

## Practica N°7 – Arreglos (Arreglos en paralelo)

Semana del 08 al 13 de Junio del 2015

**Objetivo de la Práctica:** Desarrollar una aplicación tipo consola en VB2010, en donde se utilicen datos tipo arreglos unidimensionales y/o bidimensionales, los cuales comparten los datos de un registro, con el propósito de determinar cálculos del tipo estadísticos (promedios, porcentajes, máximos y mínimos entre otros).

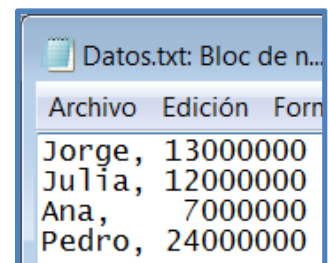
### Introducción

Como se empezó a ver en este corte, observamos que los lenguajes de programación cuentan con la posibilidad de crear estructuras de datos capaces de almacenar una cantidad finita de datos del mismo tipo en una sola variable, conociéndose este tipo de estructura como arreglo o matrices para el caso de Visual Basic.

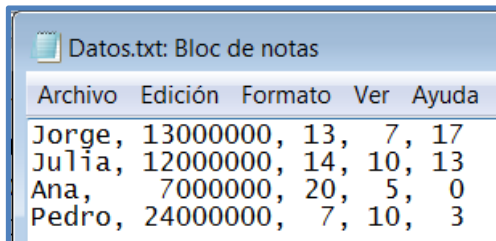
Los arreglos pueden crearse de uno a más dimensiones, haciendo las veces de vectores, matrices o estructura de  $R^3$  en adelante. Sin embargo, cuando la información a manejar posee más de un concepto o dato, con un solo arreglo no se logra almacenar todos los datos del registro, sino que se requiere construir un arreglo distinto por cada concepto a manejar o procesar.

Por otro lado, estos arreglos que almacenan los datos de cada concepto a procesar, tiene una relación con respecto a los índices que ellos manejan. Siendo este índice el que relaciona los datos con el registro a procesar, Ej. la lista de clase de un curso cualquiera generalmente manejan los siguientes datos, número de la lista, la cédula de identidad y finalmente el apellido y nombre del estudiante, es decir que por cada estudiante presente se le registra tres conceptos distintos, Número de la lista, Cédula de identidad y su Apellidos y nombre.

Interpretemos el número de la lista como el índice que asocia la cedula de identidad del estudiante y el campo del Apellido y Nombre del mismo, esto significa que si quiero grabar en memoria el registro de todos los estudiantes del curso se requiere construir dos arreglos unidimensionales, uno que guarde las cédulas de identidad (Tipo entero "**Integer**") y otro que almacene los datos de los apellidos y nombres (Tipo cadena de caracteres "**String**"). Esto significa que si yo deseo conocer la información del estudiante número tres(3) de la lista, tengo que acceder tanto en el vector que almacena las cédulas de identidad y el que almacena los apellidos y nombres en la posición o índice tres(3).



Archivo	Edición	Form
Jorge,	13000000	
Julia,	12000000	
Ana,	7000000	
Pedro,	24000000	



Archivo	Edición	Formato	Ver	Ayuda
Jorge,	13000000,	13,	7,	17
Julia,	12000000,	14,	10,	13
Ana,	7000000,	20,	5,	0
Pedro,	24000000,	7,	10,	3

Finalmente, no siempre los arreglos tienen que poseer la misma cantidad de dimensiones, se puede dar el caso que los arreglos sean de distintas dimensiones, así como pueden ser de distintos tipos de datos, continuando con el ejemplo anterior, pensemos que además de los datos de las cédulas de identidad y los apellidos y nombres de los estudiantes, se tiene además las notas que estos obtuvieron en tres evaluación realizadas. Esto quiere decir que se requiere implementar otro arreglo pero este seria de dos dimensiones, siendo las filas el índice que asocia a que estudiante se está

procesando y las columnas indican cual evaluación se procesa. Ej. si esta matriz se llama Notas, al indicar Notas(2,3), estaremos procesando al estudiante Nro 2 en su evaluación Nro 3.

**En conclusión, como se pudo observar se dice que dos o más arreglos se comportan en paralelo cuando al menos un índice está asociado a la información del registro de datos que se quiere manejar. Donde estos arreglos no tienen que ser necesariamente del mismo tipo ni tampoco del mismo número de dimensiones.**

En caso de no querer manejar arreglos en paralelos, una forma indirecta de hacerlo es creando un tipo de dato estructura que almacene en cada campo la información que se desea manejar y posteriormente crear un vector de estructura, Ej. en la última situación planteada se puede crear la estructura estudiante la cual puede tener alguna de las siguientes dos definiciones:

```
Structure Estudiante
    Dim Nom As String
    Dim CI As Integer
    Dim N1, N2, N3 As Integer
End Structure
```

```
Structure Estudiante
    Dim Nom As String
    Dim CI As Integer
    Dim Notas() As Integer
End Structure
```

Lo que implica que en vez ahora de manejar tres arreglos como describimos anteriormente se puede crear un solo vector tipo Estudiante, sin importar la definición que escogamos

Dim V(30) As Estudiante

Recordando que en este caso al acceder a un índice del arreglo estamos manejando un dato tipo Estudiantes y para hacer acceso a los datos dentro de él, significa entrar en los campos internos del mismo. Ej, si se desea acceder al nombre del

estudiante ubicado en la posición N°3 del arreglo V, la forma de acceso sería, sin importar cual se las dos definiciones de estudiante se emplee:

**V(3).Nom**

Ahora, hay algo muy importante a tener en cuenta en caso de que un campo de la estructura sea un dato tipo arreglo, y es que esta la primera vez que se requiera implementar se le debe redimensionar, para que en memoria se reserven las posiciones a ser empleadas en él, es decir: Si se quiera desarrollar el subprograma leer, el cual lee el archivo de datos "Datos.Txt" que se mostró anteriormente, el código sería para ambas definiciones del dato Estudiante la siguiente:

```
Sub Leer(ByRef V() As Estudiante,
        ByRef Nest As Integer)
    FileOpen(1, "..\..\..\Datos.Txt", OpenMode.Input)
    Nest = 0
    While Not Eof(1)
        Nest += 1
        Input(1, V(Nest).Nom)
        Input(1, V(Nest).CI)
        Input(1, V(Nest).N1)
        Input(1, V(Nest).N2)
        Input(1, V(Nest).N3)
    End While
    FileClose(1)
End Sub
```

```
Sub Leer(ByRef V() As Estudiante,
        ByRef Nest As Integer,
        ByVal Nnotas As Integer)
    FileOpen(1, "..\..\..\Datos.Txt", OpenMode.Input)
    Nest = 0
    While Not Eof(1)
        Nest += 1
        Input(1, V(Nest).Nom)
        Input(1, V(Nest).CI)
        Redim V(Nest).Notas(Nnotas) ' Solo la primera vez
        For I=1 to Nest
            Input(1, V(Nest).Notas(I))
        Next
    End While
    FileClose(1)
End Sub
```

Finalmente, si se quisiera hacer acceso a la nota que obtuvo el segundo estudiante en la evaluación N°3, la forma de acceder sería la siguiente para cada forma de definición del dato Estudiante:

**V(2).N3**

**V(2).Notas(3)**

### Escenario N° 1: Geometría Vectorial

Desarrolle los siguientes subprogramas:

- 1- Dado los vectores **X** e **Y** de **L** elementos cada uno tipo **single**, elabore un subprograma que determine el producto escalar de los vectores sabiendo que:

$$\vec{X} * \vec{Y} = x_1 * y_1 + x_2 * y_2 + x_3 * y_3 + x_4 * y_4 + \dots + x_L * y_L = \sum_{i=1}^L x_i * y_i$$

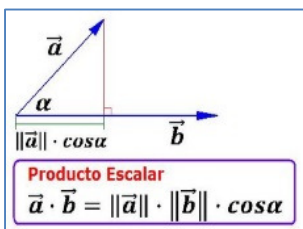
- 2- Dados el valor entero **K**, lea por teclado el vector **P** de **K** elementos tipo **single**.
- 3- Dado los vectores **A** y **B** de **N** elementos tipo **single** cada uno, determine el coseno del ángulo formado. Sabiendo que:

$$\vec{A} * \vec{B} = \|\vec{A}\| * \|\vec{B}\| * \cos(\alpha)$$

Recuerde que el módulo de un vector (V) de tv elementos se determina según la siguiente expresión:

$$\|\vec{V}\| = \sqrt{\sum_{i=1}^{tv} v_i^2}$$

### ENUNCIADO



Desarrolle un programa en VB2010 que haciendo uso de los subprogramas solicitados y aquellos que crea usted necesitar, lee por teclado el tamaño (**N**) de los vectores **V1** y **V2** paralelos entre si con sus respectivos valores y determine e imprima por pantalla si **V1** es perpendicular o no con **V2**, en caso de no ser perpendiculares determine el coseno del ángulo que entre ellos se forma.

**NOTA:** recuerde que dos vectores son perpendiculares entre sí, siempre y cuando se cumple que su producto escalar es igual a cero.

### Ejemplo de entrada y salida del programa planteado

Datos de entrada

```
N= 5
V1= 2.46  4.34  5.76  14.54  22.46
V2= 1.14  10.28  10.72  13.00  19.62
```

Salida por pantalla

```
Los vectores no son perpendiculares
cos(alpha) = 0.9535
```

## Escenario N° 2: Aritmética de matrices

- 1- Desarrolle un subprograma que lea del archivo **Narch** (Número del archivo), en la primera línea la cantidad de filas y columnas de una matriz, y luego en líneas separadas los valores tipo **single** de cada fila de la misma.
- 2- Desarrolle un subprograma que dada dos matriz **A** y **B** de **NxM** elementos tipo **single**, devuelva la matriz **C** que representa el resultado de la suma de **A+B**.

$$\begin{bmatrix} 1 & 2 & 2 \\ 1 & 0 & 3 \\ 3 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 4 \\ 2 & 4 & 1 \\ 3 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 1+0 & 2+1 & 2+4 \\ 1+2 & 0+4 & 3+1 \\ 3+3 & 2+1 & 2+4 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 6 \\ 3 & 4 & 4 \\ 6 & 3 & 6 \end{bmatrix}$$

- 3- Desarrolle un subprograma que dada la matriz (**M[fM, cM]**) elementos tipo **single**, y un valor escalar **k**, devuelva la matriz **M** como el resultado del producto del escalar **k \* M**
- 4- Desarrolle un subprograma que dada la matriz (**M[fM, cM]**) elementos tipo **single**, y la posición de una fila **f**, retorne el vector **V** el cual contenga todos los elementos contenidos en la fila **f** de la matriz.
- 5- Desarrolle un subprograma que dada la matriz (**M[fM, cM]**) elementos tipo **single**, y la posición de una columna **c**, retorne el vector **V** el cual contenga todos los elementos contenidos en la columna **c** de la matriz.
- 6- Desarrolle un subprograma que dada dos matriz **A[fA, cA]** y **B[fB, cB]** elementos tipo **single** cada uno, devuelva la matriz **C** que representa la multiplicación de **A\*B**. Para ello determine la multiplicación de las dos matrices empleando la siguiente metodología:  
El resultado del elemento de las matriz **C(i, j)** es igual al producto escalar de la fila **i** de la matriz **A** por la columna **j** de la matriz **B**. Ej.

### Producto escalar

$$A = \begin{bmatrix} 1 & 8 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 5 \\ -1 \\ 4 \end{bmatrix}, AB = \begin{bmatrix} 1 & 8 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ -1 \\ 4 \end{bmatrix} = (1)(5) + (8)(-1) + (0)(4)$$

$$AB = 5 - 8 + 0 = -3$$

### Multiplicación de matrices

$$P = QW = \begin{bmatrix} 1 & 3 & 5 \\ 0 & 0 & 1 \\ 4 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 1 & 3 & 4 & 0 \\ 0 & -8 & 9 & 1 \\ 7 & 5 & 5 & 1 \end{bmatrix}$$

$$P_{(2,3)} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 9 \\ 5 \end{bmatrix} = 5$$

$$P = \begin{bmatrix} 36 & 4 & 56 & 8 \\ 7 & 5 & 5 & 1 \\ 18 & 14 & 35 & 3 \end{bmatrix}$$

## ENUNCIADO

Desarrolle un programa en VB2010 que haciendo uso de los subprogramas antes solicitados y aquellos otros que crea usted necesitar, lea del archivo de datos **"Matriz.Txt"**, la información de dos matriz (Ver ejemplo del archivo), luego muestre por pantalla los datos de las dos matrices leídas, para finalmente solicitarle al usuario por teclado que operación desea realizar con las dos matrices. Donde el tipo de operación a realizar se indica como: **1** = sumar A+B, **2** = Restar A-B, o **3**= Multiplicar A\*B.

Recuerde que para poder realizar la suma o resta de dos matrices debe cumplirse que el tamaño de las matrices tienen que ser iguales y si la operación es de multiplicación, el nro de columnas de **A(cA)** deben ser igual al nro de filas de **B(fB)**, donde la matriz resultante su tamaño será de **C[fA, cB]**

Matriz.Txt

3,	3
14.85,	3.68, 0.61
4.32,	14.84, 1.09
10.27,	1.07, 7.35
2,	2
12.58,	9.01
4.43,	4.62

### Ejercicio Nro 3: Competencia de Patinaje artístico

Durante una competencia de patinaje artístico, se registra por cada participante las calificaciones de los jurados del comité evaluador. Con el objeto de asegurar un resultado sin parcialidad alguna, se decide que para el puntaje total se deben promediar las calificaciones obtenidas eliminando la más alta y más baja de las emitidas por los jurados.

Por ejemplo:

Nombre del Competidor	Calificaciones de jurado						Calculo del puntaje
Juan García	9	9	8	6	8	10	$(50 - 6 - 10) / 4$
Miguel Martínez	6	7	7	7	6	6	$(39 - 7 - 6) / 4$
Carlos Guillen	9	9	9	9	9	9	$(54 - 9 - 9) / 4$



#### ENUNCIADO

Dado el archivo de datos “**Calificacion.Txt**” en el cual se almacenó en la primera línea el número de jurados a evaluar (**N**) y luego en líneas siguiente para cada participante se colocó: el nombre del participante y el conjunto de calificaciones obtenidas por los **N** jueces del comité evaluador, considerando que no se conoce la cantidad de competidores, desarrolle un programa en VB2010 que determine e imprima en el archivo “**PuntajeFinal.Txt**”, la siguiente información para cada participante, nombre, las calificación dadas por el jurado y la puntuación final obtenida.

#### Ejemplo de entrada y salida del programa planteado

Calificacion.Txt							PuntajeFinal.Txt			
6							Nombre	Calificaciones		
Juan García,	9,	9,	8,	6,	8,	10	Juan García	9	9	8
Miguel Martínez,	6,	7,	7,	7,	6,	6	Miguel Martínez	6	7	7
Carlos Guillen,	9,	9,	9,	9,	9,	9	Carlos Guillen	9	9	9

#### Requerimientos:

Para la solución del problema debe definir y utilizar:

1. Un subprograma que lea el archivo de datos “**Calificacion.Txt**”, el vector (**nom**) que almacene ? elementos tipo **String** paralelo con las filas de la matriz (**m**[?, **CM**]) de elementos tipo **Integer**, con las calificaciones del jurado.
2. Un subprograma que sume de los elementos de la fila **f** de una matriz (**m**[**fM**, **CM**]) de elementos tipo **Integer**.
3. Un subprograma que retorne la posición del elemento menor en la fila **f** de la matriz (**m**[**fM**, **CM**]) de elementos que contiene datos tipo **Integer**.
4. Un subprograma que retorne la posición del elemento mayor en la fila **f** de la matriz (**m**[**fM**, **CM**]) de elementos que contiene datos tipo **Integer**.
5. Un subprograma que retorne un vector (**puntaje**) de elementos tipo **Single**, paralelo a las filas de la matriz (**m**[**fM**, **CM**]) elementos tipo **Integer**, que contiene las calificaciones registradas por el jurado según las especificaciones del ejercicio.
6. Un subprograma que imprima el vector (**nom**) de elementos tipo **String** en paralelo con la una matriz (**m**[**fM**, **CM**]) de elementos tipo **Integer** que contiene las calificaciones registradas por el comité evaluador y por último el vector (**puntaje**) de elementos tipo **Single**.