

LA FUNCIÓN INPUTBOX EN VBA

El Inputbox en VBA no es más que método que nos permite que aparezca un pop-up en el que introducir cierta información.

Como ya vimos en el caso del MsgBox, es muy frecuente en Excel usar pop-ups para interactuar con el usuario. En el caso del MsgBox lo que conseguiremos será enviar al usuario un simple aviso o mensaje sencillo. Con el Inputbox podremos pedirle información.

SINTAXIS

La sintaxis del Inputbox en VBA no es complicada, pero tiene ciertas partes que merece la pena destacar:

Application.InputBox(Prompt, Title, Default, Left, Top, HelpFile, HelpContextID, Type)

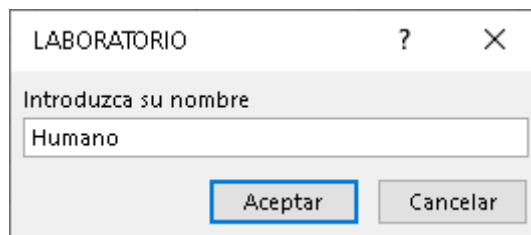
Donde las más comunes y útiles son las siguientes:

- Prompt: el texto que se mostrará en el pop-up. Es el único valor obligatorio
- Title: el título del pop-up que aparecerá en la parte superior.
- Default: el valor que mostrará por defecto el Inputbox.
- Type: el tipo de valor que podemos usar que puede ser de tipo numérico (1), texto (2), verdadero/falso (4), referencia a una celda (8), una matriz (64), etc.

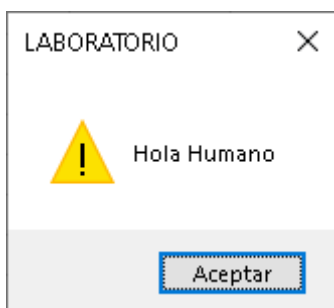
En el caso de que el valor se deje vacío o no sea del tipo que hemos especificado VBA nos mandará un error automáticamente, así que no hará falta comprobar el tipo de dato que nos devuelve el Inputbox en nuestro código VBA.

En este caso hemos usado un código muy sencillo que es el siguiente:

```
Private Sub CommandButton4_Click()  
    nombre = Application.InputBox("Introduzca su  
nombre", " LABORATORIO ")  
    MsgBox "Hola " + nombre, vbExclamation,  
"LABORATORIO"  
End Sub
```



Como vemos en el pedacito de código anterior, el resultado de la pregunta lo almacenaremos en una variable y esto será obligatorio si queremos mostrar luego la información almacenada:



LA FUNCIÓN MSGBOX EN VBA

La función MsgBox en VBA nos permite mostrar un mensaje dentro de un cuadro de diálogo en espera de que el usuario de Excel haga clic sobre alguno de los botones provistos. Si lo deseamos podemos tomar alguna acción específica después de conocer el botón pulsado por el usuario.

SINTAXIS

La función MsgBox en VBA tiene 5 argumentos, los cuales explicaré a continuación:

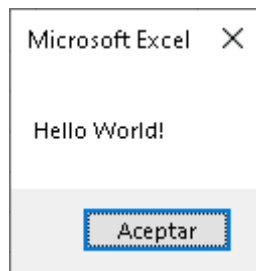
- Prompt (obligatorio): Es la cadena de texto que se mostrará como el mensaje dentro del cuadro de diálogo. La longitud máxima es de 1024 caracteres, pero depende del tipo de fuente utilizada.
- Buttons (opcional): Expresión numérica que proviene de la suma de ciertas constantes que representan el tipo de botón e iconos a desplegar.
- Title (opcional): Cadena de texto que se mostrará como el título del cuadro de diálogo. Si se omite, el título será el nombre de la aplicación.
- HelpFile (opcional): Cadena de texto con la ubicación del archivo de ayuda asociado al cuadro de diálogo. Si se especifica este argumento, debe indicarse también Context.
- Context (opcional): Valor numérico asignado por el autor al tema de ayuda. Si se especifica este argumento, debe indicarse también HelpFile.

1º EJEMPLO: LA FUNCIÓN MSGBOX EN VBA

Ya que solamente el primer argumento de la función MsgBox es obligatorio, podemos crear un mensaje informativo para el usuario con la siguiente línea de código:

```
Private Sub CommandButton1_Click()  
    MsgBox "Hello World!"  
End Sub
```

Al ejecutar este código se mostrará un cuadro de diálogo como el siguiente:

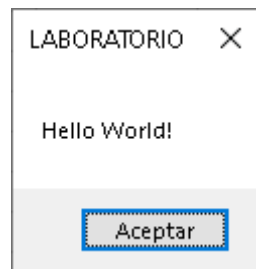


2º EJEMPLO: EL ARGUMENTO TITLE

Antes de revisar el segundo argumento de la función MsgBox, hablaremos sobre su tercer argumento que es el título del cuadro de diálogo. Para poner un título personalizado será suficiente indicarlo de la siguiente manera:

```
Private Sub CommandButton2_Click()  
    MsgBox " Hello World!", , "LABORATORIO"  
End Sub
```

Esta instrucción mostrará el título del cuadro de diálogo como “Mensaje especial” y ya no se mostrará el nombre de la aplicación:



EJEMPLO 3: EL ARGUMENTO BUTTONS

Hasta ahora solo has visto el botón Aceptar en el cuadro de diálogo, pero el segundo argumento de la función MsgBox nos permitirá indicar los botones que deseamos mostrar y también podremos elegir el icono desplegado y el comportamiento del cuadro de diálogo. La siguiente tabla indica los valores que podemos utilizar para este argumento:

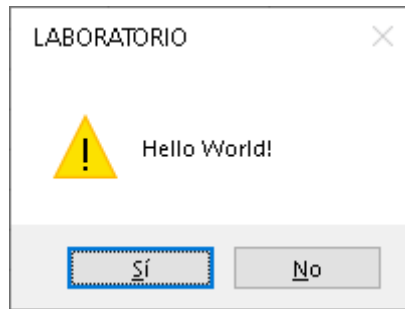
Constante	Valor	Descripción
vbOKOnly	0	Muestra el botón Aceptar
vbOKCancel	1	Muestra los botones Aceptar y Cancelar.
vbAbortRetryIgnore	2	Muestra los botones Anular, Reintentar y Omitir.
vbYesNoCancel	3	Muestra los botones Si, No y Cancelar.
vbYesNo	4	Muestra los botones Si y No.
vbRetryCancel	5	Muestra los botones Reintentar y Cancelar.
vbCritical	16	Muestra el icono Mensaje crítico.
vbQuestion	32	Muestra el icono Consulta de advertencia.
vbExclamation	48	Muestra el icono Mensaje de advertencia.
vbInformation	64	Muestra el icono Mensaje de información.
vbDefaultButton1	0	El primer botón es el predeterminado.
vbDefaultButton2	256	El segundo botón es el predeterminado.
vbDefaultButton3	512	El tercer botón es el predeterminado.
vbDefaultButton4	768	El cuarto botón es el predeterminado.
vbApplicationModal	0	Aplicación modal: el usuario debe responder al mensaje antes de continuar trabajando en la aplicación actual.
vbSystemModal	4096	Sistema modal: se suspenden todas las aplicaciones hasta que el usuario responda al cuadro de mensaje.
vbMsgBoxHelpButton	16384	Muestra el botón Ayuda.
VbMsgBoxSetForeground	65536	Especifica la ventana del cuadro de mensaje como ventana de primer plano.
vbMsgBoxRight	524288	Texto alineado a la derecha.
vbMsgBoxRtlReading	1048576	Especifica que el texto debe aparecer para ser leído de derecha a izquierda en los sistemas árabe y hebreo.

Todos estos valores los podemos dividir en cinco grupos. El primer grupo (0, 1, 2, 3, 4, 5) nos permite indicar los botones que se mostrarán en el cuadro de diálogo, el segundo grupo (16, 32, 48, 64) determinará el tipo de ícono mostrado, el tercer grupo (0, 256, 512, 768) es útil para indicar el botón predeterminado. El cuarto grupo (0, 4096) es la modalidad del cuadro de diálogo y el último grupo, que son los valores restantes, nos permiten indicar la alineación del texto y si deseamos mostrar un botón de Ayuda.

Ya que los valores de cada grupo son excluyentes, solo hace sentido elegir un valor de cada uno de ellos. De esta manera, si deseo mostrar los botones Si y No, y además mostrar un ícono de mensaje de advertencia, entonces debo utilizar la siguiente instrucción:

```
Private Sub CommandButton3_Click()
    MsgBox " Hello World!", vbYesNo + vbExclamation, "
    LABORATORIO "
End Sub
```

Para el segundo argumento puedes utilizar las constantes definidas para cada opción o utilizar directamente el valor numérico correspondiente de acuerdo con la tabla. El resultado de esta instrucción será el siguiente:



Para cada opción adicional que desees indicar deberás agregarla utilizando el símbolo de suma (+). Recuerda que este argumento de la función MsgBox es de valor numérico, así que Excel obtendrá la suma total para conocer las opciones indicadas.

EJEMPLO 4: CONOCER EL BOTÓN PULSADO

Para conocer el botón que ha sido pulsado por el usuario, es necesario guardar el valor devuelto por la función MsgBox en una variable:

```
resultado = MsgBox("Hola Mundo", vbAbortRetryIgnore, "Mensaje especial")
```

Antes de continuar debo mencionar algo importante sobre esta instrucción ya que, a diferencia de las anteriores, en esta ocasión los argumentos de la función MsgBox están encerrados en paréntesis. Esto se debe a una regla de programación en VBA la cual indica que cuando se llama a una función que devolverá un valor el cual será asignado a una variable, su lista de argumentos deberá estar rodeada por paréntesis. Así que no podemos olvidar colocar los paréntesis si queremos conocer el botón pulsado por el usuario.

Ahora bien, la variable **resultado** guardará el valor devuelto por la función MsgBox el cual puede ser cualquiera de las siguientes opciones dependiendo los botones que hayamos decidido mostrar:

Constante	Valor	Descripción
vbOK	1	Aceptar
vbCancel	2	Cancelar
vbAbort	3	Anular
vbRetry	4	Reintentar
vbIgnore	5	Omitir
vbYes	6	Si
vbNo	7	No

Una vez que el usuario hace clic sobre un botón, la función MsgBox nos devuelve el valor correspondiente y podremos comparar dicho valor con las contantes mencionadas en la tabla anterior. Para nuestro ejemplo, utilizaré una sentencia Select Case para comparar la variable resultado con todas las opciones posibles:

```
Private Sub CommandButton4_Click()  
    resultado = MsgBox("Hello World!", vbAbortRetryIgnore, "  
LABORATORIO ")  
    Select Case resultado  
        Case vbAbort:  
            MsgBox "Se pulsó el botón Anular"  
        Case vbRetry:  
            MsgBox "Se pulsó el botón Reintentar"  
        Case vbIgnore:  
            MsgBox "Se pulsó el botón Omitir"  
    End Select  
End Sub
```

Una vez identificado el botón sobre el cual el usuario ha hecho clic, mostraremos otro mensaje con el nombre de dicho botón. En un caso real, esa línea de código sería reemplazada por las instrucciones que se desean ejecutar de acuerdo con la respuesta del usuario.

En nuestro ejemplo mostramos un cuadro de diálogo con los botones Anular, Reintentar y Omitir, así que si el usuario hace clic sobre el botón Reintentar se mostrará el mensaje que nos confirmará dicha acción:

