

# **Tema 15 – Componentes personalizados**

Formación específica, cursos verano 2008  
ETS de Informática Aplicada  
Universidad Politécnica de Valencia

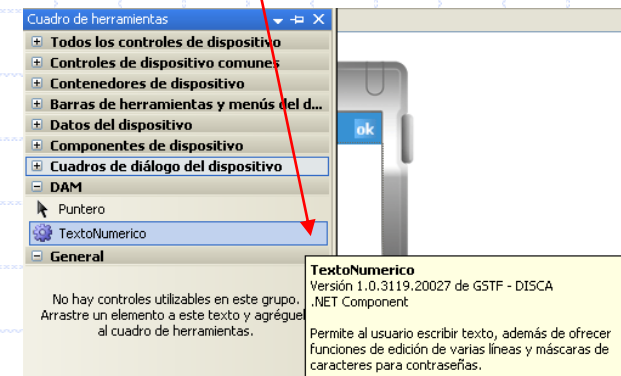
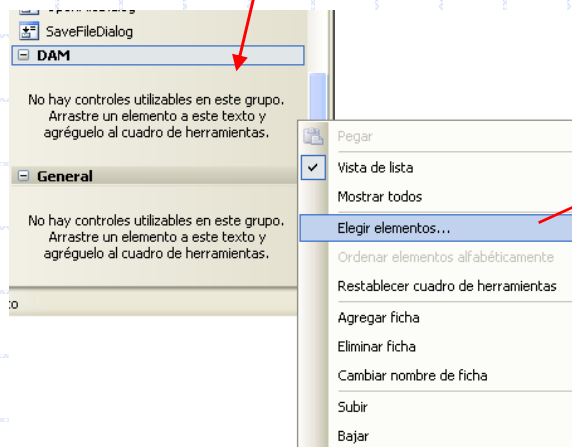
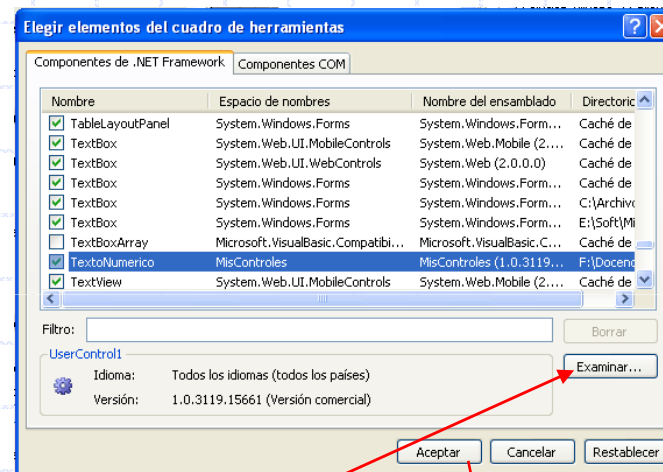
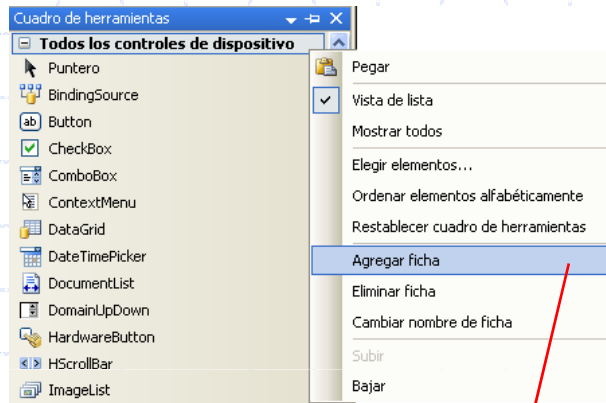
# Motivación

- ◆ .NET CF ofrece controles para gestionar diferentes tipos de datos y escenarios de interacción
  - Subconjunto del *full framework*
- ◆ Desarrollar controles para ampliar las funcionalidades ofrecidas
  - Extender los controles ya existentes
  - Crear un nuevo control desde cero
  - Integrar los nuevos controles en el Visual Studio

# Extender un control

- ◆ Extender la funcionalidad y capacidades de un control ya existente
  - Todos los controles derivan de *System.Windows.Forms.Control*
- ◆ Procedimiento
  - Crear un nuevo proyecto → *Control library*
  - Añadir un nuevo elemento → *Control de usuario*
  - Editar el código para sustituir la clase que hereda (*Control*) por la que se desea heredar
  - El resultado final es una nueva DLL


# Importar componentes de una DLL



# Nuevos componentes

- ◆ Agregar un nuevo elemento a la librería → *Control de usuario*
- ◆ Editar la GUI como si de un *Form* se tratara (atributos, métodos, eventos...)

```
public event EventHandler MiEvento;  
  
protected virtual void OnMiEvento(object sender, EventArgs e) {  
    if (this.MiEvento != null) MiEvento(sender, e);  
}  
private void okButton_Click(object sender, EventArgs e) { //por ejemplo  
    OnMiEvento(this, EventArgs.Empty);  
}
```



# Nuevos componentes

- ◆ Agregar un nuevo elemento a la librería → *Archivo de atributos en tiempo de diseño*
  - Fichero XML (.xmta) que describe atributos propios del componente
  - El Visual Studio incluye *Microsoft IntelliSense* como ayuda

# Nuevos componentes

## ◆ Fichero .xmta

### ■ Elemento *Classes*

- ◆ Un elemento *Class* para cada componente
  - *Espacio de nombres.Clase* en el atributo *Name*
  - Añadir subelementos por cada propiedad y evento que se desee describir

```
<?xml version="1.0" encoding="utf-16"?>
<Classes xmlns="http://schemas.microsoft.com/VisualStudio/2004/03/SmartDevices/XMTA.xsd">
  <Class Name="MisControles.PinControl">
    <DesktopCompatible>true</DesktopCompatible>
  </Class>
</Classes>
```

# Nuevos componentes

## ◆ Fichero .xmta

- *DesktopCompatible* – indica que puede aparecer dibujado en el entorno de desarrollo

```
<DesktopCompatible>true</DesktopCompatible>
```

- **Propiedades** – Aparecen por defecto en el apartado *Misc*

```
<Property Name="PasswordChar">  
  <Category>Behavior</Category>  
  <DefaultValue>  
    <Type>  
      char  
    </Type>  
    <Value>  
    </Value>  
  </DefaultValue>  
  <Description>  
    Carácter que debe mostrarse al escribir contraseñas  
  </Description>  
</Property>
```

# Nuevos componentes



## Fichero .xmta

- *DefaultEvent* – evento gestionado al hacer doble click sobre el componente

```
<DesktopCompatible>true</DesktopCompatible>
```

- Propiedades y eventos

```
<Property Name="PasswordChar">
```

```
  <Category>Behavior</Category>
```

```
  <DefaultValue>
```

```
    <Type>char</Type>
```

```
    <Value></Value>
```

```
  </DefaultValue>
```

```
  <Description>
```

```
    Carácter que debe mostrarse al escribir contraseñas
```

```
  </Description>
```

```
</Property>
```

```
<DefaultEvent>PinChanged</DefaultEvent>
```

```
<Event Name="PinChanged">
```

```
  <Category>Action</Category>
```

```
  <Description>Ocurre cuando el usuario introduce un nuevo PIN</Description>
```

```
</Event>
```

# Nuevos componentes

- ◆ También pueden observarse los atributos personalizados en el diagrama de clases

