

TÍTULO

DESARROLLO DE APLICACIONES I

Sesión Nro. 03

Objetivos de la sesión:

Al culminar la presente podrás:

- Construir formularios MDI , que incluyan barras de menú y de estado
- Crear un formulario de Login , para conocer los fundamentos del acceso correcto a la aplicación en base al ingreso de credenciales (Usuario y Password)
- Interactuar con los principales cuadros de dialogo , gestionando archivos de imágenes o XML.



Temario

1. Creando aplicaciones MDI
2. El Control MenuStrip , StatusStrip y Timer
3. Otros Controles de importancia
4. Principales cuadros de dialogo



Tema Nro. 1: Creando aplicaciones MDI



Aplicaciones MDI

SDI	MDI
	
Sólo hay un documento visible	Muestra varios documentos a la vez
Debe cerrarse un documento antes de abrir otro	Cada documento se muestra en su propia ventana

MDI (Interfaz de documentos múltiples), es un tipo de lógica de presentación basada en varios formularios que pueden estar abiertos en simultaneo, pero administrados por un formulario contenedor, que garantiza el orden y buen manejo de la presentación del aplicativo,



Como definir un formulario MDI

- **Crear un formulario primario**
 - Crear un nuevo proyecto
 - Configurar la propiedad **IsMdiContainer** como **True**
 - Añadir un elemento de menú para invocar el formulario secundario
- **Crear un formulario secundario**
 - Añadir un nuevo formulario al proyecto
- **Invocar un formulario secundario desde uno primario**

```
private void ejemplo1ToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmTabControl frm1 = new frmTabControl();
    frm1.MdiParent = this;
    frm1.Show();
}
```

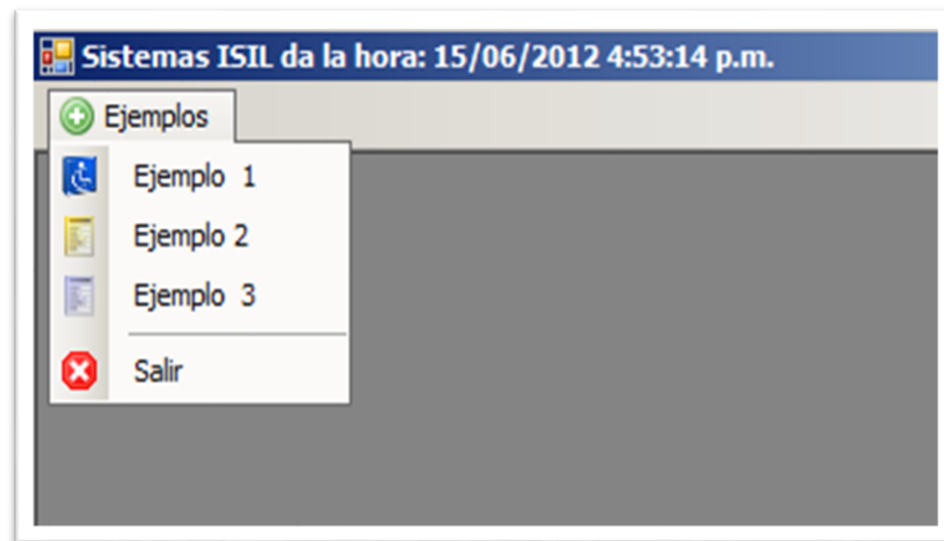


Tema Nro. 2: Controles MenuStrip, ContextMenuStrip, StatusStrip y Timer



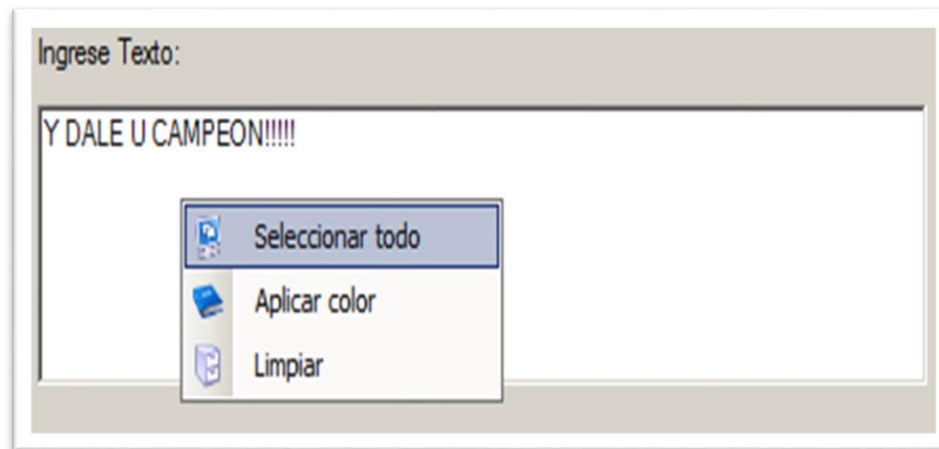
El Control MenuStrip

- Los menús exponen la funcionalidad a sus usuarios presionando comandos agrupados por un tema común.
- El control MenuStrip admite la interfaz de múltiples documentos (MDI) y combinación de menús, la información sobre herramientas y el desbordamiento. Puede mejorar la utilidad y legibilidad de sus menús agregando teclas de acceso, teclas de método abreviado, marcas de verificación, imágenes y barras separadoras.



Control ContextMenuStrip

- El control ContextMenuStrip representa los menús contextuales que aparecen cuando el usuario hace clic con el botón secundario del mouse (ratón) en un control o en un área del formulario. Por ejemplo, se puede usar un menú contextual asignado a un control TextBox para proporcionar elementos de menú que permitan seleccionar todo el texto , aplicar un color de fuente o cualquier otra acción.



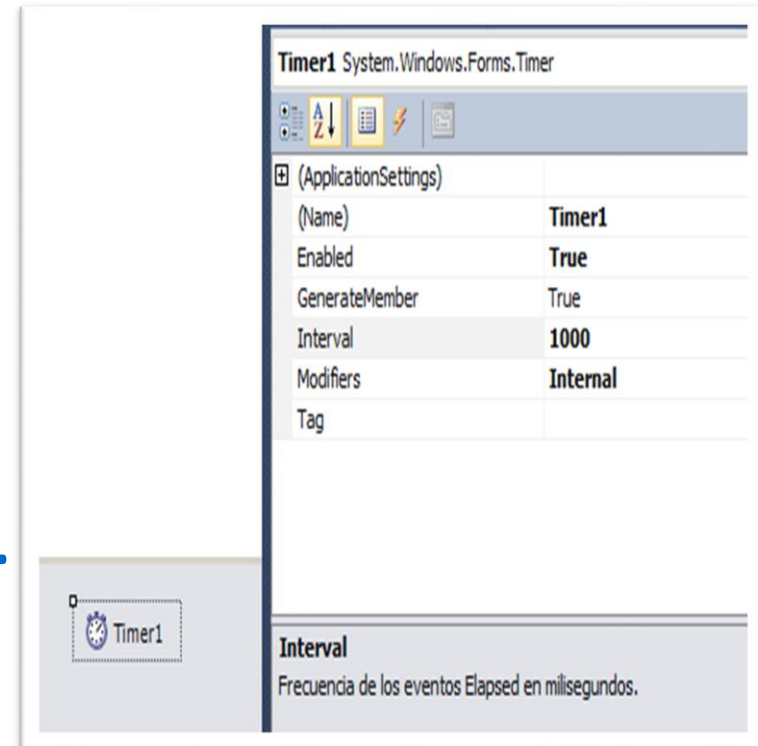
Control StatusStrip

- El control StatusStrip permite crear una barra de estado personalizada, ideal para mostrar información pertinente a la forma como se esta ejecutando el aplicativo, el usuario, si se tiene o no conexión a red o hasta el tiempo que se este ejecutando la aplicación.
- En este control podemos colocar label, combos, entre otros controles.

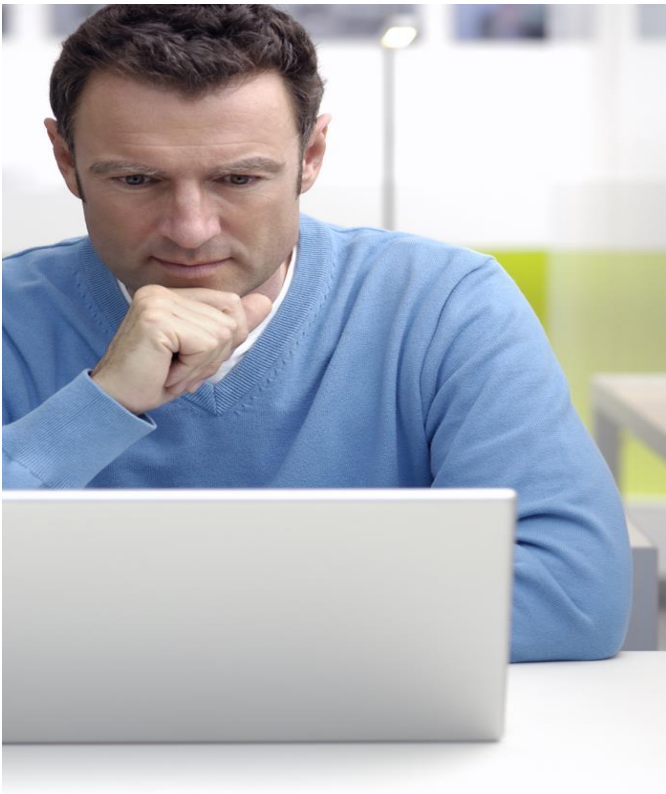


Control Timer

- El control Timer es un componente que permita la programación de acciones cada cierto intervalo de tiempo.
- Su propiedad Interval permite establecer cada cuanto tiempo se lanzara el evento Tick del control, para que en él se codifique el conjunto de acciones que se desea ejecutar cada vez que se cumpla el ciclo del intervalo.
- Esta propiedad tiene como unidades milisegundos, es decir, si por ejemplo se desea hacer algo cada segundo, el valor de la propiedad debería ser 1000



Demostración 1: Creación de una aplicación MDI



- En esta demostración, aprenderemos a crear un MDI completo , que incluya un MenuStrip y un StatusStrip con imágenes y control de tiempos



Tema Nro. 3: Controles de tipo Cuadro de Dialogo



Controles de Cuadro de Dialogo

OpenFileDialog	Permite a los usuarios abrir archivos mediante un cuadro de diálogo preconfigurado
SaveFileDialog	Selecciona los archivos a guardar y la ubicación donde deben guardarse
ColorDialog	Permite a los usuarios seleccionar un color de la paleta y agregar colores a ésta
FontDialog	Expone las fuentes actualmente instaladas en el sistema
PrintDialog	Selecciona una impresora y determina otras configuraciones relacionadas con la impresión
PageSetupDialog	Configura los detalles de la página para su impresión
PrintPreviewDialog	Muestra el aspecto que tendrá un documento cuando se imprima



Cómo utilizar las entradas de un cuadro de dialogo personalizado

Propiedad DialogResult

Utilizar el valor devuelto por esta propiedad para determinar qué acción ha realizado el usuario

Ejemplo

El valor **DialogResult.Cancel** indica que el usuario ha hecho clic en el botón **Cancel**

La propiedad **DialogResult** puede establecerse en tiempo de diseño o en tiempo de ejecución



Cómo utilizar las entradas de un cuadro de dialogo personalizado

Recuperar y utilizar resultados de cuadros de diálogo

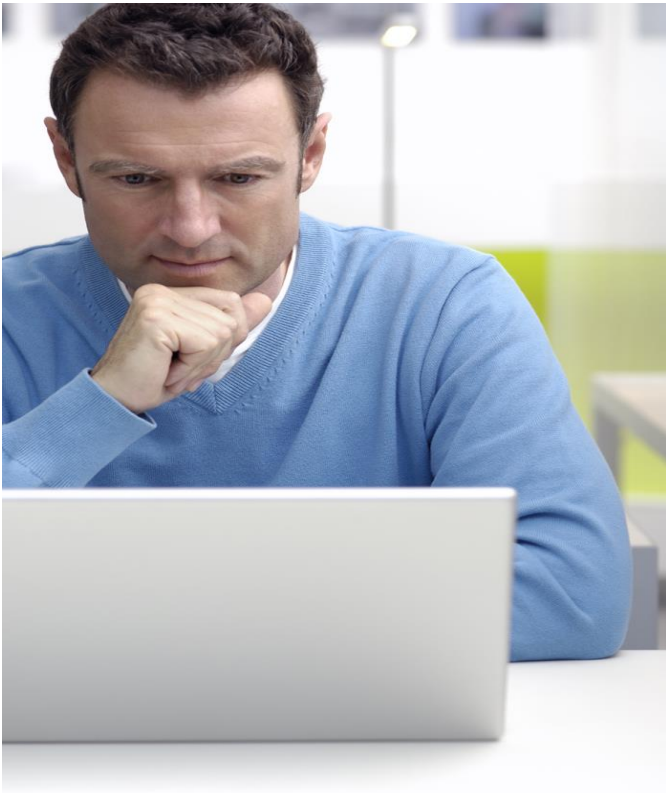
1 En la ventana de código, ir al controlador de eventos o el método para el que desea establecer la propiedad **DialogResult**

2 Añadir código para recuperar el valor **DialogResult**

```
private void MDIPrincipal_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult vrpta;
    vrpta = MessageBox.Show("Seguro de salir de la aplicación??", "Mensaje",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (vrpta == DialogResult.No)
    {
        e.Cancel = true;
    }
}
```



Demostración 2: Miscelánea de Controles



- **En base a lo aprendido hasta la presente sesión crearemos una aplicación para el manejo de una agenda de contactos.**



Conclusiones de la sesión:

- Podemos gestionar de mejor manera los formularios de nuestra aplicación desde formularios MDI, complementado con barras de menú y de estado.
- Manejar el concepto del formulario de inicio de sesión garantiza el acceso a usuarios acreditados, mediante un login y password.
- El uso de controles como DateTimePicker, MaskedTextBox entre otros, que enriquecen la lógica de presentación de las aplicaciones.

