

# Clase 4 - No Presencial. Por Guillermo Guastavino

## Manejo de Formularios y menús

Hola a todos!, Llenándonos de ganas y voluntad, a pesar del casi encierro y la Pandemia, llegamos a la 4ta clase no presencial.

Para que siga siendo lo “más semejante” posible a una clase presencial, les sigo escribiendo la clase casi en “vivo”, empiezo el día anterior, temprano hasta la noche... como si se las estuviera dando, rellenamos con un poco de imaginación.

Como les adelanté la clase pasada, hoy empezamos al revés, y vamos a seguir así para los dos cursos que tienen conmigo. La clase ahora la hago en VB, que ya demostramos que es más simple, nos dá margen a enfocarnos en lo que importa y no en el lenguaje, aprenden otro etc., y ahora ustedes arman la clase en C#. Me preguntan lo que necesitan en el foro, pegan código, o adjuntan etc. canalicen las dudas por allí.

Les estuve dando lo que vamos necesitando, y ahora vamos a necesitar cosas que nos acerquen al sistema de facturación con base de datos SQL. Así que los objetos y teoría que veamos son para cubrir esa necesidad. Investiguen en internet, sean curiosos sobre otros objetos y clases, y empiecen desde hoy a armar un sistema en sus casas, más grande y mejor que el que vamos haciendo, es TU sistema, hecho por VOS, y es lo que más vale, y lo que aprendas así superará cualquier cosa que puedas aprender en un curso. Ciento ochenta mil clases te esperan...

Hoy empezamos con el manejo de formularios y un menú (y un desafío para guerreros o magos al final).




En Visual Studio, creamos un proyecto nuevo, en VB (y repetimos... vos lo hacés también en C#...).

Vas a armar un proyecto nuevo, exactamente con el mismo camino de plantillas de antes, pero VB (que ya hiciste varios).

# Visual Studio 2019

## Abrir recientes

### Esta semana

-  **ConsumoRed\_F45.sln** 14/04/2020 01:33 p.m.  
K:\Guillermo\Sistemas\Fideliti\ConsumoRed\ConsumoRed\_F45
-  **ConsumoRed\_sin\_reportes NET 2019** 14/04/2020 01:17 p.m.  
K:\Guillermo\Sistemas\Fideliti\ConsumoRed
-  **Navegador.sln** 10/04/2020 05:45 p.m.  
K:\Guillermo\Notas, Presupuestos y CV\Universidad\Navegador

## Tareas iniciales



### Clonar o extraer código del repositorio

Obtiene código desde un repositorio en línea, como GitHub o Azure DevOps.



### Abrir un proyecto o una solución

Abre un archivo .sln o proyecto de Visual Studio local.



### Abrir una carpeta local

Navegar y editar el código en cualquier carpeta



### Crear un proyecto

Elija una plantilla de proyecto con la técnica scaffolding de código para comenzar.

[Continuar sin código →](#)



## Crear un proyecto

Elija una plantilla de proyecto con la técnica scaffolding de código para comenzar.



### Aplicación de Windows Forms (.NET Framework)

Proyecto para crear una aplicación con una interfaz de usuario de Windows Forms (WinForms)

Visual Basic

Windows

Escritorio

Siguiente

Eligen la ruta que quieran. Seguimos con framework 4.5 para que ande sin problemas también en W 7. El proyecto es “menú”, y no va a ser lo que vayamos a usar luego, sino un tablero de pruebas para ver cómo funciona todo. Como siempre... en C# es casi completamente igual todo. Si se pierden con algo, vean las clases anteriores, me preguntan o preguntan en el foro para que otros participen ayudando.

# Configure su nuevo proyecto

Aplicación de Windows Forms (.NET Framework)

Visual Basic

Windows


Escritorio

Nombre del proyecto

menú

Ubicación

K:\Guillermo\Notas, Presupuestos y CV\Universidad

Nombre de la solución 

menú

☒ Colocar la solución y el proyecto en el mismo directorio

Framework

.NET Framework 4.5

Crear

Empecemos cambiando algunas propiedades del Form1. La idea es que el usuario no pueda alterar el tamaño del formulario (y hacerlo más chico tanto que cosas no se vean, o más grande y vea cosas que ocultamos o tape un formulario que teníamos por detrás por ejemplo.). Tampoco queremos entonces, que pueda maximizar o minimizar, y que no salga con la X, sino con un botón que nosotros hagamos.

Si ponen ControlBox a false en tiempo de diseño, va a desaparecer directamente el cuadrado con X Max y Min. En cambio, podrían haber quitado el Max (propiedad MaximizeBox=false) o el Min (minimizeBox=false), para que quede sólo la X. Para que siempre esté maximizado, minimizado o normal (como ahora), se puede usar WindowsState.

Poné

**ControlBox=false**

**FormBorderStyle=FixedSingle** (Sizable se puede estirar, Fixed no...)

Entonces el tamaño que le den ahora, va a ser el tamaño que va a tener, e inamovible para el usuario.

Si ejecutan esto, habremos caído en nuestra propia trampa, porque no hay forma “legal” desde el programa, para cortar la ejecución (usá el cuadrado de Stop...)

Agregamos un botón llamado bSalir, que tenga text “Salir”.

Hay algo que he visto muchísimas veces, que los usuarios cuando cambian o abren un formulario nuevo que reemplaza al anterior, dicen “andá a la pantalla tal...”, y la imaginan como que está a la derecha de la otra, como pasar de un cuarto al siguiente. Entonces para volver, piensan que si fueron para la derecha, para volver es hacia la izquierda. Entonces yo pongo siempre los botones de volver (al form anterior) o salir (cerrar el proyecto), a la izquierda y abajo... justo al revés de dónde pone Windows la X.

Hablando de lo que entiende el usuario... Como les conté, todos imaginan que se mueven entre formularios como se moverían entre cuartos de un edificio. En la UNLZ, para llegar al aula de práctica, primero entro a la Universidad, luego camino unos pasos por el hall, subo 2 pisos la escalera, y entro a la segunda aula a la izquierda. Lamentablemente, no hay un “portero teletransportador” en la entrada, que me haga aparecer tipo Star Trek en el medio del aula, y que luego pueda elegir Cafetería y aparecer de golpe allí (doble esfuerzo de imaginación, porque ni a la Facu podemos ir...). Sin embargo los menús desplegables de Windows hacen eso... el usuario está atendiendo a un cliente en el formulario de Ventas, pero puede apretar una opción de una lista para abrir compras, también proveedores y cualquier otra cosa. La curva de aprendizaje de sistemas con menús “teletransportadores”, es muchísimo más empinada y difícil para el usuario que le meten el programa de prepo: “empezá a usar este programa que Juan se tomó vacaciones... mándame los informes en 10 minutos”.

En cambio, si primero ves el menú principal, luego entrás a clientes (imaginariamente a la derecha), de allí elegís a Gonzalez, apretás vender (para venderle a Gonzalez) y vas (a la derecha...) y aparece la lista de productos, elegís la TV que va a comprar Gonzalez, luego a la izquierda abajo está Volver... (que va a volver al “cuarto” anterior, o sea Clientes, para elegir a otro o para ver la CC de Gonzalez luego de la compra). Desde Clientes si vuelvo, vuelvo al lugar desde dónde partí, el “hall” del sistema, el menú principal, y desde allí puedo entrar a Proveedores etc...

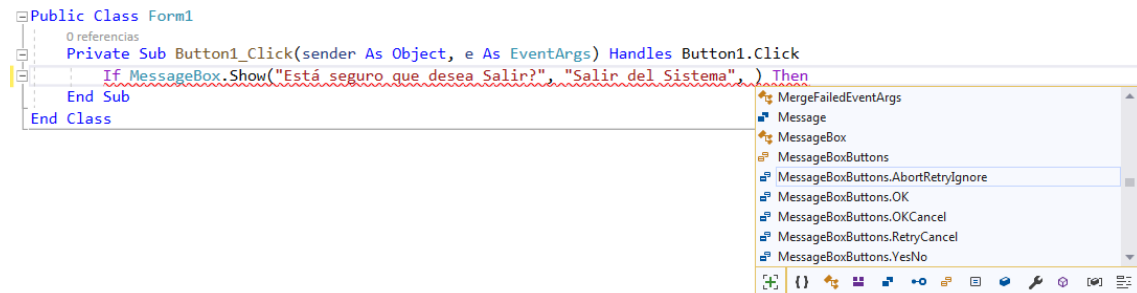
Yo uso eso en mis sistemas, y los vendo diciendo que lo aprenden a usar en 5 minutos... Cuanto más “natural” sea el sistema y su flujo, para el usuario, más rápido lo va a aprender y con más soltura se va a desenvolver.

Otra “tortura” para muchos, es cuando están vendiéndole a Gonzales, en una ventana encima del menú principal, y sin querer, aprietan sobre el menú principal. El menú “toma foco”, y queda por encima de la ventana en dónde le estaban cargando los datos a Gonzalez. Vos dirías...”hacé Alt-TAB”, pero no sabés la cantidad de gente que no sabe eso, y ni siquiera usar un mouse apropiadamente, y la empresa no les enseña. Por miedo a que “el pibe joven que recién entró”, les quite el trabajo, agachan la cabeza y..... abren otro formulario para cargar a Gonzalez de nuevo....

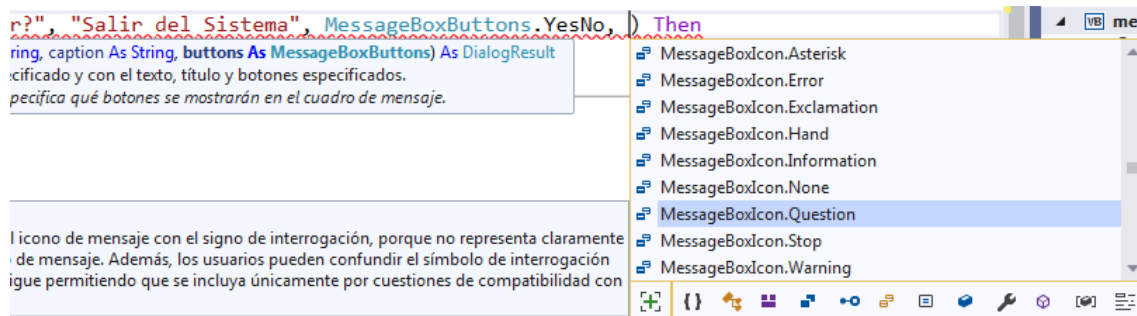
En la versión viejita de Visual Studio 6 (muchos años antes de .NET, pero su antecesor), cuando podías saltar de un formulario al que estaba abajo y cambiaba el foco, ese formulario era “no modal”, y si no podías (como en los carteles de error), era “modal”. En .NET, cuando no se puede cambiar, es una “ventana de Diálogo”, y cuando se puede es.... no tiene nombre... Entonces yo prefiero llamarlas a la vieja usanza “modal” (o Cuadro de Diálogo) y “no modal”.

Vamos a investigar hoy todas estas formas de navegar entre menús y formularios.

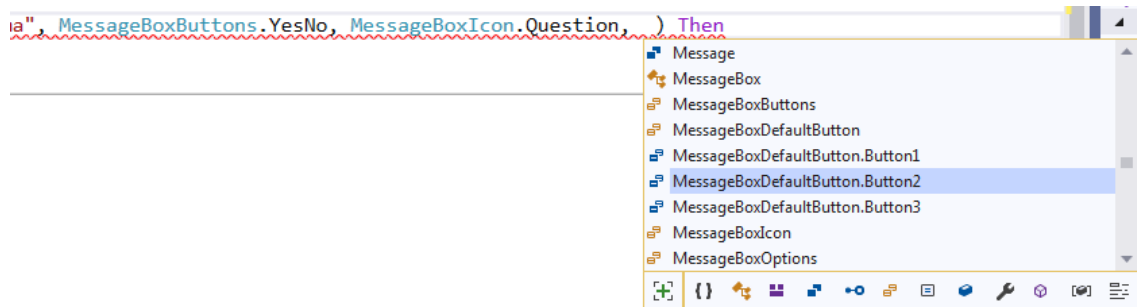
Hagamos primero una salida “elegante”, usando `messagebox.show` (también existe por compatibilidad con VB 6 en `msgbox`). Este objeto, abre un Cuadro de Diálogo, o ventana modal, de las que hablábamos. Con esto pueden hacer un mensaje de error con icono de crítico, que diga “Está seguro que desea formatear la Unidad C:?”, y que sólo haya un botón “Aceptar”... la pesadilla de cualquiera...



`MessageBox` es una función, que devuelve el valor del botón de respuesta que apretaste (OK, Cancel, SI, NO etc.). Como tiene una interacción y una respuesta, .NET los llama “Cuadros de Diálogo”. El primer valor es el mensaje, en éste caso “Está seguro que desea Salir?”. El siguiente (text), es el cartelito que va arriba a la izquierda en el cuadro. Pongamos “Salir del Sistema”. Luego te pide que elijas el tipo de “Diálogo”, o sea que botones querés que aparezcan. Elegí el que tiene “YesNo” (un botón SI y uno NO). El próximo es el ícono que va a tener. Resistí la tentación de poner el Critical... pongamos Question, que también pone un sonido más amigable (si querés probar con Critical, y un mensaje apocalíptico, probá... es tu tablero de pruebas, para eso lo estamos haciendo...).



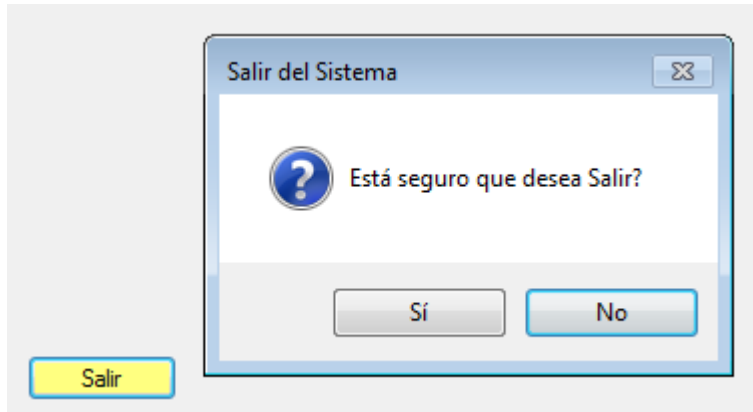
Tus respuestas son un botón SI y uno NO, aunque por defecto el foco está en SI, entonces si la pregunta es “estás seguro que querés borrar TODA la base de datos?” y vos venís embaldado apretando ENTERs... te tomaría SI... un ENTERicidio... Así que en el próximo valor ponemos `DefaultButton.Button2`, para que por defecto sea NO.



Ahora tenés que elegir con cuál “respuesta” del usuario vas a trabajar. Elegimos YES, y después del then, ponemos END. O sea que te queda: “SI estás seguro de que querés irte, entonces andate...”

```
...estion, MessageBoxDefaultButton.Button2) = DialogResult.Yes Then end
```

Probá ahora, y como quedás encerrado, apretá el botón Salir y vas a ver el cuadro de Diálogo. Es modal, así que no podés tocar en el menú principal para que cambie de foco. Te obliga a contestar.



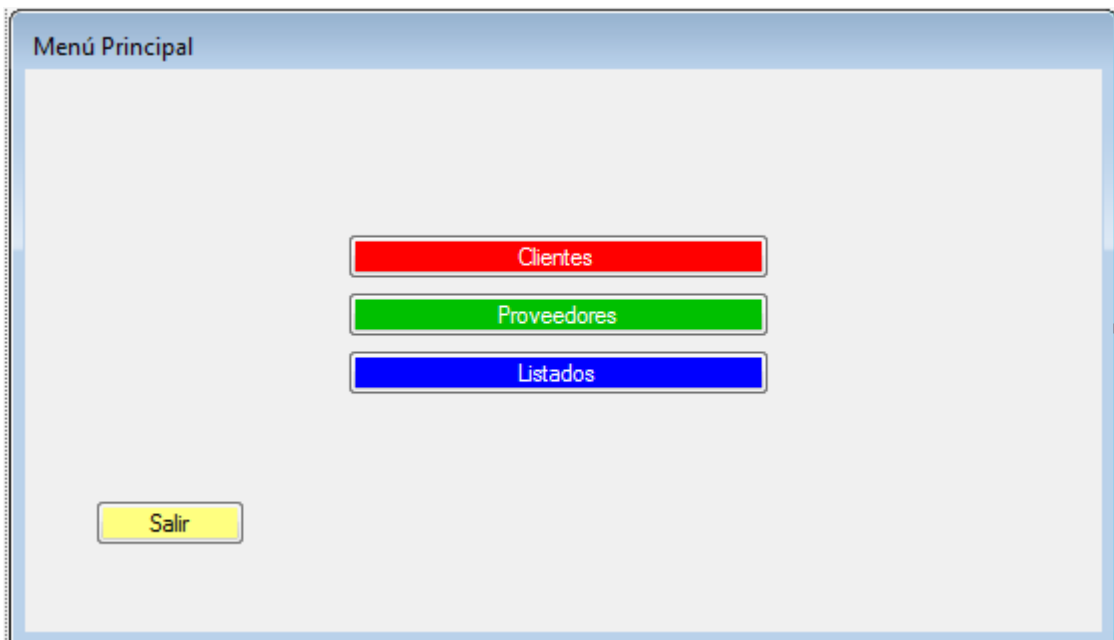
Dale... hacete otro botón con un mensaje terrorífico Critical sin salida o que la salida sea otro mensaje con otras respuestas abriendo un árbol de posibilidades de respuestas, y que alguien en tu casa lo pruebe.... Probá y experimentá.

Ahora hacé 3 botones en el medio del formulario:

bClientes con texto “Clientes” y color rojo (lo de los colores es sólo para que se aprecie más lo que vamos a hacer... jamás hagas formularios rojos o con colores resaltantes... la gente prefiere colores pastel, especialmente la gamma del azul, colores suaves, el pobre tipo que va a usar tu programa, va a estar 8 horas delante de lo que hagas... )

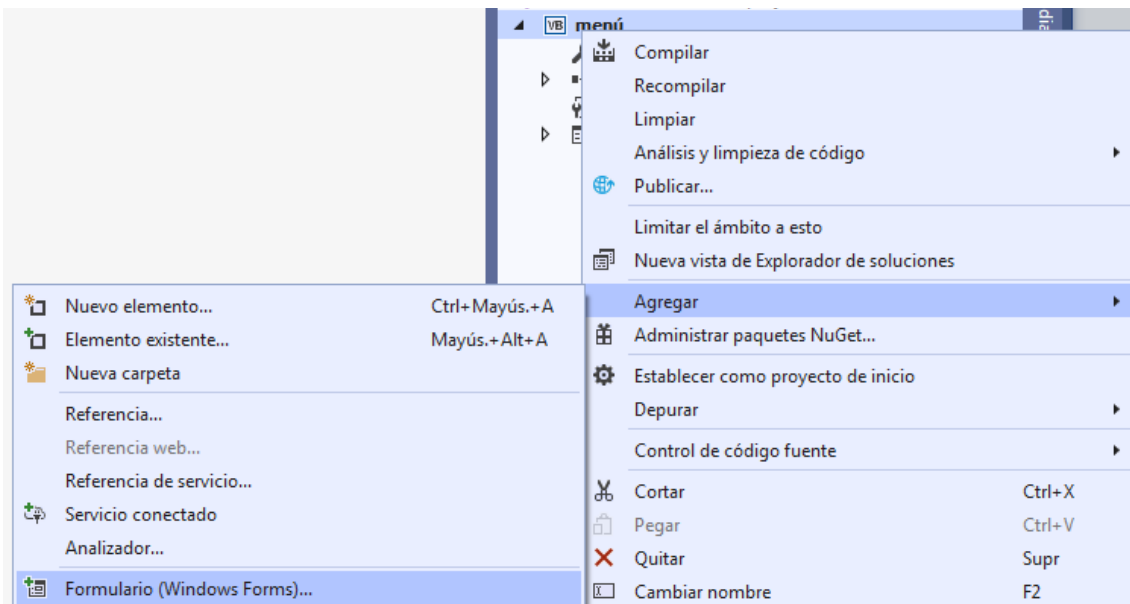
bProveedores con texto “Proveedores” y color verde.

bListados con texto “Listados” y color azul

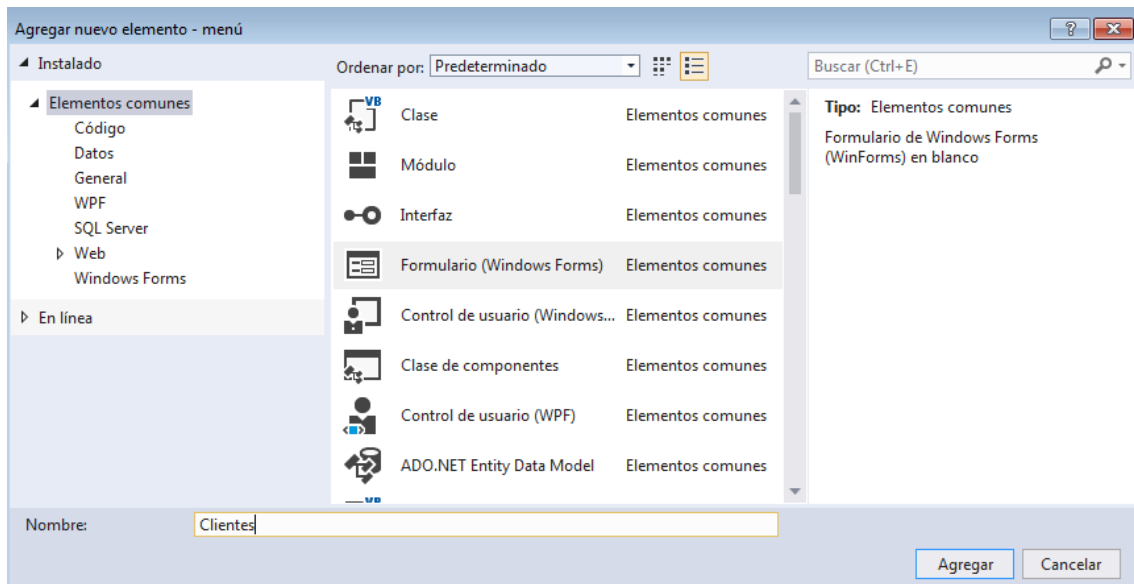


Ahora vamos a crear un nuevo formulario para Clientes

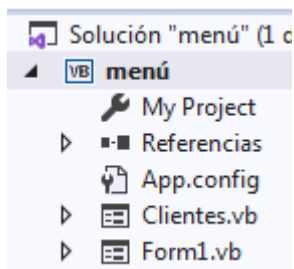
Botón derecho sobre “menú”, luego Agregar – Formulario (Windows Forms)



ATENCIÓN: aunque tengas un deseo irrefrenable de tocar Agregar directamente, primero ponle el Nombre: “Clientes” al formulario que vas a Agregar. Si no lo hacés, se va a llamar form2, y si bien es posible cambiarle luego el nombre, es bastante molesto...



Después de escribir Clientes, tocá Agregar:



Ahora tenés también a Clientes.

Para ir a clientes, hacele doble clic. Para volver a Form1, hacele doble clic.

Si...si..en la barra aparecen los dos, y podés elegir desde allí si vas a diseño o código de Form1 o diseño y código de Clientes; pero cuando tengas 50 o 100 formularios?... las pestañas van a ser de 1 milímetro cada una... mejor acostumbrarse a doble clic sobre el formulario que quieras.

Quedate en Form1, y hacé doble click en el botón bClientes

```
Private Sub bClientes_Click(sender As Object, e As EventArgs) Handles bClientes.Click
    Dim cli As New Clientes
    cli.ShowDialog()
End Sub
```

Para C# escribí:

```
Form cli = new Clientes();
cli.ShowDialog();
```

Qué pasa acá?... por qué no puedo llamar directamente al formulario Clientes?... por?... Porque es una CLASE, acabás de crear la clase Clientes, y ya sabemos que no podés usar una clase directamente, primero hay que instanciarla. Creamos entonces la instancia cli (puede ser el nombre que quieras) que es una instancia de la clase Clientes, heredando entonces todas sus propiedades, métodos y eventos.



Al usar New, creamos una Nueva instancia.

Ahora tenemos cli y dos caminos para abrir el formulario: ShowDialog que lo abre modal (tenés que salir con un botón que hagas vos), y no modal (si tocás el formulario que está debajo toma foco y se pone arriba).

Como cli hereda todas las propiedades de Clientes, podríamos cambiarle el color de fondo antes de que se abra (para eso era lo del color rojo...para probar esto).

```
Dim cli As New Clientes
cli.BackColor = Color.Red
cli.ShowDialog()
```

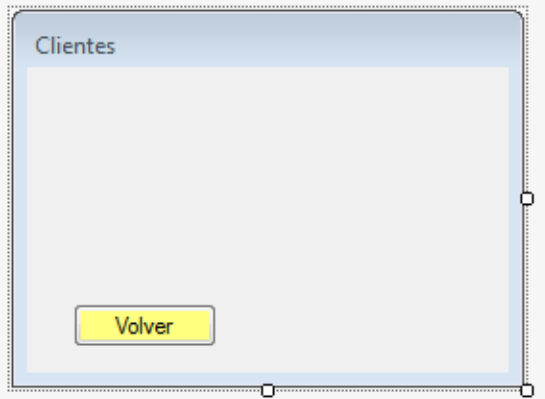
Todas las propiedades las tenés que cambiar ANTES de hacer ShowDialog.

En C#:

```
Form cli = new clientes();
cli.BackColor = Color.Red;
cli.ShowDialog();
```

Para que puedas volver del formulario Clientes (para los amigos cli, por ahora...), entrá en Clientes (era doble click sobre Clientes en el Explorador de Soluciones), y agregá el botón bVolver con text Volver.

Al formulario sacale el control box (ControlBox=false)



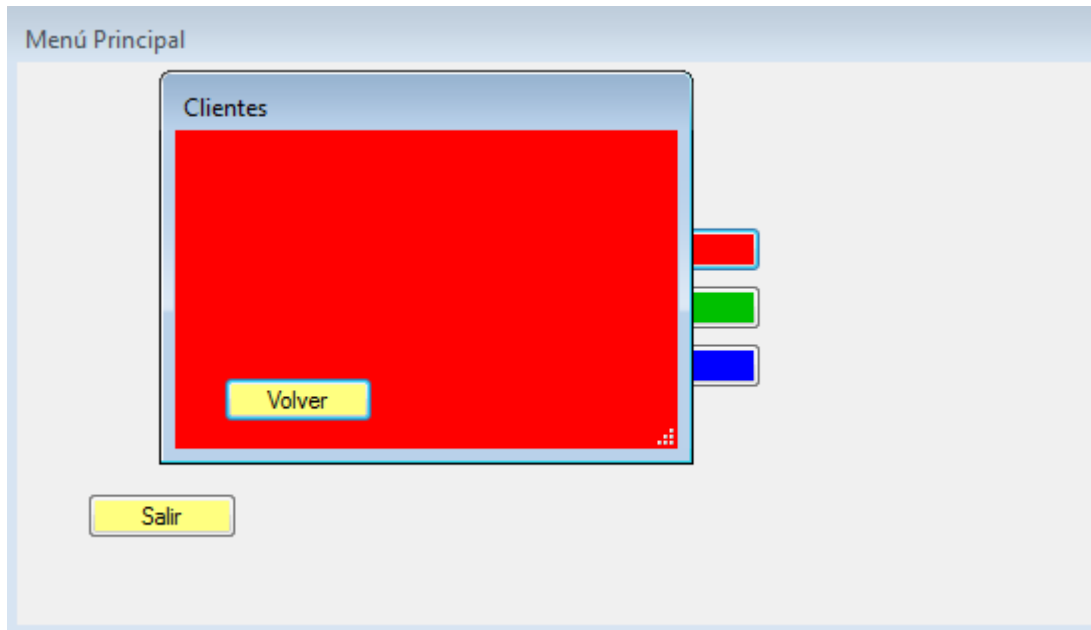
En el botón bVolver:

```
Private Sub bVolver_Click(sender As Object, e As EventArgs) Handles bVolver.Click
    Me.Close()
End Sub
```

Para C#: recordá de las clases anteriores que tenés que usar **this** en C# y **me** en VB. También para el backcolor y cualquier otra propiedad de los formularios

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

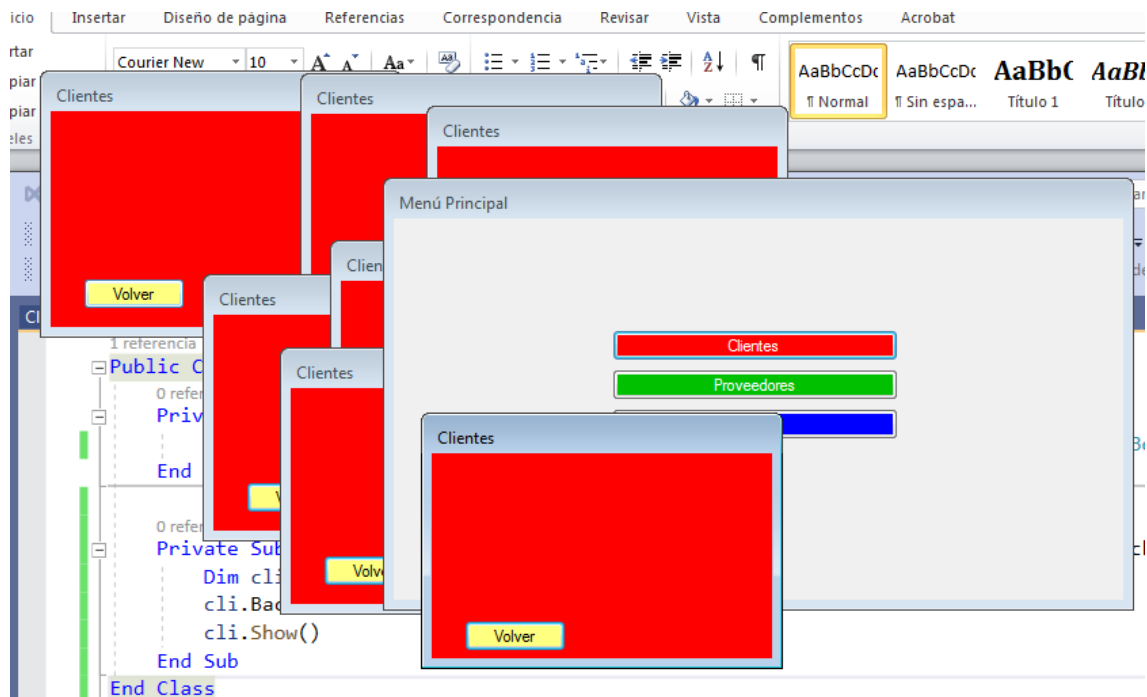
Probá ir y venir de Clientes.



Viste que no podés salir de Clientes si no es por Volver?, porque está modal... hagámoslo no modal por un ratito... (recordá: que ShowDialog es modal y Show es no modal)

```
Private Sub bClientes_Click(sender As Object, e As EventArgs) Handles bClientes.Click
    Dim cli As New Clientes
    cli.BackColor = Color.Red
    cli.Show()
End Sub
```

Ahora probá abrir varios Clientes (tocás en Form1 y apretás de nuevo Clientes sin cerrarlo)... que pasa? (!!!!!) (si pensás “No pasa nada, no anda...”, mové un poco la ventana o mirá si no están debajo del Visual...)



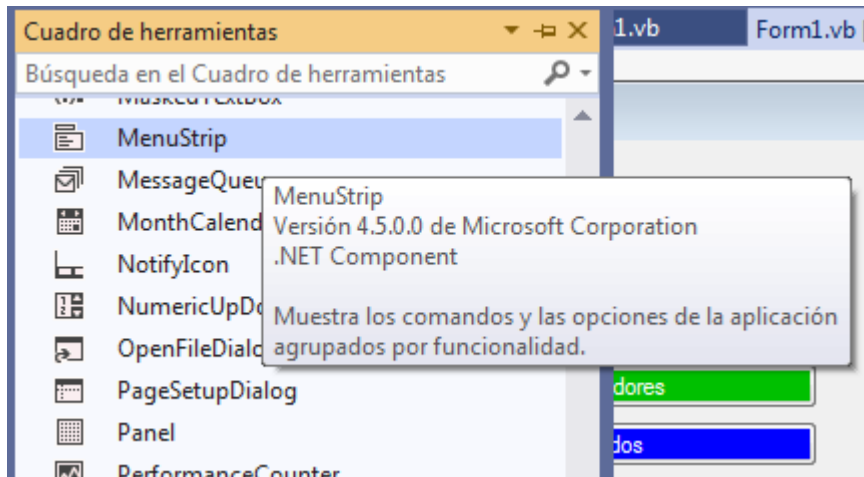
Imaginate que le pasara esto a un viejito que estaba por jubilarse en el trabajo y le dicen... cubrí a Juan que está de vacaciones... es fácil el programa... no vas a tener problemas... no hay edad para aprender... vuelvo más tarde....

Por eso te conté toda la historia de los cuartos y de usar ShowDialog.

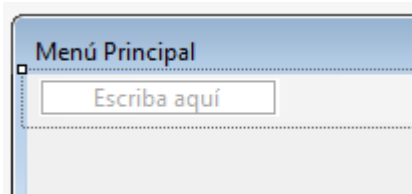
Esto podría servir únicamente para gente muy avispada, que mientras que le está vendiendo a Gonzalez que anda indeciso, abre otra ventana para venderle a Rodriguez que está mirando una heladera, y con otra ventana le vende a Perez por teléfono... si hay plata seguro que lo usan sin problemas...

Volvé a poner ShowDialog, y creá los formularios Proveedores y Listados, usando los colores que les corresponden.

Ahora y para terminar por hoy, vamos a ver el menú “teletransportador”, un menú desplegable o top down, o menuStrip como se llama acá:

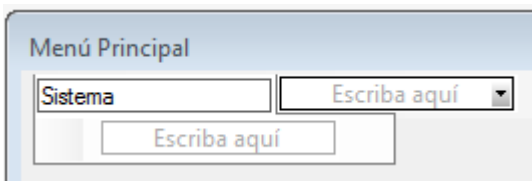


Cuando lo selecciones, aparece arriba (siempre es así):



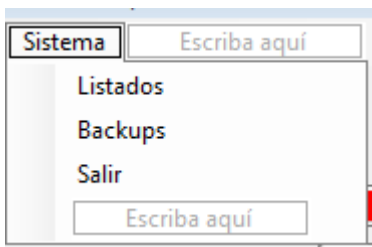
Lo que pongas en “Escriba aquí”, es un menú de primer nivel, como sería Archivo, Ver, Editar etc. en Visual.

Poné “Sistema”:

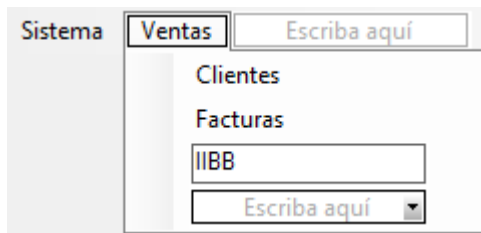


Ahora vas a poder escribir abajo para que pertenezca a Sistema, o a la derecha para hacer otro menú de primer nivel.

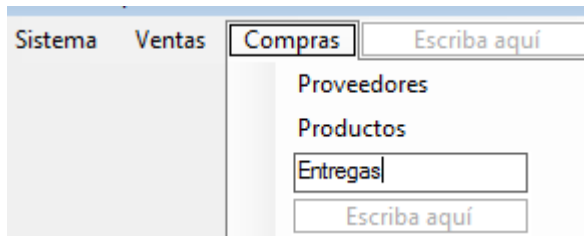
Poné esto en Sistemas:



Ahora agregá Ventas:

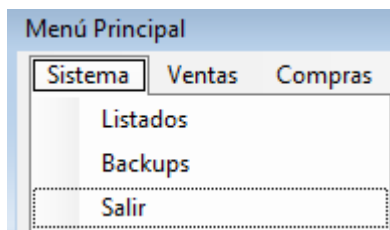


y Compras:



Podés editar cada cuadradito, y también borrarlo, o moverlo con drag and drop.

Cada ítem, funciona igual que un botón... hacele doble click a Salir en Sistema.

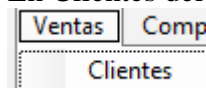


```
Private Sub SalirToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles SalirToolStripMenuItem.Click
    '
End Sub
```

Acá copiamos el messagebox que pusimos en el botón Salir, para que puedas salir por el botón o por el menú.

```
Private Sub SalirToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles SalirToolStripMenuItem.Click
    '
    If MessageBox.Show("Está seguro que desea Salir?", "Salir", MessageBoxButtons.YesNo) = DialogResult.No Then
        Return
    End If
End Sub
```

En Clientes del menú



Copió el código del botón bClientes, para que puedas abrir Clientes desde los dos lados.

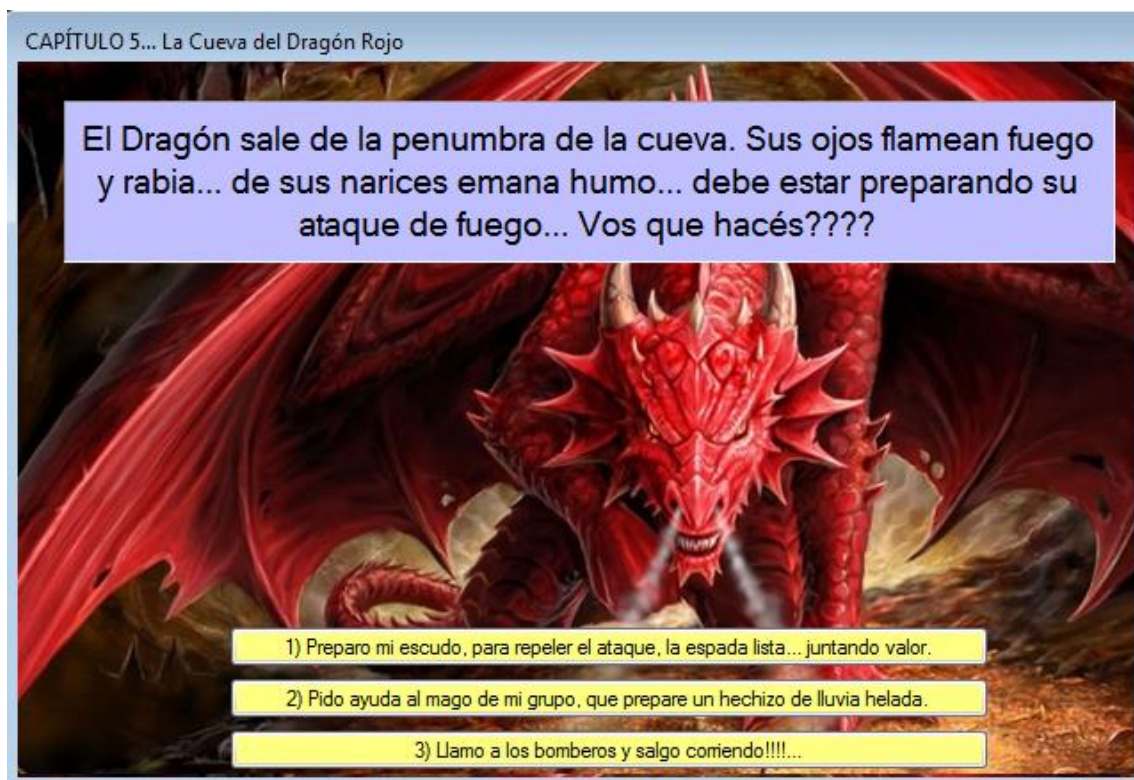
```
Private Sub ClientesToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ToolStripMenuItem1.Click
    Dim cli As New Clientes
    cli.BackColor = Color.Red
    cli.Show()
End Sub
```

Hacé lo mismo con Proveedores y Listados, y probá que todo ande.

FELICITACIONES si te anda todo lo que hicimos; sino, andá con cuidado repasando los puntos del tutorial, o escribí en el foro.

Armame la versión C# de ésta clase.

Investigá sobre los temas que vimos hoy, experimentá por tu cuenta, ármate algo super. De hecho, podrías hasta hacer un juego estilo “Elige tu propia aventura”, en donde vas enganchando cuadros de diálogo o formularios hechos por vos modales, para que puedas agregarles más botones de opciones. Te dá hasta para una mini aventura Dungeons & Dragons (no digo que lo hagas, es sólo una idea que se me vino ahora) (me da ganas de hacerlo yo...), por ejemplo:



Si te entusiasmó la idea, o al menos querés saber un par de tips... te agrego un plus a la clase..

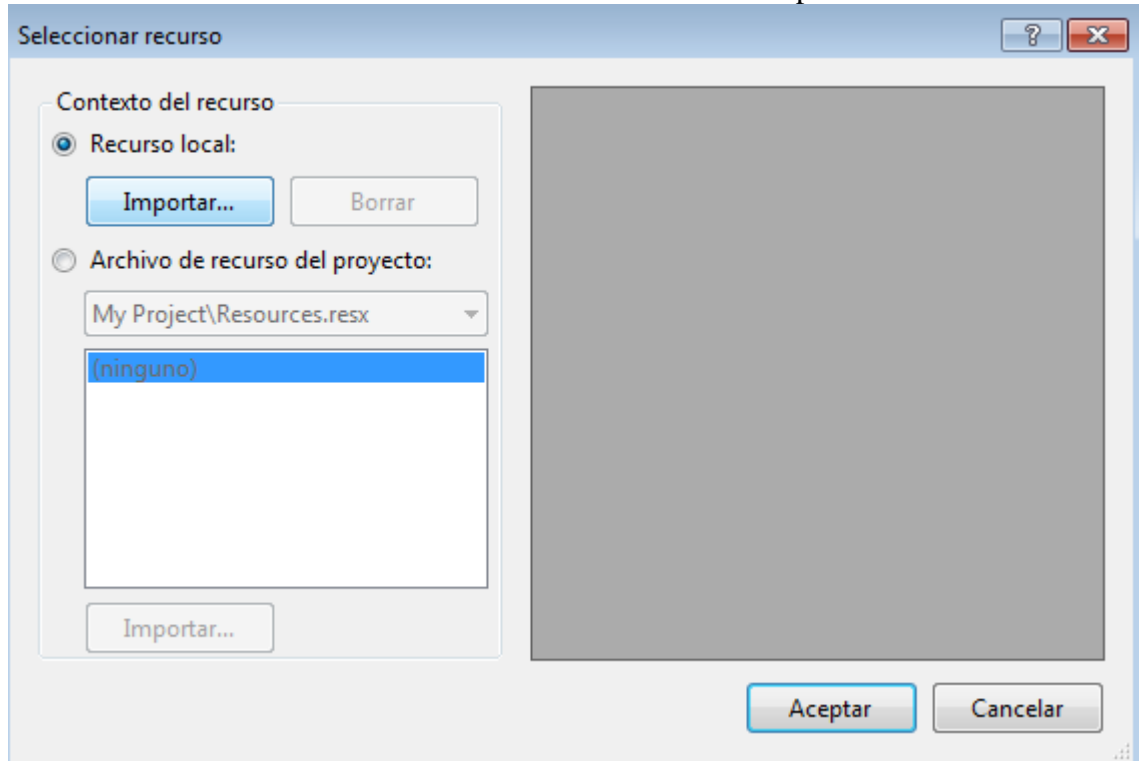
- 1) Como hacer que una label me quede en varias líneas como la label violeta de arriba?... yo escribo y me lo deja en una sola línea larguísima... Y cómo centro el texto?

AutoSize	False
TextAlign	MiddleCenter

- 2) Como pongo al dragón en el fondo???????

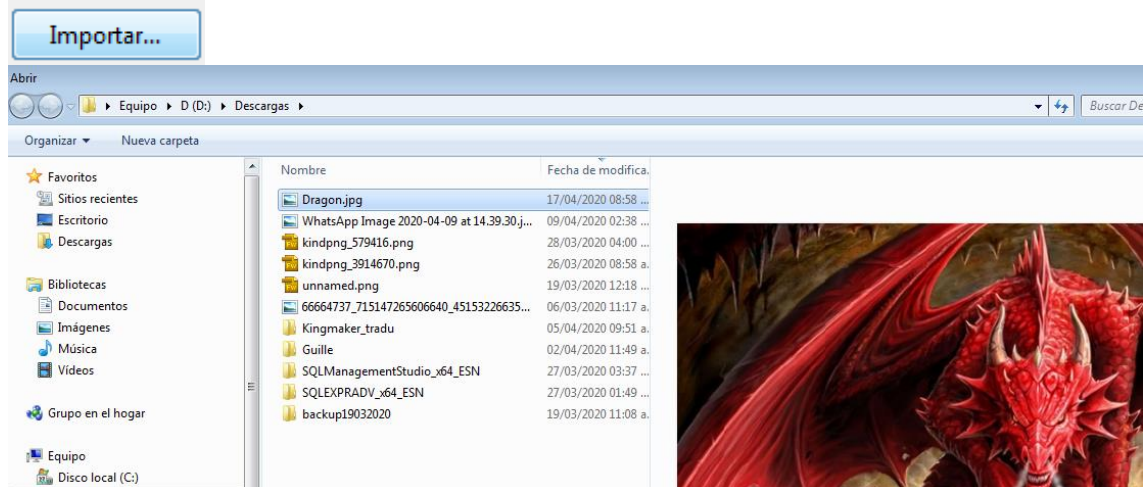
- Primero tenés que tener la imagen como un jpg en el disco (bajala de Google, botón derecho guardar imagen como)
- Seleccionas el formulario (también podés usar un objeto PictureBox)

BackgroundImage ☐ (ninguno) ... tocás los 3 puntitos



c.

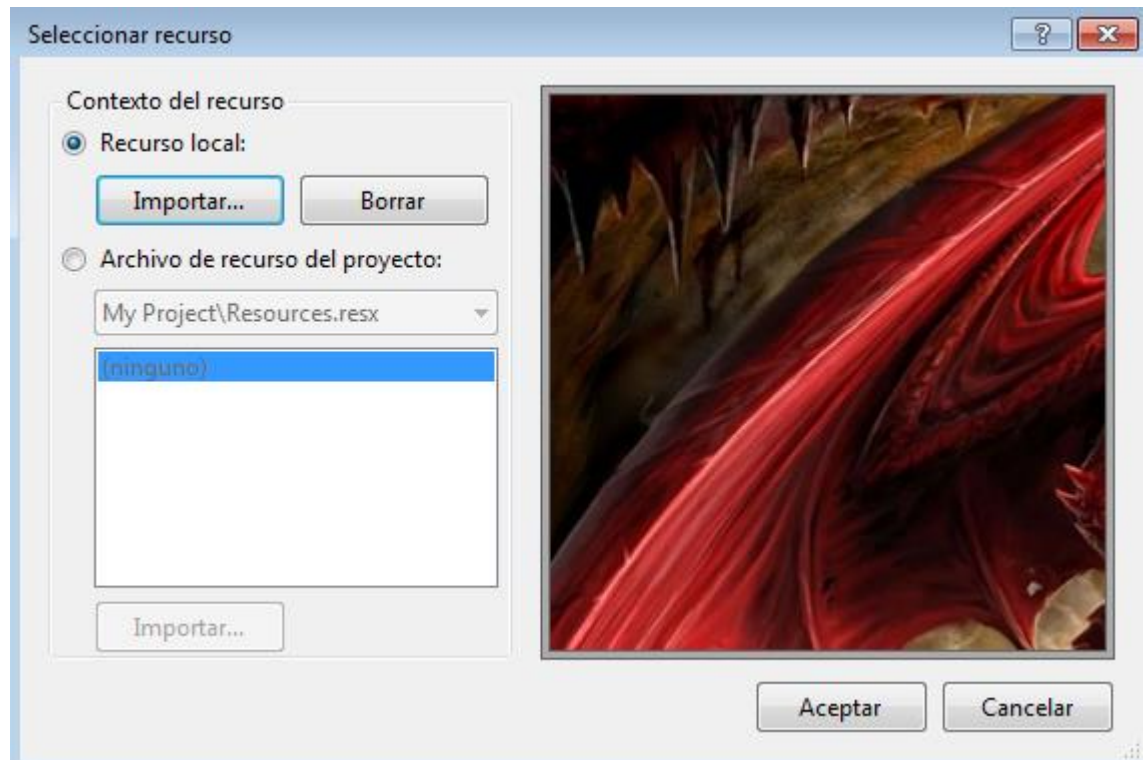
d.



e.

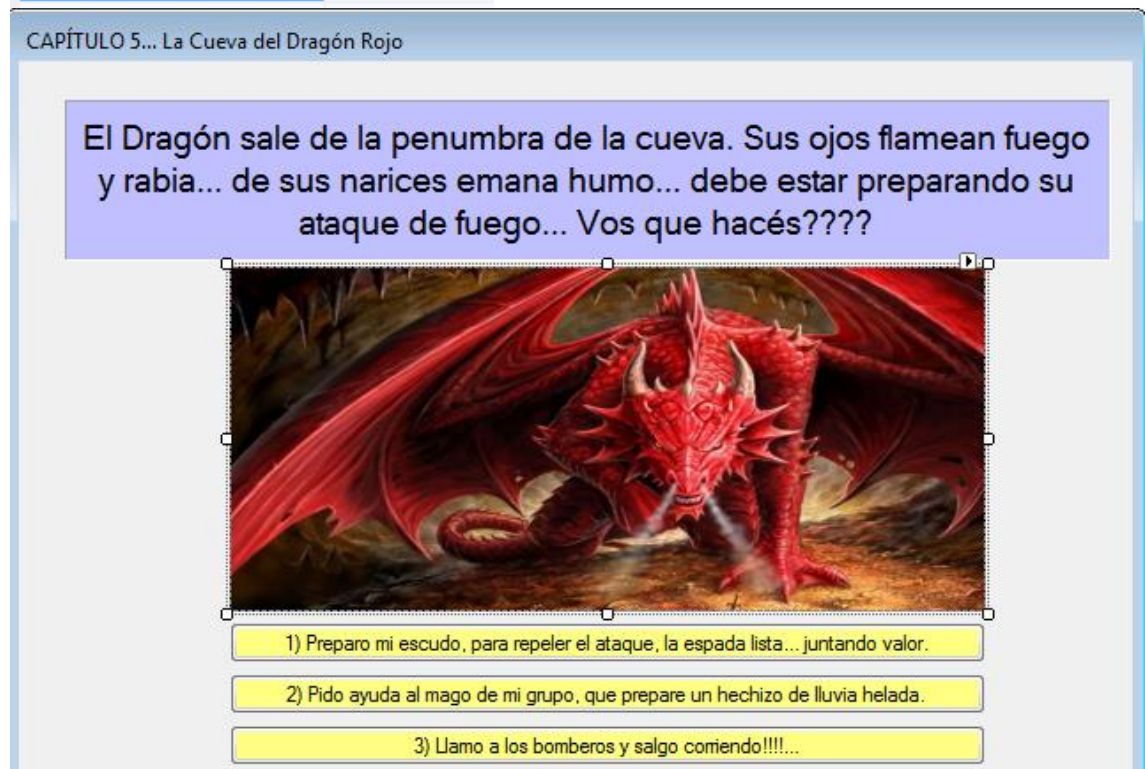
- f. Búscalo en tu disco y tocá Abrir





- g. h. Aceptar... y ajustá el tamaño del formulario a la foto. Si usaste un PictureBox, usás Image para cargar la foto, y luego podés adaptar el tamaño de la foto al PictureBox usando:

- i. **SizeMode** **StretchImage**



- j. k. Entonces, podés usar otra imagen fija para los fondos. Igualmente lo mejor es que la imagen tenga el mismo tamaño del PictureBox, porque una foto de 3 MB y 1980, en un PictureBox de 300 x 200 sigue consumiendo 3 MB.



Si vos cargás la imagen como la puse en la clase, queda incluida en el exe... probá mover el exe (como vimos en Clase 1) a otra carpeta, y vas a ver que es más grande porque tiene la imagen adentro. Si es un fondo para un sistema, con eso prevenís que venga un vivo, y te cambie el fondo poniendo un logo de él para vender tu sistema como propio, por ejemplo.

Cuando estemos en ASP .NET, podrás usar ".imageUrl" que carga directamente desde internet. En Windows no tenemos eso, pero podés hacerlo igual:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Me.BackgroundImage = Image.FromFile("D:\Descargas\dragon.jpg")
End Sub

0 referencias
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    PictureBox1.ImageLocation = "https://es.babelio.com/users/liste_Historias-de-dragones_6292.jpeg"
    PictureBox1.Load()
End Sub
```

Acá te pegué el código de dos botones, Button1 y Button2. Necesitás también un PictureBox1, con stretch.

Button1, carga la imagen desde el disco rígido, en vez de tenerla dentro del exe (cambiá la ruta y nombre del archivo)

Button2, carga la imagen desde internet, podés poner la url que quieras, siempre que apunte a una imagen. Probá con esa que busqué.

### **Cómo podría ser el juego?:**

Viste que desde el menú principal llamamos a clientes, y desde clientes vos podés volver o bien, podrías agregarle más botones a clientes, que abran otros formularios. Por ejemplo, en clientes podés agregar el botón "Ventas" que va a abrir el formulario Ventas (agregalo) desde Clientes. En Ventas, podrías agregar otro botón que abra "facturas" (hacelo...).

Cuando ejecutes, en el principal apretás clientes, en clientes apretás ventas, y desde ventas apretás facturas, y así podrías ir agregando cada vez más formularios. Imaginate ahora que en Clientes, aparte del botón ventas que abre ventas, agregás (hacelo...) el botón Deudas, que abre el formulario Deudas. Deudas tiene 2 botones, uno "Vencidas" que abre el formulario Vencidas y otro No Vencidas que abre el formulario NoVencidas. Si podés armá la estructura hasta allí. A todos los formularios nuevos ponele volver. Cuando ejecutes vas a ver que podés ir siguiendo caminos diferentes. Si en clientes elegí Ventas voy por un lado y veo determinados formularios, pero si elegí Deudas, voy por otro lado y veo otros formularios. Estarías armando una estructura de navegación con ramas que se van bifurcando.

Entonces:

Capítulo 1: Hacés un formulario que presenta al personaje, y preguntás:

"Se te presenta una gran aventura... en realidad dos... si sos un gran guerrero, deberás matar a un Dragón, pero si sos un mago, enfrentarás a un Demonio!. Que eres valiente humano?"

Entonces ponés 3 botones:

Button1: "Soy un caballero y acepto el desafío del Dragón!"

Button2: "Soy un mago, y ningún demonio está a la altura de mi magia!, mataré al Demonio!"

Button3: "Qué?... nha... ni loco... andá a buscar a otro gil..."

Si apretás button3, abris un formulario que te muestra que perdiste y end.

Si apretás Button1, abris el formulario Dragon1, que tendrá otra consigna y otros botones para responder y para mandarte a otros formularios de la rama de los dragones.

Si apretás Button2, abris el formulario Demonio1, que tendrá otra consigna y otros botones para responder y para mandarte a otros formularios de la rama de los demonios.

Lo vas armando paso a paso como con el de clientes y ventas usando tu imaginación.

Si en un formulario el usuario toca la opción incorrecta estilo:

"hay dos caminos adelante, y te persiguen demonios, no tenés tiempo de pensar... el de la izquierda es una saliente en un precipicio, el de la derecha, es un camino de tablas podridas sobre un río de lava. Cúal elegís?, rápido!!"

Botón 1: "me arriesgo por el acantilado"

Botón 2: "me arriesgo por las tablas podridas"

Botón 3: "me quedo donde estoy y enfrentaré la muerte segura con valentía"

Botón 3: te manda al formulario de muerte, con un cartel "Tu indecisión causó tu ruina, has muerto por las flamas de cientos de pequeños demonios"

Botón 2: te manda al formulario de muerte con un cartel "Las maderas estaban podridas, te acordás?... y tu armadura es muy pesada. Las maderas cedieron y caiste a la lava ardiente... no es como Terminator, tu no volverás" y termina el programa.

Botón 1: te manda a otro formulario para seguir la aventura, el usuario eligió la opción que vos como diseñador del juego, consideraste la correcta.

En cada formulario, podés poner un gráfico genérico o bien uno que muestre lo que está pasando.

PD: Si te animás a armar un juego como el que propuse, por favor subilo al foro, lo quiero ver!!! (no es un ejercicio, sólo es una idea disparatada para valientes, intrépidos y aguerridos estudiantes).

Nos leemos en el foro!

Guillermo Guastavino