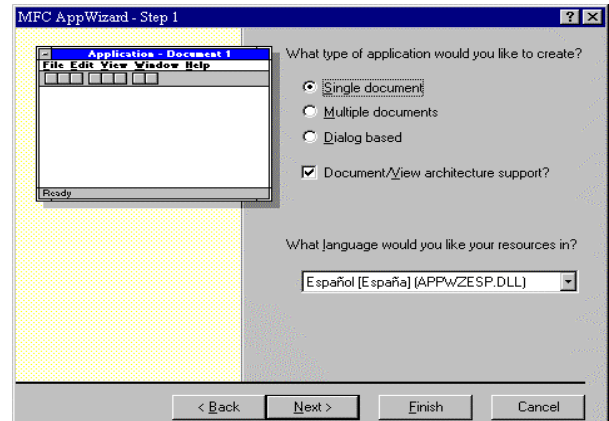
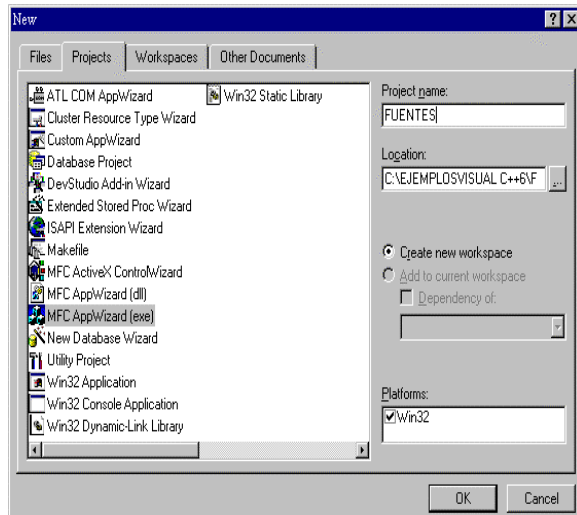


LABORATORIO: N°4

OBJETIVO: El aprendizaje y dominio de todas las posibilidades de las fuentes ofrecidas por Windows
Desarrollar aplicaciones para ilustrar diversas técnicas de manipulación de fuentes.

PARTE I**PASO 1**

Comience por seleccionar **File|New** desde el menú principal de Visual C++,
En **Projects** elija **AppWizard**
Coloque título al **Proyecto** y seleccione los pasos que se muestran



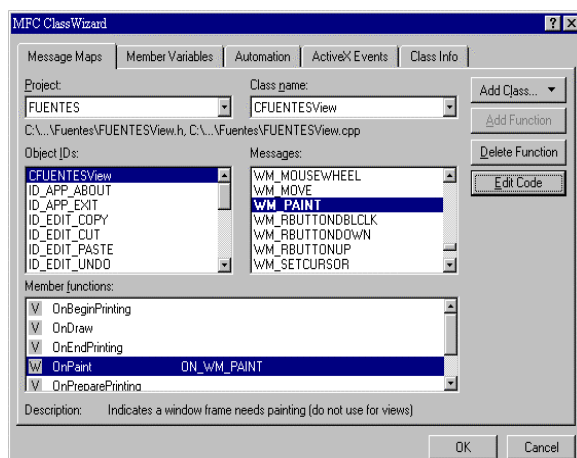
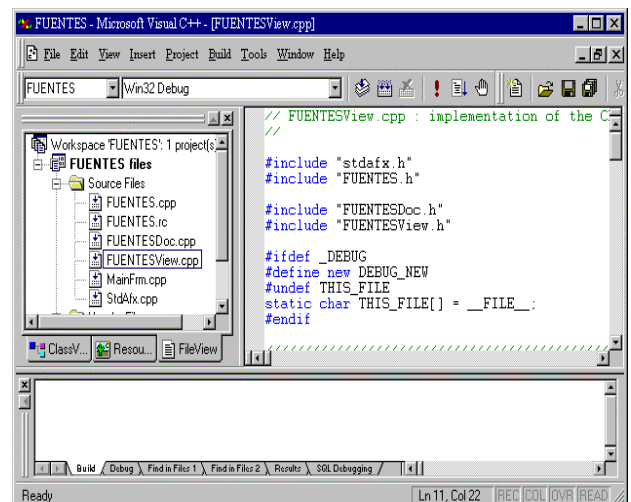
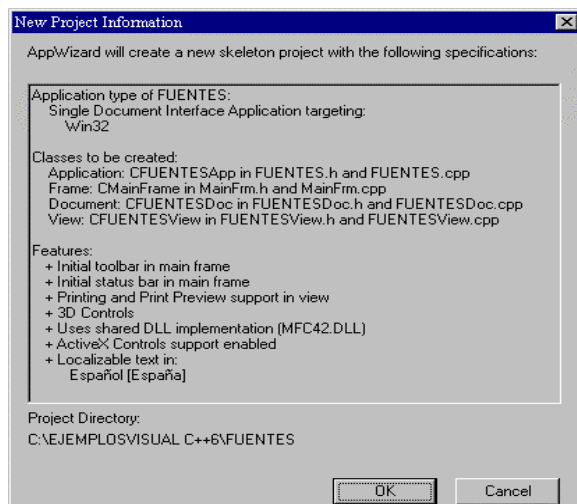
Esto dará lugar a una aplicación con una interfaz que permitirá al usuario abrir un solo documento cada vez.

SDI:

S=Single

D=Document

I=Interface



La clase **CPaintDC** se ha diseñado para utilizarla con la función **OnPaint**.

Para visualizar texto o gráficos en el área de trabajo de una ventana en respuesta a un mensaje **WM_PAINT**(se ejecuta la función **OnPaint**), hay que crear un objeto **DC** perteneciente a la clase **CPaintDC** o una clase derivada de ella.

- Presione **Edit Code** y continúe.

PASO 2

Digite el siguiente código.
Deberá obtener un resultado tal como se muestra en la gráfica.

```
void CFUENTESView::OnPaint()
{
    CPaintDC dc(this); // device context for painting

    HFONT hNFont; // crea un objeto
    int nHeight, nWidth;

    static char szTextString[]="InfoPUC.";
    static char szIntString[]="Instituto de informatica Pontificia Universidad Catolica del Peru.";
    static char szVCString[]="VISUAL C++ 6.";

    //Imprime texto InfoPUC
    nHeight=100;
    nWidth=40;
    hNFont=CreateFont(nHeight,nWidth,0,0,FW_BOLD, FALSE,FALSE,FALSE,OEM_CHARSET,
                      OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
                      VARIABLE_PITCH|FF_ROMAN,"Roman");

    dc.SelectObject(hNFont);
    dc.TextOut(0,30,szTextString,strlen(szTextString));

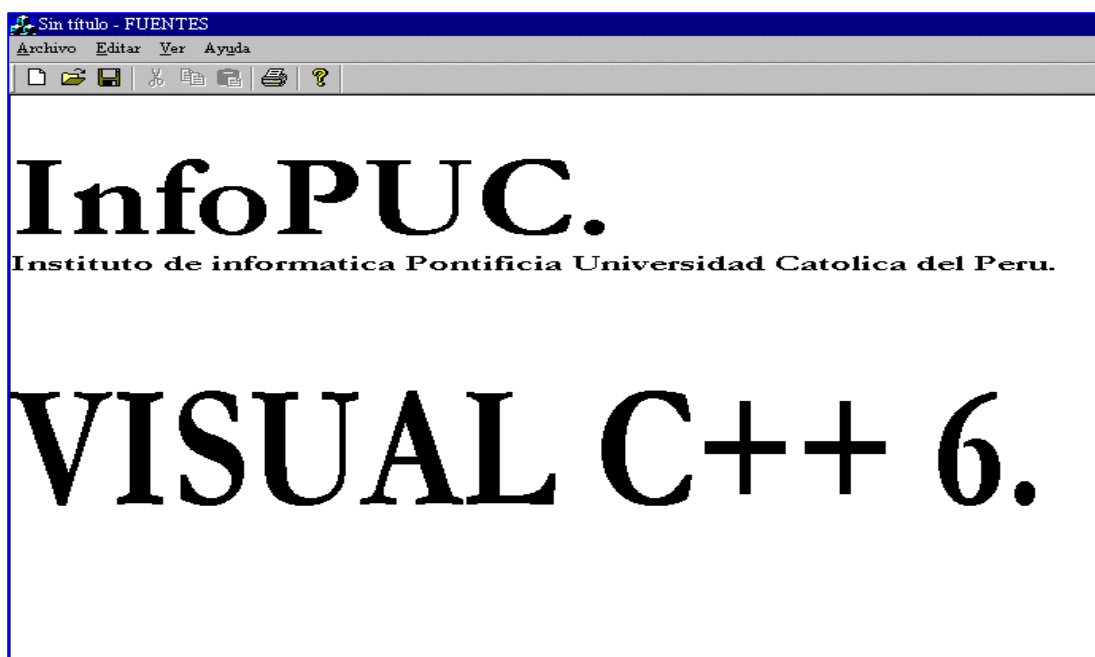
    //Imprime Texto Instituto de informatica
    //Pontificia Universidad Catolica del Peru

    nHeight=20;
    nWidth=10;
    hNFont=CreateFont(nHeight,nWidth,0,0,FW_BOLD, FALSE,FALSE,FALSE,OEM_CHARSET,
                      OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
                      VARIABLE_PITCH|FF_ROMAN,"Roman");

    dc.SelectObject(hNFont);
    dc.TextOut(0,120,szIntString,strlen(szIntString));

    // Imprime Texto VISUAL C++ 6
    nHeight=150;
    nWidth=40;
    hNFont=CreateFont(nHeight,nWidth,0,0,FW_BOLD, FALSE,FALSE,FALSE,OEM_CHARSET,
                      OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
                      VARIABLE_PITCH|FF_ROMAN,"Roman");

    dc.SelectObject(hNFont);
    dc.TextOut(0,200,szVCString,strlen(szVCString));
}
```



PARTE II**PASO 1 :**

Borrar el código anterior y Digite el siguiente código.
Deberá obtener un resultado tal como se muestra en la gráfica.

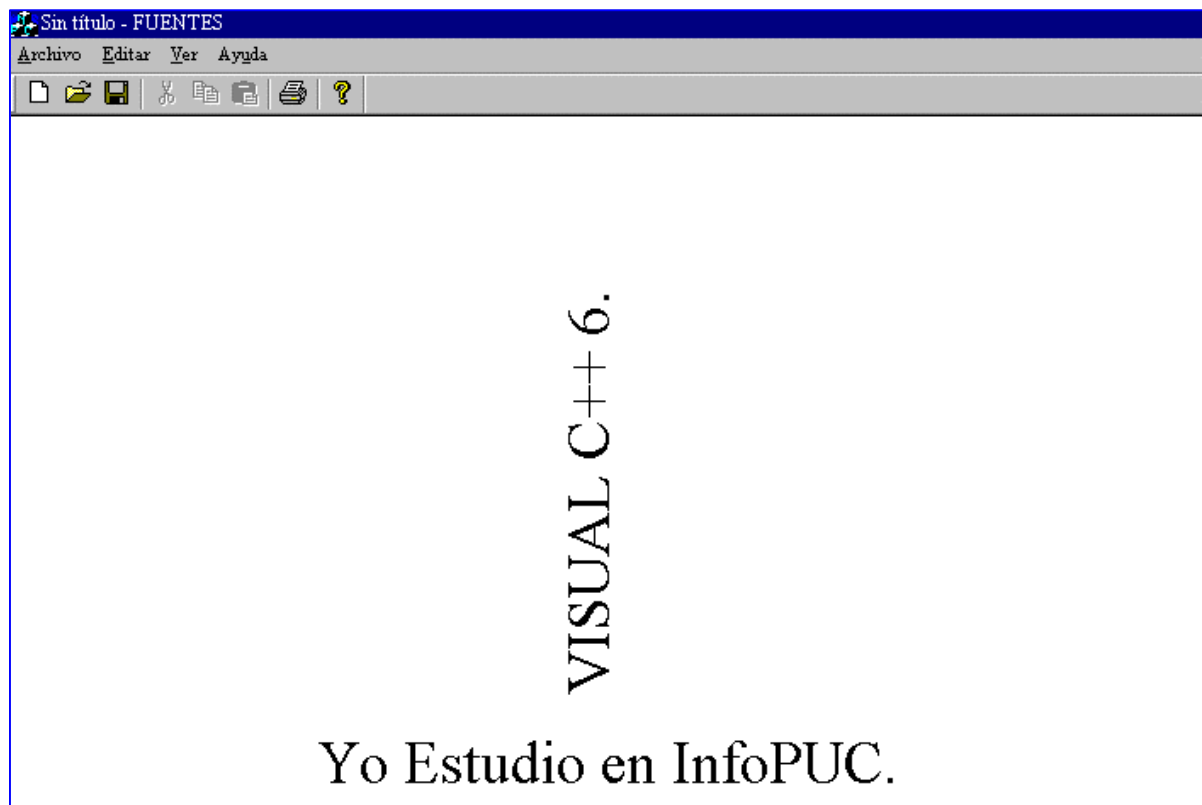
```
void CFUENTESView::OnPaint()
{
    CPaintDC dc(this); // device context for painting

    HFONT hNFont;// crea un objeto
    static char szXString[]="Yo Estudio en InfoPUC.";
    static char szYString[]="VISUAL C++ 6.";
    //Imprime texto horizontal en la pantalla
    //con la fuente Courier New de TrueType
    hNFont=CreateFont(40,0,0,0,FW_NORMAL, FALSE,FALSE,FALSE,DEFAULT_CHARSET,
        OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,49,
        "Times New Roman");

    dc.SelectObject(hNFont);
    dc.TextOut(180,360,szXString,strlen(szXString));

    //Imprime Texto Vertical en la pantalla
    //con la fuente Times New Roman de TrueType
    hNFont=CreateFont(40,0,900,900,FW_NORMAL, FALSE,FALSE,FALSE,DEFAULT_CHARSET,
        OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,18,
        "Times New Roman");

    dc.SelectObject(hNFont);
    dc.TextOut(320,340,szYString,strlen(szYString));
}
```



PASO 2

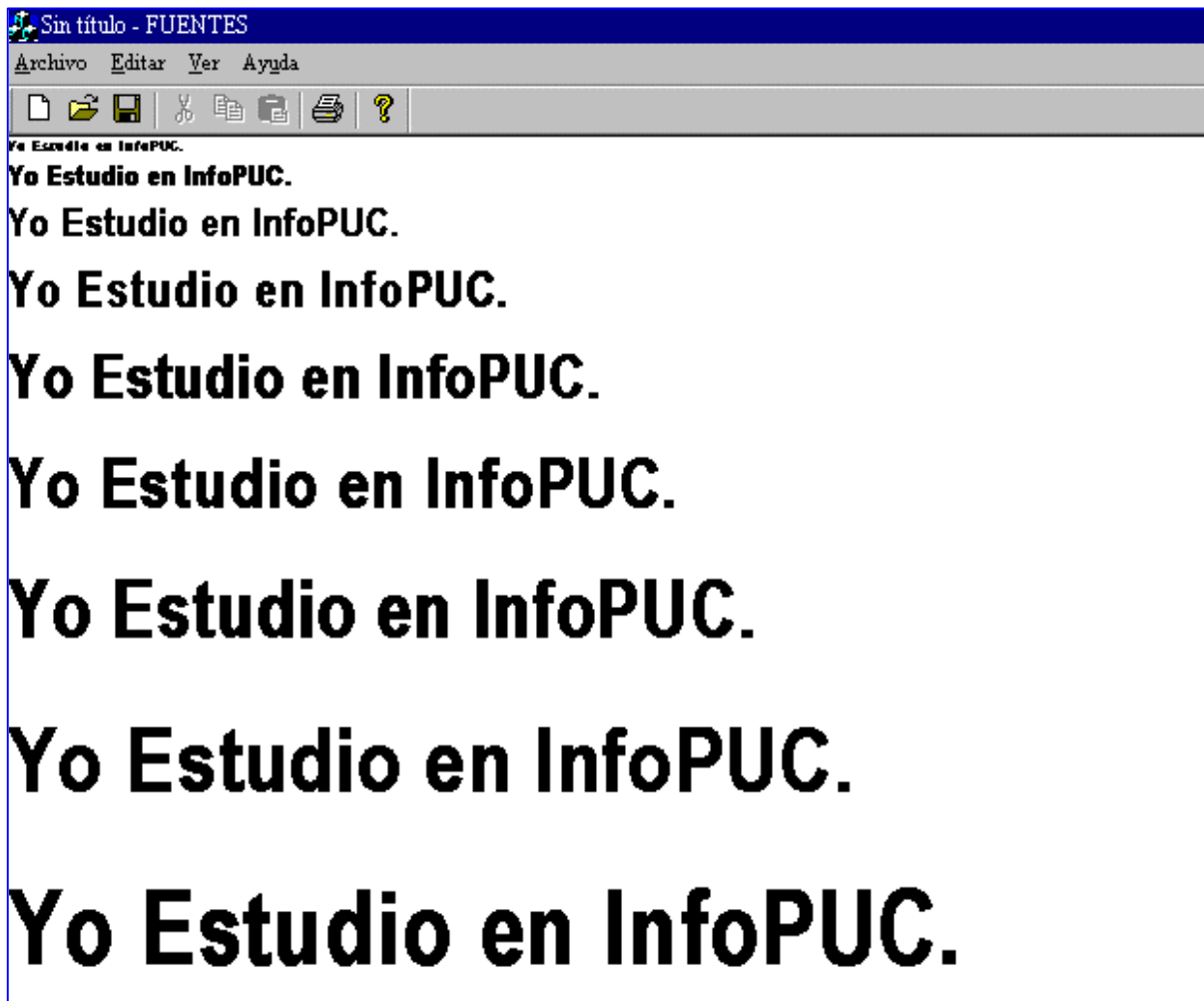
Borrar el código anterior y digite el siguiente código.
Deberá obtener un resultado tal como se muestra en la gráfica.

```
void CFUENTESView::OnPaint()
{
    CPaintDC dc(this); // device context for painting

    HFONT hNFont; // crea un objeto
    static LOGFONT lf;
    static char szXString[]="Yo Estudio en InfoPUC.";
    int i, ypos;

    lf.lfWeight=FW_HEAVY; //grado de negrita (400 normal,700 negrita)
    lf.lfCharSet=OEM_CHARSET; //juego de caracteres del tipo de letra
    lf.lfPitchAndFamily=FF_SCRIPT; //ancho y familia del tipo de letra
    ypos=0;

    for (i=1;i<10;i++){
        lf.lfHeight=6+(6*i);
        hNFont=CreateFontIndirect(&lf);
        dc.SelectObject(hNFont);
        dc.TextOut(0,ypos,szXString,strlen(szXString));
        ypos+=10*i;
    }
    DeleteObject(hNFont);
}
```



PASO 3

Borrar el código anterior y Digite el siguiente código.
Deberá obtener un resultado tal como se muestra en la gráfica.

```
void CFUENTESView::OnPaint()
{
    CPaintDC dc(this); // device context for painting

    HFONT hNFont; // crea un objeto
    static LOGFONT lf;
    static char szXString[]="VISUAL C++ 6 !QUE FACIL!";

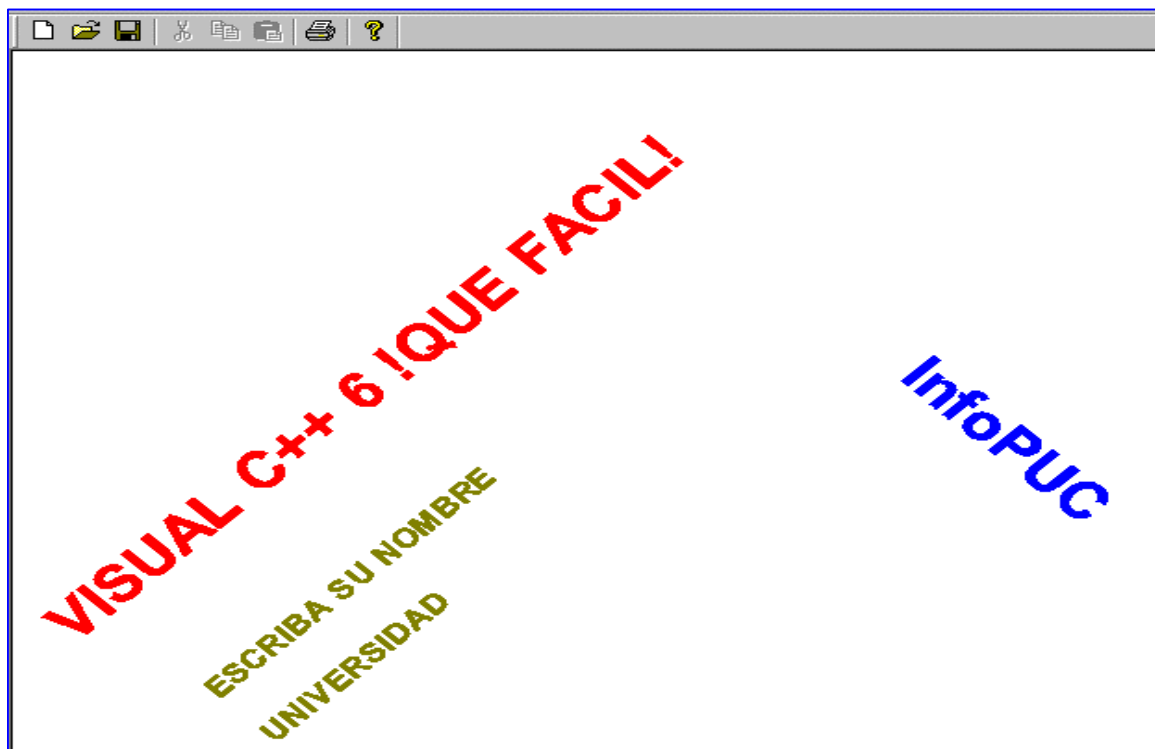
    lf.lfWeight=FW_HEAVY; //grado de negrita (400 normal,700 negrita)
    lf.lfCharSet=DEFAULT_CHARSET; //juego de caracteres del tipo de letra
    lf.lfPitchAndFamily=34; //ancho y familia del tipo de letra
    lf.lfEscapement=450; //angulo de rotación en decimas de grados
    lf.lfHeight=40; //altura
    dc.SetTextColor(RGB(255,0,0));
    hNFont=CreateFontIndirect(&lf);
    dc.SelectObject(hNFont);
    dc.TextOut(10,380,szXString,strlen(szXString));

    dc.SetTextColor(RGB(130,130,0));
    lf.lfHeight=20; //altura
    hNFont=CreateFontIndirect(&lf);
    dc.SelectObject(hNFont);
    dc.TextOut(100,430,"ESCRIBA SU NOMBRE");
    dc.TextOut(130,460,"UNIVERSIDAD");

    dc.SetTextColor(RGB(0,0,255));
    lf.lfHeight=40; //altura
    lf.lfEscapement=-450; //angulo de rotación en decimas de grados

    hNFont=CreateFontIndirect(&lf);
    dc.SelectObject(hNFont);
    dc.TextOut(500,200,"InfoPUC");

    DeleteObject(hNFont);
}
```



PARTE III**PASO 1**

Borrar el código anterior y Digite el siguiente código.
Debera obtener un resultado tal como se muestra en la grafica.

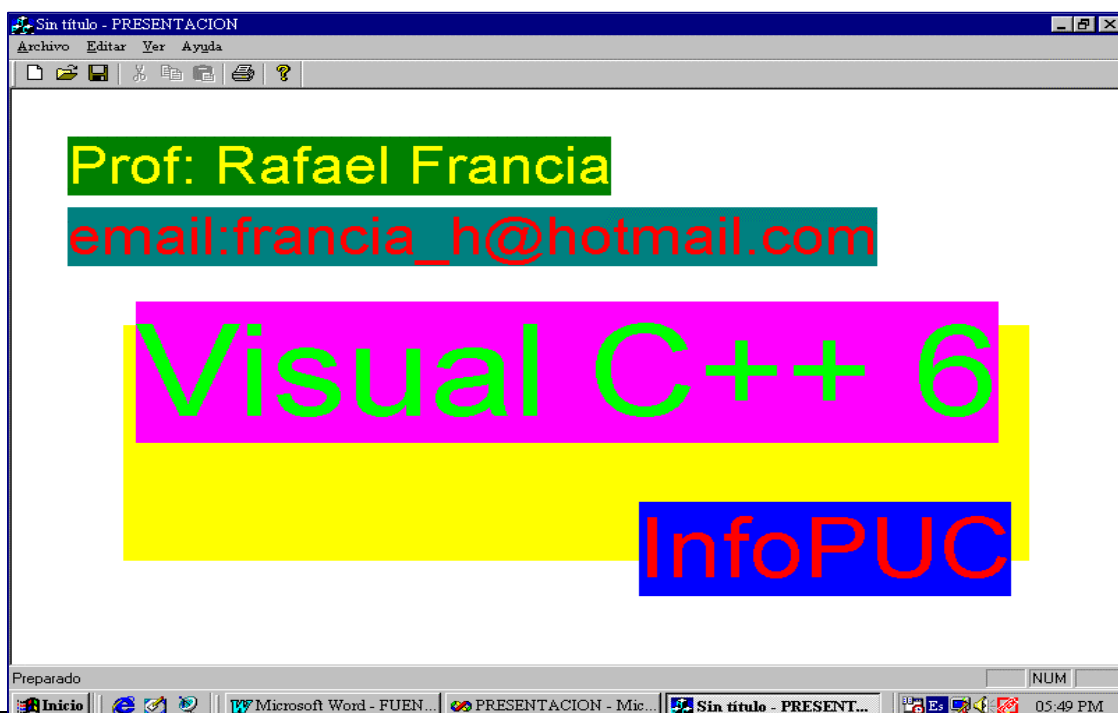
```
void CPRESENTACIONView::OnDraw(CDC* pDC)
{
    CPRESENTACIONDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here

    static LOGFONT lf;
    HFONT hFont;
    lf.lfHeight=50;
    hFont=CreateFontIndirect(&lf);
    pDC->SelectObject(hFont);
    pDC->FillSolidRect(80,200,650,200,RGB(255,255,0));
    //coloca el texto "Prof: Rafael Francia"
    pDC->SetTextColor(RGB(255,255,0)); //color de texto
    pDC->SetBkColor(RGB(0,128,0));
    pDC->TextOut(40,40,"Prof: Rafael Francia");

    //coloca el texto "email:francia_h@hotmail.com"
    pDC->SetBkColor(RGB(0,128,128));
    pDC->SetTextColor(RGB(255,0,0)); // color texto
    pDC->TextOut(40,100,"email:francia_h@hotmail.com");

    //coloca el texto "Visual C++ 6"
    lf.lfHeight=120;
    hFont=CreateFontIndirect(&lf);
    pDC->SelectObject(hFont);
    pDC->SetBkColor(RGB(255,0,255));
    pDC->SetTextColor(RGB(0,255,0));
    pDC->TextOut(90,180,"Visual C++ 6");

    // coloca el texto InfoPUC
    pDC->SetBkColor(RGB(0,0,255));
    lf.lfHeight=80;
    hFont=CreateFontIndirect(&lf);
    pDC->SelectObject(hFont);
    pDC->SetTextColor(RGB(255,0,0));
    pDC->TextOut(450,350,"InfoPUC");
    pDC->SetBkColor(RGB(255,0,255));
    DeleteObject(hFont);
}
```



OBSERVACIONES:**ESTABLECER UNA FUENTE**

- 1.- La GDI(Graphics Device Interface) de Windows emplea la funcionalidad de la clase CFont (derivada de CGdiObject) para manipular las fuentes de caracteres.
Para utilizar una determinada fuente de Windows, construya un objeto CFont y utilizando la función miembro CreateFont o CreateFontIndirect, ligue la fuente al objeto.

La función **CreateFont** tiene 14 parametros, que se corresponden directamente con los miembros de una estructura de tipo **LOGFONT**, que definen las características de un tipo de letra.

Por esto, es más cómodo crear un tipo de letra utilizando la estructura **LOGFONT** y la función CreateFontIndirect:

```
CFont m_Fuente; //objeto
LOGFONT lf
// establecer las características del tipo de letra
m_Fuente.CreateFontIndirect(&lf);
```

Después de crear un tipo de letra virtual, lo seleccionamos para el contexto de dispositivo con **SelectObject** o directamente para el objeto con SetFont. Cuando el tipo de letra no se utilice se debe destruir llamando a la función **DeleteObject**.

```
m_Fuente.DeleteObject();
```

- 2.- La función **CreateFont** es del tipo **HFONT**, tal como se define en el archivo cabecera **WINDOWS.H**
La sintaxis de **CreateFont** es:

HFONT CreateFont(Altura, Anchura, Inclinación, Orientación, Intensidad, Cursiva, Subrayado, Tachadura, JuegoCaracteres, PrecisionSalida, PrecisionClip, Calidad, PasoYFamilia, NombreTipo)

Con 14 parámetros, CreateFontF lleva bastante equipaje

(byte) PasoYFamilia	Familia y clase de espaciado de la fuente(DEFAULT_PITCH, FIXED_PITCH,.....FF_DECORATIVE, FF_DONTCARE, FF_MODERN, FF_ROMAN, FF_SCRIPT, FF_SWISS)
---------------------	---