



INTRODUCCIÓN

Para poder trabajar con archivos en formularios windows C# cuenta con dos objetos que vienen ya diseñados el **OpenFileDialog** (para abrir) y el **SaveFileDialog** (para guardar).

OpenFileDialog

El componente OpenFileDialog permite a los usuarios examinar las carpetas de su equipo o de cualquier equipo de la red y seleccionar uno o más archivos para abrirlos. El cuadro de diálogo devuelve la ruta de acceso y el nombre del archivo que seleccionó el usuario en el cuadro de diálogo.

Una vez que el usuario ha seleccionado el archivo que desea abrir, hay dos maneras de abrirlo. Si prefiere trabajar con secuencias de archivo, puede crear una instancia de la clase StreamReader. Como alternativa, puede utilizar el método OpenFile para abrir el archivo seleccionado.

Principales propiedades

openFileDialog1.Filter = "JPeg Image|*.jpg|Bitmap Image|*.bmp|Gif Image|*.gif"; //Tipos de archivos que se pueden abrir

openFileDialog1.Title = "Abrir archivo..."; Título que aparecerá en el cuadro de diálogo

openFileDialog1.ShowDialog(); //Mostrar cuadro de diálogo

Método después de seleccionar un archivo para abrir

Hay dos formas de saber si se ha seleccionado un archivo de un openFileDialog: Por medio de su método openFileDialog1_FileOk(object sender, CancelEventArgs e)

```
private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
    //Se abrió archivo y podemos saber su ruta y nombre a través de
    openFileDialog1.FileName;
    Nombre_archivo_abierto = openFileDialog1.FileName;
}
```

O bien desde otro objeto como un botón:

```
private void button1_Click(object sender, System.EventArgs e)
{
    if(openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        System.IO.StreamReader sr = new
        System.IO.StreamReader(openFileDialog1.FileName);
        MessageBox.Show(sr.ReadToEnd());
        sr.Close();
    }
}
```



SaveFileDialog

El componente SaveFileDialog permite a los usuarios examinar el sistema de archivos y seleccionar los archivos que deseen guardar. El cuadro de diálogo devuelve la ruta de acceso y el nombre del archivo que seleccionó el usuario en el cuadro de diálogo. **Sin embargo, debe implementar el código para escribir realmente los archivos en el disco.**

Principales propiedades

`saveFileDialog1.Filter` = "JPEG Image|.jpg|Bitmap Image|.bmp|Gif Image|.gif"; //Tipos de archivos como puede guardarse

`saveFileDialog1.Title` = "Salvar archive como..."; Título que aparecerá en el cuadro de diálogo

`saveFileDialog1.ShowDialog();` //Mostrar cuadro de diálogo

Método después de seleccionar un archivo para guardar

```
private void saveFileDialog1_FileOk(object sender, CancelEventArgs e)
{
    if (saveFileDialog1.CheckFileExists == false)
    {
        Nombre_archivo_para_guardar = saveFileDialog1.FileName;
        //Proceso para guardar archivo Por ejemplo con StreamWriter
    }
}
```

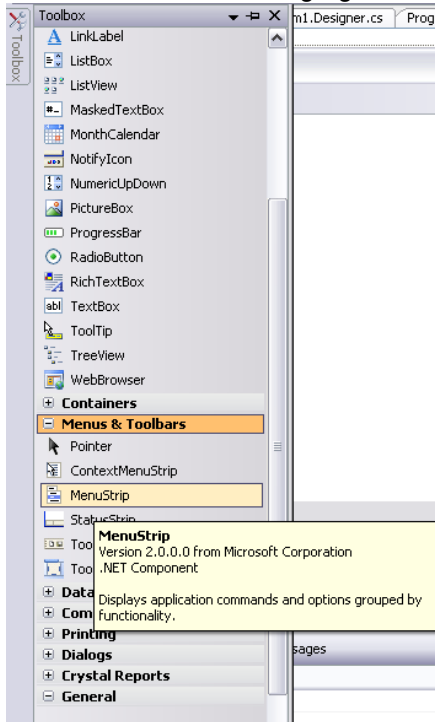


Ejercicios de la Práctica

1. Crea un nuevo proyecto seleccionando la opción **Windows Application**. Llámala a la solución Practica21_bloc_notas, y el nombre del proyecto **bloc_notas**

En el formulario adiciona los siguientes controles.

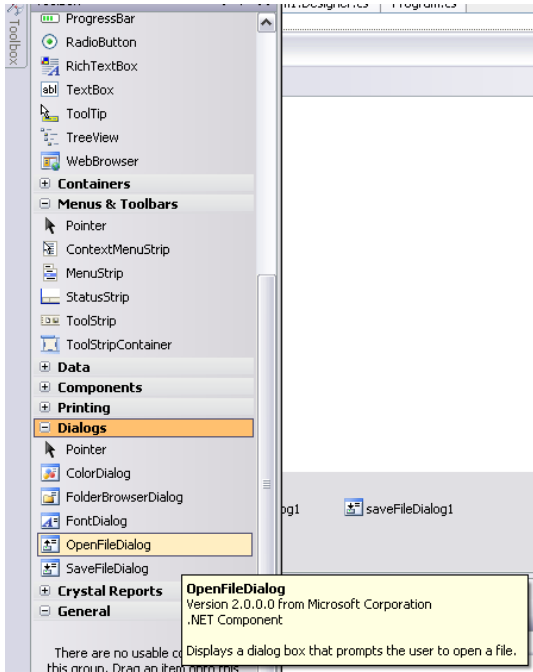
De **Menús & ToolBars** agrega un menuStrip



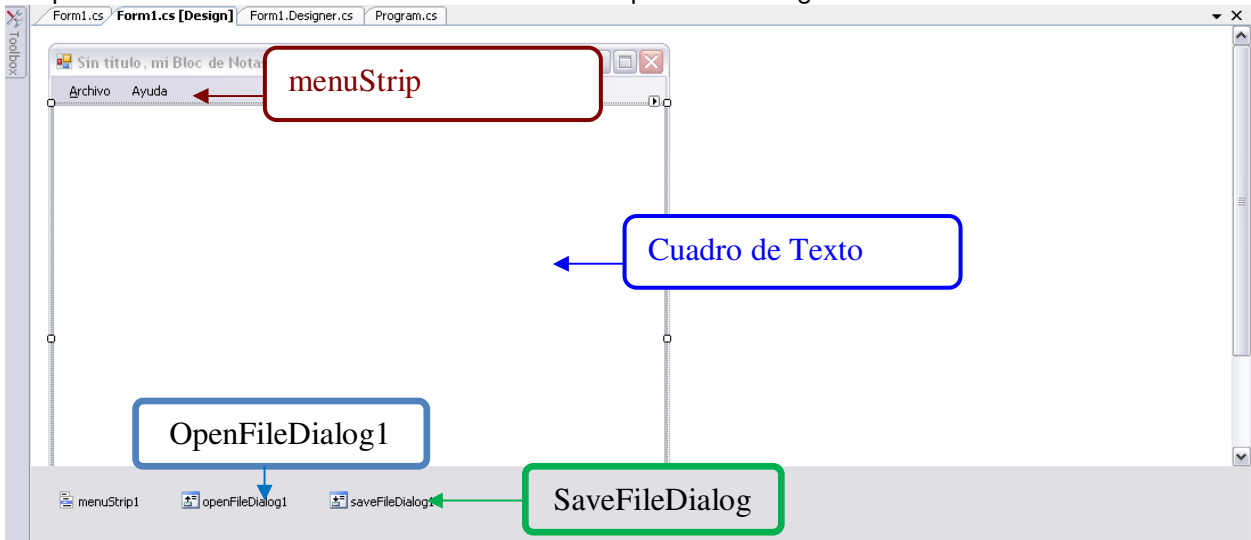
De **Dialogs** agrega OpenFileDialog y SaveFileDialog



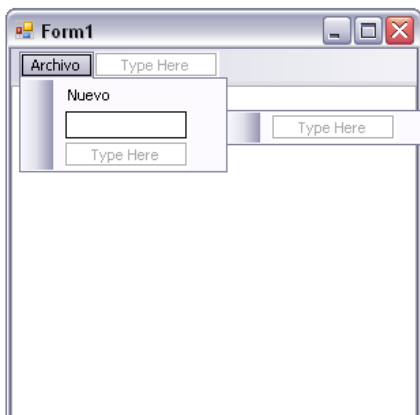
TEMA4.PROGRAMACIÓN ORIENTADA A OBJETOS



Finalmente Agrega un Cuadro de Texto (TextBox) y cámbiale su propiedad **MultiLine= True**. (Al cambiar esta propiedad usted podrá redimensionar a lo largo y a lo alto su cuadro de texto)
La apariencia Final de su ventana de diseño deberá quedar de la siguiente manera:

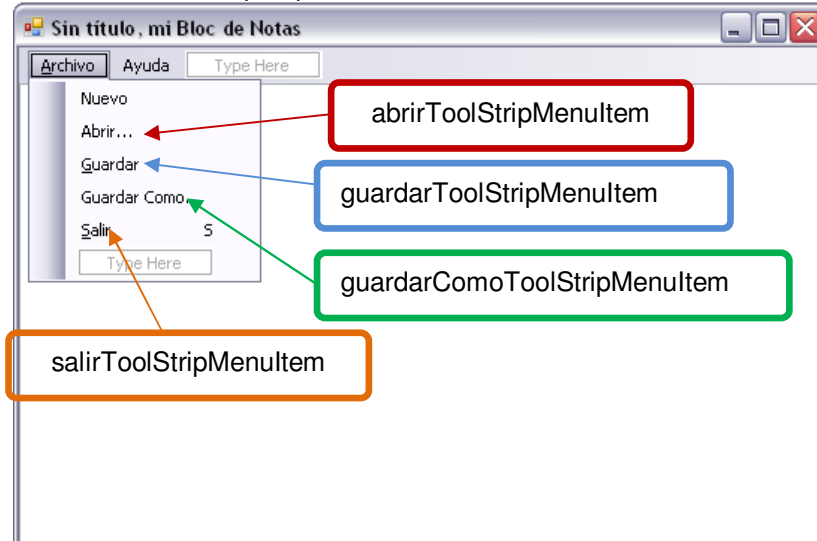


Para el menuStrip Pulsando Directamente sobre él para poder editarlo.





Edítelo de manera que quede así:



Posteriormente vamos a ir al Form1.cs y vamos a escribir el siguiente código:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace bloc_notas
{
    public partial class Form1 : Form
    {
        StreamReader lector_de_archivo;
        StreamWriter escritor_de_archivo;
        string miArchivoAbierto="";
        public Form1()
        {
            InitializeComponent();
        }

        private void abrirToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //Comando para abrir un cuadro de dialogo
            openFileDialog1.Title="Abrir archivo de texto";
            openFileDialog1.FileName = "";
            //openFileDialog1.Filter="*.txt|*.txt|*.jpg|*.jpg" indica que tipo de archivos podemos abrir
            openFileDialog1.Filter="*.txt|*.txt";
            openFileDialog1.ShowDialog();
        }

        private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
        {
            //En caso de que se haya seleccionado un archivo abrimos una instancia
        }
    }
}
```



TEMA4.PROGRAMACIÓN ORIENTADA A OBJETOS

```
//de la clase 1
miArchivoAbierto = openFileDialog1.FileName;
abrirArchivo(miArchivoAbierto);
string miCadena = lector_de_archivo.ReadToEnd();
try
{
    textBox1.Text = miCadena;
}
catch
{
    MessageBox.Show("Error al abrir el archivo");
}
finally
{
    cerrarArchivo();
}
}
cerrarArchivo();
}

private void salirToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}

private void nuevoToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.textBox1.Clear();
    miArchivoAbierto = "";
    this.Text = "Sin título, mi Bloc de Notas";
}

private void guardarToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (miArchivoAbierto == "")
    {
        saveFileDialog1.Title = "Guardar archivo...";
        saveFileDialog1.Filter = "*.txt|*.txt";
        saveFileDialog1.ShowDialog();
    }
    else
    {
        abrirArchivo(miArchivoAbierto, true);
        escritor_de_archivo.Write(textBox1.Text);
        cerrarArchivo(true);
    }
}

public void abrirArchivo(string nombre_archivo)
{
    this.Text = nombre_archivo;
    lector_de_archivo = new StreamReader(nombre_archivo);
}

public void abrirArchivo(string nombre_archivo, bool escribir)
{
    if (nombre_archivo != "")
    {
        try
        {
            this.Text = nombre_archivo;
            escritor_de_archivo = new StreamWriter(nombre_archivo);
        }
    }
}
```



TEMA4.PROGRAMACIÓN ORIENTADA A OBJETOS

```
        catch (ArgumentNullException)
        {
            MessageBox.Show("No ha seleccionado un archivo");
        }
    }
    else
    {
        saveFileDialog1.Title = "Guardar archivo como...";
        saveFileDialog1.Filter = "*.txt|.txt";
        saveFileDialog1.ShowDialog();
    }
}

public void cerrarArchivo()
{
    if(lector_de_archivo!=null)
        lector_de_archivo.Close();
}

public void cerrarArchivo(bool escribir)
{
    if (escritor_de_archivo != null)
        escritor_de_archivo.Close();
}

private void saveFileDialog1_FileOk(object sender, CancelEventArgs e)
{
    if (saveFileDialog1.CheckFileExists == false)
    {
        miArchivoAbierto = saveFileDialog1.FileName;
        this.Text = miArchivoAbierto;
        abrirArchivo(miArchivoAbierto, true);
        escritor_de_archivo.Write(textBox1.Text);
        cerrarArchivo(true);
    }
}

private void guardarComoToolStripMenuItem_Click(object sender, EventArgs e)
{
    saveFileDialog1.Title = "Guardar archivo como...";
    saveFileDialog1.Filter = "*.txt|.txt";
    saveFileDialog1.ShowDialog();
}
}
```

Analiza cada una de las partes del código y coloca tus comentarios adicionales, completa el código si es necesario.