

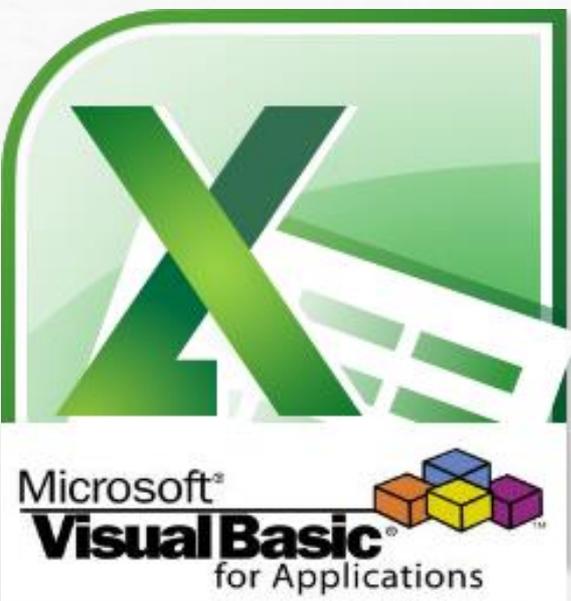


MICROSOFT EXCEL

HERRAMIENTAS INFORMÁTICAS II



Prof. Ing. Norma Cañizares



Microsoft Excel

Visual Basic for Applications



Visual Basic for Applications

Estructuras de Control

- Estructuras de Decisión
 - ✓ If Then Else
 - ✓ Case
- Estructuras De Repetición
 - ✓ Repetir Para (For)
 - ✓ Hacer Mientras (Do While)
 - ✓ Repetir Hasta (Do Until)

Microsoft Excel

ESTRUCTURA DE DECISIÓN: IF .. ELSE

Las estructuras condicionales son instrucciones de programación que permiten controlar la ejecución de un fragmento de código en función de si se cumple o no una condición.

Estructura If..Else

If Condición **Then**

Sentencia1

...

SentenciaN

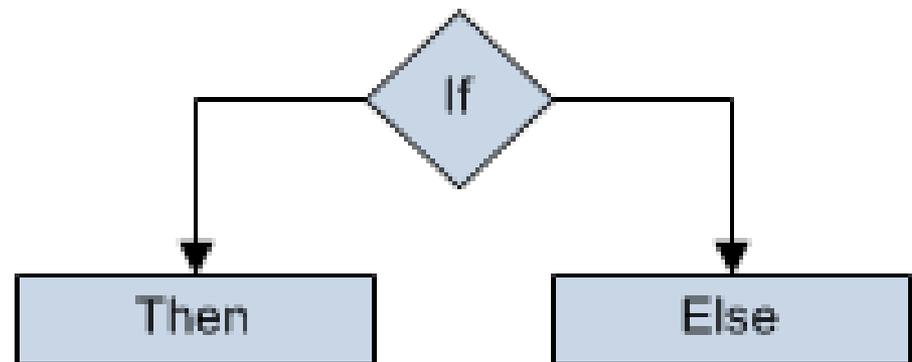
Else

Sentencia1

.

SentenciaN

End If



EJERCICIO 1:

- Efectuar la resta de los valores, de B2 y B3, y guardar el resultado en la celda B4.
- Si el resultado es positivo o 0, colocar la fuente de B4 en azul, sino en rojo.
- La resta solo se lleva a cabo cuando los valores para los operandos son valores numéricos.

	A	B	C	D
1	Operación: Resta			
2	Operando 1:		Resta	
3	Operando 2:			
4	Resultado:			
5				
6				
7				

RESOLUCIÓN: EJERCICIO 1.

	A	B	C	D
1	Operación: Resta			
2	Operando 1:		Resta	
3	Operando 2:			
4	Resultado:			
5				
6				
7				

Option Explicit

```
Private Sub cmdResta_Click()  
'EJERCICIO 1  
Range("B4").Value = ""  
If IsNumeric(Range("B2").Value) And Range("B2").Value <> "" Then  
    If IsNumeric(Range("B3").Value) And Range("B3").Value <> "" Then  
        Range("B4").Value = Range("B2").Value - Range("B3").Value  
        If Range("B4").Value >= 0 Then  
            Range("B4").Font.Color = RGB(0, 0, 255)  
        Else  
            Range("B4").Font.Color = RGB(255, 0, 0)  
        End If  
    Else  
        MsgBox ("Debe ingresar un valor numerico para el Operando 2")  
        Range("B3").Value = ""  
    End If  
Else  
    MsgBox ("Debe ingresar un valor numerico para el Operando 1")  
    Range("B2").Value = ""  
End If  
  
End Sub
```


RESOLUCIÓN EJERCICIO 2: CON ESTRUCTURA IF.. THEN.

	B	C	D	E	F	G	H	I
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								

```
Private Sub VerificarTipoDato_Click()  
'EJERCICIO 2  
Dim RespuestaTexto As String  
Dim valor As Variant  
  
valor = Range("E7").Value  
If IsError(valor) = True Then  
    RespuestaTexto = "Error"  
Else  
    If IsEmpty(valor) = True Then  
        RespuestaTexto = "Vacío"  
    Else  
        If WorksheetFunction.IsLogical(valor) = True Then  
            RespuestaTexto = "Booleano"  
        Else  
            If IsNumeric(valor) = True Then  
                RespuestaTexto = "Numero"  
            Else  
                If IsDate(valor) = True Then  
                    RespuestaTexto = "Fecha"  
                Else  
                    If WorksheetFunction.IsText(valor) = True Then  
                        RespuestaTexto = "Texto"  
                    Else  
                        RespuestaTexto = "Tipo NO Definido"  
                    End If  
                End If  
            End If  
        End If  
    End If  
End If  
End If  
End If  
End If  
  
Range("E10") = RespuestaTexto  
  
End Sub
```

ESTRUCTURA DE DECISIÓN: SELECT CASE

Las estructuras condicionales son instrucciones de programación que permiten controlar la ejecución de un fragmento de código en función de una condición.

Estructura Select Case.

Select Case Expresión

Case Etiq1
 [Sentencia1]

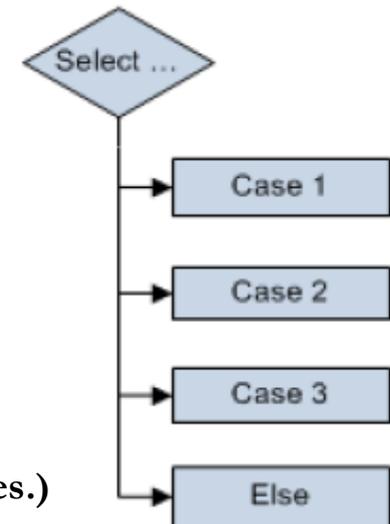
Case Etiq2
 [Sentencia2]

.....

Case Else
 [SentenciaN]

(Instrucciones en caso que no sean ninguno de los valores anteriores.)

Se usa en aquellos casos, cuando en función del valor de una variable se realizan una u otra acción.



End Select

Donde **Expresión** es una expresión numérica o alfanumérica, y **Etiq1, Etiq2,..** pueden adoptar las siguientes formas:

- 1.- Expresión.
- 2.- Expresión To Expresión
- 3.- Is Operador de Relación Expresión
- 4.- Combinación de las anteriores separadas por comas.

Ej → **Case 1 To 4, 7 To 9, 11, 13, Is > MaxNumber**

La palabra clave **To** especifica un rango de valores. Si usa la palabra clave To, el valor inferior debe aparecer antes de To.

Use la palabra clave **Is** con operadores de comparación

RESOLUCIÓN EJERCICIO 2: CON ESTRUCTURA SELECT CASE.

Option Explicit

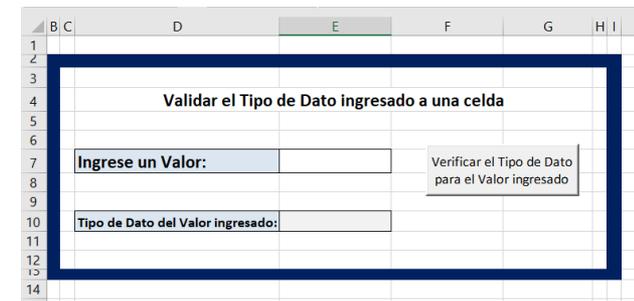
```
Private Sub VerificarTipoDato_Click()  
Dim RespuestaTexto As String  
Dim valor As Variant  
Dim Bandera As Boolean  
  
Bandera = True  
valor = Range("E7").Value  
Select Case Bandera  
    Case IsError(valor) = True  
        RespuestaTexto = "Error"  
    Case IsEmpty(valor) = True  
        RespuestaTexto = "Vacío"  
    Case WorksheetFunction.IsLogical(valor) = True  
        RespuestaTexto = "Booleano"  
    Case IsNumeric(valor) = True  
        RespuestaTexto = "Numero"  
    Case IsDate(valor) = True  
        RespuestaTexto = "Fecha"  
    Case WorksheetFunction.IsText(valor) = True  
        RespuestaTexto = "Texto"  
    Case Else  
        RespuestaTexto = "Tipo NO Definido"  
End Select  
  
Range("E10") = RespuestaTexto  
  
End Sub
```

En este ejemplo, para la Expresión de la Estructura del Case, se define una variable auxiliar de tipo Boolean, Bandera, y se la inicializa en True para que el Case se ejecute en condiciones normales.

Tipo de dato de la Expresión del Case: Valor Lógico

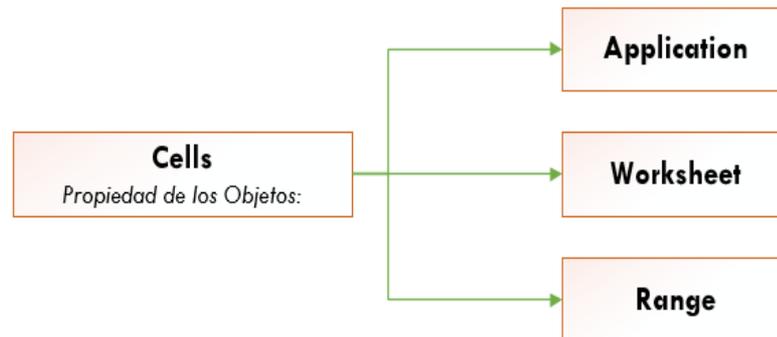
Valor devuelto por cada función para evaluar el tipo de dato: Valor Lógico

Se deben corresponder en el Tipo de Dato



PROPIEDAD CELLS

- VBA provee múltiples maneras de llevar a cabo una tarea y podemos escoger la mejor según la tarea que estemos realizando.
- Ahora hablamos de una **Forma Alternativa De Trabajar Con Rangos**, dado que en lugar de utilizar el *Objeto Range* también podemos utilizar la **Propiedad Cells** de alguno de los siguientes objetos.



- Podemos utilizar la **Propiedad Cells**, de los **objetos Application, Worksheet y Range**, para hacer referencia a un rango de celdas que el objeto contiene. Pueden ser incluso todas las celdas de una hoja, una pequeña parte o una única celda.

EJEMPLOS DE USO: PROPIEDAD CELLS

Ejemplos:

Con la Propiedad Cells

Application.Cells(*Fila,Columna*)

Propiedad Cells de Application Pertenece al **<Global>**

Application.Cells(37,"G")

Application.Cells(37,7)

Hace referencia a la celda G37 de la HOJA ACTIVA

Cells(37,"G")

Cells(37,7)

Hace referencia a una HOJA NO ACTIVA

Por ejemplo la celda F4 de la hoja Ex. 1

Worksheets("Ex. 1").Cells(4,6)

Con el Objeto Range

Range Pertenece al **<Global>**

Application.Range("G37")

Hace referencia a la celda G37 de la HOJA ACTIVA

Range("G37")

Hace referencia a una HOJA NO ACTIVA

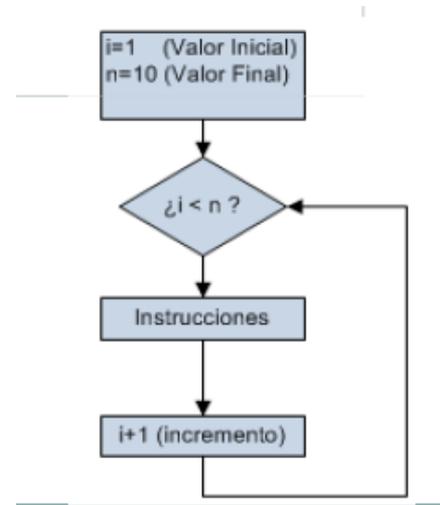
Por ejemplo la celda F4 de la hoja Ex. 1

Worksheets("Ex. 1").Range("F4")

ESTRUCTURA REPETITIVA: FOR (PARA).

Esta estructura sirve para repetir la ejecución de una sentencia o bloque de sentencias, un número definido de veces. La estructura es la siguiente:

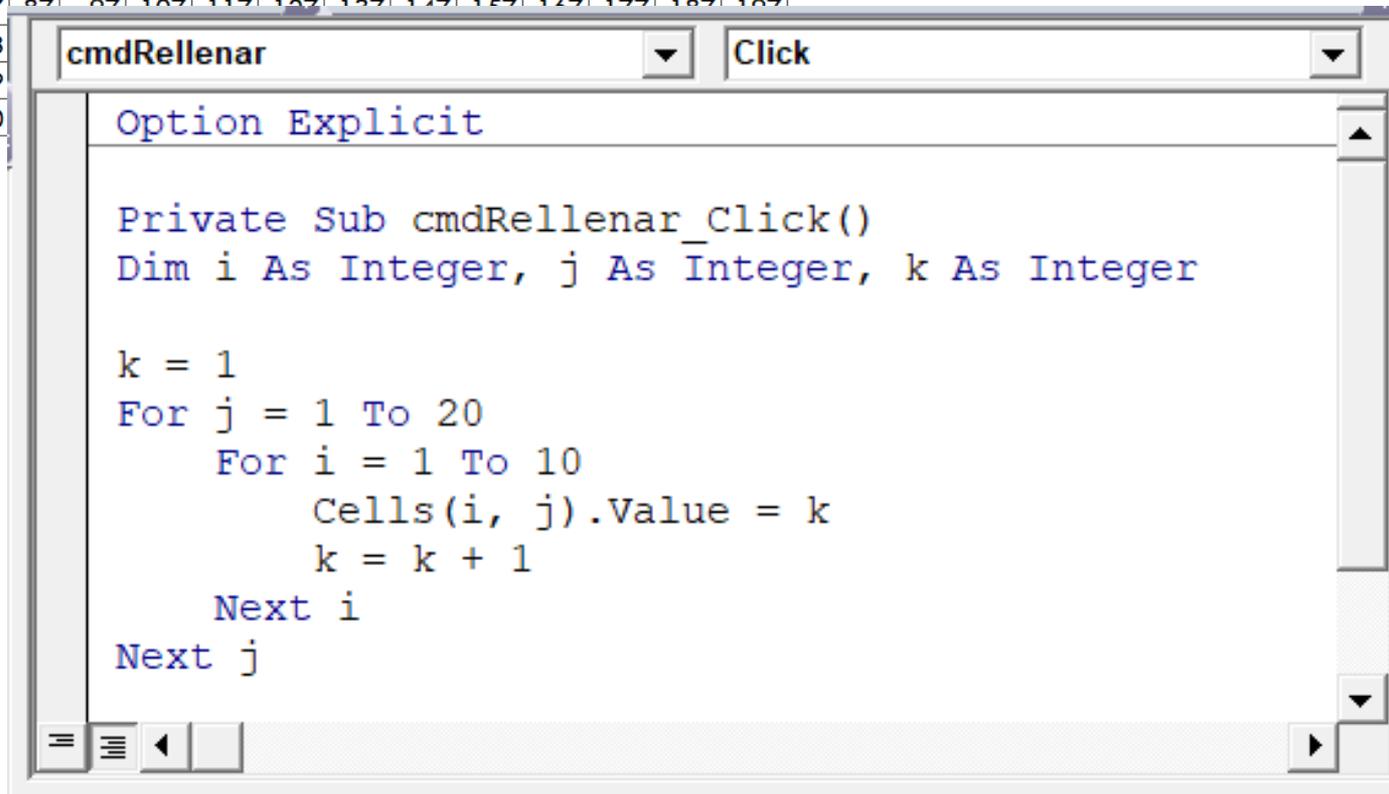
```
For Variable =Valor_Inicial To Valor_Final Step Incremento
  Sentencia 1
  Sentencia 2
  .....
  Sentencia N
Next Variable
```



Variable es una variable que la primera vez que se entra en el bucle se iguala a *Valor_Inicial*, las sentencias del bucle se ejecutan hasta que **Variable** llega al *Valor_Final*, cada vez que se ejecutan el bloque de instrucciones **Variable** se incrementa según el valor de Incremento. Si el incremento es 1, no hace falta poner Step 1.

RESOLUCIÓN EJERCICIO 3: CON ESTRUCTURA FOR...

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	1	11	21	31	41	51	61	71	81	91	101	111	121	131	141	151	161	171	181	191
2	2	12	22	32	42	52	62	72	82	92	102	112	122	132	142	152	162	172	182	192
3	3	13	23	33	43	53	63	73	83	93	103	113	123	133	143	153	163	173	183	193
4	4	14	24	34	44	54	64	74	84	94	104	114	124	134	144	154	164	174	184	194
5	5	15	25	35	45	55	65	75	85	95	105	115	125	135	145	155	165	175	185	195
6	6	16	26	36	46	56	66	76	86	96	106	116	126	136	146	156	166	176	186	196
7	7	17	27	37	47	57	67	77	87	97	107	117	127	137	147	157	167	177	187	197
8	8	18	28	38	48	58	68	78	88	98	108	118	128	138	148	158	168	178	188	198
9	9	19	29	39	49	59	69	79	89	99	109	119	129	139	149	159	169	179	189	199
10	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200
11																				



```
cmdRellenar Click

Option Explicit

Private Sub cmdRellenar_Click()
    Dim i As Integer, j As Integer, k As Integer

    k = 1
    For j = 1 To 20
        For i = 1 To 10
            Cells(i, j).Value = k
            k = k + 1
        Next i
    Next j
End Sub
```


RESOLUCIÓN EJERCICIO 4: CON ESTRUCTURA FOR...

	A	B	C	D	E	F	G	H
1	NOMBRE Y APELLIDO	TELEFONO	ÁREA	SALARIO			Area:	Producción
2	Laura Morales	290999	Producción	\$ 21.419,00				
3	Andrés Castañeda	480050	Recursos Humanos	\$ 28.058,00				
4	Carolina Crisancho	865935	Recursos Humanos	\$ 23.108,00				
5	Luisa Rodriguez	916236	Producción	\$ 24.232,00				
6	José Jimenez	306360	Recursos Humanos	\$ 26.986,00				
7	Diana Parra	680999	Producción	\$ 26.968,00				
8	Juan Perez	730304	Ventas	\$ 23.330,00				
9	Carlos Gomez	839022	Ventas	\$ 34.583,00				

Option Explicit

Private Sub cmdResaltar_Click()

Dim i As Integer

Dim sArea As String

sArea = Range("H1").Value

Range("A2:D19").Font.Bold = False

Range("A2:D19").Interior.ColorIndex = 0

For i = 2 To 19

 If Cells(i, 3).Value = sArea Then

 Range("A" & i & ":" & "D" & i).Font.Bold = True

 Range("A" & i & ":" & "D" & i).Interior.ColorIndex = 6

 End If

Next i

End Sub

ESTRUCTURA REPETITIVA: DO WHILE..LOOP.

DO WHILE..LOOP (HACER MIENTRAS).

- Esta estructura se utiliza, en general, cuando no se sabe previamente el número de veces que se debe repetir un proceso.
- Esta estructura repetitiva está controlada por una o varias condiciones, la repetición del bloque de sentencias dependerá de si se va cumpliendo la condición o condiciones.

Do While (Condición)

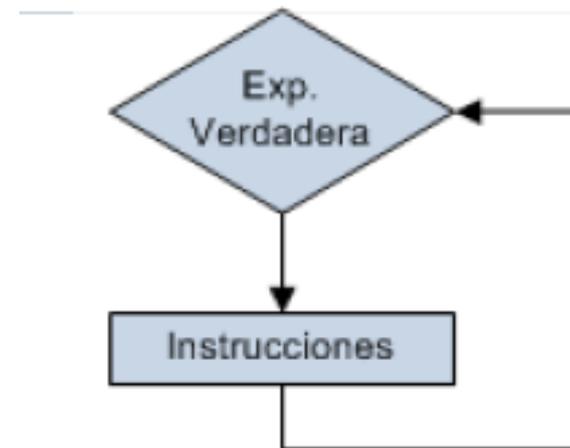
Sentencia1

Sentencia2

....

Sentencia N

Loop



ESTRUCTURA REPETITIVA: DO LOOP UNTIL.

DO LOOP UNTIL (REPETIR HASTA)

- El funcionamiento de esta estructura repetitiva es similar a la anterior salvo que la condición se evalúa al final, la inmediata consecuencia de esto es que las instrucciones del cuerpo del bucle se ejecutaran al menos una vez .

Do

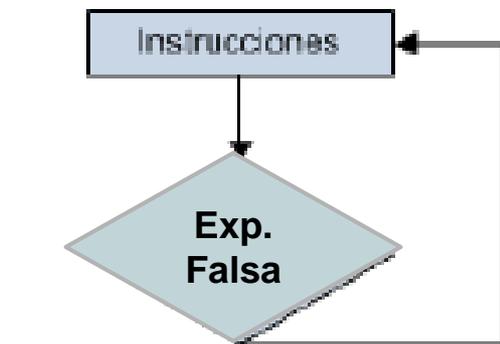
Sentencia1

Sentencia2

....

Sentencia N

Loop Until (Condición)



RESOLUCIÓN EJERCICIO 4: CON ESTRUCTURA DO WHILE

	A	B	C	D	E	F	G	H
1	NOMBRE Y APELLIDO	TELEFONO	ÁREA	SALARIO			Area:	Producción
2	Laura Morales	290999	Producción	\$ 21.419,00				
3	Andrés Castañeda	480050	Recursos Humanos	\$ 28.058,00			Resaltar	
4	Carolina Crisancho	865935	Recursos Humanos	\$ 23.108,00				
5	Luisa Rodfriguez	916236	Producción	\$ 24.232,00				
6	José Jimenez	306360	Recursos Humanos	\$ 26.986,00				
7	Diana Parra	680999	Producción	\$ 26.968,00				
8	Juan Perez	730304	Ve	\$ 22.222,00				
9	Carlos Gomez	839022	Ve	\$ 22.222,00				
10	Yineth Roa	587059	Re	\$ 22.222,00				
11	Camilo Mendez	408125	Re	\$ 22.222,00				
12	Miguel Ceron	748841	Re	\$ 22.222,00				
13	Daniel Rincon	534661	Ve	\$ 22.222,00				
14	Sandra Muñoz	335912	Pr	\$ 22.222,00				
15	Lina Gonzales	242005	Pr	\$ 22.222,00				
16	Jeimmy Neuta	732173	Pr	\$ 22.222,00				
17	Nataly Cuervo	425163	Pr	\$ 22.222,00				
18	Milena Quevedo	462310	Re	\$ 22.222,00				
19	Paty Bernal	246825	Ve	\$ 22.222,00				
20								

Option Explicit

```
Private Sub cmdResaltar_Click()
```

```
Dim i As Integer
```

```
Dim sArea As String
```

```
sArea = Range("H1").Value
```

```
Range("A2:D19").Font.Bold = False
```

```
Range("A2:D19").Interior.ColorIndex = 0
```

```
i = 2
```

```
Do While (i <= 19)
```

```
    If Cells(i, 3).Value = sArea Then
```

```
        Range("A" & i & ":" & "D" & i).Font.Bold = True
```

```
        Range("A" & i & ":" & "D" & i).Interior.ColorIndex = 6
```

```
    End If
```

```
    i = i + 1
```

```
Loop
```

```
End Sub
```

RESOLUCIÓN EJERCICIO 4: CON ESTRUCTURA DO LOOP UNTIL

	A	B	C	D	E	F	G	H
1	NOMBRE Y APELLIDO	TELEFONO	ÁREA	SALARIO			Area:	Producción
2	Laura Morales	290999	Producción	\$ 21.419,00				
3	Andrés Castañeda	480050	Recursos Humanos	\$ 28.058,00			Resaltar	
4	Carolina Cristancho	865935	Recursos Humanos	\$ 23.108,00				
5	Luisa Rodríguez	916236	Producción	\$ 24.232,00				
6	José Jimenez	306360	Recursos Humanos	\$ 26.986,00				
7	Diana Parra	680999	Producción	\$ 26.968,00				
8	Juan Perez	730304	Ventas	\$ 23.330,00				
9	Carlos Gomez	839						
10	Yineth Roa	587						
11	Camilo Mendez	408						
12	Miguel Ceron	748						
13	Daniel Rincon	534						
14	Sandra Muñoz	335						
15	Lina Gonzales	242						
16	Jeimmy Neuta	732						
17	Nataly Cuervo	425						
18	Milena Quevedo	462						
19	Paty Bernal	246						
20								

Option Explicit

```
Private Sub cmdResaltar_Click()
```

```
Dim i As Integer
```

```
Dim sArea As String
```

```
sArea = Range("H1").Value
```

```
Range("A2:D19").Font.Bold = False
```

```
Range("A2:D19").Interior.ColorIndex = 0
```

```
i = 2
```

```
Do
```

```
    If Cells(i, 3).Value = sArea Then
```

```
        Range("A" & i & ":" & "D" & i).Font.Bold = True
```

```
        Range("A" & i & ":" & "D" & i).Interior.ColorIndex = 6
```

```
    End If
```

```
    i = i + 1
```

```
Loop Until (i > 19)
```

```
End Sub
```

PROCEDIMIENTOS Y FUNCIONES.

- Se define como Procedimiento y/o Función a un bloque de código que realiza alguna tarea.
- La filosofía de utilizar Procedimientos y/o Funciones es la antigua fórmula del "divide y vencerás", es decir, con los procedimientos y/o funciones podremos tratar cada problema o tarea de forma más o menos aislada de forma que construiremos el programa paso a paso evitando tener que resolver o controlar múltiples cosas a la vez.
- Los Procedimientos y/o Funciones son también un eficaz mecanismo para evitar la repetición de código en un mismo programa e incluso en diferentes programas.

PROCEDIMIENTOS.

- **Definir un procedimiento.**

Sub Nombre_Procedimiento

Sentencias.

End Sub.

- **Llamar a un procedimiento.**

Para llamar un procedimiento desde otro se utiliza la instrucción **Call** Nom_Procedimiento.

Sub P_Uno

Sentencias.

•

Call P_Dos

•

Sentencias

•

End Sub

Las secuencias del procedimiento P_Uno se ejecutan hasta llegar a la línea Call P_Dos, entonces se salta al procedimiento P_Dos, se ejecutan todas las sentencias de este procedimiento y el programa continua ejecutándose en el procedimiento P_Uno a partir de la sentencia que sigue a Call P_Dos.

FUNCIONES.

- Una función es lo mismo que un procedimiento con la salvedad que este devuelve un valor al procedimiento o función que lo llama.

La estructura general de una Funcion:

Function *Nombre_funcion*(var1 As Tipo, Var2 As Tipo,..., Varn As Tipo) As Tipo.

Invocacion de una funcion:

Nom_variable:= *Nombre_Funcion*(Var1, Var2,..., VarN)

EJERCICIO 5:

Función que devuelve la suma de dos valores que se le pasan como parámetros. Observe las diferentes formas de llamar a la función.

Option Explicit

Sub Ejemplo()

Dim x As Integer

Dim n1 As Integer, n2 As Integer

X = Suma(5, 5)

n1= Val (InputBox("Entrar un número : ", "Entrada"))

n2= Val (InputBox("Entrar otro número : ", "Entrada"))

X= Suma(n1,n2)

ActiveCell.Value = Suma(ActiveSheet.Range("A1").Value , ActiveSheet.Range("A2").Value)

X = Suma(5, 4) + Suma (n1, n2)

End Sub

Function Suma(V1 As Integer, V2 As Integer) As Integer

Dim Total As Integer

Total = V1 + V2

Suma = Total

End Function

EJERCICIO 6: PROPIEDADES ROW Y COLUMN.

Llenar un rango de celdas, empezando por una celda, que se debe especificar desde teclado, con una serie de 10 valores correlativos (comenzando por el 1).

Option Explicit

Sub Ejemplo()

Dim Casilla_Inicial **As String**

Dim i **As Integer**

Dim Fila **As Integer**, Columna **As Integer**

Casilla_Inicial = InputBox("Introducir la casilla Inicial : ", "Casilla Inicial")

ActiveSheet.Range(Casilla_Inicial).Activate

Fila = ActiveCell.Row

Columna = ActiveCell.Column

For i = 1 **To** 10

 ActiveSheet.Cells(Fila, Columna).Value = i

 Fila = Fila + 1

Next i

End Sub

FUNCIONES DE AYUDA

- **Función INT** : Devuelve la parte entera de un número.

Sintaxis

Int(número)

- **Función RND**: Devuelve un numero aleatorio real entre 0 y 1.

Sintaxis

Rnd[(número)]

El argumento opcional número es un tipo Single o cualquier expresión numérica válida.

Para producir enteros aleatorios en un intervalo dado, use esta fórmula:

Int((Límite_superior - límite_inferior + 1) * Rnd() + límite_inferior)

Aquí, **límite_superior** es el número mayor del intervalo y **límite_inferior** es el número menor del intervalo.

VARIABLES IMPORTANTES

- Contador.
- Acumulador.
- Interruptor o Bandera.

VARIABLES IMPORTANTES

Contador: Es una variable cuyo valor aumenta o disminuye en una *cantidad constante* cada vez que se produce un determinado suceso o acción. Por ej. Contar el n° de aprobados.

Operaciones sobre un contador:

1. Inicialización:

$$\text{Contador} \leftarrow \text{Valor_inicial}$$

2. Incremento /decremento:

$$\begin{aligned} \text{Contador} &\leftarrow \text{Contador} + \text{Valor_Constante} \\ \text{Contador} &\leftarrow \text{Contador} - \text{Valor_Constante} \end{aligned}$$

Ej.

Cont = 0

Cont = Cont + 1

VARIABLES IMPORTANTES

Acumulador: es una variable cuyo valor aumenta o disminuye en una *cantidad variable* cada vez que se produce un determinado suceso o acción.

Operaciones sobre un acumulador

1. Inicialización:

$$\text{Acumulador} \leftarrow \text{Valor_inicial}$$

2. Acumulación:

$$\text{Acumulador} \leftarrow \text{Acumulador} + \text{variable}$$
$$\text{Acumulador} \leftarrow \text{Acumulador} * \text{variable}$$
$$\text{Acumulador} \leftarrow \text{Acumulador} - \text{variable}$$
$$\text{Acumulador} \leftarrow \text{Acumulador} / \text{variable}$$

Ej. Se desea acumular 3 valores ingresados; 7, 4, 9

Sum = 0

Sum = Sum + 7 (el valor de Sum queda con el valor 7)

Sum = Sum + 4 (el valor de Sum queda con el valor 11)

Sum = Sum + 9 (el valor de Sum queda con el valor 20)

VARIABLES IMPORTANTES

Interruptor o bandera: es el valor asignado a una variable que hace que se ejecute un bloque de instrucciones.

Por lo general existen dos puntos críticos dentro del algoritmo:

- Cuando se pregunta por el valor de la bandera y
- Cuando se cambia de valor.

Se utiliza para transmitir información de un punto a otro dentro del algoritmo.

Se inicializan en un valor y en donde se considere que sea necesario se cambia su valor al contrario.

Es de mucha utilidad cuando se trabaja con bucles y estructuras de control.

Solo pueden tomar dos valores:

- Lógico: puede ser true o false.

EJERCICIO 7: SEUDOCODIGO

```
bandera = cierto;
contador = 1;
REPITA MIENTRAS (bandera =
cierto)
    Escribir(contador)
    contador = contador + 1
    Si (contador = 10)
        bandera = falso;
    Fin_si
Fin_RM
```

EJERCICIO 7: VBA

Option Explicit

Sub VarImportantes()

Dim Contador As Integer

Dim Bandera As Boolean

Contador = 1

Bandera = True

Do While (Bandera = True)

MsgBox ("Valor de Contador: " & Contador)

Contador = Contador + 1

If Contador = 10 Then

Bandera = False

End If

Loop

End Sub

FUNCIONES DE COMPROBACIÓN

Las **Funciones de Comprobación**, son útiles para validar o comprobar los tipos de datos, bien los insertados en una celda o los introducidos por teclado. Devuelven un valor lógico.

Lista de Funciones de Comprobación.

- **IsNuméric**(Expresión). Comprueba si expresión tiene un valor que se puede interpretar como numérico.
- **IsDate**(Expresión). Comprueba si expresión tiene un valor que se puede interpretar como tipo fecha.
- **IsEmpty**(Expresión). Comprueba que expresión tenga algún valor.
- **IsError**(Expresión). Comprueba si expresión devuelve algún valor de error.
- **WorksheetFunction.IsText**(Expresión). Verifica si el valor es alfanumérico.
- **WorksheetFunction.IsLogical**(Expresión). Verifica si el valor es lógico.