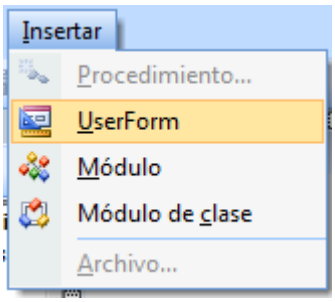
CREANDO FORMULARIOS Y PROGRAMÁNDOLOS

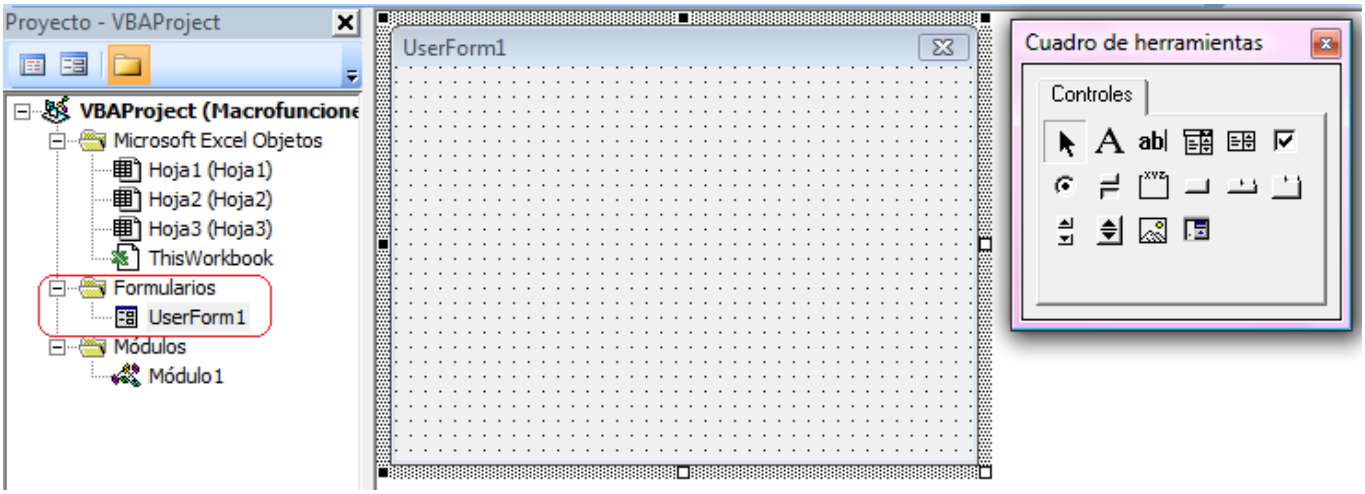
Ahora aprenderemos a dominar lo máximo de Excel que es crear formularios y programarlos, bueno un formulario es una ventana que se programa por medio de controles y estos controles responden a sucesos que nosotros programamos.

Todo esto se encuentra dentro del Editor de Visual Basic. A continuación se muestra como crear un formulario y como programarlo:

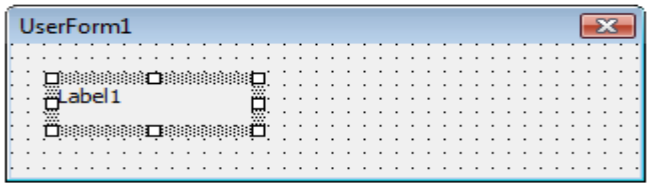
Del **Menú Insertar** elija la Opción **UserForm**, observe:



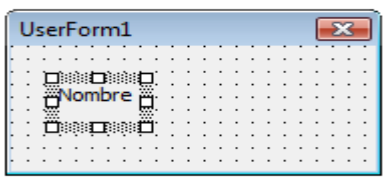
A continuación nos aparece en el explorador la carpeta de formularios y el UserForm que se muestra a la derecha junto con el cuadro de herramientas, observe:



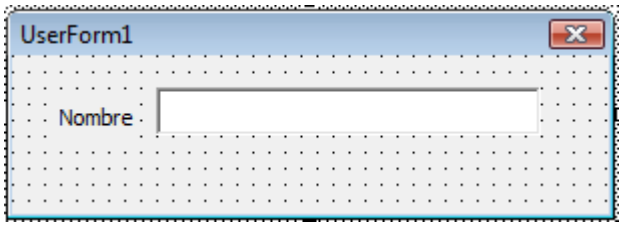
Elija del **Cuadro de Herramientas** el **Control Etiqueta** el que tiene la **A** y Arrastre dibujando en el Formulario **USERFORM1** la etiqueta. Quedara el nombre Label1, observe:



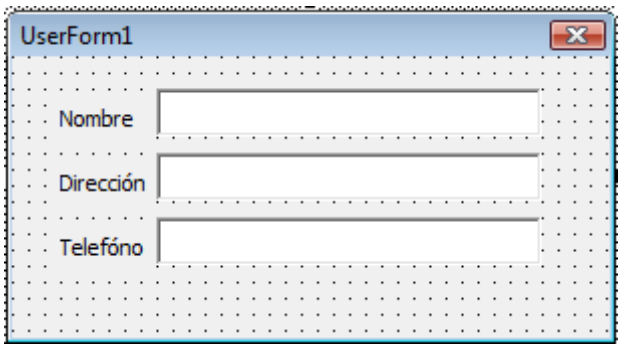
Luego de un clic en la etiqueta dibujada y podrá modificar el nombre de adentro y lo cambiaremos por **Nombre**. Si por error da doble clic en la etiqueta y lo manda a la pantalla de programación de la etiqueta, solo de doble clic en USERFORM1 que se encuentra en el Explorador de Proyecto.



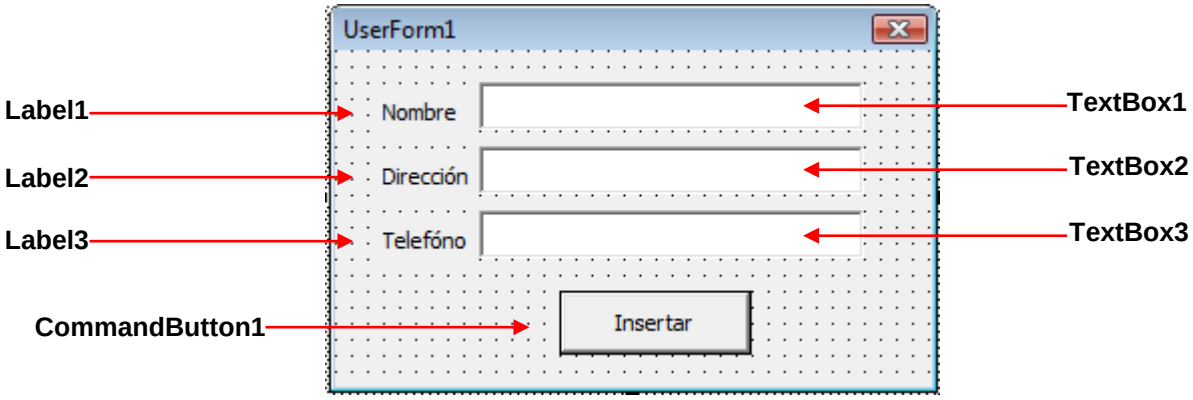
Elija del **Cuadro de Herramientas** el control **Cuadro de Texto** el que tiene **ab|** y arrastre dibujando en el formulario USERFORM1 el cuadro de texto a un lado de la etiqueta que dice Nombre. El cuadro de texto debe de estar vacío y su nombre será Textbox1, el nombre solo aparecerá en el control, observe:



Haga los dos pasos anteriores agregando dos controles de etiqueta y dos cuadros de texto. Esto quedara así después de haberlo hecho:



Por último elija del **Cuadro de Herramientas** el control **Botón de Comando** el que es y arrastre dibujando en el formulario USERFORM1 el botón debe quedar ubicado en la siguiente posición con el nombre Insertar:

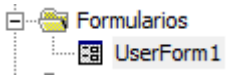


Ahora de doble clic sobre el control **Textbox1** para programarlo y después inserte el siguiente código:

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

Esto indica que se valla a la celda **A9** y escriba lo que hay en el **Textbox1**, es decir cuando ejecute el formulario en Excel en la celda A9 se escribirá el nombre que se digite en el formulario.

Nota: Lo que esta en azul lo genera Excel automáticamente, usted solo escribirá lo que esta en Negrita. Para volver al Formulario y programar el siguiente Textbox de doble clic en **Userform1** que se encuentra en el Explorador de la izquierda:



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 la celda **B9** y escriba lo que hay en el **Textbox2**.

Volver al **UserForm1** doble clic sobre el control **Textbox3** para programarlo y después inserte el siguiente código:

```
Private Sub TextBox3_Change()
Range("C9").Select
ActiveCell.FormulaR1C1 = TextBox3
End Sub
```

Esto indica que se valla a la celda **C9** y escriba lo que hay en el **Textbox3**.

Para crear un comentario en visual Basic debes hacerlo utilizando comilla simple **‘**

Volver al **UserForm1** doble clic sobre el control **CommandButton1** 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 o ‘** son empleados para poner comentarios dentro de la programación, el comando **Empty** es empleado para vaciar los Textbox, es decir dejarlos en blanco para escribir un nuevo dato.

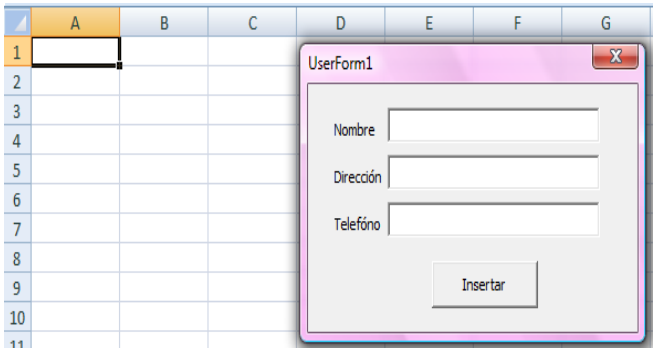
Guarde los cambios para que visual Basic guarde el formulario creado y configurado.

Como Ejecutar el Formulario

Hay 4 formas:

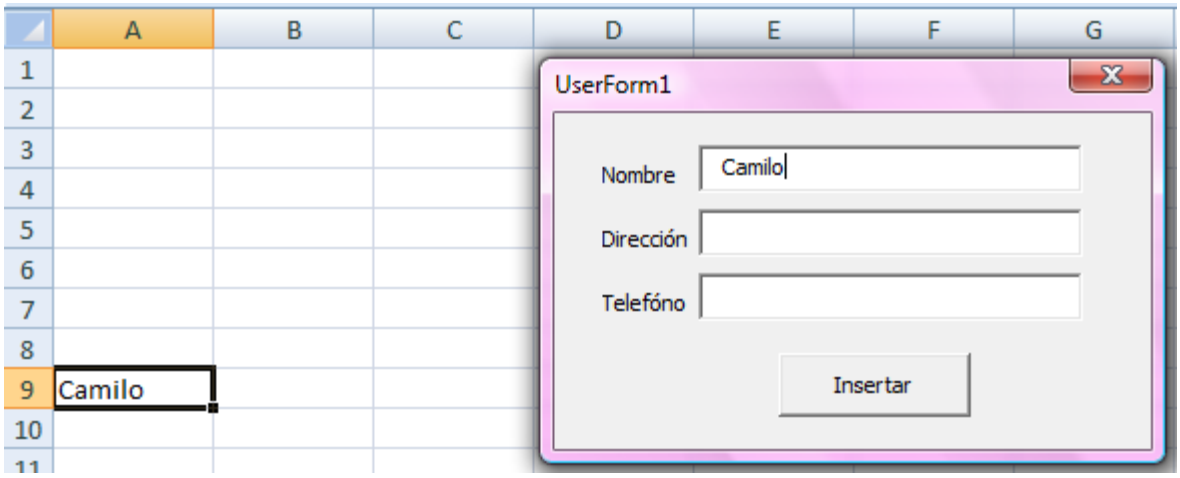
1. Una es pulsando la tecla **F5.**
2. Otra es dar clic en el botón **Ejecutar** .
3. Otra es seleccionar el **Menú Ejecutar** y dar clic en **Ejecutar Sub/UserForm.**
4. La última es ejecutarlo utilizando un botón de formulario en la Hoja de Excel.

Pulsemos la tecla **F5**, y observe que el formulario aparece activado en la hoja de Excel, observe:

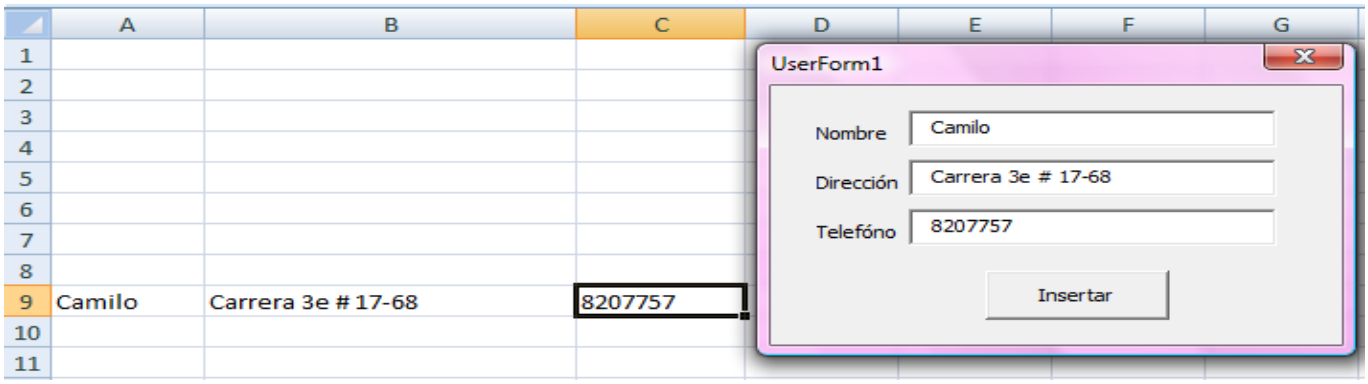


Se activará el **Userform1** y todo lo que escriba en los Textbox se escribirá en Excel y cuando presione el botón Insertar, se insertara un renglón y se vaciaran los Textbox y después se mostrara el cursor en el **Textbox1**.

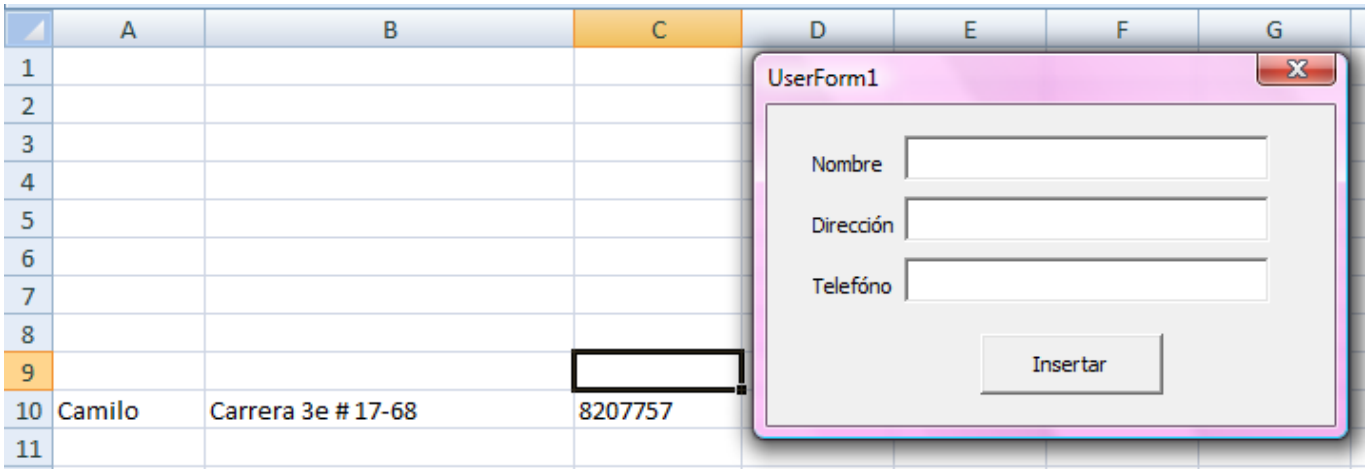
Veamos insertaremos el siguiente registro en el formulario, observe que automáticamente cuando se escribe el nombre en el formulario este se escribe inmediatamente en la celda A9, mire:



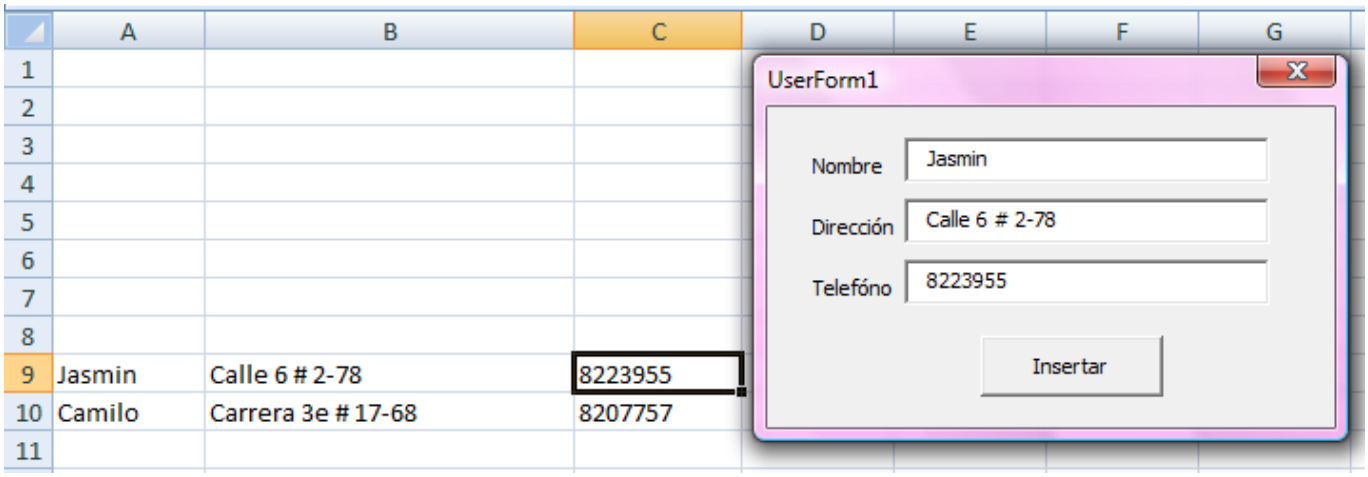
Continuemos con el registro observe como debe quedar:



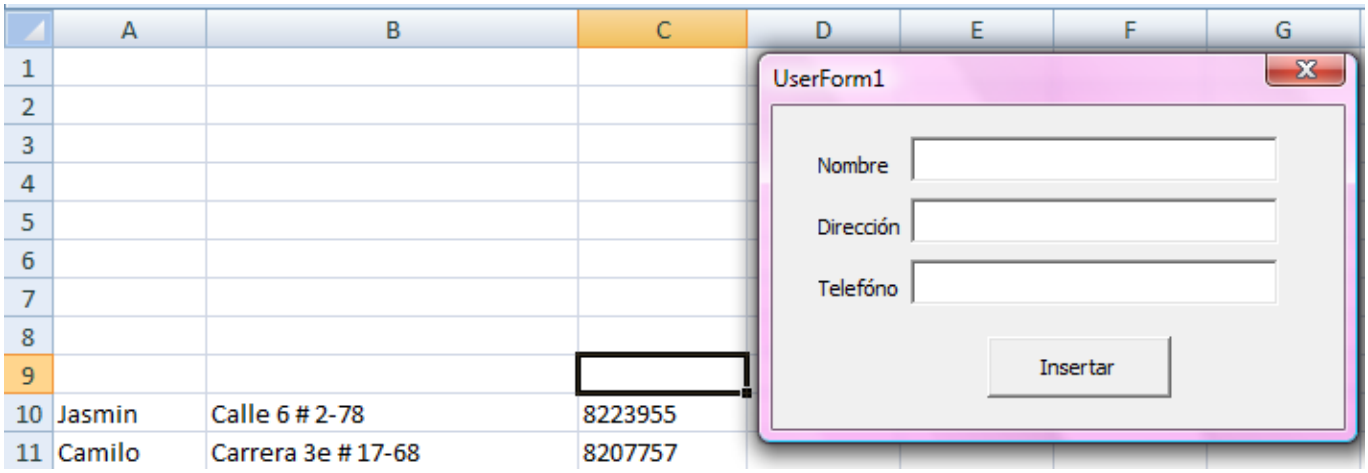
Damos clic en el botón **Insertar**, observe que los cuadros de texto del formulario quedan en blanco por si se quiere agregar otro registro y el registro que se digito queda guardado en la celda A10 de la hoja de Excel, observe:



Agreguemos un registro más:



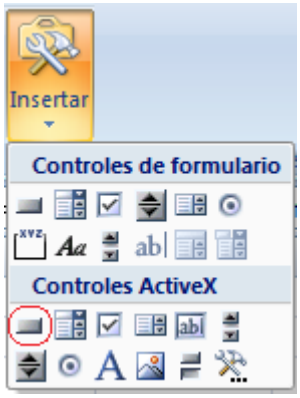
Damos clic en Insertar y observe como queda la lista ahora:



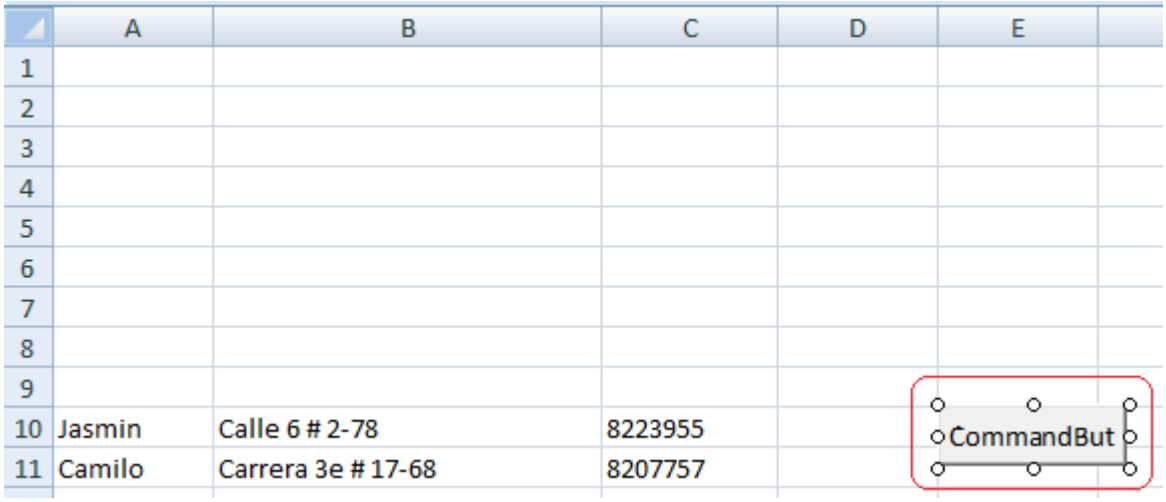
Cada vez que se agrega un nuevo registro este automáticamente inserta una fila por debajo de la celda que se escribe y los anteriores registros insertados bajan una posición.

Asignar el Formulario a un Botón desde Excel

Para cargar el formulario desde la hoja de Excel, en la vista Programador, en la opción Insertar, en Controles ActiveX seleccionamos el botón de comando, observe:



Damos clic en la hoja para ubicarlo donde queremos, observe:



Para programarlo damos doble clic en el botón y nos aparece el editor de visual Basic, observe:

```
Private Sub CommandButton1_Click()

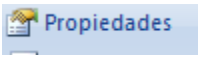
End Sub
```

En medio digitamos las siguientes líneas de código, observe:

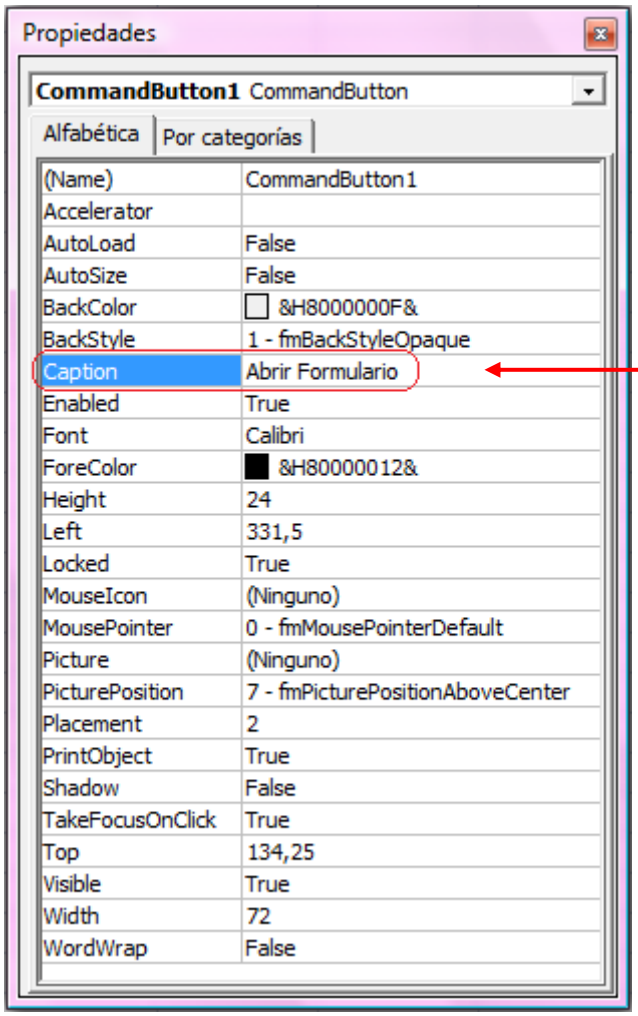
```
Private Sub CommandButton1_Click()
Load UserForm1
UserForm1.Show
End Sub
```

Load UserForm1 Carga el formulario
UserForm1.Show Hace visible el formulario lo muestra

Guardamos los cambios en el editor y regresamos a Excel.

Para cambiarle el nombre al botón, lo seleccionamos y damos clic en la opción 

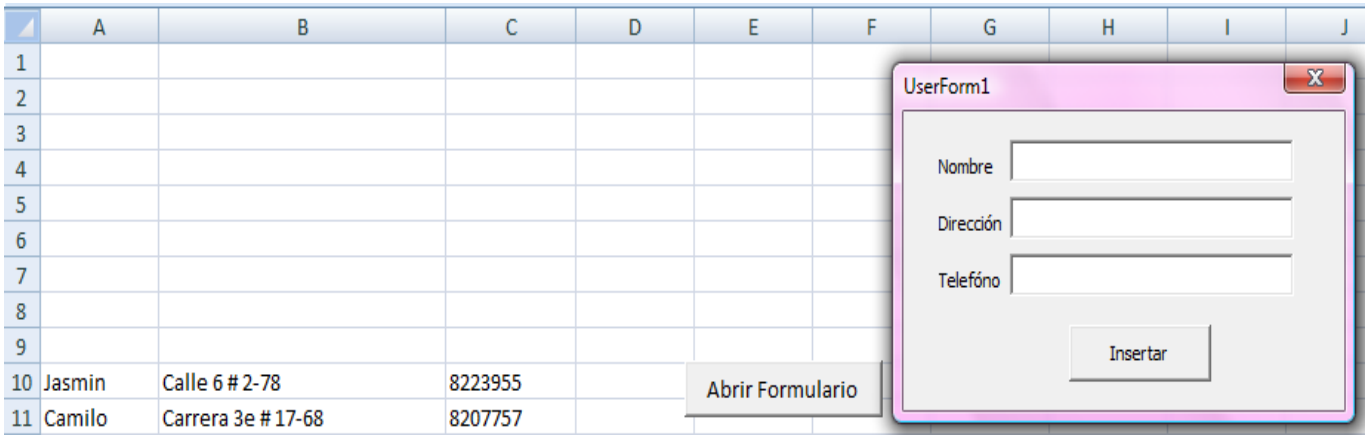
y nos aparece la siguiente ventana:



Nos Ubicamos al frente de la opción **Caption** y escribimos el nombre que deseamos ponerle al botón para nuestro caso será **Abrir Formulario** y cerramos la ventana.

Para activar el botón damos clic en la opción 

Damos clic en el botón Abrir Formulario y observe que ya nos muestra el formulario:



Otros Controles de Formulario

En Este nuevo formulario vamos a hacer uso de otros controles de formulario como los son: Cuadros combinados, Cuadro de lista, Casillas de Verificación, Botones de opción.

Primero iniciemos creando los controles que vimos en el ejemplo anterior, observe como ordenarlos:



Son 4 etiquetas y 3 Cuadros de texto

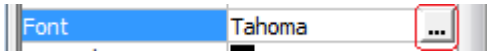
Ahora vamos a insertar un Marco , observe como debe quedar:



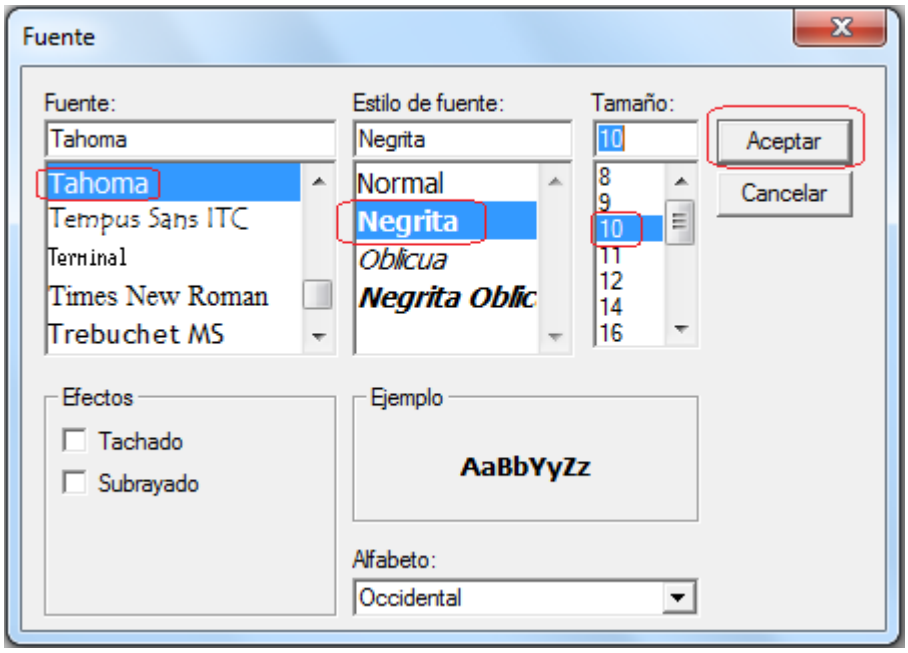
Para cambiarle el nombre al marco, lo seleccionamos y a la izquierda en la Propiedades, buscamos la opción **Caption** y al lado digitamos el texto que queremos aparezca, observe:



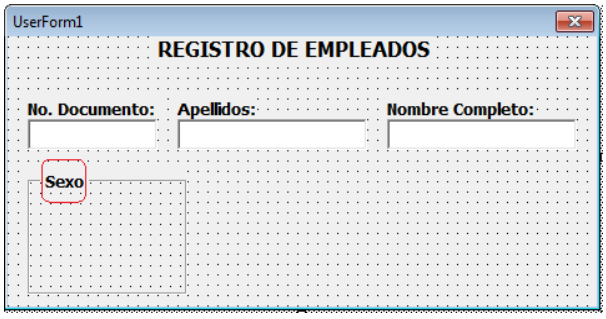
Para ponerlo en negrita y tamaño 10 hacemos lo siguiente, en las propiedades buscamos la opción **Font *Tahoma*** y damos clic en los puntos que aparecen, observe:



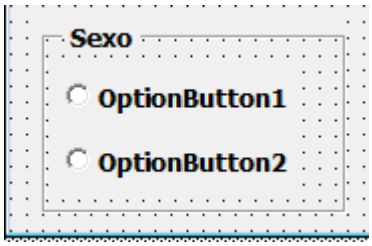
En la ventana que se muestra a continuación, podemos cambiar el tipo de Fuente, el estilo y el Tamaño, observe:



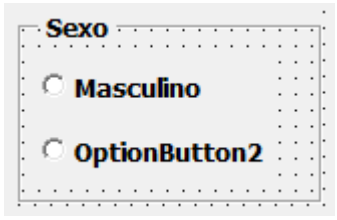
Escogemos fuente Tahoma, Estilo Negrita y Tamaño 10, damos clic en Aceptar y observe como queda:



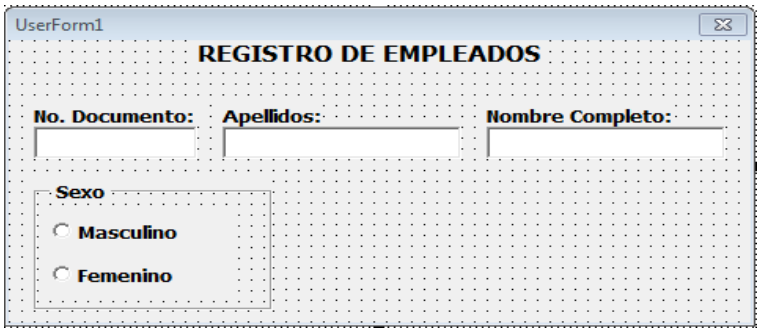
Ahora dentro del marco vamos a agregar dos Botones de opción , observe:



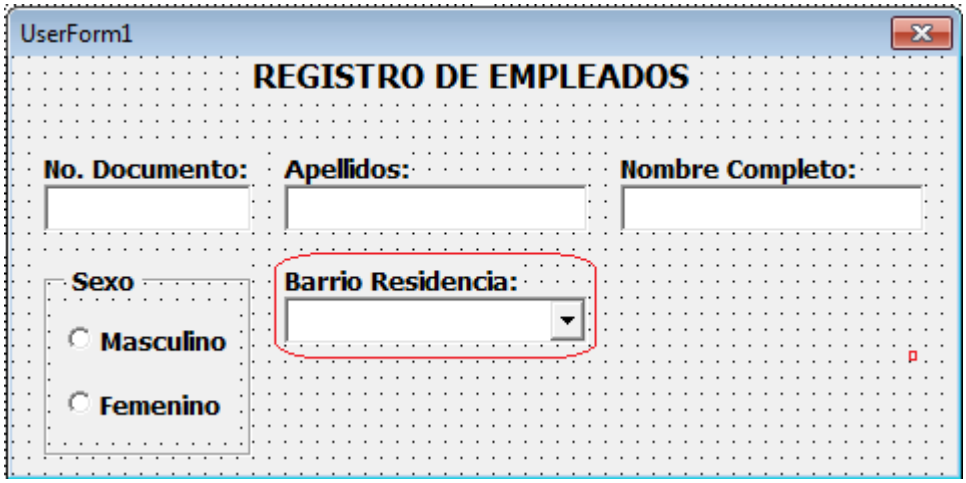
Para cambiarles el nombre, seleccionamos el botón, en las propiedades nos ubicamos en Caption y al lado digitamos el texto que queramos aparezca, observe:



De igual forma hacemos con el otro botón y lo configuramos así:



Ahora agregaremos un cuadro combinado , ordenándolo de la siguiente forma ,con su etiqueta, observe:



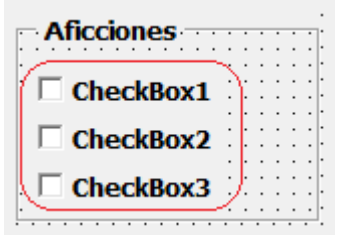
Ahora agregaremos un cuadro de lista , ordenándolo de la siguiente forma ,con su etiqueta, observe:



Insertamos un nuevo marco y lo configuramos de esta forma:



Luego le agregamos al marco 3 casillas de verificación, de esta forma:



Luego les ponemos los siguientes nombres, observe:



Luego seleccionamos el formulario y en la propiedad Caption le digitamos el nombre Empleados al formulario, Observe:



Por último agregamos dos botones de comando y los configuramos de esta forma:



Damos clic en el botón Ejecutar  para ver el formulario en funcionamiento, recuerde que no está programado, observe:



Cerramos el formulario para volver al editor de Visual Basic nuevamente y programar los controles:

Empezamos programando los textbox, similar en el ejemplo1, para ello damos doble clic en el textbox 1, observe:

```
Private Sub TextBox1_Change()

End Sub
```

Le digitamos el siguiente código:

```
Private Sub TextBox1_Change()
Range("A2").Select
ActiveCell.FormulaR1C1 = TextBox1
End Sub
```

De igual forma para el textbox 2 y textbox 3, observe:

```
Private Sub TextBox2_Change()
Range("B2").Select
ActiveCell.FormulaR1C1 = TextBox2
End Sub
Private Sub TextBox3_Change()
Range("C2").Select
ActiveCell.FormulaR1C1 = TextBox3
End Sub
```

NOTA: Recuerden que lo que hace estas líneas de código es ir a la hoja activa de Excel, seleccionar la celda asignada y escribir lo que contienen los textbox.

Para los botones de opción debemos validarlo de acuerdo a la opción que escojamos, para ello vamos a programarlos utilizando la condición IF, observe:

```
Private Sub OptionButton1_Click()
If OptionButton1 = True Then
Range("D3").Select
ActiveCell.FormulaR1C1 = "Masculino"
End If
If OptionButton2 = True Then
Range("D3").Select
ActiveCell.FormulaR1C1 = "Femenino"
End If
End Sub
Private Sub OptionButton2_Click()
If OptionButton1 = True Then
Range("D3").Select
ActiveCell.FormulaR1C1 = "Masculino"
End If
If OptionButton2 = True Then
Range("D3").Select
ActiveCell.FormulaR1C1 = "Femenino"
End If
End Sub
```

Vamos a evaluar los botones de opción con dos If

```
If OptionButton1 = True Then
Range("D3").Select
ActiveCell.FormulaR1C1 = "Masculino"
End If
```

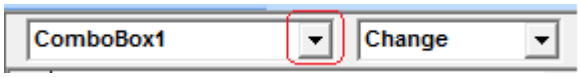
La primer línea indica si se selecciona el OptionButton1 que corresponde a la opción Masculino, sea igual a True es decir verdadero, este es el valor que toma un OptionButton al darle clic, si se cumple entonces que en la celda D3 escriba la palabra Masculino.

De igual forma funciona para el OptionButton2, que al seleccionarlo en la celda D3 pondría la palabra Femenino:

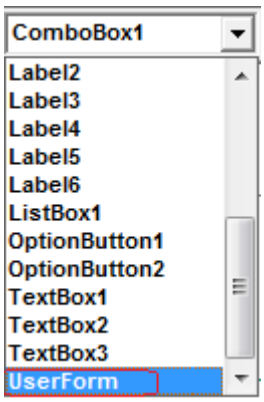
```
If OptionButton2 = True Then
Range("D3").Select
ActiveCell.FormulaR1C1 = "Femenino"
End If
```

Al ComboBox, primero hay que agregarle los valores, es decir las opciones que se pueden seleccionar al darle clic, para ello, debemos asignarlos en el momento en que el formulario sea activado, para ello se usa la propiedad actívate del userform1, veamos como activarla:

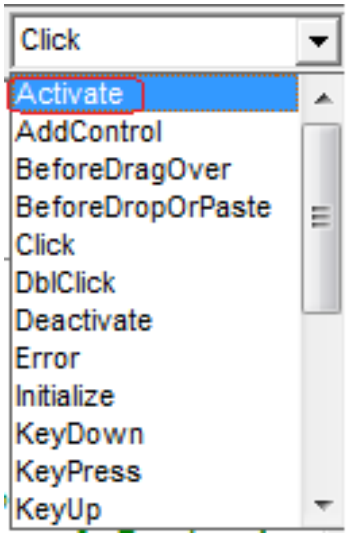
En la hoja donde se programan los controles den clic en este botón que esta al inicio



Luego seleccionar UserForm, observe:



Y en la opción de al lado escogen, Activate, observen:



Nos genera este código:

```
Private Sub UserForm_Activate()

End Sub
```

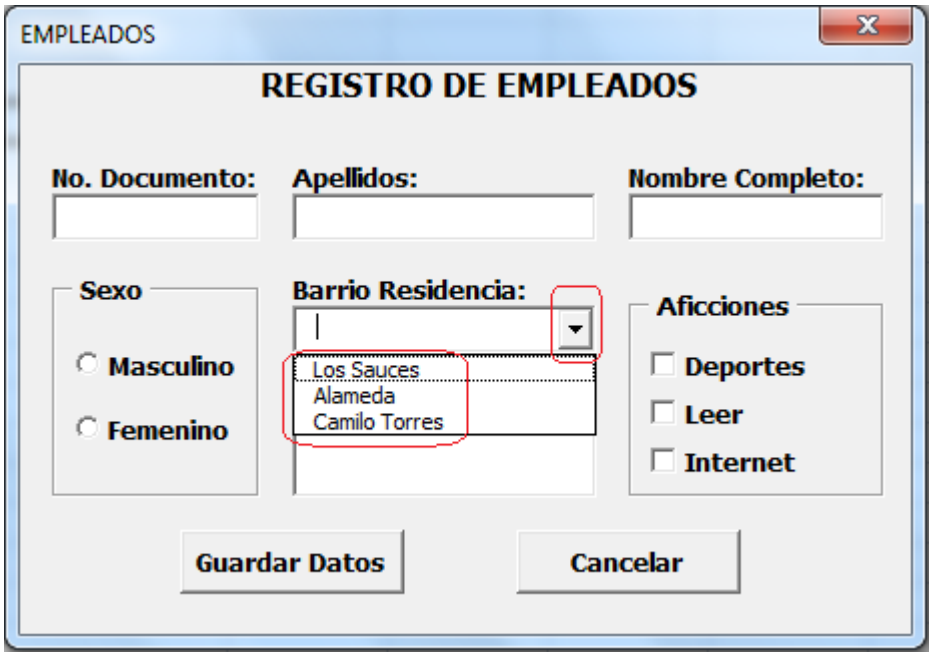
Ahora le escribimos el código que permite asignarle valores al ComboBox, observen como:

```
Private Sub UserForm_Activate()
ComboBox1.AddItem "Los Sauces"
ComboBox1.AddItem "Alameda"
ComboBox1.AddItem "Camilo Torres"
End Sub
```

ComboBox1.AddItem "Los Sauces"

Quiere decir que al cuadro combinado 1, agréguele la opción Los Sauces.

Si ejecutamos el formulario y hacemos clic en el cuadro combinado, podemos ver como aparecen las opciones, observen:



Con solo ejecutar el formulario los valores del cuadro combinado aparecen automáticamente, esto es lo que permite el código

```
Private Sub UserForm_Activate()
ComboBox1.AddItem "Los Sauces"
ComboBox1.AddItem "Alameda"
ComboBox1.AddItem "Camilo Torres"
End Sub
```

Ahora para que el ComboBox agregue en la Hoja el valor que se selecciona se programa de la siguiente, observe:

```
Private Sub ComboBox1_Change()
Range("E3").Select
ActiveCell.FormulaR1C1 = ComboBox1
End Sub
```

Para el ListBox (lista de valores) funciona similar al ComboBox, los valores que se desean mostrar también se deben agregar al activarse el formulario para ello también deben añadirse en el UserForm_Activate similar al ComboBox, observe:

```
Private Sub UserForm_Activate()
ComboBox1.AddItem "Los Sauces"
ComboBox1.AddItem "Alameda"
ComboBox1.AddItem "Camilo Torres"

ListBox1.AddItem "Estrato 1"
ListBox1.AddItem "Estrato 2"
ListBox1.AddItem "Estrato 3"
End Sub
```

Lo único que cambia es el nombre del control al cual se le van a asignar los valores.

Si ejecutamos el formulario podemos ver como aparecen las opciones del ListBox, en la opción Estrato, observen:



Ahora para que el ListBox agregue en la Hoja el valor que se selecciona se programa de la siguiente, observe:

```
Private Sub ListBox1_Click()
Range("F3").Select
ActiveCell.FormulaR1C1 = ListBox1
End Sub
```

Por último vamos a programar las Aficiones, en este caso los CheckBox o casillas de verificación, las cuales pueden seleccionarse varias opciones o al menos 1, para ello vamos hacer varias rutinas IF anidadas para verificar las opciones que se estén seleccionando, veamos:

Damos doble clic en el CheckBox 1 y nos aparece este código:

```
Private Sub CheckBox1_Click()

End Sub
```

Ahora digitaremos el código, veamos:

Primero definimos una variable llamada cadena de tipo String es decir que en esa variable vamos a guardar texto, la palabra Dim indica que vamos a definir una variable.

Luego a la variable cadena la inicializamos vacía es decir que no contenga nada, Observe como queda el código:

```
Private Sub CheckBox1_Click()
Dim cadena As String
cadena = ""
End Sub
```

Ahora vamos a evaluar el CheckBox1, que pasa cuando se ejecute el formulario y se dé clic en él:

```
Private Sub CheckBox1_Click()
Dim cadena As String
cadena = ""

If CheckBox1 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "Deportes"
End If

End Sub
```

If CheckBox1 = True Then

Esta línea indica Si el CheckBox1 es verdadero, es decir cuando se dé clic en él, que hace

If Len(cadena) = 0 Then cadena = ""	Len() es una función para evaluar la longitud de un texto
End If	significa el final del segundo if que esta anidado.

Estas líneas indican si la longitud de la variable cadena es igual a cero o vacio que haga cadena igual a vacio, es decir que la deje en blanco, en este caso cuando el formulario se ejecute y no sé de clic en ninguna opción que deje la celda vacía.

cadena = cadena + "Deportes"	Significa que a la variable cadena se le asigne la misma variable pero agregándole la palabra deportes
End If	significa el final del primer if.

Para evaluar los CheckBox2 y CheckBox3 el código es el mismo, lo único que cambia es cuando se le asigna la palabra leer e Internet de los dos checkbox.

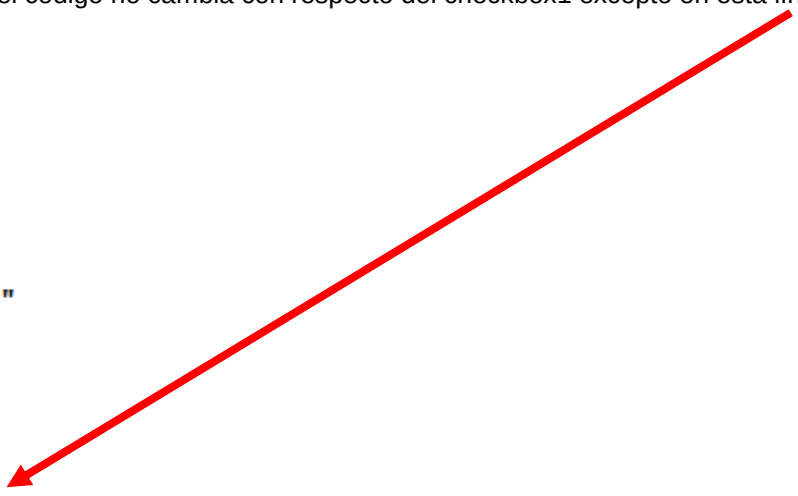
Veamos para el checkbox2, observen que el código no cambia con respecto del checkbox1 excepto en esta línea:

```
Private Sub CheckBox1_Click()
Dim cadena As String
cadena = ""

If CheckBox1 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "Deportes"
End If

If CheckBox2 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Leer"
End If

End Sub
```



En la línea indicada se le asigna a la variable cadena si ha seleccionado el checkbox1, en los if anteriores, también se le agrega la palabra Leer en el caso de que haya escogido las dos opciones, es decir si se ha dado clic tanto en el CheckBox1 y el CheckBox2, estos signos `+ "," +` quieren decir que una o concatene las dos palabras.

Para el checkBox3 es el mismo código salvo en la misma línea del CheckBox2, observe:

```
Private Sub CheckBox1_Click()
Dim cadena As String
cadena = ""

If CheckBox1 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "Deportes"
End If

If CheckBox2 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Leer"
End If

If CheckBox3 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Internet"
End If

End Sub
```

Por último evaluamos la longitud de la variable cadena nuevamente:

```
Private Sub CheckBox1_Click()
Dim cadena As String
cadena = ""

If CheckBox1 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "Deportes"
End If

If CheckBox2 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Leer"
End If

If CheckBox3 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Internet"
End If

If Len(cadena) <> 0 Then
Range("G3").Select
ActiveCell.FormulaR1C1 = cadena
End If
End Sub
```

Aquí evaluamos si la cadena no esta vacía, entonces que en la celda G3 se digite o se agregue el contenido de la variable cadena, que puede ser que se hayan seleccionado 1, 2 o las tres opciones.

Fin del checkbox1

Ahora lo que debemos hacer es copiar y pegar todo el código en los CkeckBox 2 y CkeckBox3. Observen:

```
Private Sub CheckBox2_Click()
Dim cadena As String
cadena = ""
If CheckBox1 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "Deportes"
End If

If CheckBox2 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Leer"
End If

If CheckBox3 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Internet"
End If
If Len(cadena) <> 0 Then
Range("G3").Select
ActiveCell.FormulaR1C1 = cadena
End If
End Sub
```

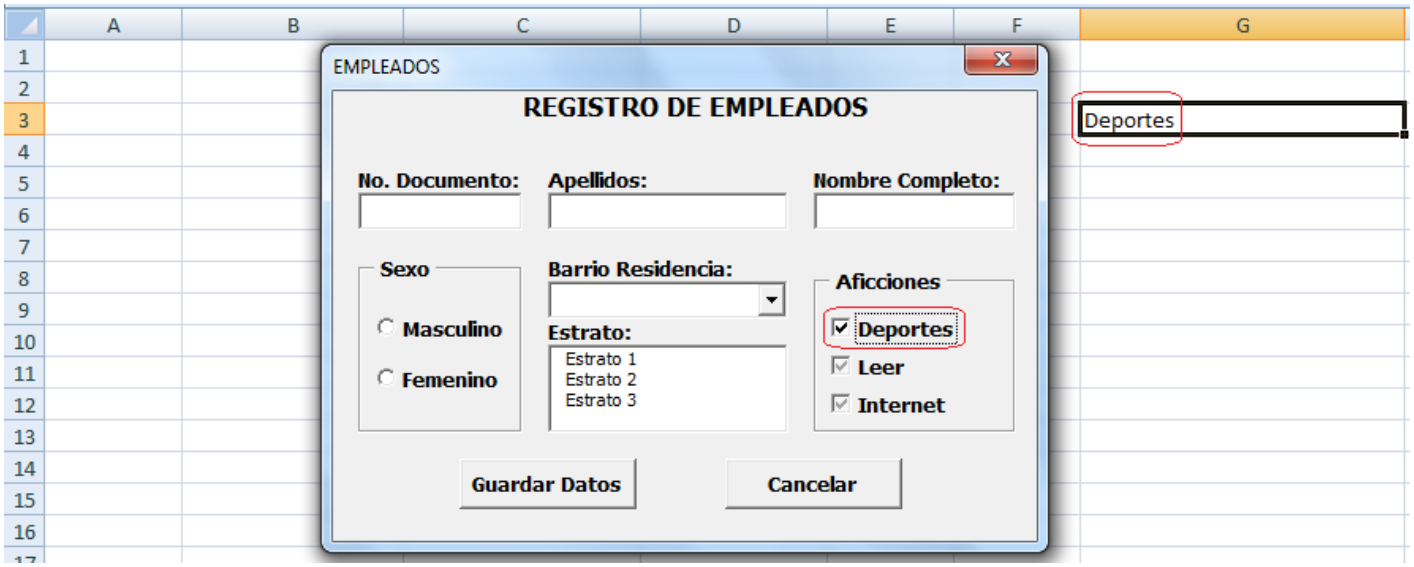
```
Private Sub CheckBox3_Click()
Dim cadena As String
cadena = ""
If CheckBox1 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "Deportes"
End If

If CheckBox2 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Leer"
End If

If CheckBox3 = True Then
    If Len(cadena) = 0 Then
        cadena = ""
    End If
    cadena = cadena + "," + "Internet"
End If
If Len(cadena) <> 0 Then
Range("G3").Select
ActiveCell.FormulaR1C1 = cadena
End If
End Sub
```

Listo ejecutemos el formulario y veamos como es el funcionamiento de los CheckBox:

Al dar clic en la opción Deportes que corresponde al CheckBox1, observen que automáticamente aparece la palabra en la celda G3:



Ahora si se da clic en la opción Leer que corresponde al CheckBox2, observen que automáticamente se agrega la palabra separada de una coma:



Ocurre lo mismo si se da clic en la opción Internet que corresponde al CheckBox3, observen:



Ahora en el caso de que se desea quitar una de las opciones solo habría que desactivar o dar nuevamente clic en la opción, por ejemplo desactivemos la opción Leer, observen que automáticamente la palabra se borra de la celda G3, pero se mantienen las otras dos opciones escogidas:



Cerremos el formulario y ahora programemos los dos botones, Guardar Datos y Cancelar:

Demos doble clic en el botón Guardar Datos, observe el código que aparece:

```
Private Sub CommandButton1_Click()

End Sub
```

Digitamos el siguiente código que ya saben que hace cuando se creó el ejemplo1 del primer formulario:

Selection.EntireRow.Insert lo que hace es insertar los datos escritos y crea una nueva fila para agregar otro nuevo
Empty lo que hace es limpiar o dejar en blanco los controles.
TextBox1.SetFocus lo que hace es luego de insertar los datos digitamos y poner nuevamente las celdas y controles en blanco, ubica el cursor en el primer textbox.

```
Private Sub CommandButton1_Click()
Selection.EntireRow.Insert

TextBox1 = Empty
TextBox2 = Empty
TextBox3 = Empty
OptionButton1 = Empty
OptionButton2 = Empty
ComboBox1 = Empty
ListBox1 = Empty
CheckBox1 = Empty
CheckBox2 = Empty
CheckBox3 = Empty

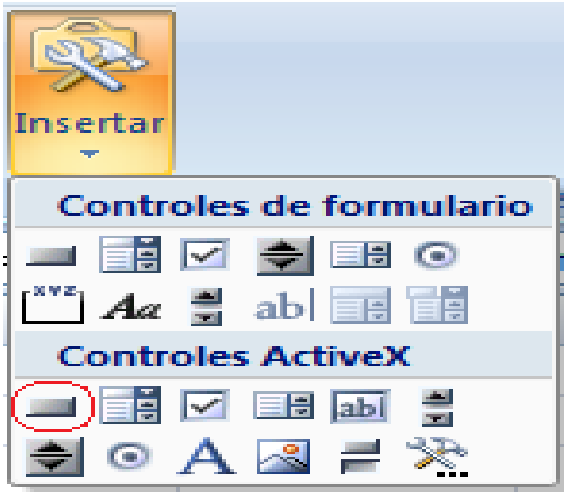
TextBox1.SetFocus
End Sub
```

Este es el código para el botón Cancelar el cual deja los controles en Blanco:

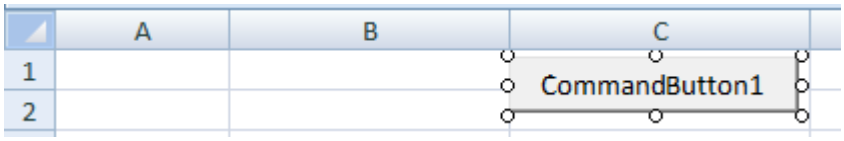
```
Private Sub CommandButton2_Click()
TextBox1 = ""
TextBox2 = ""
TextBox3 = ""
ComboBox1 = ""
Range("D3").Select
ActiveCell.FormulaR1C1 = ""
Range("F3").Select
ActiveCell.FormulaR1C1 = ""
Range("G3").Select
ActiveCell.FormulaR1C1 = ""
End Sub
```

Vayan a la Hoja de Excel y Borren todo, dejen la hoja en blanco.

Ahora desde la hoja de Excel creemos un botón para ejecutar el formulario, en la vista Programador, en la opción Insertar, en Controles ActiveX seleccionamos el botón de comando, observe:



Luego forme el botón y ubíquelo en una posición de la Hoja, Observe:



Para programarlo damos doble clic en el botón y nos aparece el editor de visual Basic, observe:

```
Private Sub CommandButton1_Click()

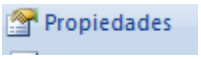
End Sub
```

En medio digitamos las siguientes líneas de código, observe:

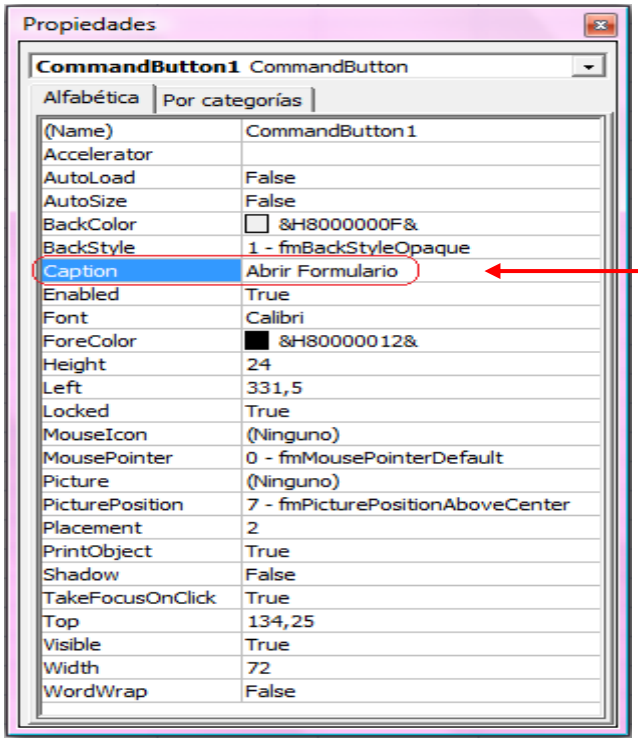
```
Private Sub CommandButton1_Click()
Load UserForm1
UserForm1.Show
End Sub
```

Load UserForm1 Carga el formulario
UserForm1.Show Hace visible el formulario lo muestra

Guardamos los cambios en el editor y regresamos a Excel.

Para cambiarle el nombre al botón, lo seleccionamos y damos clic en la opción 

y nos aparece la siguiente ventana:



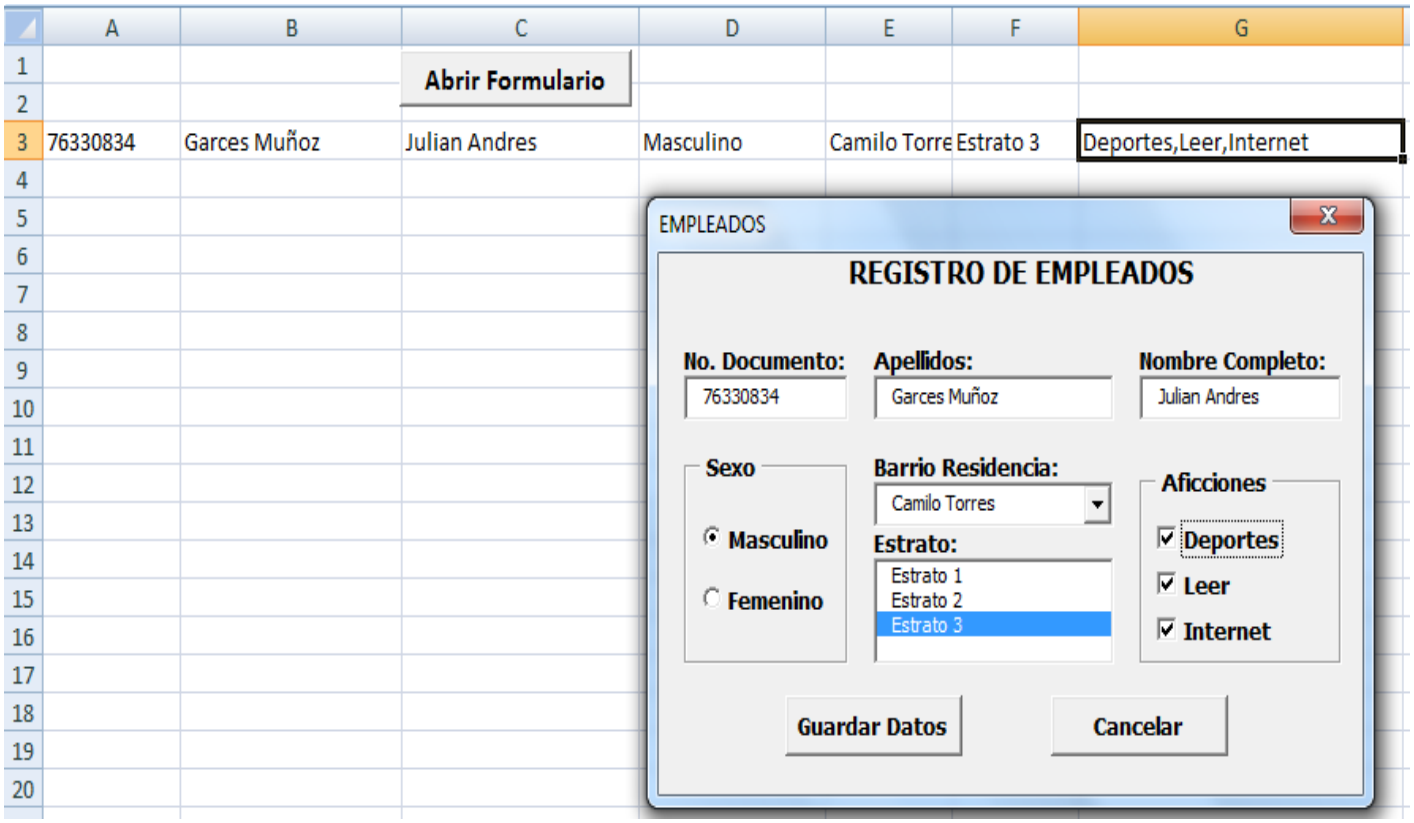
Nos Ubicamos al frente de la opción **Caption** y escribimos el nombre que deseamos ponerle al botón para nuestro caso será **Abrir Formulario** y cerramos la ventana.

Para activar el botón damos clic en la opción 

Damos clic en el botón Abrir Formulario y observe que ya nos muestra el formulario:

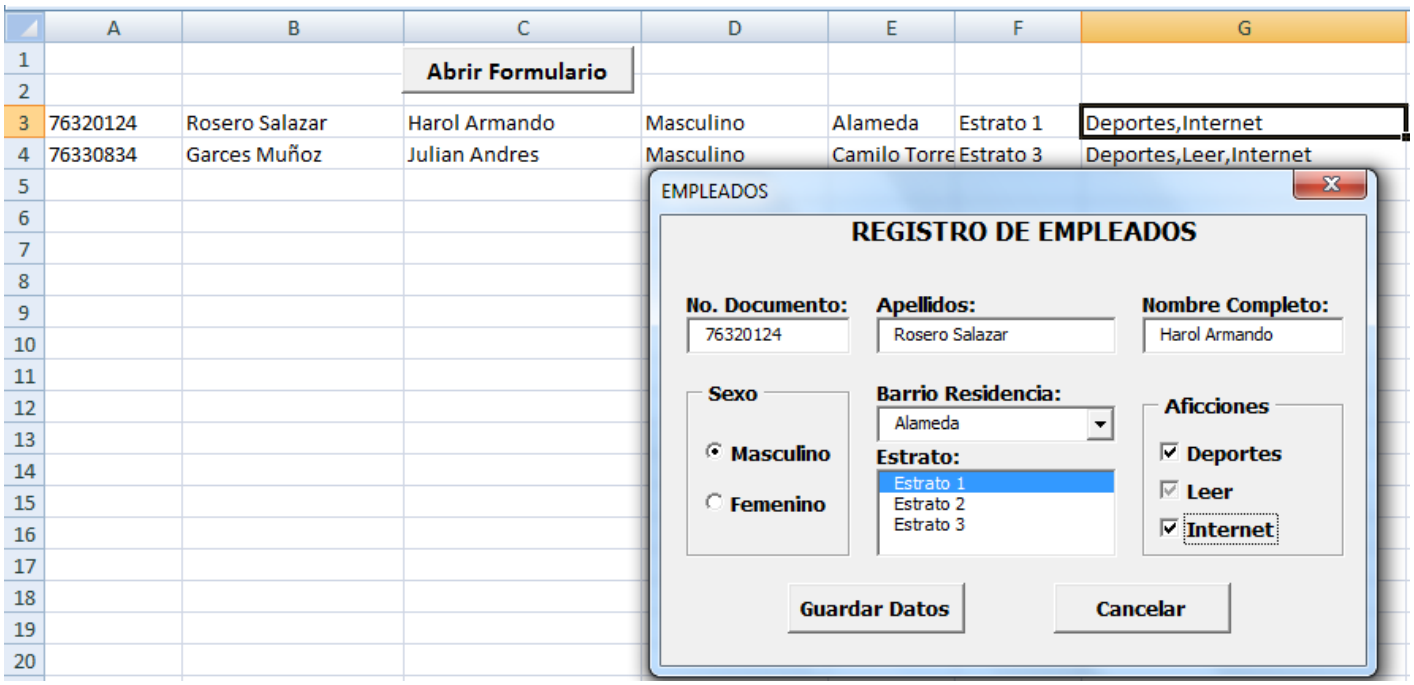


Ahora empecemos a insertar datos, observen que al ir digitando los valores automáticamente se van agregando a las celdas asignadas:



Damos clic en el Botón Guardar datos para agregar nuevos datos:

Miremos otro registro, observe que el anterior ha bajado una fila lo que indica que ya ha sido almacenado y los nuevos valores aparecen por encima, listos para guardar:



Bueno Aprendices con este Material damos por culminado los documentos del curso que espero les hayan servido de apoyo en sus labores cotidianas y laborales, los invito a realizar con entusiasmo el último Taller del curso.