



Objetivos:

- ❖ Diseño de diagramas de flujo con **funciones** y **procedimientos**
- ❖ Codificación de **funciones** y **procedimientos** en Visual Basic
- ❖ Llamada a subprogramas propias y del sistema.

Programa de demostración del uso de funciones y procedimientos

Interfaz:

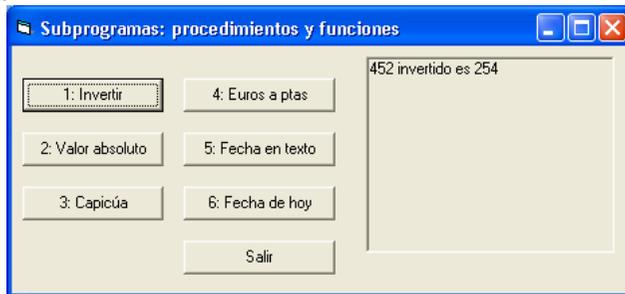


Figura 1. Objetos presentes en la interfaz: botones y caja de dibujo

Funcionamiento general:

1. Cada **ejercicio** tendrá un botón de ejecución propio (cmdEj1, cmdEj2, ..., cmdEj5).
2. Al hacer **click** sobre cada botón, borraremos inicialmente el contenido de la **caja de dibujo** (PictureBox) del resultado, **pctRes**. Utilizaremos para ello la función **Cls** (pctRes.Cls).
3. Al hacer **click** en el botón **Salir**, el programa **finalizará**.
4. Se proporciona un modelo de programa **ejecutable** para clarificar los enunciados.

Ejercicio 1: invertir número positivo no terminado en 0 (resolución)

Funcionamiento:

Llamaremos **cmdEj1** al botón asociado al ejercicio 1. Cuando el usuario pulse el botón etiquetado "**1: Invertir**", el programa pedirá un número positivo que no termine en 0 mediante una instrucción **InputBox** y mostrará en la caja de dibujo (**PictureBox**) mediante una sentencia **Print** (pctRes.Print) este mismo número invertido, tal y como se ejemplifica en la Figura 1.



Algoritmo:

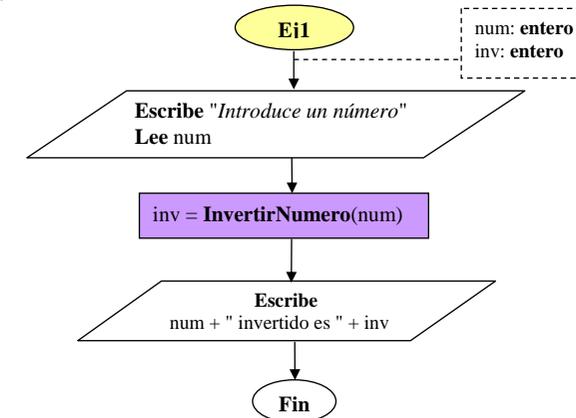


Figura 2. Diagrama de flujo del ejercicio 1

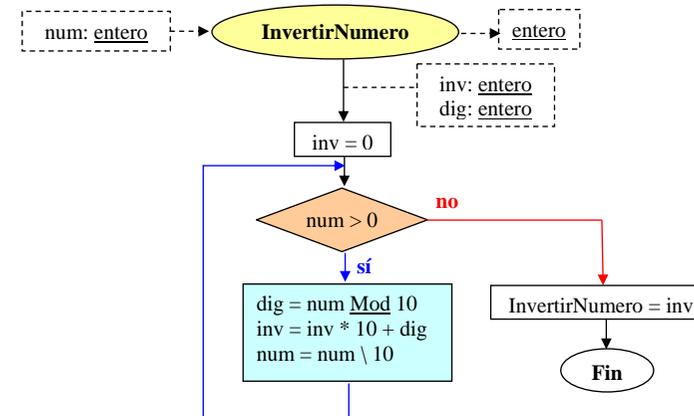


Figura 3. Diagrama de flujo de la función InvertirNumero del ejercicio 1

Pasos a seguir:

1. Crearemos los objetos del tipo y forma mostrados en la Figura 1. Guardaremos todo.



2. Añadir el código a los eventos, es decir, la carga del formulario y el clic sobre los botones:

- **Código del botón “1: Invertir”:** es el procedimiento o subprograma asociado al evento clic, como hemos venido haciendo. Nótese que llama a la función InvertirNumero.

```
Private Sub cmdEj1_Click()
    Dim num As Integer
    Dim inv As Integer

    pctRes.Cls

    num = InputBox("Introduce un número positivo que no termine en 0")
    inv = InvertirNumero(num)
    pctRes.Print num & " invertido es " & inv
End Sub
```

Habrá que definir igualmente (normalmente a continuación) la función nueva:

```
Private Function InvertirNumero(ByVal num As Integer) As Integer
    Dim dig As Integer
    Dim inv As Integer

    inv = 0
    While num > 0
        dig = num Mod 10
        inv = inv * 10 + dig
        num = num \ 10
    Wend
    InvertirNumero = inv
End Function
```

- **Código del botón Salir:**

```
Private Sub cmdSalir_Click()
    End
End Sub
```

Ejercicio 2: valor absoluto de un número

Diseña los **diagramas de flujo** y escribe un **procedimiento** que lea un número y calcule su **valor absoluto**, mostrando el resultado en el cuadro de dibujo. Utiliza para ello una **función** que calcule el valor absoluto de un número:



Figura 4. Cabecera de la función que calcula el valor absoluto



Ejercicio 3: números capicúas

Diseña los **diagramas de flujo** y escribe un **procedimiento** que lea un número y muestre en el cuadro de dibujo **si es capicúa** (se lee igual del derecho y del revés). Utiliza para ello la **función** EsCapicua que llame a la función InvertirNumero vista en el ejercicio 1.



Figura 5. Cabecera de la función que dice si un número es capicúa

Ejercicio 4: conversor de euros a pesetas

Diseña los **diagramas de flujo** y escribe un **procedimiento** que pida una cantidad en euros y nos muestre su valor en pesetas, sabiendo que 1 €son 166,386 pesetas.



Figura 6. Cabecera de la función que convierte euros a pesetas

Ejercicio 5: mostrar una fecha

Diseña los **diagramas de flujo** y escribe el **procedimiento** que pida un **día, mes y año** (mediante tres llamadas a la función InputBox) y muestre la **cadena de la fecha** con el siguiente formato: **día de mes de año**, por ejemplo “21 de Junio de 2006”. Para obtener la cadena del mes utilizará una función específica CadenaMes. Esta función no verifica si el día, mes y año se corresponden con una fecha correcta, por ejemplo: “0 de No-mes de -123”.

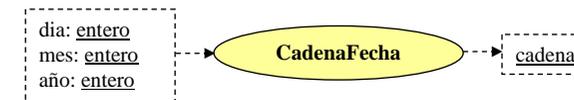


Figura 7. Cabecera de la función que obtiene la **cadena** de una **fecha**



Figura 8. Cabecera de la función que obtiene la **cadena** de un **mes**



Ejercicio 6: mostrar la fecha del sistema

Diseña los **diagramas de flujo** y escribe un **procedimiento** que muestre la fecha del sistema. Para ello hará uso de las funciones del sistema que se estimen necesarias de la tabla 1.

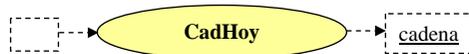


Figura 9. Cabecera de la función que obtiene la **cadena** de la fecha del sistema

Date () As Date	Fecha actual del sistema DD/MM/AAAA
Day (ByVal dat As Date) As Integer	Día de una fecha
Month (ByVal dat As Date) As Integer	Mes de una fecha
Year (ByVal dat As Date) As Integer	Año de una fecha

Tabla 1. Lista de funciones con fechas de Visual Basic