COLEGIO DE INGENIEROS DEL PERU
CONSEJO DEPARTAMENTAL DE LAMBAYEQUE
INSTITUTO DE ESTUDIOS PROFESIONALES DE INGENIERÍA IEPI
CENTRO DE CAPACITACIÓN

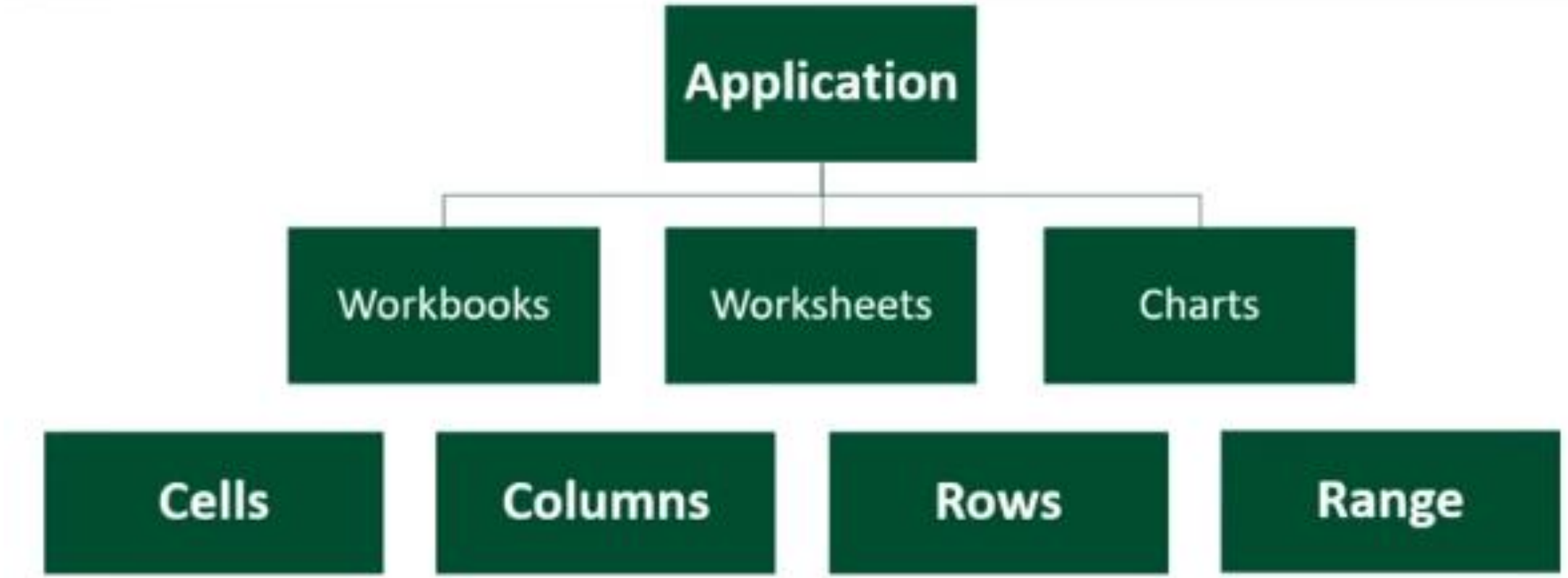
SESIÓN 03: EXCEL AVANZADO

Docente: Ing. Eric Alberto Heredia Mendoza

MACROS EN VBA

Para poder programar una Macros en VBA es necesario manejar la estructura de Jerarquía de Objetos.

JERARQUIA DE OBJETOS



JERARQUÍA DE OBJETO

Objeto Aplication- Representa la Aplicación Excel y todo lo que se puede encontrar dentro del programa

Objeto Workbooks- Representa el Libro de Excel en el cual trabajarás la macro

Objeto Worksheets-Representa la hoja de calculo en la cual trabajaremos la macro.

Objeto Charts-Representa una referencia a un gráfico.

JERARQUÍA DE OBJETO

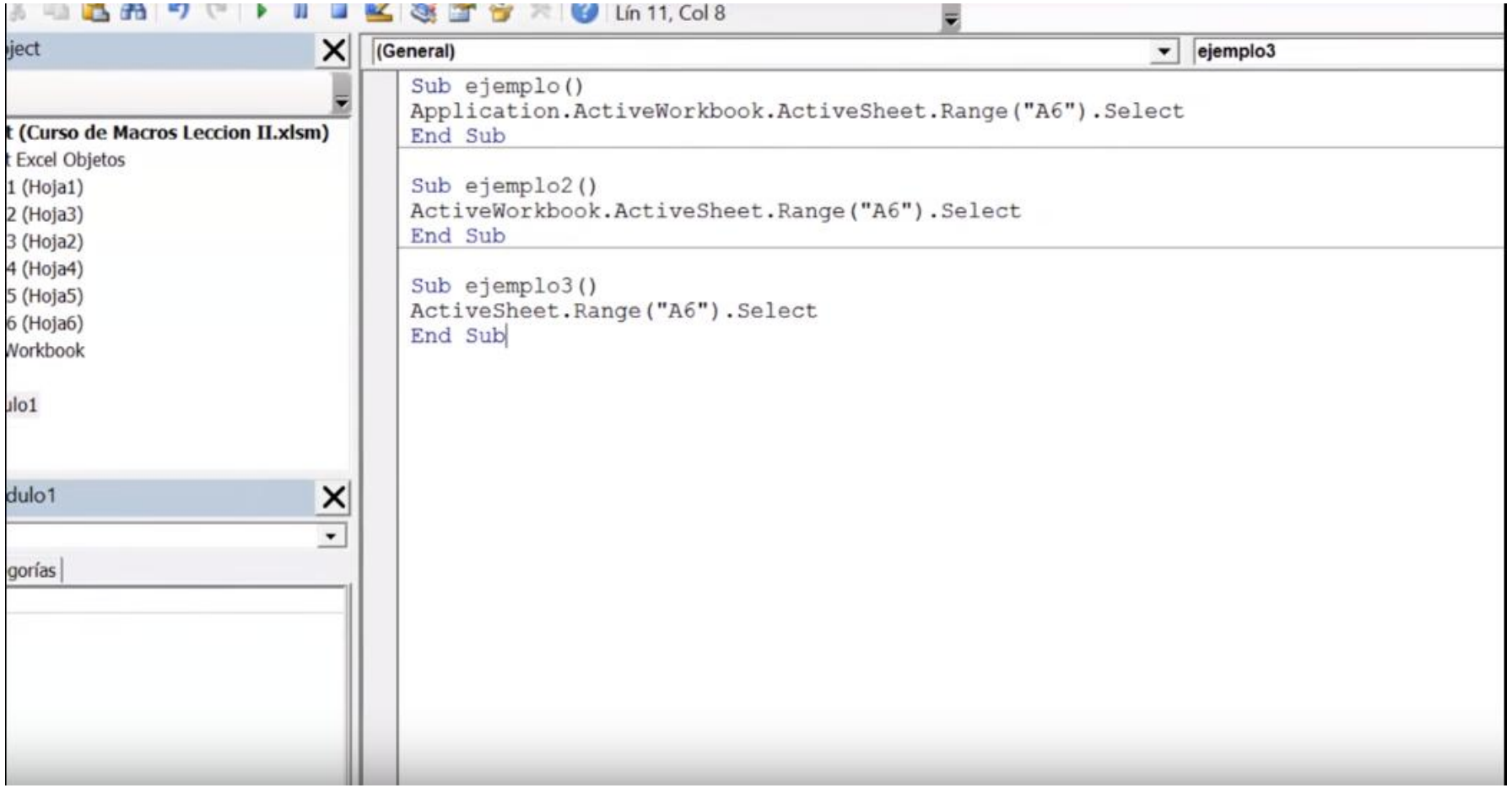
Objeto Cells- Representa a una celda estableciendo el número de columna y el número de fila

Objeto Column- Representa a una columna correspondiente a una hoja

Objeto Rows-Representa a una fila correspondiente a una hoja.

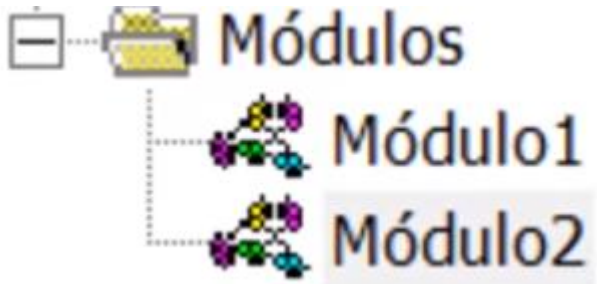
Objeto Range- Representa a una celda o a un rango de celdas de una hoja.

EJEMPLO DE PROGRAMACIÓN



MODULOS EN VBA

Nos permite agrupar los procedimientos y funciones que serán programados para realizar una acción concreta. En los módulos nosotros trabajaremos con procedimientos los cuales se detallarán a continuación



TIPOS DE PROCEDIMIENTOS

- Procedimientos SUB
- Procedimientos Function
- Procedimientos Property

PROCEDIMIENTOS SUB

Son aquellos que realizan una serie de acciones concretas y deben ser llamadas a través de la aplicación

Es recomendable segmentar los procedimientos en varias para facilitar su modificación.

Dentro de los Procedimientos Sub encontramos dos tipos con los cuales se pueden trabajar:

-Procedimientos Generales.

-Procedimientos de Eventos.

PROCEDIMIENTOS GENERALES

- Son aquellos que se declaran dentro un módulo y se deben llamar a través de la aplicación por medio de un botón, un comando u otros elementos

PROCEDIMIENTOS DE EVENTOS

- Son los que se ejecutan al identificar un evento provocado por el usuario o el sistema

-Al abrir

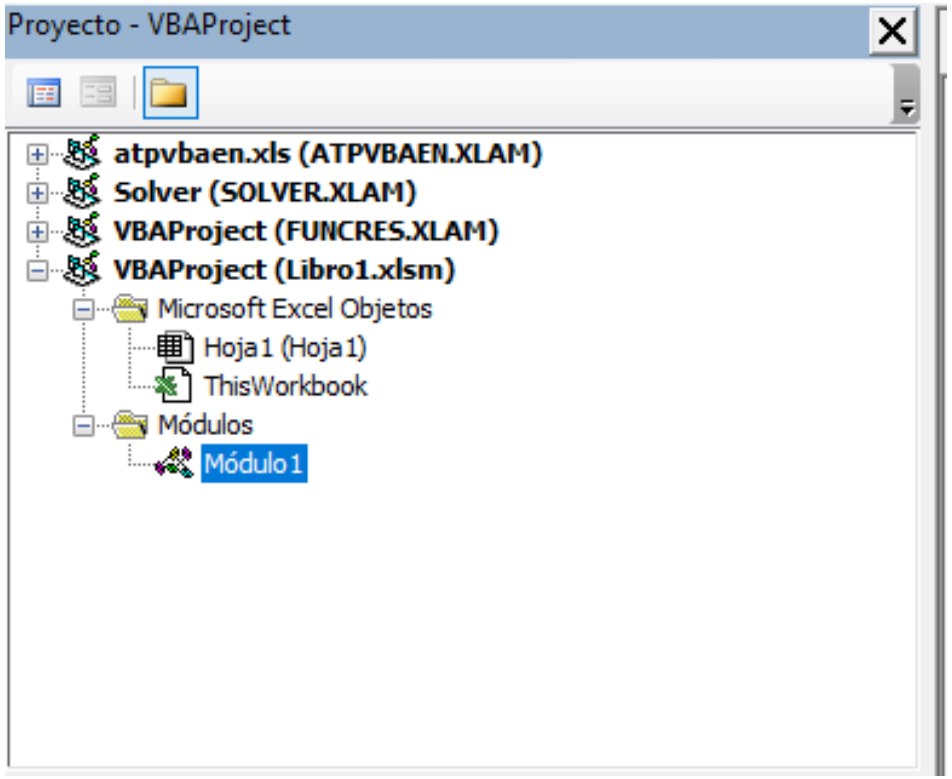
-Al Cerrar

-Después de

-Antes de

EXPLORADOR DE PROYECTOS

- Proyecto: Un proyecto de VBA es un libro de Excel con un conjunto de objetos para navegar entre los elementos que lo componen



VARIABLES EN VBA

- Es un elemento del código que sirve para guardar valores de manera temporal, que luego utilizaremos para la ejecución de una instrucción. El contenido que se guarda dentro de la variable se refiere a la información que deseamos guardar y se le llama valor de una variable
- Son las palabras que utilizaremos para hacer referencia a determinada información dentro de una macro. Se utiliza la variable debido a que muchos casos no se tiene la información de inicio sino que se va a generar y solicitar en el momento.

DECLARACIÓN DE UNA VARIABLE

- Declaración Implícita

Es aquella en donde no es necesario declarar una variable antes de utilizarla ya que la aplicación se encargara de crearla de forma automática cuando le asignemos un valor a dicha variable

Ejemplo:

```
Sub macro2()
a = 25
compra = 50

End Sub
```

DECLARACIÓN DE UNA VARIABLE

- Declaración Explícita

Debemos definir los nombres de todas las variables que vamos a utilizar antes de aplicarlas en el código

```
Sub macro2()
Dim a
Dim impuesto
a = 25
impuesto = 40

End Sub
```

TIPO DE DATOS DE UNA VARIABLE

- Los tipos de datos que pueden trabajar una variable son los siguientes:

- **Datos numéricos**
- **Datos de fecha y hora**
- **Datos de texto**
- **Datos booleanos**
- **Datos variant y,**
- **Datos de objeto**

TIPO DE DATOS DE UNA VARIABLE

Tipo	Valores
Byte	0 - 255
Integer	-32,768 a 32,768
Long	- 2,147,483,648 a 2,147,483,648
Single	32 bits y 7 decimales
Double	64 bits y 16 decimales
Currency	15 decimales y 4 a la derecha
Date	Fechas y Horas
String	Datos de texto
Boolean	Datos Lógicos
Variant	Cualquier valor
Object	Referencias a Objetos

FUNCIÓN INPUTBOX

- Nos muestra un elemento flotante, en el cual el usuario podrá introducir la información que le sea requerida. Si el usuario captura la información y presiona el botón Aceptar los caracteres se guardaran en una variable

InputBox(promt [,title][,default][,xpost][,ypost][,helpfile, context])

FUNCION INPUTBOX

- Ejemplo:

```
Sub ejemplo2()
Dim edad As Byte
edad = InputBox("Captura tu edad", "Edad del Usuario")
End Sub
```

```
Sub ejemplo3()
Dim edad As Byte
edad = InputBox("Captura tu edad", "Edad del Usuario",
End Sub
```

FUNCIÓN MSGBOX

- Nos ayudará cuando necesitamos una respuesta del tipo lógica, errores, alertas , advertencias, etc.

MsgBox(prompt[, buttons][, title][, helpfile, context])

Botones por mostrar	Valor	Constante
Aceptar	0	vbOkOnly
Aceptar y Cancelar	1	vbOkCancel
Anular reintentar Ignor.	2	bAbortRetryIgnore
Sí no Cancelar	3	vbYesNoCancel
Sí y No	4	vbYesNo
Reintentar y Cancelar	5	vbRetryCancel

CONDICIONAL IF THEN ELSE

Las instrucciones ***If...Then...Else*** se pueden presentar en varios formatos, con unas características determinadas. Normalmente, se presentan anidadas en tantos niveles como sea necesario. Esto, sin embargo, puede hacer menos legible el código, por lo que es aconsejable utilizar una instrucción ***Select Case*** en vez de recurrir a múltiples niveles de instrucciones ***If...Then...Else*** anidadas (únicamente en caso de que el excesivo número de anidamientos pudiera dar problemas en la legibilidad del programa, o errores en la depuración de éste).

```
If b > 0 And c > 0 Then
        Hoja1.Cells(i, 8) = a
Else
        If b > 0 And c < 0 Then
                Hoja1.Cells(i, 8) = a + 360
        Else
                Hoja1.Cells(i, 8) = a + 180
        End If
End If
```