



Objetivos:

- ❖ Toma de contacto con el **entorno** de trabajo **Visual Basic** (VB)
- ❖ Diseño gráfico de un programa VB: **botón de comando** (cmd)
 - Propiedades: **(Nombre)** y **Caption**
 - Evento **Click**
- ❖ Diseño de **diagramas de flujo** y **codificación** de programas **secuenciales**
- ❖ Finalización de un programa VB: **End**
- ❖ Declaración de **constantes y variables**: **Const** y **Dim**
- ❖ Instrucciones de lectura y escritura: **InputBox** y **MsgBox**
- ❖ Tipos **String**, **Integer** y **Double** y conversiones **CInt** y **Cdbl**
- ❖ **Asignación** y **expresiones** sencillas
- ❖ **Concatenación** de cadenas y salto de línea: **&** y **vbCrLf**

Programa de saludo

Como ejercicio resuelto vamos a **diseñar** y **codificar** un programa Visual Basic (VB) que nos pregunte el nombre y nos envíe un saludo.

Interfaz

La Figura 1.1 nos muestra la interfaz propuesta para nuestro programa, que constará de dos controles de tipo botón (*command button*) etiquetados “Salúdame” y “Salir” para poder realizar las acciones respectivas.

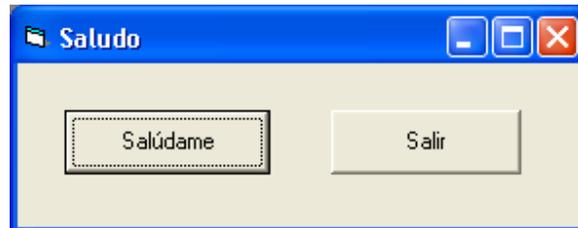


Figura 1.1. Interfaz del programa de saludo.

Diagramas de flujo

A efectos del programa vamos a identificar los botones como `cmdSaludo` y `cmdSalir`. Anteponeamos el acrónimo `cmd` (del inglés *command*) para facilitar la tarea de reconocerlos.

Cada uno de estos botones tendrá asociado un subprograma. Al ejecutar el programa, cuando hagamos clic sobre los botones se ejecutará el código asociado a ese evento (el evento “hacer clic”).

La Figura 1.2 muestra el diagrama de flujo asociado al botón que finaliza el programa. Es un diagrama de flujo sencillo que utilizaremos en todos los programas VB en situaciones parecidas. La acción “**FinPrograma**” es una orden VB que veremos más adelante en el apartado de codificación.

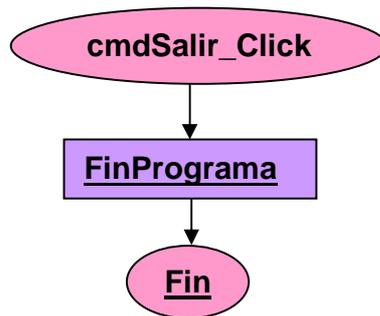


Figura 1.2. Diagrama de flujo asociado al botón cmdSalir.

El diagrama de flujo asociado al botón de saludo se muestra en la Figura 1.3. En él declaramos dos variables tipo **cadena**: nombre y saludo. Leemos el nombre y calculamos el saludo, anteponiendo la cadena constante "Hola " al nombre. Nótese que utilizamos el símbolo "+" para expresar la **concatenación** de cadenas y, aunque sea válido, en VB utilizaremos un símbolo distinto, como veremos más adelante.

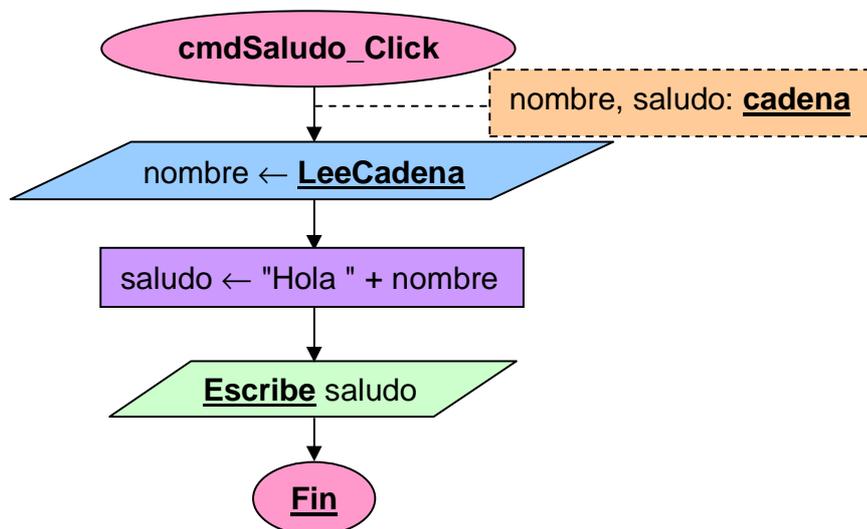


Figura 1.3. Diagrama de flujo asociado al botón cmdSaludo.

Codificación

Un subprograma comienza por la instrucción **Sub** seguida del nombre de subprograma, cmdSalir_Click(); finaliza con la orden **End Sub**.

El código del evento **Click** asociado al botón **cmdSalir** es sencillo, como se muestra en la Figura 1.4. Bastará con llamar a la instrucción **End** de VB, que se corresponde con la orden "**FinPrograma**" que habíamos especificado en el diagrama de flujo de la Figura 1.2.

```

Sub cmdSalir_Click()
End
End Sub
  
```

Figura 1.4. Código del botón cmdSalir.

El código del evento **Click** asociado al botón **cmdSaludo** de la Figura 1.5 se corresponde con el diagrama de flujo de la Figura 1.3. Podremos identificar los distintos elementos.

```

Sub cmdSaludo_Click()
    Dim nombre As String, saludo As String
    nombre = InputBox ("Introduce tu nombre")
    saludo = "Hola " & nombre 'Construimos el saludo
    MsgBox saludo
End Sub

```

Figura 1.5. Código del botón **cmdSaludo**.

En primer lugar declaramos las variables **nombre** y **saludo** como cadenas (**String**).

A continuación pedimos que el usuario introduzca su nombre, mediante la instrucción **InputBox**, recogiendo el resultado en la variable **nombre**. El cuadro de diálogo resultante se muestra en la Figura 1.6.

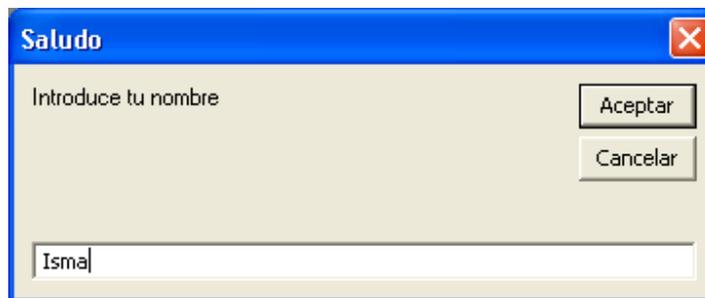


Figura 1.6. Cuadro de diálogo correspondiente al **InputBox**.

Se calculará el texto a mostrar asignando a la variable **saludo** la concatenación de la cadena literal "Hola " y el texto introducido en **nombre**. El operador de concatenación en VB es el signo denominado **and**, **et**, **ampersand** o signo "y comercial": "&".

Acto seguido mostraremos un cuadro de diálogo, **MsgBox**, con el saludo como se muestra en la Figura 1.7.



Figura 1.7. Cuadro de diálogo correspondiente al **MsgBox**.

Introduciendo el programa en el entorno VB

Para crear el primer programa sigue los siguientes pasos:

1. Lanza el programa: Inicio → Todos los Programas → Microsoft Visual Basic 6.0 → Microsoft Visual Basic 6.0

2. Si obtienes un cuadro de diálogo como el de la Figura 1.8 selecciona “Exe estándar” → Abrir.

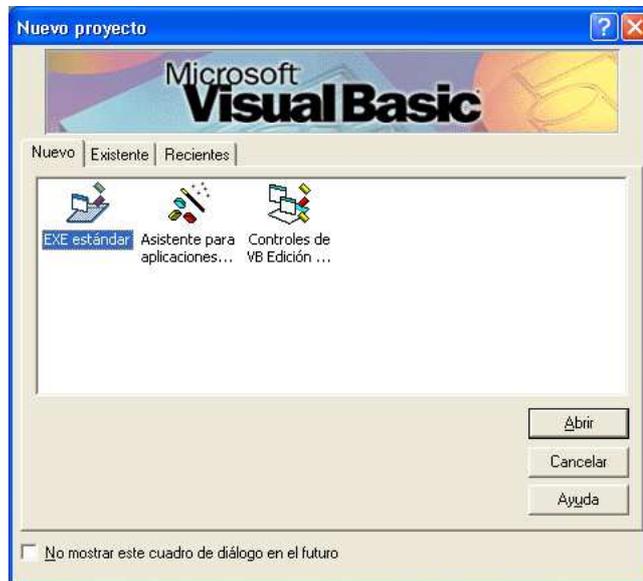


Figura 1.8. Proyecto VB nuevo.

3. Menú → Ver → Cuadro de Herramientas (en la parte de abajo, si no está seleccionado). El entorno resultante se muestra en la Figura 1.9.

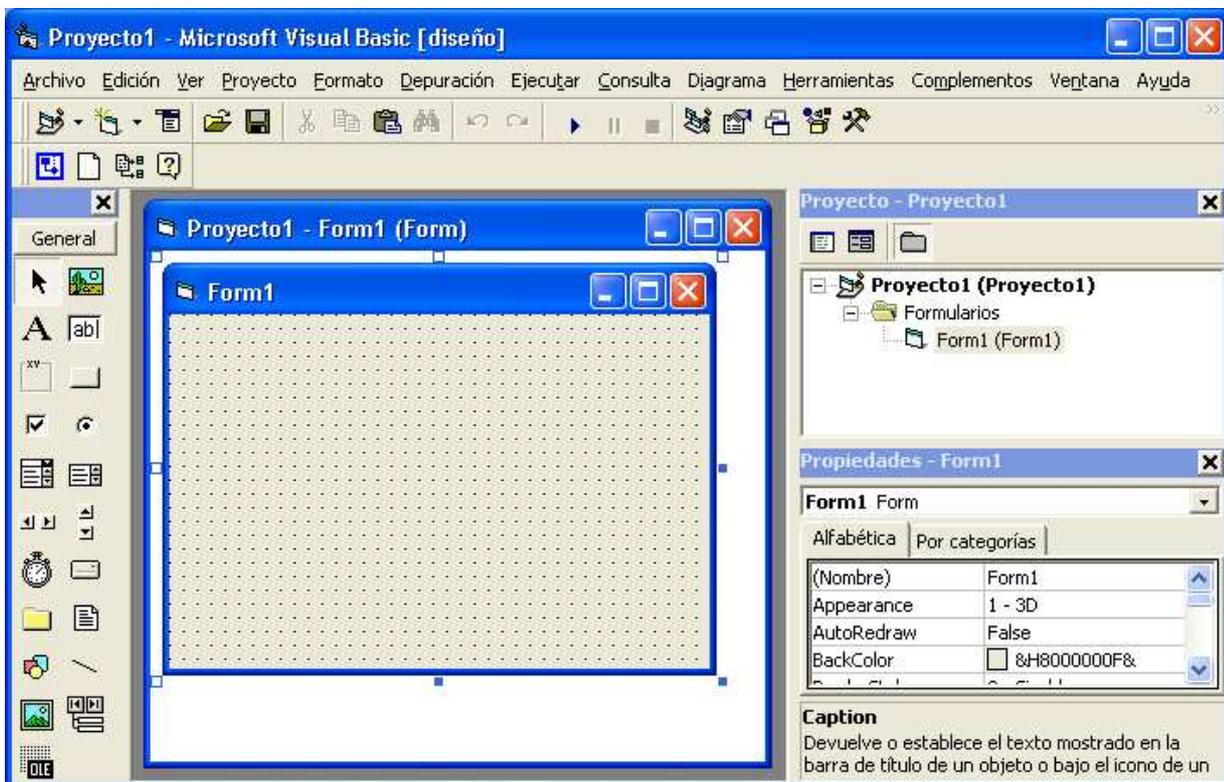


Figura 1.9. Entorno VB.

4. Elige (haz clic) del Cuadro de Herramientas el botón  (Figura 1.10) y dibuja un botón en el formulario, haciendo clic y arrastrando.

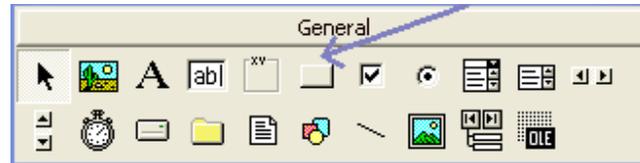


Figura 1.10. Botón en el Cuadro de Herramientas.

5. En el panel de propiedades aparecerán las propiedades del botón. Vamos a modificar las siguientes:
- (Nombre):** nombre del botón en el código, `cmdSaludo`
 - Caption:** texto a mostrar en el botón, **Salúdame**
6. Haremos otro tanto con el botón **Salir**:
- (Nombre):** `cmdSalir`
 - Caption:** **Salir**
7. Si hacemos doble clic en el botón `cmdSaludo` nos aparecerá una ventana de edición de código con el esqueleto del subprograma correspondiente, como se muestra en la Figura 1.11. En ella introduciremos el código de la Figura 1.5. Nótese que en el código propuesto en la línea anterior pone "Option Explicit" y además antepone "Private" a la palabra clave "Sub". Los mantendremos sin entrar a discutir su utilidad.

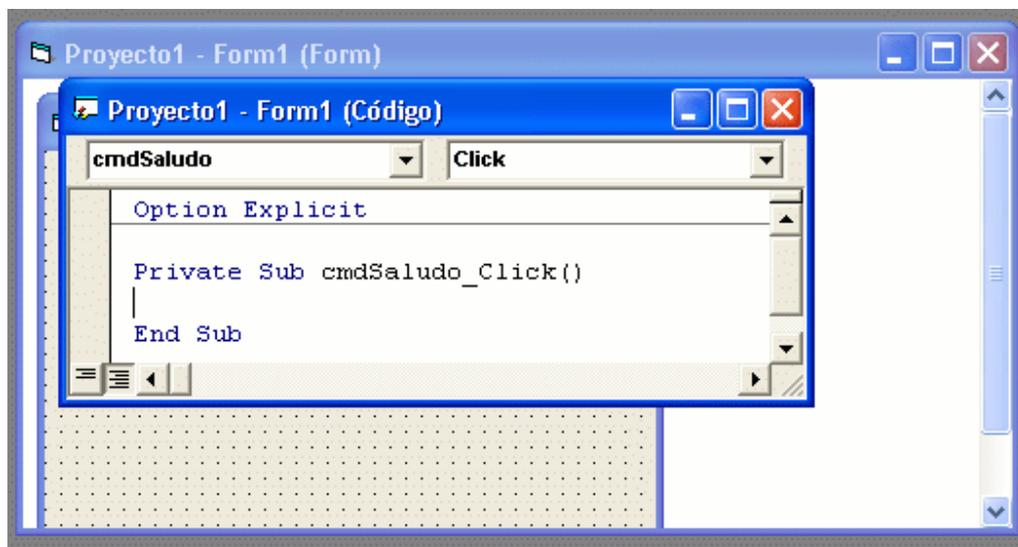


Figura 1.11. Edición de código.

8. Para ejecutar: tecla F5, hacer clic sobre  o elegir en el menú (Ejecutar → Iniciar)
9. Para guardar cada proyecto se recomienda utilizar un lápiz de memoria con interfaz USB:
Menú → Archivo → Guardar Proyecto

- i. Nombre del **formulario**: Saludo (.frm)
 - ii. Nombre del **proyecto**: Saludo (.vbp) – se visualiza en los **InputBox** y **MsgBox**
 - iii. **Puedes borrar** los archivos innecesarios (.vbw)
10. Las posteriores veces que modifiques algo sólo tienes que hacer: Archivo → Guardar.

Constantes

Para declarar una constante, por ejemplo, la relación entre el euro y las antiguas pesetas, pondremos junto a las declaraciones de variables (si las hubiera) el nombre deseado y el valor asignado mediante una igualdad, como se muestra en la figura 1.13.

----- Pts1eur = 166,386

Figura 1.12. Declaración de una constante.

El código a escribir en Visual Basic se ilustra en la Figura 1.13. Nótese que en el diagrama de flujo utilizamos las convenciones del castellano (coma para separar los decimales) y en el programa utilizamos las convenciones de Visual Basic.

Const Pts1eur As Double = 166.386

Figura 1.13. Código de inicialización de una constante.

Lectura de enteros y reales

La instrucción **InputBox** de Visual Basic devuelve una cadena. Para poder leer enteros o reales leeremos primero el texto sobre una cadena y luego los convertiremos a los tipos correspondientes con las órdenes **CInt** y **CDBl** respectivamente.

En la Figura 1.14 se muestra una porción ilustrativa de cómo leeríamos un entero *i* (en el diagrama de flujo **LeeEntero**) y un real *d* (en el diagrama de flujo **LeeReal**).

```

Dim s As String
Dim i As Integer
Dim d As Double
s = InputBox("Entero:")
i = CInt(s)
s = InputBox("Real:")
d = CDBl(s)
...
```

Figura 1.14. Código de lectura de enteros y reales.

Un aspecto a tener en cuenta es que para la conversión de cadenas a reales mediante **CDBl**, Visual Basic espera que se introduzcan los números mediante la convención local, es decir, si utilizamos las convenciones del castellano, espera que los decimales se separen mediante una coma y no mediante un punto (como espera que se utilicen en el programa). Además, para enteros (en Visual Basic de 2 bytes, 16 bits) se producirá un sobrepasamiento a partir del 32768, lo cual no se considerará un error de programa.

Trucos de diseño gráfico

Una buena práctica para que todos los elementos gráficos de un formulario (por ejemplo, los botones) sean de igual tamaño es decidir desde un principio la forma y el tamaño de uno de ellos, tomando como referencia los textos más largos.

Dado el primero de los elementos se podrá copiar al portapapeles mediante **Menú** → **Edición** → **Copiar** o pulsando **Control-C** y pegarlo para duplicarlo las veces que sea necesario mediante **Menú** → **Edición** → **Pegar** o pulsando **Control-V**.

Hay que tener **mucho cuidado** porque al cuadro de diálogo como el de la Figura 1.15 preguntando si se desea crear un botón con el mismo nombre pero índice distinto, al que se tendrá que elegir el botón **No**.

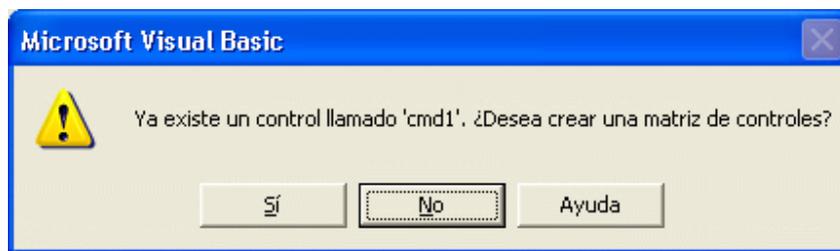


Figura 1.15. Cuadro de diálogo al pegar un botón.

Ejercicios propuestos

Para los siguientes ejercicios utiliza un único proyecto, codificando cada ejercicio en un botón distinto (cmd1, cmd2, ..., cmd9 y cmdSalir). En la Figura 1.16 se propone una interfaz.



Figura 1.16. Interfaz de los ejercicios propuestos.

1. **Diseña** el diagrama de flujo y **codifica** un programa que lea **dos números** enteros y calcule su **suma**. Pruebas: 12+15, 12000 + 20000, 40000+12.
2. **Diseña** el diagrama de flujo y **codifica** un programa que lea el radio (r) calcule el **área** (a) y el **volumen** (v) de una **esfera**. Declara π (pi) como una constante que valga 3,14159265358979. Muestra el resultado en un único cuadro de diálogo en tres líneas separadas (para radio, área y volumen) por un salto de línea ([vbCrLf](#)). Pruebas: r = 1, r = 2, r = 45, r = 4.5, r = 4,5.

$$a = 4\pi r^2 \quad v = 4\pi r^3/3$$

3. **Diseña** el diagrama de flujo y **codifica** un programa que lea un número entero y muestre los cinco números siguientes.

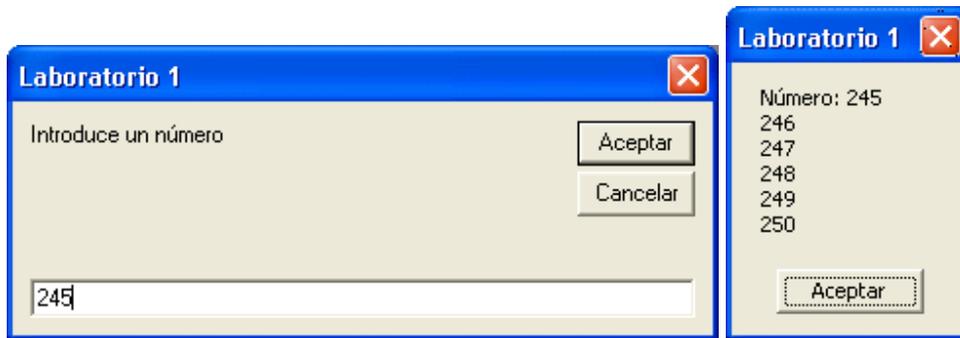


Figura 1.17. InputBox y MsgBox del ejercicio 3.

4. **Diseña** y **codifica** un programa para **convertir** la temperatura de grados **Celsius** a grados **Fahrenheit**.
- $$^{\circ}\text{C} = 5/9(^{\circ}\text{F}-32)$$
5. **Diseña** y **codifica** un programa para leer **dos números** enteros y calcular su **media entera**.
6. **Diseña** y **codifica** un programa para leer **tres números** enteros y calcular su **media entera**.
7. **Diseña** y **codifica** un programa para leer **dos números** enteros e **intercambiar sus valores**, mostrándolos a continuación. Nótese que no es suficiente con mostrar los valores en orden invertido sino que **hay que cambiar el contenido de las variables** en el programa.
- $$X = 23, Y = 45 \quad X = 45, Y = 23$$
8. **Diseña** y **codifica** un programa para leer **tres números** enteros e **intercambiar sus valores**, de manera similar al ejercicio anterior, mostrándolos a continuación.
- $$A = 23, B = 45, C = 67 \quad A = 45, B = 67, C = 23$$
9. **Diseña** y **codifica** un programa para leer una cantidad en **segundos** y nos muestre por pantalla las **horas, minutos y segundos** correspondientes. Ejemplo: Si introducimos 4005 nos mostrará “1 horas 6 minutos 45 segundos”. Si introducimos una cantidad alta de segundos se producirá un sobrepasamiento.