CBS

Colegio Bautista Shalom

Ofimática
Quinto BACO
Segundo Bimestre

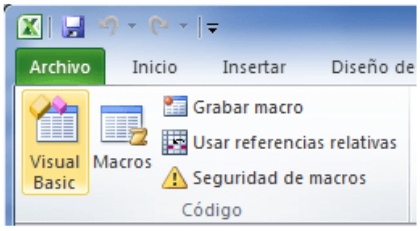
Contenidos

VISUAL BASIC EN RELACIÓN A EXCEL

- ✓ CÓMO ACTIVAR VISUAL BASIC EN EXCEL.
- ✓ CÓMO INSERTAR MACROS EN EXCEL CON VISUAL BASIC.
- ✓ CREAR MACROS EN EXCEL CON VISUAL BASIC.
- ✓ FICHA PROGRAMADOR.
- ✓ PROBLEMAS DE SEGURIDAD.
- ✓ EDITOR DE VISUAL BASIC.
- ✓ CÓDIGOS DE UNA MACRO DE EXCEL.
- ✓ CÓDIGOS MÁS COMUNES.
- ✓ USO DE LA GRABADORA DE MACROS.
- ✓ MODIFICACIÓN DEL CÓDIGO GRABADO.
- ✓ CUADRO DE CONTROL – CONTROLES ACTIVEX.
- ✓ CREANDO FORMULARIOS Y PROGRAMÁNDOLOS.
- ✓ TRABAJANDO CON FORMULAS.
- ✓ OBSERVANDO LOS CÓDIGOS DE UNA MACRO DE EXCEL.
- ✓ CONVERSIÓN DE TIPOS DE DATOS.

NOTA: conforme vayas avanzando en tu aprendizaje, encontrarás ejercicios a realizar. Sigue las instrucciones de tu catedrático(a).

VISUAL BASIC EN RELACIÓN A EXCEL

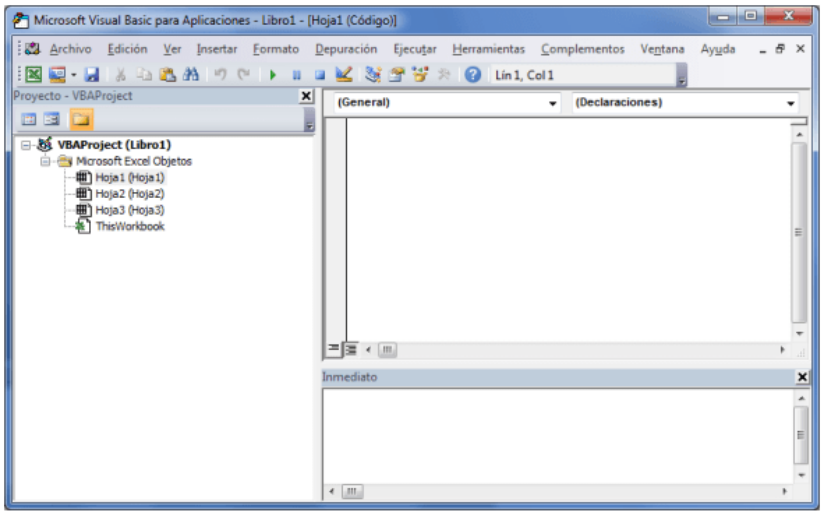


Excel nos proporciona muchas herramientas para manipular la información de la hoja de cálculo pero en ocasiones, deseamos realizar las acciones repetitivas en Excel de una manera más sencilla o crear funcionalidades que no están contempladas en la aplicación de Office**.**

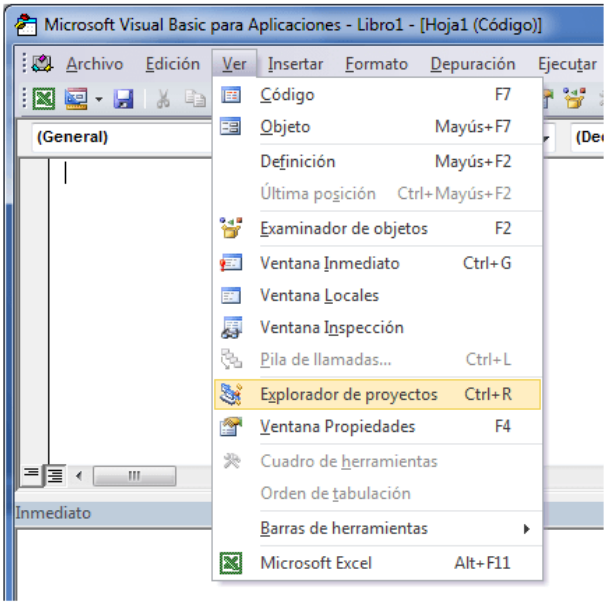
Visual Basic funcionará a través de macros, que nos permitirán crear tareas automatizadas paso a paso. Este lenguaje de programación nos servirá para facilitarnos el trabajo en Excel.

Existen al menos dos alternativas para abrir este editor, la primera de ellas es a través del botón Visual Basic de la ficha Programador.

El segundo método para abrir este programa es, en mi opinión, el más sencillo y rápido y que es a través del atajo de teclado: ALT + F11. El Editor de Visual Basic contiene varias ventanas y barras de herramientas.



En la parte izquierda se muestra el Explorador de proyectos el cual muestra el proyecto VBA creado para el libro actual y además muestra las hojas pertenecientes a ese libro de Excel. Si por alguna razón no puedes visualizar este módulo puedes habilitarlo en la opción de menú Ver y seleccionando la opción *Explorador de proyectos*.



El Explorador de proyectos también nos ayuda a crear o abrir módulos de código que se serán de gran utilidad para reutilizar todas las funciones de código VBA que vayamos escribiendo.

Dentro del Editor de Visual Basic puedes observar una ventana llamada Inmediato que está en la parte inferior. Esta ventana es de mucha ayuda al momento de escribir código VBA porque permite introducir instrucciones y observar el resultado inmediato. Además, desde el código VBA podemos imprimir mensajes hacia la ventana Inmediato con el comando *Debug.Print* de manera que podamos depurar nuestro código. Si no puedes observar esta ventana puedes mostrarla también desde el menú Ver.

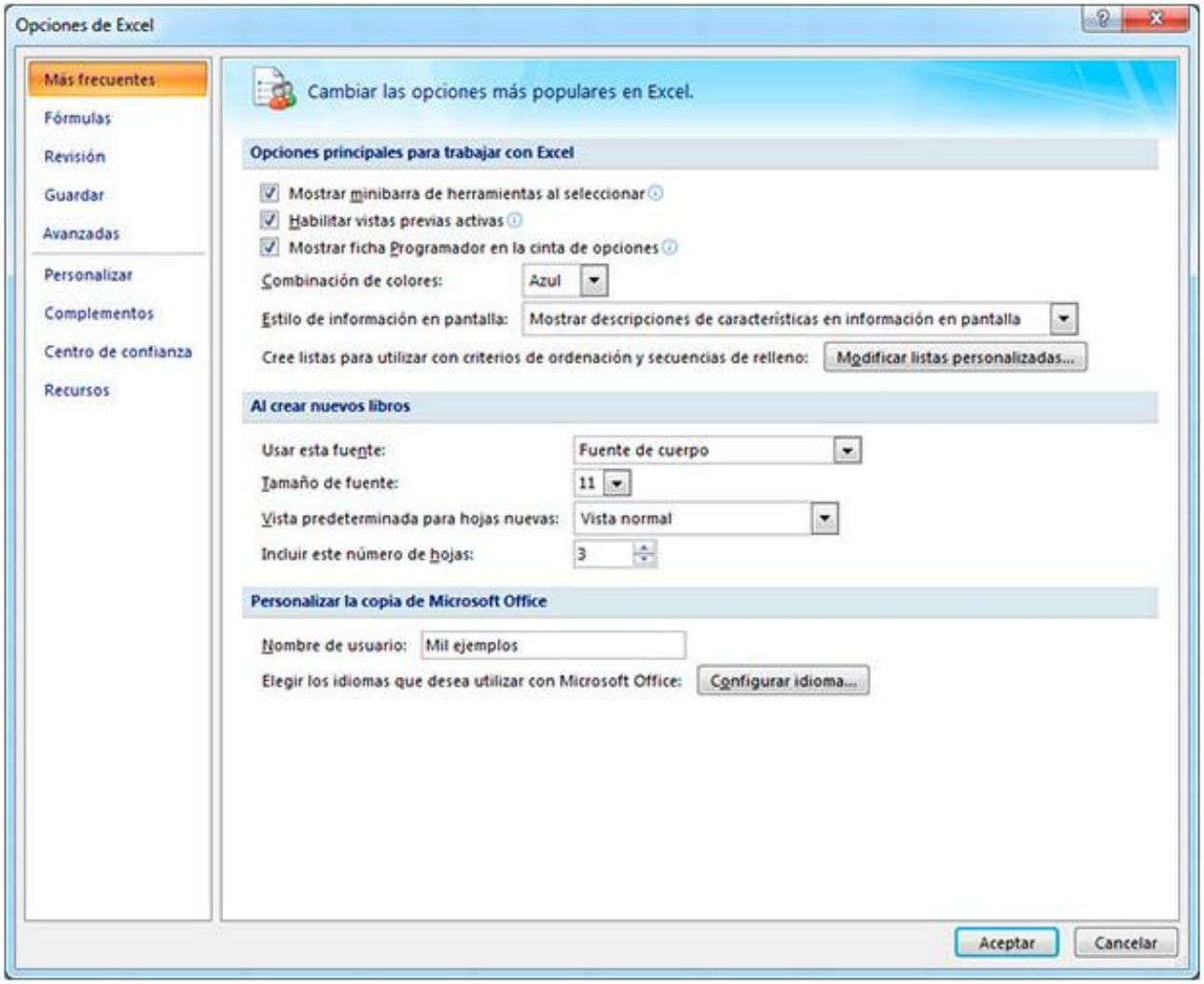
El área más grande en blanco es donde escribiremos el código VBA. Es en esa ventana en donde escribimos y editamos las instrucciones VBA que dan forma a nuestras macros.

Es importante familiarizarnos con el Editor de Visual Basic antes de iniciar con la creación de macros

CÓMO ACTIVAR VISUAL BASIC EN EXCEL

Para empezar a trabajar con este lenguaje de programación, deberemos **activar Visual Basic en Excel**. Para ello, habilitaremos la **ficha de Programador** siguiendo los pasos indicados:

- ✓ Hacer clic en el botón de Office y seleccionar Opciones de Excel.
- ✓ Elegiremos la opción más frecuente de la lista de opciones.
- ✓ En el panel de la derecha, seleccionamos la opción mostrar ficha de programador en la cinta de opciones y aceptamos la cambios.
- ✓ Se mostrará la ficha de programador en la cita de opciones.



CÓMO INSERTAR MACROS EN EXCEL CON VISUAL BASIC

En el momento que creemos un **macro en Excel** podremos utilizarlo cuando lo necesitemos. Para **insertar macros en Excel con Visual Basic** debes seguir los pasos que te indique tu catedrático/a, ya que posee mayor conocimiento acerca del lenguaje de programación **Visual Basic** para acceder a las funcionalidades de Excel. Además, podremos utilizar las plantillas de Excel plantillas de Excel para formatear la hoja de cálculo e **insertar macros** de forma fácil.

CREAR MACROS EN EXCEL CON VISUAL BASIC

Podremos crear **macros en Excel** gracias a la herramienta que dispone Excel conocida como **grabar macro**.

La **grabación del macro** permitirá realizar de forma automática las tareas repetitivas sin necesidad de saber programar. Y entonces, ¿por qué necesitamos saber Visual Basic? Para poder crear nuevas funcionalidades de las que no dispone Excel tendremos que introducirnos en el mundo de la programación.

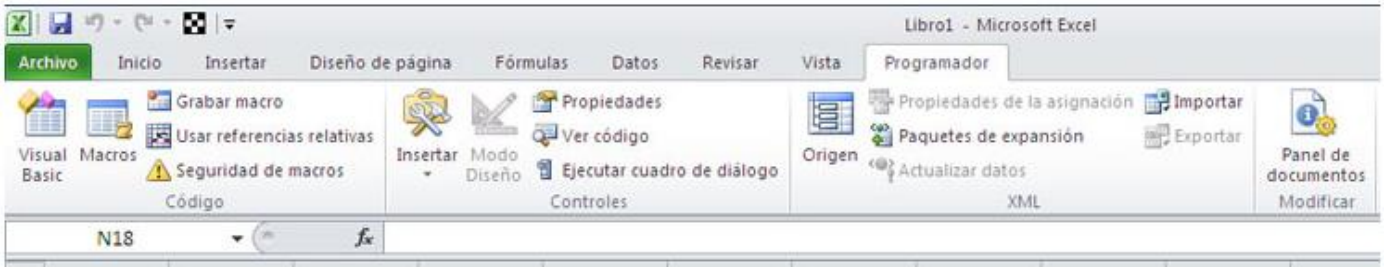
FICHA PROGRAMADOR

Las aplicaciones de Office (en este caso Excel) utilizan la cinta de opciones. La ficha **Programador** es una de las fichas incluidas en la cinta de opciones, donde se puede tener acceso al Editor de Visual Basic y a otras herramientas de programador. Debido a que Excel (en su versión 2010) no muestra la ficha **Programador** de manera predeterminada, debes habilitarla mediante el siguiente procedimiento:

Para habilitar la ficha Programador:

1. En la ficha **Archivo**, elija **Opciones** para abrir el cuadro de diálogo **Opciones de Excel**.
2. Cliquea en **Personalizar cinta de opciones** en el lado izquierdo del cuadro de diálogo.
3. En **Comandos disponibles en:** en el lado izquierdo del cuadro de diálogo, selecciona **Comandos más utilizados**.
4. En **Personalice esta cinta de opciones**, en el lado derecho del cuadro de diálogo, selecciona **Fichas principales** y, a continuación, activa la casilla de verificación **Programador**.
5. Cliquea en **Aceptar**.

Después de que Excel muestre en pantalla la ficha Programador, observa la ubicación de los botones Visual Basic, Macros y Seguridad de macros en la ficha.



PROBLEMAS DE SEGURIDAD

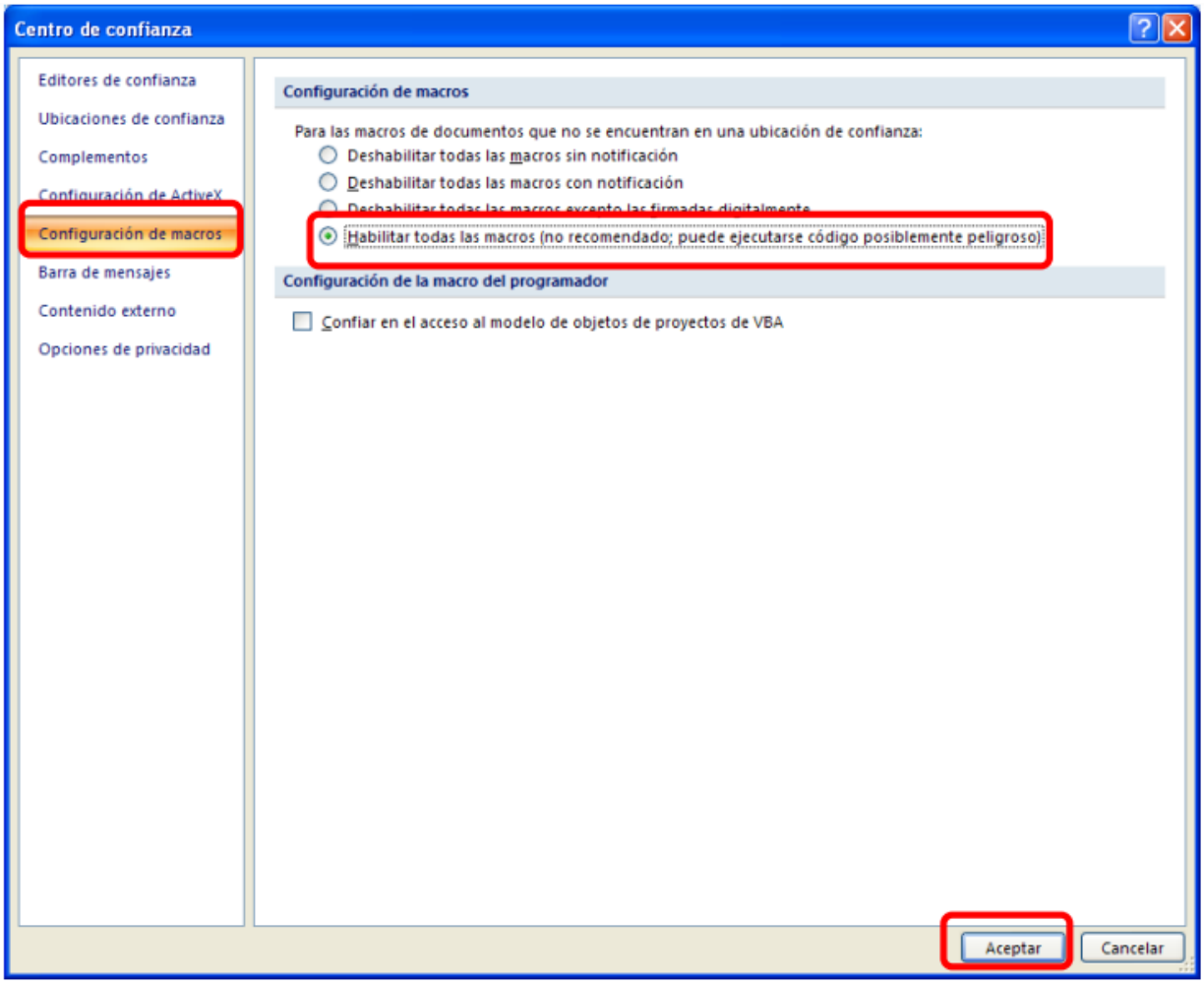
Cliquea en el botón de **Seguridad de macros** para especificar qué macros pueden ejecutarse y en qué condiciones. Aunque el código de macros de sistemas no confiables puede dañar gravemente el equipo, las condiciones de seguridad que impiden ejecutar macros útiles pueden disminuir en gran medida la productividad. La seguridad de macros es un tema complejo que se debe estudiar y comprender, si se trabaja con macros de Excel.

Para el propósito del curso, es importante que tengas en cuenta, que si la barra **Advertencia de seguridad: las macros se han deshabilitado** aparece entre la cinta de opciones y el libro, cuando abre un libro que contiene una macro, puede hacer clic en el botón **Habilitar contenido** para habilitar las macros.

Además, como medida de seguridad, no puedes guardar una macro en el formato de archivo predeterminado de Excel (.xlsx); debes guardar la macro en un archivo con extensión especial (.xlsm).
Para establecer el nivel de seguridad de manera que estén habilitadas temporalmente todas las macros, realiza lo siguiente:

En la ficha Programador, en el grupo Código, cliquea en Seguridad de macros .

Se visualiza:



En Configuración de macros, cliquea en **Habilitar todas las macros** (no recomendado; puede ejecutarse código posiblemente peligroso) y, a continuación, cliquea en **Aceptar.**

NOTA. Para ayudar a evitar que se ejecute el código potencialmente peligroso, recomendamos que vuelvas a cualquiera de las configuraciones que deshabilitan todas las macros cuando termine de trabajar con las macros. En la ficha Programador, en el grupo Código, cliquea en Grabar macro. En el cuadro Nombre de la macro, escribe un nombre para la macro.

NOTA. El primer carácter del nombre de la macro debe ser una letra. Los caracteres siguientes pueden ser letras, números o caracteres de subrayado. No se permiten espacios en un nombre de macro, caracteres especiales ni palabras reservadas; puede utilizarse un carácter de subrayado como separador de palabras.

Si utilizas un nombre de macro que también es una referencia de celda, puede aparecer un mensaje indicando que el nombre de la macro no es válido. Para asignar una combinación de tecla de método abreviado (método abreviado: tecla o combinación de teclas de función, como F5 o CTRL + a, que utiliza para ejecutar un comando. Una tecla de acceso, por lo contrario es un combinación de teclas, como ALT + f, que mueve el enfoque a un menú, comando o control.) Con CTRL para ejecutar la macro, en el cuadro Tecla de método abreviado, escribe cualquier letra en mayúsculas o minúsculas que desee utilizar.

NOTA. La tecla de método abreviado suplantará a cualquier tecla de método abreviado predeterminada equivalente en Excel mientras esté abierto el libro que contiene la macro.

En la lista Guardar macro en, selecciona el libro en el que deseas almacenar la macro.

En la lista Guardar macro en, selecciona el libro en el que deseas almacenar la macro. Sugerencia Si deseas que la macro esté disponible siempre que utilices Excel, selecciona Libro de macros personal. Cuando se selecciona Libro de macros personal, Excel crea un libro oculto de macros personal (Personal.xlsb), si no existe todavía, y guarda la macro en este libro.

Si deseas que se ejecute automáticamente una macro del libro de macros personal en otro libro, también debe guardar ese libro en la carpeta XLStart, de forma que ambos libros se abran cuando se inicie Excel.

1. Para incluir una descripción de la macro, escriba el texto que desee en el cuadro Descripción.
2. Cliquea en **Aceptar** para iniciar la grabación.
3. Realiza las acciones que deseas grabar.
4. En la ficha **Programador**, en el grupo **Código**, cliquea en **Detener grabación.**

SUGERENCIA. También puedes hacer clic en **Detener grabación** en el lado izquierdo de la barra de estado.

REALIZA LO SIGUIENTE...

Genera las siguientes Macros:

1. Graba una Macro que se active con Control + b y que esta macro permita abrir un archivo.
2. Graba una Macro que inserte una tabla con datos.
3. Graba una Macro que abra un archivo existente.
4. Graba una Macro que abra un nuevo archivo.
5. Graba una Macro que inserte un logotipo.
6. Graba una Macro que ordene alfabéticamente una lista de nombres.
7. Graba una Macro que imprima un formulario.

EDITOR DE VISUAL BASIC

El siguiente procedimiento muestra cómo crear un nuevo libro en blanco donde se almacenarán las macros. A continuación, podrás guardar el libro con el formato .xlsm.

Para crear un nuevo libro en blanco:

1. Clique en el botón **Macros**, de la ficha Programador.
2. En el cuadro de diálogo **Macro** que aparece, escribe **Hello** en **Nombre de macro**.
3. Cliquea en el botón **Crear** para abrir el Editor de Visual Basic que incluirá los esquemas de una nueva macro ya escritos.

El lenguaje de programación Visual Basic es un lenguaje muy completo; y, en consecuencia, tiene un entorno de programación completo. En este artículo solo se estudian las herramientas que se usan para empezar a trabajar en programación sin incluir la mayoría de las herramientas del Editor de Visual Basic. Realizada esta salvedad, cierra la ventana **Propiedades** en el lado izquierdo del Editor de Visual Basic e ignora las dos listas desplegables que aparecen sobre el código.

El Editor de Visual Basic contiene el siguiente código.

```vb
Sub Hello()

End Sub
```

Sub se refiere a *Subrutina*que, por el momento, se puede definir como "macro". Al ejecutar la macro Hello se ejecuta cualquier código que se encuentre entre Sub Hello() y End Sub.

Ahora, edita la macro para que tenga un aspecto similar al siguiente código.

```vb
Sub Hello()
   MsgBox ("Hello, world!")
End Sub
```

Vuelve a la ficha **Programador** en Excel y cliquea de nuevo en el botón **Macros**.

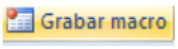
Selecciona la macro **Hello** en la lista que aparece y, a continuación, cliquea en **Ejecutar** para mostrar un cuadro de mensaje pequeño que contiene el texto "Hello, world!".

Acabas de crear e implementar correctamente código de VBA personalizado en Excel. Cliquea en **Aceptar** en el cuadro de mensaje para cerrarlo y terminar de ejecutar la macro.

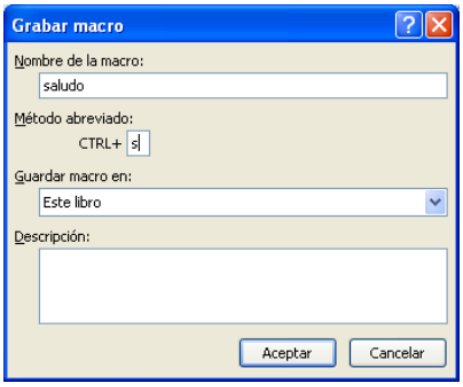
Si no aparece el cuadro de mensaje, comprueba la configuración de seguridad de la macro y reinicie Excel.

CÓDIGOS DE UNA MACRO DE EXCEL

Para observar los códigos de una macro debemos seguir los pasos:

1. En primer lugar selecciona la celda B5 antes de empezar la grabación de la Macro.
2. Presiona el Botón 

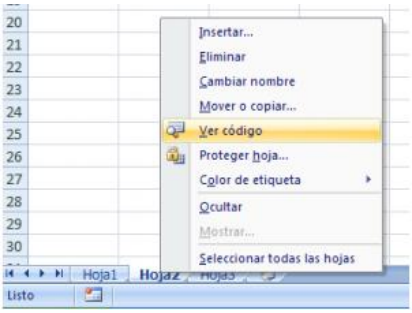
del grupo **Código MS Excel** muestra el cuadro de dialogo **Grabar Macro**:



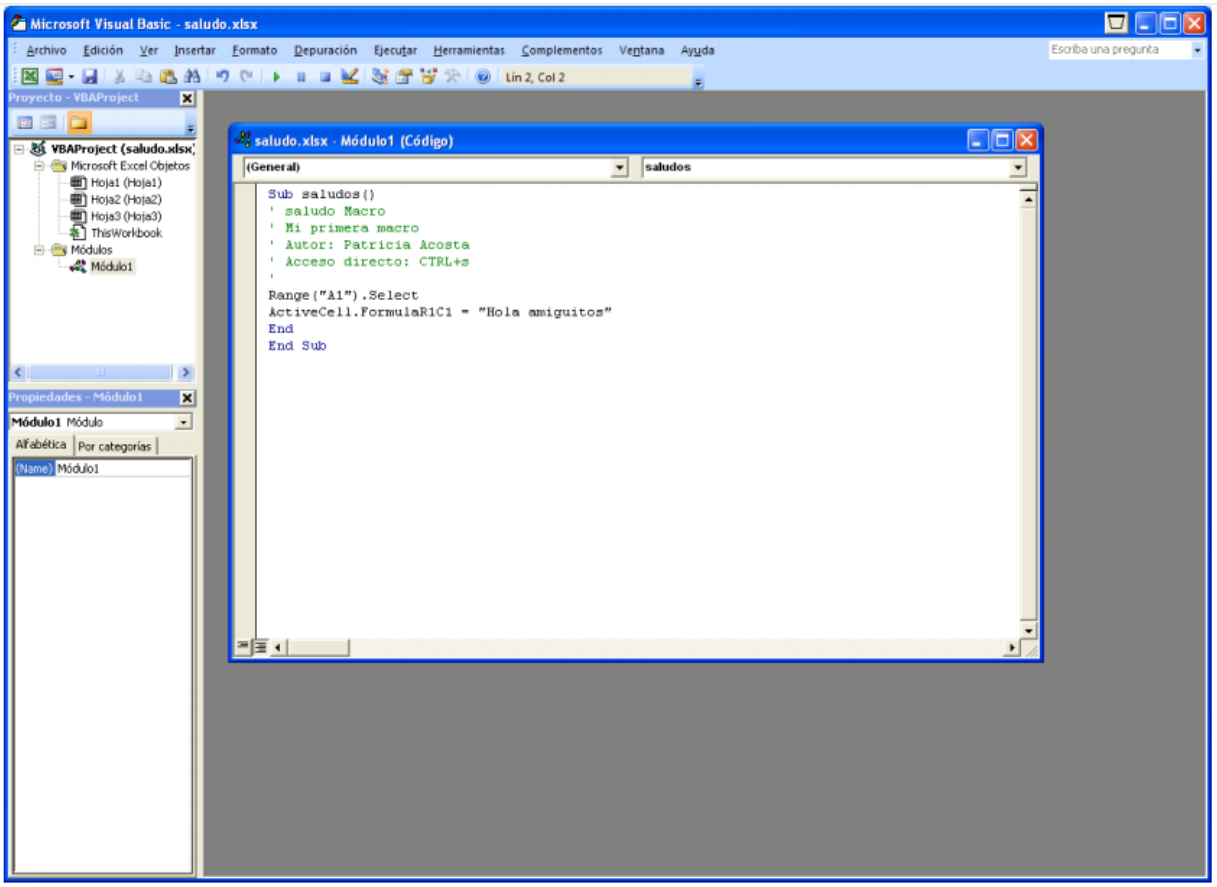
3. Ingresa un nombre de la macro por ejemplo saludo.
4. En la opción **Método Abreviado** escribe la letra s, por lo tanto la macro se llamara con Control + s.
5. En **Guardar macro** en: Selecciona en el lugar en donde deseas guardar la macro, por ejemplo Este libro.
6. En **Descripción** puedes agregar una descripción de lo que hace la macro, este punto es opcional. Solo te sirve para que recuerdes acerca de lo que hace la macro, pues este código no es interpretado por el compilador.
7. Presiona el botón **Aceptar**. Excel inicia la grabación de la Macro.

8. Trasládate a la celda A1 y escriba Hola amiguitos, después presiona **Enter** para aceptar el valor en la celda.
9. Detén la grabación de la macro presionando el botón 

del **grupo Códig**o. Excel ha grabado los pasos y ha generado un código.
10. Para visualizar el código generado, presiona la tecla Alt + la tecla de función F11(Alt + F11), o da un clic derecho en la hoja de cálculo:



11. Selecciona la opción **Ver código**. También puedes acceder **al grupo Código**, al cliquear en la opción Visual Basic.
12. Excel nos traslada al **Editor** de Visual Basic. Se visualiza:



13. Active los siguientes cuadros o ventanas:

    ✓ Cliquea en el Menú Ver y elija la opción Explorador de Proyectos.
    ✓ Cliquea en el Menú Ver y elija la opción Ventana Propiedades.

14. Del **cuadro Proyecto** da doble clic en Módulos o simplemente presiona el signo de + que aparece en la **opción Módulos**. Se activara debajo de *Módulos* la **Opción Modulo1**.

15. Da doble clic en **Modulo1**. Se mostrara en el Editor de Visual Basic el código de la macro que grabamos de la siguiente forma:

```
Sub saludo()
    '' saludo Macro
    ' Mi primera macro
    ' Autor: Patricia Acosta
    ' Acceso directo: CTRL+s
    'Range("A1").Select
    ActiveCell.FormulaR1C1 = "Hola amiguitos"
    End Sub
```

16. Que es lo que significa esto nos preguntaremos asombrados, a continuación se da una explicación de lo que ha hecho Excel:

- ✓ Sub y End Sub indican el inicio y el final del procedimiento de la macro saludo.
- ✓ Todo lo que aparece con un apóstrofe ´ indica que no se tomara en cuenta que es solo texto o comentarios y ese texto aparece en color verde.
- ✓ Range("A1").Select Indica que lo primero que hicimos al grabar la macro fue trasladarnos a la celda A1. La orden Range nos permite trasladarnos a una celda.
- ✓ ActiveCell.FormulaR1C1 = "Hola amiguitos" Esto indica que se escribirá en la celda en que se encuentra el valor de texto Hola amiguitos. Todo lo que aparece entre comillas siempre será un valor de texto. La orden ActiveCell.FormulaR1C1 nos permite escribir un valor en la celda activa. Para comprender alteraremos el código dentro del editor de Visual Basic.

```
Sub saludo()
' saludo Macro
' Mi primera macro
' Autor: Patricia Acosta
' Acceso directo: CTRL+s
Range("A1").Select
ActiveCell.FormulaR1C1 = "Hola amiguitos"
Range("B1").Select
ActiveCell.FormulaR1C1 = "Bienvenidos al curso de Excel"
End Sub
```

17. Al alterar el código, y cuando regreses a Excel y ejecutes la macro con Control + s hará lo siguiente: En A1 escribirá Hola amiguitos. En B1 escribire Bienvenidos al curso de Excel.

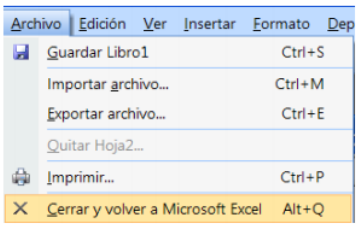
Se visualiza:

```
Sub saludos()
' saludo Macro
' Mi primera macro
' Autor: Patricia Acosta
' Acceso directo: CTRL+s
'
Range("A1").Select
ActiveCell.FormulaR1C1 = "Hola amiguitos"
'
Range("B1").Select
ActiveCell.FormulaR1C1 = "Bienvenidos al seminario de Excel"
End
End Sub
```

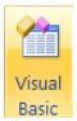
Al alterar el código y cuando regreses a Excel y ejecutes la macro con Control + s hará: En A1 escribirás Hola amiguitos. En B1 escribirás Bienvenidos al seminario de Excel.

	A	B
1	Hola amiguitos	Bienvenidos al seminario de Excel
2		

Para salir del editor cliquea en el **Menú Archivo** y elije la **opción Cerrar** y volver a Microsoft Excel.



Si no desea salir por completo de clic en el botón Microsoft Excel.

Cuando desees volver al editor da clic en: la **pestaña Programador**. De clic en el icono 

del grupo Código

REALIZA LO SIGUIENTE...

1. Genera una Macro que escriba un nombre en una celda y lo ponga negrita y observe el Código.
2. Genera una Macro que escriba un nombre en una celda y lo Centre y observe el Código.
3. Genera una Macro que escriba un nombre en una celda y cambie el tamaño de la letra a 20 puntos y observa el Código.

ACCESIBILIDAD DE LAS MACROS

También puedes tener acceso al cuadro de diálogo **Macros** desde la ficha **Ver**, pero si utilizas una macro con frecuencia, te resultará más cómodo tener acceso a ella mediante un método abreviado de teclado o un botón de la **Barra de herramientas de acceso rápido**.

Para crear un botón para la macro **Hello** en la **Barra de herramientas de acceso rápido**, utiliza el siguiente procedimiento.

El siguiente procedimiento describe cómo crear un botón para una macro en la Barra de herramientas de acceso rápido:

Para crear un botón para una macro en la **Barra de herramientas de acceso rápido**:

1. Cliquea en la pestaña **Archivo**.
2. Cliquea en **Opciones** para abrir el cuadro de diálogo **Opciones de Excel** y, a continuación, cliquea en **Barra de herramientas de acceso rápido**.
3. En la lista que se encuentra en **Comandos disponibles en:** elija **Macros**. Busca en la lista el texto que es similar a **Book1!Hello** y selecciónalo.
4. Cliquea en el botón **Agregar >>** para agregar la macro a la lista en el lado derecho y, a continuación, cliquea en el botón **Modificar...**, a fin de seleccionar una imagen del botón para asociar a la macro.
5. Cliquea en **Aceptar**. El nuevo botón deberá mostrarse en la **Barra de herramientas de acceso rápido**, encima de la ficha **Archivo**.

Ahora puedes ejecutar rápidamente la macro en cualquier momento sin tener que usar la ficha **Programador**: inténtalo.

CÓDIGOS MÁS COMUNES

Trasladarse a una Celda

```
Range("A1").Select
```

Escribir en una Celda

```
Activecell.FormulaR1C1="Paty Acosta"
```

Letra Negrita

```
Selection.Font.Bold = True
```

Letra Cursiva

```
Selection.Font.Italic = True
```

Letra Subrayada

```
Selection.Font.Underline = xlUnderlineStyleSingle
```

Centrar Texto

```
With Selection
    .HorizontalAlignment = xlCenter
End With
```

Alinear a la izquierda

```
With Selection
    .HorizontalAlignment = xlLeft
End With
```

Alinear a la Derecha

```
With Selection
    .HorizontalAlignment = xlRight
End With
```

Tipo de Letra(Fuente)

```
With Selection .Font
    .Name = "AGaramond"
End With
```

Tamaño de Letra(Tamaño de Fuente)

```
With Selection.Font
    .Size = 15
End With
```

Copiar

```
Selection.Copy
```

Pegar

```
ActiveSheet.Paste
```

Cortar

```
Selection.Cut
```

Ordenar Ascendente

```
Selection.Sort Key1:=Range("A1"), Order1:=xlAscending, Header:=xlGuess, _ OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
```

Orden Descendente

```
Selection.Sort Key1:=Range("A1"), Order1:=xlDescending, Header:=xlGuess, _ OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
```

Buscar

```
Cells.Find(What:="Paty Acosta", After:=ActiveCell, LookIn:=xlFormulas, LookAt _ :=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:= _ False).Activate
```

Insertar Fila

```
Selection.EntireRow.Insert
```

Eliminar Fila

```
Selection.EntireRow.Delete
```

Insertar Columna

```
Selection.EntireColumn.Insert
```

Eliminar Columna

```
Selection.EntireColumn.Delete
```

Abrir un Libro

```
Workbooks.Open Filename:="C:\Mis documentos\miarchivo.xls"
```

Grabar un Libro

```
ActiveWorkbook.SaveAs Filename:="C:\Mis documentos\tauro.xls", FileFormat _ :=xlNormal, Password:="", WriteResPassword:="", ReadOnlyRecommended:= _ False, CreateBackup:=False
```

USO DE LA GRABADORA DE MACROS

A veces una simple macro grabada es todo lo que se necesita; en estos casos, incluso, no es necesario mirar el código. Por lo general, la grabación por sí sola no es suficiente, pero sí, un punto de inicio en el siguiente proceso.

Para usar la grabadora de macros como punto de inicio de la solución:

1. Graba las acciones que desea codificar.
2. Revisa el código y busca las líneas que realizan esas acciones.
3. Elimina el resto del código.
4. Modifica el código grabado.
5. Agrega variables, estructuras de control y otro código que la grabadora de macros no pueda grabar.

Comienza tu investigación con la grabación de una macro que cambie el nombre de una hoja de cálculo a **New Name**. Después podrás utilizar la macro grabada para crear tu propia macro que cambie el nombre de varias hojas de cálculo según su contenido.

Para grabar una macro que cambia el nombre de una hoja de cálculo:

1. Cliquea en **Grabar macro** en la ficha **Programador**.
2. Coloca el nombre **RenameWorksheets** a la macro, cambia el nombre Sheet1 a **New Name** y, a continuación, cliquea en **Detener grabación**.
3. Vé a la ficha **Programador** o **Ver**, cliquea en el botón **Macros** y elije **Editar** para abrir el Editor de Visual Basic.

En el Editor de Visual Basic, el código debe verse similar al siguiente.

```vb
Sub RenameWorksheets()
'
' RenameWorksheets Macro
'
'

    Sheets("Sheet1").Select
    Sheets("Sheet1").Name = "New Name"
End Sub
```

Las primeras cuatro líneas que aparecen después de la línea **Sub** son comentarios. Toda línea que comienza con un apóstrofe es un comentario y no tiene efecto alguno sobre la acción que ejecuta la macro.

Los principales usos de los comentarios son los siguientes:

- ✓ Facilitar la comprensión del código, no solo para el usuario, sino para cualquier persona que necesite modificarlo en el futuro.
- ✓ Deshabilitar temporalmente una línea de código (se denomina *marcar como comentario*).

Los cuatro comentarios en esta macro grabada no cumplen ningún propósito, por lo tanto, elimínelos.

La siguiente línea usa el método **Select** para seleccionar el miembro Sheet1 del objeto de la colección **Sheets**. En el código de VBA, por lo general, no es necesario seleccionar objetos antes de manipularlos, aunque eso es lo que hace la grabadora de macros. En otras palabras, esta línea de código es redundante, por lo tanto, puede eliminarla.

La última línea de la macro grabada modifica la propiedad Nombre del miembro **Sheet1** de la colección **Sheets**.

Esta es la línea que debe conservar.

Después de realizar los cambios, el código grabado ahora debe ser similar al siguiente.

```vb
Sub RenameWorksheets()
    Sheets("Sheet1").Name = "New Name"
End Sub
```

Vuelve a cambiar la hoja denominada New Name a Sheet1 de forma manual y, a continuación, ejecuta la macro. El nombre debe volver a cambiar a New Name.

MODIFICACIÓN DEL CÓDIGO GRABADO

Ahora ha llegado el momento de investigar la colección **Sheets** que la grabadora de macros usó. El tema Sheets en la referencia del modelo de objetos incluye el siguiente texto.

"La colección **Sheets** puede contener los objetos **Chart** o **Worksheet**. Si necesita trabajar con hojas de un solo tipo, consulta el tema de objetos para ese tipo de hoja".

Como está trabajando solo con **Worksheets**, entonces, cambie el código para que se vea de la siguiente forma.

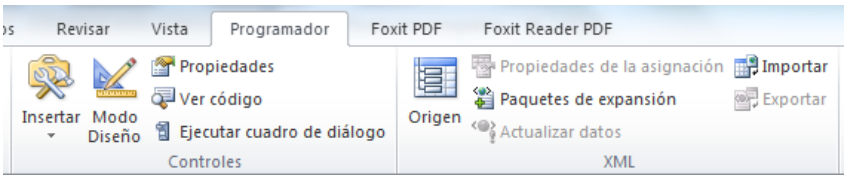
VB

```
Sub RenameWorksheets()
    Worksheets("Sheet1").Name = "New Name"
End Sub
```

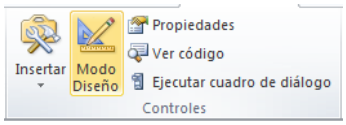
CUADRO DE CONTROL – CONTROLES ACTIVEX

Una de las opciones más interesantes que tiene el Excel es la de utilizar los "cuadros de control". Los cuadros de control se usan para crear verdaderos programas en Excel y pueden ser de mucha utilidad.

Esta herramienta está ubicada en:

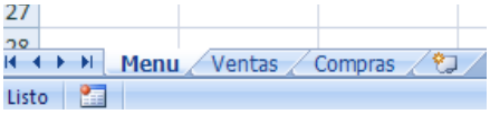


Este grupo de Controles cuenta con tres opciones muy importantes como:



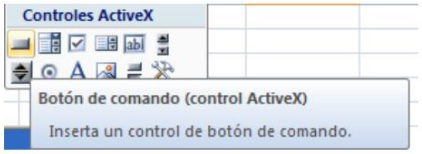
1. Modo diseño: permitirá trabajar en el diseño de los controles de ActiveX.
2. Propiedades: permiten activar la propiedad de cada control.
3. Ver código: permite agregar código a cada control.

Para iniciar crea las hojas: Menú, Ventas y Compras:



Selecciona la hoja Menú para allí crear dos botones.

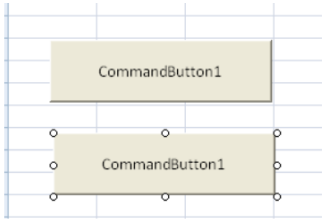
Para trabajar con estos controles en necesario Activar el modo de diseño y dar clic en **Insertar**, selecciona el **Botón de comando**.

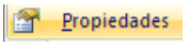


Esta vez haremos un botón que cuando se presione pase a otra hoja del Excel. Por ejemplo se puede hacer un menú con varios botones que al presionarlos pasen a las distintas opciones.

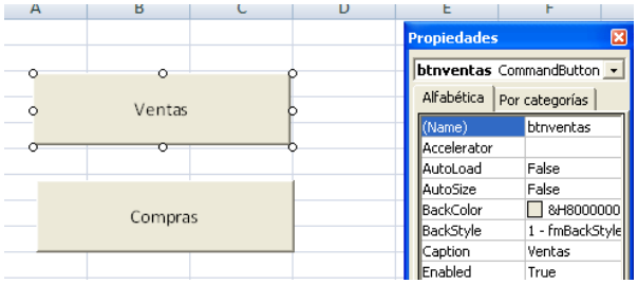
En la hoja Menú crea dos "botones de comando".

Por Ejemplo:



Selecciona el primer botón y cliqueando en la opción y 

muestre las propiedades. Cambia la Propiedad "Caption" por: "Ventas" En Name: btnventas

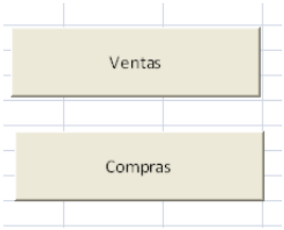


Selecciona el segundo botón y muestra las propiedades…

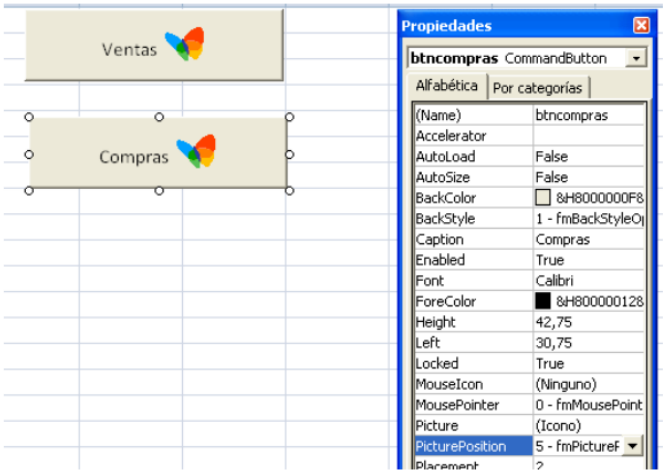
Cambia la Propiedad "Caption" por: "Compras"

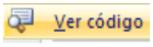
En Name: btncompras

Si pudiste realizar correctamente estos pasos. Debería de ver lo siguiente:



Si deseas colocar iconos en los botones seleccione la **propiedad Picture** e inserte una imagen de extensión .ico.



Para que se visualice el texto cambia la posición **PicturePosition** a: 5 Selecciona el primer botón y cliquea en ver código 

.

En esta parte se abrirá el Editor de Visual Basic y debe escribir lo siguiente: Hoja2.Activate

Cierra el editor de Visual Basic (nota: cada vez que cierre el editor de Visual Basic, hazlo del cuadro de cerrar "X" que está más arriba, porque puede confundirse y cerrar la ventana de editar código, no te preocupes que no está cerrando Excel.) Selecciona el segundo botón y cliquea en ver código 

Escriba: Hoja3.activate

Salga del modo de diseño y navegue con los botones que programó. Más adelante utilizaremos estos botones para cargar formularios desde VBA en Excel.

O añada el código de las macros que grabó con la grabadora.

CREANDO FORMULARIOS Y PROGRAMÁNDOLOS

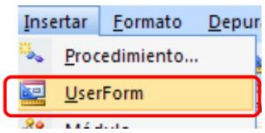
Ahora aprenderemos a dominar lo máximo de Excel que es crear formularios y programarlos, bueno un formulario es una ventana que se programa por medio de controles y estos controles responden a sucesos que nosotros programamos. Todo esto se encuentra dentro de Visual Basic.

A continuación Muestro como crear un formulario y como programarlo:

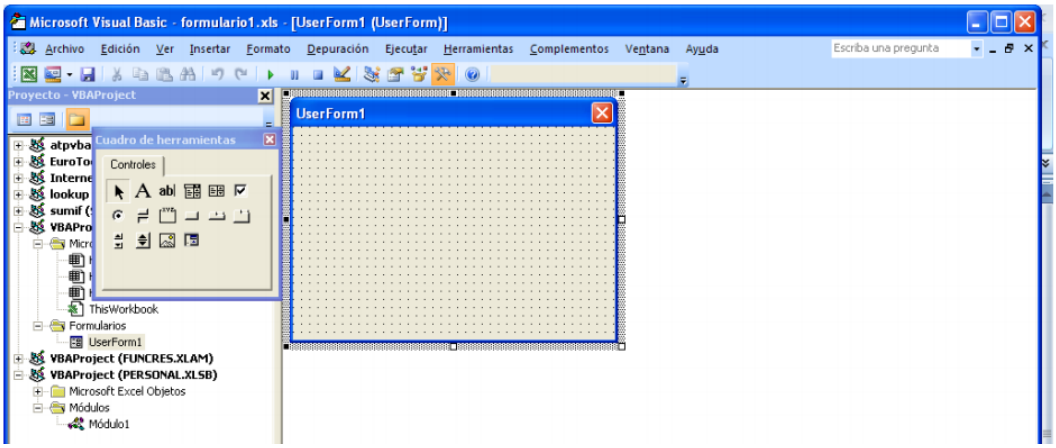
1. Presione La Teclas Alt + F11, para entrar al editor de Visual Basic.
2. Activa las siguientes opciones:

   - ✓ De clic en el Menú Ver y elija la opción Explorador de Proyectos
   - ✓ De clic en el Menú ver y elija la opción Ventana Propiedades

3. Del Menú Insertar elija la Opción UserForm.

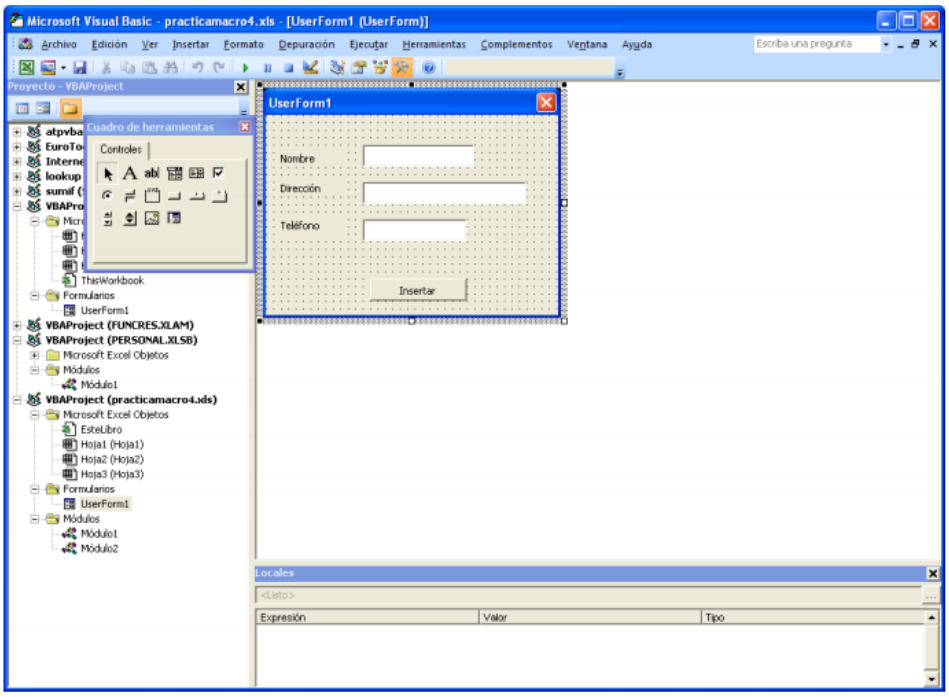


Esto inserta el Formulario que programaremos con controles. En el Explorador de Proyecto se observara que se insertó el UserForm.

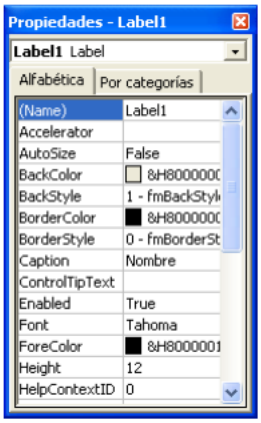


También cuando de clic en el Formulario USERFORM1 se debe de activar el Cuadro de Herramientas, si no se activa de clic en el Menú Ver y elija la opción Cuadro de Herramientas.

4. Elija del Cuadro de Herramientas el Control Etiqueta el que tiene la A y Arrastre dibujando en el Formulario USERFORM1 la etiqueta. Quedará el nombre Label1, después de un clic en la etiqueta dibujada y podrá modificar el nombre de adentro y pondremos ahí Nombre. Si por error da doble clic en la etiqueta y lo manda a la pantalla de programación de la etiqueta, solo de doble clic en UserForm1 que se encuentra en el Explorador de Proyecto.
5. Elija del Cuadro de Herramientas el control Cuadro de Texto el que tiene ab y arrastre dibujando en el formulario USERFORM1 el cuadro de texto a un lado de la etiqueta que dice Nombre. El cuadro de texto debe de estar vacío y su nombre será Textbox1, el nombre solo aparecerá en el control.
6. Haga los dos pasos anteriores igualmente poniendo Dirección en la Label2 y Teléfono en la Label3 y también dibújeles su Textbox. Esto quedara así después de haberlo hecho.



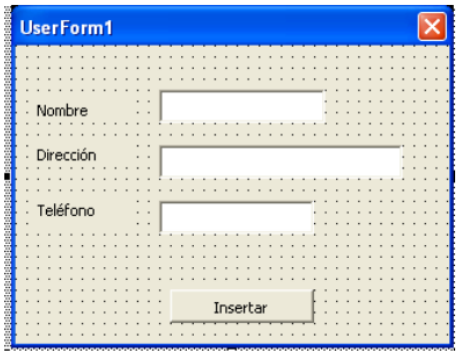
Si tiene algún problema al dibujar las etiquetas o los cuadros de texto, solo cámbiele el nombre a la etiqueta o el cuadro de texto en la Ventana Propiedades la opción se llama (Name). El Error que marque puede ser Nombre Ambiguo, pero si le cambia el Nombre al control se quitara el error. Puede ponerle cualquier nombre en lugar de Label1.



Los controles como las Etiquetas y Cuadros de Textos pueden modificárseles algunas opciones en la Ventana Propiedades Para hacer esto es necesario tener conocimiento sobre las propiedades de los controles. No altere las propiedades si no las conoce.

7. Elija del Cuadro de Herramientas el control Botón de Comando y Arrastre dibujando en el Formulario USERFORM1 el Botón, después de un clic en el nombre del Botón dibujado y podrá modificar el nombre

y pondremos ahí Insertar. Si por error da doble clic en la Botón y lo manda a la pantalla de programación de la etiqueta, solo de doble clic en UserForm1 que se encuentra en el Explorador de Proyecto. Así quedara el Formulario formado por los controles:



8. Ahora de doble clic sobre el control Textbox1 para programarlo y después inserte el siguiente código:

```
Private Sub TextBox1_Change()
Range("A9").Select
ActiveCell.FormulaR1C1 = TextBox1
End Sub
```

Esto indica que se valla a A9 y escriba lo que hay en el **Textbox1.**

Nota. Lo que está en azul lo genera Excel automáticamente, usted solo escribirá lo que está en Negrita. Para volver al Formulario y programar el siguiente Textbox de doble clic en UserForm1 que se encuentra en el Explorador de Proyecto, o simplemente de clic en Ver Objeto en el mismo Explorador de Proyecto.

9. Ahora de doble clic sobre el control Textbox2 para programarlo y después inserte el siguiente código:

```
Private Sub TextBox2_Change()
Range("B9").Select
ActiveCell.FormulaR1C1 = TextBox2
End Sub
```

Esto indica que se valla a B9 y escriba lo que hay en el Textbox2. Para volver al Formulario y programar el siguiente Textbox de doble clic en UserForm1 que se encuentra en el Explorador de Proyecto, o simplemente de clic en Ver Objeto en el mismo Explorador de Proyecto.

10. Ahora de doble clic sobre el control Textbox3 para programarlo y después inserte el siguiente código:

```
Private Sub TextBox3_Change()
Range("C9").Select
ActiveCell.FormulaR1C1 = TextBox2
End Sub
```

Esto indica que se valla a C9 y escriba lo que hay en el Textbox3 Para volver al Formulario y programar el Botón de Comando Insertar de doble clic en UserForm1 que se encuentra en el Explorador de Proyecto, o simplemente de clic en Ver Objeto en el mismo Explorador de Proyecto.

11. Ahora de doble clic sobre el control Botón de Comando para programarlo y después inserte el siguiente código:

```
Private Sub CommandButton1_Click()
'inserta un renglón
Selection.EntireRow.Insert
'Empty Limpia Los Textbox
TextBox1 = Empty
TextBox2 = Empty
TextBox3 = Empty
'Textbox1.SetFocus Envía el cursor al Textbox1 para volver a capturar los
datos
TextBox1.SetFocus
    End Sub
```

Nota. El comando Rem es empleado para poner comentarios dentro de la programación, el comando Empty es empleado para vaciar los Textbox.

12. Ahora presione el botón Ejecutar User/Form que se encuentra en la barra de herramientas o simplemente la tecla de función F5. Se activará el Userform1 y todo lo que escriba en los Textbox se escribirá en Excel y cuando presione el botón Insertar, se insertara un renglón y se vaciaran los Textbox y después se mostrara el cursor en el Textbox1.

TRABAJANDO CON FORMULAS

Es de suma importancia saber aplicar Formulas en Macros de Excel, ya que la mayoría de las hojas de cálculos las involucran, por ejemplo los Inventarios, las Nóminas o cualquier otro tipo de hoja las llevan, es por eso que en la siguiente Fase se muestra cómo manejar Formulas en Macros de Excel.

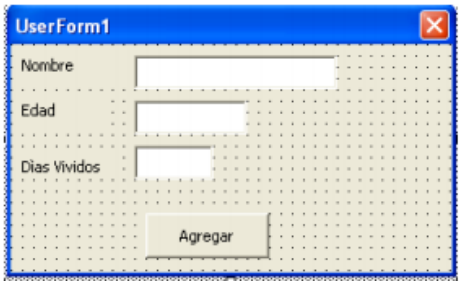
Presiona La Teclas Alt + F11, para entrar al editor de Visual Basic.

Activa las siguientes opciones:

- ✓ Cliquea en el Menú Ver y elija la opción Explorador de Proyectos.
- ✓ Cliquea en el Menú ver y elija la opción Ventana Propiedades.

Del Menú Insertar elija la Opción UserForm. Esto inserta el Formulario que programaremos con controles. En el Explorador de Proyecto se observara que se insertó el UserForm.

Ahora crearas un formulario con el siguiente aspecto:



El formulario tendrá:

- ✓ Tres etiquetas

- ✓ Tres Textbox
- ✓ Un Botón de Comando Los datos que se preguntaran serán Nombre y Edad, los Días Vividos se generaran automáticamente cuando insertes la edad.

A continuación se muestra como se deben de programar estos Controles Programación de los Controles:

```
Private Sub CommandButton1_Click()
Selection.EntireRow.Insert
TextBox1 = Empty
TextBox2 = Empty
TextBox3 = Empty
TextBox1.SetFocus
End Sub
Private Sub TextBox1_Change()

Range("A9").Select

ActiveCell.FormulaR1C1 = TextBox1

End Sub

Private Sub TextBox2_Change()

Range("B9").Select

ActiveCell.FormulaR1C1 = TextBox2

' aquí se crea la Fórmula
 TextBox3 = Val(TextBox2) * 365

 'El Textbox3 guardara el total de la multiplicación del Textbox2 por 365

 'El Comando Val permite convertir un valor de Texto a un Valor Numérico

 'Esto se debe a que los Textbox no son Numéricos y debemos de Convertirlos

 End Sub

 Private Sub TextBox3_Change()

 Range("C9").Select

 ActiveCell.FormulaR1C1 = TextBox3

 End Sub
```

OBSERVANDO LOS CÓDIGOS DE UNA MACRO DE EXCEL

Crearemos una macro y veremos sus códigos:

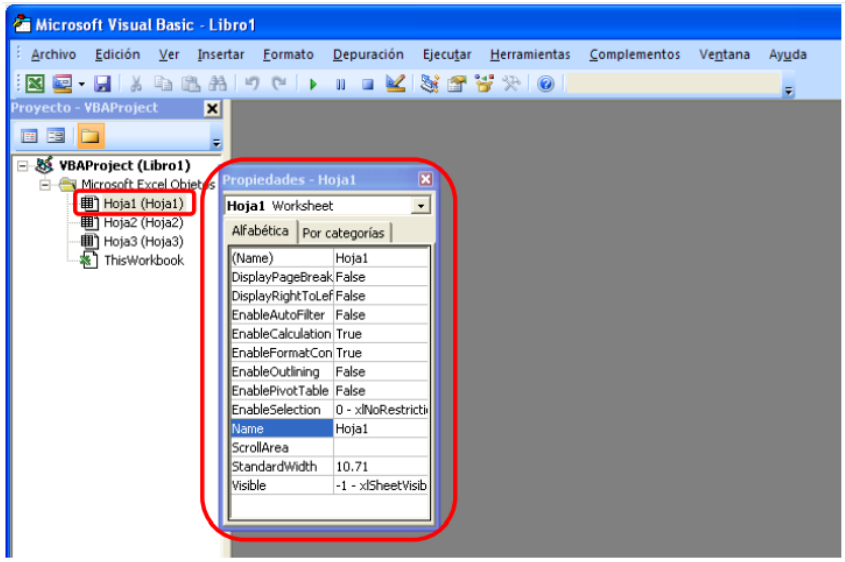
Para observar los códigos de una macro debemos seguir los pasos:

1. Primero debes ubicarte en la celda A5 antes de empezar la grabación de la Macro.
2. Presiona el botón **Grabar Macro** de la **barra de Herramientas** Visual Basic. Excel muestra el **cuadro de Dialogo Grabar Macro**.
3. En la **opción Método Abreviado** escribe la letra r, por lo tanto la macro se llamara con Control + r
4. Presiona el botón **Aceptar**. Excel inicia la grabación de la Macro1
5. Trasládate a la celda A1 y escribe su nombre, después presiona Enter para aceptar el valor en la celda.
6. Detén la grabación de la macro presionando el botón **Detener Grabación** de la barra de herramientas Visual Basic. Excel a grabado los pasos y ha generado un código.

7. Presiona la tecla Alt + la tecla de función F11(Alt + F11). Excel nos traslada al Editor de Visual Basic. Si este editor no se activa es que Excel no está bien instalado o se ha borrado.
8. Active los siguientes cuadros o ventanas:

   - ✓ De clic en el Menú Ver y elija la opción Explorador de Proyectos
   - ✓ De clic en el Menú ver y elija la opción Ventana Propiedades

   Estas dos opciones deben de estar siempre activadas ya que de ahí depende todo lo que vallamos a hacer:



9. Del cuadro **Proyecto** haz doble clic en **Módulos** o simplemente presione el signo de + que aparece en la opción **Módulos**. Se activara debajo de **Módulos** la **Opción Modulo1**
10. Haz doble clic en **Modulo1**. Se mostrara en el Editor de Visual Basic el código de la macro que grabamos de la siguiente forma:

```
Sub Macro1()

    ' Macro1 Macro
    ' Macro grabada el 08/04/2005 por ....
    ' Acceso directo: CTRL+r
      Range("A1").Select
      ActiveCell.FormulaR1C1 = "Paty"
      Range("A2").Select
End Sub
```

Que es lo que significa esto nos preguntaremos asombrados, a continuación se da una explicación de lo que ha hecho Excel:

- ✓ Sub y End Sub indican el inicio y el final del procedimiento de la Macro1
- ✓ Todo lo que aparece con un apóstrofe ´ indica que no se tomara en cuenta que es solo texto o comentarios y ese texto debe de aparecer en un color, ya sea el color verde.
- ✓ Range("A1").Select Indica que lo primero que hicimos al grabar la macro fue trasladarnos a la celda A1. La orden Range nos permite trasladarnos a una celda
- ✓ ActiveCell.FormulaR1C1 = "Paty" Esto indica que se escribirá en la celda en que se encuentra el valor de texto Paty. Todo lo que aparece entre comillas siempre será un valor de texto. La orden ActiveCell.FormulaR1C1 nos permite escribir un valor en la celda activa.
- ✓ Range("A2").
- ✓ Select Otra vez indicamos que se traslade a la celda A2. Esto se debe a que cuando escribimos el nombre de Paty en A1 presionamos Enter y al dar Enter bajo a la celda A2. Para comprender alteraremos el código dentro del editor de Visual Basic.

```
Sub Macro1()
' Macro1 Macro
' Macro grabada el 01/01/2009 por ....
' Acceso directo: CTRL+r
Range("A1").Select
ActiveCell.FormulaR1C1 = "Paty"
Range("B1").Select
ActiveCell.FormulaR1C1 = "Almagro 1822"
Range("C1").Select
ActiveCell.FormulaR1C1 = "092794313"
Range("D1").Select
ActiveCell.FormulaR1C1 = "Alpallana"
Range("E1").Select
ActiveCell.FormulaR1C1 = "SACCEC"

End Sub
```

Recién se alteró el código y cuando regrese a Excel y ejecute la macro con Control + r hará lo siguiente:

En A1 escribirá Paty

En B1 escribirá Almagro 1822

En C1 escribirá 092794313

En D1 escribirá Alpallana

En E1 escribirá SACCEC

Así que salgamos del editor dando clic en el **Menú Archivo** y eligiendo la opción **Cerrar** y volver a Microsoft Excel. Si no deseas salir por completo da clic en el botón Microsoft Excel que se encuentra activado en la barra de tareas y cuando quieras volver al editor da clic en el botón Microsoft Visual Basic que se encuentra en la barra de Tareas.

Ahora ya que salimos de Visual Basic y estamos en Excel de Nuevo ejecutemos la macro presionando **Control + r** y veamos los resultados de nuestra modificación.

REALIZA LO SIGUIENTE...

1. Genera una Macro que escriba un nombre en una celda y lo ponga negrita y observa el Código.
2. Genera una Macro que escriba un nombre en una celda y lo Centre y observa el Código.
3. Genera una Macro que escriba un nombre en una celda y cambie el tamaño de la letra a 20 puntos y observa el Código.

A continuación vamos a repetir el programa Ejemplo1, pero en lugar de poner "Hola" en la celda A1 de la hoja activa, dejaremos que el usuario entre un texto desde teclado y a continuación guardaremos ese valor en esa celda. Observa que el valor que entre del usuario debe guardarse en algún lugar para poder ponerlo después en la celda A1; pues bien, ese valor se guardará en una variable. Una variable es simplemente un trozo de memoria que la función o procedimineto se reserva para guardar datos, la forma general de declarar una variable es:

DIM variable **AS** ***tipo***.

Siendo variable el nombre que se asigna a la misma y Tipo el tipo de datos que se guardarán (números, texto, fecha, boleanos,...). En nuestro ejemplo, declararemos la variable de tipo String (tipo texto), y lo haremos de la siguiente forma:

Dim Texto **As** ***String***

Con esto estamos indicando que se reserve un trozo de memoria, que se llama Texto y que el tipo de datos que se guardarán ahí serán caracteres.

La Función *InputBox*.

Esta función muestra una ventana para que el usuario pueda teclear datos. Cuando se pulsa sobre **Aceptar**, los datos entrados pasan a la variable a la que se ha igualado la función. Atento a la línea siguiente:

```
Texto = InputBox("Introduzca el texto", "Entrada de datos").
```

Sintaxis de InputBox.

InputBox(*Mensaje, Título, Valor por defecto, Posición horizontal, Posición Vertical, Archivo ayuda, Número de contexto para la ayuda*).

- *Mensaje: es el mensaje que se muestra en la ventana. Si desea poner más de una línea ponga Chr(13) para cada nueva línea, vea el ejemplo siguiente.*
- *Título: es el título para la ventana InputBox. Es un parámetro opcional.*
- *Valor por defecto: es el valor que mostrará por defecto el cuadro donde el usuario entra el valor.*
- *Parámetro opcional.*
- *Posición Horizontal: la posición X de la pantalla donde se mostrará el cuadro, concretamente es la posición para la parte izquierda. Si se omite el cuadro se presenta horizontalmente centrado a la pantalla.*
- *Posición Vertical: la posición Y de la pantalla donde se mostrará el cuadro, concretamente es la posición para la parte superior. Si se omite el cuadro se presenta verticalmente centrado a la pantalla.*
- *Archivo Ayuda: es el archivo que contiene la ayuda para el cuadro. Parámetro opcional.*
- *Número de contexto para la ayuda: número asignado que corresponde al identificador del archivo de ayuda, sirve para localizar el texto que se debe mostrar.*

Si se especifica este parámetro, debe especificarse obligatoriamente el parámetro Archivo Ayuda.

Por ejemplo:

```
Sub Entrar_Valor

    Dim Texto As String

    ' Chr(13) sirve para que el mensaje se muestre en dos Líneas

    Texto = InputBox("Introducir un texto " & Chr(13) & "Para la celda A1",
    "Entrada de datos")

    ActiveSheet.Range("A1").Value = Texto

End Sub
```

Este ejemplo también se puede hacer sin variables.

```
Sub Entrar_Valor

    ActiveSheet.Range("A1").Value = InputBox("Introducir un texto " & Chr(13) &
    "Para la celda

    A1", "Entrada de datos")

End Sub
```

Ejemplo:

Repetiremos el ejemplo anterior, pero en lugar de entrar los valores sobre la celda A1, haremos que el usuario pueda elegir en que celda quiere entrar los datos, es decir, se le preguntará al usuario mediante un segundo Inputbox sobre que celda quiere entrar el valor del primer Inputbox. Serán necesarias dos variables, una para guardar la celda que escoja el usuario y otra para guardar el valor.

```
Option Explicit

Sub Entrar_Valor

    Dim Celda As String
    Dim Texto As String

    Celda = InputBox("En que celda quiere entrar el valor", "Entrar Celda")

    Texto = InputBox("Introducir un texto " & Chr(13) & "Para la celda " & Celda ,

    "Entrada de

    datos")

    ActiveSheet.Range(Celda).Value = Texto

End Sub
```

En Visual Basic no es necesario declarar las variables, por ejemplo, en el programa anterior se hubiera podido prescindir de las líneas

La sentencia Option Explicit.

En visual basic no es necesario declarar las variables, por ejemplo, en el programa anterior se hubiera podido prescindir de las líneas:

```
Dim Celda As String

Dim Texto As String
```

A pesar de ello, le recomendamos que siempre declare las variables que va a utilizar, de esta forma sabrá cuales utiliza el procedimiento y que tipo de datos guarda cada una, piense que a medida que vaya aprendiendo, creará procedimientos cada vez más complicados y que requerirán el uso de más variables, si no declara las variables al principio del procedimiento ocurrirán dos cosas. Primero, las variables no declaradas son asumidas como tipo Variant (este es un tipo de datos que puede almacenar cualquier valor, número, fechas, texto, etc. pero tenga en cuenta que ocupa 20 Bytes y para guardar una referencia a una celda, la edad de alguien, etc. no son necesarios tantos bytes); segundo, reducirá considerablemente la legibilidad de sus procedimientos ya que las variables las irá colocando a medida que las necesite, esto, a la larga complicará la corrección o modificación del procedimiento.

La sentencia Option Explicit al principio del módulo fuerza a que se declaren todas las variables. Si al ejecutar el programa, se encuentra alguna variable sin declarar se producirá un error y no se podrá ejecutar el programa hasta que se declare. Si todavía no se ha convencido sobre la conveniencia de declarar las variables y utilizar Option Explicit, pruebe el procedimiento siguiente, cópielo tal cual (Texto y Testo están puestos adrede simulando que nos hemos equivocado al teclear).

```
Sub Entrar_Valor

    Texto = InputBox("Introducir un texto " & Chr(13) & "Para la celda A1",

    "Entrada de datos")

    ActiveSheet.Range("A1").Value = Testo
```

```
End Sub
```

Observa que el programa no hace lo que se pretendíamos que hiciera. Efectivamente, Texto y Testo son dos variables diferentes, como no se ha declarado ninguna ni se ha utilizado **Option Explicit** Visual Basic no da ningún tipo de error y ejecuta el programa. Prueba el siguiente módulo e intente ejecutarlo.

```
Option Explicit

Sub Entrar_Valor

    Dim Texto As String

    Texto = InputBox("Introducir un texto " & Chr(13) & "Para la celda A1",
    "Entrada de datos")

    ActiveSheet.Range("A1").Value = Testo

End Sub
```

Observa que el programa no se ejecuta, al poner **Option Explicit**, forzamos a que se declaren todas las variables. Visual Basic detecta que la variable Testo no ha sido declarada y así lo indica mostrando Error, entonces es cuando es más fácil darnos cuenta del error que hemos cometido al teclear y cambiamos Testo por Texto. Ahora imagina que el error se produce en un programa de cientos de líneas que necesita otras tantas variables. Tipos de datos en Visual Basic para Excel:

Tipos de datos	Tamaño de almacenamiento	Intervalo
Byte	1 byte	0 a 255
Boolean	2 bytes	True o False
Integer	2 bytes	-32.768 a 32.767
Long(entero largo)	4 bytes	-2.147.483.648 a 2.147.483.647
Single (coma flotante/ precisión simple)	4 bytes	-3,402823E38 a -1,401298E-45 para valores negativos; 1,401298E-45 a 3,402823E38 para valores positivos
Double (coma flotante/ precisión doble)	8 bytes	-1,79769313486232E308 a -4,94065645841247E-324 para valores negativos; 4,94065645841247E-324 a 1,79769313486232E308 para valores positivos
Currency (entero a escala)	8 bytes	-922.337.203.685.477,5808 a 922.337.203.685.477,5807

Decimal	14 bytes	+/- 79.228.162.514.264.337.593.543.950.335 sin punto decimal; +/- 7,9228162514264337593543950335 con 28 posiciones a la derecha del signo decimal; el número más pequeño distinto de cero es+/- 0,0000000000000000000000000001
Date	8 bytes	1 de enero de 100 a 31 de diciembre de 9999
Object	4 bytes	Cualquier referencia a tipo Object
String (longitud variable)	10 bytes + longitud de la cadena	Desde 0 a 2.000 millones
String(longitud fija)	Longitud de la cadena	Desde 1 a 65.400 aproximadamente
Variant(con números)	16 bytes	Cualquier valor numérico hasta el intervalo de un tipo Double
Variant(con caracteres)	22 bytes + longitud de cadena	El mismo intervalo que para un tipo String de longitud variable
Definido por el usuario (Type)	Número requerido por los elementos	El intervalo de cada elemento es el mismo que el Intervalo de su tipo de datos.

CONVERSIÓN DE TIPOS DE DATOS

Copia el siguiente Ejemplo. Simplemente se piden dos números, se suman y se guardan en la celda A1 de la hoja activa.

```
Option Explicit

Sub Sumar()
    Dim Numero1 As Integer
    Dim Numero2 As Integer
    Numero1 = InputBox("Entrar el primer valor", "Entrada de datos")
    Numero2 = InputBox("Entrar el primer valor", "Entrada de datos")
    ActiveSheet.Range("A1").Value = Numero1 + Numero2
End Sub
```

Ejecuta el procedimiento y pon respectivamente los valores 25 y 25. Observa que todo ha ido correctamente y en la celda A1 de la hoja activa aparece un 50. Ahora, vuelve a ejecutar el programa y cuando se te pida el primer valor teclee "Hola". Observa que el programa se detiene indicando un error en el tipo de datos. Efectivamente, observa que la función InputBox devuelve siempre datos tipo String, en el primer ejemplo no ha habido ningún problema, al entrar caracteres numéricos, estos pueden asignarse a variables tipo Integer porque Visual Basic hace automáticamente la conversión, pero al entrar texto e intentarlo asignar a una variable Integer Visual Basic muestra un error indicando que la variable no es adecuada para los datos que se desean guardar

Para solucionar estos problemas se deben utilizar funciones de conversión de tipo. Estas funciones, como su nombre indica, convierten datos de un tipo a otro, de String a Integer, de Integer a String, de Date a String,... Así el procedimiento anterior quedaría.

```
Option Explicit

Sub Sumar()
    Dim Numero1 As Integer
    Dim Numero2 As Integer
    Numero1 = Val(InputBox("Entrar el primer valor", "Entrada de datos"))
    Numero2 = Val(InputBox("Entrar el primer valor", "Entrada de datos"))
    ActiveSheet.Range("A1").Value = Numero1 + Numero2
End Sub
```

La función ***Val****(Dato String),* convierte una cadena de caracteres a valor numérico. Si la cadena a convertir contiene algún carácter no numérico devuelve 0. Así, si al pedir un valor se teclea "Hola", la función Val, devolverá un cero.

INFORMACIÓN (INCLUÍDA EN ESTE DOCUMENTO EDUCATIVO) TOMADA DE:

Sitios web:

1. https://exceltotal.com/el-editor-de-visual-basic/
2. https://es.justexw.com/tutoriales/tutorial-de-visual-basic-para-excel
3. https://es.justexw.com/tutoriales/como-usar-el-editor-de-visual-basic-en-excel
4. https://es.justexw.com/tutoriales/como-insertar-macros-en-excel-con-visual-basic
5. https://es.justexw.com/tutoriales/como-insertar-un-modulo-en-visual-basic-para-excel
6. https://es.justexw.com/tutoriales/como-usar-las-funciones-de-excel-con-visual-basic
7. http://www3.uji.es/~berbel/Visual%20Basic/Manuales/Excelvbaplication%202010.pdf