



NOMBRE DEL ALUMNO: VALERIA TRUJILLO YAÑEZ

NOMBRE : SUPER NOTA

PARCIAL: 4

**NOMBRE DE LA MATERIA: PROCESAMIENTO DE LA INFORMACIÓN CON
HOJA DE CÁLCULO**

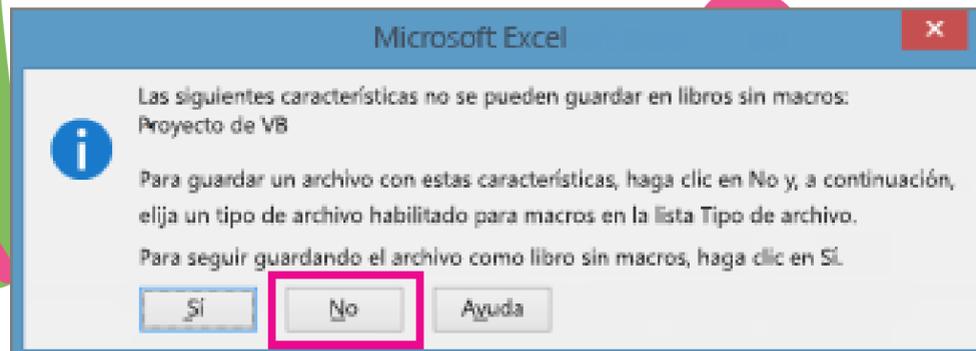
NOMBRE DEL PROFESOR: ANDRES ALEJANDRO REYES MOLINA

NOMBRE DE LA LICENCIATURA: PSICOLOGÍA

CUATRIMESTRE: 5

Guardar un libro con macros

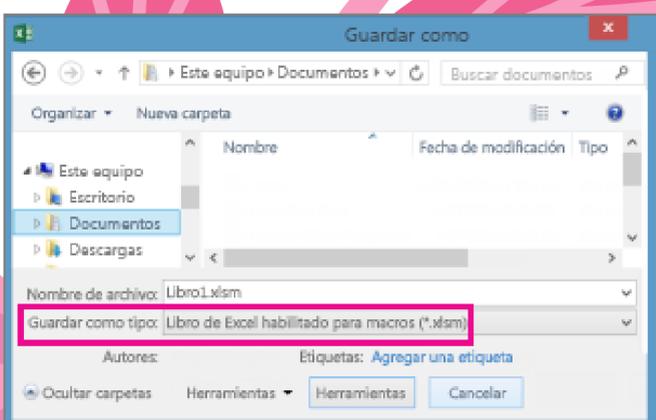
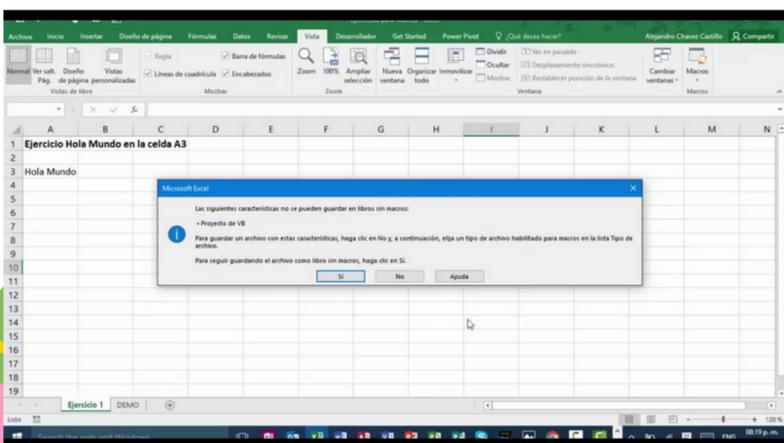
Si solo necesita la macro en el libro donde la creó, haga clic en Guardar o Guardar como lo haría normalmente. Pero guardar un libro con macros es algo diferente, ya que debe estar en un formato de archivo especial "habilitado para macros". Por lo tanto, cuando intenta guardarlo, Excel le pide dos opciones:



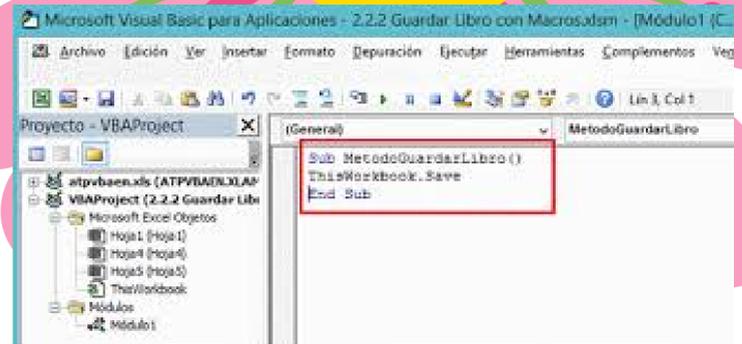
- Para guardarlo como un libro habilitado para macros (tipo de archivo *.xlsm), haga clic en no.
- Para guardarlo como libro sin macros, haga clic en sí.

Para guardarlo como libro habilitado para macros:

1. Haga clic en No.
2. En el cuadro Guardar como, en el cuadro de lista Guardar como tipo, elija Excel Macro-Enabled libro (*.xlsm).
3. Haga clic en Guardar.



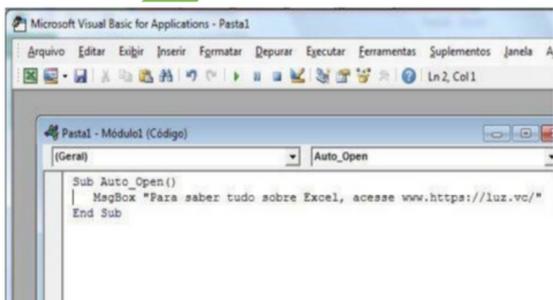
1) Mostrando un mensaje Sólo tienes que insertar en tu macro el comando MsgBox. En el ejemplo siguiente: `MsgBox "texto del mensaje"`



Esta línea de código se puede insertar en cualquier macro, muy útil en los casos donde se hace necesario emitir advertencias al usuario de la hoja de trabajo que está ejecutando la macro

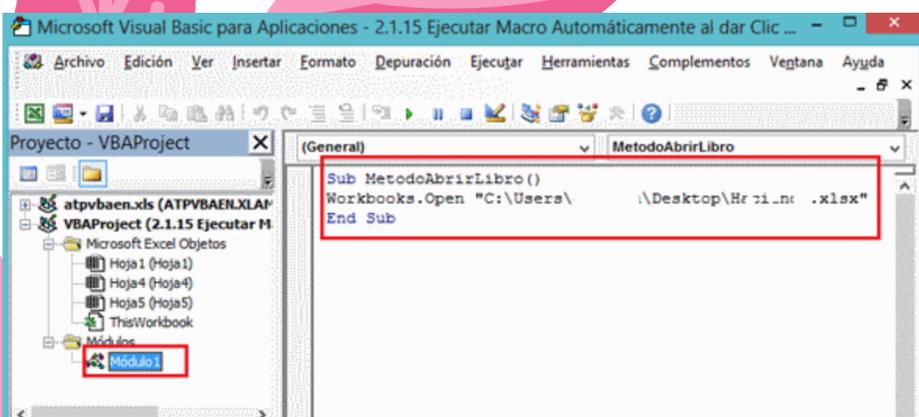
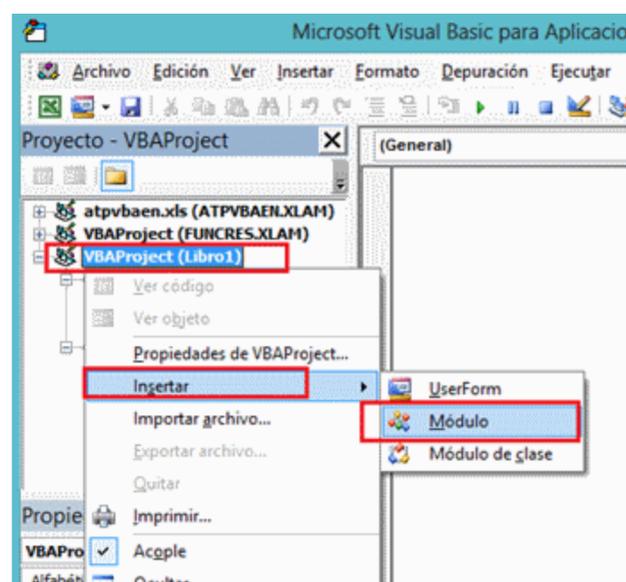
Abrir un libro con macro

Vamos a aplicar en esta ocasión, el Método Open (Abri) mediante el cual podemos programar la apertura de cualquier libro de Excel que tengamos en nuestro equipo de trabajo.



Método Open: Con la aplicación de este método, lo que estamos indicando desde VBA es la opción de Abrir un libro de Excel indicando la ruta donde se encuentra en nuestro equipo

Desde Excel abrimos el Editor VBA (Alt + F11), insertamos un Módulo (Clic derecho sobre el proyecto Excel, Insertar, Módulo):

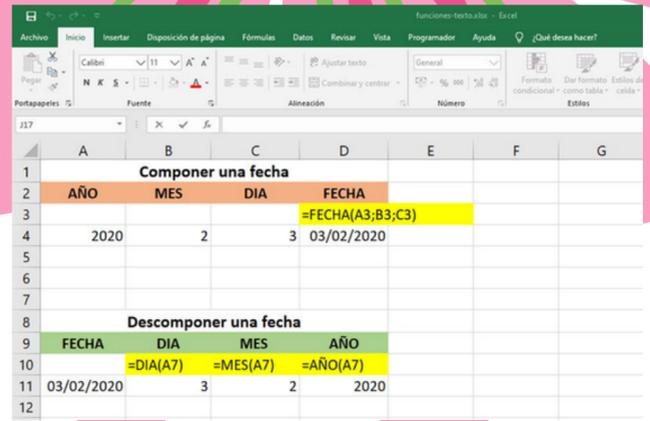


1. La primera línea `Sub MetodoAbrirLibro ()`, indica el inicio de la macro.
2. En la segunda línea estamos utilizando del Objeto Workbooks el Método Open el cual nos permite Abrir cualquier libro de Excel que tengamos en nuestro computador y para lograrlo, basta con indicar entre comillas, la ruta donde se encuentra el archivo Excel que queremos abrir.
3. La última línea marca el fin (`End`) de la Macro o Subrutina (`Sub`)

Macros y seguridad

Hoja de trabajo con fecha y hora actual

Puede escribir la fecha y la hora actual en la hoja de cálculo. El siguiente ejemplo inserta estos datos en la celda A1, vea. Sub escribirDataEHora () Range ("A1") = NowEnd Sub



Componer una fecha			
AÑO	MES	DIA	FECHA
2020	2	3	03/02/2020

Descomponer una fecha			
FECHA	DIA	MES	AÑO
03/02/2020	3	2	2020

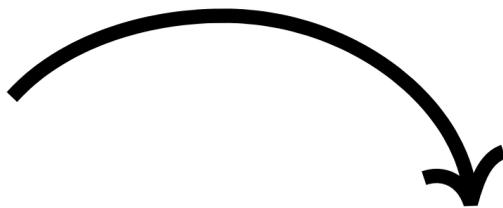


Realice la misma acción en cada celda seleccionada

```
Microsoft Visual Basic for Applications - Pasta1
Arquivo Editar Exibir Inserir Formato Depurar Ejecutar Herramientas Suplementos Janela Ayuda
Ln 5, Col 8

Pasta1 - Módulo1 (Código)
(Geral) Auto_Open
Sub hacerAlgoACadaCelula()
  For Each cell In Selection.Cells
    MsgBox cell
  Next
End Sub
```

En algún momento usted necesitará ejecutar una acción en las células que fueron seleccionadas por el usuario. Para ello, cree una macro igual al ejemplo siguiente. Sub hacerAlgoACadaCelula () For Each cell In Selection.Cells MsgBox cell



La misma cosa en todas las celdas seleccionadas

Usted puede hacer la misma macro anterior pero de forma un poco más amplia, o sea que la acción no se realizará en etapas. El ejemplo siguiente escribe "Hola" en todas las celdas. Sub hacerAlgoATodasAsCelulas () Selection.Cells.Value = "Hola" End Sub

```
Microsoft Visual Basic for Applications - Pasta1
Arquivo Editar Exibir Inserir Formato Depurar Ejecutar Herramientas Suplementos Janela Ayuda
Ln 2, Col 1

Pasta1 - Módulo1 (Código)
(Geral) Auto_Open
Sub hacerAlgoATodasAsCelulas()
  Selection.Cells.Value = "014"
End Sub
```

Usted puede hacer una infinidad de inserciones en el código, como por ejemplo cambiar el color del texto, el tamaño de fuente, el tipo de fuente y mucho más

Identificación de las fórmulas dentro de cada celda

Es muy recomendable verificar si hay algo dentro de la celda, siempre que sea necesaria una modificación de la celda. En este ejemplo comprobar si existe alguna fórmula en la celda en cuestión.

```
Microsoft Visual Basic for Applications - Pasta1
Arquivo Editar Exibir Inserir Formato Depurar Ejecutar Herramientas Suplementos Janela Ayuda
Ln 7, Col 8

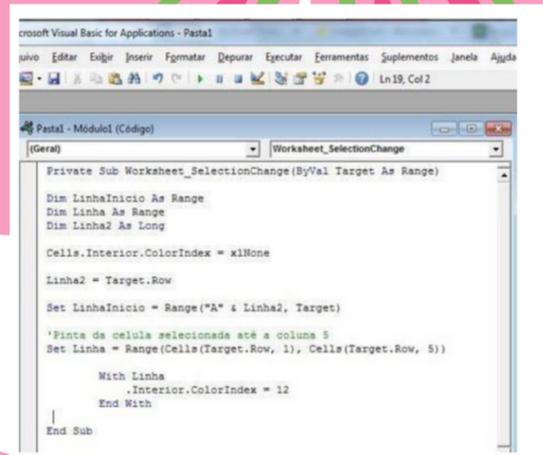
Pasta1 - Módulo1 (Código)
(Geral) Auto_Open
Sub verificarFormula()
  IF Range("A1").HasFormula = True Then
    MsgBox "Existe Formula"
  ELSE
    MsgBox "No é uma Formula"
  END IF
End Sub
```

Sub comprobarFormula () ("A1"). HasFormula = True Then MsgBox "Hay Fórmula" otro MsgBox "no es una fórmula" Si terminar End Sub



Cambiar el color de la celda al pasar el ratón

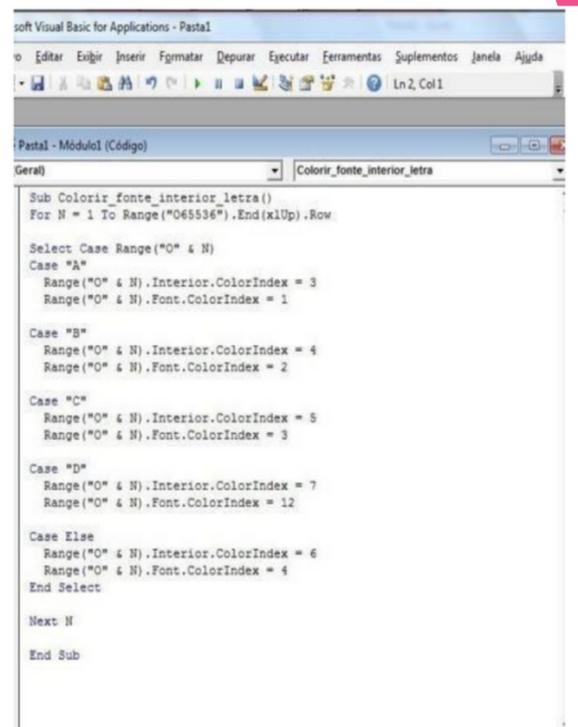
También es posible que desee proporcionar más interactividad con los usuarios de la hoja de cálculo. Lo ideal es que la interactividad tenga el objetivo de facilitar el servicio y visualización de las informaciones. La siguiente macro es un ejemplo de cómo puede hacer que su tabla muestre las celdas con el mismo tono de color cada vez que el ratón pasa por un dado que correlaciona estas celdas



```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Dim LinhaInicio As Range
    Dim Linha As Range
    Dim Linha2 As Long
    Cells.Interior.ColorIndex = xlNone
    Linha2 = Target.Row
    Set LinhaInicio = Range("A" & Linha2, Target)
    'Pinta da célula selecionada até a coluna 5
    Set Linha = Range(Cells(Target.Row, 1), Cells(Target.Row, 5))
    With Linha
        .Interior.ColorIndex = 12
    End With
End Sub
```

Cambiando el color del interior y de las fuentes

Esta macro, cambia los colores en el interior y en la fuente de las celdas, de acuerdo con la letra de las celdas.



```
Sub Colorir_fonte_interior_letra()
    For N = 1 To Range("O65536").End(xlUp).Row
        Select Case Range("O" & N)
            Case "A"
                Range("O" & N).Interior.ColorIndex = 3
                Range("O" & N).Font.ColorIndex = 1
            Case "B"
                Range("O" & N).Interior.ColorIndex = 4
                Range("O" & N).Font.ColorIndex = 2
            Case "C"
                Range("O" & N).Interior.ColorIndex = 5
                Range("O" & N).Font.ColorIndex = 3
            Case "D"
                Range("O" & N).Interior.ColorIndex = 7
                Range("O" & N).Font.ColorIndex = 12
            Case Else
                Range("O" & N).Interior.ColorIndex = 6
                Range("O" & N).Font.ColorIndex = 4
        End Select
    Next N
End Sub
```

Macro que habla

Ya pensó que interesante tener una hoja de trabajo que habla los datos contenidos en ella, es exactamente eso que esa macro hace. En el ejemplo siguiente la hoja de cálculo tenía palabras que iban desde la celda A1 hasta la A5, pero usted puede elegir el intervalo que le parezca más conveniente, para eso basta con cambiar el A1: A5 presentes en el código abajo, por el intervalo que prefiera

