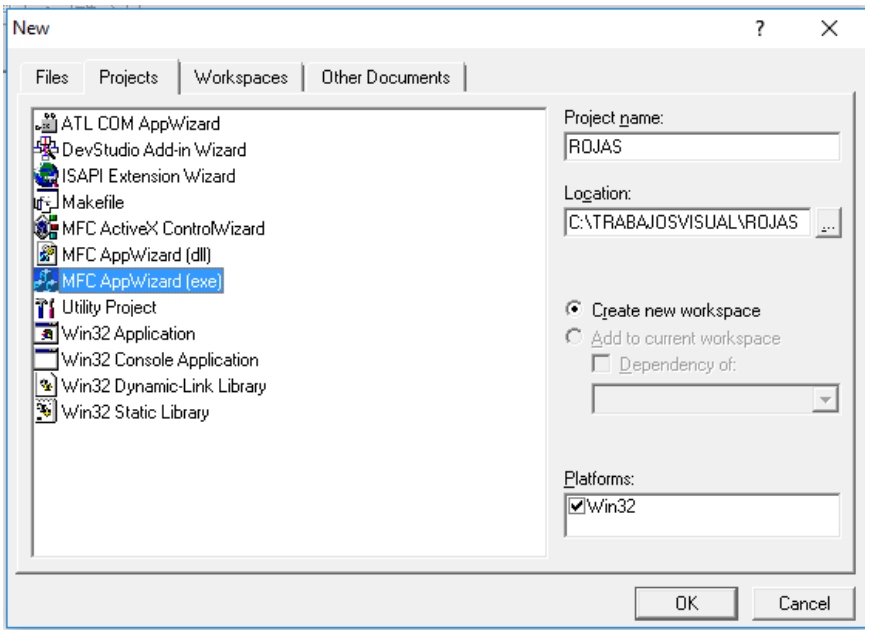
ULTIMA SESIÓN

<<File/New/MFC AppWizard(exe)/Proyect Name=ROJAS/

Location = C:\TRABAJOS VISUAL/Ok>>



<<Step1= Document Single/Next>>

<<Step2= None/Next>>

<<Step3= None/ActiveX Contorls/Next>>

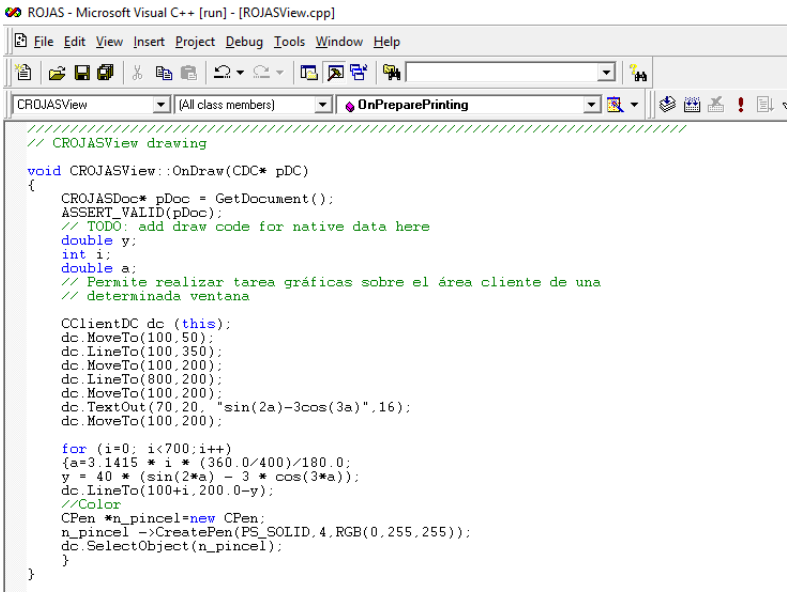
<<Step4= Docking Toolbar/Inicial StatusBar/Printing and Print Preview / 3D Controls / Normal/Next >>

<<Step5= MFC Standard / Yes pleace/As shared DLL / Next >>

<<Step6= Finish/Ok >>

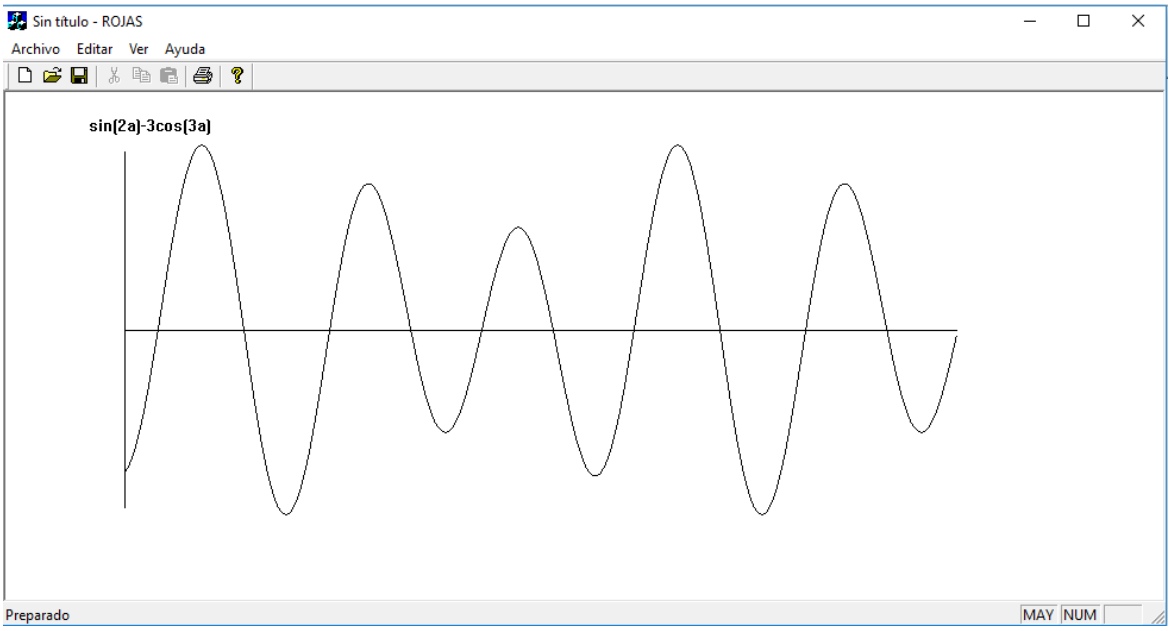
Despues nos ubicamos en la función OnDraw(CDC* pDC) de la Clase CAppVasquezView, para editar su código. No olvidar incluir la librería “math.h” para poder utilizar las funciones trigonométricas

Insertamos el siguiente código para la gráfica de: sen (2a) – 3 cos (3a)

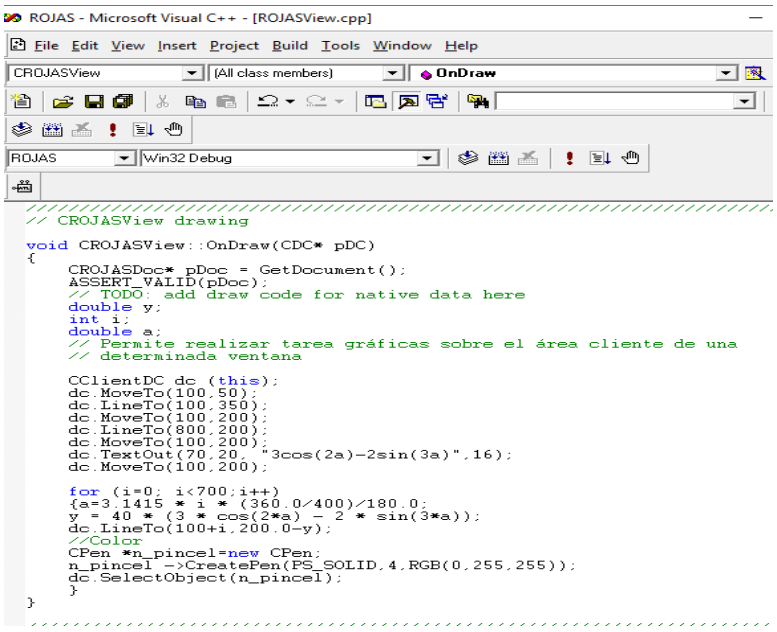



```
////////////////////////////////////////////////////////////////////////////
// CROJASView drawing

void CROJASView::OnDraw(CDC* pDC)
{
    CROJASDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    double y;
    int i;
    double a;
    // Permite realizar tarea gráficas sobre el área cliente de una
    // determinada ventana

    CClientDC dc (this);
    dc.MoveTo(100,50);
    dc.LineTo(100,350);
    dc.MoveTo(100,200);
    dc.LineTo(800,200);
    dc.MoveTo(100,200);
    dc.TextOut(70,20, "sin(2a)-3cos(3a)",16);
    dc.MoveTo(100,200);

    for (i=0; i<700;i++)
    {a=3.1415 * i * (360.0/400)/180.0;
    y = 40 * (sin(2*a) - 3 * cos(3*a));
    dc.LineTo(100+i,200.0-y);
    //Color
    CPen *n_pincel=new CPen;
    n_pincel ->CreatePen(PS_SOLID,4,RGB(0,255,255));
    dc.SelectObject(n_pincel);
    }
}
```

Al ejecutarlo nos mostrara lo siguiente:

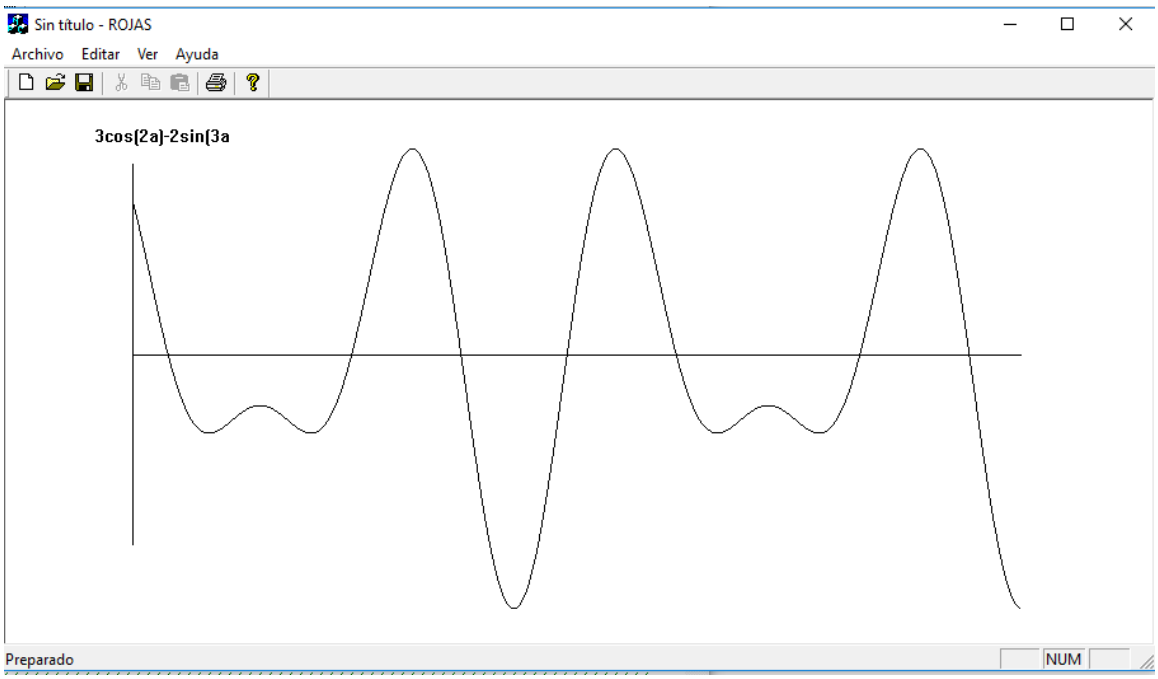


Insertamos el siguiente código para la gráfica de: 3cos(2a)-2sin(3a)




```
////////////////////////////////////////////////////////////////////////////
// CROJASView drawing

void CROJASView::OnDraw(CDC* pDC)
{
    CROJASDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    double y;
    int i;
    double a;
    // Permite realizar tarea gráficas sobre el área cliente de una
    // determinada ventana

    CClientDC dc (this);
    dc.MoveTo(100,50);
    dc.LineTo(100,350);
    dc.MoveTo(100,200);
    dc.LineTo(800,200);
    dc.MoveTo(100,200);
    dc.TextOut(70,20, "3cos(2a)-2sin(3a)",16);
    dc.MoveTo(100,200);

    for (i=0; i<700;i++)
    {a=3.1415 * i * (360.0/400)/180.0;
    y = 40 * (3 * cos(2*a) - 2 * sin(3*a));
    dc.LineTo(100+i,200.0-y);
    //Color
    CPen *n_pincel=new CPen;
    n_pincel ->CreatePen(PS_SOLID,4,RGB(0,255,255));
    dc.SelectObject(n_pincel);
    }
}
////////////////////////////////////////////////////////////////////////////
```

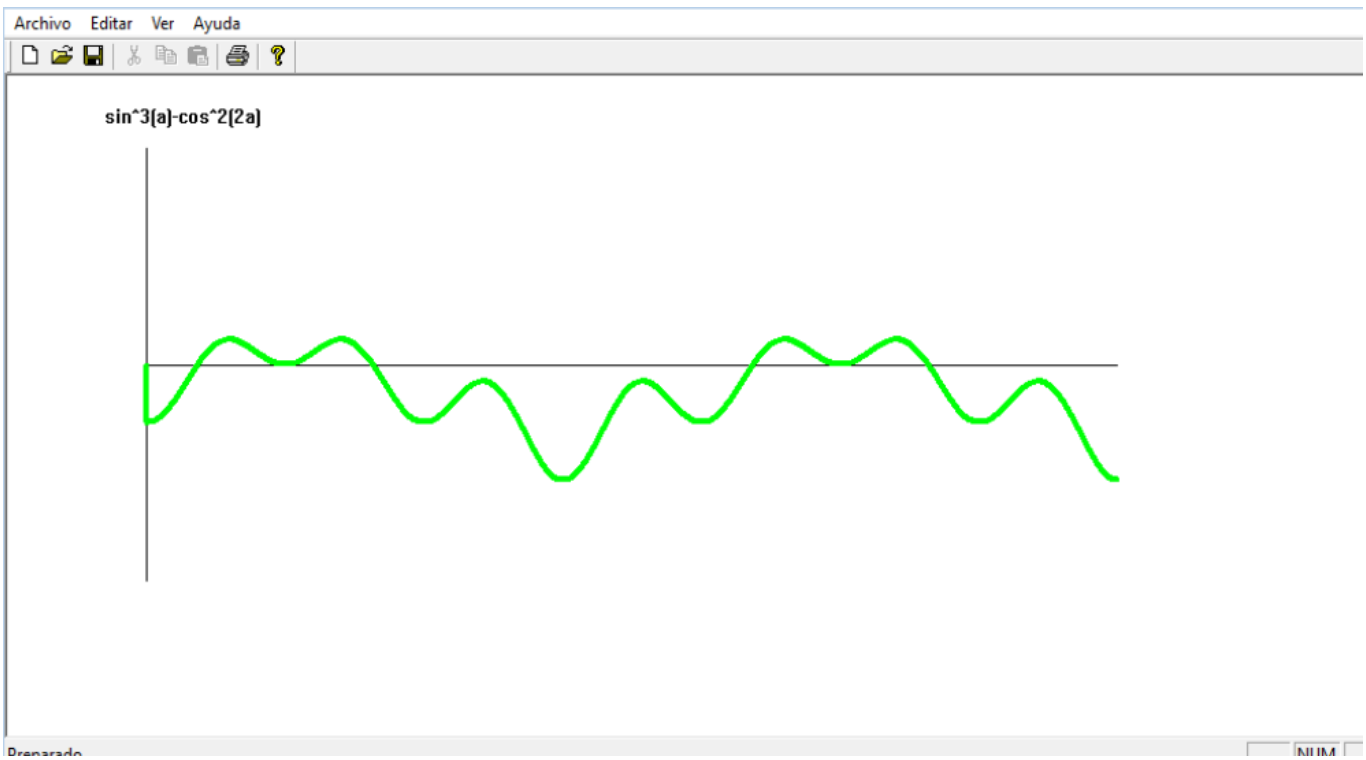
Al ejecutarlo nos mostrara lo siguiente:



Insertamos el siguiente código para la gráfica de: sen^3 (a) – cos^2 (2a)

```
    // TODO: add draw code for native data here
    double y;
    int i;
    double a, p, q;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana

    CClientDC dc (this);
    dc.MoveTo(100,50);
    dc.LineTo(100,350);
    dc.MoveTo(100,200);
    dc.LineTo(800,200);
    dc.MoveTo(100,200);
    dc.TextOut(70,20, "cot^3(a)-sin^2(2a)",18);
    dc.MoveTo(100,200);

    for (i=0; i<700;i++)
    {a=3.1415 * i * (360.0/400)/180.0;
      p = atan (a);
      q = sin (2*a);
    y = 40 * (pow(p,3) - pow(q,2));
    dc.LineTo(100+i,200.0-y);
     //Color
      CPen *n_pincel=new CPen;
      n_pincel ->CreatePen(PS_SOLID,4,RGB(0,255,64));
      dc.SelectObject(n_pincel);
    }
}
```

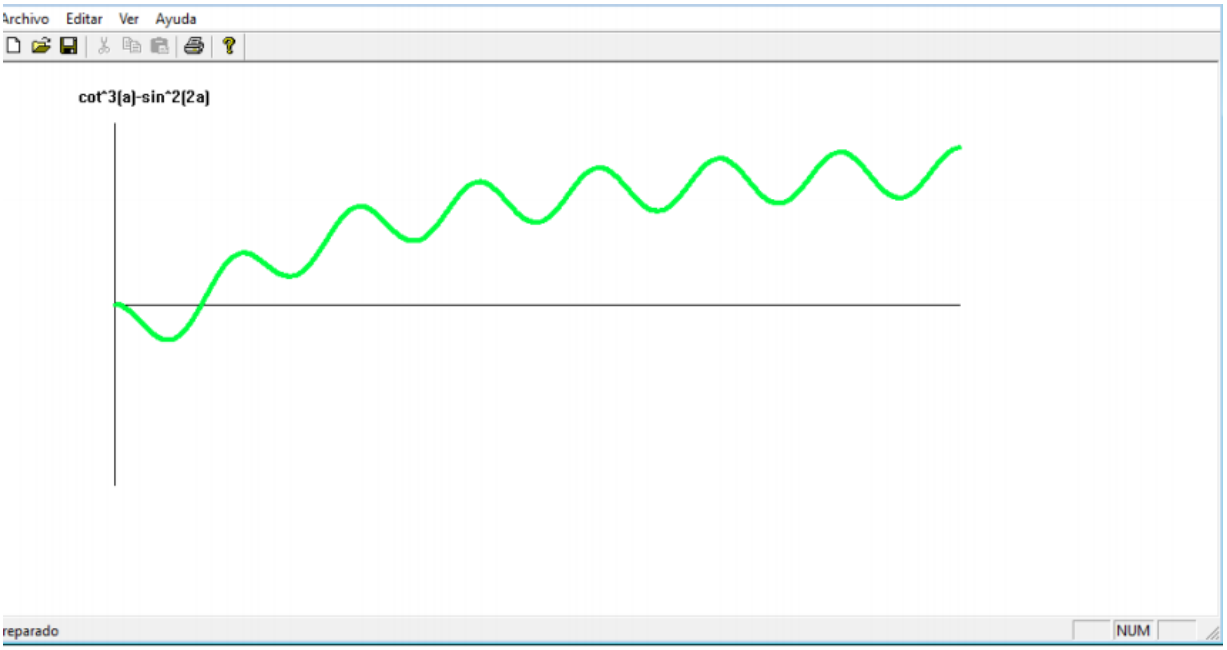
Al ejecutarlo nos mostrara lo siguiente:



Insertamos el siguiente código para la gráfica de: cot^3(a)-sin^2(2a)

```
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    double y;
    int i;
    double a, p, q;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana

    CClientDC dc (this);
    dc.MoveTo(100,50);
    dc.LineTo(100,350);
    dc.MoveTo(100,200);
    dc.LineTo(800,200);
    dc.MoveTo(100,200);
    dc.TextOut(70,20, "cot^3(a)-sin^2(2a)",18);
    dc.MoveTo(100,200);

    for (i=0; i<700;i++)
    {a=3.1415 * i * (360.0/400)/180.0;
      p = atan (a);
      q = sin (2*a);
    y = 40 * (pow(p,3) - pow(q,2));
    dc.LineTo(100+i,200.0-y);
     //Color
      CPen *n_pincel=new CPen;
      n_pincel ->CreatePen(PS_SOLID,4,RGB(0,255,64));
      dc.SelectObject(n_pincel);
    }
}
```

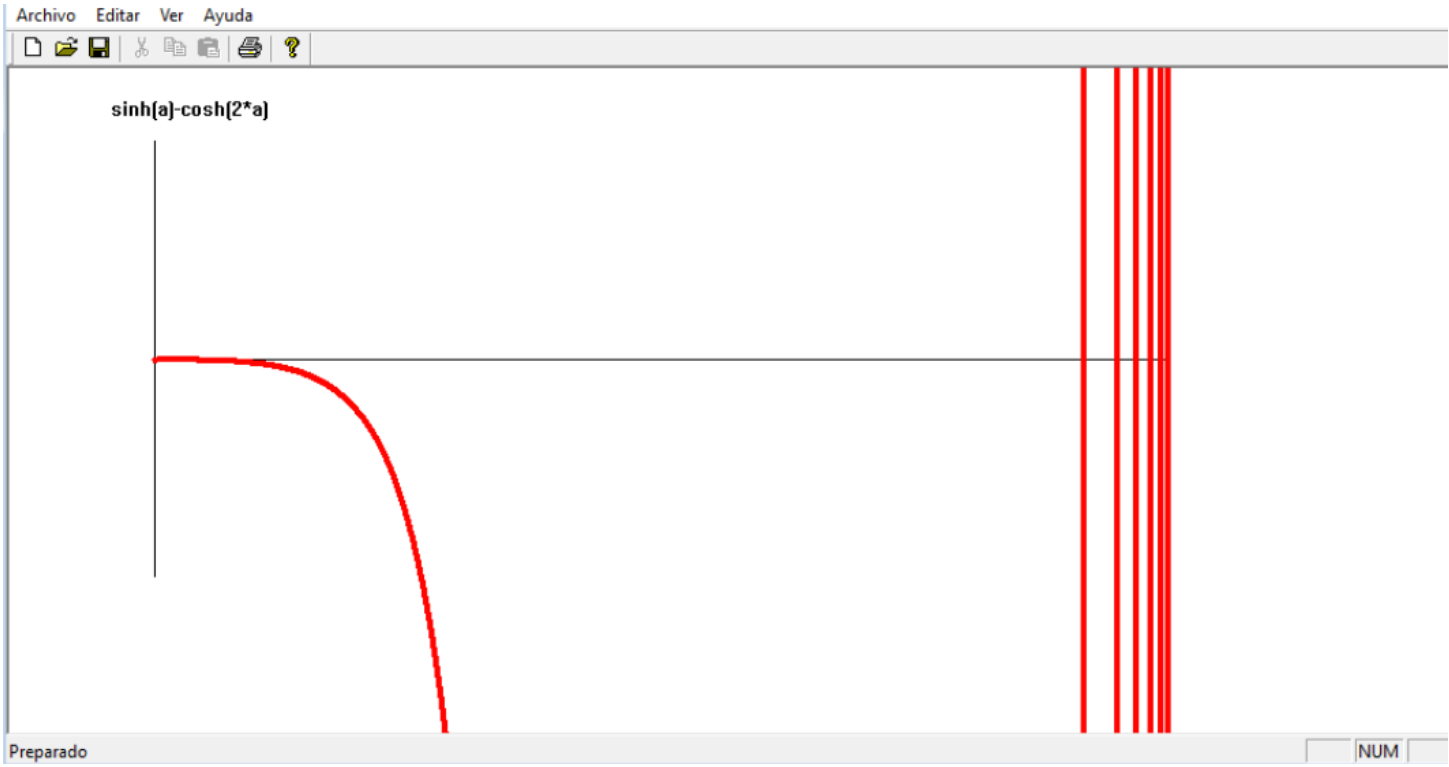
Una vez que ejecutamos nos mostrara lo siguiente:



Insertamos el siguiente código para la gráfica de: cot^3(a)-sin^2(2a)

```
        // TODO: add draw code for native data here
        double y;
        int i;
        double a;
        // Permite realizar tareas gráficas sobre el área cliente de una
        // determinada ventana
        CClientDC dc (this);
        dc.MoveTo(100,50);
        dc.LineTo(100,350);
        dc.MoveTo(100,200);
        dc.LineTo(800,200);
        dc.MoveTo(100,200);
        dc.TextOut(70,20, "sinh(a)-cosh(2*a)",20);
        dc.MoveTo(100,200);

        for (i=0; i<700;i++)
        { a=3.1415 * i * (360.0/400)/180.0;
          y =  (sinh(a) - cosh(2*a));
          //Color
          CPen *n_pincel=new CPen;
          n_pincel ->CreatePen(PS_SOLID,4,RGB(255,4,0));    //Inicializaci
          dc.SelectObject(n_pincel);
          dc.LineTo(100+i,200.0-y);
        }

}
```

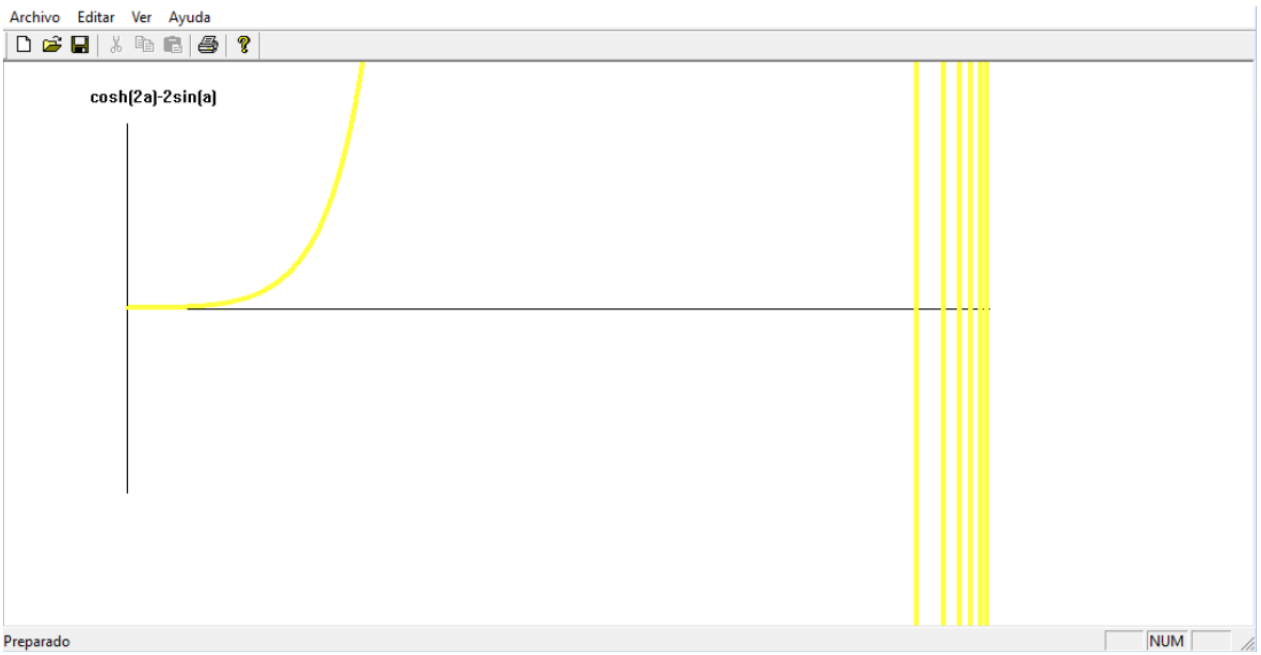
Al ejecutar nos mostrara lo siguiente:



Insertamos el siguiente código para la gráfica de: senh (a) – cosh (2a)

```
        // TODO: add draw code for native data here
        double y;
        int i;
        double a;
        // Permite realizar tareas gráficas sobre el área cliente de una
        // determinada ventana

        CClientDC dc (this);
        dc.MoveTo(100,50);
        dc.LineTo(100,350);
        dc.MoveTo(100,200);
        dc.LineTo(800,200);
        dc.MoveTo(100,200);
        dc.TextOut(70,20, "cosh(2a)-2sin(a)",18);
        dc.MoveTo(100,200);

        for (i=0; i<700;i++)
        {a=3.1415 * i * (360.0/400)/180.0;
        y = cosh (2*a) - 2 * sin (a);
        dc.LineTo(100+i,200.0-y);
         //Color
          CPen *n_pincel=new CPen;
          n_pincel ->CreatePen(PS_SOLID,4,RGB(255,255,64));
          dc.SelectObject(n_pincel);
        }
}
```

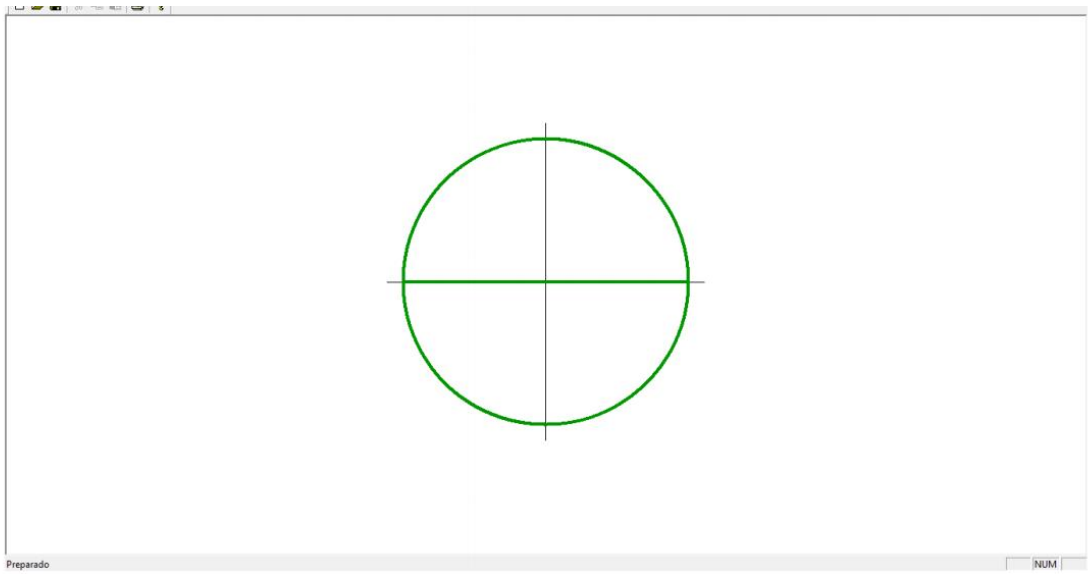
Luego ejecutamos y nos mostrara lo siguiente:



Insertamos el siguiente código para la gráfica del Círculo:

```
// TODO: add draw code for native data here
double y;
int i;
double a;
// Permite realizar tareas gráficas sobre el área cliente de una
// determinada ventana
CClientDC dc (this);
dc.MoveTo(680,135);
dc.LineTo(680,535);
dc.MoveTo(480,335);
dc.LineTo(880,335);
dc.TextOut(1360,670, "+",1);

for (i=-180; i<181;i++)
{
  y = sqrt(pow(180,2) - i*i);
  dc.LineTo(680+i,335-y);
 //Color
  CPen *n_pincel=new CPen;
  n_pincel ->CreatePen(PS_SOLID,4,RGB(0,150,0));
  dc.SelectObject(n_pincel);
}

for (i=-180; i<181;i++)
{
  y = sqrt(pow(180,2) - i*i);
  dc.LineTo(680+i,335+y);
}
dc.MoveTo(680,335);
```

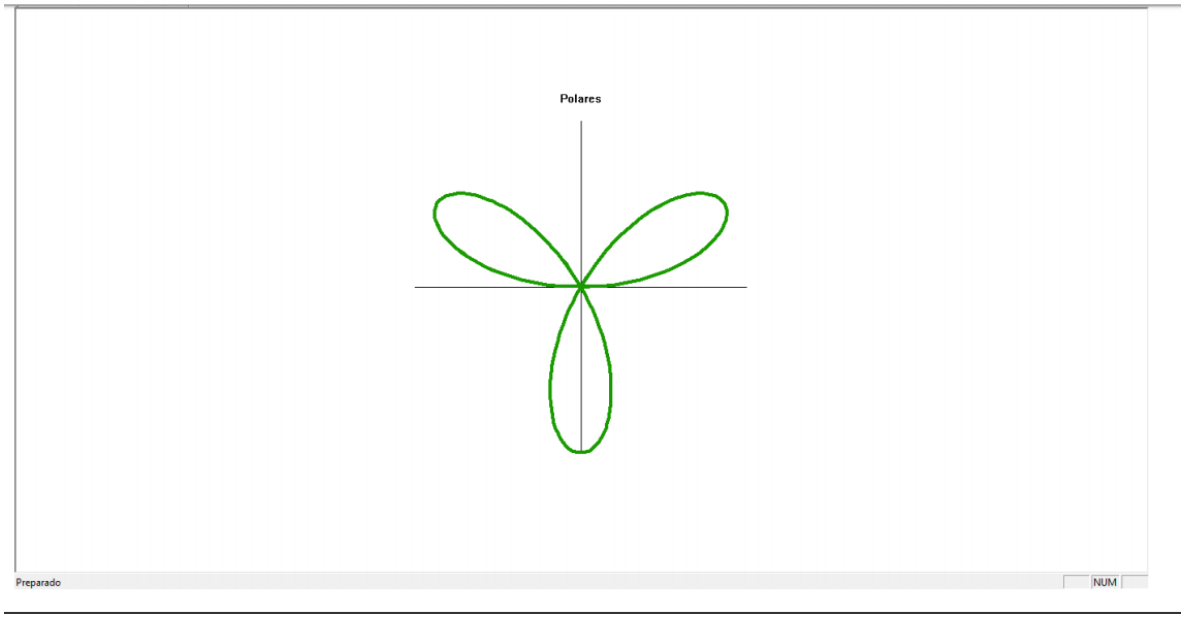
Al ejecutarlo nos mostrara lo siguiente:



Insertamos el siguiente código para la gráfica del Coordenadas polares:

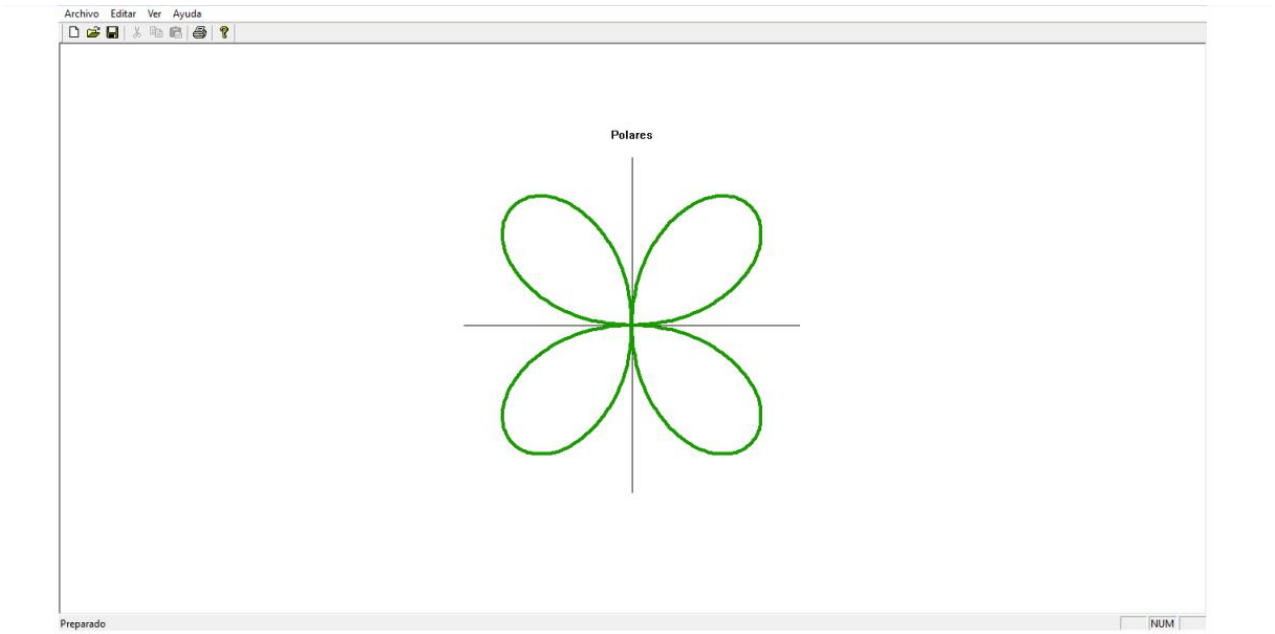
```
    // TODO: add draw code for native data here
    double y;
    int i;
    double a, b;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana
    CClientDC dc (this);
    dc.MoveTo(680,135);
    dc.LineTo(680,535);
    dc.MoveTo(480,335);
    dc.LineTo(880,335);
    dc.TextOut(1360,670, "Polares",7);
    dc.MoveTo(680,335);
    for (i=0; i<360;i++)
    {
      y = 150 * sin(3.1415 * 3 * i/180.0);
      a = sin (3.1415 * i/180);
      b = cos (3.1415 * i/180);
     //Color
      CPen *n_pincel=new CPen;
      n_pincel ->CreatePen(PS_SOLID,3,RGB(0,150,0));
      dc.SelectObject(n_pincel);
      dc.LineTo(680+y*b,335-y*a);
    }
}
```

Al ejecutarlo nos mostrara lo siguiente:



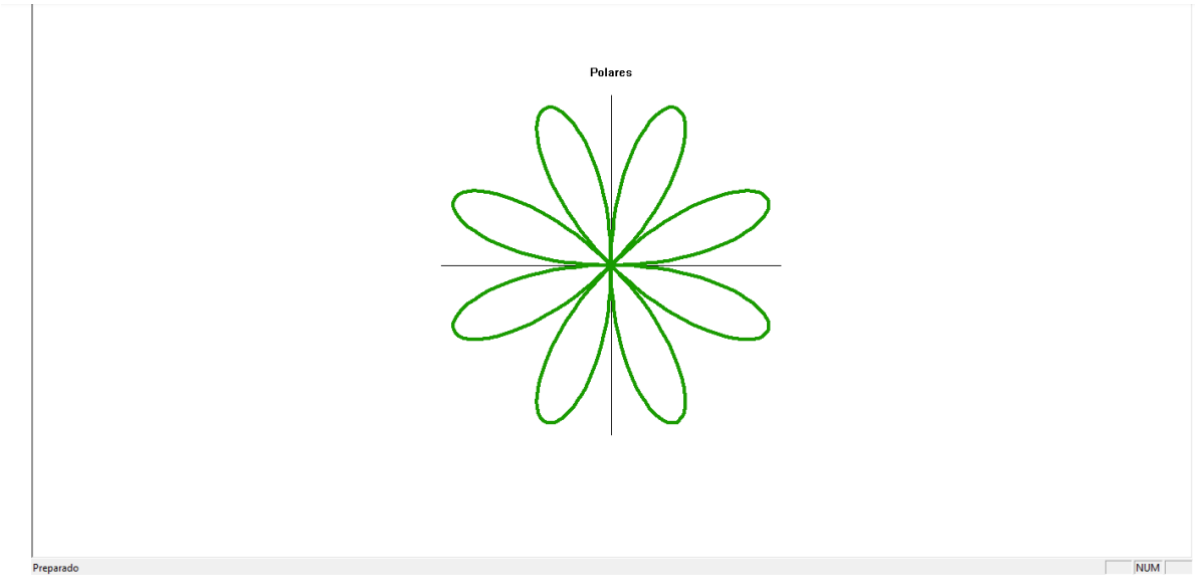
```
    // TODO: add draw code for native data here
    double y;
    int i;
    double a, b;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana

    CClientDC dc (this);
    dc.MoveTo(680,135);
    dc.LineTo(680,535);
    dc.MoveTo(480,335);
    dc.LineTo(880,335);
    dc.TextOut(670,120, "Polares",7);
    dc.MoveTo(680,335);

    for (i=0; i<360;i++)
    {
     y = 200 * sin(3.1415 * 2 * i/180.0);
     a = sin (3.1415 * i/180);
     b = cos (3.1415 * i/180);
    dc.LineTo(680+y*b,335-y*a);
     //Color
      CPen *n_pincel=new CPen;
      n_pincel ->CreatePen(PS_SOLID,4,RGB(25,155,0));
      dc.SelectObject(n_pincel);
    }
}
```

Al ejecutarlo nos mostrara lo siguiente:



```
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    double y;
    int i;
    double a, b;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana

    CClientDC dc (this);
    dc.MoveTo(680,135);
    dc.LineTo(680,535);
    dc.MoveTo(480,335);
    dc.LineTo(880,335);
    dc.TextOut(655,100, "Polares",7);
    dc.MoveTo(680,335);

    for (i=0; i<360;i++)
    {
     y = 200 * sin(3.1415 * 4 * i/180.0);
     a = sin (3.1415 * i/180);
     b = cos (3.1415 * i/180);
    dc.LineTo(680+y*b,335-y*a);
     //Color
      CPen *n_pincel=new CPen;
      n_pincel ->CreatePen(PS_SOLID,4,RGB(25,155,0));
      dc.SelectObject(n_pincel);
    }
}
```

Al ejecutarlo nos mostrara lo siguiente:



```
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    double y;
    int i;
    double a, b;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana

    CClientDC dc (this);
    dc.MoveTo(680,135);
    dc.LineTo(680,535);
    dc.MoveTo(480,335);
    dc.LineTo(880,335);
    dc.TextOut(655,100, "Polares",7);
    dc.MoveTo(680,335);

    for (i=0; i<360;i++)
    {
     y = 200 * sin(3.1415 * 6 * i/180.0);
     a = sin (3.1415 * i/180);
     b = cos (3.1415 * i/180);
    dc.LineTo(680+y*b,335-y*a);
     //Color
      CPen *n_pincel=new CPen;
      n_pincel ->CreatePen(PS_SOLID,4,RGB(25,155,0));
      dc.SelectObject(n_pincel);
    }
}
```

Al ejecutarlo nos muestra lo siguiente: