
CODIGO GRAFICADOR 3D

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.IO;
using System.Drawing.Imaging;
using System.Windows.Forms; //PictureBox

namespace graficador3D
{
    class unitario3D
    {
        int x, y, z, ancho, alto, centrox, centroy;
        public Graphics grafico;
        public Pen lapiz = new Pen(Color.Violet, 0.9f);
        public Pen lapiz2 = new Pen(Color.Cyan, 0.9f);
        public Font letra = new Font("Arial", 5, FontStyle.Italic, GraphicsUnit.Pixel);
        public SolidBrush rrelleno = new SolidBrush(Color.Cyan);
        public struct punto2D
        {
            public float xr, yr; //coordenadas del punto a graficar

            public punto2D(double x, double y, double z)
            {
                xr = (float)(y - x * (Math.Sqrt(2) / 2));
                yr = (float)(z - x * (Math.Sqrt(2) / 2));
            }
            public punto2D(punto3D punto)
            {
                xr = (float)(punto.Y - punto.X * (Math.Sqrt(2) / 2));
                yr = (float)(punto.Z - punto.X * (Math.Sqrt(2) / 2));
            }
        }
        public struct punto3D
        {
            public double X, Y, Z; //coordenadas en 3D de un punto
```

```

public punto3D(double x, double y, double z)
{
    X = x;
    Y = y;
    Z = z;
}
}

public void set_x(int valor)
{
    x = valor;
}

public void set_y(int valor)
{
    y = valor;
}

public void DibujarPlanoCartesiano(PictureBox pbox)
{
    int centroX = pbox.Width / 2, centroY = pbox.Height / 2;
    Bitmap bmp = new Bitmap(pbox.Width, pbox.Height);
    grafico = Graphics.FromImage(bmp); //Desde un pictureBox existente
    grafico.Clear(Color.Transparent);
    grafico.SmoothingMode = SmoothingMode.HighQuality;
    grafico.TranslateTransform(centroX, centroY);
    grafico.ScaleTransform(1, -1);
    grafico.DrawLine(lapiz, centroX * -1, 0, centroX * 2, 0); //dibujamos eje x
    grafico.DrawLine(lapiz, 0, centroY, 0, centroY * -1); //dibujamos eje y
    //subdivisiones de los ejes:
    for (int i = -centroX; i < centroX; i += 8)
    {
        //divisiones para el ejeY
        grafico.DrawLine(lapiz, 5, i, -5, i); //linea horizontal
        //divisiones para el ejeX
        grafico.DrawLine(lapiz, i, 5, i, -5); //linea vertical
    }
    pbox.Image = bmp;
}

public Point Convertir3Da2D(float cpx, float cpy, float cpz)
{
    double xr, yr;
    xr = cpy - cpx * (Math.Sqrt(2) / 2);
    yr = cpz + cpx * (Math.Sqrt(2) / 2);
    Point p2DGraficar = new Point((int)xr, (int)yr);
    return p2DGraficar;
}

```

```

public void dibujarEjes3D(PictureBox pbox)
{
    int centroX = pbox.Width / 2, centroY = pbox.Height / 2;

    Bitmap bmp = new Bitmap(pbox.Width, pbox.Height);
    grafico = Graphics.FromImage(bmp);
    grafico.Clear(Color.Transparent);
    grafico.SmoothingMode = SmoothingMode.HighQuality;
    grafico.TranslateTransform(centroX, centroY);
    grafico.ScaleTransform(1, -1);

    // lapiz.CustomStartCap = new AdjustableArrowCap(2, 4, false);
    lapiz.CustomEndCap = new AdjustableArrowCap(2, 4, false); //ponemos punta al vector dibujado

    // dibujamos los tres vectores unitarios en 3D
    grafico.DrawLine(lapiz, 0, 0, -centroX * (float)(Math.Sqrt(2) / 2), -centroY * (float)(Math.Sqrt(2) / 2)); //x
    grafico.DrawLine(lapiz, 0, 0, centroX, 0); //y
    grafico.DrawLine(lapiz, 0, 0, 0, centroY); //z

    grafico.DrawEllipse(lapiz, (float)-5.4142, (float)-4.4142, 1, 1); //dibujar punto

    grafico.DrawLine(lapiz, 0, 0, (float)-5.4142, (float)-10.58); //dibujar linea

    pbox.Image = bmp;
}

public void dibujarLinea3D(PictureBox pbox)
{
    int centroX = pbox.Width / 2, centroY = pbox.Height / 2;
    Bitmap bmp = new Bitmap(pbox.Width, pbox.Height);
    grafico = Graphics.FromImage(bmp);
    grafico.Clear(Color.Transparent);
    grafico.SmoothingMode = SmoothingMode.HighQuality;
    grafico.TranslateTransform(centroX, centroY);
    grafico.ScaleTransform(1, -1);

    lapiz.CustomEndCap = new AdjustableArrowCap(2, 4, false); //ponemos punta al vector dibujado

    // dibujamos los tres vectores unitarios en 3D
    grafico.DrawLine(lapiz, 0, 0, -centroX * (float)(Math.Sqrt(2) / 2), -centroY * (float)(Math.Sqrt(2) / 2)); //x
    grafico.DrawLine(lapiz, 0, 0, centroX, 0); //y
    grafico.DrawLine(lapiz, 0, 0, 0, centroY); //z

    grafico.DrawLine(lapiz, 0, 0, (float)-5.4142, (float)-10.58); //dibujar la linea

```

```

pbox.Image = bmp;
}

public void dibujarEjes(PictureBox pbox)
{
    int centroX = pbox.Width / 2, centroY = pbox.Height / 2;
    Bitmap bmp = new Bitmap(pbox.Width, pbox.Height);
    // grafico = pbox.CreateGraphics(); //Desde un pictureBox existente para ver solo este basta sin crear el bitmap
    grafico = Graphics.FromImage(bmp);
    grafico.Clear(Color.Transparent);
    grafico.SmoothingMode = SmoothingMode.HighQuality;
    grafico.TranslateTransform(centroX, centroY); //grafico.TranslateTransform(centroX, centroY);
    grafico.ScaleTransform((float)1, -1); //grafico.ScaleTransform(1, -1);

    lapiz.CustomEndCap = new AdjustableArrowCap(2, 3, false); //ponemos punta al vector dibujado
    // dibujamos los tres vectores unitarios en 3D
    grafico.DrawLine(lapiz, 0, 0, -centroX * (float)(Math.Sqrt(2) / 2), -centroY * (float)(Math.Sqrt(2) / 2)); //x
    grafico.DrawLine(lapiz, 0, 0, centroX, 0); //y
    grafico.DrawLine(lapiz, 0, 0, 0, centroY); //z

    lapiz2.CustomEndCap = new AdjustableArrowCap(2, 3, false);
    // dibujamos los tres - vectores unitarios en 3D
    grafico.DrawLine(lapiz2, 0, 0, centroX * (float)(Math.Sqrt(2) / 2), centroY * (float)(Math.Sqrt(2) / 2)); //x
    grafico.DrawLine(lapiz2, 0, 0, -centroX, 0); //y
    grafico.DrawLine(lapiz2, 0, 0, 0, -centroY); //z

    dibujarPunto(25.36, 89.36, 47.23, 1);
    dibujarLinea(25.36, 89.36, 47.23, 50.369, 78.26, 45.219);

    Random ram = new Random();
    List<punto3D> nube = new List<punto3D>();
    for (int i = 0; i < 100; i++)
    {
        nube.Add(new punto3D(25 + ram.Next(0, 100), 35 - ram.Next(0, 100), 0.004 * ram.Next(0, 50)));
    }
    dibujarNubePuntos(nube);
    dibujarPlanoCuadrado();

    //for (float i = 0; i < 500; i+=5)
    //{
    //    dibujarLinea(25.36, 89.36, 47.23, 50.369+i, 78.26+i, 45.219+i);
    //    dibujarPunto(i, i, i, 2);
    //}

```

```

// dibujar cualquier plano 3D - inicio
//mirar el video de jugen en youtube
// dibujar cualquier plano 3D - fin

pbox.Image = bmp;
}
public void dibujarPlanoCuadrado()
{
    // metodo 1
    punto3D p1 = new punto3D(100, 100, 150);
    punto3D p2 = new punto3D(200, 100, 150);
    punto3D p3 = new punto3D(200, 200, 150);
    punto3D p4 = new punto3D(100, 200, 150);
    dibujarLinea(p1, p2);
    dibujarLinea(p2, p3);
    dibujarLinea(p3, p4);
    dibujarLinea(p4, p1);
    // metodo 2
    dibujarLinea(p1.X, p1.Y, p1.Z, p2.X, p2.Y, p2.Z);
    dibujarLinea(p2.X, p2.Y, p2.Z, p3.X, p3.Y, p3.Z);
    dibujarLinea(p3.X, p3.Y, p3.Z, p4.X, p4.Y, p4.Z);
    dibujarLinea(p4.X, p4.Y, p4.Z, p1.X, p1.Y, p1.Z);

    // dibujar el relleno
    punto2D pun1 = new punto2D(p1.X, p1.Y, p1.Z);
    punto2D pun2 = new punto2D(p2.X, p2.Y, p2.Z);
    punto2D pun3 = new punto2D(p3.X, p3.Y, p3.Z);
    punto2D pun4 = new punto2D(p4.X, p4.Y, p4.Z);
    // Create solid brush.
    SolidBrush blueBrush = new SolidBrush(Color.FromArgb(150, 23, 56, 78));
    // Color.FromArgb(127, 23, 56, 78) 127=transparencia, 23=rojo, 56=verde, 78=azul;
    // todos los valores comprendidos entre 0 y 255, (0, 255)
    // Color.Blue

    // Create points that define polygon.
    PointF point1 = new PointF(pun1.xr, pun1.yr);
    PointF point2 = new PointF(pun2.xr, pun2.yr);
    PointF point3 = new PointF(pun3.xr, pun3.yr);
    PointF point4 = new PointF(pun4.xr, pun4.yr);

    PointF[] curvePoints = { point1, point2, point3, point4 };

```

```

// Define fill mode.
FillMode newFillMode = FillMode.Winding; // FillMode.Alternate y FillMode.Winding

// Fill polygon to screen.
grafico.FillPolygon(blueBrush, curvePoints, newFillMode);

}

private void dibujarLinea()
{
    throw new NotImplementedException();
}

public void dibujarNubePuntos(List<punto3D> puntos)
{
    //dibujar cualquier nube de puntos 3D - inicio
    foreach (var p3D in puntos)
    {
        //punto2D p = new punto2D(p3D.X, p3D.Y, p3D.Z);
        //grafico.DrawEllipse(lapiz, p.xr, p.yr, 2, 2);
        dibujarPunto(p3D, 1);
    }
    //dibujar cualquier nube de puntos 3D - fin
}

public void dibujarLinea(double xi, double yi, double zi, double xf, double yf, double zf)
{
    //dibujar cualquier linea 3D - inicio
    punto3D punto1 = new punto3D(xi, yi, zi); //punto inicial
    punto3D punto2 = new punto3D(xf, yf, zf); //punto final
    punto2D puntoini = new punto2D(punto1.X, punto1.Y, punto1.Z); //punto inicial a graficar
    punto2D puntofin = new punto2D(punto2.X, punto2.Y, punto2.Z); //punto final a graficar
    grafico.DrawLine(lapiz, puntoini.xr, puntoini.yr, puntofin.xr, puntofin.yr); //dibujar la linea
    //dibujar cualquier linea 3D - fin

    //grafico.DrawString("P(x:" + xi + ",y:" + yi + ",z:" + zi + ")", letra, rrelleno, puntoini.xr, puntoini.yr);
    //grafico.DrawString("P(x:" + xf + ",y:" + yf + ",z:" + zf + ")", letra, rrelleno, puntofin.xr, puntofin.yr);
}

public void dibujarLinea(punto3D punto1, punto3D punto2)
{
    //dibujar cualquier linea 3D - inicio
    //punto3D punto1 = new punto3D(xi, yi, zi); //punto inicial
    //punto3D punto2 = new punto3D(xf, yf, zf); //punto final
    punto2D puntoini = new punto2D(punto1.X, punto1.Y, punto1.Z); //punto inicial a graficar
    punto2D puntofin = new punto2D(punto2.X, punto2.Y, punto2.Z); //punto final a graficar

```

```

grafico.DrawLine(lapiz, puntoini.xr, puntoini.yr, puntofin.xr, puntofin.yr); //dibujar la linea
//dibujar cualquier linea 3D - fin

//grafico.DrawString("P(x:" + xi + ",y:" + yi + ",z:" + zi + ")", letra, rrelleno, puntoini.xr, puntoini.yr);
//grafico.DrawString("P(x:" + xf + ",y:" + yf + ",z:" + zf + ")", letra, rrelleno, puntofin.xr, puntofin.yr);
}

public void dibujarPunto(double xi, double yi, double zi, float r)
{
    //dibujar cualquier punto 3D - inicio
    punto3D punto1 = new punto3D(xi, yi, zi); //punto inicial
    punto2D puntoini = new punto2D(punto1.X, punto1.Y, punto1.Z); //punto inicial a graficar
    grafico.DrawEllipse(lapiz, puntoini.xr, puntoini.yr, r, r);
    //dibujar cualquier punto 3D - fin

    //grafico.DrawString("P(x:" + xi + ",y:" + yi + ",z:" + zi + ")", letra, rrelleno, puntoini.xr, puntoini.yr);
}

public void dibujarPunto(punto3D punto1, float r)
{
    //dibujar cualquier punto 3D - inicio
    //punto3D punto1 = new punto3D(xi, yi, zi); //punto inicial
    punto2D puntoini = new punto2D(punto1.X, punto1.Y, punto1.Z); //punto inicial a graficar
    grafico.DrawEllipse(lapiz, puntoini.xr, puntoini.yr, r, r);
    //dibujar cualquier punto 3D - fin

    //grafico.DrawString("P(x:" + xi + ",y:" + yi + ",z:" + zi + ")", letra, rrelleno, puntoini.xr, puntoini.yr);
}
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Drawing.Imaging;

```

```

namespace graficador3D
{
    public partial class Form1 : Form
    {

```

```
public Form1()
{
    InitializeComponent();
}
unitario3D este = new unitario3D();
private void button1_Click(object sender, EventArgs e)
{
    este.dibujarEjes3D(this.pictureBox1);
}

private void button2_Click(object sender, EventArgs e)
{
    este.dibujarLinea3D(this.pictureBox1);
}

private void button3_Click(object sender, EventArgs e)
{
    este.dibujarEjes(this.pictureBox1);
    string directorio1 = Environment.GetFolderPath(Environment.SpecialFolder.Desktop) +
        "\\grafico_cartesiano.png";
    //Bitmap bmp = new Bitmap();
    var mm = (Bitmap)this.pictureBox1.Image.Clone();
    mm.Save(directorio1, ImageFormat.Png);
}
}
}
```
