

# Lenguajes Visuales

---

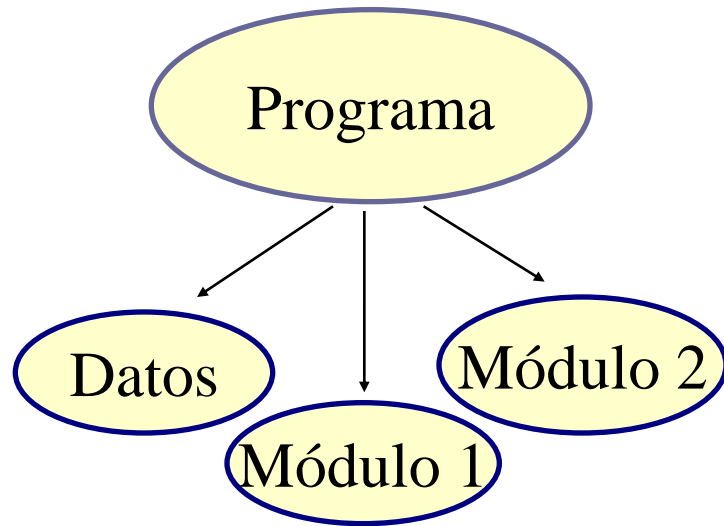
- Uso de componentes
  - Su apariencia y comportamiento pueden ser adaptados fácilmente.
  - Ej: botones y menús desplegables.
  
- Esto facilita el desarrollo de aplicaciones porque el programador puede dedicarse a resolver el problema sin preocuparse por la parte visual.

# Paradigmas de Programación

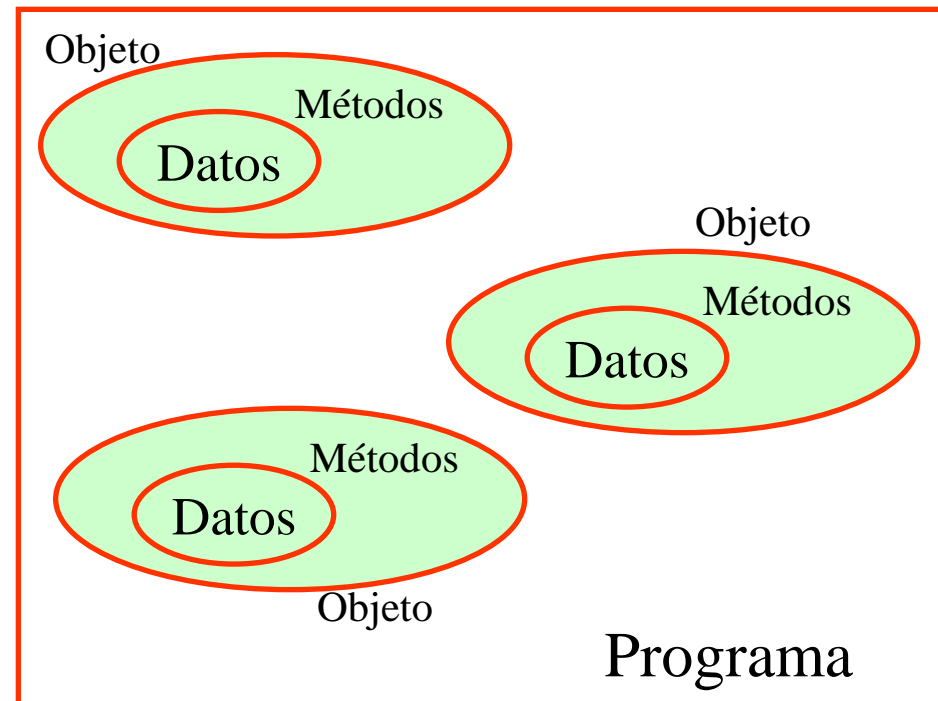
<b>Programación Imperativa</b>	<b>Programación Orientada a Objetos</b>
Divide el sistema en <b>módulos</b> (procedimientos y/o funciones) para operar sobre los datos.	Divide el sistema en componentes que contienen operaciones y datos. Cada componente se denomina <b>objeto</b> .
El flujo de control está manejado por las estructuras de control.	Los objetos de un sistema se comunican entre sí mediante mensajes.

# Un programa según ...

## Programación Estructurada



## Programación Orientada a Objetos



# Objeto y Componente

---

- **Objeto:** Representación informática de lo que se conoce como objeto en el sentido habitual, es decir, algo con características propias que funciona como un todo.
  - Ej. de mundo real : un reloj, un auto, un termostato, etc.
  - Ej. informático : botón, ventana, menú desplegable, etc.
- **Componente**
  - Objeto particular que puede ser reusado en diferentes contextos.
- **Usaremos “objeto” y “componente” como sinónimos.**

# Objeto

---

- Todo objeto tiene dos partes
  - Características o Propiedades con información descriptiva del objeto
  - Comportamiento

**Auto**

**Características**

marca, modelo, color, velocidad máxima, velocidad actual, tamaño tanque de combustible, cantidad actual de combustible en el tanque.

**Comportamiento**

frenar, acelerar, llenar tanque de combustible, etc

# Objeto

---

- Todo objeto tiene dos partes
  - Características o Propiedades con información descriptiva del objeto
  - Comportamiento

## Botón

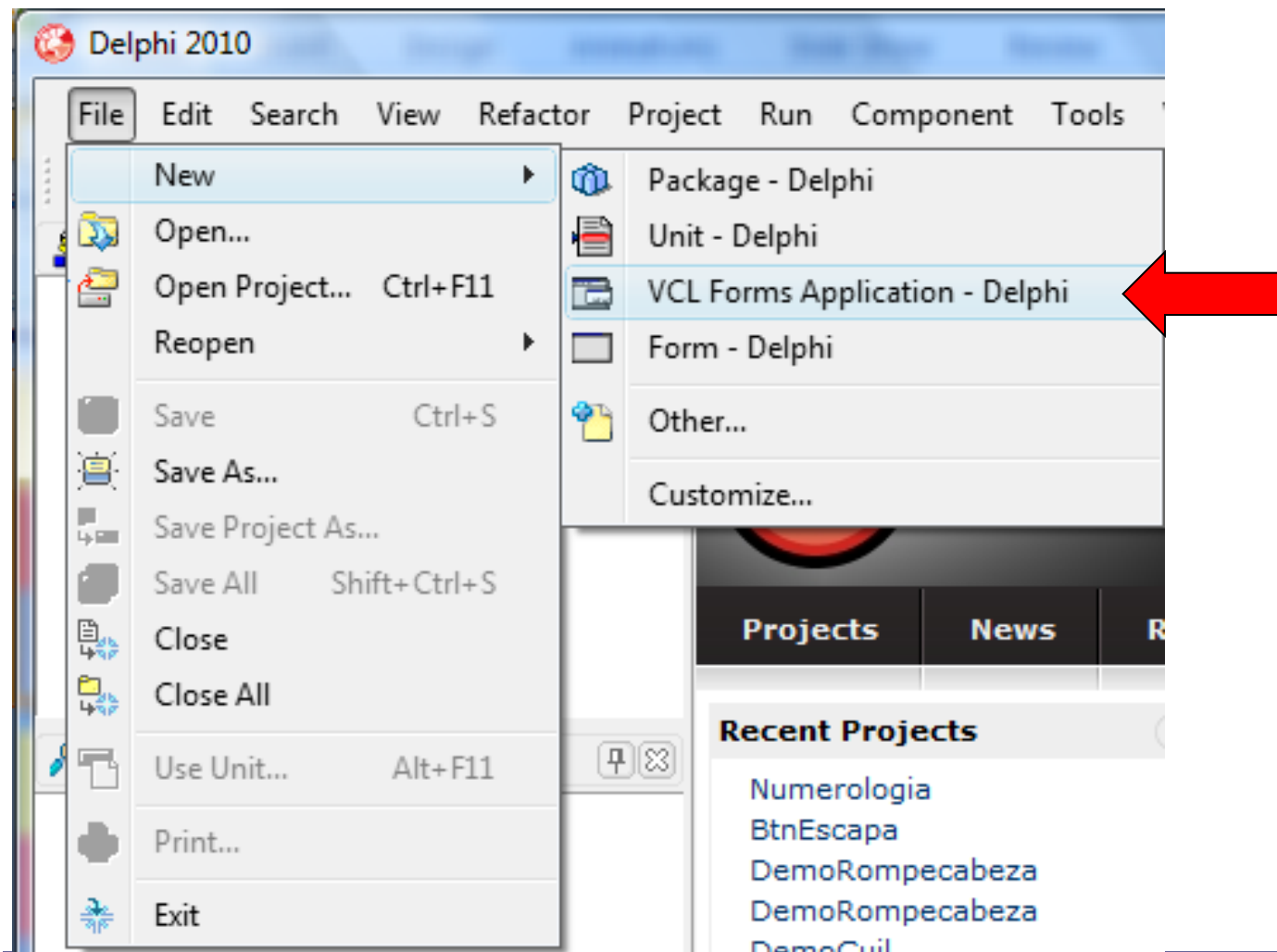
### **Características**

tamaño (ancho y alto), texto que se muestra dentro del botón, color, etc.

### **Comportamiento**

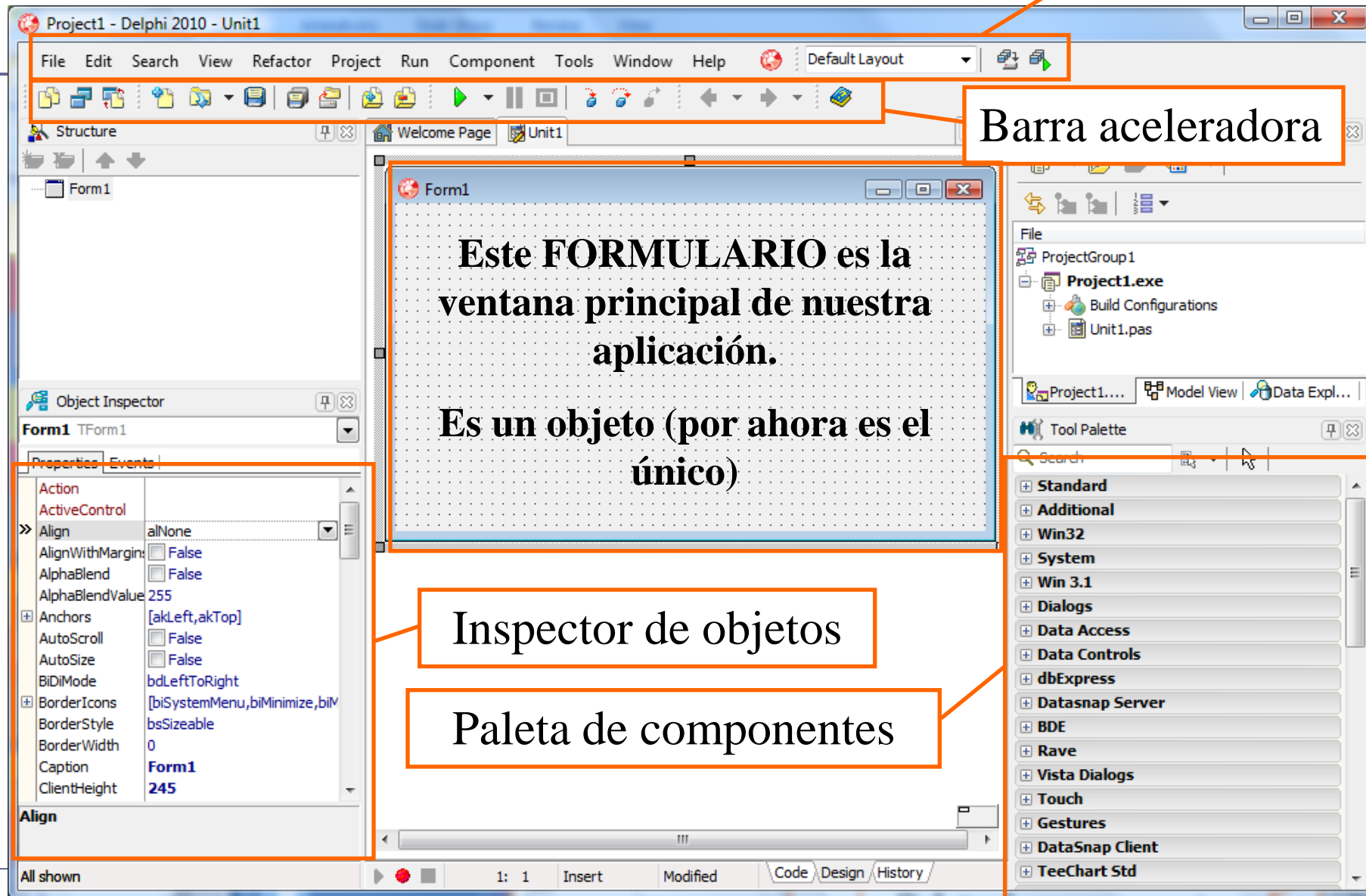
Lo que debe hacer cuando se clickea sobre el botón, idem para el doble click, etc.

# Cómo empezar ...



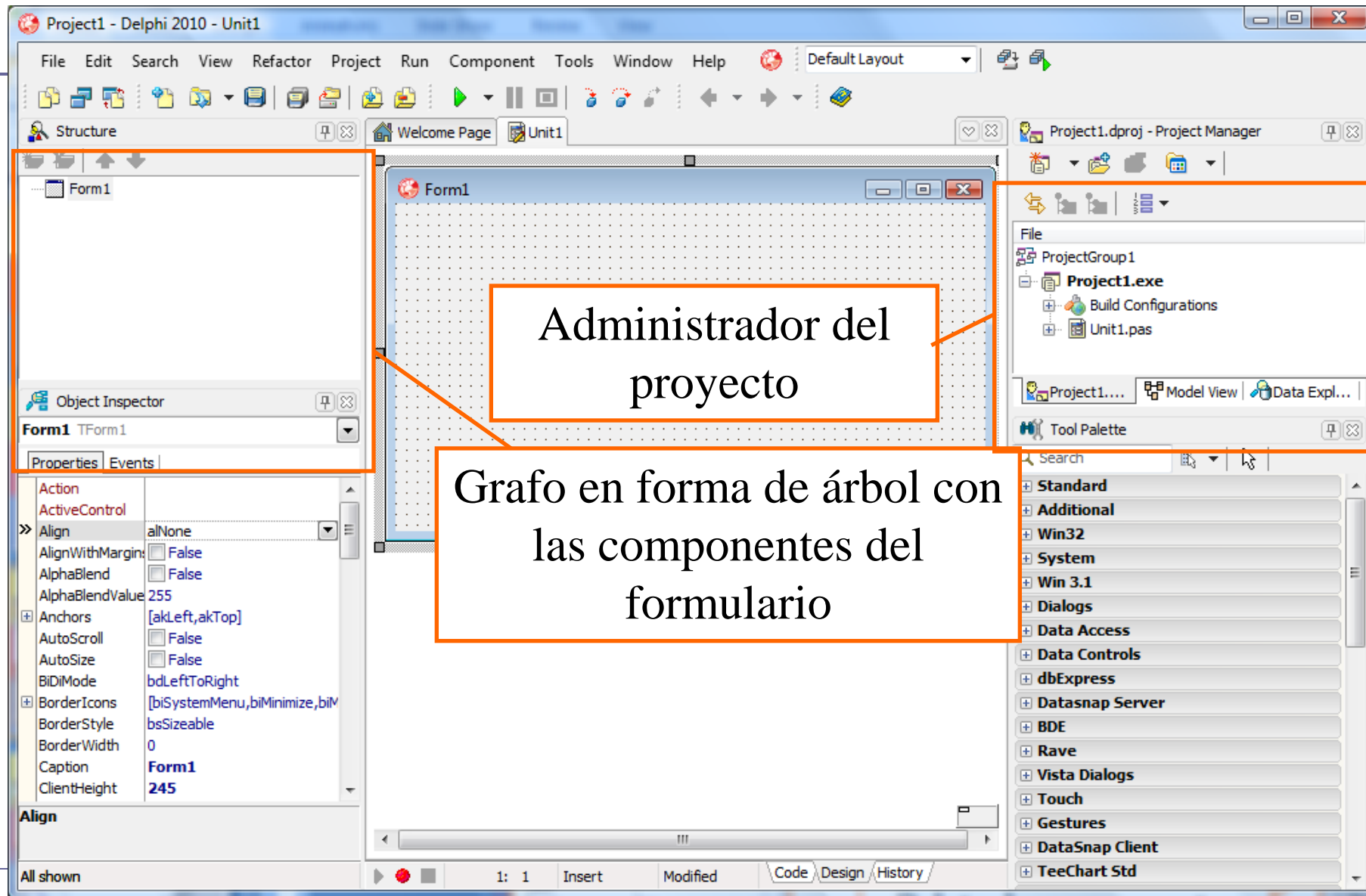
# Pantalla inicial de Delphi

Menú principal





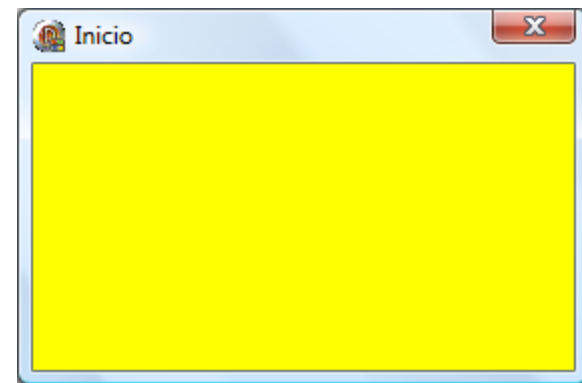
# Pantalla inicial de Delphi



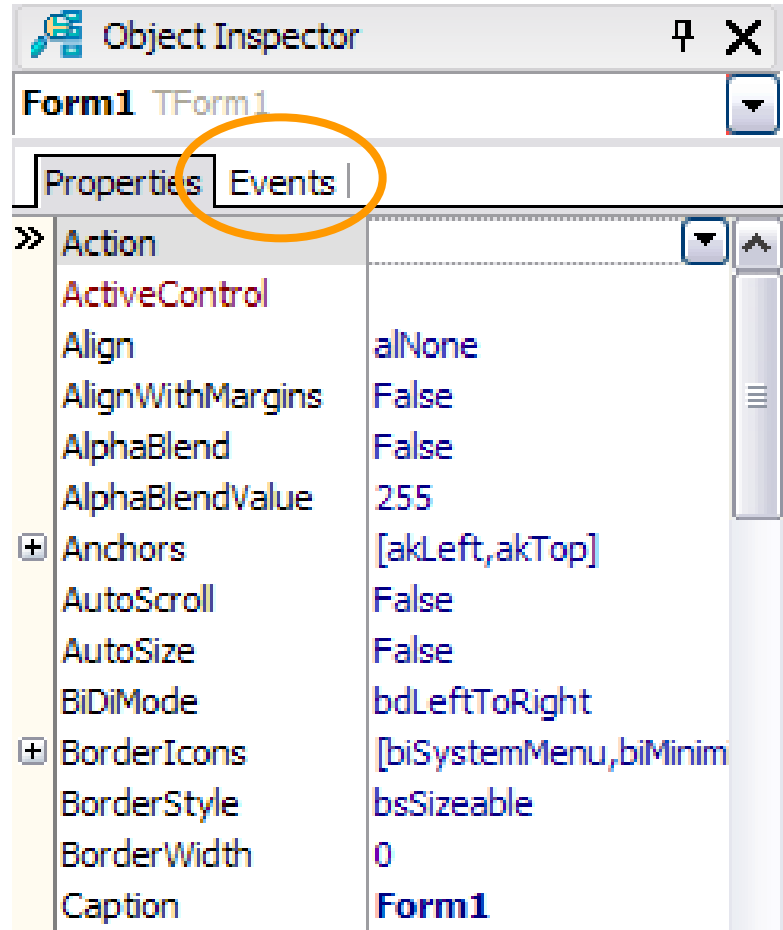
# Ejemplo 1.1. Inicio.dpr

---

- ❑ Utilizar el Inspector de Objetos para modificar las propiedades del *formulario*.
- ❑ Cambiar las siguiente propiedades desde el Inspectos de Objetos
  - Caption (título)
  - Color
  - Height
  - Width

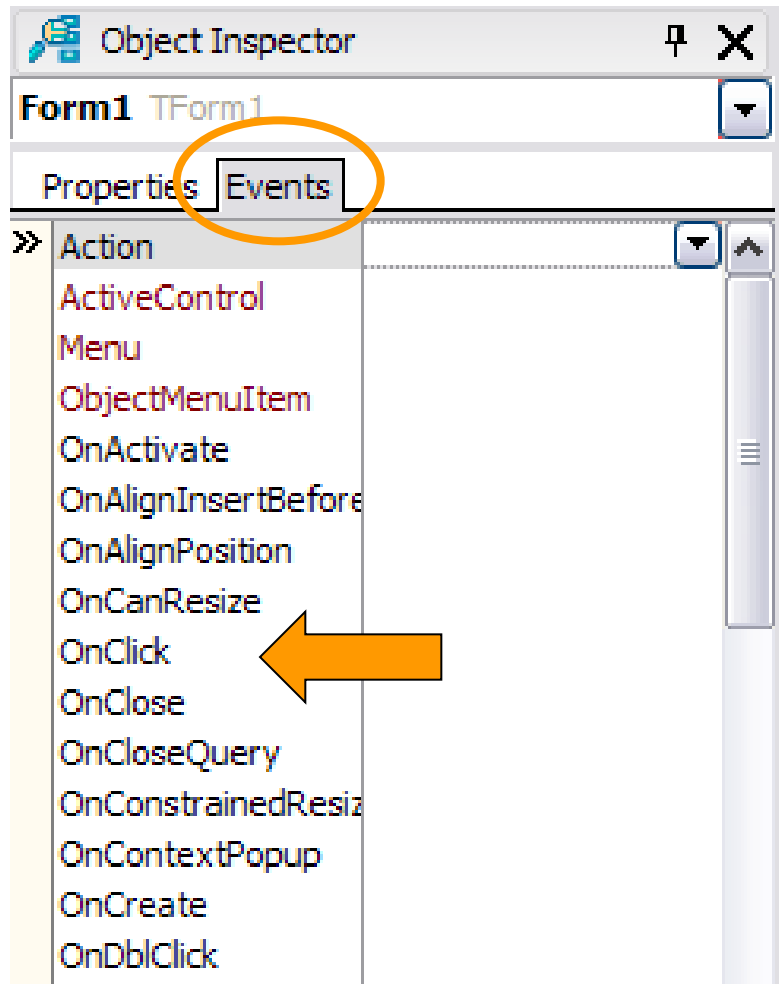


# Eventos



- Los objetos poseen la capacidad de responder a eventos

# Eventos



- Un **evento** es un hecho que se produce en un momento dado bajo ciertas condiciones y puede desencadenar reacciones
- Ejemplo
  - Click con el mouse sobre el formulario

# Objeto

---

- Todo objeto tiene dos partes
  - Características o Propiedades con información descriptiva del objeto
  - Comportamiento

# Objeto

---

- Todo objeto tiene dos partes
  - Características o Propiedades con información descriptiva del objeto
  - **Comportamiento**



Los distintos comportamientos de un objeto se indican a través de sus métodos

Un **método** es una descripción, mediante una secuencia de instrucciones, de lo que debe hacerse para obtener el resultado esperado.

# Objeto

---

- Todo objeto tiene dos partes
  - Características o Propiedades con información descriptiva del objeto
  - **Comportamiento**



Los distintos comportamientos de un objeto se indican a través de sus métodos

Por ejemplo, el Formulario tiene definido el método **Close** que permite cerrarlo.

**Form1.Close**

# Objeto

---

- Todo objeto tiene dos partes
  - Características o **Propiedades** con información descriptiva del objeto
  - **Comportamiento** definido a través de métodos.
- Delphi permite que las componentes (objetos) reaccionen a **eventos**. Es decir, que podemos invocar a métodos predefinidos o indicar las acciones a realizar ante determinadas situaciones.
  - Por ejemplo, qué debe hacerse al clickear un cierto botón

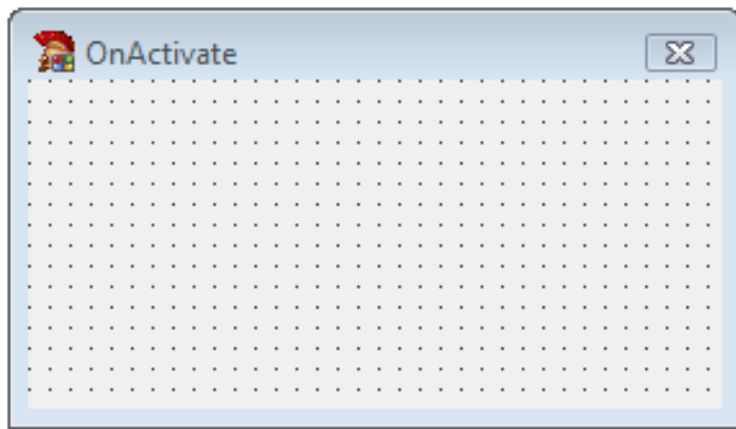
Definamos un evento sencillo



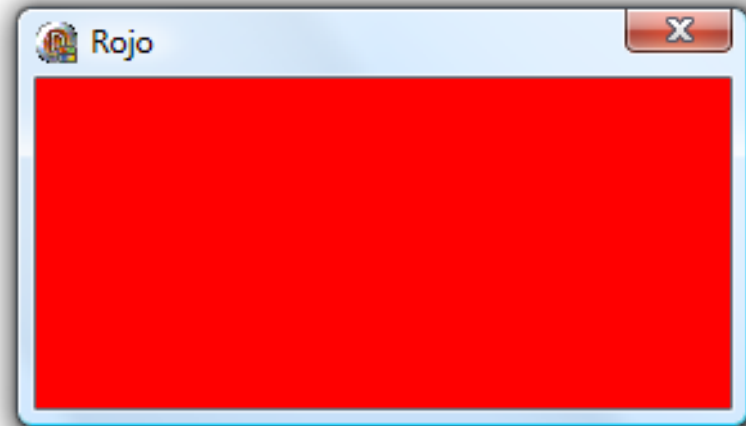
## Ejemplo 1.2. OnActivate.dpr

---

- Indique el color del formulario utilizando el evento **OnActivate**



(en edición)



(en ejecución)

**Ejemplo :** El color inicial del formulario puede indicarse por código en el evento **OnActivate**

---

```
procedure TForm1.FormActivate(Sender: TObject);  
begin
```

```
    Form1.caption := 'Rojo';
```

```
    Form1.color := clred;
```

```
end;
```

Nombre de la  
propiedad

Nombre del objeto  
(valor de su propiedad name)

## Ejemplo 1.3. CambioColor.dpr

---

- Escribir un programa que presente una ventana de color rojo con el título *"ROJO"*. Al clickear sobre el formulario debe cambiar de rojo a azul y viceversa. No se olvide de actualizar el título.

Nota: Utilice el evento **OnClick** del Formulario.

# Ejemplo 1.3. CambioColor.dpr

---

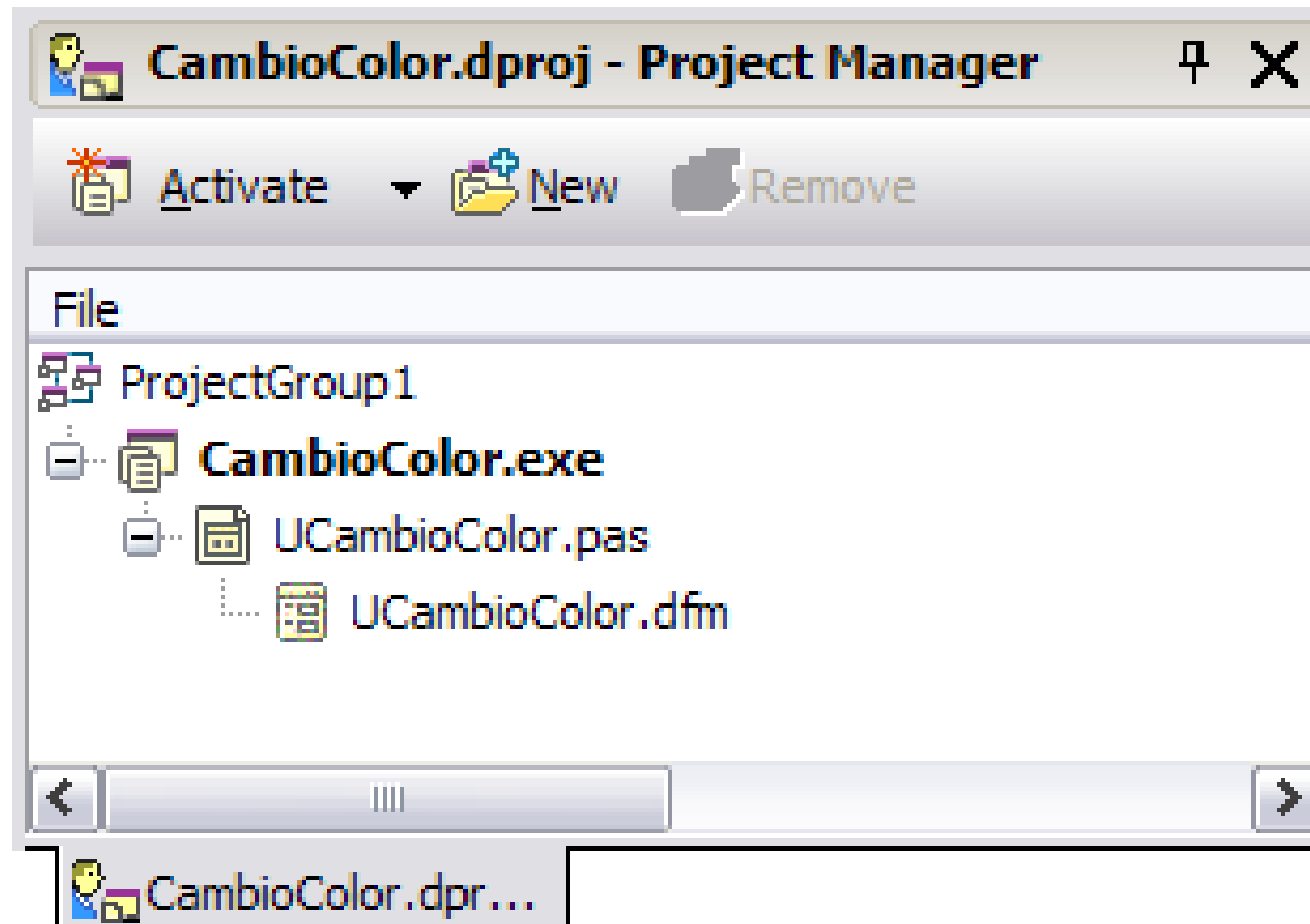
```
procedure TForm1.FormClick(Sender: TObject);
begin
    if Form1.color = clred then
        begin
            Form1.color := clblue;
            Form1.caption := 'AZUL';
        end
    else begin
        Form1.color := clred;
        Form1.caption := 'ROJO';
    end;
end;
```

# Esquema de un programa en Delphi

---

- ❑ La aplicación se encuentra dividida en una o varias unidades (extensión PAS). Todas las unidades están reunidas dentro de un mismo Project (extensiones DPR y DPROJ).
- ❑ Cada formulario tiene una unidad asociada. Allí se definen sus propiedades y eventos y los de las componentes que en él se utilizan.
- ❑ Inicialmente el proyecto tiene una única unidad correspondiente al único formulario existente.

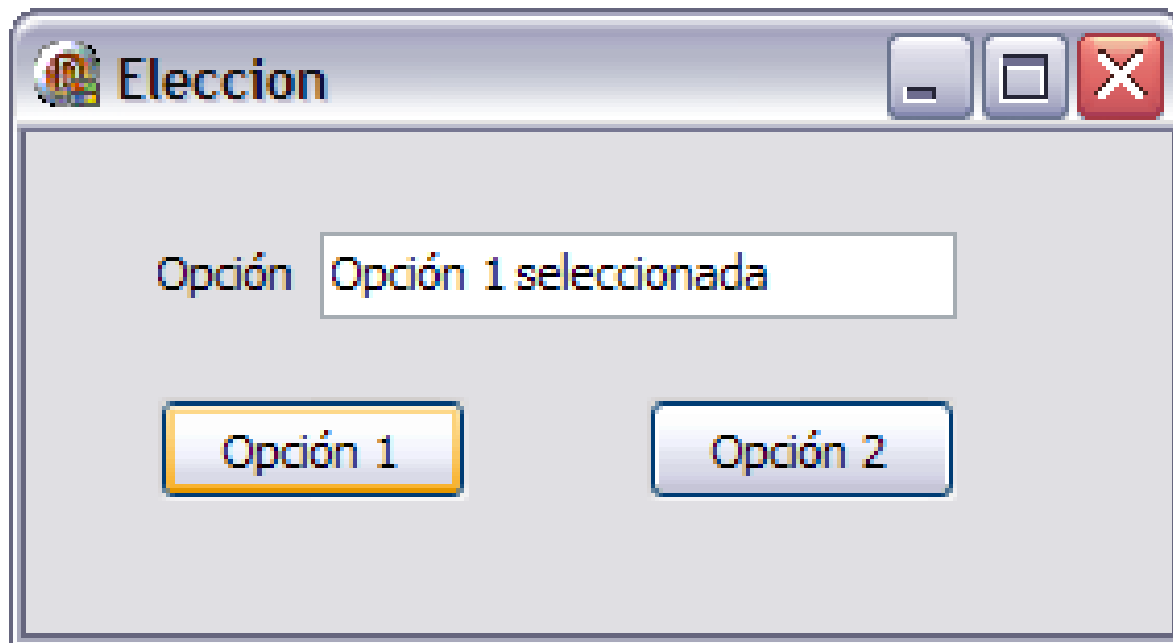
# Esquema de un programa en Delphi



# Ejemplo 1.4. Eleccion.dpr

---

- Diseñar la siguiente pantalla



## Ejemplo 1.4. Eleccion.dpr

---

- Si se clickea sobre el botón de la izquierda, se modifica la propiedad **Text** de la componente **Edit** colocando el cartel: "Opción 1 seleccionada" y si se selecciona el de la derecha aparecerá: "Opción 2 seleccionada".
- Si se clickea sobre el formulario, debe mostrarse el mensaje "Seleccione uno de los dos botones" con **ShowMessage** o con **MessageDlg**.



# Ejemplo 1.4. Eleccion.dpr

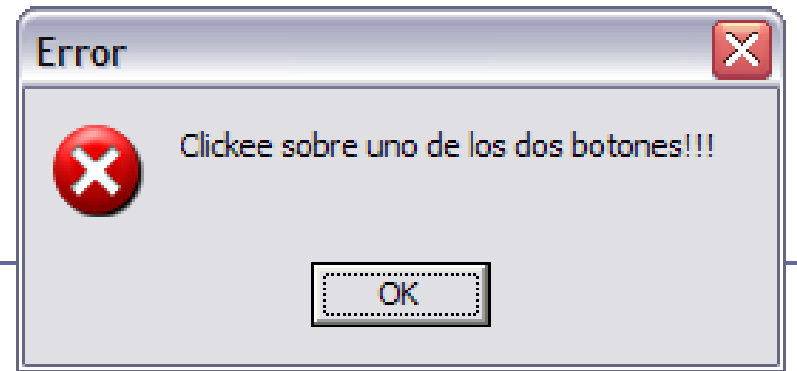
---

```
procedure TForm1.Opcion1Click(Sender: TObject);  
begin  
    Edit1.text := 'Opción 1 seleccionada';  
end;
```

```
procedure TForm1.Opcion2Click(Sender: TObject);  
begin  
    Edit1.text := 'Opción 2 seleccionada';  
end;
```

# Eleccion.dpr

Mensaje a visualizar en la ventana



```
procedure TForm1.FormClick(Sender: TObject);  
begin
```

```
    MessageDlg('Clickee sobre uno de los dos botones!',  
                mterror, [mbok], 0);
```

```
end;
```

Tipo de ventana :  
mtError, mtWarning,  
mtInformation, etc

Referencia a  
una página de  
ayuda

Conjunto de botones  
Ej: [mbYes, mbCancel]

# Aspectos generales

---

## □ Propiedades más comunes

- Name
- Caption o Text
- Enabled
- Visible
- Width
- Height
- Top
- Left
- Color
- Font

# Aspectos generales

---

## □ Eventos más comunes

- OnClick
- OnMouseDown
- OnMouseUp
- OnMouseMove
- OnEnter
- OnExit

# Resumen

---

## □ Componentes

### ■ **Propiedades**

- Características; por ejemplo el color.

### ■ **Métodos**

- Comportamiento definido previamente según el objetivo de la componente.

### ■ **Eventos**

- Mecanismo para invocar métodos y/o definir la acción esperada.

- **Modificamos las propiedades** de las componentes (desde el Inspector de Objetos y desde el código)

- **Escribimos eventos** para que la componente reaccione según lo pedido.

- **Esquema de un programa en Delphi**

- **MessageDlg**