

Introducción al desarrollo de aplicaciones de escritorio con .NET

Objetivo

Conocer los elementos involucrados en el desarrollo de una aplicación de escritorio Windows con Visual Studio 2005 y la plataforma Microsoft .NET, presentando las novedades introducidas al respecto en la nueva versión 2.0

Temario (1/2)

- Introducción a Windows Forms
- El diseñador de formularios
- El objeto Form
- Controles
- Diseño de Interfaz de Usuario
- Herencia visual

Temario (2/2)

- Configuración
- Diálogos comunes
- Distribución de la aplicación
- Referencias

Temario (1/2)

➤ Introducción a Windows Forms

- ¿ Qué es Windows Forms ?
- ¿ Qué es un formulario ?
- Cómo crear un proyecto de aplicación para Windows

➤ El diseñador de formularios

➤ El objeto Form

➤ Controles

➤ Diseño de Interfaz de Usuario

➤ Herencia visual

¿ Qué es Windows Forms ?

- ▶ Windows Forms es un subconjunto de la .NET Framework Class Library que permite el desarrollo de aplicaciones de escritorio ricas bajo Microsoft Windows.
- ▶ Incluye clases base, interfaces, enumeraciones y controles gráficos diversos.

¿ Qué es un formulario ?

- ▶ Un formulario Windows Forms actúa como interfaz del usuario local de Windows.
- ▶ Los formularios pueden ser ventanas estándar, interfaces de múltiples documentos (MDI), cuadros de diálogo, etc.
- ▶ Los formularios son clases que exponen propiedades, métodos que definen su comportamiento y eventos que definen la interacción con el usuario.

Cómo crear un proyecto de aplicación para Windows

➤ Utilizando los entornos de desarrollo:

➤ Visual C# 2005 Express Edition o

➤ Visual Basic 2005 Express Edition,

se deben seguir estos pasos:

1. *En el menú File, seleccionar New Project*
2. *En la ventana que aparece seleccionar Windows Application*

➤ No es necesario crear un directorio para los archivos del proyecto, éstos son creados en un directorio temporal hasta que se decida grabarlos.

Temario (1/2)

- Introducción a Windows Forms
- El diseñador de formularios
- El objeto Form
- Controles
- Diseño de Interfaz de Usuario
- Herencia visual

El diseñador de formularios

- Al momento de diseñar un formulario, el diseñador de Visual Studio Express escribe de forma automática el código que describe a cada uno de los controles y al propio formulario.
- El concepto de Partial class que incorpora .NET 2.0 permite separar el código de una clase en varios archivos fuentes diferentes.
- El diseñador de formularios utiliza esta técnica para escribir en un archivo aparte todo el código que él mismo genera.
- Esto permite organizar más claramente el código, manteniendo separada la lógica de la aplicación en un archivo diferente.

Temario (1/2)

- Introducción a Windows Forms
- El diseñador de formularios
- El objeto Form
 - Generalidades
 - Eventos, Métodos
 - Ciclo de vida
 - Trabajando con el Mouse
 - Trabajando con el Teclado
 - Foco de controles y orden de tabulación
 - Message Box
- Controles
- Diseño de Interfaz de Usuario
- Herencia visual

Generalidades (1/2)

- ▶ El objeto Form es el principal componente de una aplicación Windows.
- ▶ Algunas de sus propiedades admiten valores de alguno de los tipos nativos de .NET
 - ▶ Ejemplo Código C#

```
miForm.ShowInTaskBar = false;  
miForm.Opacity = 0.83;
```

Generalidades (2/2)

- Otras propiedades requieren la asignación de objetos
 - Ejemplo en C#

```
miForm.Size = new Size(100, 100);  
miForm.Location = new Location(0, 0);
```

Métodos

► Show()

- Visualiza el formulario. Puede especificarse su formulario Owner.
 - Si un formulario A es owner (dueño) de otro B, el formulario B siempre se visualizará sobre el A, sin importar si otro formulario está activo.

► ShowDialog()

- Visualiza el formulario como cuadro de diálogo Modal.
 - Un formulario visualizado de forma modal no permite que otro formulario perteneciente a la misma aplicación tome foco. Esta opción es utilizada para mostrar cuadros de diálogo y focalizar la atención del usuario.

Eventos (1/2)

- Manejadores de eventos
 - Por cada evento soportado por el Form (o por cualquier otro objeto) es posible definir varios métodos manejadores.
 - A su vez, un método manejador puede controlar eventos disparados por diferentes objetos.

Eventos (2/2)

► Ejemplos:

► Código C#

```
// Varios manejadores para un evento  
this.Click += new EventHandler(MetodoManejador1);  
this.Click += new EventHandler(MetodoManejador2);  
// Un mismo manejador para diferentes eventos  
this.Load += new EventHandler(ManejadorCentralizado);  
this.Activated += new EventHandler(ManejadorCentralizado);
```


Ciclo de vida del formulario

- Muchos de los eventos a los que responde el *objeto Form* pertenecen al ciclo de vida del formulario
- Entre estos eventos se encuentran los siguientes, en orden de ocurrencia:
 - Load: El formulario está en memoria, pero invisible.
 - Paint: Se "pinta" el formulario y sus controles.
 - Activated: El formulario recibe foco.
 - FormClosing: Permite cancelar el cierre.
 - FormClosed: El formulario es invisible.
 - Disposed: El objeto está siendo destruido.

Trabajando con el Mouse

- ▶ El mouse puede ser controlado escribiendo código para alguno de estos eventos:
 - ▶ `MouseClicked`
 - ▶ `MouseEnter`
 - ▶ `MouseMove`
- ▶ A través de los argumentos que reciben los manejadores de estos eventos se puede obtener:
 - ▶ La posición del puntero
 - ▶ Qué botón fue presionado
 - ▶ Cantidad de "pasos" que fue girada la rueda

Trabajando con el Teclado

- El manejador del evento `KeyPress` informa a través del argumento `e.KeyChar` el código de la tecla presionada.
- Es posible cancelar el comportamiento por defecto asignando "true" al argumento `e.Handled`.
- Los argumentos que reciben los manejadores de los eventos `KeyDown` y `KeyUp` informan del estado de las teclas `Alt`, `Ctrl` y `Shift`.
- El evento `HelpRequested` es disparado cuando se presiona la tecla `F1`.

Foco de controles y orden de tabulación

- ▶ El objeto Form expone diferentes propiedades, métodos y eventos que permiten controlar la navegabilidad del formulario:
 - ▶ Propiedad *CanFocus*: Indica si el control puede tomar foco.
 - ▶ Propiedad *Focused*: Indica si el control tiene el foco actualmente.
 - ▶ Método *Focus()*: "Mueve" el foco al objeto deseado.
- ▶ Orden de tabulación (Propiedad *TabIndex*)
 - ▶ En forma visual, desde el diseñador de formularios, es posible configurar el orden en el que el foco se irá moviendo por los controles.

MessageBox

- Para mostrar información o pedir intervención del usuario, es posible utilizar la clase `MessageBox`.
- Esta clase contiene métodos estáticos que permiten mostrar un cuadro de mensaje para interactuar con el usuario de la aplicación.
- Los parámetros se especifican a través de enumerados que facilitan la legibilidad del código, por ejemplo:
 - `MessageBoxButtons.AbortRetryIgnore`
 - `MessageBoxIcon.Error`
 - `MessageBoxDefaultButton.Button1`

Temario (1/2)

- Introducción a Windows Forms
- El diseñador de formularios
- El objeto Form
- **Controles**
 - Controles de Windows
 - Controles contenedores
 - Menú
- Diseño de Interfaz de Usuario
- Controles Extender Providers
- Herencia visual

Controles de Windows (1/3)

- ▶ Gran parte del éxito de una aplicación Windows consiste en elegir y manejar adecuadamente los controles que ofrece .NET.
- ▶ Entre los controles nativos se encuentran controles totalmente nuevos y versiones mejoradas de sus pares de .NET 1.1.
- ▶ Nuevos controles como el control BindingSource mejoran notablemente el enlace de datos provenientes de muy diferentes fuentes de datos.

Controles de Windows (2/3)

➤ MaskedEdit

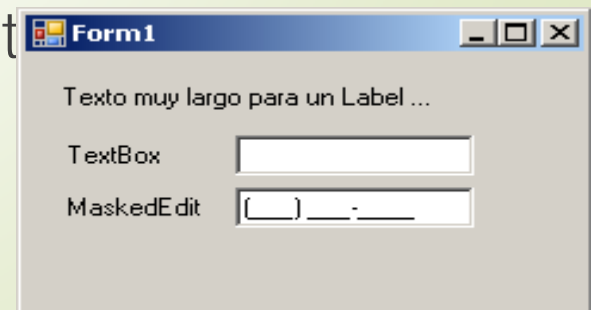
- Es un control que permite el uso de máscaras personalizadas para facilitar la entrada de datos.

➤ TextBox

- Cuadro de texto que, entre otras mejoras tiene la funcionalidad de auto completar.

➤ Label

- Si el texto ocupa más lugar que el largo del control, gracias a la nueva propiedad *AutoEllipsis* incorporada en .NET 2.0, el exedente se reemplaza automáticamente con t

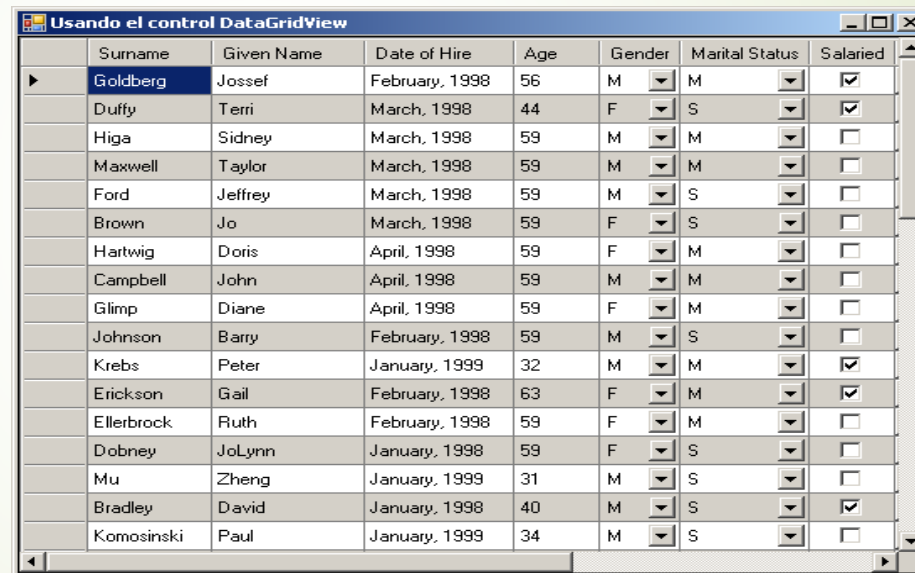


Controles de Windows (3/3)

► DataGridView

- Es una versión mejorada del DataGrid control de NET 1.1 con funcionalidad de modo "Virtual". Permite enlazar datos originados en una Base de Datos a medida que se necesitan.

DataGridView



Surname	Given Name	Date of Hire	Age	Gender	Marital Status	Salaried
Goldberg	Jossef	February, 1998	56	M	M	<input checked="" type="checkbox"/>
Duffy	Terri	March, 1998	44	F	S	<input checked="" type="checkbox"/>
Higa	Sidney	March, 1998	59	M	M	<input type="checkbox"/>
Maxwell	Taylor	March, 1998	59	M	M	<input type="checkbox"/>
Ford	Jeffrey	March, 1998	59	M	S	<input type="checkbox"/>
Brown	Jo	March, 1998	59	F	S	<input type="checkbox"/>
Hartwig	Doris	April, 1998	59	F	M	<input type="checkbox"/>
Campbell	John	April, 1998	59	M	M	<input type="checkbox"/>
Glimp	Diane	April, 1998	59	F	M	<input type="checkbox"/>
Johnson	Barry	February, 1998	59	M	S	<input type="checkbox"/>
Krebs	Peter	January, 1999	32	M	M	<input checked="" type="checkbox"/>
Erickson	Gail	February, 1998	63	F	M	<input checked="" type="checkbox"/>
Ellerbrock	Ruth	February, 1998	59	F	M	<input type="checkbox"/>
Dobney	JoLynn	January, 1998	59	F	S	<input type="checkbox"/>
Mu	Zheng	January, 1999	31	M	S	<input type="checkbox"/>
Bradley	David	January, 1998	40	M	S	<input checked="" type="checkbox"/>
Komosinski	Paul	January, 1999	34	M	S	<input type="checkbox"/>

TreeView

Utilizando la nueva propiedad DrawMode es posible sobrescribir la manera en que el sistema operativo "dibuja" cada nodo del árbol.

Controles Contenedores

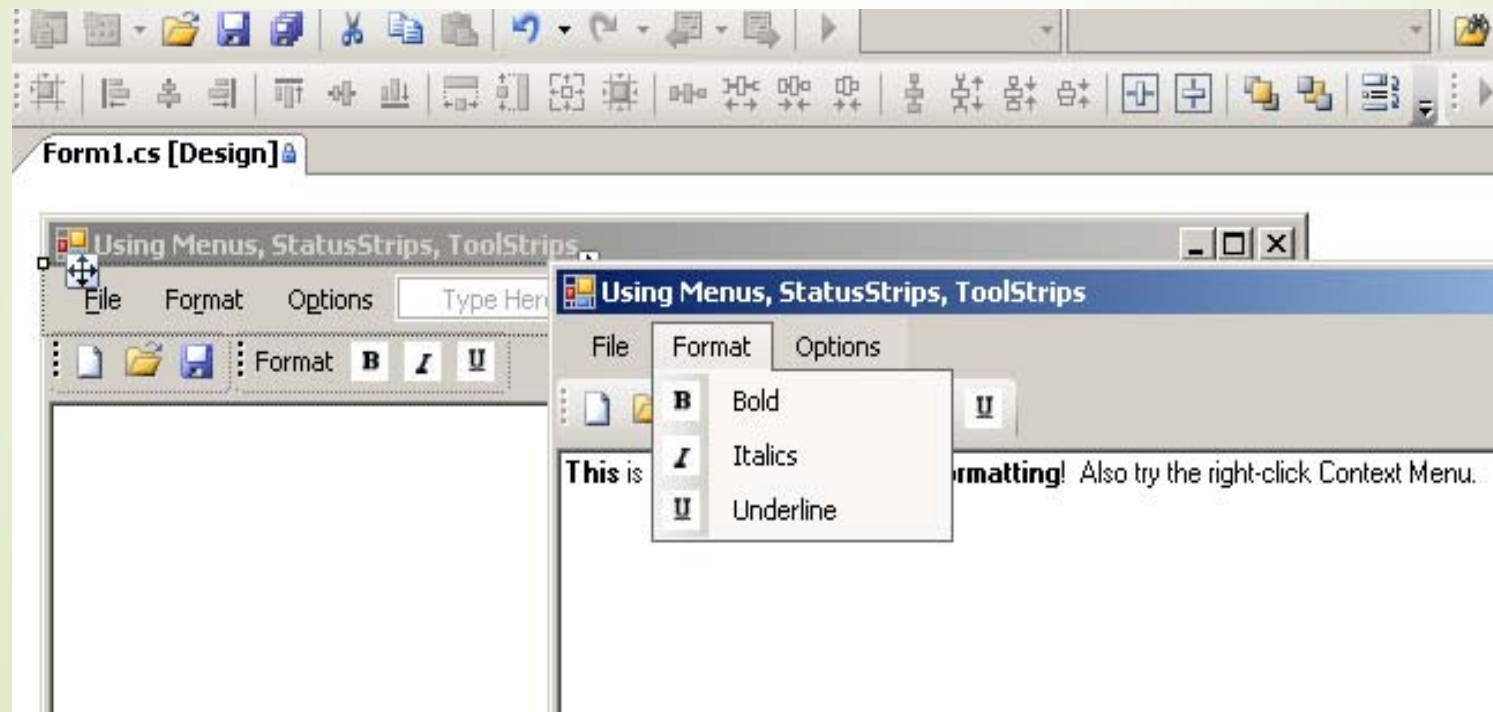
- Algunos controles como el propio Form, Panel o GroupBox heredan de la clase ContainerControl en lugar de hacerlo directamente de Control.
- Por este motivo, poseen una colección mediante la que se puede acceder a los controles que contiene.
- Sólo se puede acceder a los controles de nivel superior, no a todos los controles contenidos.

Menú (1/2)

- ▶ El nuevo control `MenuStrip` provee un sistema de menú para un formulario.
- ▶ `MenuStrip` es contenedor de objetos como `ToolStripMenuItem`, `ToolStripComboBox`, `ToolStripSeparator`, `ToolStripTextBox`.
- ▶ El control `ContextMenuStrip` representa un menú que será mostrado al usuario cuando presione el botón derecho del mouse. También puede contener los mismos controles que `MenuStrip`.
- ▶ Las propiedades `MergeAction` y `MergeIndex` del objeto `ToolStripItem` permiten controlar la manera en que los menú de dos diferentes ventanas se "mezclarán".

Menú (2/2)

- ▶ En la imagen se ve una aplicación que utiliza los controles MenuStrip y ToolStrip. En segundo plano se ve el diseñador de formularios.

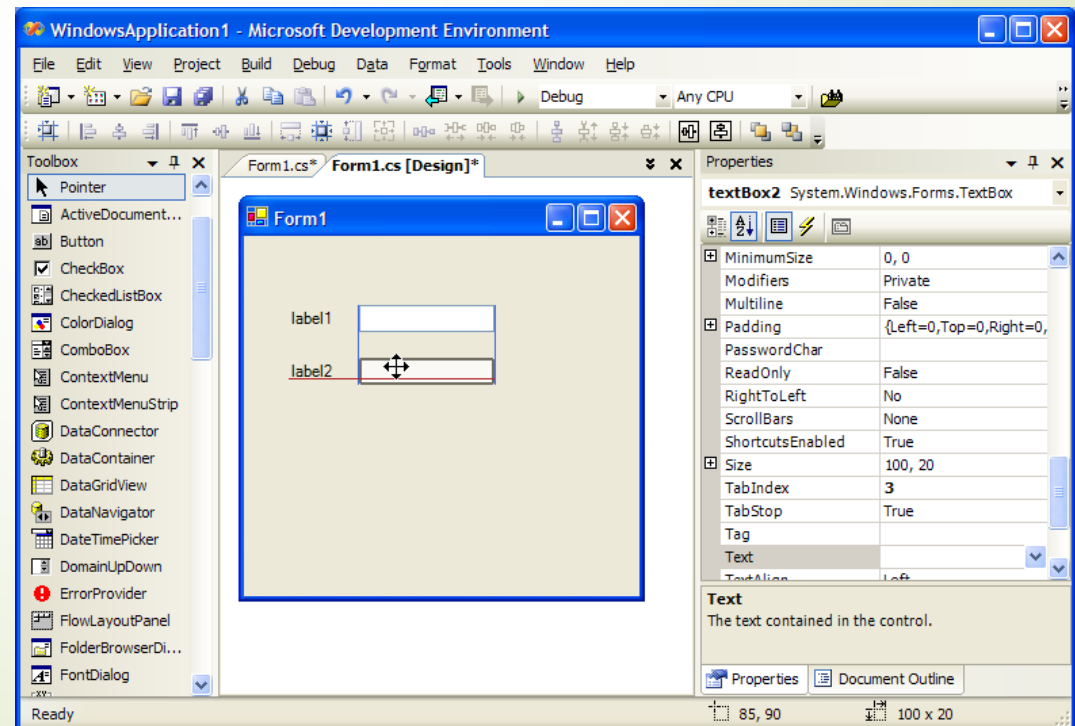


Temario (1/2)

- Introducción a Windows Forms
- El diseñador de formularios
- El objeto Form
- Controles
- Diseño de Interfaz de Usuario
 - Snaplines
 - Document Outline
 - Paneles de Layout
 - Anchor y Docking
- Controles Extender Providers
- Herencia visual

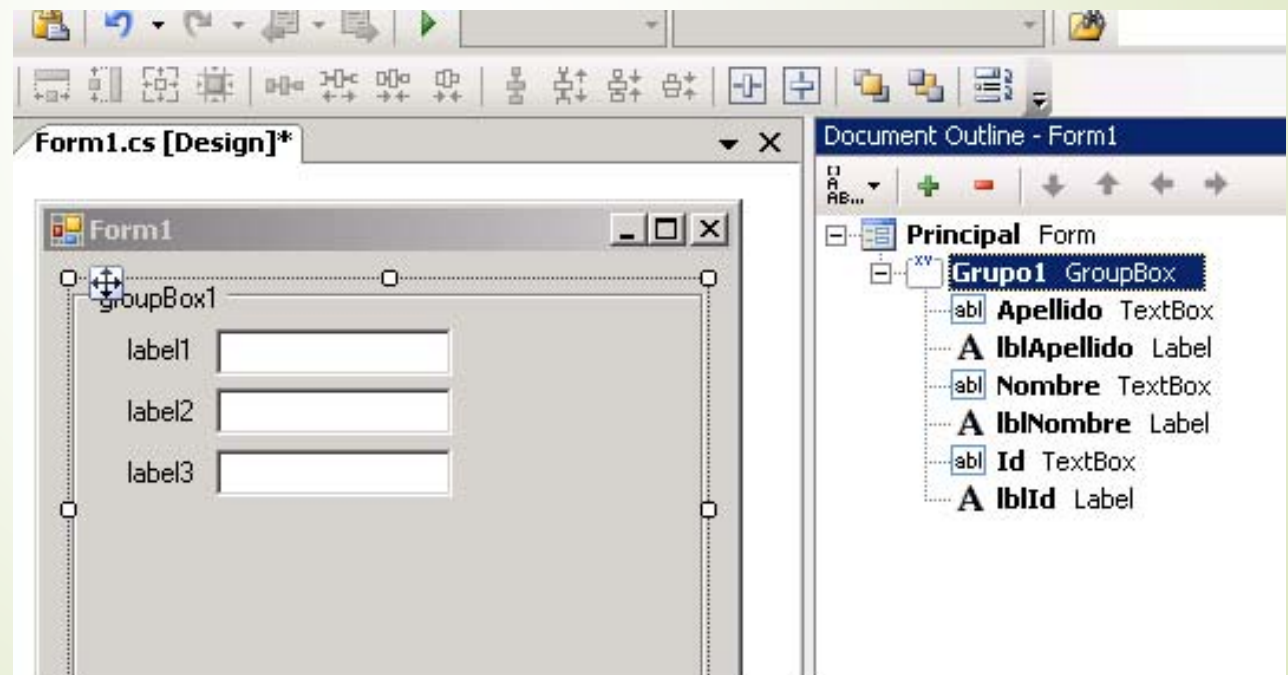
Snaplines

- Son líneas que se dibujan automáticamente en el diseñador de formularios al momento de posicionar el control.
- Ayudan a mantener la correcta distancia entre los controles y entre éstos y su contenedor.



Document Outline

- Mediante esta herramienta es posible ver la jerarquía de controles del formulario en forma de TreeView, y además editar el nombre de esos controles.



Layout Panels

- ▶ `TableLayoutPanel`

- ▶ Es similar a diseñar una tabla en un formulario HTML.
- ▶ Facilita la ubicación de los controles en escenarios de localización.
- ▶ Facilita la creación de interfaces redimensionables.

- ▶ `FlowLayoutPanel`

- ▶ Los controles contenidos “fluyen” como en el modo por defecto de un formulario HTML.

Anchor y Docking

➤ Anchor

- Automatiza el redimensionamiento y posicionamiento de los controles cuando se redimensiona el formulario.
- Los controles pueden **anclarse** contra cualquier combinación de los cuatro bordes del formulario.

➤ Docking

- La propiedad Dock (que exponen todos los controles Windows) permite **pegar** un control a alguno de los cuatro bordes del formulario.

Temario (1/2)

- Introducción a Windows Forms
- El diseñador de formularios
- El objeto Form
- Controles
- Diseño de Interfaz de Usuario
- Controles Extender Providers
- Herencia visual

Controles Extender Providers

- ▶ Son controles que, una vez colocados en un formulario, agregan nuevas propiedades a los otros controles existentes.
 - ▶ `ErrorProvider`: Permite asociar un error a un control mostrando un ícono que parpadea al lado de dicho control.
 - ▶ `HelpProvider`: Permite asociar a un control desde una simple cadena de texto un archivo Help que serán mostrados al presionar F1.
 - ▶ `ToolTip`: Es el clásico rectángulo que aparece asociado a un control y que es mostrado cuando el mouse se detiene sobre él.

Temario (1/2)

- Introducción a Windows Forms
- El diseñador de formularios
- El objeto Form
- Controles
- Diseño de Interfaz de Usuario
- Controles Extender Providers
- Herencia visual

Herencia Visual

- ▶ Dado que un formulario Windows es como cualquier otra clase .NET, es posible aplicar herencia.
- ▶ Al heredar de un formulario base, además de sus miembros, se heredan todos los controles que en él se encuentren.
- ▶ Permite entre otras cosas:
 - ▶ Unificar el diseño de las interfaces de usuario.
 - ▶ Reutilizar funcionalidad de formularios similares.

Temario (2/2)

- Configuración
- Diálogos comunes
- Enlace a datos
- Distribución de la aplicación
- Referencias

Configuración

- Las Propiedades Dinámicas permiten almacenar preferencias del usuario en archivos de configuración asociados a la aplicación.
- Estos valores pueden ser leídos y grabados tanto en diseño como en ejecución.
- Por cada valor que se almacena se puede definir el nombre, tipo de dato y alcance (usuario o aplicación).
- Es posible además enlazar (binding) propiedades dinámicas a controles del formulario.

Temario (2/2)

- Configuración
- Diálogos comunes
- Enlace a datos
- Distribución de la aplicación
- Referencias

Diálogos Comunes

- ▶ Los cuadros de diálogo comunes permiten interacción con el usuario para ejecutar acciones comunes como abrir un archivo, configurar la impresión, seleccionar un color del sistema, etc.
- ▶ Sólo basta configurar algunas propiedades e invocar su método `ShowDialog()`.
- ▶ Alguno de los controles que muestran estos diálogos son:
 - ▶ `ColorDialog`
 - ▶ `PrintDialog`
 - ▶ `SaveDialog`
 - ▶ `OpenDialog`

Temario (2/2)

- Configuración
- Diálogos comunes
- Enlace a datos
 - Colecciones
 - Objeto BindingSource
 - ADO.NET
- Distribución de la aplicación
- Referencias

Colecciones

- ▶ Enlace de un ComboBox a datos provenientes de un ArrayList:
 - ▶ Código C#

```
System.Collections.ArrayList Países =  
    new System.Collections.ArrayList();  
Países.Add("Argentina");  
Países.Add("Brasil");  
Países.Add("Uruguay");  
comboBox1.DataSource = Países;
```

Objeto DataSource

- ▶ El objeto DataSource permite el enlace de controles a datos provenientes de fuentes de datos (DataSource) de tres tipos
 - ▶ DataBase: Crea internamente un dataset.
 - ▶ WebService: Crea una referencia web a un servicio que es el que proporciona los datos
 - ▶ Object: Utiliza una clase de negocios como fuente de datos creando automáticamente una colección de elementos de esa clase.
- ▶ Usándolo junto a un control DataBindingNavigator y un DataGridView conforman un formulario de ABM sin escribir código alguno.

ADO.NET

- ▶ Además de utilizar el objeto BindingSource, es posible enlazar datos utilizando ADO.NET de manera directa.
 - ▶ Ejemplo en C#

```
using( SqlConnection cn = new SqlConnection("....") )
{
    cn.Open();
    SqlDataAdapter da = new SqlDataAdapter("Select *
        from Employee", cn);
    DataTable dt = new DataTable();
    da.Fill(dt);
    this.dataGridView1.DataSource = dt;
}
```

Temario (2/2)

- Configuración
- Diálogos comunes
- Enlace a datos
- Distribución de la aplicación
 - Click Once
- Referencias

Distribución de la aplicación

- ▶ La distribución de una aplicación Windows involucra varios pasos de cierta complejidad dependiendo de sus requerimientos.
- ▶ .NET 2.0 incorpora ClickOnce, una tecnología que permite la distribución de la aplicación, versionado y rollback, entre otras funciones.
- ▶ Es posible:
 - ▶ Decidir donde será instalada físicamente.
 - ▶ Especificar la frecuencia de revisión de actualizaciones.
 - ▶ Forzar actualizaciones críticas.
 - ▶ Publicar actualizaciones en Servidores Web, Servidores de archivo (File Servers) o Servidores FTP.

Temario (2/2)

- Configuración
- Diálogos comunes
- Enlace a datos
- Distribución de la aplicación
- Referencias

Referencias Adicionales

► Windows Forms:

- <http://windowsforms.net>
- <http://samples.gotdotnet.com/quickstart/winforms/>
- <http://msdn.microsoft.com/library/en-us/cpref/html/frlrfsystemwindowsforms.asp>
- <http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vbcon/html/vbconintroductiontowfcforms.asp>

Referencias Adicionales

- ▶ Smart Client Developer Center

- ▶ <http://msdn.microsoft.com/smartclient/understanding/windowsforms/default.aspx>

- ▶ ClickOnce:

- ▶ <http://msdn.microsoft.com/msdnmag/issues/04/05/ClickOnce/>