

# Computación II

## CIA-3351

Unidad II  
Programación para objetos Excel

Carlos Valle Vidal  
Primer Semestre 2018

# Programación para objetos de Excel

- En este capítulo veremos como hacer programas que interactúen con formularios, tablas y gráficos.
- Para poder manipularlos, Visual Basic los trata como objetos.

# ¿Qué es un objeto?

- Un objeto es un “tipo de dato” especial que tiene asociados **atributos** y **métodos**.
- Los atributos definen las características individuales del objeto.
- Los métodos son funciones que le permiten al objeto realizar ciertas acciones.
- Por ejemplo, una silla sería un objeto. El color, la altura, el material, serían atributos. Y mover silla, pintar silla, serían métodos.
- Para acceder a un atributo: nombre\_objeto.atributo
- Para acceder a un método: nombre\_objeto.metodo(parametros)

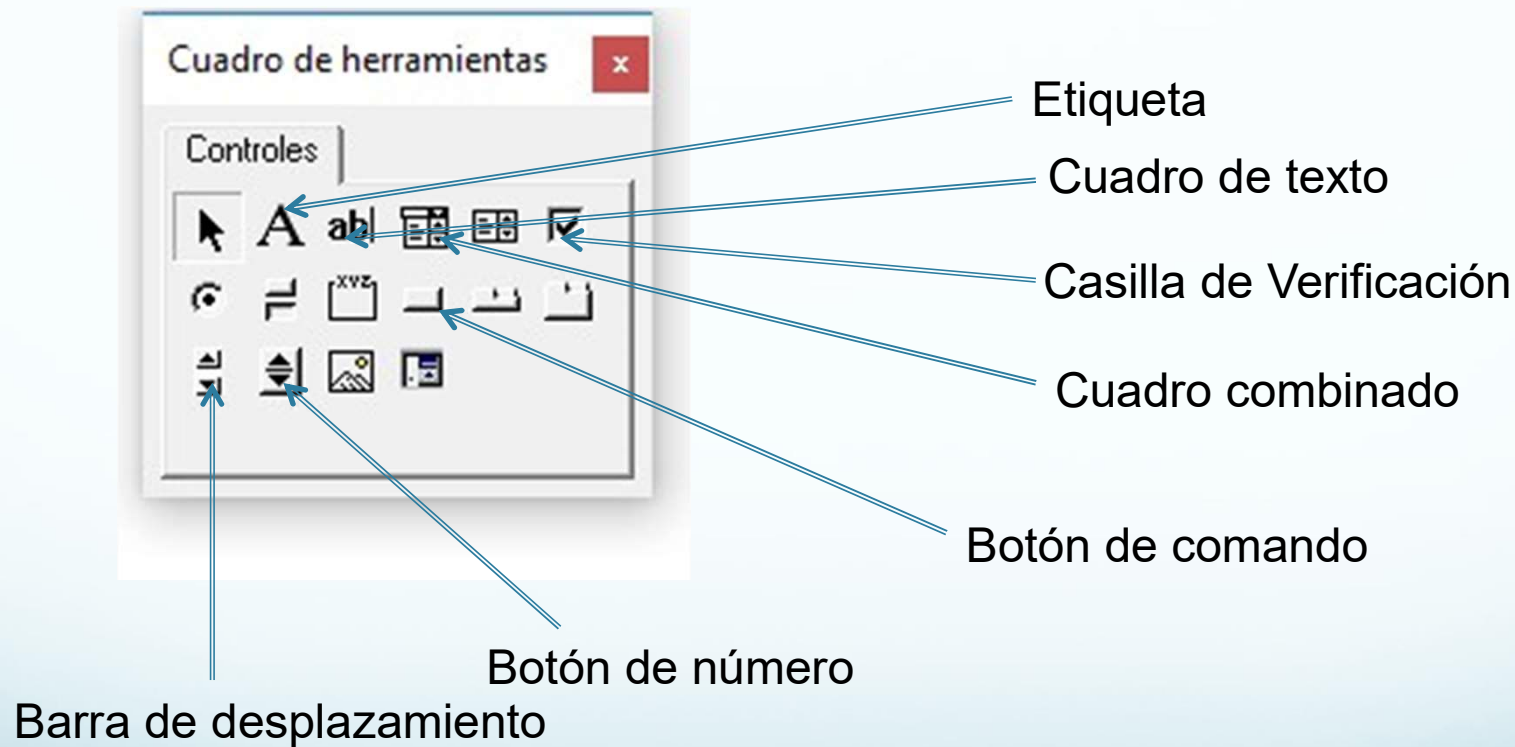
# Formularios

- Los formularios nos permiten ingresar los datos de manera más amistosa.
- Los formularios Excel utilizan objetos especiales conocidos como *controles de formulario* que son precisamente los que nos permiten agregar campos de texto, listas, botones de opción entre otras cosas más.

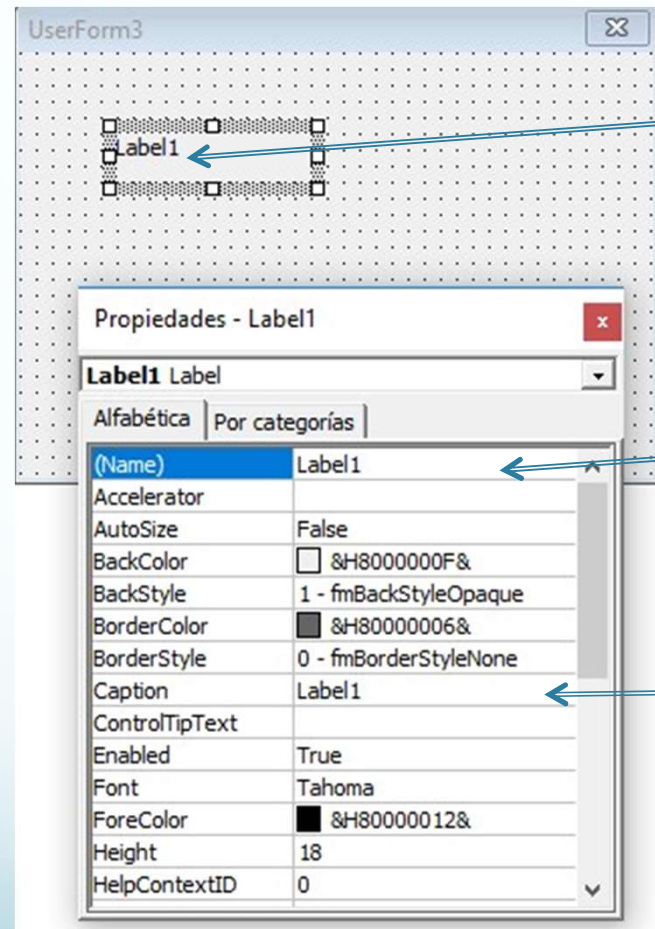
# Elementos de un formulario

- Revisaremos un poco más en detalle algunos de los elementos que podemos incluir en un formulario:
- Etiqueta
- Campo de texto
- Botón de comando
- Casilla
- Cuadro Combinado

# Elementos de un formulario (2)



# Etiqueta



Contenido de la etiqueta

etiqueta

Caption es el atributo que define el contenido de la etiqueta

# Modificar etiqueta dentro de una macro

```
Sub Macro()
```

```
    UserForm.Label1.Caption = "Etiqueta"
```

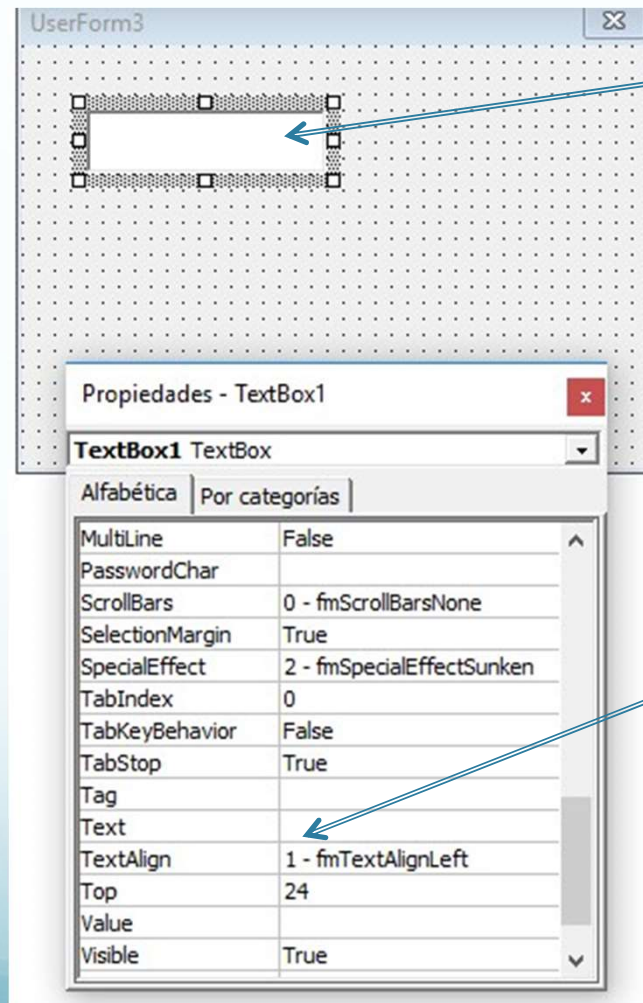
```
    Load UserForm 'carga el formulario
```

```
    UserForm.Show 'muestra el formulario
```

```
End Sub
```



# Cuadro de texto



Cuadro de texto

Text es el atributo que define el contenido del cuadro de texto

# Botón de comando

Al hacer dobleclick sobre el botón, Visual Basic nos permite declarar la función nombre\_boton\_Click(). Esta función indica lo que hace el botón, cuando el usuario presiona click sobre él.

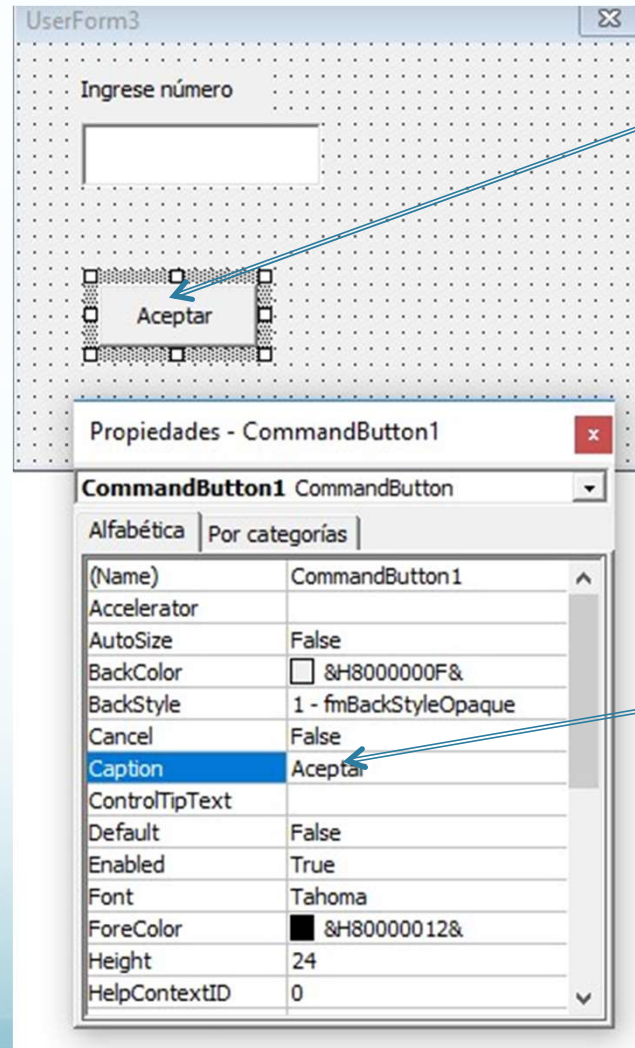
Private Sub

    CommandButton1\_Click()

    Cells(1, 1) = TextBox1.Text

End Sub

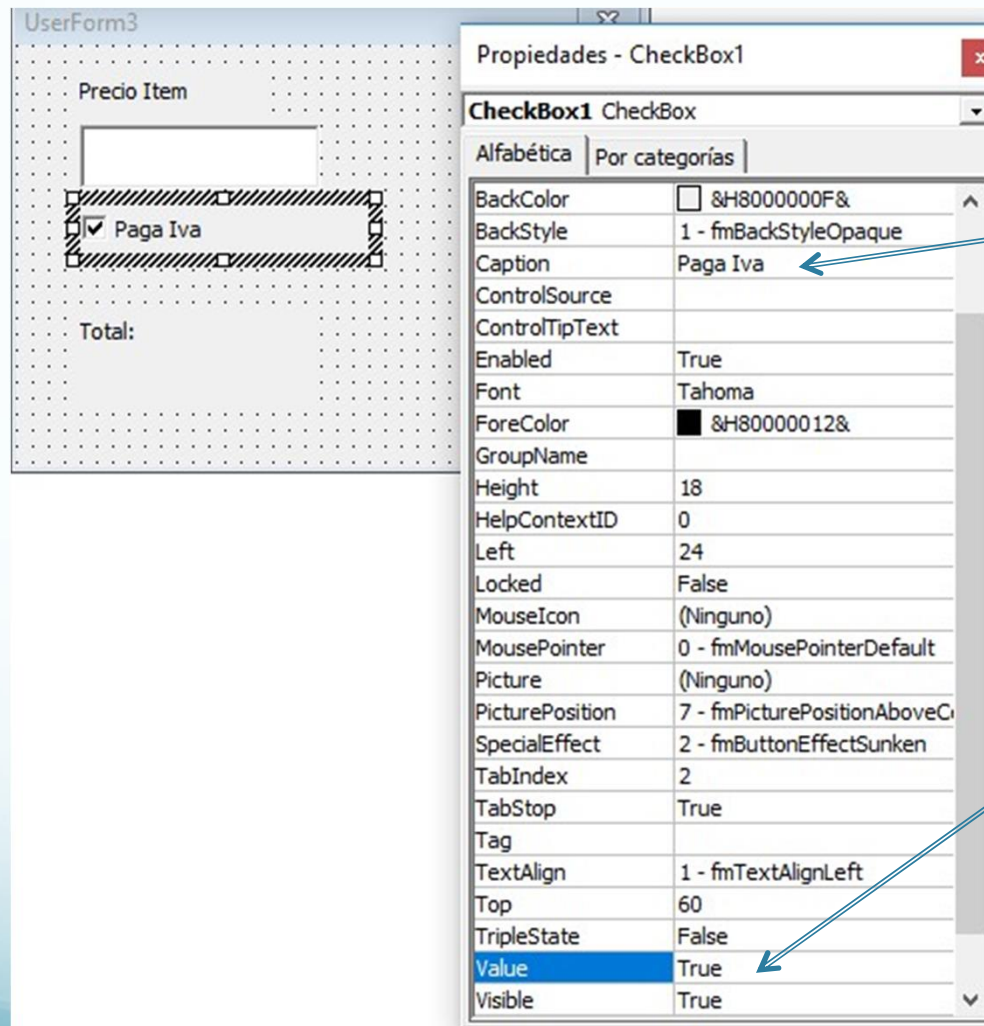
En este caso, el contenido del cuadro de texto se copia en la celda A1 de la planilla Excel.



Botón de comando

Caption es el atributo que define el mensaje del botón

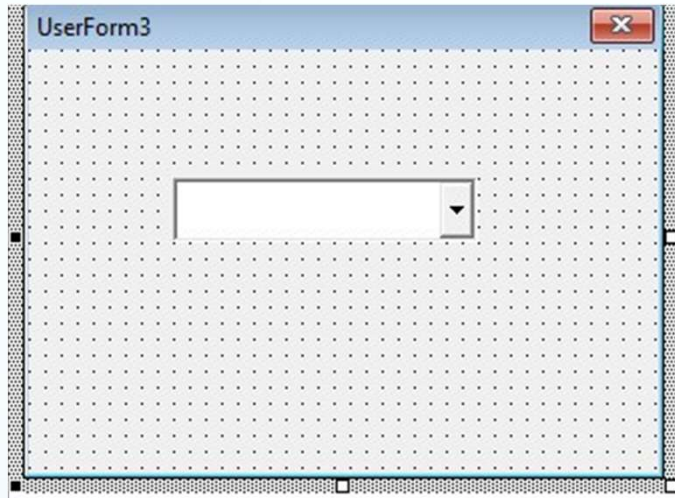
# Casilla



Contenido

Value es un atributo booleano: verdadero si la casilla está marcada, y falso en caso contrario.

# Cuadro Combinado



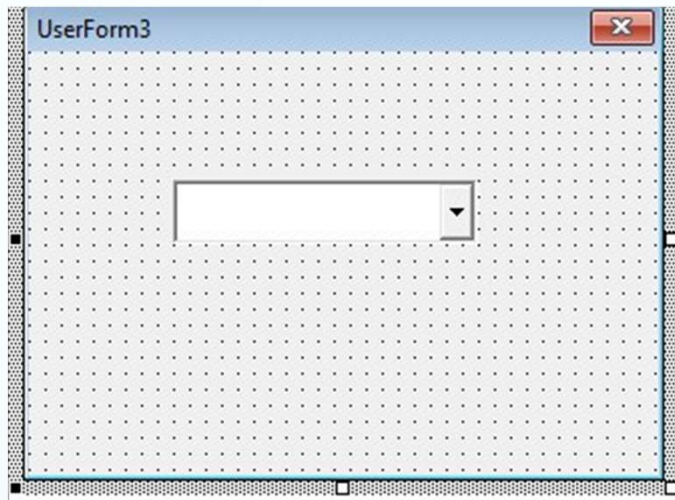
Para agregar opciones a la lista tenemos que definir la función Initialize en el formulario.

Para esto hacemos doble click en el formulario y seleccionamos initialize en la pestaña de la derecha

El método AddItem nos permite agregar una nueva opción.

El atributo .Value nos define el valor por defecto, si no ponemos nada, el cuadro por defecto se ve inicialmente en blanco.

# Cuadro Combinado



```
Private Sub UserForm_Initialize()  
    With ComboBox1  
        .AddItem "Ingeniería Civil Ambiental"  
        .AddItem "Ingeniería Civil Industrial"  
        .AddItem "Ingeniería Ambiental"  
        .AddItem "Ingeniería Informática"  
        .AddItem "Ingeniería Estadística"  
        .Value = "Ingeniería Civil Industrial"  
    End With  
End Sub
```

# Creando nuestro primer formulario

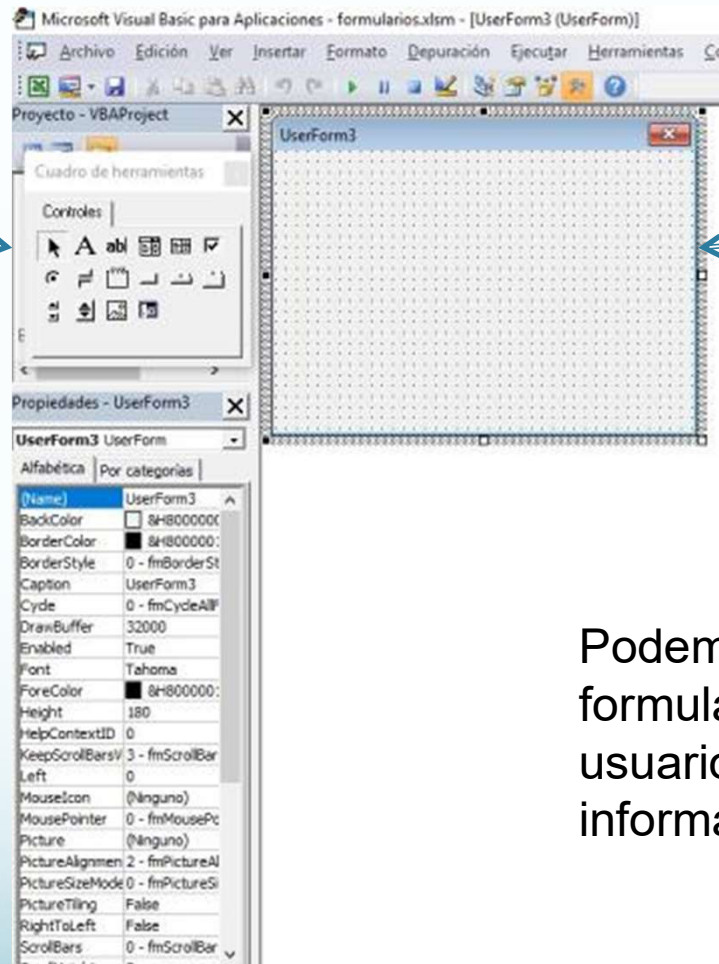
- En la pestaña programador, presionamos el botón Visual Basic
- Esto nos llevará a nuestra ya conocido ambiente de programación.
- Una vez allí, vamos a la pestaña Insertar y luego seleccionamos **UserForm**.



# Creando nuestro primer formulario (2)

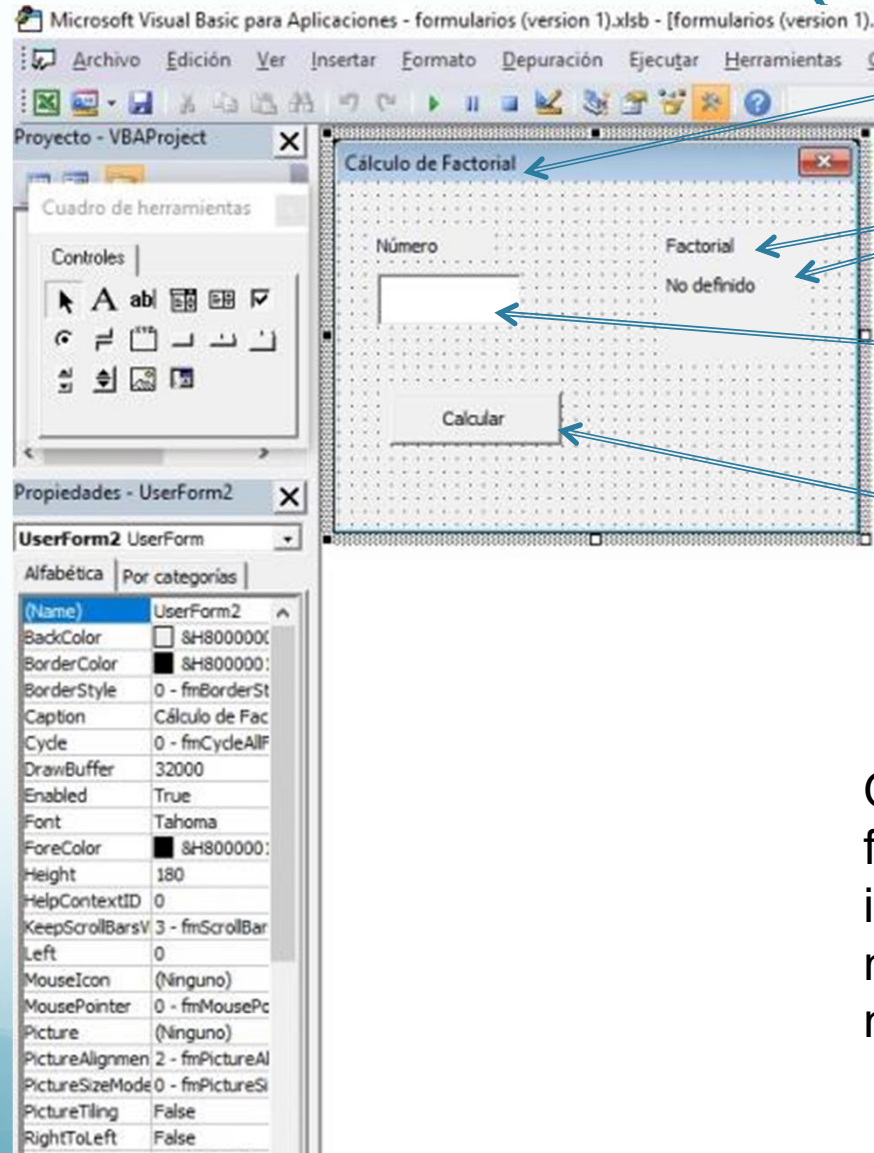
Controles

Formulario



Podemos agregar controles al formulario, de manera que el usuario pueda ingresar la información que requerimos.

# Creando nuestro primer formulario (3)



Campo  
caption  
del  
formulario

Etiqueta

Campo de texto

Botón

Queremos hacer un formulario que nos permita ingresar un número y mostrar el factorial de ese número



# Usando nuestro formulario

```
Function factorial(n As Integer) As Double
```

```
    Dim i As Integer
```

```
    If n < 0 Then
```

```
        factorial = -1
```

```
        Exit Function
```

```
    ElseIf n = 0 Then
```

```
        factorial = 1
```

```
        Exit Function
```

```
    Else
```

```
        factorial = 1
```

```
        For i = n To 1 Step -1
```

```
            factorial = factorial * i
```

```
        Next i
```

```
        Exit Function
```

```
    End If
```

```
End Function
```

# Usando nuestro formulario (2)

- Al hacer click sobre el botón Calcular, en el formulario, Excel nos permitirá programar lo que sucederá cuando el usuario haga click sobre dicho botón.

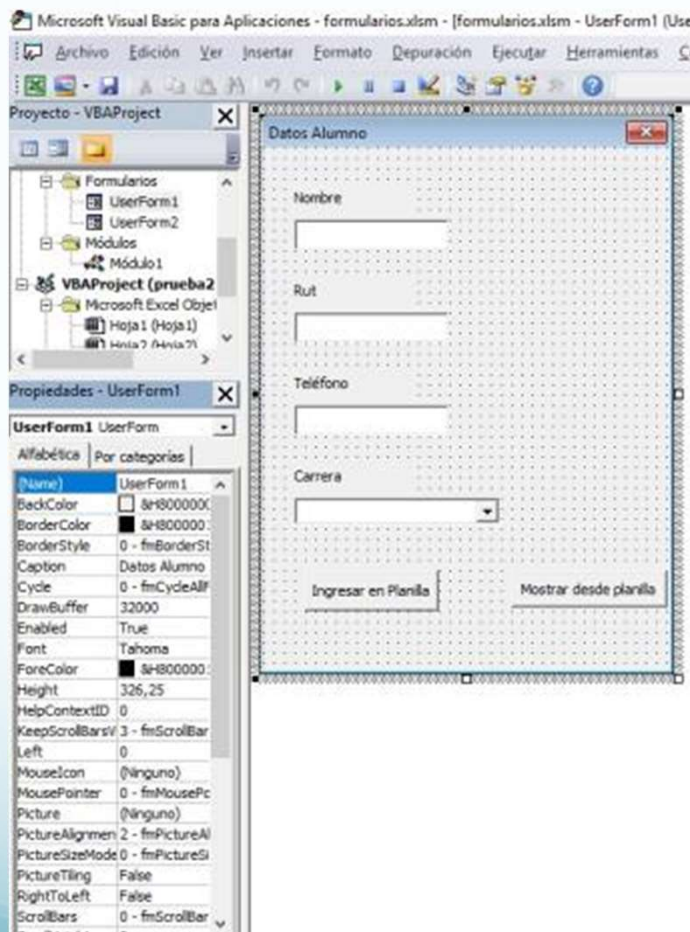
```
Private Sub CommandButton1_Click()  
    If factorial(TextBox1.Text) = -1 Then  
        Label2.Caption = "No definido"  
    Else  
        Label2.Caption = factorial(TextBox1.Text)  
    End If  
End Sub
```

# Usando nuestro formulario (3)

- Finalmente creamos una macro para cargar el formulario.

```
Sub Macro2()  
    Load UserForm2  
    UserForm2.Show  
End Sub
```

# Ejemplo



Se quiere hacer un formulario para ingresar los datos de un alumno.

El botón ingresar planilla debe poner los datos ingresados en la planilla en el siguiente formato:

	A	B	C	D
1	Nombre	Rut	Teléfono	Carrera
2	Juan Perez	24654356-1	94123123	Ingeniería Estadística

El botón “Mostrar desde planilla” coloca los datos que están en la planilla en el formulario.

# Ejemplo (2)

```
Private Sub UserForm_Initialize()  
    With ComboBox1  
        .AddItem "Ingeniería Civil Ambiental"  
        .AddItem "Ingeniería Civil Industrial"  
        .AddItem "Ingeniería Ambiental"  
        .AddItem "Ingeniería Informática"  
        .AddItem "Ingeniería Estadística"  
    End With  
End Sub
```

```
Private Sub CommandButton1_Click()  
    Cells(2, 1) = TextBox1.Text  
    Cells(2, 2) = TextBox2.Text  
    Cells(2, 3) = TextBox3.Text  
    Cells(2, 4) = ComboBox1.Value  
    TextBox1.Text = ""  
    TextBox2.Text = ""  
    TextBox3.Text = ""  
    ComboBox1.Text = ""  
    Cells.EntireColumn.AutoFit  
End Sub
```

# Ejemplo (3)

```
Private Sub CommandButton2_Click()  
    Dim i As Integer  
    Dim enc As Boolean  
    enc = False  
    TextBox1.Text = Cells(2, 1)  
    TextBox2.Text = Cells(2, 2)  
    TextBox3.Text = Cells(2, 3)  
    For i = 0 To (ComboBox1.ListCount - 1)  
        If ComboBox1.List(i) = Cells(2, 4) Then  
            ComboBox1.Text = ComboBox1.List(i)  
            enc = True  
        End If  
    Next i  
    If Not (enc) Then  
        MsgBox "Error: la carrera no está en la lista"  
    End If  
End Sub
```

# Ejemplo (4)

```
Sub Macro1()
```

```
    Range(Cells(1, 1), Cells(1, 4)).Interior.ColorIndex = 20
```

```
    Range(Cells(1, 1), Cells(1, 4)).Font.Bold = True
```

```
    Cells(1, 1) = "Nombre"
```

```
    Cells(1, 2) = "Rut"
```

```
    Cells(1, 3) = "Teléfono"
```

```
    Cells(1, 4) = "Carrera"
```

```
    MsgBox Worksheets(1).Rows.Count
```

```
    With Range(Cells(1, 1), Cells(1, 4)).Borders
```

```
        .LineStyle = xlContinuous
```

```
        .Weight = xlThin
```

```
    End With
```

```
    With Range(Cells(1, 1), Cells(1, 4)).Borders
```

```
        .LineStyle = xlContinuous
```

```
        .Weight = xlThin
```

```
    End With
```

# Ejemplo (5)

```
With Range(Cells(2, 1), Cells(2, 4)).Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlThin
End With
With Range(Cells(2, 1), Cells(2, 4)).Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlThin
End With
With Range(Cells(2, 1), Cells(2, 4)).Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
End With
Load UserForm1
UserForm1.Show
End Sub
```



# Desafío

- Modifique el código para que si el rut del alumno que se desea ingresar no está en la planilla, entonces escriba los datos del alumno en la siguiente fila desocupada, y arregle los bordes de la planilla para que encierren también a este nuevo alumno.

# Tablas como objetos

- Las tablas son una herramienta poderosa para el análisis y visualización de datos.
- Una tabla es simplemente un rango estructurado donde puede consultar diferentes secciones que se asignan automáticamente (como la fila del encabezado).
- Podemos agregar filtros sobre las columnas y así visualizar información de una parte de la tabla.
- Para crear una tabla usando Visual Basic:

```
Nombre_Hoja.ListObjects.Add(xlSrcRange, Rango, ,  
xlYes).Name = Nombre_Tabla
```

# Tablas como objetos (2)

- Para manipular la tabla, podemos crear una variable de tipo ListObject:

```
Dim Table As ListObject
```

```
Set Table = ActiveSheet.ListObjects("Table1")
```

- Para ver algunas de las operaciones que se pueden hacer con el objeto tabla:

<https://www.thespreadsheetguru.com/blog/2014/6/20/the-vba-guide-to-listobject-excel-tables>

# Ejemplo

- Supongamos que tenemos información en una hoja Excel, como se muestra en la siguiente imagen:

	A	B	C	D	E
1	Id	Vendedor	Producto	Precio	Mes
2	1	Juan Henriquez	PS4	260.000	Enero
3	2	Pedro Araya	Móvil	150.000	Febrero
4	3	Maria Bustamante	Tablet	80.000	Enero
5	4	Carolina Perez	Video Juego	38.000	Enero
6	5	Luis Quiñonez	Tablet	100.000	Febrero
7	6	Daniela Vidal	Video Juego	40.000	Febrero
8	7	Paz Martinez	Móvil	180.000	Marzo
9	8	Carolina Perez	Tablet	100.000	Febrero
10	9	Luis Quiñonez	Video Juego	42.000	Febrero
11	10	Carolina Perez	PS4	260.000	Enero
12	11	Luis Quiñonez	Móvil	170.000	Enero
13	12	Daniela Vidal	Video Juego	40.000	Febrero
14	13	Pedro Araya	Móvil	150.000	Febrero
15	14	Maria Bustamante	Tablet	110.000	Febrero
16	15	Daniela Vidal	Video Juego	35.000	Marzo
17	16	Paz Martinez	Móvil	250.000	Enero
18	17	Juan Henriquez	Tablet	100.000	Marzo
19	18	Pedro Araya	Video Juego	40.000	Marzo
20	19	Maria Bustamante	Móvil	150.000	Marzo

# Ejemplo

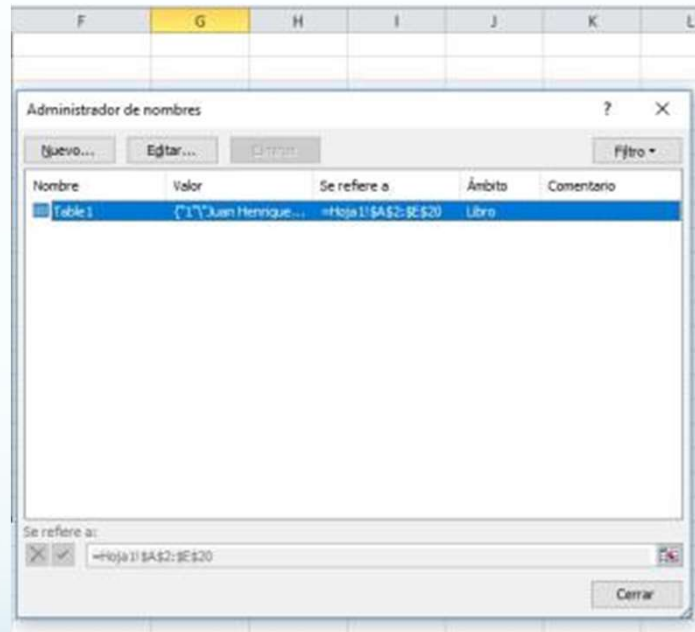
- Para crear la tabla:

```
Sub CrearTabla()  
    ActiveSheet.ListObjects.Add(xlSrcRange, Range("A1:E20"), , xlYes).Name =  
    "Table1"  
    ActiveSheet.ListObjects("Table1").TableStyle = ""  
End Sub
```

- El atributo TableStyle da el estilo de la tabla, si queremos personalizarlo, ponemos doble comilla.

## Ejemplo (2)

- En la pestaña **Fórmulas**, botón **Administrador de nombres** podemos ver la tabla creada.



- Usando el método `ListObjects(nombre_tabla)`, podemos referenciarla.

## Ejemplo (3)

- Observemos que el encabezado de nuestra tabla ha quedado con unas pestañas que nos permiten filtrar información

	A	B	C	D	E
1	Id	Vendedor	Producto	Precio	Mes
2	1	Juan Henriquez	PS4	260.000	Enero
3	2	Pedro Araya	Móvil	150.000	Febrero
4	3	Maria Bustamante	Tablet	80.000	Enero
5	4	Carolina Perez	Video Juego	38.000	Enero
6	5	Luis Quiñonez	Tablet	100.000	Febrero
7	6	Daniela Vidal	Video Juego	40.000	Febrero
8	7	Paz Martinez	Móvil	180.000	Marzo
9	8	Carolina Perez	Tablet	100.000	Febrero
10	9	Luis Quiñonez	Video Juego	42.000	Febrero
11	10	Carolina Perez	PS4	260.000	Enero
12	11	Luis Quiñonez	Móvil	170.000	Enero
13	12	Daniela Vidal	Video Juego	40.000	Febrero
14	13	Pedro Araya	Móvil	150.000	Febrero
15	14	Maria Bustamante	Tablet	110.000	Febrero
16	15	Daniela Vidal	Video Juego	35.000	Marzo
17	16	Paz Martinez	Móvil	250.000	Enero
18	17	Juan Henriquez	Tablet	100.000	Marzo
19	18	Pedro Araya	Video Juego	40.000	Marzo
20	19	Maria Bustamante	Móvil	150.000	Marzo



# Ejemplo (4)

	A	B	C	D	E
1	Id	Vendedor	Producto	Precio	Mes
			Ordenar de A a Z	260.000	Enero
			Ordenar de Z a A	150.000	Febrero
			Ordenar por color	80.000	Enero
			Borrar filtro de "Producto"	38.000	Enero
			Filtrar por color	100.000	Febrero
			Filtros de texto	40.000	Febrero
			Buscar	180.000	Marzo
			(Seleccionar todo)	100.000	Febrero
			Móvil	42.000	Febrero
			PS4	260.000	Enero
			Tablet	170.000	Enero
			Video Juego	40.000	Febrero
				150.000	Febrero
				110.000	Febrero
				35.000	Marzo
				250.000	Enero
				100.000	Marzo
				40.000	Marzo
				150.000	Marzo

	A	B	C	D	E
1	Id	Vendedor	Producto	Precio	Mes
2	1	Juan Henriquez	PS4	260.000	Enero
4	3	Maria Bustamante	Tablet	80.000	Enero
6	5	Luis Quiñonez	Tablet	100.000	Febrero
9	8	Carolina Perez	Tablet	100.000	Febrero
11	10	Carolina Perez	PS4	260.000	Enero
15	14	Maria Bustamante	Tablet	110.000	Febrero
18	17	Juan Henriquez	Tablet	100.000	Marzo

A la derecha vemos nuestra pequeña base de datos filtrada



# Agregando subtotales

- Nos gustaría tener una fila con el total de dinero ingresado por las ventas.
- Para eso podemos usar el atributo TotalsCalculation en las columnas:
- Para ver todas las operaciones que podemos incorporarle al atributo TotalsCalculation:
- <https://msdn.microsoft.com/en-us/vba/excel-vba/articles/xltotalscalculation-enumeration-excel>

# Agregando totales (2)

- Si queremos la suma:

```
ActiveSheet.ListObjects("Table1").ShowTotals = True  
ActiveSheet.ListObjects("Table1").ListColumns("Precio").TotalsC  
alculation = xlTotalsCalculationSum  
Cells.EntireColumn.AutoFit
```

- El atributo showTotals permite visualizar los totales.
- ¿Por qué no usar una simple fórmula?
- Porque si filtramos la tabla, el calculo se hace sobre los valores que aparecen en ese momento en la tabla (luego del filtro).

# Agregando totales (3)

	A	B	C	D	E	F
1	Id	Vendedor	Producto	Precio	Mes	Forma de Pago
2	1	Juan Henriquez	PS4	260.000	Enero	Contado
3	2	Pedro Araya	Móvil	150.000	Febrero	Credito
4	3	Maria Bustamante	Tablet	80.000	Enero	Credito
5	4	Carolina Perez	Video Juego	38.000	Enero	Credito
6	5	Luis Quiñonez	Tablet	100.000	Febrero	Contado
7	6	Daniela Vidal	Video Juego	40.000	Febrero	Contado
8	7	Paz Martinez	Móvil	180.000	Marzo	Contado
9	8	Carolina Perez	Tablet	100.000	Febrero	Contado
10	9	Luis Quiñonez	Video Juego	42.000	Febrero	Contado
11	10	Carolina Perez	PS4	260.000	Enero	Credito
12	11	Luis Quiñonez	Móvil	170.000	Enero	Credito
13	12	Daniela Vidal	Video Juego	40.000	Febrero	Credito
14	13	Pedro Araya	Móvil	150.000	Febrero	Contado
15	14	Maria Bustamante	Tablet	110.000	Febrero	Credito
16	15	Daniela Vidal	Video Juego	35.000	Marzo	Contado
17	16	Paz Martinez	Móvil	250.000	Enero	Credito
18	17	Juan Henriquez	Tablet	100.000	Marzo	Contado
19	18	Pedro Araya	Video Juego	40.000	Marzo	Credito
20	19	Maria Bustamante	Móvil	150.000	Marzo	Contado
21	Total			2.295.000		
22						

	A	B	C	D	E	F
1	Id	Vendedor	Producto	Precio	Mes	Forma de Pago
2	1	Juan Henriquez	PS4	260.000	Enero	Contado
3	2	Pedro Araya	Móvil	150.000	Febrero	Credito
4	3	Maria Bustamante	Tablet	80.000	Enero	Credito
5	4	Carolina Perez	Video Juego	38.000	Enero	Credito
6	5	Luis Quiñonez	Tablet	100.000	Febrero	Contado
8	7	Paz Martinez	Móvil	180.000	Marzo	Contado
9	8	Carolina Perez	Tablet	100.000	Febrero	Contado
10	9	Luis Quiñonez	Video Juego	42.000	Febrero	Contado
11	10	Carolina Perez	PS4	260.000	Enero	Credito
14	13	Pedro Araya	Móvil	150.000	Febrero	Contado
16	15	Daniela Vidal	Video Juego	35.000	Marzo	Contado
17	16	Paz Martinez	Móvil	250.000	Enero	Credito
18	17	Juan Henriquez	Tablet	100.000	Marzo	Contado
20	19	Maria Bustamante	Móvil	150.000	Marzo	Contado
21	Total			1.895.000		
22						

# Ejemplo (5)

- Para manipularla:

```
Sub ManipularTabla()  
    Dim Table As ListObject  
    Set Table = ActiveSheet.ListObjects("Table1")  
    Table.ListColumns.Add  
    Table.HeaderRowRange(Table.ListColumns.Count) = "Forma de Pago"  
    Worksheets(1).Range(Cells(2, Table.ListColumns.Count),  
Cells(Table.ListRows.Count + 1, Table.ListColumns.Count)).Value =  
Hoja2.Range("F2:F20").Value  
    Cells.EntireColumn.AutoFit  
End Sub
```

- Agregamos una columna a la tabla, cuya información está en la Hoja 2.

# Ejemplo (6)

	A	B	C	D	E	F
1	Id	Vendedor	Producto	Precio	Mes	Forma de Pago
2	1	Juan Henriquez	PS4	260.000	Enero	Contado
3	2	Pedro Araya	Móvil	150.000	Febrero	Credito
4	3	Maria Bustamante	Tablet	80.000	Enero	Credito
5	4	Carolina Perez	Video Juego	38.000	Enero	Credito
6	5	Luis Quiñonez	Tablet	100.000	Febrero	Contado
7	6	Daniela Vidal	Video Juego	40.000	Febrero	Contado
8	7	Paz Martinez	Móvil	180.000	Marzo	Contado
9	8	Carolina Perez	Tablet	100.000	Febrero	Contado
10	9	Luis Quiñonez	Video Juego	42.000	Febrero	Contado
11	10	Carolina Perez	PS4	260.000	Enero	Credito
12	11	Luis Quiñonez	Móvil	170.000	Enero	Credito
13	12	Daniela Vidal	Video Juego	40.000	Febrero	Credito
14	13	Pedro Araya	Móvil	150.000	Febrero	Contado
15	14	Maria Bustamante	Tablet	110.000	Febrero	Credito
16	15	Daniela Vidal	Video Juego	35.000	Marzo	Contado
17	16	Paz Martinez	Móvil	250.000	Enero	Credito
18	17	Juan Henriquez	Tablet	100.000	Marzo	Contado
19	18	Pedro Araya	Video Juego	40.000	Marzo	Credito
20	19	Maria Bustamante	Móvil	150.000	Marzo	Contado
21						

# Tablas dinámicas (pivot tables)

- Las **tablas dinámicas**, también llamadas *pivot tables*, son una herramienta para análisis de bases de datos (BD).
- Se encargan de resumir y ordenar la información de la BD.
- Permiten analizar sólo una porción de la BD, es decir, con una BD con gran cantidad de campos o columnas, ayudan a visualizar únicamente la información que queremos.
- Con esto podemos facilitar el análisis y la toma de decisiones.
- Para poder crear una pivot table necesita una pivot caché que contiene la información base.
- Se pueden crear varias pivot tables a partir de una pivot caché.



# Tablas dinámicas: Objetos, atributos y métodos útiles

- Para visualizar mejor los datos que analizaremos en la Tabla dinámica, se recomienda alojar la tabla en una nueva hoja de cálculo.
- Para esto, es recomendable tener una variable de tipo **Worksheet** (hoja de cálculo).
- El atributo **name** de un objeto tipo worksheet, permite conocer el nombre de la hoja.
- También necesitamos una variable de tipo PivotCaché para manejar los datos de origen, y una **PivotTable** para manipular la tabla dinámica destino.

# Tablas dinámicas: Objetos, atributos y métodos útiles

- La pivot Caché se crea desde el libro:

```
ActiveWorkbook.PivotCaches.Create(SourceType:=xlData  
base, SourceData:=hoja!rango_datos)
```

- La PivotTable se crea usando un método de la Pivotcaché

```
pvtCache.CreatePivotTable(TableDestination:=ubicación_  
destino, TableName:="nombre_Tabla")
```



# Tablas dinámicas: Objetos, atributos y métodos útiles (2)

- El método `hoja.PivotTables("nombre_tabla")` devuelve la Tabla pivote con el nombre especificado, que se encuentra en la hoja de cálculo especificada.
- El método `pivotFields("nombre_campo")` de un objeto `PivotTable`, devuelve el campo de la tabla especificado.
- Ese campo pivote tiene el atributo `Orientation` que puede ser: **`xlColumnField`** (columna), **`xlRowField`** (fila), **`xlPageField`** (página: para ir como campo de filtro).
- También tiene otro atributo llamado `Position` para fijar la posición (cuando tenemos más de un campo con la misma orientación)

# Creando Tablas dinámicas

```
Sub CrearPivotTable()
```

```
    Dim sht As Worksheet  
    Dim pvtCache As PivotCache  
    Dim pvt As PivotTable  
    Dim StartPvt As String  
    Dim SrcData As String
```

```
    'Define el rango de los datos que se quieren incluir en la tabla  
    SrcData = ActiveSheet.Name & "!" & Range("A1:F20").Address(ReferenceStyle:=xlR1C1)
```

```
    'Crear una nueva hoja luego de la original  
    Set sht = Sheets.Add(After:=Sheets(ActiveSheet.Name))
```

```
    'Ubicación destino donde poner la pivot table  
    StartPvt = sht.Name & "!" & sht.Range("A3").Address(ReferenceStyle:=xlR1C1)
```

```
    'Crear Pivot Cache desde los datos de origen  
    Set pvtCache = ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase,  
    SourceData:=SrcData)
```

```
    'Crear Pivot table a partir de la Pivot Cache  
    Set pvt = pvtCache.CreatePivotTable(TableDestination:=StartPvt,  
    TableName:="PivotTable1")
```

```
End Sub
```

# Agregando campos a una tabla dinámica

```
Sub agregar_campos_a_pivotTable()
```

```
    Dim pvt As PivotTable
```

```
    Set pvt = ActiveSheet.PivotTables("PivotTable1")
```

```
    'Agregar item a las filas
```

```
    pvt.PivotFields("Vendedor").Orientation = xlRowField
```

```
    pvt.PivotFields("Vendedor").Position = 1
```

```
    'Agregar un campo pivote
```

```
    pvt.AddDataField pvt.PivotFields("Precio"), "Suma de Precio", xlSum
```

```
    'Agregar un item de filtro
```

```
    pvt.PivotFields("Mes").Orientation = xlPageField
```

```
    'Agregar un item de filtro
```

```
    pvt.PivotFields("Producto").Orientation = xlPageField
```

```
    'Posición del item
```

```
    pvt.PivotFields("Producto").Position = 1
```

```
    'Actualiza automáticamente la tabla si hay cambios en la Pivot Table
```

```
    pvt.ManualUpdate = False
```

```
End Sub
```

# Agregando campos a una tabla dinámica

	A	B
1	Mes	(Todas) ▼
2	Producto	(Todas) ▼
3		
4	<b>Etiquetas de fila ▼</b>	<b>Suma de Precio</b>
5	Carolina Perez	398000
6	Daniela Vidal	115000
7	Juan Henriquez	360000
8	Luis Quiñonez	312000
9	Maria Bustamante	340000
10	Paz Martinez	430000
11	Pedro Araya	340000
12	<b>Total general</b>	<b>2295000</b>
13		

# Creando Tablas dinámicas

- Más información en:
- <https://www.thespreadsheetguru.com/blog/2014/9/27/vba-guide-excel-pivot-tables>