



**Instituto Politécnico Nacional**  
*Escuela Superior de Cómputo*



**Genetic Algorithms**  
Profesor: M. en C. Maria Elena Cruz Meza

**Practica 1**  
**Graficando una lista de datos generados aleatoriamente.**

Grupo: 3CM8

Objetivo: Mostrar los conocimientos en la manipulación de histogramas.

Pérez Garduño José Emiliano

*Fecha: 10 de septiembre de 2019*

## Marco Teórico:

Histograma: Es una gráfica de la distribución de un conjunto de datos. Es un tipo especial de gráfica de barras, en la cual una barra va pegada a la otra, es decir no hay espacio entre las barras. Cada barra representa un subconjunto de los datos.

Un histograma muestra la acumulación ó tendencia, la variabilidad o dispersión y la forma de la distribución, es una gráfica adecuada para representar variables continuas, aunque también se puede usar para variables discretas. Es decir, mediante un histograma se puede mostrar gráficamente la distribución de una variable cuantitativa o numérica.

## Desarrollo de la Práctica:

### main.c :

```
/*
Codigo para generar una lista de numeros aleatorios y mostrar su histograma
Autor: José Emiliano Pérez Garduño
Materia: Genetic Algorithms
Grupo: 3CM8
*/
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "practica0.h"

int main(int argc, char const *argv[])
{
    Practica0();
    printf("\n\nPrograma para mostrar una lista de 10 a N numeros aleatorios junto a su histograma\n");
    double * Lista;
    Lista = CrearLista(Lista);
    Lista = FuncionCuadrado(Lista);
    MostrarHistograma(Lista);
    return 0;
}

int LongitudLista(double * L)
{
    int i = 0;
    while(L[i] != 0)
        i++;
    return i;
}

double * CrearLista(double *L)
{
    int n;
    printf("Introducir cuantos numeros a partir de 10 tendra la lista (10 + n)\n");
    scanf("%d",&n);
    printf("Se creara una lista de tamano: 10 + %d = %d\n", n, 10+n);
    L = calloc(sizeof(double) * (10 + n),0);
    L = PoblarLista(n,L);
    return L;
}

double * PoblarLista(int n, double * L)
{
    srand(time(NULL));
    int i;
    for (i = 0; i < 10+n; i++)
        L[i] = rand() % 50 + 1;
    MostrarLista(L);
    return L;
}

void MostrarLista(double *L)
{
    printf("Lista de Numeros Aleatorios -> ");
    int i;
    int l = LongitudLista(L);
    printf("A[%d]=(",l);
    for(i = 0; i < l; i++)
    {
        if (i == l-1)
            printf("%.0f)\n",L[i]);
        else
            printf("%.0f,",L[i]);
    }
}
```

```

double * FuncionCuadrado(double * L)          //funcion para evaluar numeros enteros f(x) = x^2
{
    int i;
    int l = LongitudLista(L);
    for (i = 0; i < l; i++)
    {
        L[i] = L[i] * L[i];
    }

    printf("\nLista de Numeros x^2 -> A^2[%d]={"", l);
    for(i = 0; i < l; i++)
    {
        if (i == l-1)
            printf("%0.0lf}\n", L[i]);
        else
            printf("%0.0lf,", L[i]);
    }
    fflush(stdin);
    getchar();
    fflush(stdin);
    getchar();
    return L;
}

void MostrarHistograma(double *L)
{
    printf("\nHistograma: \n\n");
    int i,j,ubicacion;
    int l = LongitudLista(L);
    ubicacion = NumeroMaximo(l,L);
    //Parte del eje Y
    printf("El numero maximo es: %0.1f\n", L[ubicacion]);

    printf("-----\n");

    for (i = L[ubicacion]; i > 0; --i) {
        printf("%3d|", i);
        for (j = 0; j < l; ++j)
            (L[j] >= i) ? printf(" X ") : printf(" ");
        printf("\n");
    }
    printf(" ");

    for (i = 0; i < l; ++i) //Parte del eje X
        printf("%3d", i+1);
    printf("\n-----\n");
}

int NumeroMaximo(int l, double *L)
{
    double max;
    max = L[0];
    int i;
    int ubicacion;
    for (i = 0; i < l; i++)
    {
        if(L[i] > max)
        {
            max = L[i];
            ubicacion = i;
        }
    }
    return ubicacion;
}

int ConseguirLongitudLista(char * L)          //Función para conseguir la longitud de una lista
{
    int l;
    l = 0;
    while (L[l] != '\0')
        l++;
    return l;
}

```

```

char * PoblarListaPractica0(int l, char * L)
{
    int i, n, fin, inicio;
    time_t t;
    srand((unsigned) time(&t));
    printf("Escribir el inicio del rango de numeros aleatorios\n");
    scanf("%d",&inicio);
    printf("Escribir el fin del rango de numeros aleatorios:\n");
    scanf("%d",&fin);
    for (i = 0; i <= l; i++)
    {
        if(i == l)
        {
            L[i] = '\0';
        }
        else
        {
            n = rand() % (fin-inicio) + inicio;
            L[i] = n;
        }
    }
    return L;
}

void MostrarListaPractica0(char * L)
{
    int l = ConseguirLongitudLista(L);
    int i;
    printf("\n");
    for (i = 0; i < l; i++)
    {
        if(i != l-1)
            printf("%d,",L[i]);
        else
            printf("%d\n",L[i]);
    }
}

char * AcomodarLista(char * L)
{
    int i, j, aux;
    int l = ConseguirLongitudLista(L);
    printf("Acomodando Lista.\n");
    for(i = 0; i < l; i++)
    {
        for(j = i + 1; j < l; j++)
        {
            if(L[i] >= L[j])
            {
                aux = L[j];
                L[j] = L[i];
                L[i] = aux;
            }
        }
    }
    MostrarListaPractica0(L);
    return L;
}

void ListaBinario(char * L)
{
    int i, l;
    l = ConseguirLongitudLista(L);
    printf("Index\tNumeroEntero\tNumeroBinario\n");
    for (i = 0; i < l; i++)
    {
        printf("%d.- \t%d\t\t",i,L[i]);
        NumBinario(L[i]);
    }
}

void NumBinario(int num)
{
    int i, j;
    for (i = 15; i >= 0; i--)
    {
        j = num >> i;
        if (j & 1)
            printf("1");
        else
            printf("0");
    }
    printf("\n");
}

```

```

[emiliano@localhost Practical]$ gcc main.c -o practical
[emiliano@localhost Practical]$ ./practical
Programa para generar una lista de numeros aleatorios.

Escribir cuantos numeros quiere que tenga la lista:
10
Creando Lista de 10 numeros.
Escribir el inicio del rango de numeros aleatorios
0
Escribir el fin del rango de numeros aleatorios:
100
[69,92,59,62,32,39,7,5,87,43]
Acomodando Lista.
[5,7,32,39,43,59,62,69,87,92]
Index  NumeroEntero  NumeroBinario
0.-    5             0000000000000101
1.-    7             0000000000000111
2.-    32            0000000000100000
3.-    39            0000000000100111
4.-    43            0000000000101011
5.-    59            0000000000111011
6.-    62            0000000000111110
7.-    69            00000000001000101
8.-    87            00000000001010111
9.-    92            00000000001011100

Programa para mostrar una lista de 10 a N numeros aleatorios junto a su histograma
Introducir cuantos numeros a partir de 10 tendra la lista (10 + n)
3
Se creara una lista de tamano: 10 + 3 = 13
Lista de Numeros Aleatorios -> A[13]={12,72,84,21,75,6,63,55,97,11,23,24,98}

Lista de Numeros x^2 -> A^2[13]={144,5184,7056,441,5625,36,3969,3025,9409,121,529,576,9604}

Histograma:

El numero maximo es: 9604
-----
9604|          X
9603|          X
9602|          X
9601|          X
049| X X X X X X X X X X X X X
048| X X X X X X X X X X X X X
047| X X X X X X X X X X X X X
046| X X X X X X X X X X X X X
045| X X X X X X X X X X X X X
044| X X X X X X X X X X X X X
043| X X X X X X X X X X X X X
042| X X X X X X X X X X X X X
041| X X X X X X X X X X X X X
040| X X X X X X X X X X X X X
039| X X X X X X X X X X X X X
038| X X X X X X X X X X X X X
037| X X X X X X X X X X X X X
036| X X X X X X X X X X X X X
035| X X X X X X X X X X X X X
034| X X X X X X X X X X X X X
033| X X X X X X X X X X X X X
032| X X X X X X X X X X X X X
031| X X X X X X X X X X X X X
030| X X X X X X X X X X X X X
029| X X X X X X X X X X X X X
028| X X X X X X X X X X X X X
027| X X X X X X X X X X X X X
026| X X X X X X X X X X X X X
025| X X X X X X X X X X X X X
024| X X X X X X X X X X X X X
023| X X X X X X X X X X X X X
022| X X X X X X X X X X X X X
021| X X X X X X X X X X X X X
020| X X X X X X X X X X X X X
019| X X X X X X X X X X X X X
018| X X X X X X X X X X X X X
017| X X X X X X X X X X X X X
016| X X X X X X X X X X X X X
015| X X X X X X X X X X X X X
014| X X X X X X X X X X X X X
013| X X X X X X X X X X X X X
012| X X X X X X X X X X X X X
011| X X X X X X X X X X X X X
010| X X X X X X X X X X X X X
009| X X X X X X X X X X X X X
008| X X X X X X X X X X X X X
007| X X X X X X X X X X X X X
006| X X X X X X X X X X X X X
005| X X X X X X X X X X X X X
004| X X X X X X X X X X X X X
003| X X X X X X X X X X X X X
002| X X X X X X X X X X X X X
001| X X X X X X X X X X X X X
    1  2  3  4  5  6  7  8  9 10 11 12 13
-----
[emiliano@localhost Practical]$

```

```

void Practica0()
{
    int lBin;
    char *LBin;
    printf("Programa para generar una lista de numeros aleatorios.\n\n");
    printf("Escribir cuantos numeros quiere que tenga la lista:\n");
    scanf("%d",&lBin);
    printf("Creando Lista de %d numeros.\n",lBin);
    LBin = calloc(sizeof(int) * (lBin+1),0);
    LBin = PoblarListaPractica0(lBin,LBin);
    MostrarListaPractica0(LBin);
    LBin = AcomodarLista(LBin);
    ListaBinario(LBin);
    getch();
}

```

## practica0.h :

```

#include <stdio.h>
int ConseguirLongitudLista(char * L); //Función para conseguir la longitud y crear la lista
char * PoblarListaPractica0(int l, char * L); //Función para poblar aleatoriamente la lista
void MostrarListaPractica0(char * L); //Función para mostrar la Lista
char * AcomodarLista(char * L); //Función para acomodar la Lista
void ListaBinario(char * L); //Función para imprimir la lista en binario
void NumBinario(int n); //Función para imprimir un numero entero en binario (16 bits)
void Practica0(); //Función para realizar la practica 0
double * CrearLista(); //Función para CrearLista de doubles
double * PoblarLista(int n, double * L); //Función para Poblar la lista creada
double * FuncionCuadrado(double * L); //Función para elevar al cuadrado a la lista
int LongitudLista(double * L); //Función para conseguir la longitud de una lista double
void MostrarLista(double * L); //Función para mostrar lista double
void MostrarHistograma(double *L); //Función para imprimir histograma
int NumeroMaximo(int l, double *L); //Función para encontrar el numero mas grande

```

## Referencias:

<http://asesorias.cuautitlan2.unam.mx/Laboratoriovirtualdeestadistica/DOCUMENTOS/TEMA%201/7.%20HISTOGRAMAS.pdf>