



# Arreglos en VBA-Excel

Pedro Godoy Barrera

1



## Estructuras de datos

- **Definición:** Una **estructura de datos** es una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente.

Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas.

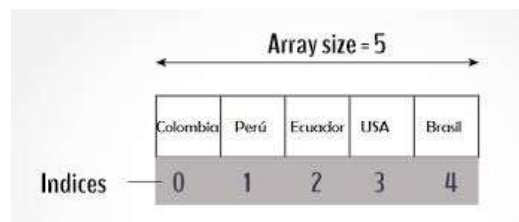
Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de internet.

2

## Definición de Arreglos



- **Definición:** Es un conjunto de valores que están lógicamente relacionados unos con otros. Un grupo de variables o valores que están contenidas en una estructura.
- Para su uso, se hace referencia a estos valores, que tienen el mismo nombre, pero que se diferencian por el uso de un número que se llama índice o subíndice, para obtener un valor específico.
- Los valores individuales son llamados los elementos del arreglo.



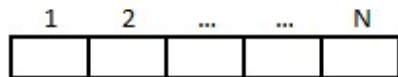
3

## Declaración de Arreglos (1 de 4)



- **Sintaxis:**

```
Dim <Variable>(1 To <N>) As <tipo_de_dato>
```



Se declara una variable con el tipo de dato que quiere que el arreglo contenga (Integer, String, Float, etc.). La diferencia radica en que se le define cuantos elementos tendrá.

En el caso anterior se observa que hay **N elementos** en total

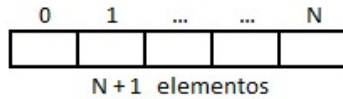
4

## Declaración de Arreglos (2 de 4)



### • Sintaxis:

Dim <Variable>(<N>) As <tipo\_de\_dato>



- Para este caso se tiene que:
  - El primer índice **parte en 0**
  - **Hay N + 1** elementos en total, pues parte enumerando desde 0.

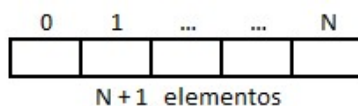
5

## Declaración de Arreglos (3 de 4)



### • Sintaxis:

Dim <Variable>(0 To <N>) As <tipo\_de\_dato>



- Funciona igual que la declaración anterior.

6

## Declaración de Arreglos (4 de 4)



Un vector es un arreglo de **una dimensión**, lo que es equivalente a una lista de elemento. La forma explicada de definir un array previamente, es lo que se considera un vector.

### Sintaxis:

```
Dim VectorNumeros As Variant
VectorNumeros = Array(1,3,5)
```

|               | 0 | 1 | 2 |
|---------------|---|---|---|
| VectorNumeros | 1 | 3 | 5 |

Esto declara un tipo de dato por defecto, el cual es el **Variant**. Si no se declaró el tipo de dato, por defecto, el intérprete asume que la variable es de tipo **Variant**, el cual es equivalente a cualquier tipo existente.

Para trabajar con **Array()**, se **necesita** que sea de tipo **Variant**.

7

## Agregar valor a un arreglo



### • Sintaxis:

<Variable>( <índice> ) = <valor\_segun\_tipo\_de\_dato>

### • Ejemplo:

```
Dim x(1 to 3) As Double
```

```
x(1) = 2.5
```

```
x(2) = 3.7
```

```
x(3) = 6.3
```

|   | 1   | 2   | 3   |
|---|-----|-----|-----|
| x |     |     |     |
| x | 2.5 |     |     |
| x | 2.5 | 3.7 |     |
| x | 2.5 | 3.7 | 6.3 |

8

## Modificar un valor en un arreglo



- **Sintaxis:**

`<Variable>( <índice> ) = <nuevo_valor_segun_tipo_de_dato>`

- **Ejemplo:**

Dim x(1 to 3) As Double

x(1) = 2.5

x(2) = 3.7

x(3) = 6.3

x(2) = 4.1

|   |     |     |     |
|---|-----|-----|-----|
| x | 1   | 2   | 3   |
|   |     |     |     |
| x | 1   | 2   | 3   |
|   | 2.5 |     |     |
| x | 1   | 2   | 3   |
|   | 2.5 | 3.7 |     |
| x | 1   | 2   | 3   |
|   | 2.5 | 3.7 | 6.3 |
| x | 1   | 2   | 3   |
|   | 2.5 | 4.1 | 6.3 |

9

## Recorrer los valores de un arreglo



- Para obtener los valores de un arreglo se debe considerar una estructura repetitiva que permita generar las posiciones de los elementos que se quieren obtener, o en su defecto, usar una estructura que obtenga directamente los elementos del arreglo (por ej. For each)

- For
- Do While
- For each

10

## Recorrer los valores de un arreglo (1 de 3)



### Sintaxis:

```
For <variable> = <valor_inicial> To <valor superior>
  ' instrucciones a realizar
Next <variable>
```

Si sabemos las dimensiones del arreglo:

### Sub recorrer()

....

```
For i = 1 to 3
  MsgBox( x(i))
Next i
```

End Sub

Si **No** sabemos las dimensiones del arreglo:

### Sub recorrer()

....

```
Inf = Lbound(x) ' Entrega el índice menor del arreglo
Sup = Ubound(x) 'Entrega el índice mayor del arreglo
```

```
For i = inf to sup
  MsgBox( x(i))
Next i
```

End Sub

11

## Recorrer los valores de un arreglo (2 de 3)



### Sintaxis:

```
<variable> = <límite inferior>
Do while <variable> <= <límite superior>
  ' instrucciones a realizar
  variable = variable + 1
Loop
```

Si sabemos las dimensiones del arreglo:

### Sub recorrer()

....

```
i = 1
Do while i <= 3
  MsgBox( x(i))
  i = i+1
Loop
```

End Sub

Si **No** sabemos las dimensiones del arreglo:

### Sub recorrer()

....

```
inf = Lbound(x) ' Entrega el índice menor del arreglo
sup = Ubound(x) 'Entrega el índice mayor del arreglo
```

```
Do While inf <= sup
  MsgBox( x(inf))
  inf = inf + 1
```

Loop

End Sub

12

## Recorrer los valores de un arreglo (3 de 3)



### Sintaxis:

```
For each <variable> IN Arreglo ' Variable debe ser del tipo Variant
    ' instrucciones a realizar
Next
```

Si se usa esta estructura, da lo mismo si **se sabe o no** el valor del índice inicial y final, pues se obtienen los elementos del arreglo directamente, sin el uso de su índice.

### Sub recorrer()

....

```
For each y in x
    MsgBox( y)
Next
```

### End Sub

13

## Ejemplo de uso de arreglos



El siguiente ejemplo generará una tabla con valores que están en un array. Note cómo se utiliza el For ... Next para referenciar las posiciones.

```
Sub agregarPaises()
    Dim Paises(1 To 5) As String

    Paises(1) = "Argentina"
    Paises(2) = "Colombia"
    Paises(3) = "España"
    Paises(4) = "México"
    Paises(5) = "Perú"

    Worksheets("Ej1").Activate
    Range("A1").Value = "Arreglo"
    Range("A1:B1").Select
    With Selection
        .HorizontalAlignment = xlCenter
        .MergeCells = True
    End With
```

```
Range("A2").Select
Range("A2").Value = "Indice"
Range("B2").Select
Range("B2").Value = "Valor"

For i = 1 To 5
    Range("A" & (i + 2)).Value = i
    Range("B" & (i + 2)).Value = Paises(i)
Next i
End Sub
```



|   | A       | B         |
|---|---------|-----------|
| 1 | Arreglo |           |
| 2 | Indice  | Valor     |
| 3 | 1       | Argentina |
| 4 | 2       | Colombia  |
| 5 | 3       | España    |
| 6 | 4       | México    |
| 7 | 5       | Perú      |
| 8 |         |           |

14

## Modificar tamaño de un arreglo



- Por definición los arreglos son de tamaño fijo, pero únicamente, los arreglos de una dimensión, pueden ser modificados en tamaño. Para lograr lo anterior, se puede usar el operador REDIM.

```
Dim MiArreglo() As Integer
' cuando está definido así "()" es porque aún no se le define un tamaño
Redim MiArreglo(5)           ' se asigna un espacio para 5
For i=1 To 5                 ' se inicializa el arreglo
    MiArreglo(i) = i        ' se asignan valores al arreglo
Next i

Redim MiArreglo(10)         ' redimensiona a 10 y OJO: se BORRAN los elementos existentes
For i=1 To 10               ' se inicializa el arreglo con 10 elementos
    MiArreglo(i) = i
Next i
Redim Preserve MiArreglo(15) ' si no se quiere borrar los elementos, se usa la
                             ' palabra "Preserve"
```

15



# ¡Muchas gracias!

Cualquier consulta realizarla a:  
[Pedro.Godoy@usm.cl](mailto:Pedro.Godoy@usm.cl)

16