

TEXTO DEL PROGRAMA EN C++ BORLAND 6

```
/------  
#include <vcl.h>  
#pragma hdrstop  
#include "Unit1.h"  
#include "mem.h"  
#include "string.h"  
#include "math.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//Tamaño de las imagenes  
const int imHeight = 150;  
const int imWidth = 330;  
const int D1=16;  
const int gray_max = 255;  
const int gray_min = 0;  
int num_elem=imHeight*imWidth;  
int h = 40;  
int w = 40;  
int image_min = gray_max;  
int image_max = gray_min;
```

```
const numOfClasses = 3;
int image[imHeight][imWidth];
//Definición de graphics
Graphics::TBitmap *ImageBitmap=new Graphics::TBitmap();
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Imagen1Click(TObject *Sender)
{
    int i,j,br,br1;
    TOpenDialog *dlg = new TOpenDialog(this);
    dlg->Title="Open Micro Image";
    dlg->Filter="Windows Bitmap *.bmp";
    dlg->InitialDir="TextureImageBase";
    dlg->Execute();
    ImageBitmap->LoadFromFile(dlg->FileName);
    TStrings *dlg->Files();
    for(i=0;i<imHeight;i++)
        for(j=0;j<imWidth;j++)
            {
                br=ImageBitmap->Canvas->Pixels[j][i];
```

```
        br1=(br & 0xff)+((br>>8)&0xff)+((br>>16)&0xff);
        br1=br1/3;
        image[i][j]=br1;
        if (br1>image_max)
            image_max=br1;
        else if (br1<image_min)
            image_min=br1;
    }
    ShowMessage("min:"+IntToStr(image_min)"
max:"+IntToStr(image_max));
    try
    {
        Canvas->Draw(5,5,ImageBitmap);
    }
    catch (...)
    {
        MessageBeep(0);
    }
    delete dlg;
}
//-----
//-----
void __fastcall TForm1::Contraste1Click(TObject *Sender)
{
```

```
int
ContrastHistogram[D1],Contrast[imHeight][imWidth],d,i,j,d1,d2,m,aux,imHeight2,imWidth
2;

int ContrastHistogram_min=gray_max;
int ContrastHistogram_max=gray_min;
int ValMax=0;
int k=15;
memset(ContrastHistogram,0,64);
imHeight2=imHeight-1;
imWidth2=imWidth-1;
for(i=0;i<imHeight2;i++)
    for(j=0;j<imWidth2;j++)
        {
            d1=image[i][j]-image[i][j+1];
            d2=image[i+1][j]-image[i][j+1];
            if (d1<0) d1=-d1;
            if (d2<0) d2=-d2;
            if (d1<d2)
                d=d1;
            else
                d=d2;
            Contrast[i][j]=d1; //matriz de contrastes
            if (d>ContrastHistogram_max)
                ContrastHistogram_max=d;
            else if (d<ContrastHistogram_min)
```

```
        ContrastHistogram_min=d;
    }
    for(i=0;i<imHeight2;i++)
        for(j=0;j<imWidth2;j++)
            ValMax=ValMax+Contrast[i][j];

    ShowMessage("Contraste min:"+IntToStr(ContrastHistogram_min)+" Contrast
max:"+IntToStr(ContrastHistogram_max));

    aux=ContrastHistogram_max-ContrastHistogram_min;
    for(i=0;i<imHeight2;i++)
        for(j=0;j<imWidth2;j++)
            {
                m=((Contrast[i][j]-ContrastHistogram_min)*k)/aux;
                ContrastHistogram[m]++;
            }
        for(j=0;j<D1-1;j++)
            {
                Canvas->TextOutA(350+j*40,410,IntToStr(ContrastHistogram[j]));
                Canvas->Rectangle(425+j*40,400,360+j*40,400-
(500*ContrastHistogram[j]/num_elem));
            }
    }
//-----
```