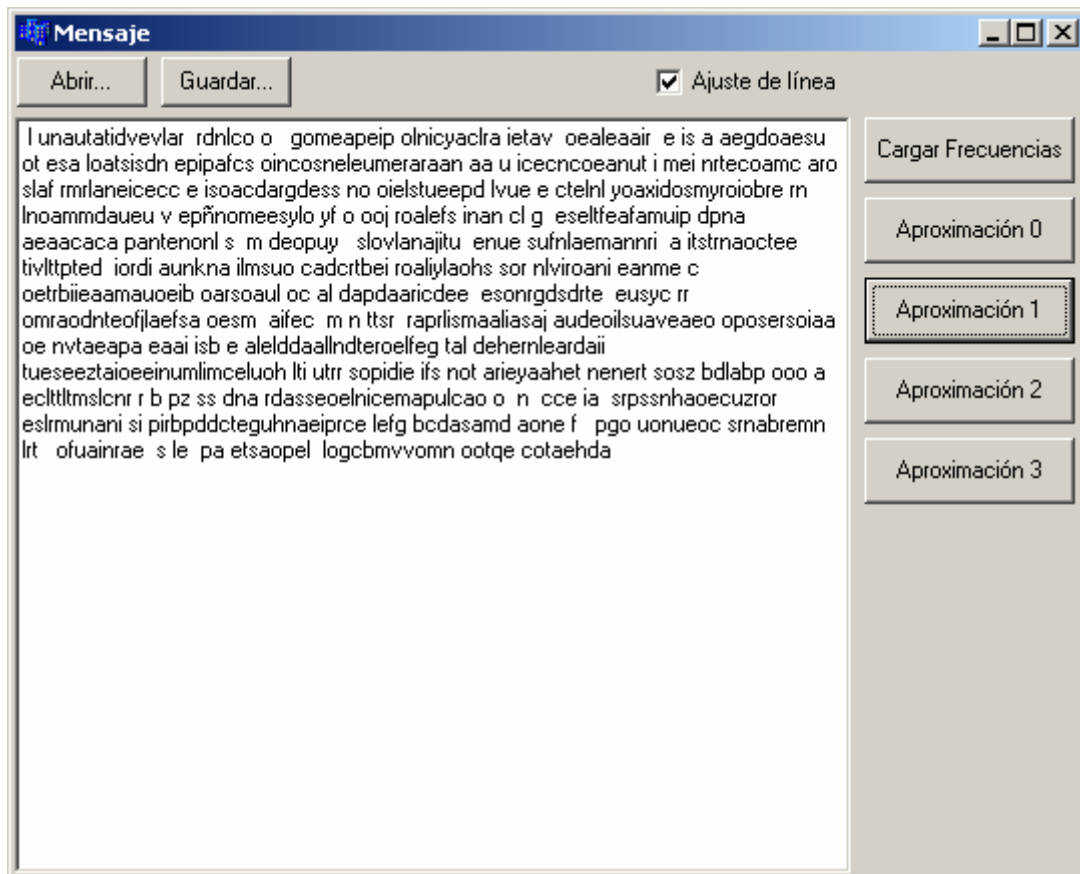


Figura 3: Aproximación 1

Para la aproximación 2 es necesario cargar un texto primero, por lo que pulsamos en el botón “Abrir...” y abrimos el fichero texto.txt. Pulsamos el botón “Aproximación 2” y se genera una 2º aproximación al idioma español como muestra la figura 4.

Además, hemos incluido una 3º aproximación al idioma español como parte opcional, aunque tarda bastante en ejecutarse por motivos obvios.

Apéndice A: Código Fuente

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
  
#include <iostream>  
#include <fstream>  
  
TfrmMensaje *frmMensaje;  
  
/*****  
IS CHARACTER  
*****/  
int is_char(char ch)  
{  
    if(ch >='a' && ch <='z') return 1;  
    if(ch >='A' && ch <='Z') return 1;  
    if(ch == 'ñ' || ch == 'Ñ') return 1;  
    return 0;  
}  
  
void mostrar_pareja(char ch1, char ch2)
```

```

{
    ch1 = ch1=='ñ' ? 0xA4 : ch1;
    ch2 = ch2=='ñ' ? 0xA4 : ch2;
    //cout << ch1 << ch2 << "\t";
}

char to_minuscula(char ch)
{
    if(ch >='A' && ch <='Z') return ch-'A'+'a';
    if(ch == 'Ñ') return 'ñ';
    else return ch;
}

/*char lsc(ifstream handle)
{
    char ch;
    ch = (char)handle.get();
    return ch;
} */

//-----

__fastcall TfrmMensaje::TfrmMensaje(TComponent* Owner)
    : TForm(Owner)
{
    srand(time(NULL));
}

//-----

```

```

void __fastcall TfrmMensaje::btnAbrirClick(TObject *Sender)
{
    if(dlgAbrir->Execute())
        txtBox->Lines->LoadFromFile(dlgAbrir->FileName);
}
//-----
void __fastcall TfrmMensaje::btnGuardarClick(TObject *Sender)
{
    if(dlgGuardar->Execute())
        txtBox->Lines->SaveToFile(dlgGuardar->FileName);
}

//-----
void __fastcall TfrmMensaje::FormResize(TObject *Sender)
{
    txtBox->Height = Height - txtBox->Top - 27;
    txtBox->Width = Width - txtBox->Left - 124;
}
//-----
void __fastcall TfrmMensaje::chkAjusteClick(TObject *Sender)
{
    txtBox->WordWrap = chkAjuste->Checked;
}

//-----

#define NLETRAS 28

// Frecuencia relativa de las letras:

```

```

double prob_abc[NLETRAS];

// Frecuencia absoluta de las letras
double f_abs[NLETRAS];

void __fastcall TfrmMensaje::btnFreccsClick(TObject *Sender)
{
//   ifstream handle("probab.txt");
   for(int i=0; i<NLETRAS; i++){
       prob_abc[i] = txtBox->Lines->Strings[i].ToDouble();
   }

       f_abs[0] = prob_abc[0];
       for(int i=1; i<NLETRAS; i++) f_abs[i] = f_abs[i-1] + prob_abc[i];

//   handle.close();
}

```

```
//-----
```

```
#define RISTRATAM 1024
```

```

void __fastcall TfrmMensaje::btnAprox0Click(TObject *Sender)
{
    char ristra[RISTRATAM+1];

```

```

for(int i=0; i<RISTRATAM; i++)
{
    char ch = rand()%28;
    if(ch == 27) ch = ' ';
    else if(ch == 26) ch = 'ñ';
    else ch = (ch+'a');
    ristra[i] = ch;
}
ristra[RISTRATAM] = '\0';
txtBox->Lines->Clear();
txtBox->Lines->Add(AnsiString(ristra));
//cout << ristra << endl;
}

```

//-----

```

/*****
ELEGIR LETRA
*****/
char elegirletra(double v, double *vector)
{
    char ch;

    int j = 0;
    while(v > vector[j]) j++;
}

```

```

        if(j == 27) ch = ' ';
        else if(j == 26) ch = 'ñ';
        else ch = (j+'a');

        return ch;
    }

/*****

ALEATORIO
*****/

double Aleatorio(void)
{
    return (double)rand() / (double) RAND_MAX;
}

void __fastcall TfrmMensaje::btnAprox1Click(TObject *Sender)
{
    char ristra[RISTRATAM+1];

    for(int i=0; i<RISTRATAM; i++)
    {
        double v = Aleatorio();
        ristra[i] = elegirletra(v, f_abs);
    }
    ristra[RISTRATAM] = '\0';
    txtBox->Lines->Clear();
    txtBox->Lines->Add(AnsiString(ristra));
}

```

```

//-----

/*****

OBTENER INDICE
*****/

inline int obtener_indice(char ch)
{
    if(ch >= 'a' && ch <= 'z') return ch-'a';
    if(ch == 'ñ') return 26;
    return 27;
}

double matriz_frec[NLETRAS][NLETRAS];

void __fastcall TfrmMensaje::btnAprox2Click(TObject *Sender)
{
    char ristra[RISTRATAM];
    // int suma = 0;

    for(int i=0; i<NLETRAS; i++)
    for(int j=0; j<NLETRAS; j++)
        matriz_frec[i][j] = 0.0;

    char ch1 = txtBox->Lines->Text[1];

    int Largo = txtBox->Lines->Text.Length();

```

```

for(int i = 2; i <= Largo; i++){
    char ch2 = txtBox->Lines->Text[i];
    matriz_frec[obtener_indice(ch1)][obtener_indice(ch2)] ++;
    ch1 = ch2;
//    suma++;
};

for(int i=0; i<NLETRAS; i++)
{
    double suma_fila = 0.00000000000000000001;

    for(int j=0; j<NLETRAS; j++)
suma_fila += matriz_frec[i][j];

    for(int j=0; j<NLETRAS; j++)
matriz_frec[i][j] /= suma_fila;

    for(int j=1; j<NLETRAS; j++)
matriz_frec[i][j] += matriz_frec[i][j-1];
}

double v = Aleatorio();
ch1 = elegirletra(v, f_abs);
ristra[0] = ch1;

for(int i=1; i<RISTRATAM; i++)
{
    v = Aleatorio();

    int k = obtener_indice(ch1);
    char ch2 = elegirletra(v, matriz_frec[k]);
    ristra[i] = ch2;
    ch1 = ch2;
}

```

```

    }
    ristra[RISTRATAM] = '\0';
    txtBox->Lines->Clear();
    txtBox->Lines->Add(AnsiString(ristra));
}
//-----

double matriz3d_frec[NLETRAS][NLETRAS][NLETRAS];

void __fastcall TfrmMensaje::btnAprox3Click(TObject *Sender)
{
    //    int suma = 0;

    for(int i=0; i<NLETRAS; i++)
    for(int j=0; j<NLETRAS; j++)
    for(int k=0; k<NLETRAS; k++)
        matriz3d_frec[i][j][k] = 0.0;
/*
    cout << "Introduzca fichero a tratar (3 aproximacion): " << endl;
    cin >> filename1;

    ifstream in(filename1);
*/

    char ch1, ch2, ch3;

    ch1 = txtBox->Lines->Text[1];

    int Largo = txtBox->Lines->Text.Length();

    for(int i=3; i <= Largo; i++){
        ch2 = txtBox->Lines->Text[i-1];

```

```

        ch3 = txtBox->Lines->Text[i];

matriz3d_frec[obtener_indice(ch1)][obtener_indice(ch2)][obtener_indice(ch3)]
++;

        ch1 = ch2;
        ch2 = ch3;
        // suma++;
    };

// in.close();

for(int i=0; i<NLETRAS; i++)
for(int j=0; j<NLETRAS; j++){
    double suma_fila = 0.000000000000000001;

        for(int k=0; k<NLETRAS; k++)
suma_fila += matriz3d_frec[i][j][k];

        for(int k=0; k<NLETRAS; k++)
matriz3d_frec[i][j][k] /= suma_fila;

        for(int k=1; k<NLETRAS; k++)
matriz3d_frec[i][j][k] += matriz3d_frec[i][j][k-1];
    }

char ristra[RISTRATAM];

        ch1 = elegirletra(Aleatorio(), f_abs);
        ristra[0] = ch1;

int j = obtener_indice(ch1);
        ch2 = elegirletra(Aleatorio(), matriz_frec[j]);
        ristra[1] = ch2;

```

```
for(int i=2; i<RISTRATAM; i++)
{
    double v = Aleatorio();

    int k = obtener_indice(ch2);
    ch3 = elegirletra(v, matriz3d_frec[j][k]);
    ristra[i] = ch3;

    ch1 = ch2;
    ch2 = ch3;
    j = k;
}
ristra[RISTRATAM] = '\0';
txtBox->Lines->Clear();
txtBox->Lines->Add(AnsiString(ristra));
}
//-----
```