

## ARREGLOS

En visual Basic un arreglo se **declara** con la siguiente sintaxis:

```
Dim nombre_arreglo As tipo de datos()
```

El nombre\_arreglo es el nombre de la variable que deseas darle y el tipo de datos del arreglo son: enteros, caracteres, flotantes, etcétera, los paréntesis después del tipo de datos indican que se está declarando un arreglo.

Para **crear** el arreglo se debe de usar la siguiente sintaxis:

```
nombre_arreglo = New tipo de datos(tamaño) {}
```

El nombre\_arreglo será el nombre del arreglo previamente declarado con la instrucción Dim, el tipo de datos deberá de ser el mismo anteriormente usado, el tamaño es el número de elementos que el arreglo almacenará y en las llaves, en forma opcional, se pueden escribir los valores separados por comas que contendrá el arreglo.

Es posible declarar el arreglo y crearlo de la siguiente forma:

```
Dim nombre_arreglo = New tipo de datos(tamaño) {}
```

o también,

```
Dim nombre_arreglo = New tipo de datos() {valor_1,valor_2,valor_3, valor_n}
```

Veamos un ejemplo para crear un arreglo de 5 números enteros y luego ordenarlos para mostrarlos en pantalla:

```
Module Module1

    Sub Main()
        Dim x, y, aux As Integer
        Dim a As Integer() 'declaración del arreglo
        a = New Integer(5) {} 'creación del arreglo con tamaño de 5
        elementos

        'carga del arreglo
        For x = 0 To 4 'el rango es de 0 a 4 para guardar 5 valores
            Console.WriteLine("Ingrese número {0}: ", x + 1)
            a(x) = Console.ReadLine()
        Next

        'ordenación del arreglo por el método de la burbuja
        For x = 0 To 3 'el índice "x" nos va a servir para acceder a la
        posición actual
            For y = x + 1 To 4 'el índice "y" se va a utilizar para las
            siguientes posiciones a la actual
```

```
        If (a(x) > a(y)) Then
            aux = a(x)
            a(x) = a(y)
            a(y) = aux
        End If
    Next
Next

'impresión del arreglo
Console.WriteLine()
Console.WriteLine("+++++")
Console.WriteLine("El arreglo ordenado es: ")
For x = 0 To 4
    Console.WriteLine("{0}", a(x))
Next
Console.WriteLine("+++++")
Console.Read()
End Sub

End Module
```

Probemos ahora con declarar el arreglo y crearlo con valores iniciales:

```
Module Module1

    Sub Main()
        Dim x As Integer
        Dim a As Integer() = New Integer() {6, 9, 2, 4, 1}

        Console.WriteLine("Tus números son:")
        For x = 0 To 4
            Console.WriteLine("{0}", a(x))
        Next
        Console.Read()

    End Sub

End Module
```

Como podrás observar en la línea:

```
Dim a As Integer() = New Integer() {6, 9, 2, 4, 1}
```

Se declara e inicializa el arreglo con 5 valores, los cuáles se pueden manipular posteriormente en el programa.

Hasta aquí, he mostrado el trabajo de arreglos unidimensionales, es decir con una sólo columna de datos, pero en el mundo real, hay varias aplicaciones que requieren arreglos bidimensionales o multidimensionales, ¿como se declaran estas estructuras?, para comenzar veamos la siguiente sintaxis:

```
Dim a as integer(,) = New integer(,) {{1,2,3},{4,5,6}}
```

Los valores toman las posiciones del arreglo que se muestran a continuación:

	Columna 0	Columna 1	Columna 2
Fila 0	1 (0,0)	2 (0,1)	3 (0,2)
Fila 1	4 (1,0)	5 (1,1)	6 (1,2)

Nota: las coordenadas entre paréntesis representan la posición dentro del arreglo.

Veamos su implementación:

```
Module Module1

    Sub Main()
        Dim x, y As Integer
        Dim a As Integer(,) = New Integer(,) {{1, 2, 3}, {4, 5, 6}}
        Console.WriteLine(vbTab & "+++++")
        Console.WriteLine(vbTab & "La impresión del arreglo es:")
        For x = 0 To 1 'el índice "x" representa las filas
            For y = 0 To 2 'el índice "y" representa las columnas
                Console.Write(vbTab & "{0}" & vbTab, a(x, y))
            Next
            Console.WriteLine()
        Next
        Console.WriteLine(vbTab & "+++++")
        Console.Read()
    End Sub

End Module
```

Para reforzar el tema de arreglos te propongo otro ejemplo en el que dando una lista de números enteros el programa realiza todas las posibles combinaciones entre ellos de 3 en 3, como por ejemplo: con los números 1,2, 3 y 4 se forman las combinaciones: 1-2-3, 1-2-4, 1-3-2, 1-3-4, 1-4-2, 1-4-3, 2-1-3, 2-1-4,..., etcétera. Analiza la siguiente codificación:

```
Module Module1

    Sub Main()
        Dim x, y, z As Integer
        Dim c As Integer = 0
        Dim a As Integer() = New Integer(4) {}
        'carga de datos al arreglo
        For x = 0 To 3
            Console.Write("Ingrese el número {0}: ", x + 1)
            a(x) = Console.ReadLine()
        Next
        Console.WriteLine("+++++")
        Console.WriteLine("Las combinaciones son: ")
        For x = 0 To 3
            For y = 0 To 3
                If (x <> y) Then
                    For z = 0 To 3
                        If (x <> z And y <> z) Then
                            Console.WriteLine(a(x) & "-" & a(y) & "-" & a(z))
                            c = c + 1
                        End If
                    Next
                End If
            Next
        Next
        Console.WriteLine("Se formaron " & c & " combinaciones.")
    End Sub

End Module
```

```
        Console.WriteLine("    | {0} | {1} |  
{2} |", a(x), a(y), a(z))      Console.WriteLine(" +-----+-----+-----  
-+")  
                                c += 1  
                                End If  
                            Next  
                        End If  
                    Next  
                Next  
            Console.WriteLine("En total se formaron {0} combinaciones.", c)  
            Console.Read()  
        End Sub  
  
End Module
```

Espero que esta información te sea de gran utilidad.

Atentamente

Prof. Gilberto Manzano Peñaloza

Director técnico

Apuntes de lenguajes de de programación orientados a objetos.