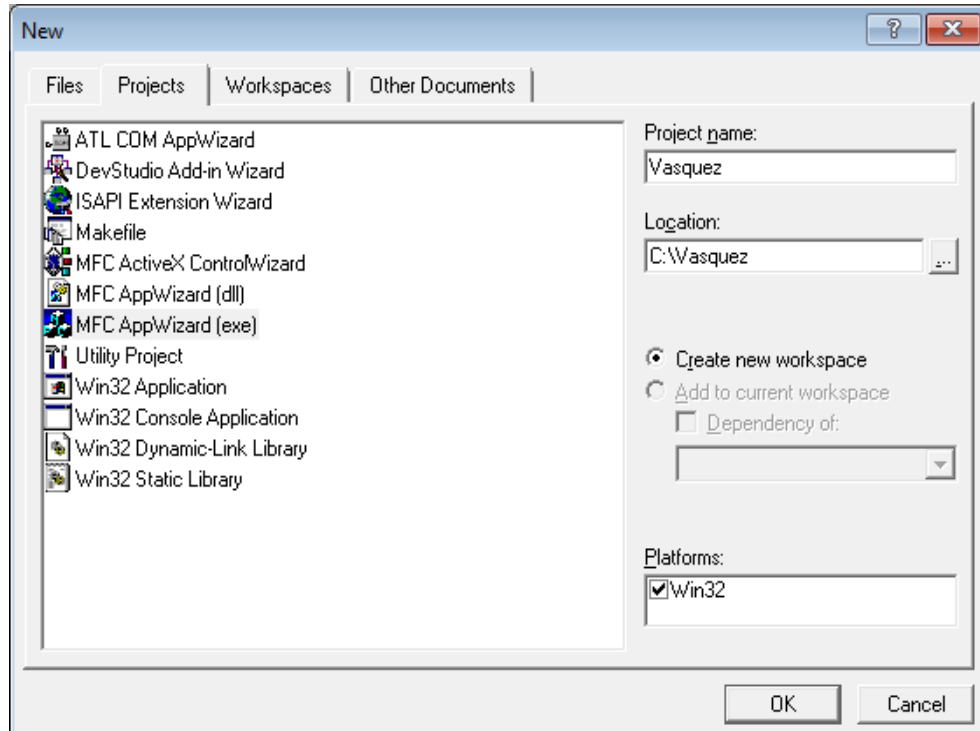


TAREA DE LA SESION 10

PRIMERO:

<<File/New/MFC AppWizard(exe)/Project Name=Vasquez/

Location = C:\Vasquez/Ok>>



<<Step1= Document Single/Next>>

<<Step2= None/Next>>

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

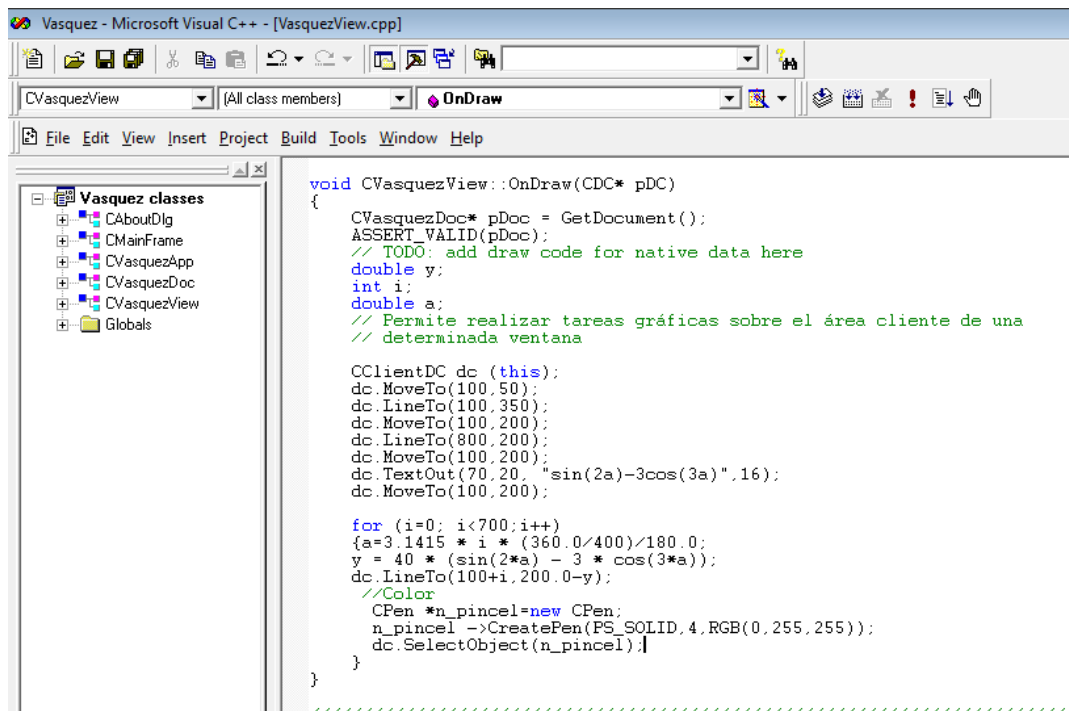
<<Step4= Docking Toolbar/Initial StatusBar/Printing and Print Preview / 3D Controls / Normal/Next >>

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

<<Step6= Finish/Ok >>

SEGUIDAMENTE: 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: $\sin(2a) - 3\cos(3a)$

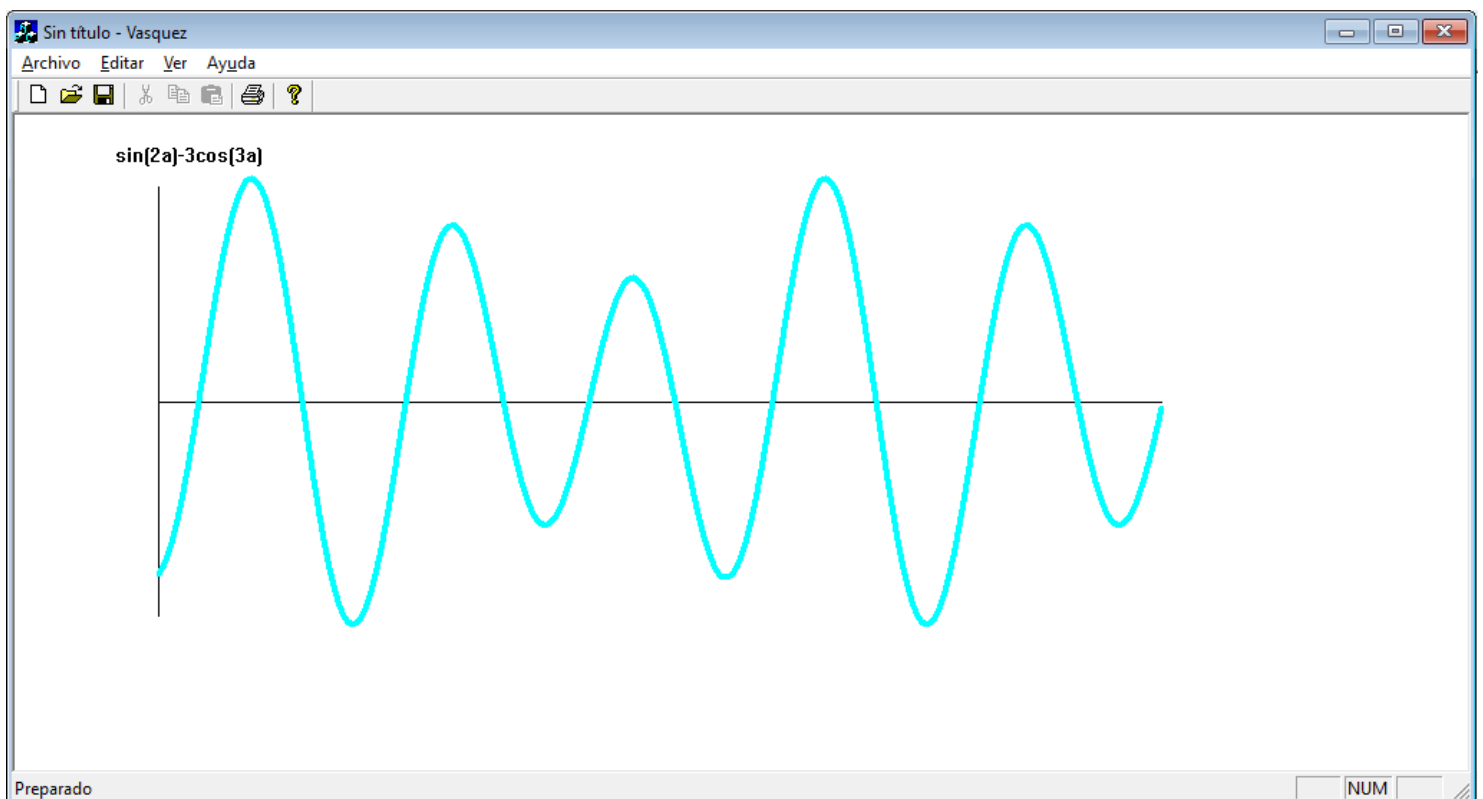


```
void CVasquezView::OnDraw(CDC* pDC)
{
    CVasquezDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    double y;
    int i;
    double a;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana

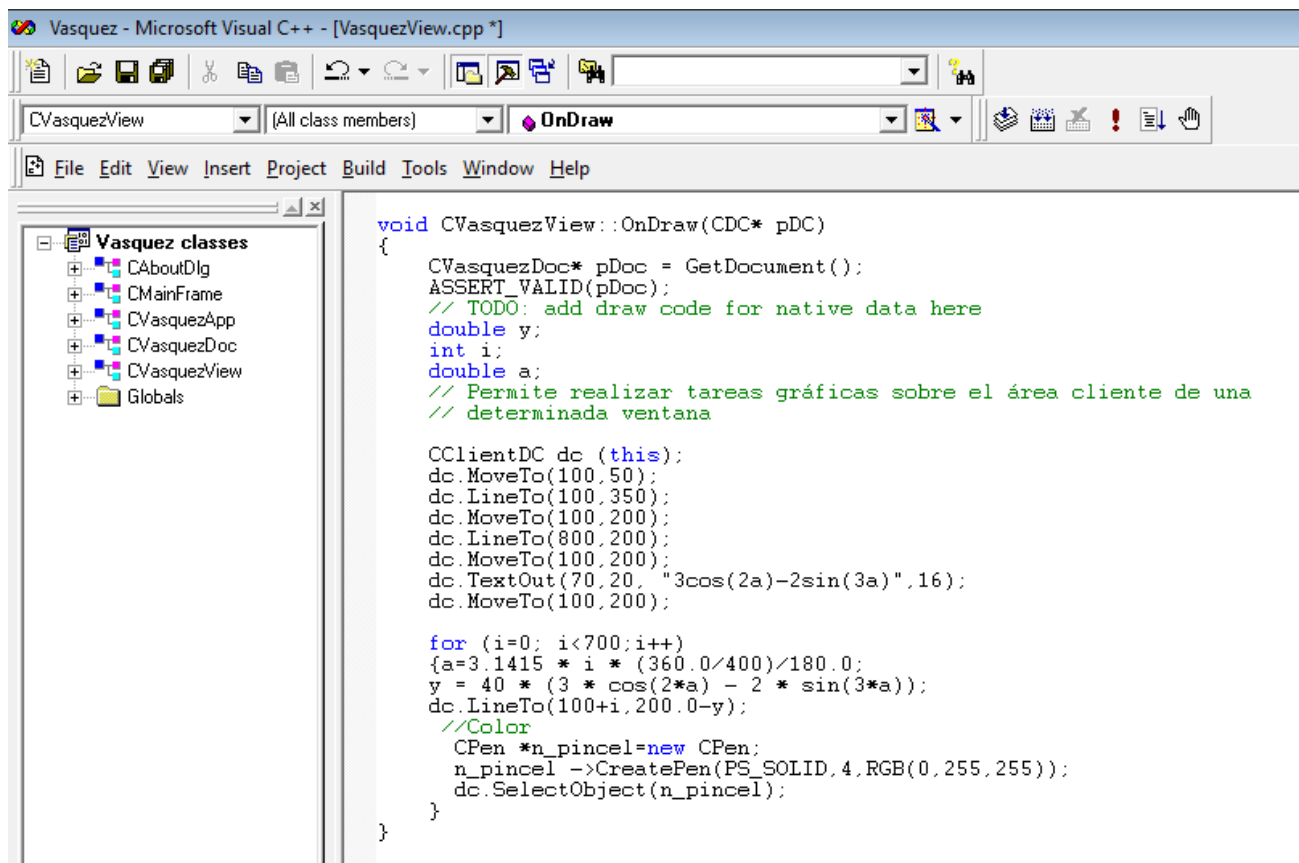
    CClientDC dc(this);
    dc.MoveTo(100, 50);
    dc.LineTo(100, 350);
    dc.MoveTo(100, 200);
    dc.LineTo(800, 200);
    dc.MoveTo(100, 200);
    dc.TextOut(70, 20, "sin(2a)-3cos(3a)", 16);
    dc.MoveTo(100, 200);

    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: $3\cos(2a)-2\sin(3a)$

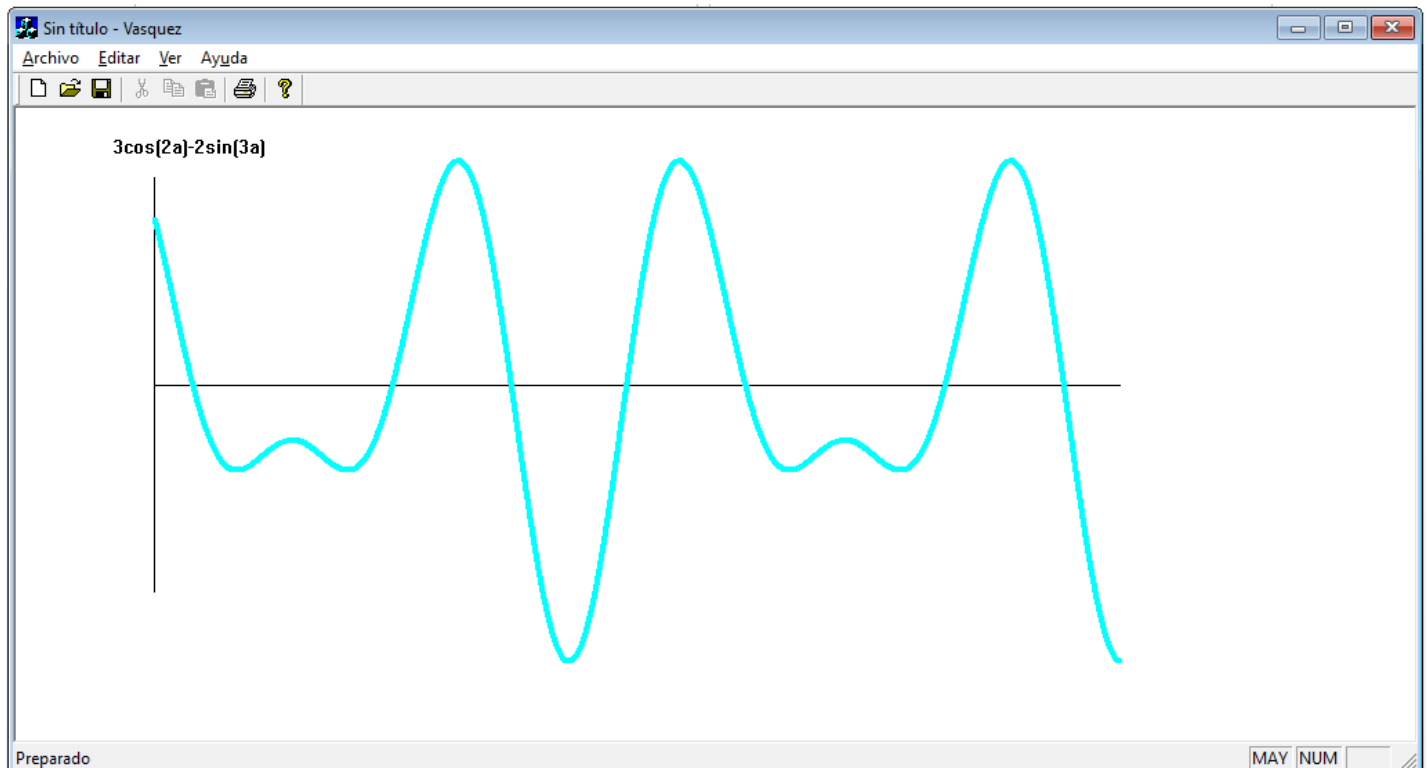


```
void CVasquezView::OnDraw(CDC* pDC)
{
    CVasquezDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    double y;
    int i;
    double a;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana

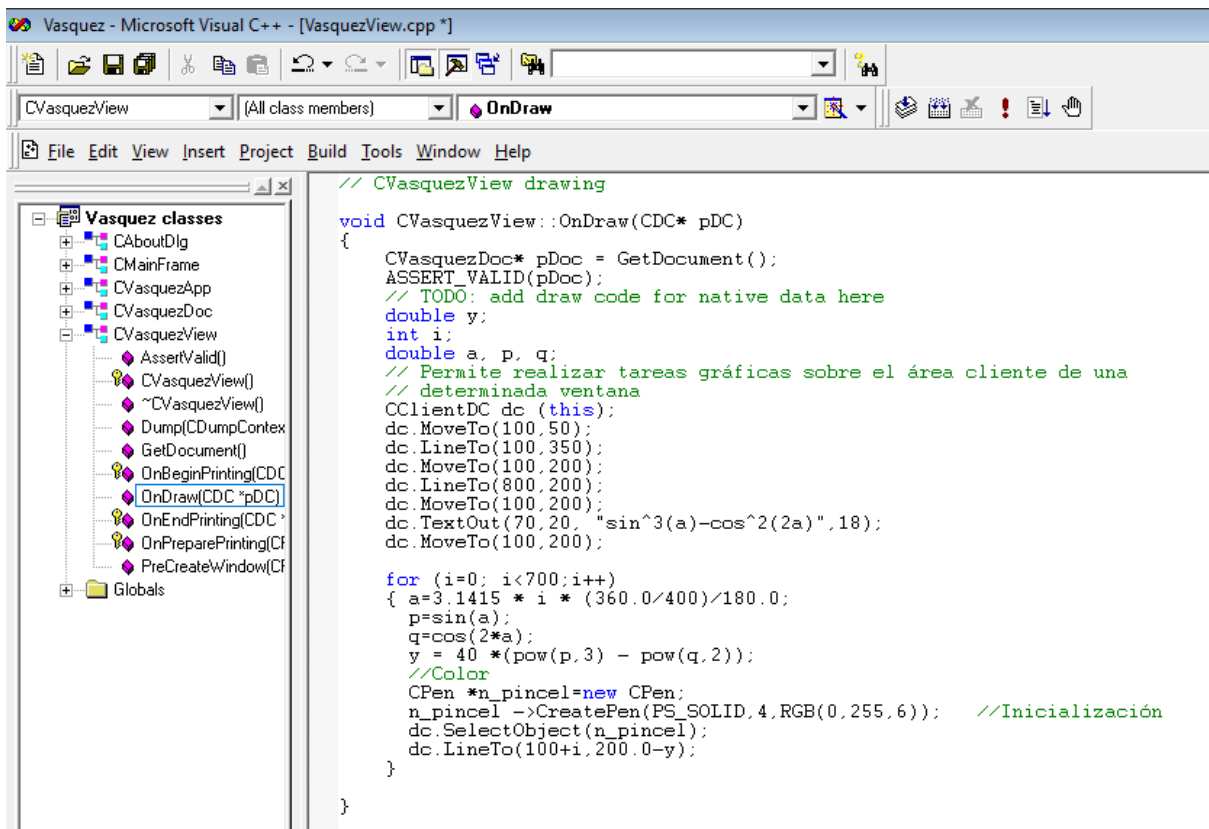
    CClientDC dc (this);
    dc.MoveTo(100,50);
    dc.LineTo(100,350);
    dc.MoveTo(100,200);
    dc.LineTo(800,200);
    dc.MoveTo(100,200);
    dc.TextOut(70,20, "3cos(2a)-2sin(3a)",16);
    dc.MoveTo(100,200);

    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);
    }
}
```

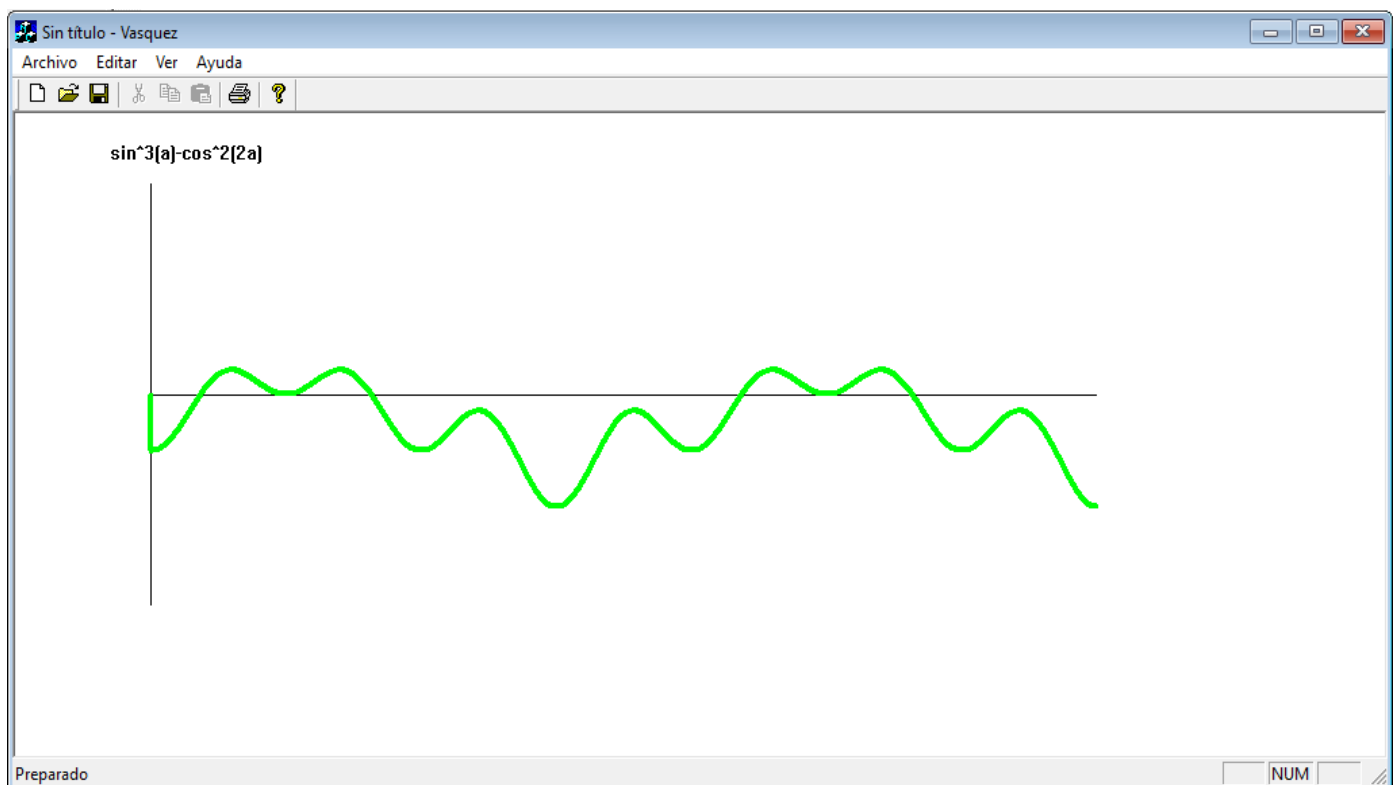
Al ejecutarlo nos mostrara lo siguiente:



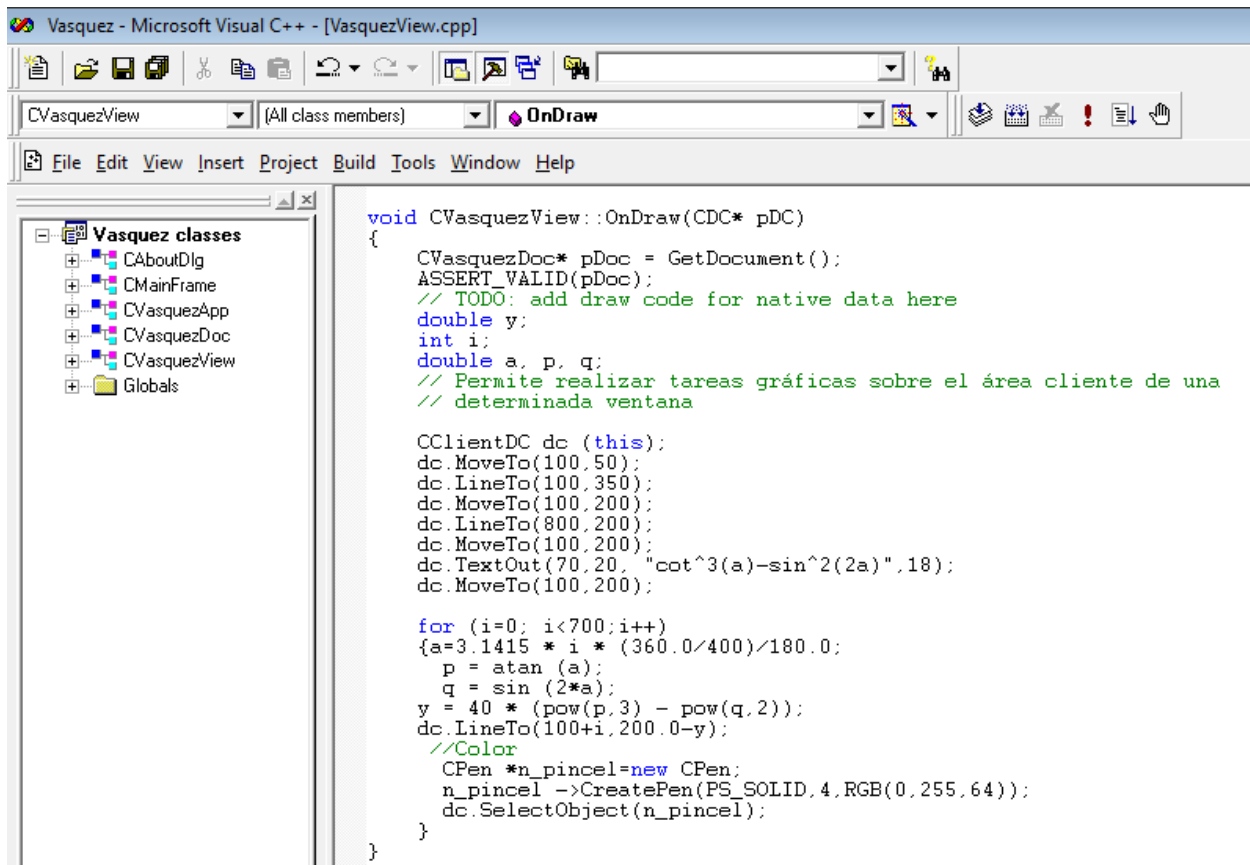
Insertamos el siguiente código para la gráfica de: $\sin^3(a) - \cos^2(2a)$



Al ejecutarlo nos mostrara lo siguiente:



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

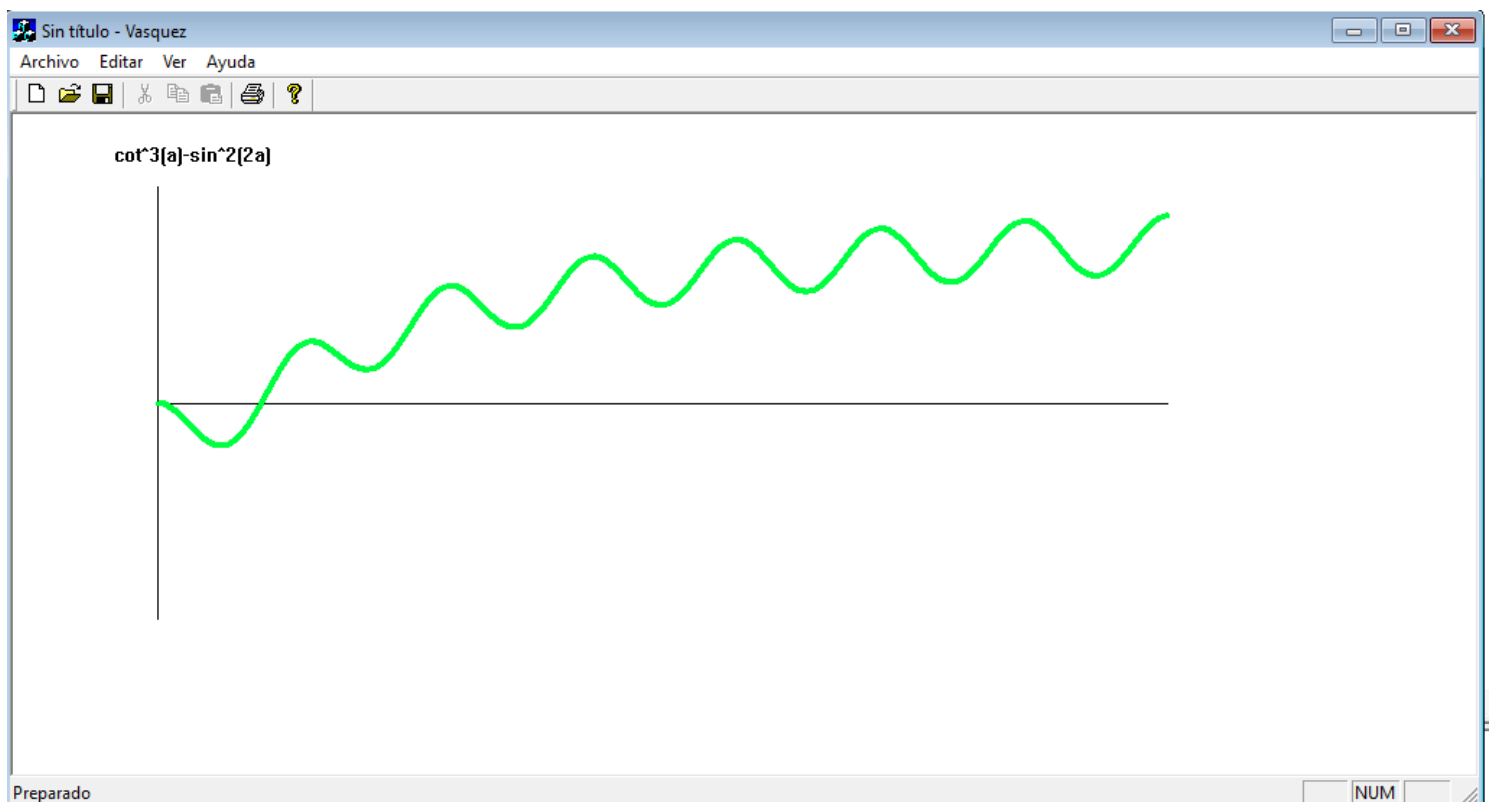


```
void CVasquezView::OnDraw(CDC* pDC)
{
    CVasquezDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    double y;
    int i;
    double a, p, q;
    // Permite realizar tareas gráficas sobre el área cliente de una
    // determinada ventana

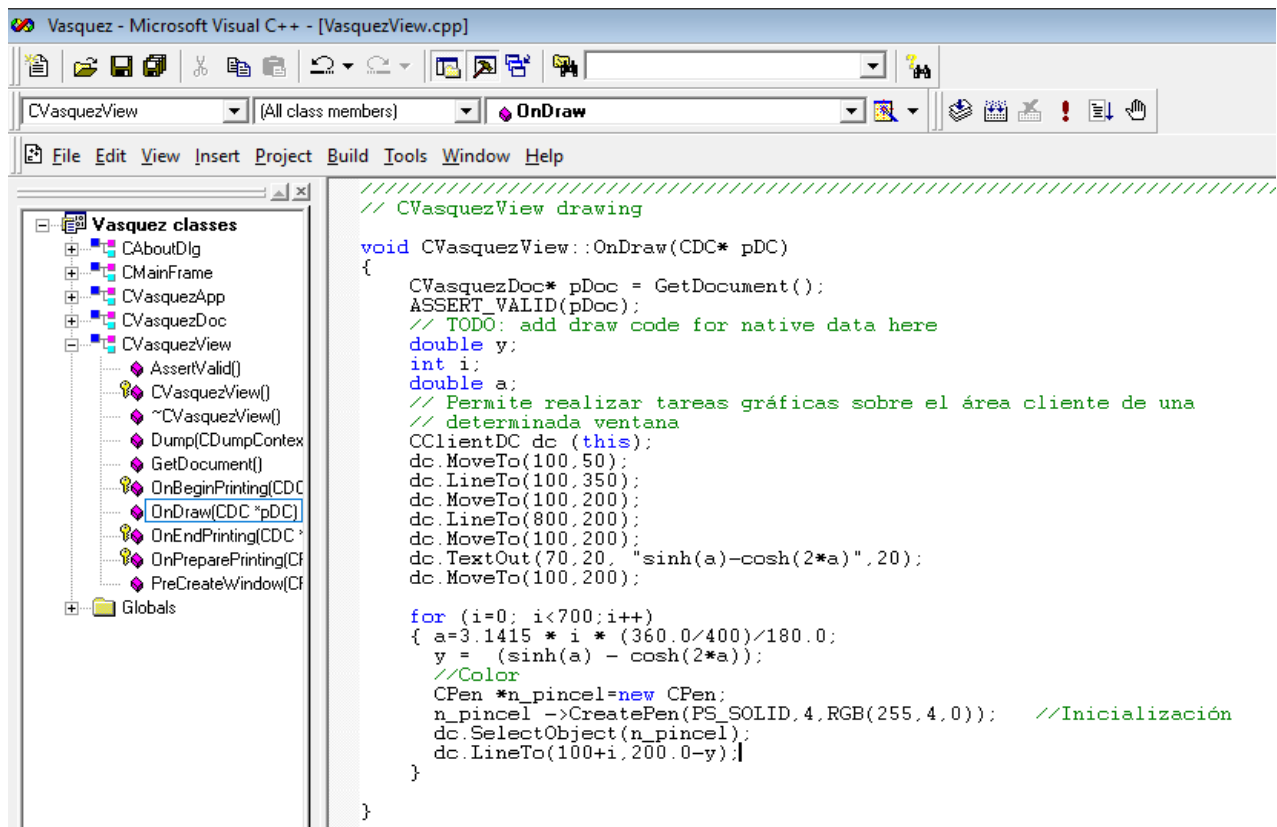
    CClientDC dc (this);
    dc.MoveTo(100,50);
    dc.LineTo(100,350);
    dc.MoveTo(100,200);
    dc.LineTo(800,200);
    dc.MoveTo(100,200);
    dc.TextOut(70,20, "cot^3(a)-sin^2(2a)",18);
    dc.MoveTo(100,200);

    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);
    }
}
```

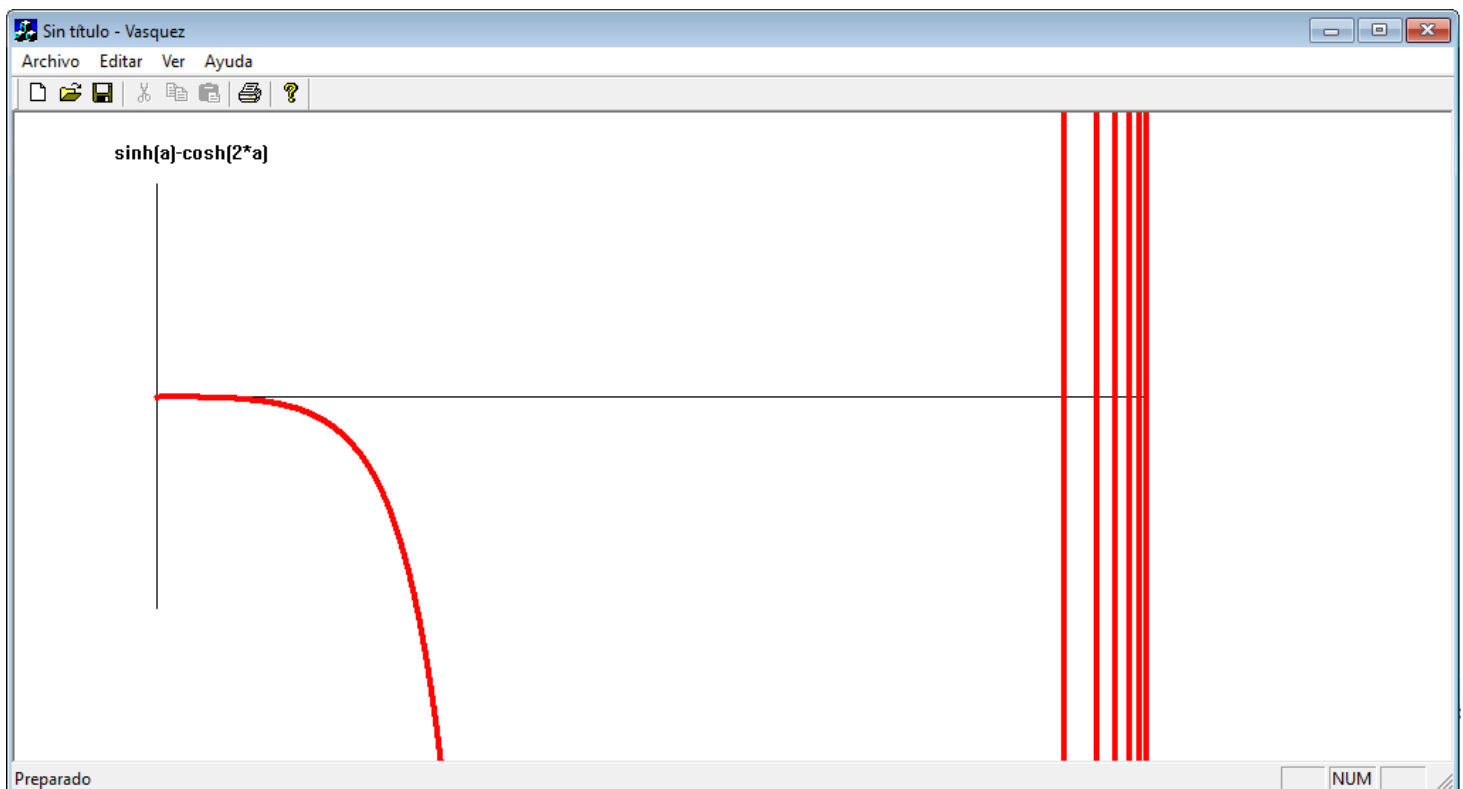
Al ejecutarlo nos mostrara lo siguiente:



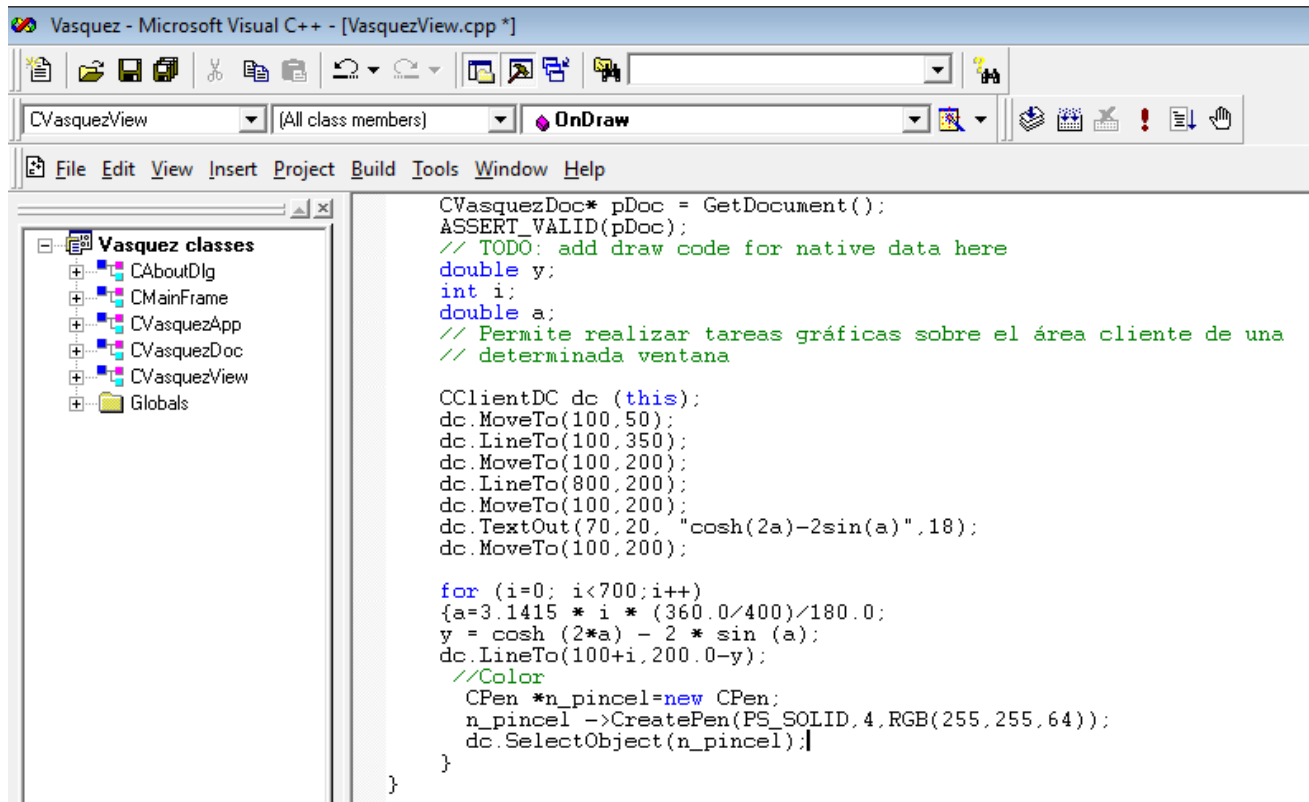
Insertamos el siguiente código para la gráfica de: $\sinh(a) - \cosh(2a)$



Al ejecutarlo nos mostrara lo siguiente:



Insertamos el siguiente código para la gráfica de: $\cosh(2a)-2\sin(a)$

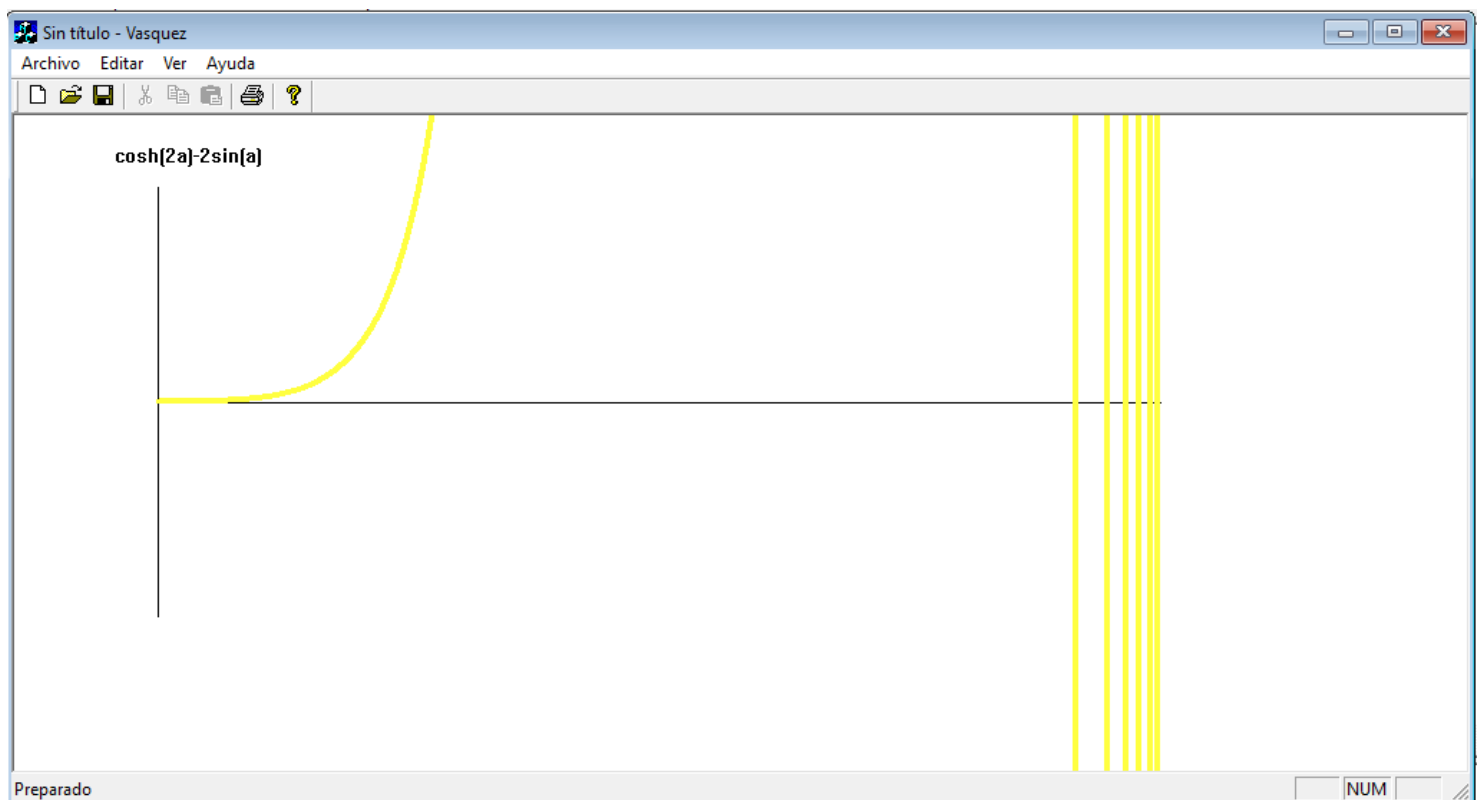


```
CVasquezDoc* pDoc = GetDocument();
ASSERT_VALID(pDoc);
// TODO: add draw code for native data here
double y;
int i;
double a;
// Permite realizar tareas gráficas sobre el área cliente de una
// determinada ventana

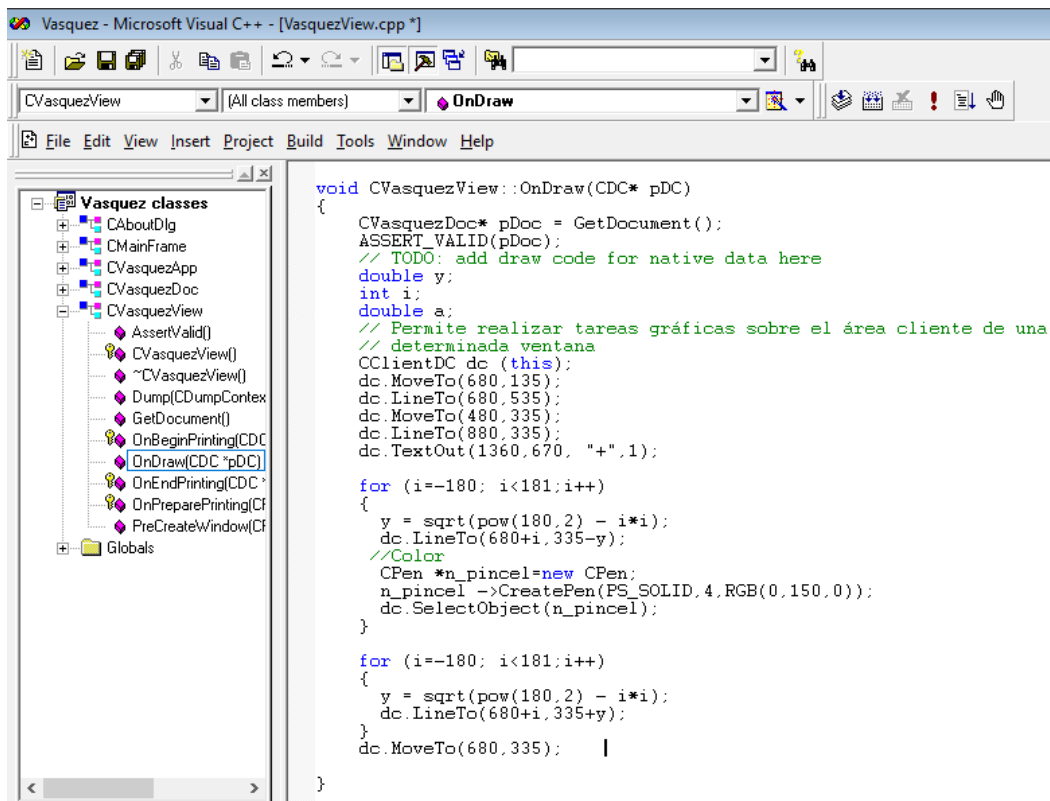
CClientDC dc (this);
dc.MoveTo(100,50);
dc.LineTo(100,350);
dc.MoveTo(100,200);
dc.LineTo(800,200);
dc.MoveTo(100,200);
dc.TextOut(70,20, "cosh(2a)-2sin(a)",18);
dc.MoveTo(100,200);

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);}
}
```

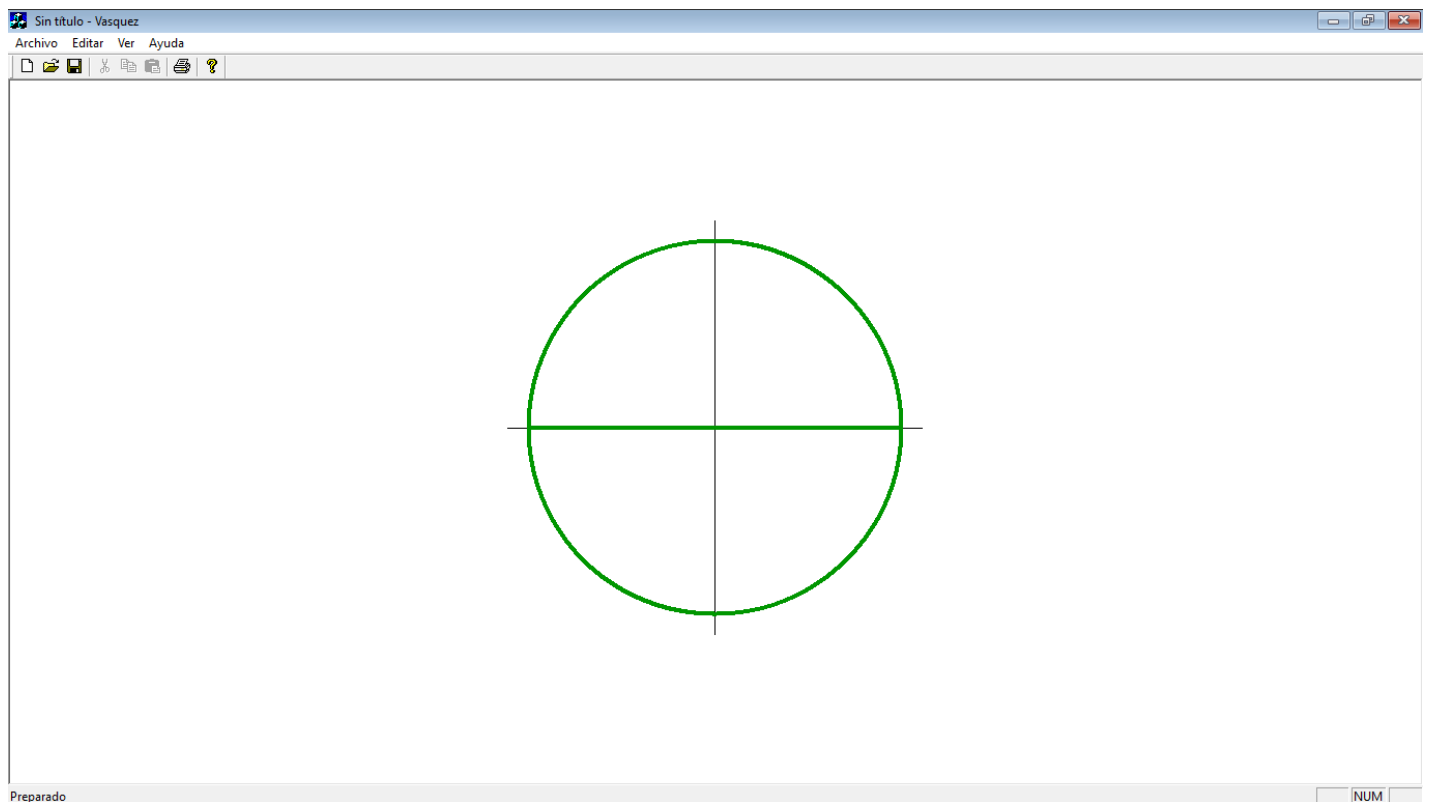
Al ejecutarlo nos mostrara lo siguiente:



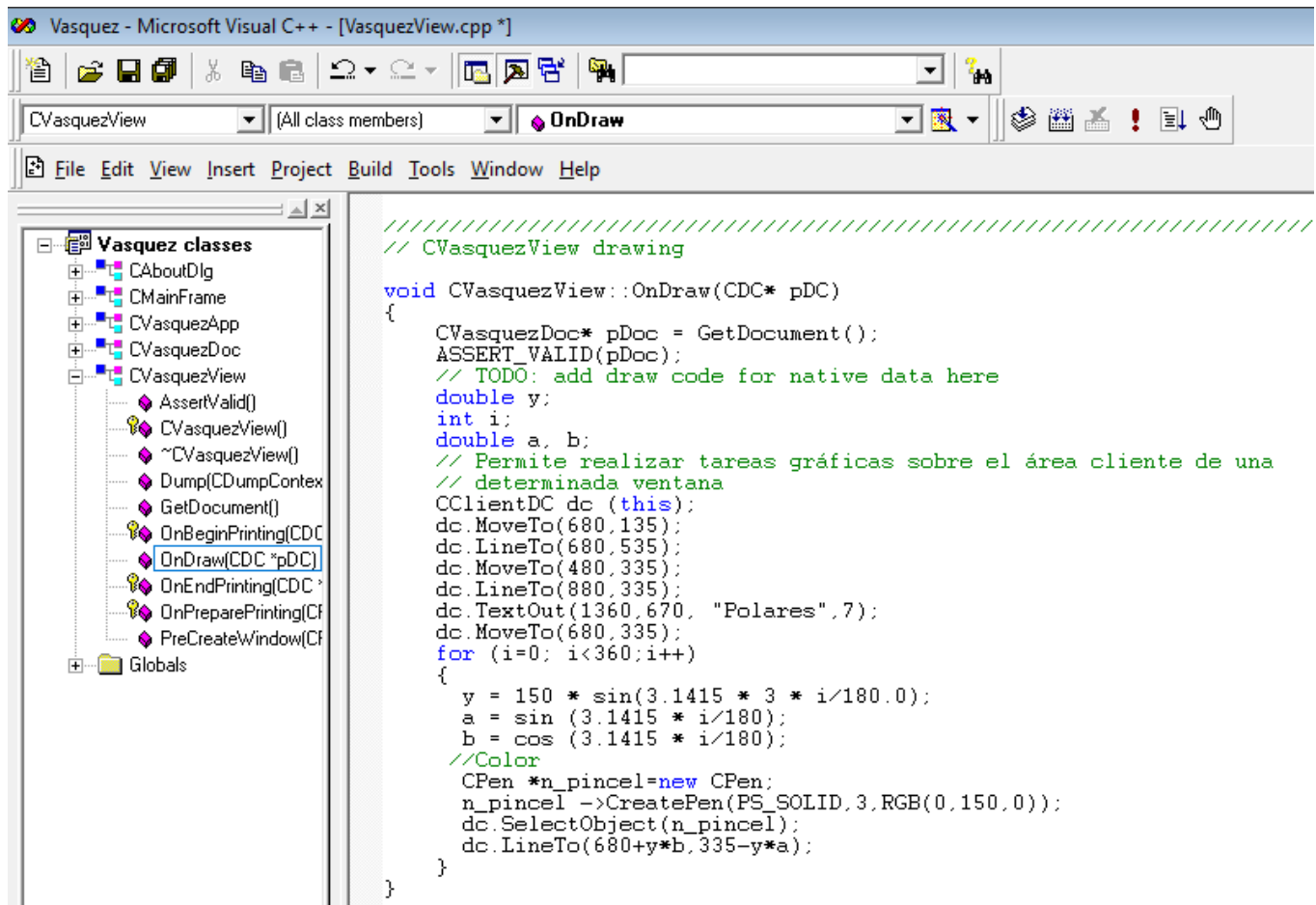
Insertamos el siguiente código para la gráfica del Círculo:



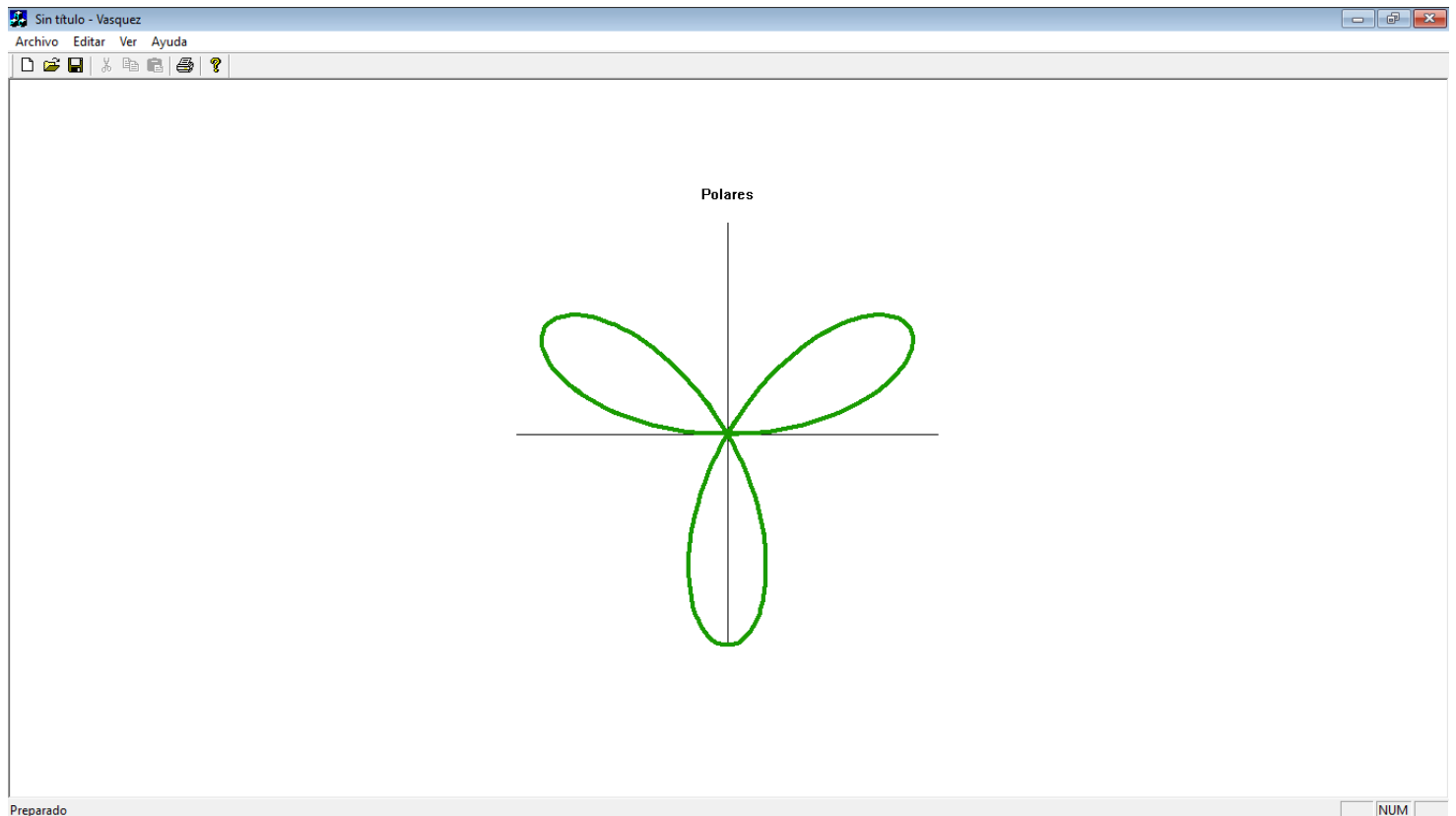
Al ejecutarlo nos mostrara lo siguiente:

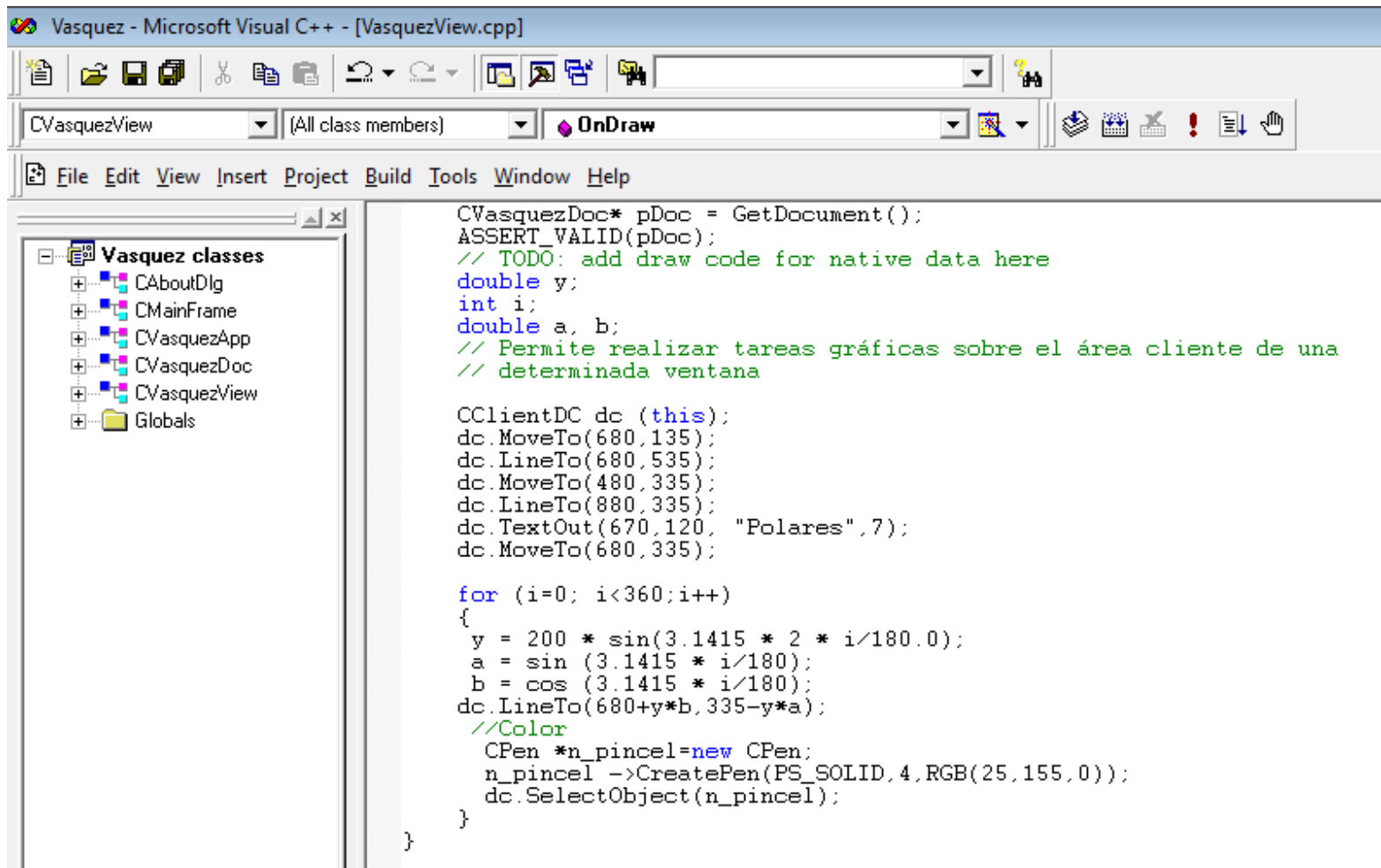


Insertamos el siguiente código para la gráfica del Coordenadas polares:

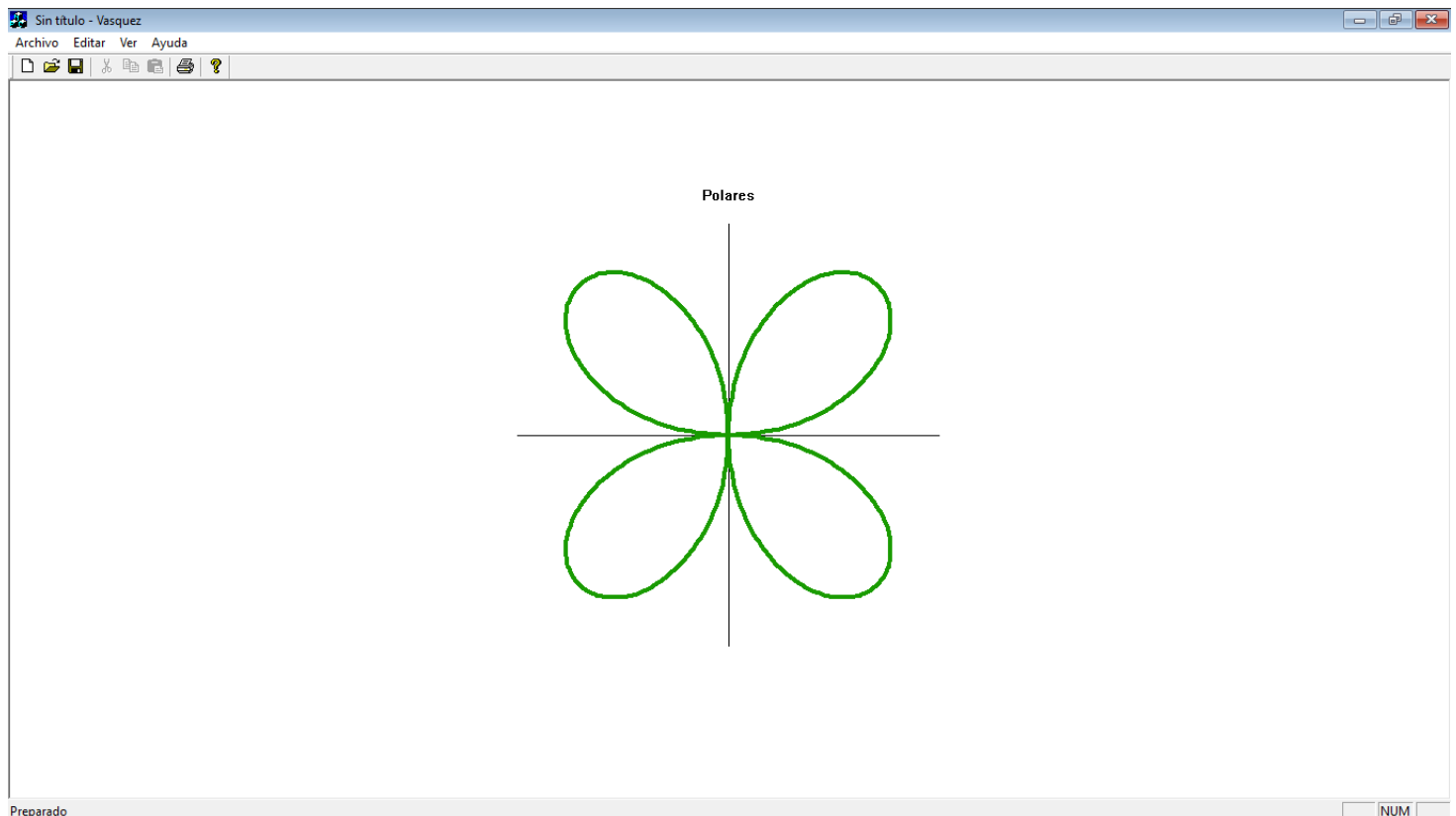


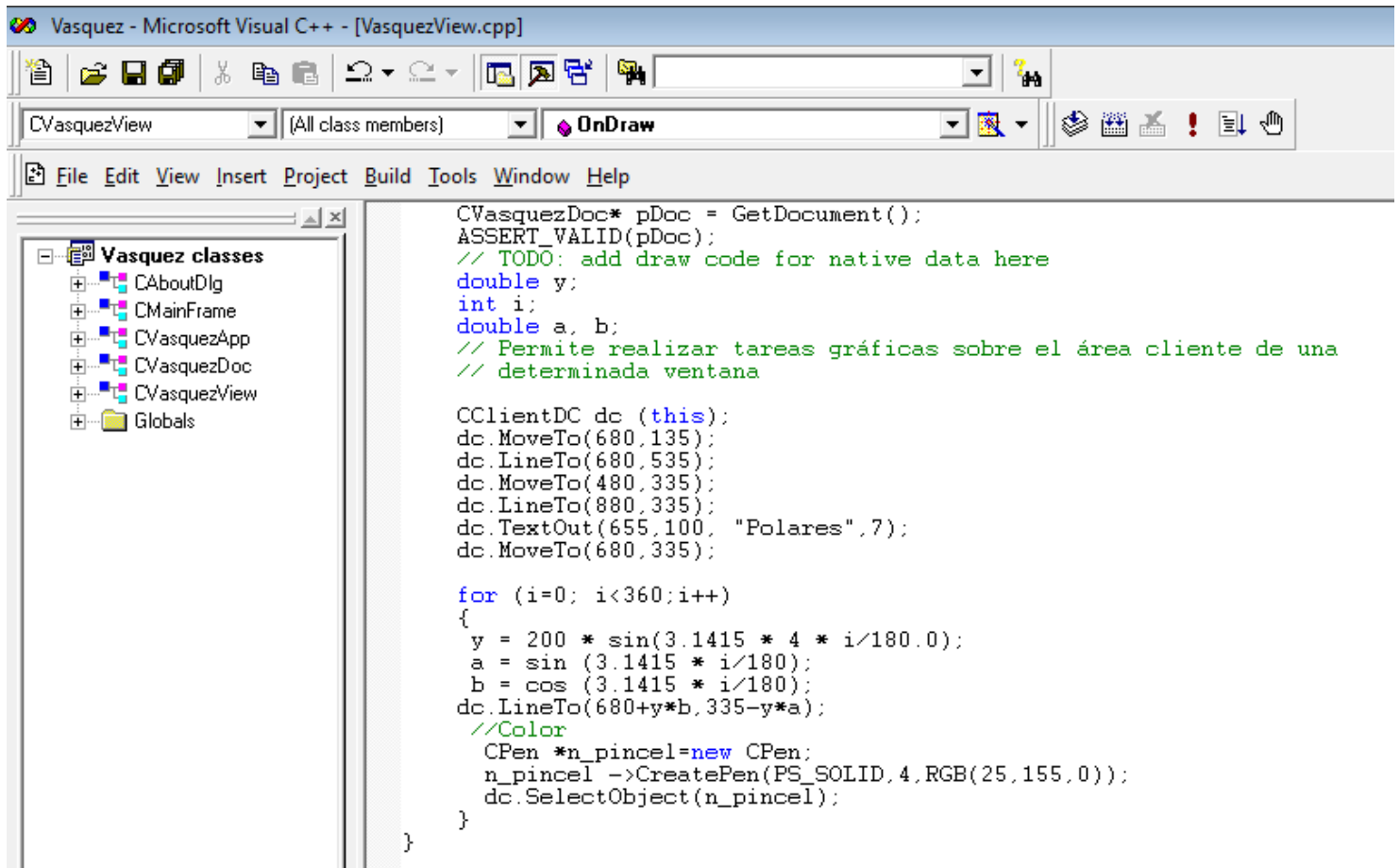
Al ejecutarlo nos mostrara lo siguiente:



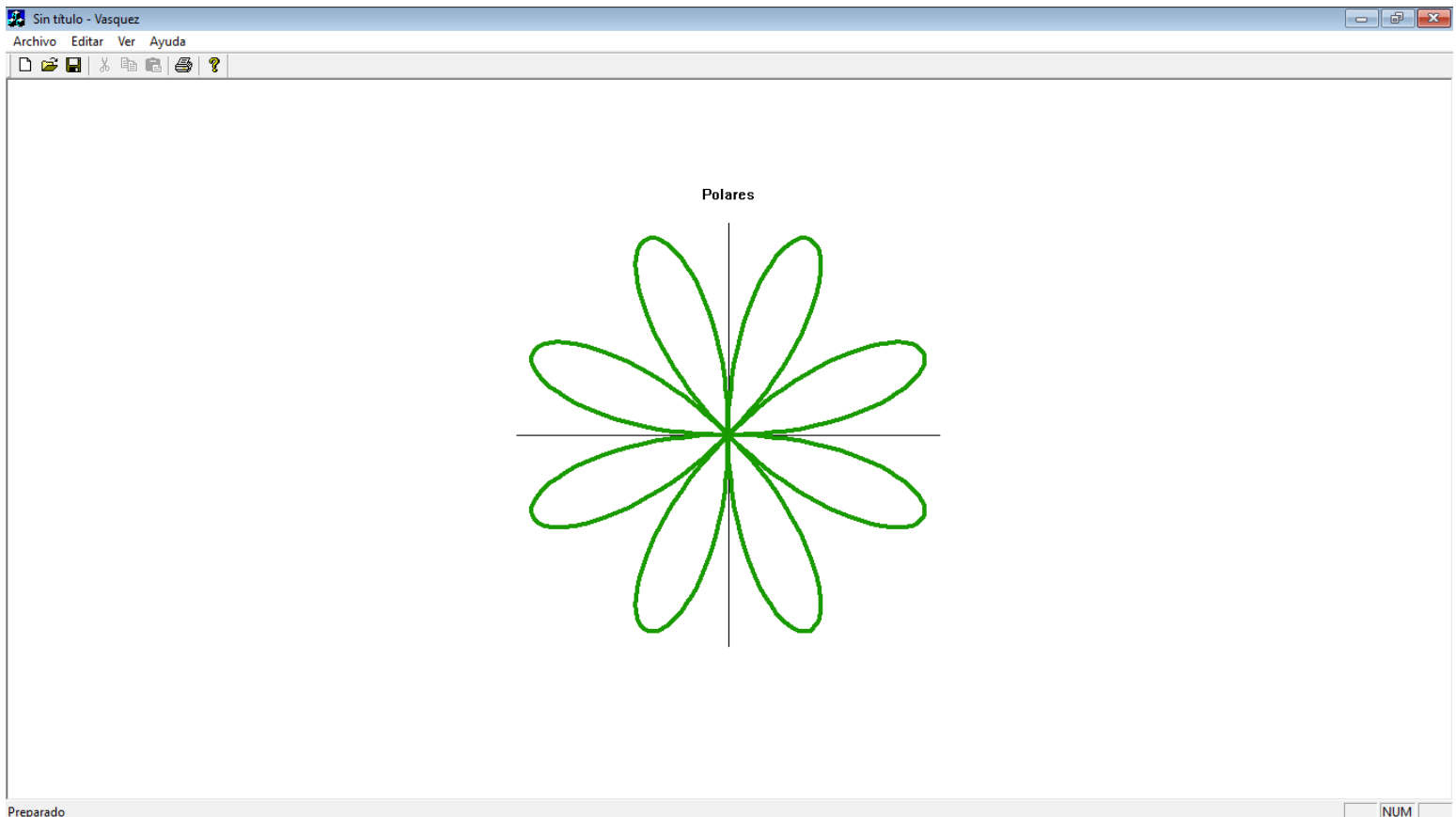


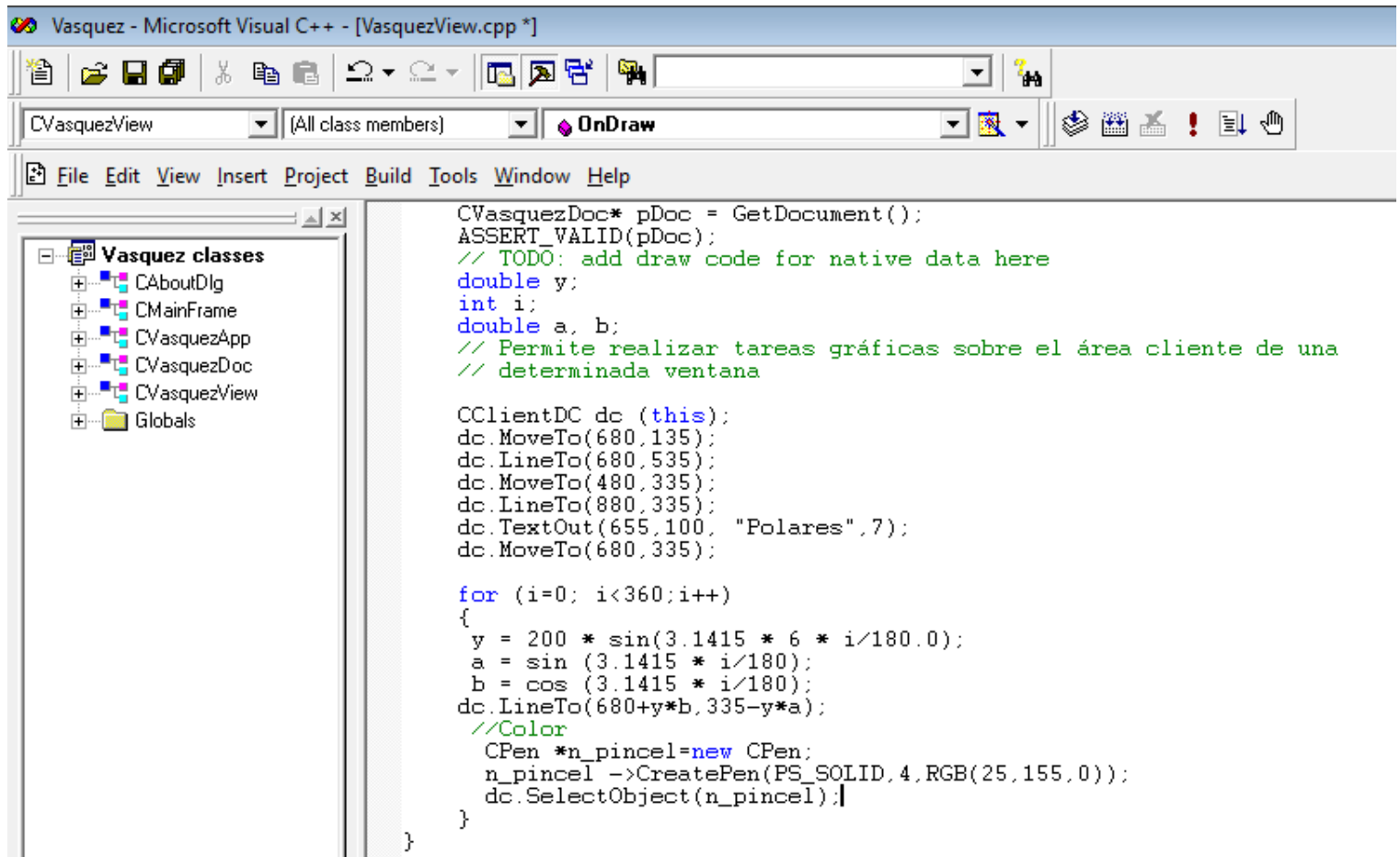
Al ejecutarlo nos mostrara lo siguiente:





Al ejecutarlo nos mostrara lo siguiente:





Al ejecutarlo nos mostrara lo siguiente:

