

GRAFICOS

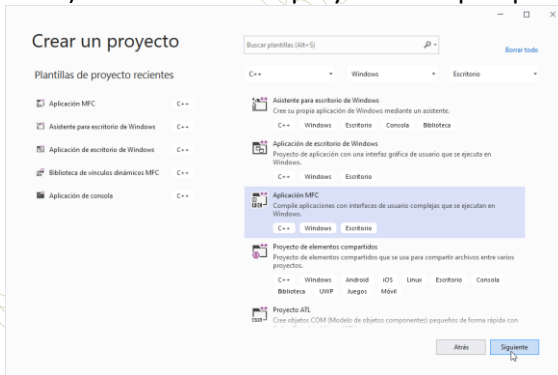
Para ver los diferentes métodos de la clase CDC, usar el siguiente link:

<https://docs.microsoft.com/es-es/cpp/mfc/reference/cdc-class?view=msvc-160>

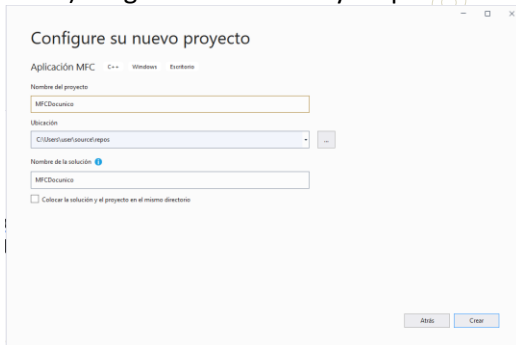
Existen 2 espacios donde graficar, uno de ellos es sobre algún control ubicado sobre una aplicación basado en cuadro de diálogo. También se puede graficar sobre el área de trabajo de una aplicación basado en documentos.

Aplicación basado en un documento

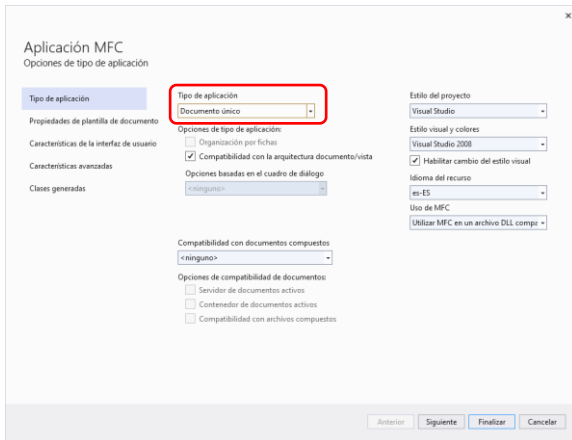
1) Crear un nuevo proyecto de tipo Aplicación MFC



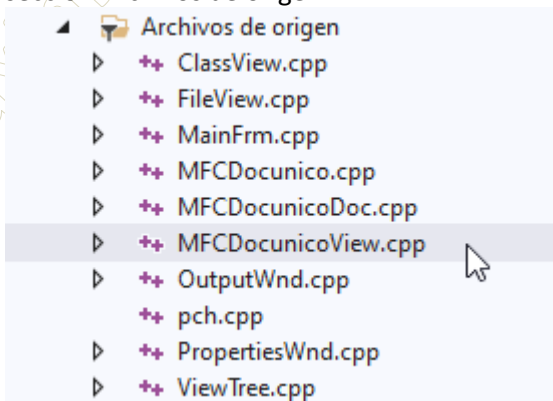
2) Ingresar el nombre y carpeta de trabajo de nuestro programa



3) Elegir la opción documento único y Finalizar.



- 4) Ingresar al archivo cuyo nombre termina en View.cpp ubicado en la sección Archivos de origen.



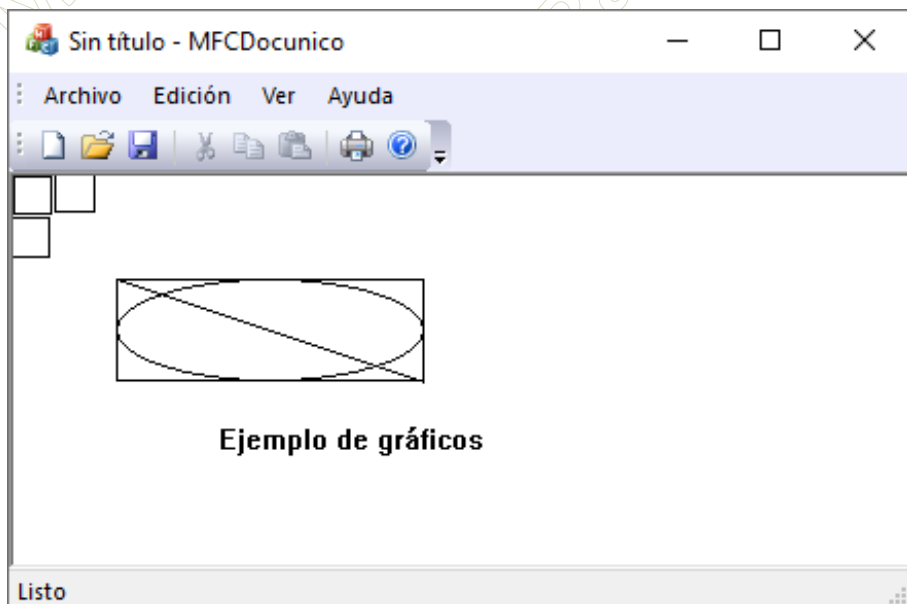
- 5) Ubicar el método OnDraw ya generado de la clase xxxxView.

```
void CMFCDocunicoView::OnDraw(CDC* /*pDC*/)
{
    CMFCDocunicoDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    // TODO: agregar aquí el código de dibujo para datos nativos
}
```

- 6) Este método tiene un parámetro de tipo puntero a la clase CDC, que apunta a la parte gráfica del documento.
- 7) Agregar la variable en el parámetro y usar los métodos de la clase CDC para graficar. Por ejemplo puede modificar de la siguiente manera:

```
void CMFCDocunicoView::OnDraw(CDC* pDC)
{
    CMFCDocunicoDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;
    pDC->Rectangle(0, 0, 19, 19);
    pDC->Rectangle(-1, 20, 18, 40);
    pDC->Rectangle(20, -1, 40, 18);
    pDC->Rectangle(50, 50, 200, 100);
    pDC->Ellipse(50, 50, 200, 100);
    pDC->MoveTo(50, 50);
    pDC->LineTo(200, 100);
    pDC->SetBkMode(TRANSPARENT);
    pDC->TextOutW(100, 120, _T("Ejemplo de gráficos"));
}
```

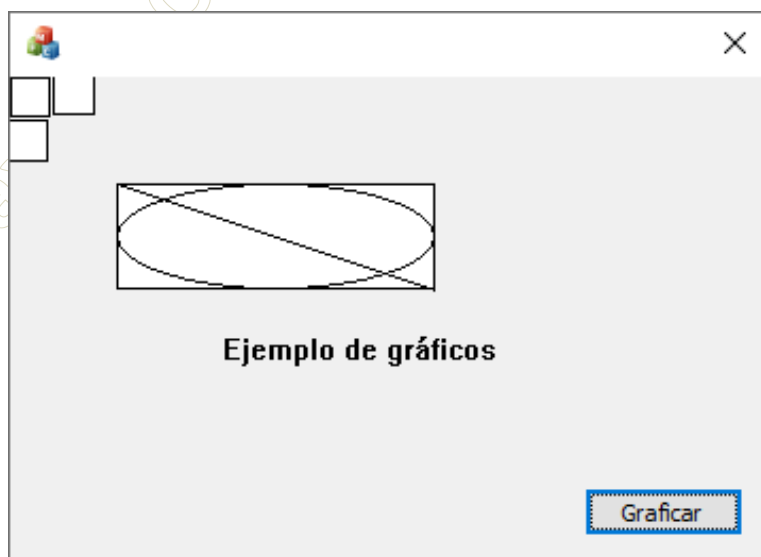


METODOS BASICOS PARA GRAFICAR

Aplicaciones basado en cuadro de dialogo

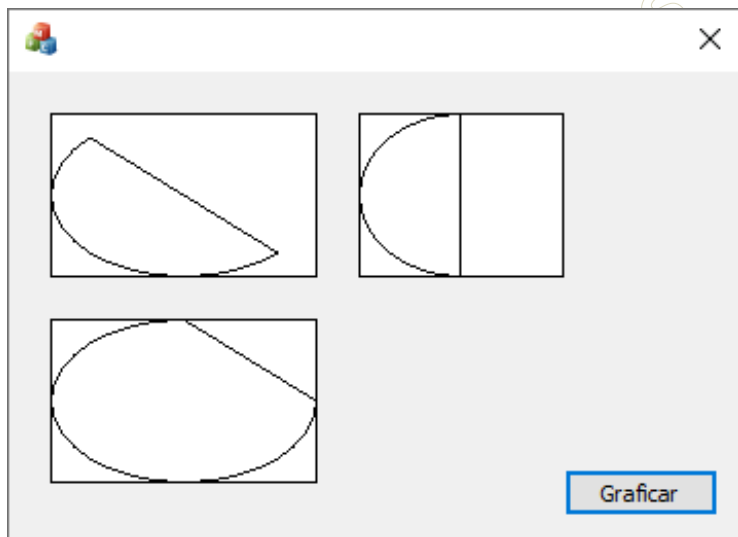
Ejemplo 1: Métodos básicos

```
CClientDC p(this);
p.Rectangle(0, 0, 19, 19);
p.Rectangle(-1, 20, 18, 40);
p.Rectangle(20, -1, 40, 18);
p.Rectangle(50, 50, 200, 100);
p.Ellipse(50, 50, 200, 100);
p.MoveTo(50, 50);
p.LineTo(200, 100);
p.SetBkMode(TRANSPARENT);
p.TextOutW(100, 120, _T("Ejemplo de gráficos"));
```



Ejemplo 2: Manejo de cuerdas

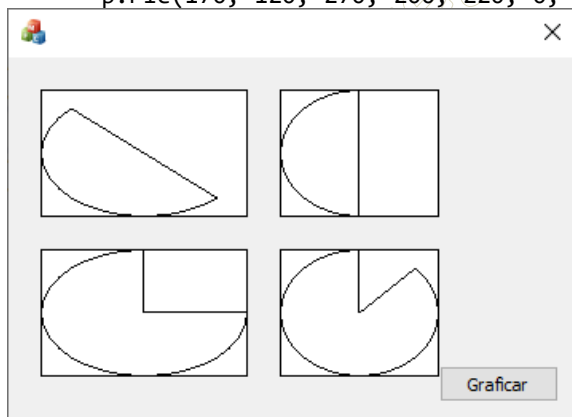
```
CClientDC p(this);
p.Rectangle(20, 20, 150, 100);
p.Chord(20, 20, 150, 100, 20, 20, 150, 100);
p.Rectangle(170, 20, 270, 100);
p.Chord(170, 20, 270, 100, 220, 0, 220, 150);
p.Rectangle(20, 120, 150, 200);
p.Chord(20, 120, 150, 200, 85, 120, 150, 160);
```



Ejemplo 3: Manejo de Tortas

```

CClientDC p(this);
p.Rectangle(20, 20, 150, 100);
p.Pie(20, 20, 150, 100, 20, 20, 150, 100);
p.Rectangle(170, 20, 270, 100);
p.Pie(170, 20, 270, 100, 220, 0, 220, 150);
p.Rectangle(20, 120, 150, 200);
p.Pie(20, 120, 150, 200, 85, 120, 150, 160);
p.Rectangle(170, 120, 270, 200);
p.Pie(170, 120, 270, 200, 220, 0, 270, 120);
  
```

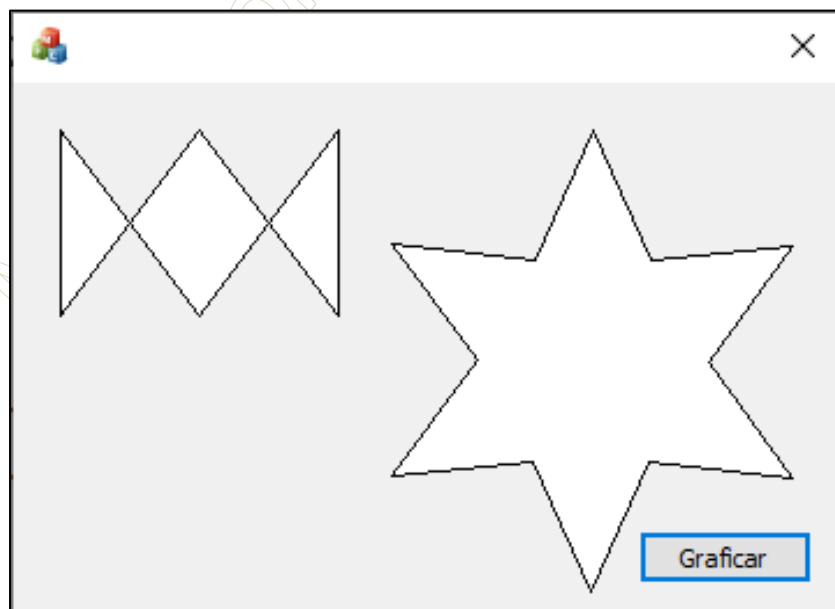


Ejemplo 4: Manejo de polígonos

```

CClientDC p(this);
POINT v1[6] = { 20,20,80,100,140,20,140,100,80,20,20,100 },v2[100];
float pi = atan(1.0) * 4;
p.Polygon(v1, 6);
int r,ri = 50, re = 100,cx=250,cy=120,np=12;
for (int a = 0; a < np; a++)
{
    r = re;
    if (a % 2 == 0) r = ri;
    v2[a].x = cx + r * cos(a * 2* pi / np);
    v2[a].y = cy + r * sin(a * 2* pi / np);
}
p.Polygon(v2, np);

```



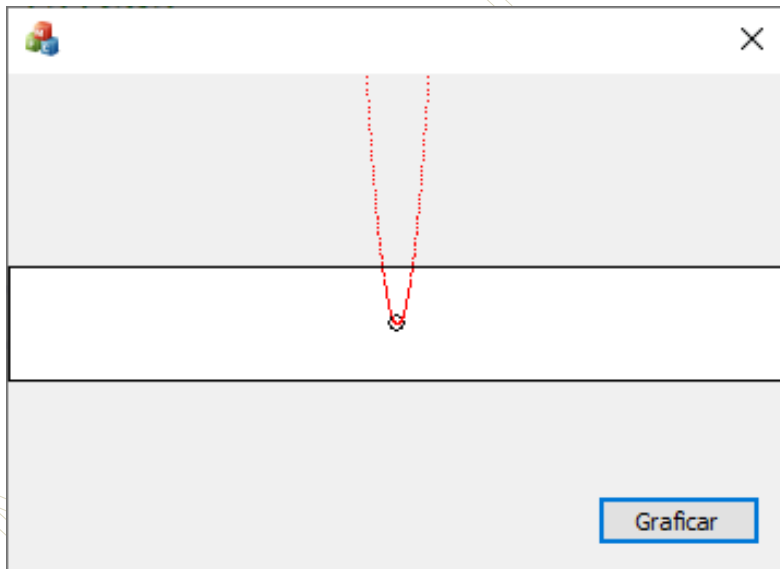
CAMBIO DEL SISTEMA DE REFERENCIA

Ejemplo 1: Cambio de escala forma 1

```

CClientDC p(this);
CRect rect;
GetClientRect(rect);
int cx = (rect.right - rect.left);
int cy = (rect.bottom - rect.top);
p.SetMapMode(MM_ISOTROPIC);
p.SetViewportOrg(cx/2, cy/2); //coordenadas fisicas
p.SetViewportExt(cx, cy);
p.SetWindowOrg(0, 0); //Coordenadas lógicas
p.SetWindowExt(2000, -300);
p.Rectangle(-1000, -150, 1000, 150);
p.Ellipse(-20, -20, 20, 20);
for (int x = -250; x <= 250; x++)
{
    int y = x * x * 0.1;
    p.SetPixel(x, y, RGB(255, 0, 0));
}

```

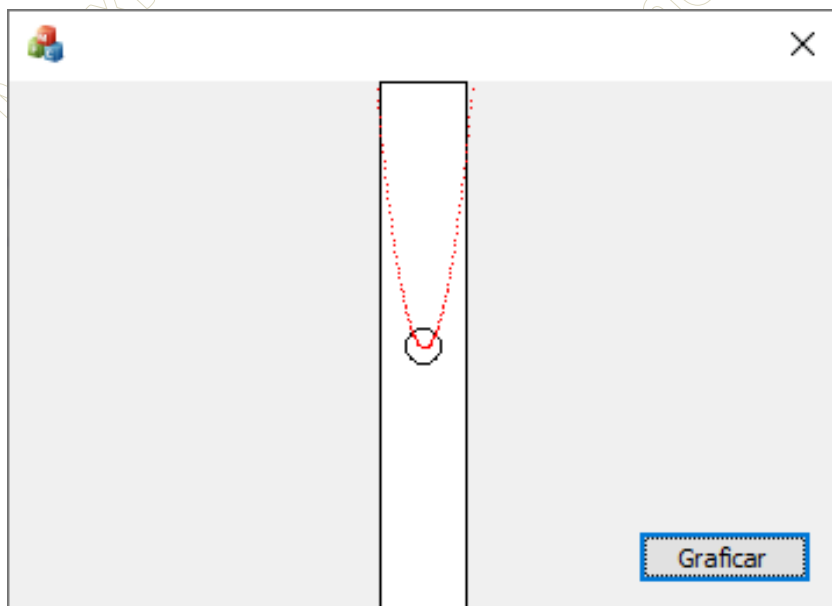


Ejemplo 2: Cambio de escala forma 2

```

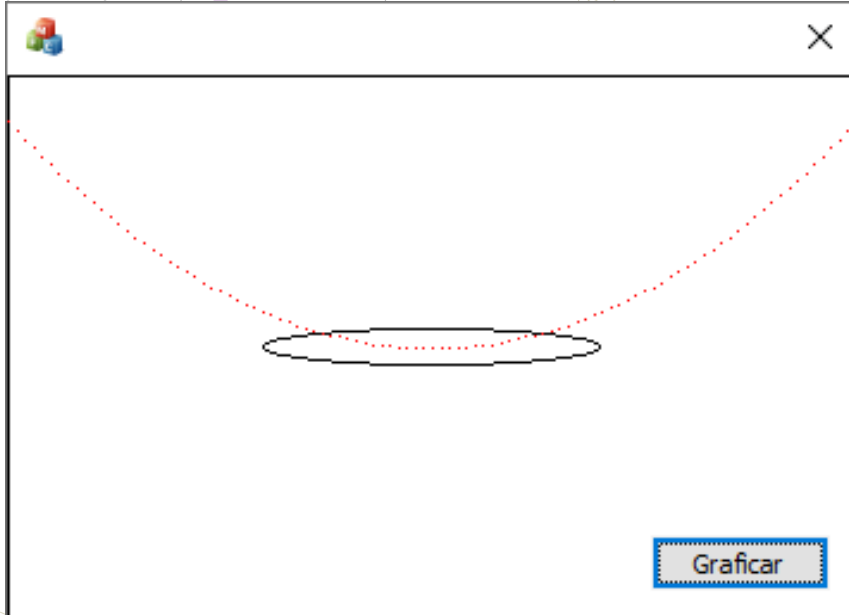
CClientDC p(this);
CRect rect;
GetClientRect(rect);
int cx = (rect.right - rect.left);
int cy = (rect.bottom - rect.top);
p.SetMapMode(MM_ISOTROPIC);
p.SetViewportOrg(cx/2, cy/2); //coordenadas fisicas
p.SetViewportExt(cx, cy);
p.SetWindowOrg(0, 0); //Coordenadas lógicas
p.SetWindowExt(100, -600);
p.Rectangle(-50, -300, 50, 300);
p.Ellipse(-20, -20, 20, 20);
for (int x = -250; x <= 250; x++)
{
    int y = x * x * 0.1;
    p.SetPixel(x, y, RGB(255, 0, 0));
}

```



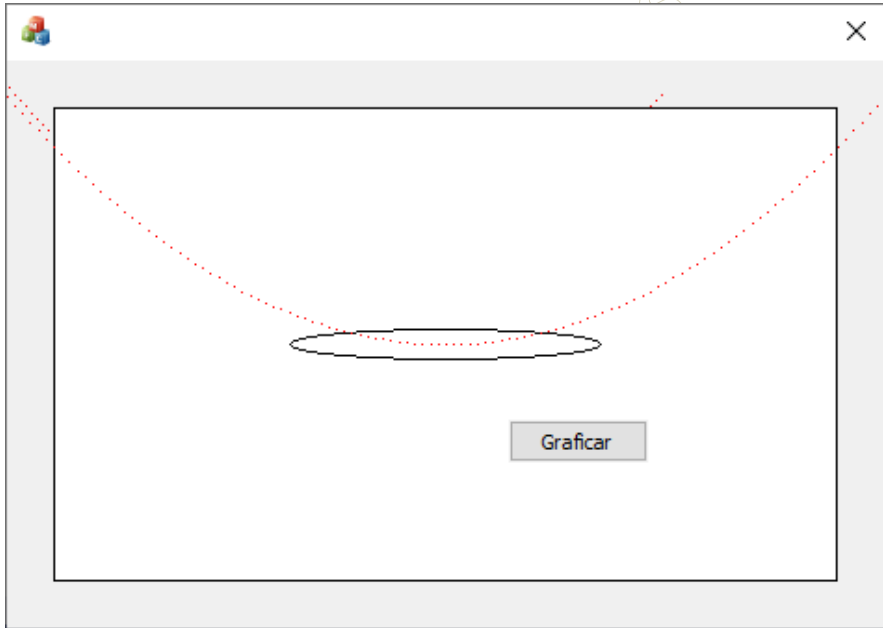
Ejemplo 3: Cambio de escala forma 3

```
p.SetMapMode(MM_ANISOTROPIC);
```



Ejemplo 4: Cambio de escala forma 4

```
CClientDC p(this);
CRect rect;
GetClientRect(rect);
int cx = (rect.right - rect.left);
int cy = (rect.bottom - rect.top);
int margen = 25;
p.SetMapMode(MM_ANISOTROPIC);
p.SetViewportOrg(cx/2, cy/2); //coordenadas fisicas
p.SetViewportExt(cx-margen*2, cy-margen*2);
p.SetWindowOrg(0, 0); //Coordenadas lógicas
p.SetWindowExt(100, -600);
p.Rectangle(-50, -300, 50, 300);
p.Ellipse(-20, -20, 20, 20);
for (int x = -250; x <= 250; x++)
{
    int y = x * x * 0.1;
    p.SetPixel(x, y, RGB(255, 0, 0));
}
```

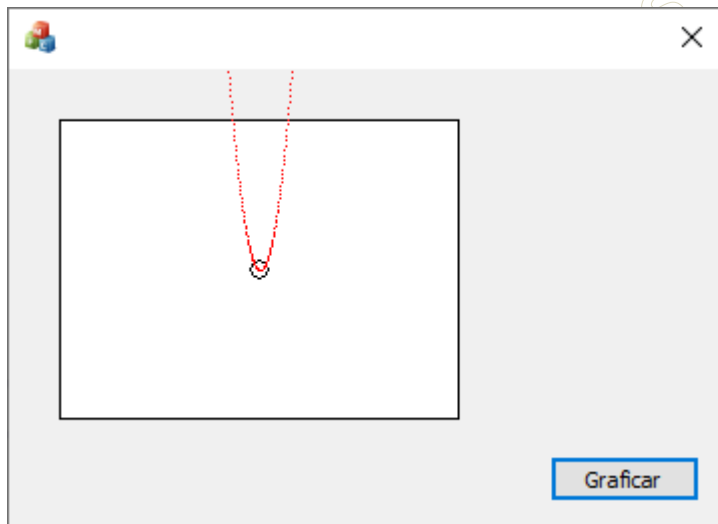


Ejemplo 5: Cambio de escala forma 5

```

CClientDC p(this);
int msup = 25, mizq=25, ancho=200, alto=150;
p.SetMapMode(MM_ANISOTROPIC);
p.SetViewportOrg(mizq+ancho/2, msup+alto/2); //coordenadas fisicas
p.SetViewportExt(ancho, alto);
p.SetWindowOrg(0, 0); //Coordenadas lógicas
p.SetWindowExt(800, -600);
p.Rectangle(-400, -300, 400, 300);
p.Ellipse(-20, -20, 20, 20);
for (int x = -200; x <= 200; x++)
{
    int y = x * x * 0.1;
    p.SetPixel(x, y, RGB(255, 0, 0));
}

```



CONFIGURANDO UNA APLICACIÓN BASADO EN DOCUMENTO ÚNICO

MENÚ PERSONALIZADO

Para cambiar el menú que viene por defecto, deberá ingresar al archivo de la aplicación principal MFC[nombreakcion].cpp, y reemplazar a IDR_MAINFRAME, por el ID del nuevo menú creado.

MFCApplication1.cpp

```
pDocTemplate = new CSingleDocTemplate(
    IDR_MAINFRAME,
    RUNTIME_CLASS(CMFCApplication1Doc),
    RUNTIME_CLASS(CMainFrame),    // Ventana de marco SDI principal
    RUNTIME_CLASS(CMFCApplication1View));
```

BARRA DE HERRAMIENTAS

Para cancelar la barra de herramientas anular algunas líneas de código MainFrm.cpp

```
// TODO: Elimine estas cinco líneas si no desea que la barra de herramientas y la barra de menús se puedan acoplar
//m_wndMenuBar.EnableDocking(CBRS_ALIGN_ANY);
//m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
EnableDocking(CBRS_ALIGN_ANY);
//DockPane(&m_wndMenuBar);
DockPane(&m_wndToolBar);
```

ICONO DE LA APLICACIÓN Y TÍTULO

MFCApplication1.rc

```
////////////////////////////////////
////////////////////////////////////
//
// Icon
//

// Icon with lowest ID value placed first to ensure application
icon
// remains consistent on all systems.
IDR_MAINFRAME      ICON          "res\\MFCApplication4.ico"
IDR_MENU1          ICON          "res\\MFCApplication4.ico"
```


TIPOS DE TRAZOS , FONDO Y LETRAS

Código de colores: En internet existen aplicaciones online que te permiten generar código de colores en RGB, por ejemplo:

https://www.rapidtables.com/web/color/RGB_Color.html

CPen: Es la clase para el tipo de trazo.

<https://docs.microsoft.com/es-es/cpp/mfc/reference/cpen-class?view=msvc-160#createpen>

CBrush: Es la clase que maneja el tipo de fondo

<https://docs.microsoft.com/es-es/cpp/mfc/reference/cbrush-class?view=msvc-160>

Ejemplo1: Manejando tipo y color de trazo

Cree una aplicación basada en documento único o basado en cuadro de diálogo.

```
void CMFCApplication2View::OnDibujarGarabatos()
```

```
{
    CClientDC p(this);
    CPen t;
    t.CreatePen(PS_DOT, 1, RGB(255, 0, 0));p.SelectObject(t);
    p.Rectangle(10, 10, 50, 50);
    t.DeleteObject();t.CreatePen(PS_DASH, 1, RGB(0, 0, 255));p.SelectObject(t);
    p.Rectangle(60, 10, 100, 50);
    t.DeleteObject();t.CreatePen(PS_SOLID, 1, RGB(0, 0, 255));p.SelectObject(t);
    p.Rectangle(10, 60, 50, 100);
    t.DeleteObject();t.CreatePen(PS_SOLID, 5, RGB(25, 25, 112));p.SelectObject(t);
    p.Rectangle(60, 60, 100, 100);
}
```



Ejemplo2: Manejando tipo y color de fondo

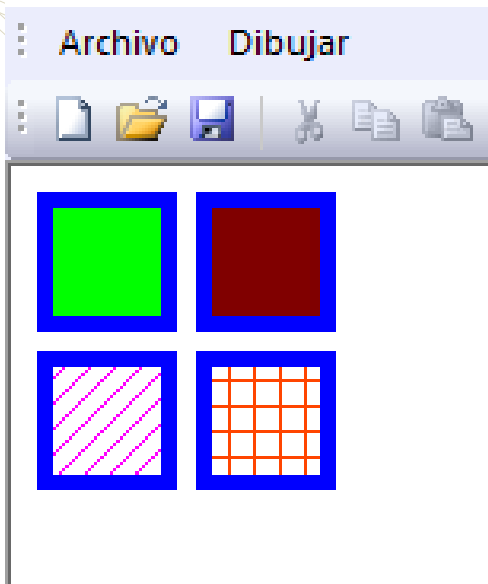
```

void CMFCAApplication2View::OnDibujarGarabatos()
{
    CClientDC p(this);
    CPen t;
    CBrush f;
    t.CreatePen(PS_SOLID, 5, RGB(0, 0, 255)); p.SelectObject(t);
    f.CreateSolidBrush(RGB(0, 255, 0)); p.SelectObject(f);
    p.Rectangle(10, 10, 50, 50);
    f.DeleteObject(); f.CreateSolidBrush(RGB(128, 0, 0));p.SelectObject(f);
    p.Rectangle(60, 10, 100, 50);
    f.DeleteObject(); f.CreateHatchBrush(HS_BDIAGONAL,RGB(255, 0, 255)); p.SelectObject(f);
    p.Rectangle(10, 60, 50, 100);
    f.DeleteObject(); f.CreateHatchBrush(HS_CROSS, RGB(255, 69, 0)); p.SelectObject(f);
    p.Rectangle(60, 60, 100, 100);
}

```



Sin título -



El sol sale para todos

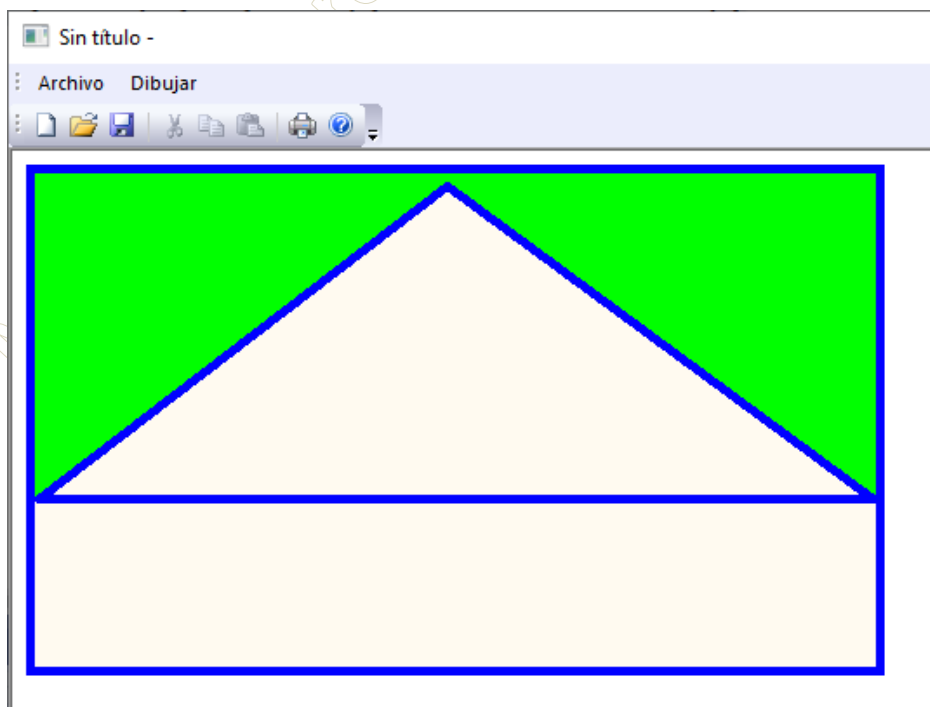
El sol sale para todos

Ejemplo 4: Usando el bote de relleno modelo 1

```

CClientDC p(this);
CPen t; CBrush f;
t.CreatePen(PS_SOLID, 5, RGB(0, 0, 255)); p.SelectObject(t);
f.CreateSolidBrush(RGB(255, 250, 240)); p.SelectObject(f);
p.Rectangle(10, 10, 500, 300);
t.DeleteObject(); t.CreatePen(PS_SOLID, 5, RGB(0, 0, 255)); p.SelectObject(t);
p.MoveTo(15, 200);p.LineTo(495, 200);p.LineTo(250, 20);p.LineTo(15, 200);
f.DeleteObject(); f.CreateSolidBrush(RGB(0, 255, 0)); p.SelectObject(f);
p.ExtFloodFill(20, 20, RGB(0, 0, 255),FLOODFILLBORDER);

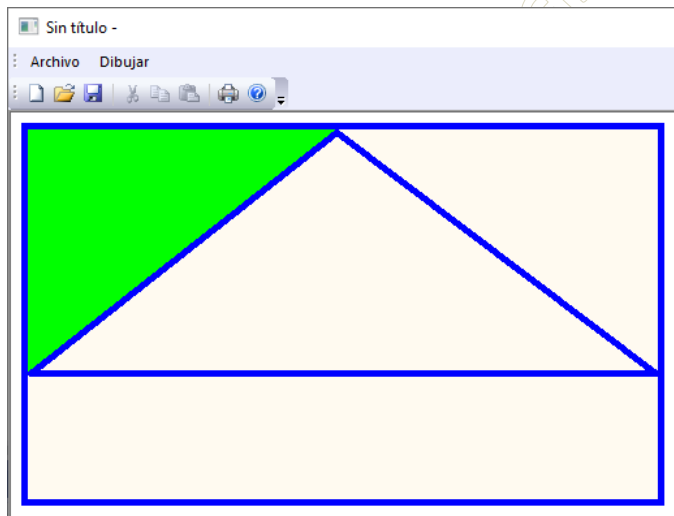
```



Ejemplo 5: Usando el bote de relleno modelo 2

Modificamos solo esta línea:

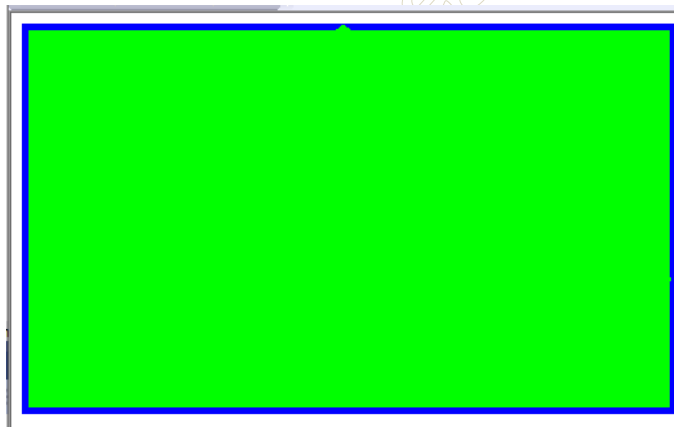
```
p.MoveTo(15, 200);p.LineTo(495, 200);p.LineTo(250, 15);p.LineTo(15, 200);
```



Ejemplo 5: Usando el bote de relleno modelo 2

Modificamos solo esta línea:

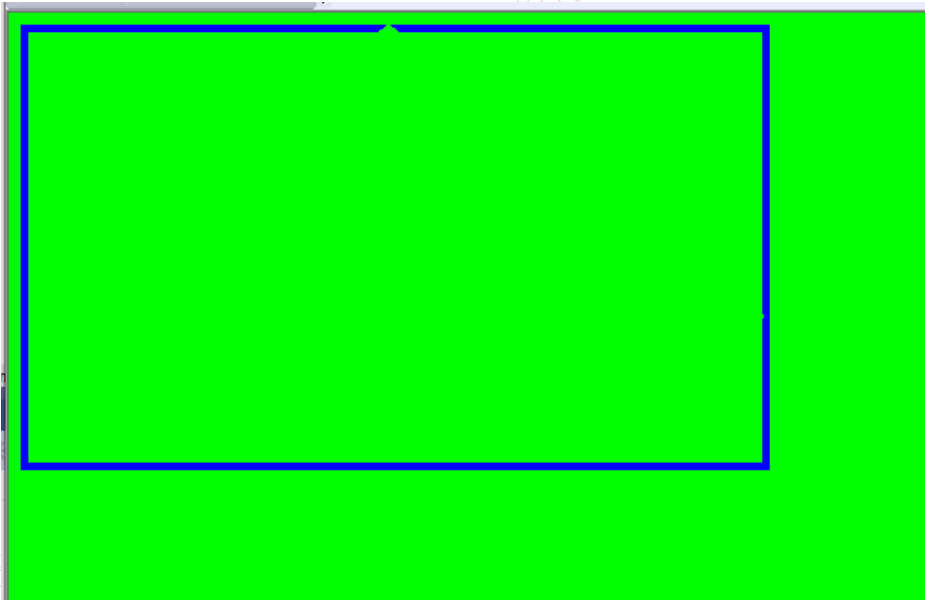
```
t.DeleteObject(); t.CreatePen(PS_SOLID, 5, RGB(255, 0, 0)); p.SelectObject(t);  
p.MoveTo(15, 200);p.LineTo(495, 200);p.LineTo(250, 11);p.LineTo(15, 200);
```



Ejemplo 6: Usando el bote de relleno modelo 2

Modificamos solo esta línea:

```
p.MoveTo(15, 200);p.LineTo(495, 200);p.LineTo(250, 10);p.LineTo(15, 200);
```



Ejemplo 7: Usando el pixel básico

```

CClientDC p(this);
CPen t;   CBrush f;
t.CreatePen(PS_SOLID, 5, RGB(0, 0, 255)); p.SelectObject(t);
f.CreateSolidBrush(RGB(255, 250, 240)); p.SelectObject(f);
p.Rectangle(10, 10, 500, 300);
p.SetPixel(100, 100, RGB(255, 0, 0));
p.SetPixel(100, 101, RGB(255, 0, 0));
p.SetPixel(101, 100, RGB(255, 0, 0));
p.SetPixel(101, 101, RGB(255, 0, 0));

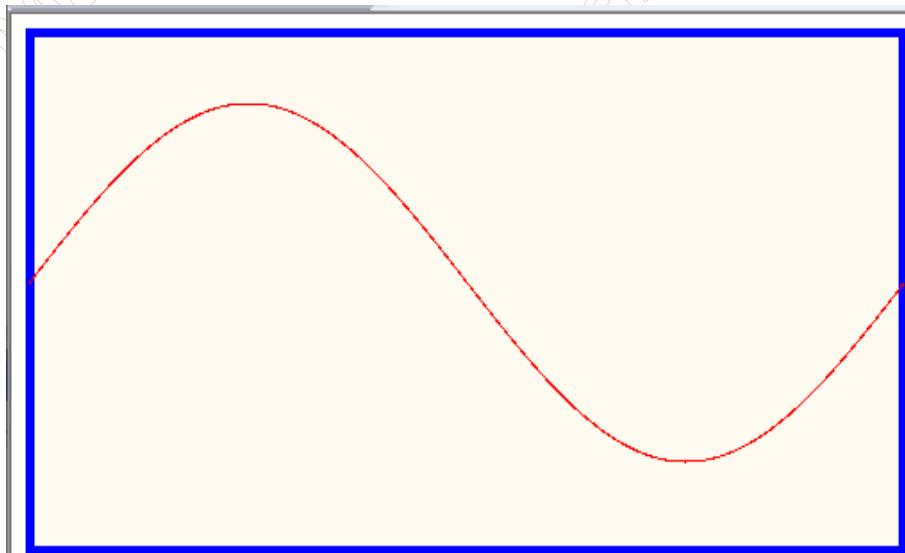
```

Ejemplo 8: Usando el pixel

```

CClientDC p(this);
CPen t;   CBrush f;
float pi = atan(1.0) * 4;
t.CreatePen(PS_SOLID, 5, RGB(0, 0, 255)); p.SelectObject(t);
f.CreateSolidBrush(RGB(255, 250, 240)); p.SelectObject(f);
p.Rectangle(10, 10, 500, 300);
for (int i = 0; i < 490*2; i++)
    p.SetPixel(i/2 + 10, 150 - 100*sin(i * 2 * pi / (2*490)), RGB(255, 0, 0));

```



REUTILIZANDO LA CLASE CCLIENTDC

Agregar esta clase:

```
class CDibujar :public CClientDC {
public:
    CDibujar(CWnd* x);
};

CDibujar::CDibujar(CWnd* x):CClientDC(x){}
```

Con esta clase puede agregar más funcionalidad para dibujar.

Ejemplo 1: Agregando más funcionalidad a los gráficos

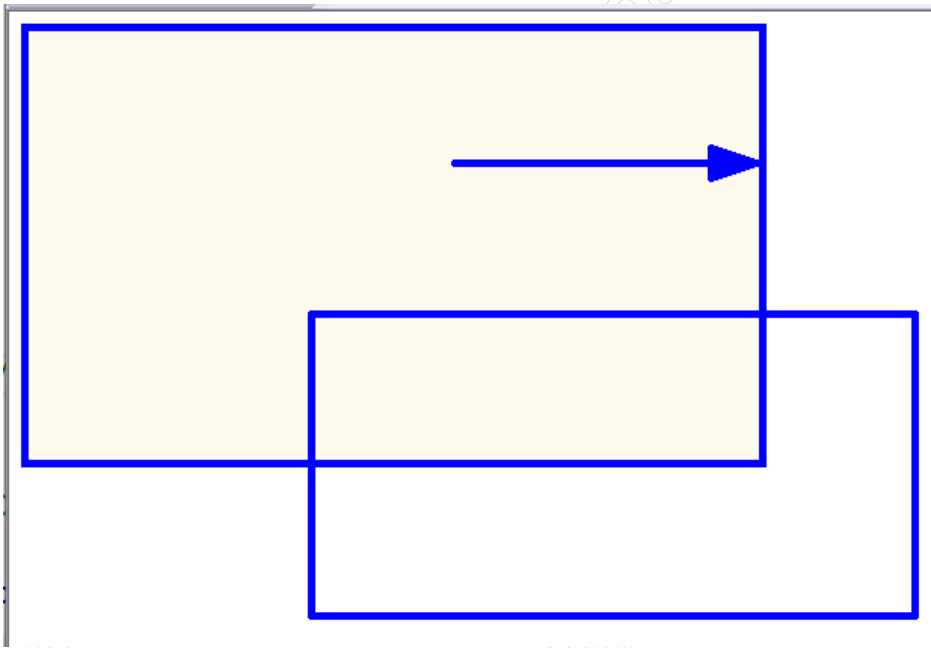
Agregue los siguientes métodos:

```
void CDibujar::dib_rectangulo(int x1, int y1, int x2, int y2)
{
    MoveTo(x1, y1); LineTo(x2, y1); LineTo(x2, y2);
    LineTo(x1, y2); LineTo(x1, y1);
}

void CDibujar::dib_flechader(int x, int y, int longitud, int cabancho, int cablon)
{
    CPen t; t.CreatePen(PS_SOLID, 2, RGB(0,0,255));
    MoveTo(x - longitud, y); LineTo(x - cablon, y);
    MoveTo(x, y); LineTo(x - cablon, y - cabancho / 2);
    LineTo(x - cablon, y + cabancho / 2); LineTo(x, y);
    CBrush f; f.CreateSolidBrush(RGB(0, 0, 255)); SelectObject(f);
    ExtFloodFill(x - cablon / 2, y, RGB(0, 0, 255), FLOODFILLBORDER);
}
```

Agregar el código a una opción del menú.

```
CDibujar p(this);
CPen t; CBrush f;
t.CreatePen(PS_SOLID, 5, RGB(0, 0, 255)); p.SelectObject(t);
f.CreateSolidBrush(RGB(255, 250, 240)); p.SelectObject(f);
p.Rectangle(10, 10, 500, 300);
p.dib_rectangulo(200, 200, 600, 400);
p.dib_flechader(495, 100, 200, 20, 30);
```



Ejemplo 2: Agregue el siguiente método

```
void CDibujar::dib_flecha(int x1, int y1, int x2, int y2, int cabancho, int cablon,
                           int grosor, COLORREF col)
{
    CPen t, *tb; t.CreatePen(PS_SOLID, grosor, col);
    tb = GetCurrentPen(); SelectObject(t);
    CBrush f, *fb; f.CreateSolidBrush(col);
    fb = GetCurrentBrush(); SelectObject(f);
    float lon = sqrt(pow(y2 - y1, 2) + pow(x2 - x1, 2));
    float ux = (x2 - x1) / lon, uy = (y2 - y1) / lon;
    float unx = -uy, uny = ux;
    MoveTo(x1, y1); LineTo(x2 - cablon*ux, y2 - cablon*uy);
    MoveTo(x2, y2); LineTo(x2 - cablon * ux + unx*cabancho/2, y2 - cablon * uy + uny*cabancho/2);
    LineTo(x2 - cablon * ux - unx * cabancho / 2, y2 - cablon * uy - uny * cabancho / 2); LineTo(x2, y2);
    ExtFloodFill(x2 - cablon*ux/1.5, y2 - cablon*uy/1.5, col, FLOODFILLBORDER);
    SelectObject(*tb); SelectObject(*fb);
}
```

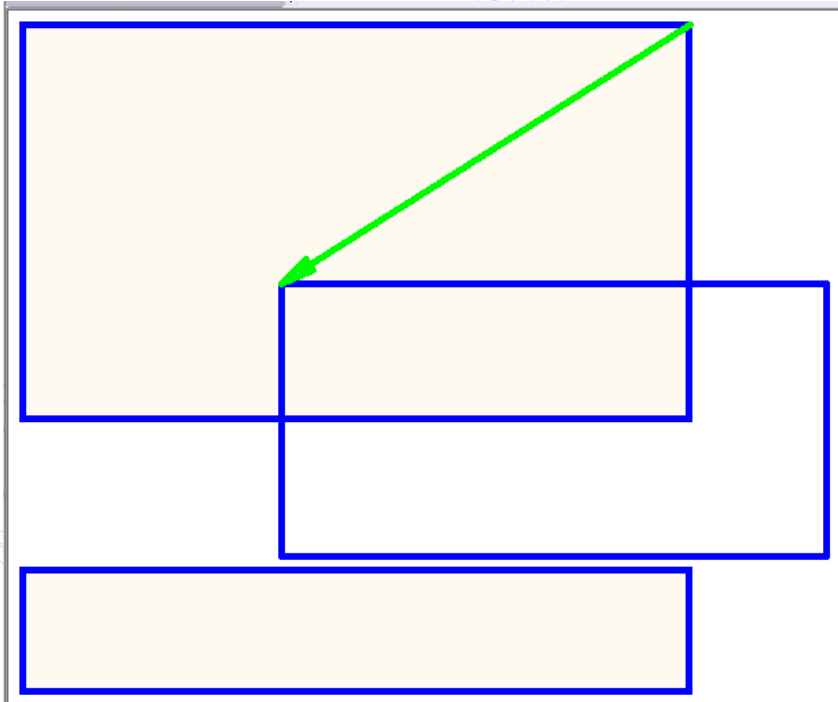
Agregar el código a una opción del menú.

```
CDibujar p(this);
CPen t; CBrush f;
t.CreatePen(PS_SOLID, 5, RGB(0, 0, 255)); p.SelectObject(t);
f.CreateSolidBrush(RGB(255, 250, 240)); p.SelectObject(f);
```

```

p.Rectangle(10, 10, 500, 300);
p.dib_rectangulo(200, 200, 600, 400);
p.dib_flecha(500, 10, 200, 200, 10, 25, 5, RGB(0, 250, 0));
p.Rectangle(10, 410, 500, 500);

```

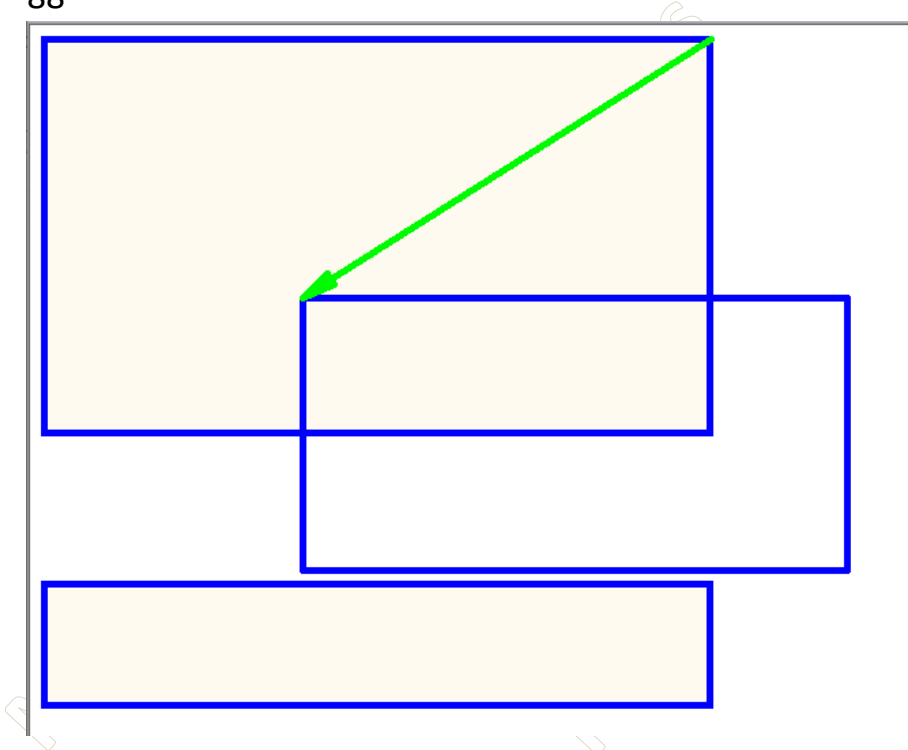


Versión mejorada para dibujar la flecha:

```

void CDibujar::dib_flecha(int x1, int y1, int x2, int y2, int cabancho, int cablon,
    int grosor, COLORREF col)
{
    CPen t, *tb; t.CreatePen(PS_SOLID, grosor, col);
    tb = GetCurrentPen(); SelectObject(t);
    CBrush f, *fb; f.CreateSolidBrush(col);
    fb = GetCurrentBrush(); SelectObject(f);
    float lon = sqrt(pow(y2 - y1, 2) + pow(x2 - x1, 2));
    float ux = (x2 - x1) / lon, uy = (y2 - y1) / lon;
    float unx = -uy, uny = ux;
    POINT tri[3] = { x2, y2, x2 - cablon * ux + unx * cabancho / 2, y2 - cablon * uy + uny * cabancho / 2,
        x2 - cablon * ux - unx * cabancho / 2, y2 - cablon * uy - uny * cabancho / 2 };
    MoveTo(x1, y1); LineTo(x2 - cablon * ux, y2 - cablon * uy);
    Polygon(tri, 3);
    SelectObject(*tb); SelectObject(*fb);
}

```

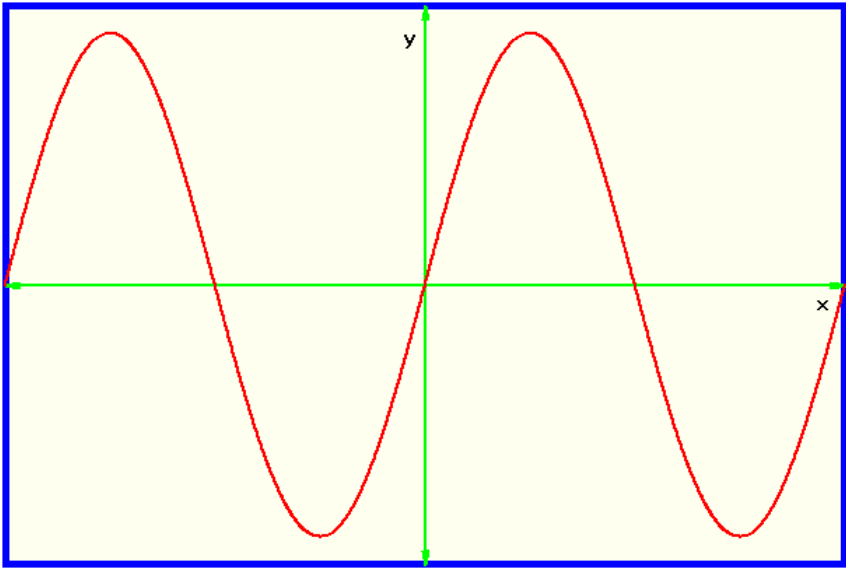
Ejemplo 3: Gráfico de la función seno usando un marco de fondo

Agregar la siguiente función miembro a la anterior

```
void CDibujar::dib_fondografico(int x1, int y1, int x2, int y2, COLORREF colp, COLORREF colb, COLORREF colf)
{
    CPen t;    CBrush f; int cablon = 10, cabancho = 4, grosor = 2;
    t.CreatePen(PS_SOLID, 5, colp); SelectObject(t);
    f.CreateSolidBrush(colb); SelectObject(f);
    Rectangle(x1, y1, x2, y2);
    dib_flecha(x1 + (x2 - x1) / 2, y1 + (y2 - y1) / 2, x2, y1 + (y2 - y1) / 2, cabancho, cablon, grosor, colf);
    dib_flecha(x1 + (x2 - x1) / 2, y1 + (y2 - y1) / 2, x1, y1 + (y2 - y1) / 2, cabancho, cablon, grosor, colf);
    dib_flecha(x1 + (x2 - x1) / 2, y1 + (y2 - y1) / 2, x1 + (x2 - x1) / 2, y1, cabancho, cablon, grosor, colf);
    dib_flecha(x1 + (x2 - x1) / 2, y1 + (y2 - y1) / 2, x1 + (x2 - x1) / 2, y2, cabancho, cablon, grosor, colf);
    SetBkMode(TRANSPARENT);
    TextOutW(x2 - 20, y1 + (y2 - y1) / 2 + 5, _T("x"));
    TextOutW(x1 + (x2 - x1) / 2 - 15, y1 + 15, _T("y"));
}
```

Programa en alguna opción del menú

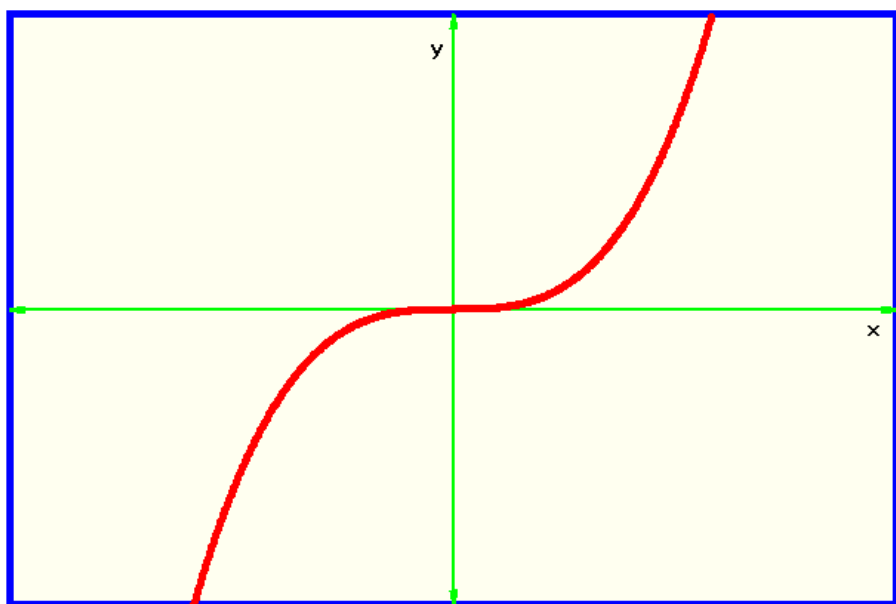
```
void CMFCAApplication4View::OnArchivoGarabatos()
{
    CDibujar p(this);
    float pi = atan(1.0) * 4;
    // p.dib_fondografico(10, 10, 300, 200, RGB(0, 0, 255), RGB(255, 255, 240), RGB(255, 0, 0));
    // p.dib_fondografico(310, 10, 600, 200, RGB(0, 0, 255), RGB(255, 255, 240), RGB(255, 0, 0));
    p.dib_fondografico(10, 10, 610, 410, RGB(0, 0, 255), RGB(255, 255, 240), RGB(0, 255, 0));
    CPen t; t.CreatePen(PS_SOLID, 2, RGB(255, 0, 0)); p.SelectObject(t);
    int npx = 2000;
    for (int i = 0; i <= npx; i++)
    {
        float x = -2*pi+4 * pi * i / npx;
        int xx = 10 + 600 * i / npx;
        float y = sin(x);
        int yy = 210 - 180 * y;
        //p.SetPixel(x, y, RGB(0, 0, 255));
        if (i == 0)
            p.MoveTo(xx, yy);
        else
            p.LineTo(xx, yy);
    }
}
```



Ejemplo 4: Gráfico de la función cúbica.

Usando la estructura anterior, modifique el código correspondiente.

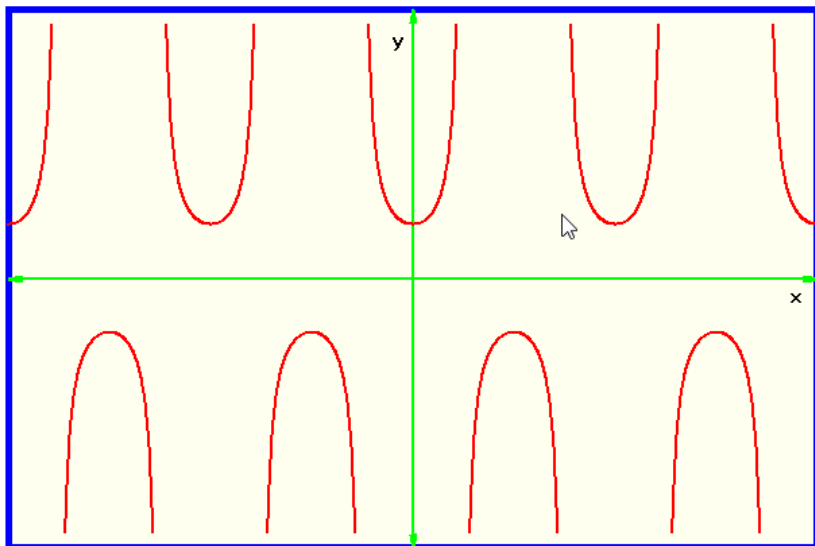
```
int npx = 1000;
for (int i = 0; i <= npx; i++)
{
    float x0 = -10, xf = 10;
    float x = x0 + (xf - x0) * i / npx;
    int xx = 10 + 600 * i / npx;
    float y = pow(x, 3);
    int yy = 210 - y;
    if (yy > 10 && yy < 410)
    {
        if (yy == 409)
            p.MoveTo(xx, yy);
        else
            p.LineTo(xx, yy);
    }
}
```



Ejemplo 5: Gráfico de la función secante

Usando la estructura anterior, modifique el código correspondiente.

```
int npx = 2000;
int ya;
for (int i = 0; i <= npx; i++)
{
    float x0 = -4 * pi, xf = 4 * pi;
    float x = x0+(xf-x0) * i / npx;
    int xx = 10 + 600 * i / npx;
    float y = 1/cos(x);
    int yy = 210 - y*40;
    if (yy > 10 && yy < 410)
    {
        if (i == 0 || abs(ya-yy)>10)
            p.MoveTo(xx, yy);
        else
            p.LineTo(xx, yy);
        ya = yy;
    }
}
```

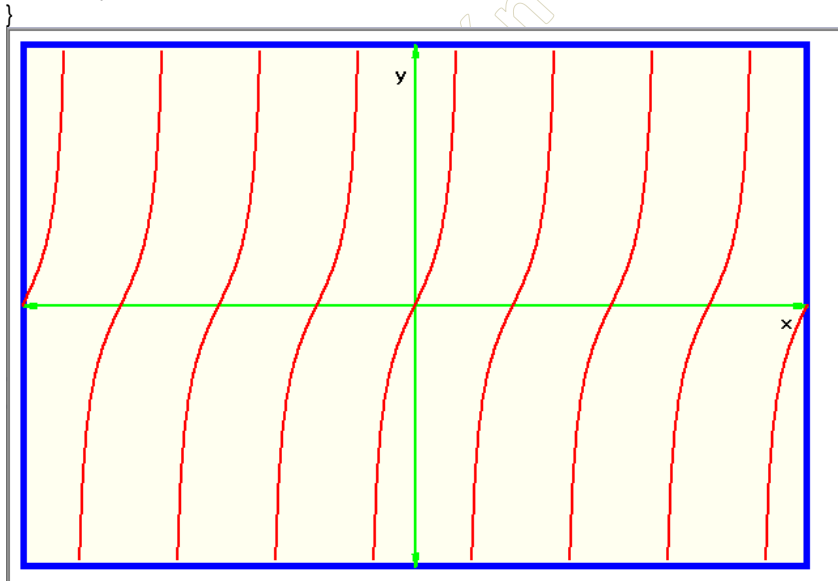


Ejemplo 6: Gráfico de la función tangente

Usando la estructura anterior, modifique el código correspondiente.

```
void CMFCApplication4View::OnArchivoGarabatos()
```

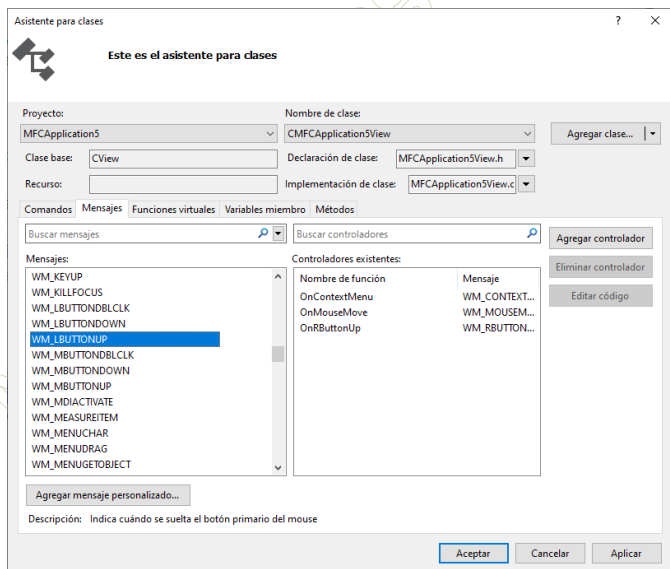
```
{
    CDibujar p(this);
    float pi = atan(1.0) * 4;
    p.dib_fondografico(10, 10, 610, 410, RGB(0, 0, 255), RGB(255, 255, 240), RGB(0, 255, 0));
    CPen t; t.CreatePen(PS_SOLID, 2, RGB(255, 0, 0)); p.SelectObject(t);
    int npix = 2000;
    int ya;
    for (int i = 0; i <= npix; i++)
    {
        float x0 = -4*pi, xf = 4*pi;
        float x = x0 + (xf - x0) * i / npix;
        int xx = 10 + 600 * i / npix;
        float y = tan(x);
        int yy = 210 - y*50;
        if (yy > 10 && yy < 410)
        {
            if (i == 0 || ya < yy)
                p.MoveTo(xx, yy);
            else
                p.LineTo(xx, yy);
            ya = yy;
        }
    }
}
```



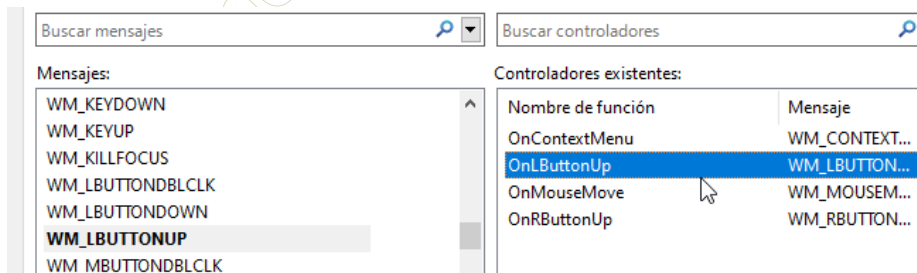
MANEJO DE EVENTOS ASOCIADOS AL MOUSE APLICADO SOBRE EL AREA DE TRABAJO

Esto es valido para aplicaciones basados en cuadro de dialogo(Dlg) como basado en documento único(View).

Presione Ctrl+Shift+X para llamar al asistente para clases, luego elija la clase CMF[nombre de la aplicación]View y luego elija la pestaña Mensaje:



Elija un Mensaje disponible WM_LBUTTONDOWN, haga doble click y se generara el controlador, es decir un método asociado a ese evento disponible, finalmente haga doble click para ingresar a la programación de dicho evento:



```

void CMFCApplication5View::OnLButtonUp(UINT nFlags, CPoint point)
{
    // TODO: Agregue aquí su código de controlador de mensajes
    o llame al valor predeterminado

    CView::OnLButtonUp(nFlags, point);
}

```

Este método tiene un parámetro importante de tipo CPoint, donde se registrará la coordenada del punto donde se realizó el click.

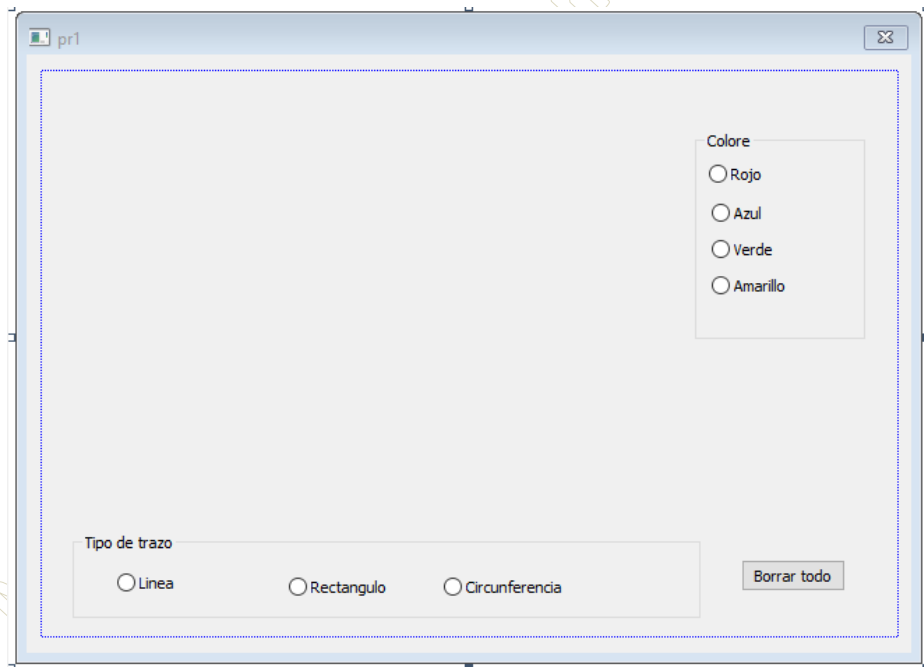
Ejemplo 1: Dibuja un cuadrado de 20px de lado centrado al pixel donde se hace clic.

```

void CMFCApplication2View::OnLButtonUp(UINT nFlags, CPoint point)
{
    CClientDC p(this);
    p.Rectangle(point.x-10,point.y-10, point.x + 10, point.y + 10);
    CView::OnLButtonUp(nFlags, point);
}

```

Ejemplo 3: Dibujando con eventos(LButtonUp y MouseMove) y variables globales



Identificador de control	Tipo	Miembro
<Variable personalizada>	HICON	m_hlcon
IDC_BUTTON1		
IDC_RADIO1	int	m_tt
IDC_RADIO4	int	m_tc

```

int cx,cy,c=0,r;
void Cpr1Dlg::OnLButtonUp(UINT nFlags, CPoint point)
{
    if (c==0)
    {
        cx=point.x;
        cy=point.y;
        c++;
    }
    else
    {
        {CClientDC g(this);

```



```

CPen t; CBrush f;
UpdateData(true);
switch (m_tc)
{
case 0: t.CreatePen(PS_DOT,2,RGB(255,0,0));break;
case 1: t.CreatePen(PS_DOT,2,RGB(0,0,255));break;
case 2: t.CreatePen(PS_DOT,2,RGB(0,255,0));break;
case 3: t.CreatePen(PS_DOT,2,RGB(255,255,0));break;
}
f.CreateSolidBrush(RGB(240, 240, 240));
g.SelectObject(t);g.SelectObject(f);
r=sqrt(pow(point.x-cx,2.0)+pow(point.y-cy,2.0));
switch (m_tt)
{
case 0: g.MoveTo(cx,cy);
        g.LineTo(point.x,point.y);break;
case 1: g.Rectangle(cx,cy,point.x,point.y);break;
case 2: g.Ellipse(cx-r,cy-r,cx+r,cy+r);break;
}
c=0;
}
CDialogEx::OnLButtonUp(nFlags, point);
}

```

Ejemplo 4: Clases y gráficos

En el archivo .h

```

class Cgrafico
{public:
    int x1,y1,x2,y2;
    void setear(int vx1,int vy1,int vx2,int vy2);
};

```

```

class cuadro1Dlg : public CDialogEx
{
    DECLARE_DYNAMIC(cuadro1Dlg)

```

```

public:
    cuadro1Dlg(CWnd* pParent = NULL);    // Constructor estándar
    virtual ~cuadro1Dlg();
    void graficar(Cgrafico g);

```

En el archivo .cpp

```

void cuadro1Dlg::graficar(Cgrafico g)
{CPen cp;float y;
    CClientDC p(this);

```

```
cp.CreatePen(PS_SOLID,2,RGB(0,0,255));
p.SelectObject(cp);
p.Rectangle(g.x1,g.y1,g.x2,g.y2);

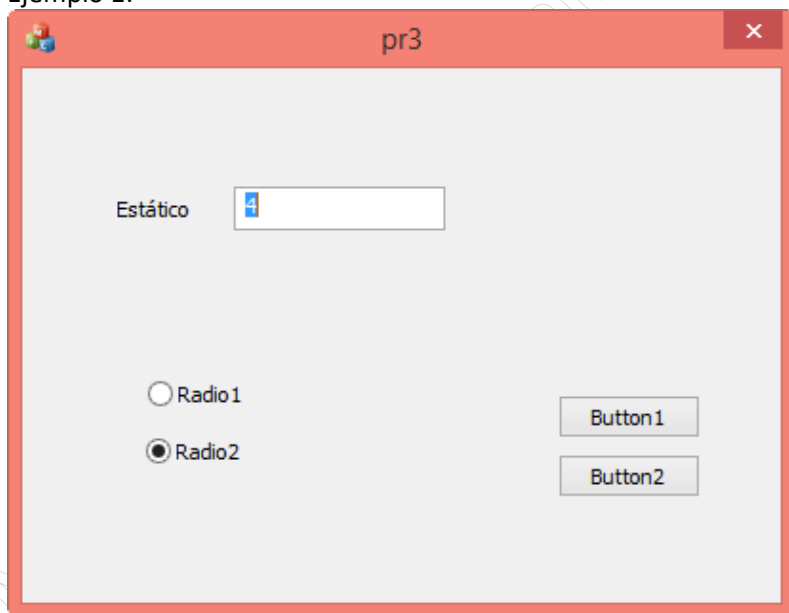
for(int x=0;x<=(g.x2-g.x1);x++)
{
    y=-(g.y2-g.y1)*sin(x*4*3.1415/(g.x2-g.x1))/2.0;
    if (x==0)
        p.MoveTo(g.x1+x,y+(g.y1+g.y2)/2);
    else
        p.LineTo(g.x1+x,y+(g.y1+g.y2)/2);
}
}

void Cgrafico::setear(int vx1,int vy1,int vx2,int vy2)
{
    x1=vx1;y1=vy1;x2=vx2;y2=vy2;
}

void cuadro1Dlg::OnBnClickedButton1()
{
    Cgrafico g1,g2;
    g1.setear(0,0,200,200); graficar(g1);
    g2.setear(250,0,450,200); graficar(g2);
}
```

TIMER

Ejemplo 1:



```
void Cprog2Dlg::OnBnClickedButton1()
{
    SetTimer(1,300,NULL);
    SetTimer(2,1000,NULL);
}

void Cprog2Dlg::OnTimer(UINT_PTR nIDEvent)
{
    UpdateData(true);
    if (nIDEvent==(m_op+1))
        m_n++;
    UpdateData(false);
}

void Cpr3Dlg::OnBnClickedButton2()
{
    KillTimer(1);
    KillTimer(2);
}
```

Ejemplo 2: Usando un Timer

```
void Cq1Dlg::OnBnClickedButton1()
```

99

```
{ SetTimer(1,10,NULL);
    // TODO: Agregue aquí su código de controlador de
    notificación de control
}

int r=150;
float a=0,inc=0.5;
void Cq1Dlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Agregue aquí su código de controlador de mensajes
    o llame al valor predeterminado
    if (nIDEvent==1)
    {
        CClientDC p(this);
        CPen t1,t2;
        int x0=200,y0=200;
        int dx=r*cos(a*3.14/180);
        int dy=-r*sin(a*3.14/180);
        int x2=x0+dx; int y2=y0+dy;
        dx=r*cos((a-inc)*3.14/180);
        dy=-r*sin((a-inc)*3.14/180);
        int x1=x0+dx; int y1=y0+dy;

        t1.CreatePen(PS_DOT,4,RGB(240,240,240));
        p.SelectObject(t1);
        p.MoveTo(x0,y0);
        p.LineTo(x1,y1);

        t2.CreatePen(PS_DOT,4,RGB(255,0,0));
        p.SelectObject(t2);
        p.MoveTo(x0,y0);
        p.LineTo(x2,y2);
        a=a+inc;
    }
    CDialogEx::OnTimer(nIDEvent);
}
```

CAPTURA DE TECLA EN UNA CAJA

Agregar manualmente esta función miembro, en el .cpp y .h

```

BOOL CcaptureteclaDlg::PreTranslateMessage(MSG* pMsg)
{
    if(GetFocus() && GetFocus()->GetDlgCtrlID()==IDC_EDIT1)
    {
        if (pMsg->message==WM_KEYDOWN && pMsg->wParam==VK_RETURN)
        {
            AfxMessageBox(_T("Se presiono la tecla ENTER"));
            //return true;
        }
    }
    return CDialog::PreTranslateMessage(pMsg);
}

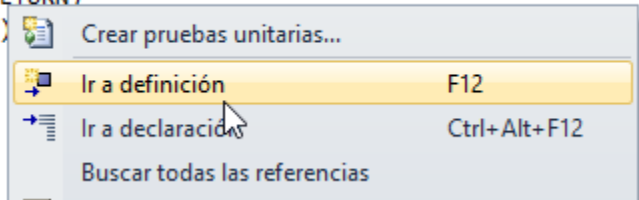
```

Para ver las otras constantes hacer click derecho sobre la constante e ir a definición

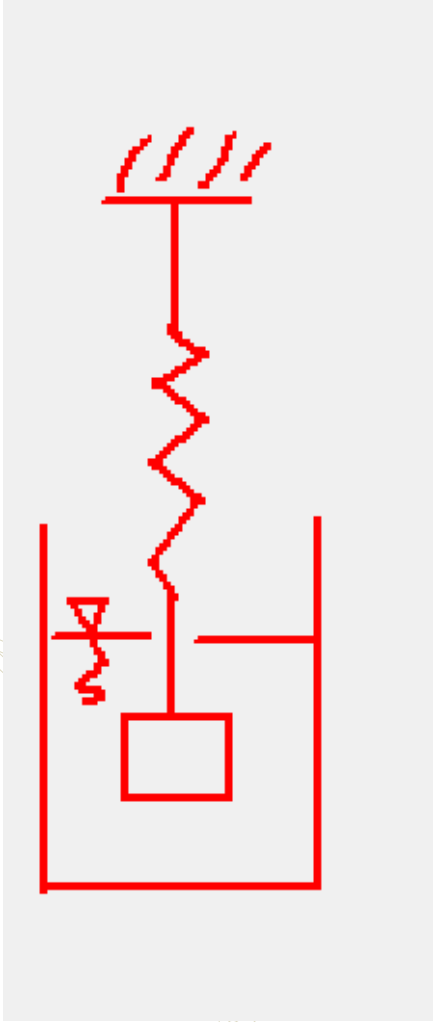
```

lgCtrlID()==IDC_EDIT1)
pMsg->wParam==VK_RETURN)
ono la tecla ENTER")
e(pMsg);

```



Ejercicio: Realice la animación de este mecanismo.



ANEXO

Como instalar MFC si no lo instalo desde un inicio

- 1) Abrir el Visual Studio Installer y seleccionar la opción Modificar