

# Menús

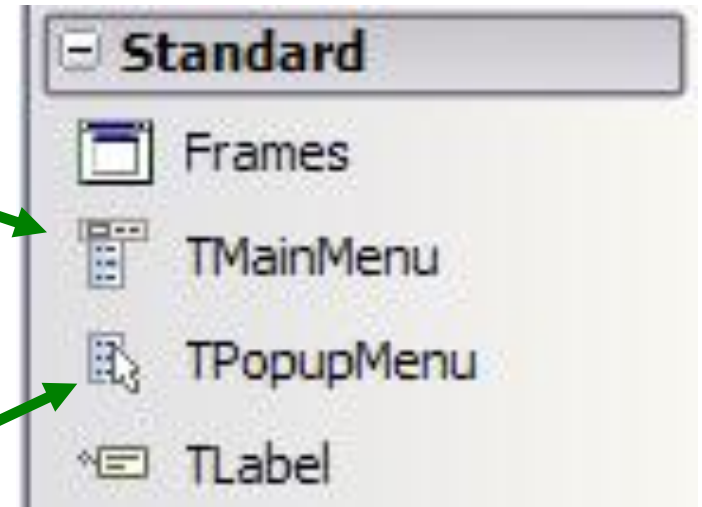
✳ Delphi posee dos tipos de menús

## MainMenu

Menú desplegable. Es el menú principal del formulario

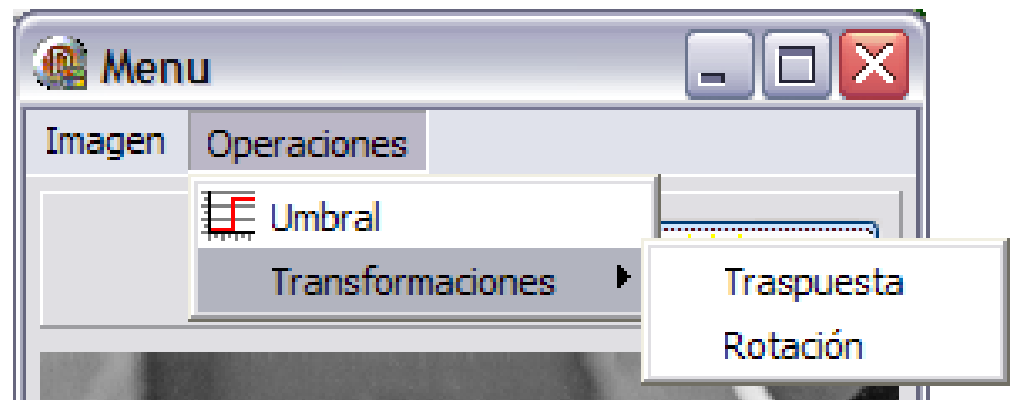
## PopupMenu

Menú que se despliega cuando se presiona el botón derecho del mouse



# Ejemplo Menu.dpr

⚡ Comenzaremos diseñando el siguiente menú desplegable



# Componente MainMenu

## ✦ Propiedad Items

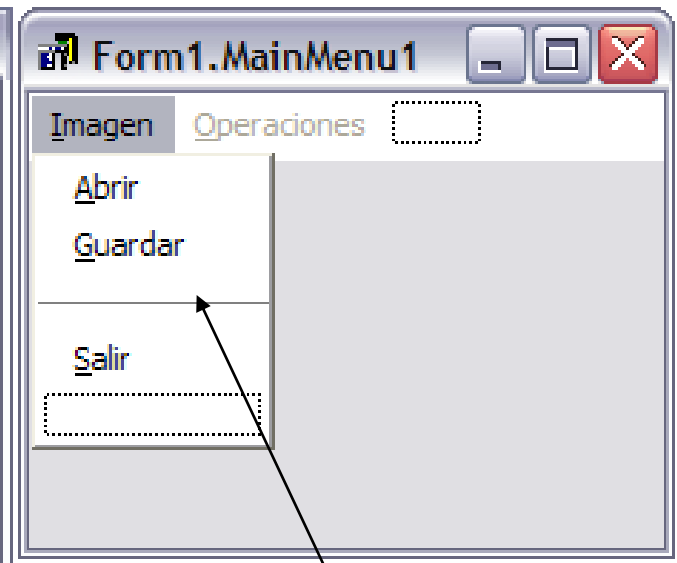
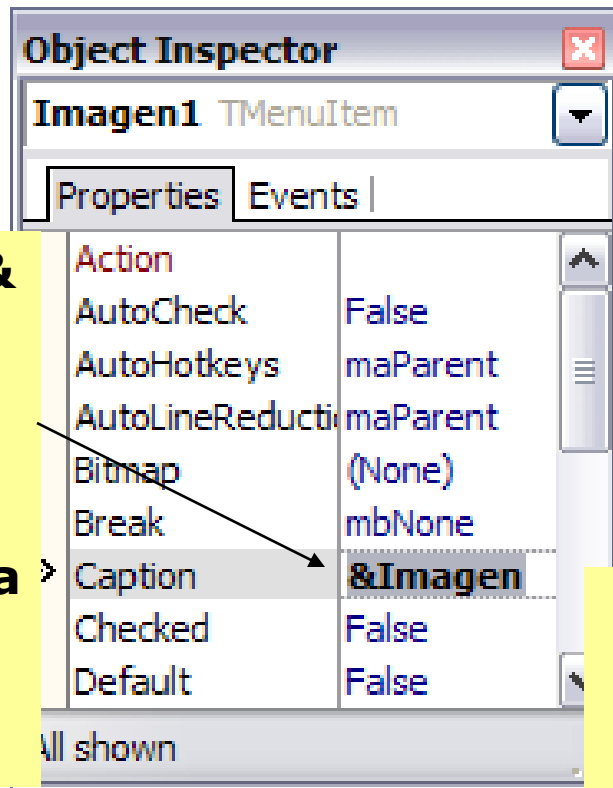
- Representa cada una de las opciones del menú.
- Count (propiedad de items) indica la cantidad de opciones que tiene actualmente el menú.

✦ Para acceder al diseñador de menús basta con **dar doble click sobre la componente MainMenu** insertada en el formulario.

# Edición de un menú

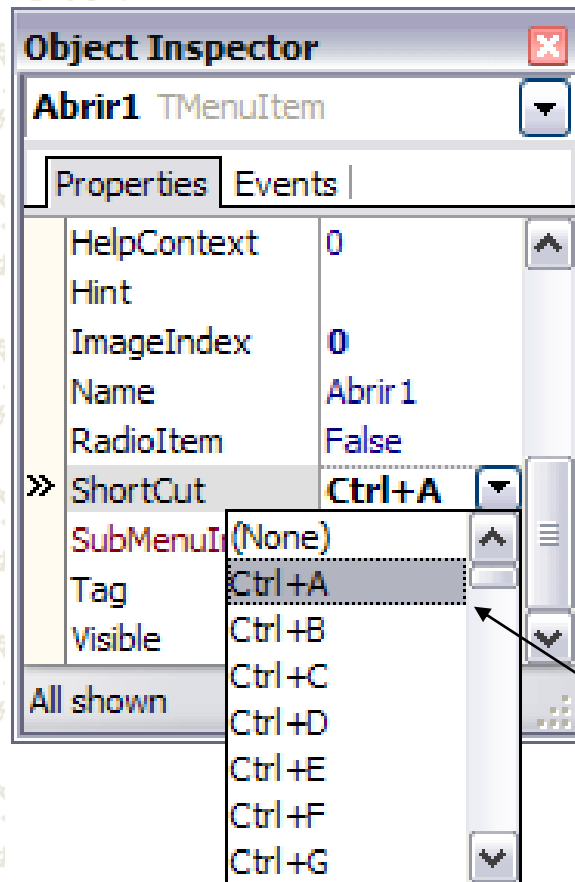
Ingresa el carácter **&** delante de la letra que quiera utilizar como acceso rápido.

Funciona presionando **ALT y la letra** que aparece subrayada.



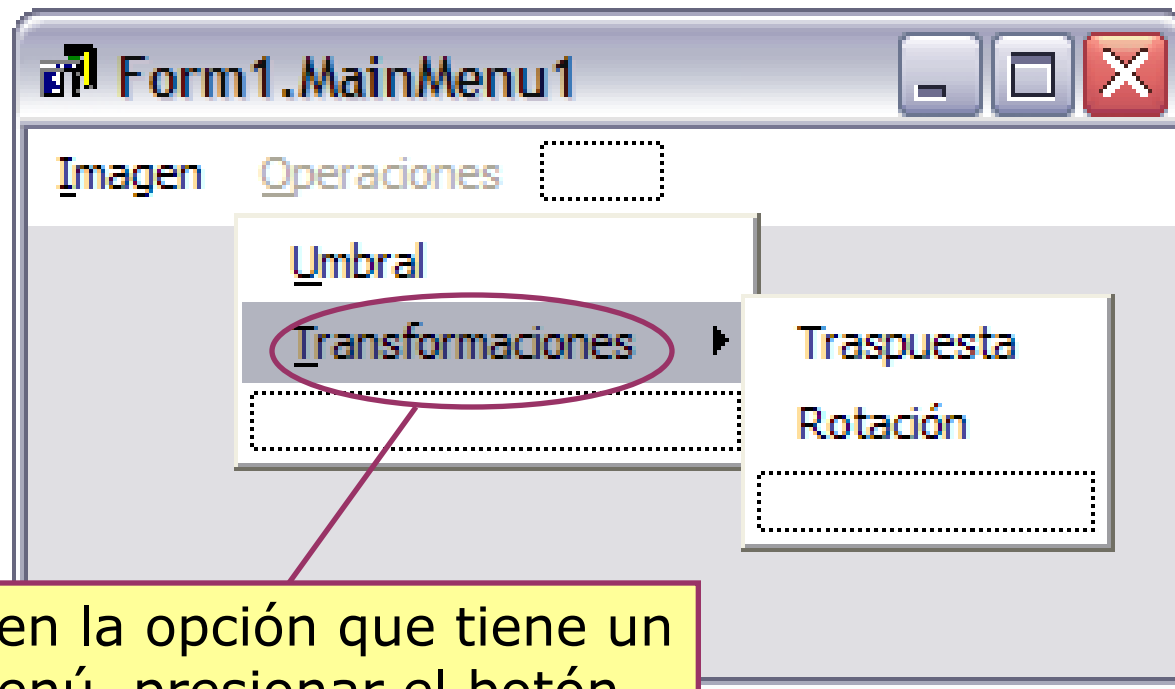
Para que aparezca esta línea sólo debe tipear un guión en el caption. Note que la línea no puede recibir el control.

# Tecclas de acceso rápido



La propiedad ShortCut permite asociar a cada opción un acceso mediante CTRL y una letra.

# Creación de submenús



Parado en la opción que tiene un submenú, presionar el botón derecho del mouse y elegir **Create Submenu.**

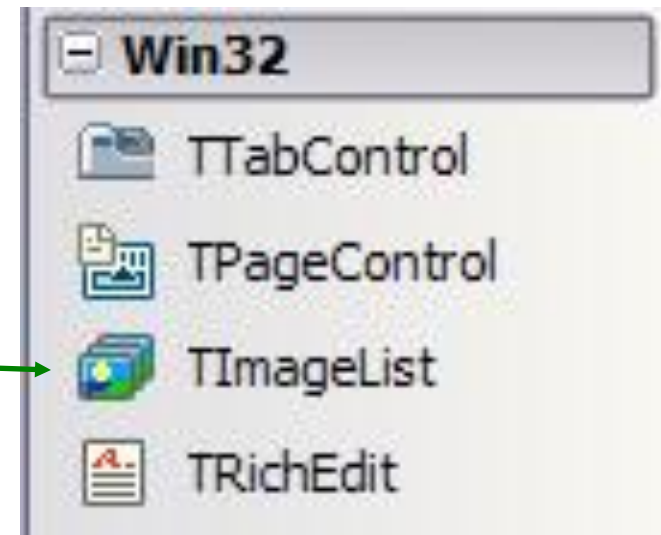
# Opciones con imágenes

- ✦ Hay dos formas de asociar una imagen a una opción del menú
  - Usando la propiedad **Bitmap** que permite ingresar una imagen.
  - Usando una lista de imágenes mediante una componente **ImageList**.
- ✦ La primer opción suele utilizarse cuando se trata de muy pocas imágenes pero en general es más flexible y eficiente el uso de la componente ImageList.

# Componente ImageList

## ImageList

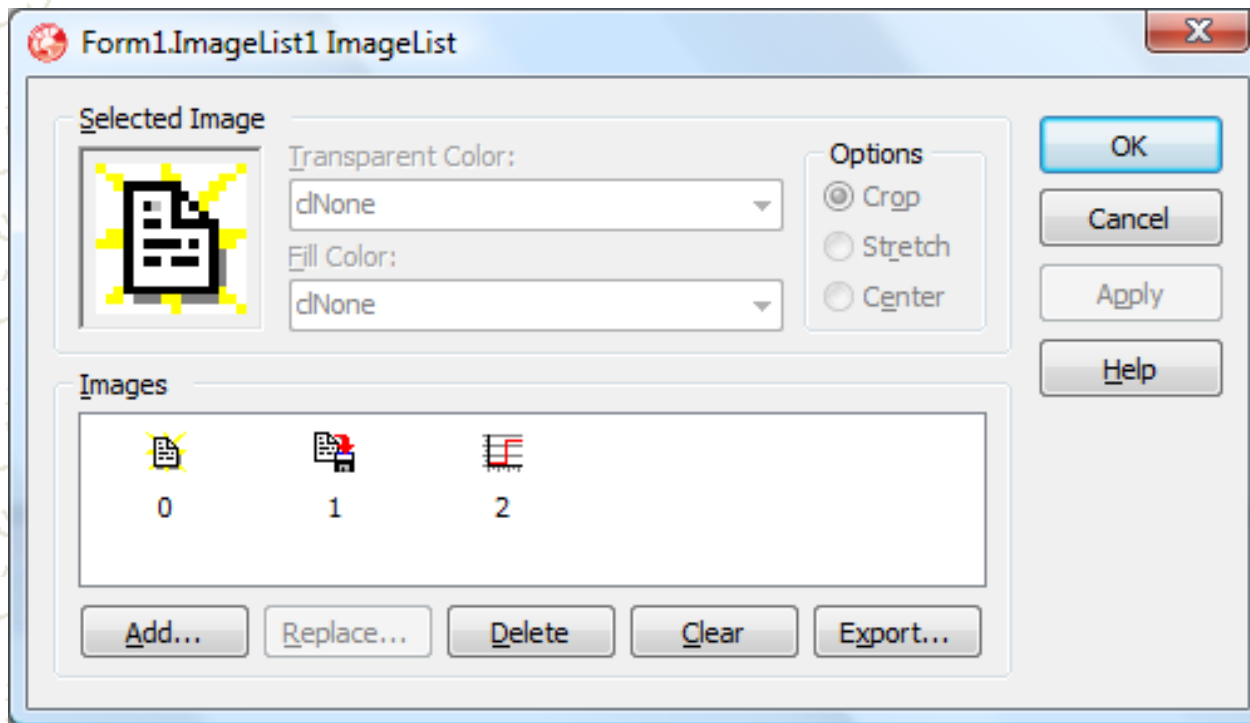
Contiene una colección de imágenes que pueden ser accedidas mediante un índice.



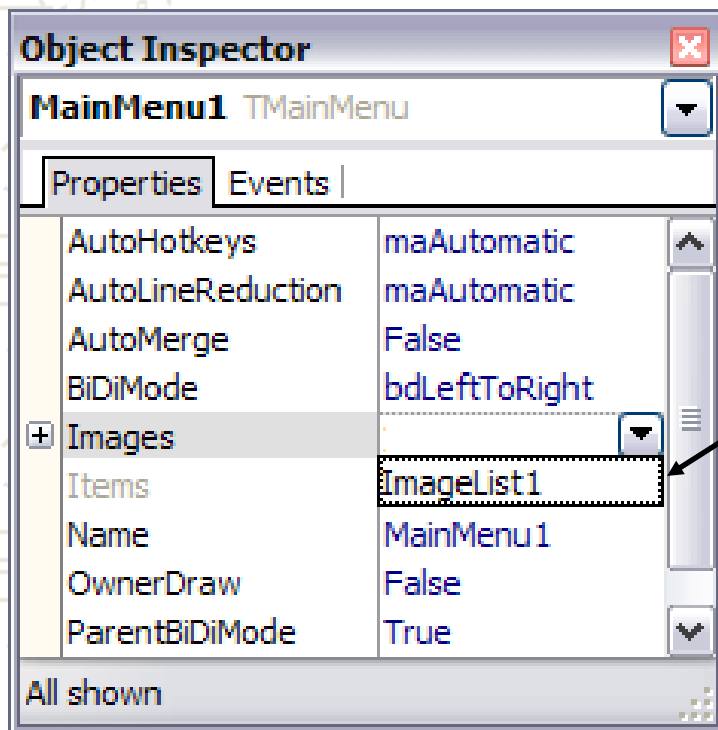


# Componente ImageList

✚ Dando doble click sobre la componente se accede a la lista de imágenes:

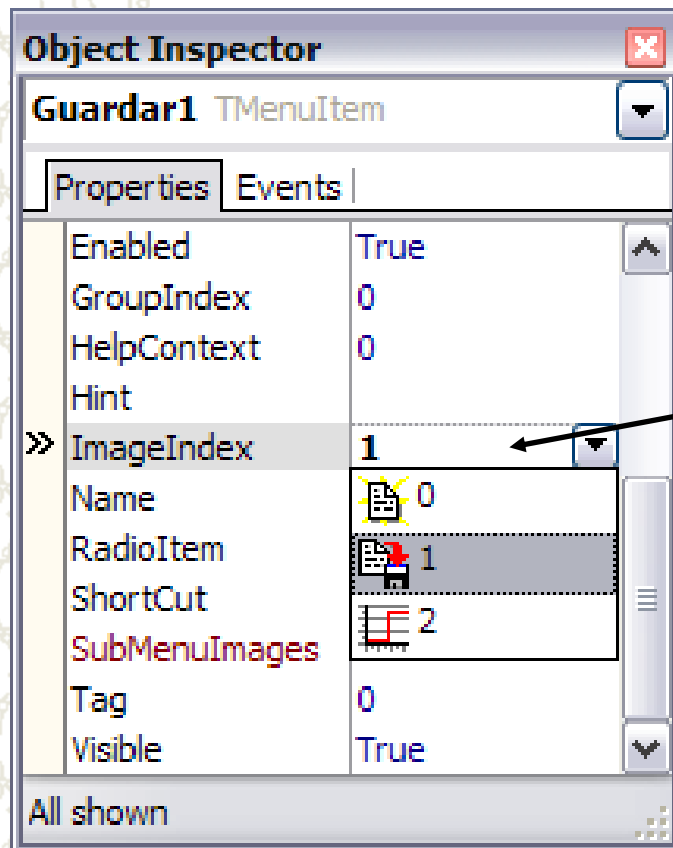


# Asociando las imágenes al menú



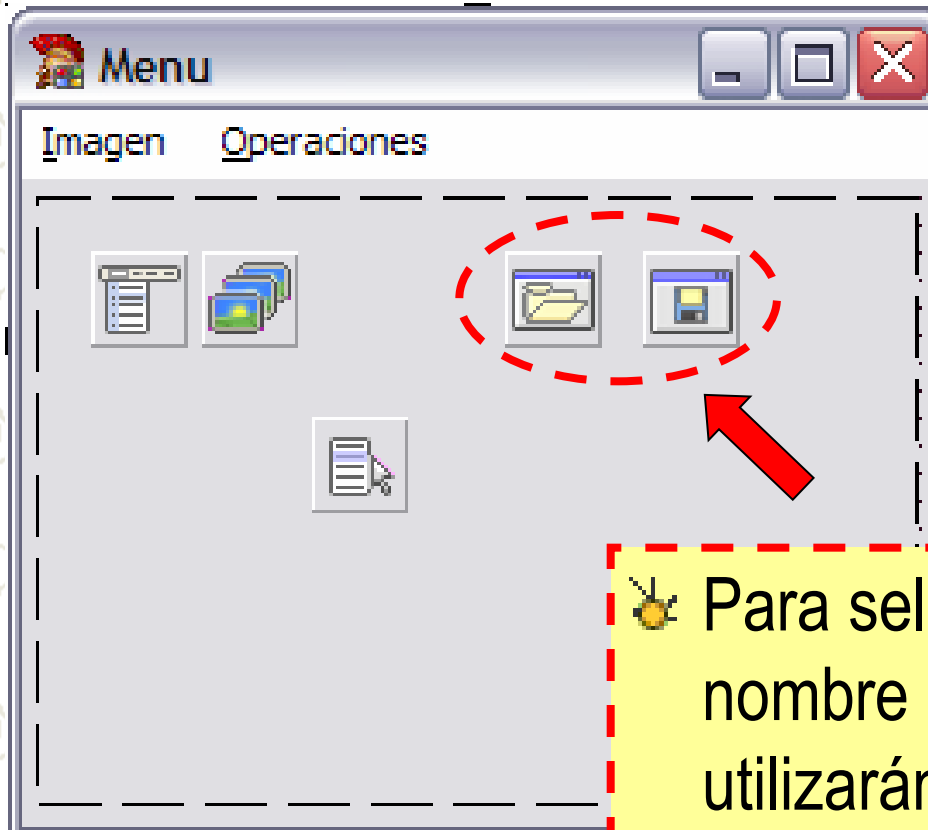
La propiedad Images del MainMenu indica de donde se obtienen las imágenes. Asociar aquí el ImageList anterior.

# Asociando las imágenes al menú



Finalmente modificar la propiedad **ImageIndex** de c/u de las opciones del menú que tengan un dibujo asociado para que contengan el número correspondiente. La 1er.imagen de la lista tiene el nro.0

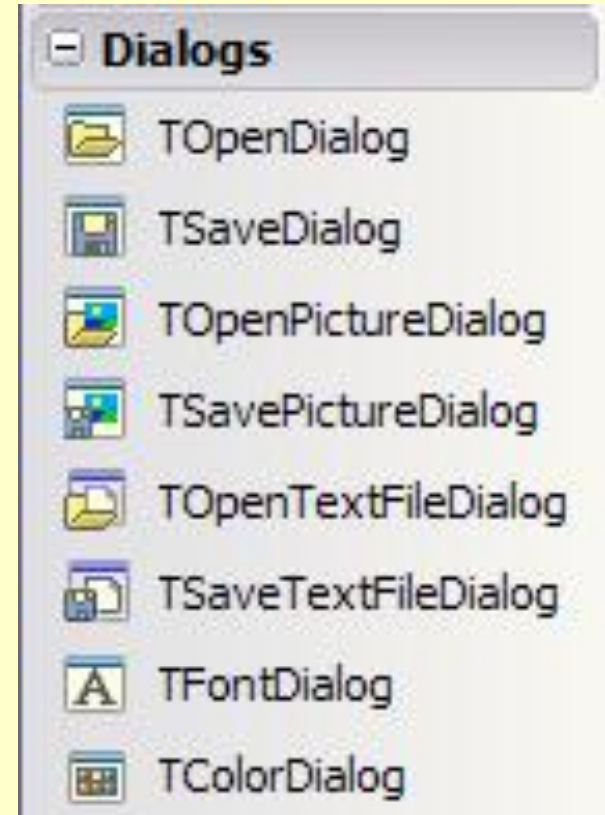
# Ejemplo Menu.dpr



✎ Para seleccionar el nombre del archivo se utilizarán componentes de la paleta Dialogs

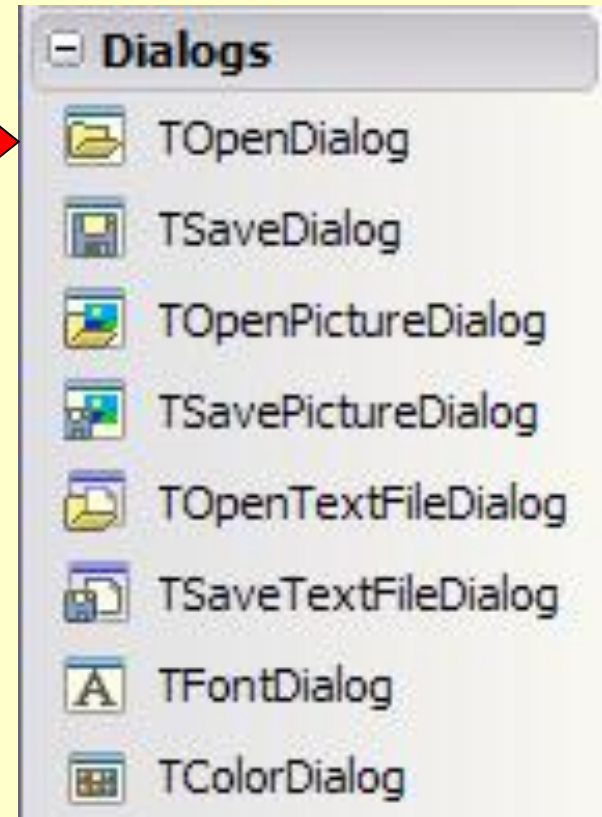
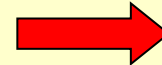
# Paleta Dialogs

- ✱ Son componentes no visuales que no aparecen en ejecución a menos que se llame al método **Execute**.
- ✱ Casi todas las componentes de esta paleta son modales, es decir, que cuando se visualizan no retornan el control hasta que se cierran.

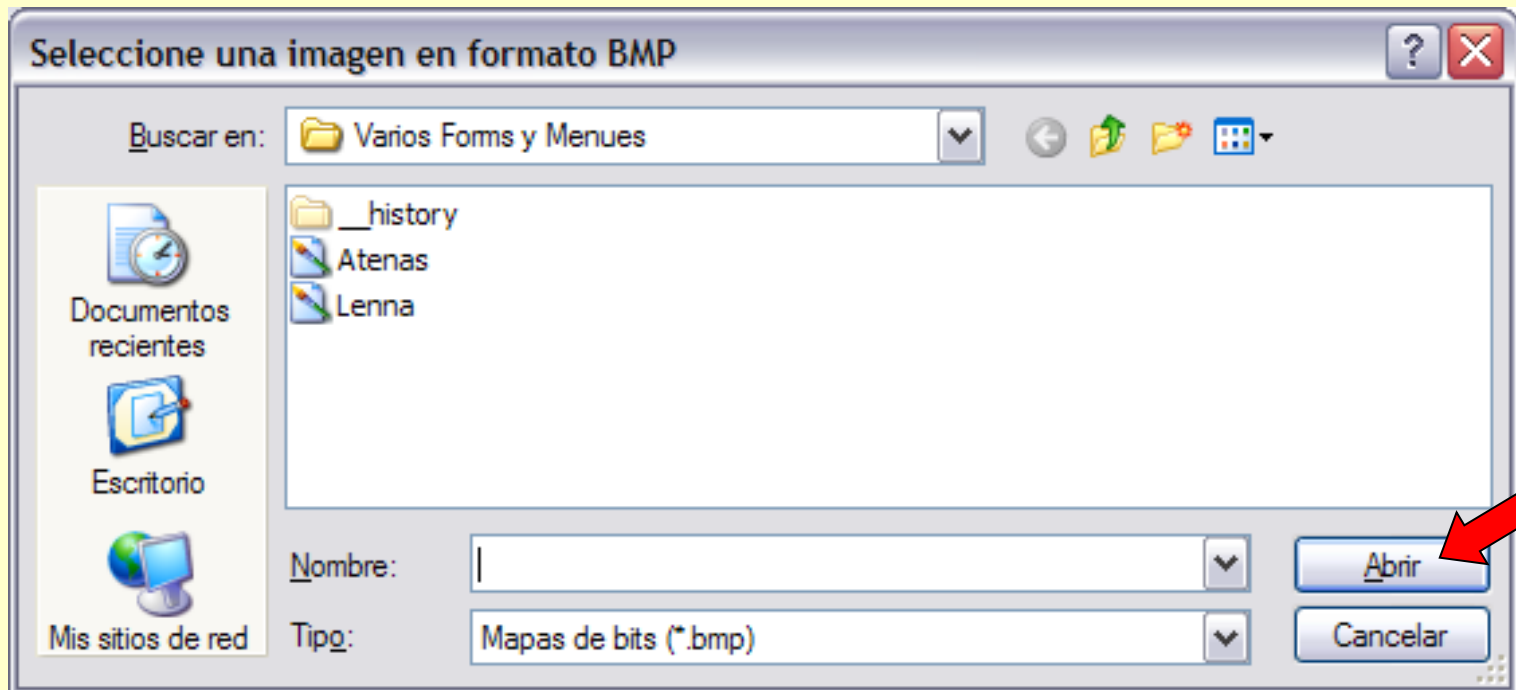


# OpenDialog

☀ Muestra una ventana de diálogo para seleccionar o ingresar un nombre de archivo.



# OpenDialog



✦ Cuando el usuario presiona *Abrir*, la propiedad **FileName** contiene el nombre del archivo.

# Ejemplo Menu.dpr

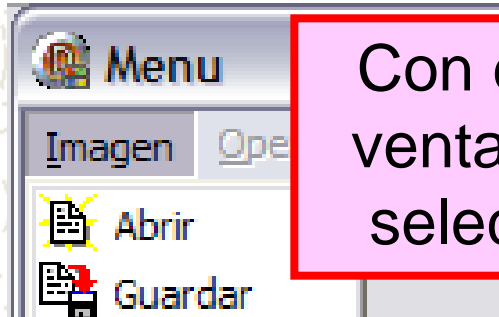


- 💡 Codifique lo siguiente en el OnClick de la opción Abrir del menú

```
procedure TForm1.Abrir1Click(Sender: TObject);  
begin  
    OpenDialog1.Execute;  
    Image1.Picture.LoadFromFile(OpenDialog1.FileName);  
end;
```



# Ejemplo Menu.dpr



Con el método **Execute** se abre la ventana de diálogo para ingresar o seleccionar el nombre del archivo

```
procedure TForm1.Abrir1Click(Sender: TObject);  
begin  
    OpenFileDialog1.Execute;  
    Image1.Picture.LoadFromFile(OpenDialog1.FileName);  
end;
```

Nombre seleccionado o ingresado

# Ejemplo del uso de OpenFileDialog

☛ ¿Qué problemas puede presentar este código ?

```
procedure TForm1.Abrir1Click(Sender: TObject);  
begin  
    OpenFileDialog1.Execute;  
    Image1.Picture.LoadFromFile(OpenDialog1.FileName);  
end;
```

☛ Antes de intentar abrir el archivo debemos asegurarnos de que el archivo exista.

- Opciones de validación dentro del componente OpenFileDialog
- No tomar la respuesta del usuario a menos que haya salido de la ventana de diálogo con el botón de *Abrir*. Esto nos asegura que la validación fue realizada.

# OpenDialog

## ✶ Propiedad **Options**

- Permiten validar el nombre de archivo indicado.
- Considere que además de indicarlo mediante selección, el usuario puede tipearlo directamente.
- Opciones que deberían setearse en TRUE
  - **OfPathMustExist** : el camino indicado debe existir
  - **OfFileMustExist** : el archivo debe existir.
  - Ver las que faltan

✶ El usuario NO podrá cerrar la ventana con el botón de *Abrir* si las opciones seteadas en TRUE no se verifican.

# OpenDialog

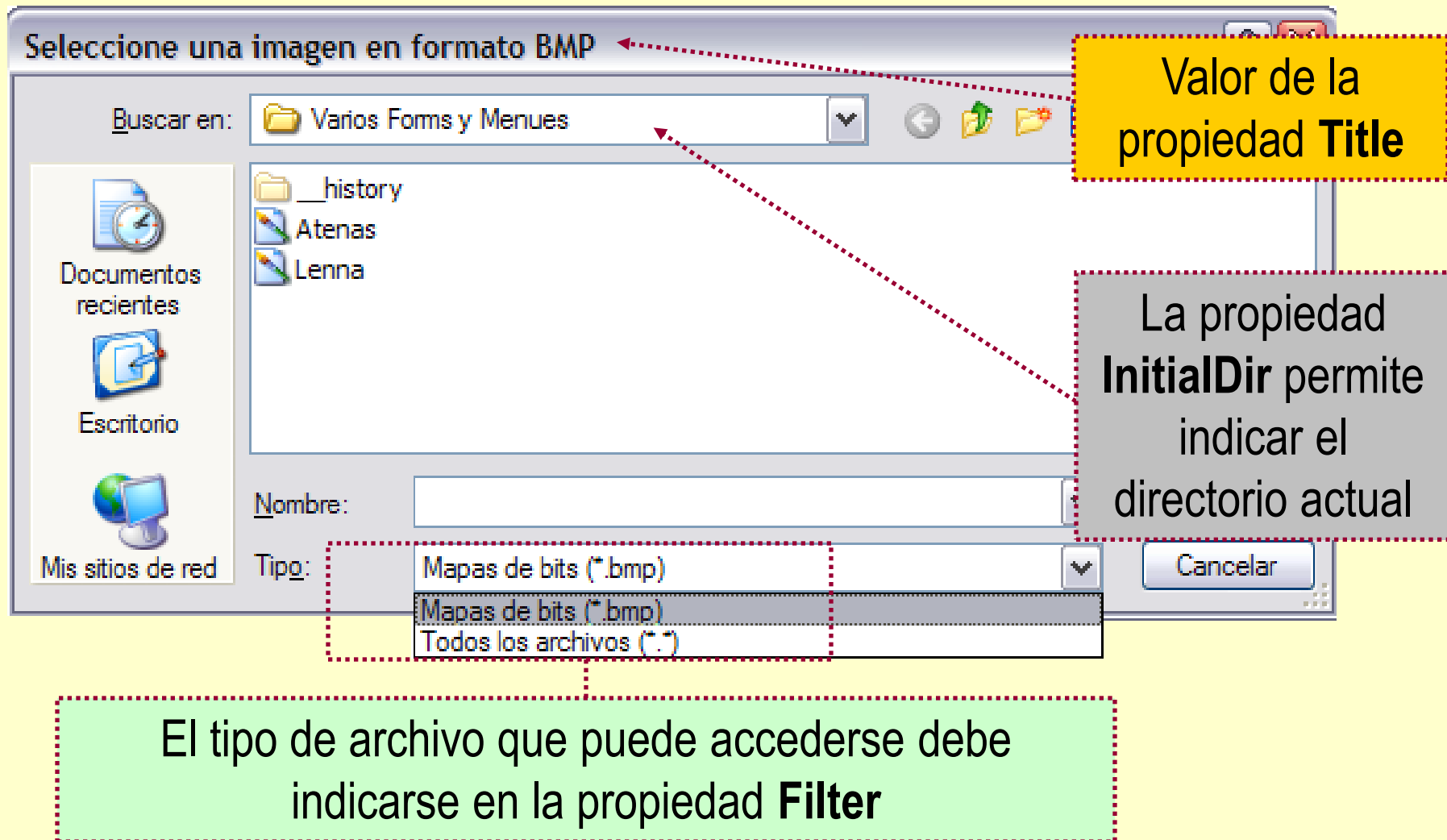
- ✱ Sólo resta saber la forma en que el usuario cerró la ventana de diálogo.
- ✱ El nombre del archivo sólo debe utilizarse si cumple con las opciones de validación (botón *Abrir*).
- ✱ El método **Execute** retorna
  - TRUE si se cerró la ventana mediante el botón *Abrir*
  - FALSE si se cerró con el botón *Cancelar*.

# Opción Abrir

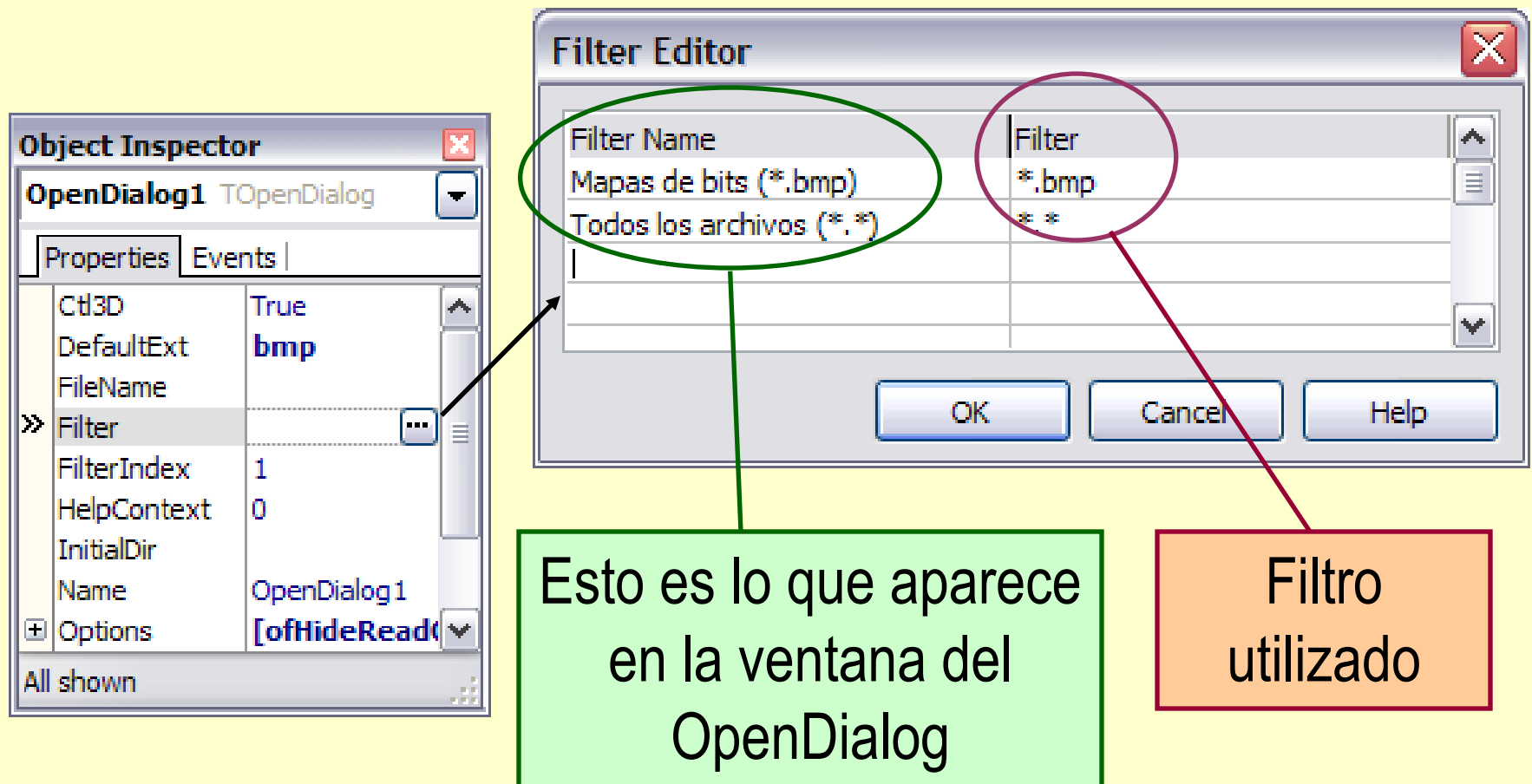
```
procedure TForm1.Abrir1Click(Sender: TObject);  
begin  
    if OpenFileDialog1.execute then  
    begin  
        Image1.Picture.LoadFromFile(OpenDialog1.FileName);  
        Operaciones1.Enabled := true;  
    end;  
end;
```

Este nombre se puede obtener  
del Inspector de Objetos

# Propiedades de OpenFileDialog

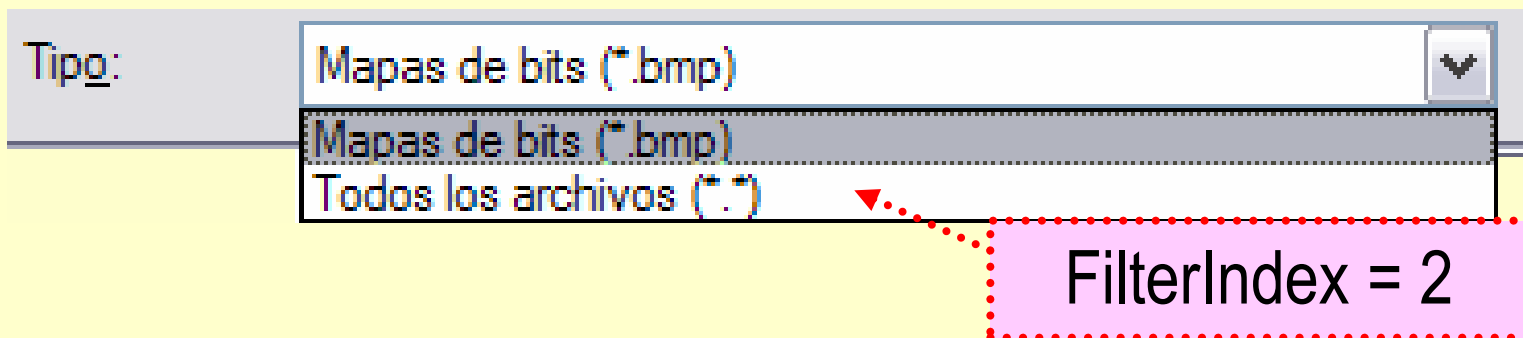


# OpenDialog : propiedad Filter



# OpenDialog : Otras propiedades

- ✦ **DefaultExt** : indica la extensión del archivo por defecto. Se agrega al nombre del archivo cuando el usuario no la especifica.
- ✦ **FilterIndex** : Es un número entero que indica el filtro a seleccionar cuando se abre la ventana de diálogo.
  - El primer filtro lleva valor 1.
  - Si su valor está fuera de rango se tomará el primer filtro definido.





# Ejemplo Menu.dpr

- Ejecute su aplicación y verifique si funciona correctamente.



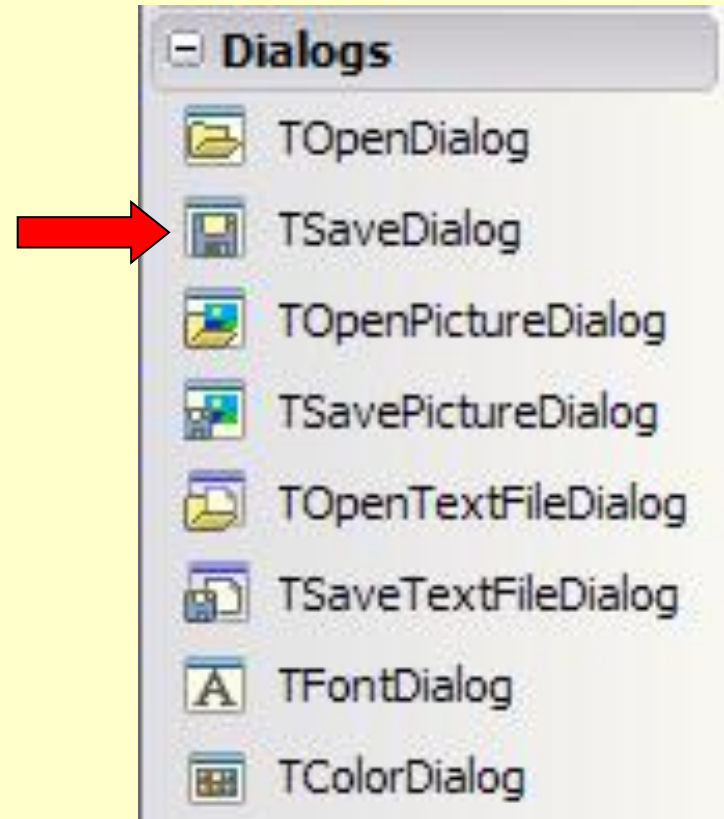
# Ejemplo Menu.dpr

- ✦ Para implementar la opción Guardar del menú utilizaremos el componente **SaveDialog**

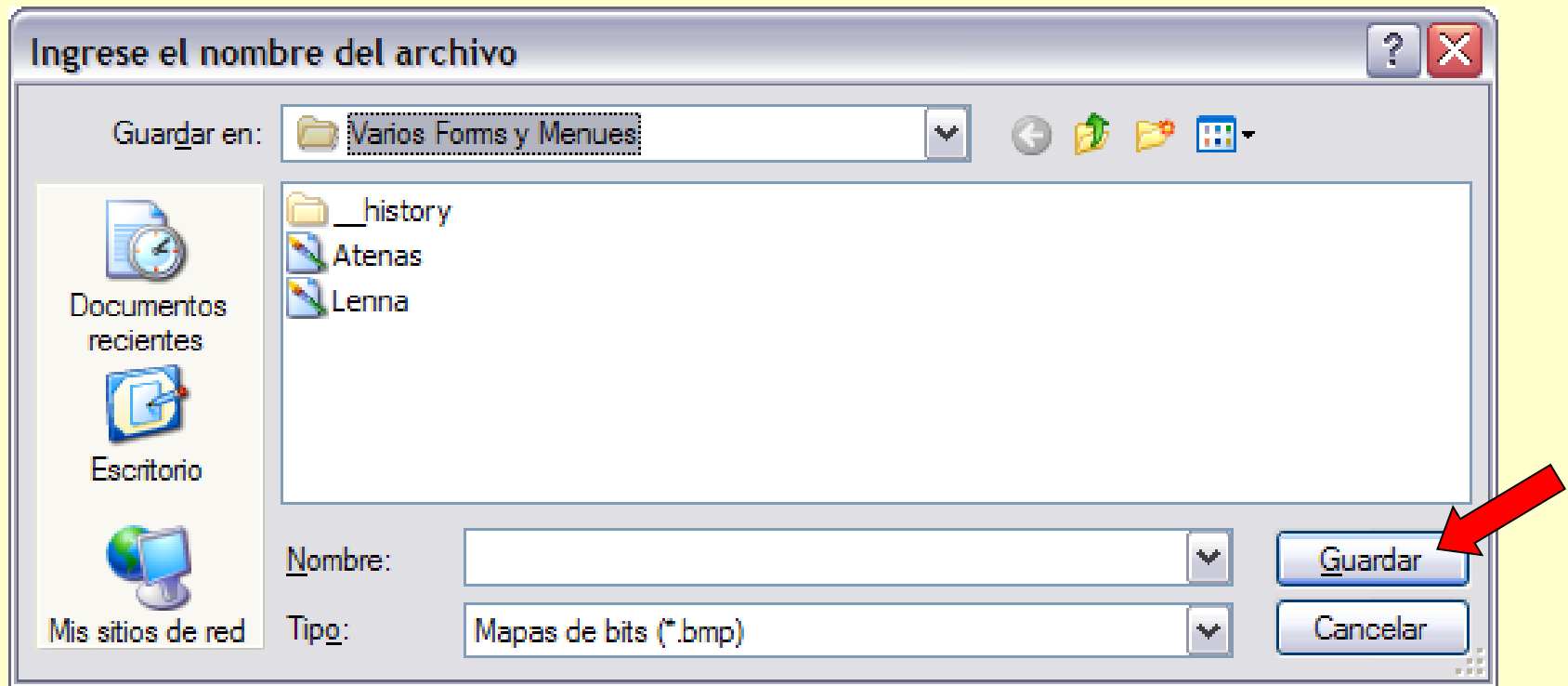


# SaveDialog

☀ Muestra una ventana de diálogo para seleccionar o ingresar un nombre de archivo para salvar.



# SaveDialog



💡 Cuando el usuario presiona *Guardar*, la propiedad **FileName** contiene el nombre del archivo.

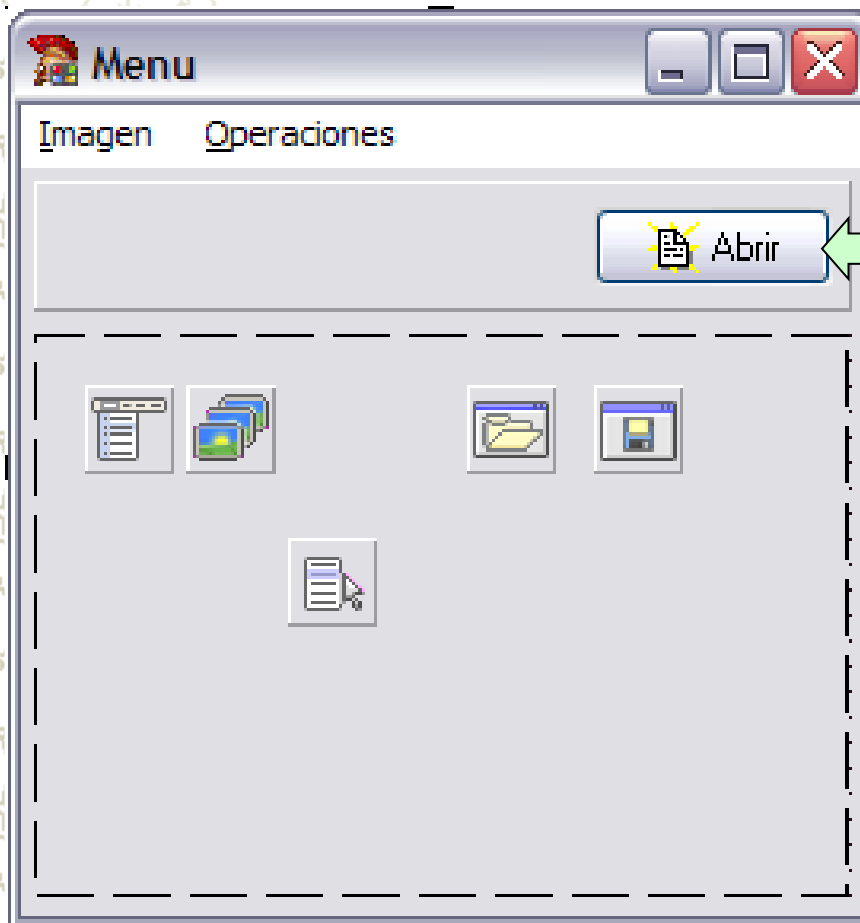
# SaveDialogs

- ✦ Tiene las mismas propiedades que el componente OpenFileDialog.
- ✦ También utiliza el método `execute` para visualizar la ventana. Este método retorna `TRUE` si se cerró la ventana mediante el botón *Guardar* y `FALSE` en caso contrario.
- ✦ En la propiedad **Options** habría que considerar:
  - **OfPathMustExist**
  - **OfOverwritePrompt** : pregunta antes de sobrescribir.

# Opción Guardar

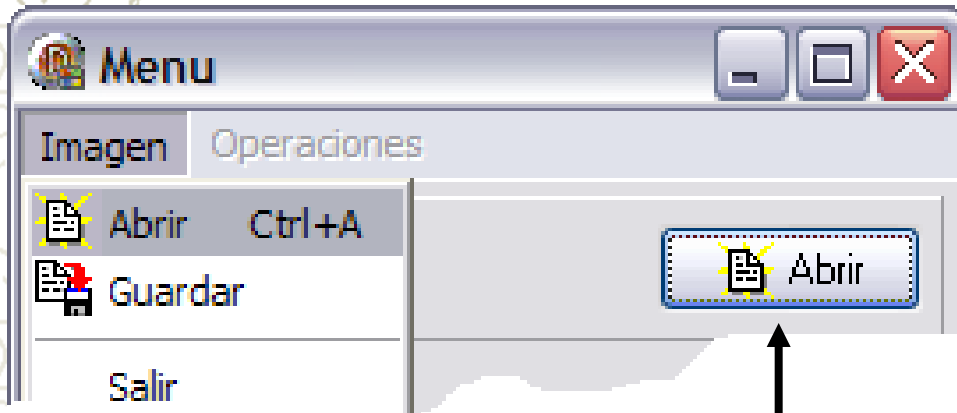
```
procedure TForm1.Guardar1Click(Sender: TObject);  
begin  
    if SaveDialog1.execute then  
        Image1.Picture.SaveToFile(SaveDialog1.FileName);  
end;
```

# Ejemplo Menu.dpr

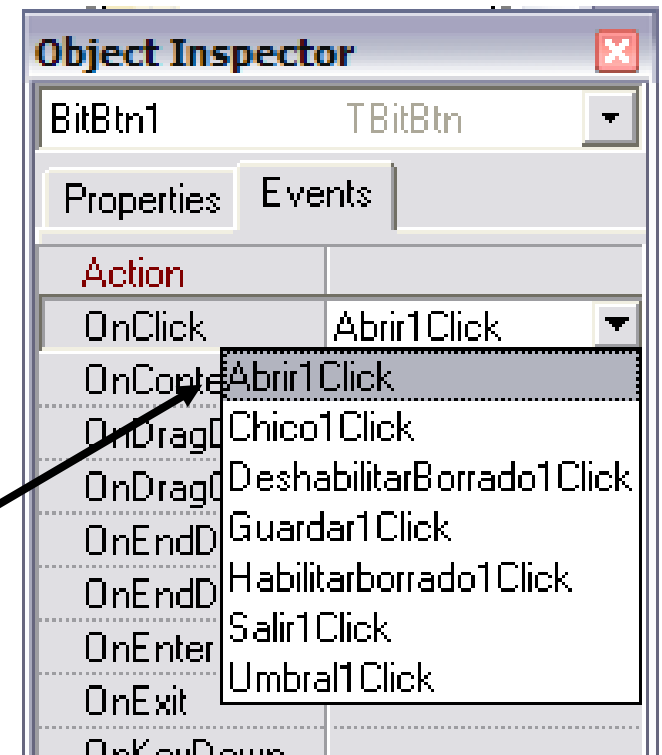


✱ Agregar un botón que realizará la misma operación que la opción "Abrir" del menú

# Compartiendo eventos

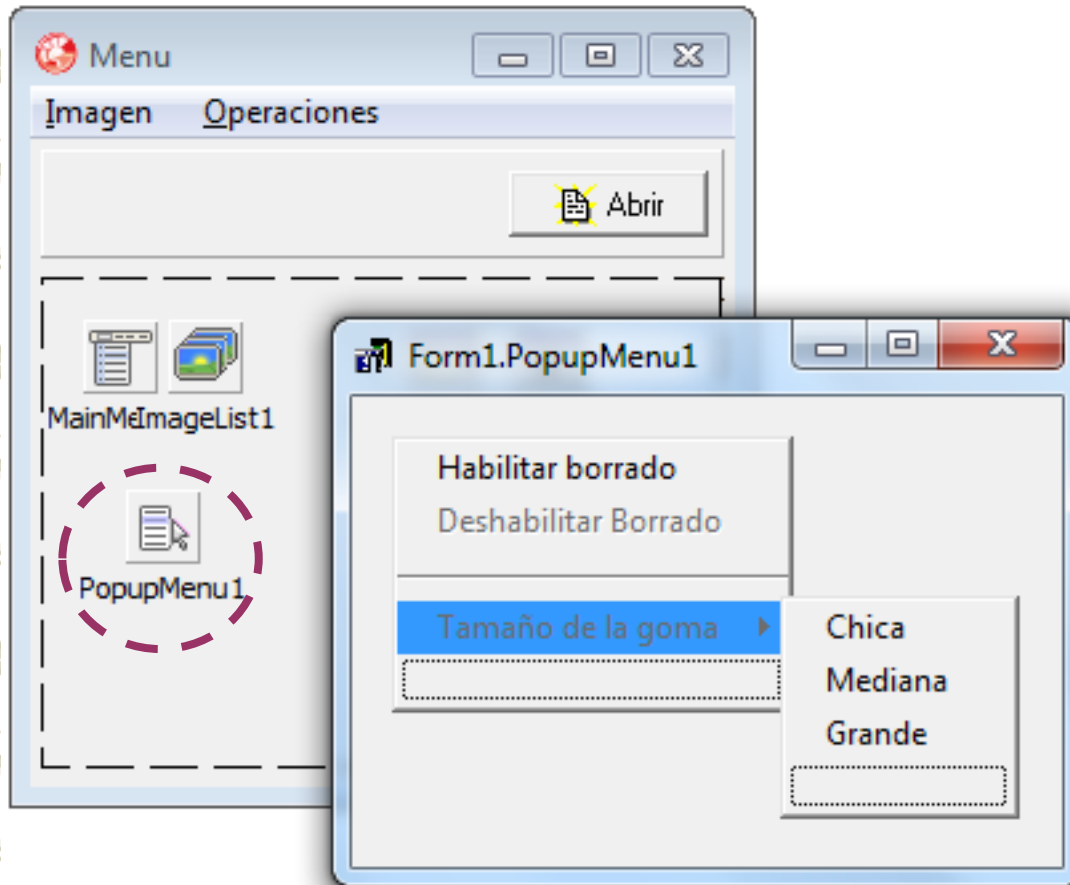


En lugar de escribir un nuevo proceso para el OnClick del botón, debemos seleccionar el proceso ya hecho para la opción del menú



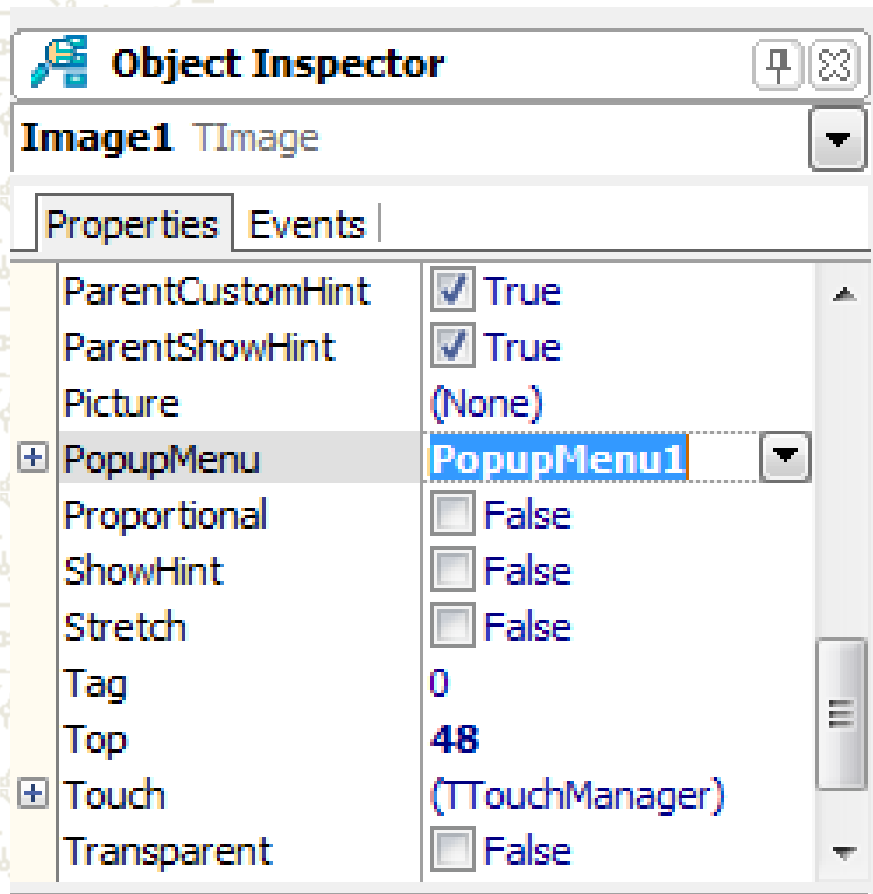


# PopupMenu



- ✿ Insertar un **PopupMenu** que se activará al presionar el botón derecho del mouse sobre la imagen.
- ✿ Diseñelo con las opciones del dibujo.

# Asociando el PopupMenu a la imagen



✦ Modificar la propiedad PopupMenu de la imagen indicándole qué menú debe utilizar

Ver **Menu.dpr**