



UNIDAD CUATRO ARREGLOS

1. Arreglos
 - 1.2 Partes de un arreglo
 - 1.3 Tipos de arreglos
 - 1.3.1 Arreglos unidimensionales
 - 1.3.2 Arreglos bidimensionales
 - 1.3.3 Arreglos multidimensionales
2. Ejercicios con arreglos
3. Registros
 - 3.1 Definición de registros
 - 3.2 Diferencias con arreglos



1. Arreglos

Los arreglos son una forma diferente de almacenar y organizar información. Un arreglo es una estructura de datos utilizada para almacenar un conjunto de datos del mismo tipo. Es una colección o conjunto de elementos finito, ordenado y homogéneo.

- Finito: porque tiene un límite es decir, debe determinarse cual será el número máximo de elementos que podría contener.
- Ordenado: se puede determinar cual es el primer elemento, el segundo, el tercero hasta el último elemento.
- Homogéneo: todos los elementos de un arreglo son del mismo tipo (todos enteros, todos char etc)

1.2 Partes de un arreglo

Un arreglo básicamente está dividido en: Componentes, subíndice o subíndices.

- Componentes: hace referencia a los elementos que conforma el arreglo, es decir a los valores que se almacenan en cada una de las casillas o campos del mismo.
- Subíndice: es una variable, constante o expresión que debe dar como valor un número entero ordinal, es decir un valor que tenga claramente definido un sucesor y un antecesor, por lo tanto los reales no pueden ser subíndice.

El subíndice indica la posición en la que debe dejar un elemento en el arreglo, o la posición o la posición en la que debe buscar un elemento en el arreglo.

Para nombrar o hacer referencia a un arreglo se debe siempre escribir el nombre del arreglo y el subíndice o los subíndices respectivos ejemplo: VEC1[K], VEC1[2], VEC1[C+1]

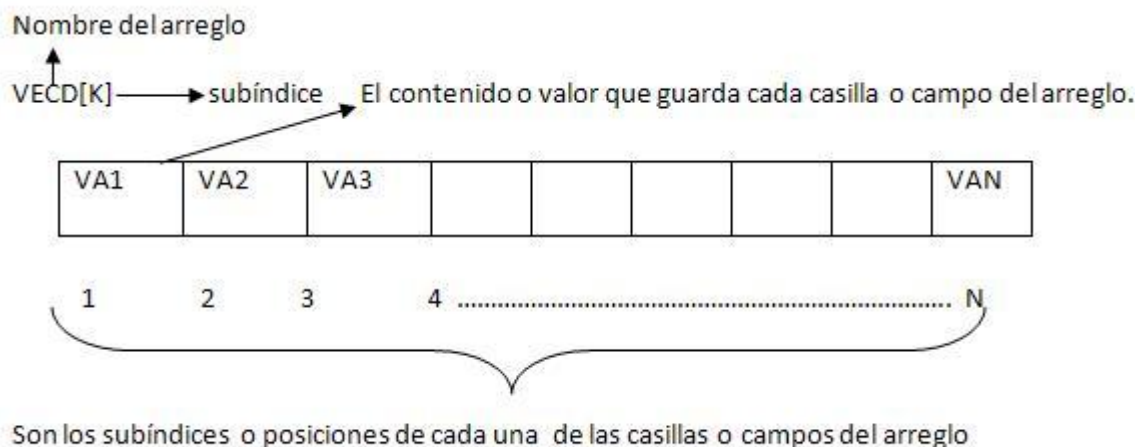


Imagen1. Arreglo

- El subíndice puede ser una variable de tipo entero o enumerado, pero que se haya definido antes.
- El subíndice va saltando o posicionándose, en cada casilla el arreglo.
- El contenido o elemento de los campos de un arreglo puede ser: numérico, carácter, cadena de caracteres etc.
- El índice o subíndice de un arreglo se encierra entre corchetes cuadrados []
- Siempre que se haga referencia a un arreglo se debe escribir su nombre y entre paréntesis su subíndice.

1.3 Tipos de arreglos

- Arreglos Unidimensionales, llamados también vectores
- Arreglos bidimensionales, también llamados matrices
- Otros: tridimensionales, poli dimensionales.



1.3.1 Arreglos unidimensionales

Definición: es una colección o conjunto de datos o elementos ordenados en forma de lista y que se conoce también con el nombre de vector.

Características: Es un arreglo de una sola dimensión (unidimensional)

Ejemplo VECN

80	12	45	8	19
----	----	----	---	----

imagen2. Vector unidimensional

Este vector posee 5 casillas o campos y en cada posición guarda un valor, en este caso un número entero; por ejemplo cuando R vale 3, el vector es igual a 45, $VECN[3] = 45$, es decir el vector en la posición 3 posee o guarda el valor de 45.

1.3.2 Arreglos bidimensionales

Definición: es una colección o conjunto de elementos o valores dispuestos en forma de filas y columnas, formando una matriz.

Características:

- Es un arreglo de dos dimensiones (bidimensional)
- Necesita de dos subíndices para el posicionamiento de la información, lo mismo que para su localización o búsqueda. El primer subíndice busca las posiciones de las filas, el segundo busca las posiciones de las columnas.



Ejemplo

80	12	45	8	19
----	----	----	---	----

Nombre de la matriz. Esta matriz posee 4 filas y 3 columnas

Subíndice de las filas.
MAT1 [K, J] → subíndice de las columnas.

23	78	100
67	99	45
1	74	325
22	10	20

imagen3. Vector bidimensional

Esta es una matriz de 4 filas y 3 columnas, en total posee 12 casillas o campos.

MAT1 [3,2] = 74

- El arreglo bidimensional o de dos dimensiones, está formado por un conjunto de elementos de un mismo tipo de datos que se almacenan bajo un mismo nombre y que al igual que en el unidimensional, se diferencian por la posición que tiene cada elemento dentro del arreglo de datos, con la aclaración de que la disposición de los elementos es en forma rectangular o cuadrada, donde la primera dimensión está dada por los renglones y la segunda por las columnas.
- Un arreglo de este tipo, también conocido como matriz, es del orden $M \times N$, donde M es el número de renglones y N el número de columnas, es decir, en forma de tabla.



1.3.3 Arreglos Multidimensionales

Pueden existir arreglos de tres o más dimensiones, dependiendo su número y del tipo de lenguaje.

Los arreglos tridimensionales, tal como un cubo, el método mas usual, es definir los puntos con tres coordenadas x, y, z . La declaración de los arreglos bidimensionales, caso particular de los arreglos multidimensionales, se hace como en el siguiente ejemplo:

```
double a[3][4];  
int pos[10][40];  
char list[25][25];
```

En la primera línea se reserva espacio para $3 \times 4 = 12$ elementos doble precisión. El primer subíndice varia entre 0 y 2, y el segundo varia entre 0 y 3. Usualmente, de manera análoga a las matrices, se dice que el primer subíndice indica la fila y el segundo subíndice indica la columna. Un arreglo tridimensional se declararía así :

```
double c[20][30][10];
```

Los sitios para los elementos de a están contiguos en el orden fila por fila, o sea, $a[0][0]$, $a[0][1]$, $a[0][2]$, $a[0][3]$, $a[1][0]$, $a[1][1]$, $a[1][2]$, $a[1][3]$, $a[2][0]$, $a[2][1]$, $a[2][2]$, $a[2][3]$.

En este ejemplo, el programa sirve para leer matrices, escribirlas y calcular el producto. Lo hace mediante la utilización de funciones que tienen como parámetros arreglos bidimensionales.

En muchas ocasiones se requiere el proceso simultáneo de más de un arreglo, que teniendo igual número de elementos, el tipo de datos de los mismos es distinto, estos vectores o matrices se denominan arreglos paralelos.



2. Ejercicios de arreglos

1. Leer un vector de 5 casillas con números enteros y posteriormente imprimir los valores.

- a. Análisis de la solución

Se debe definir que nombre se le da al vector y cual será su subíndice, posteriormente con ellos hacer referencia la vector y llenarlo. Se puede utilizar el subíndice como contador de la estructura que se utilice, ya que el apuntador del vector tendrá que ir avanzando y posicionándose en cada casilla o campos.

- b. Definición de variable y nombres

VNUM nombre del vector que contendrá 5 números enteros.

I nombre del subíndice que servirá para el posicionamiento y posterior localización de la información en el vector.

c. Diagrama de flujo

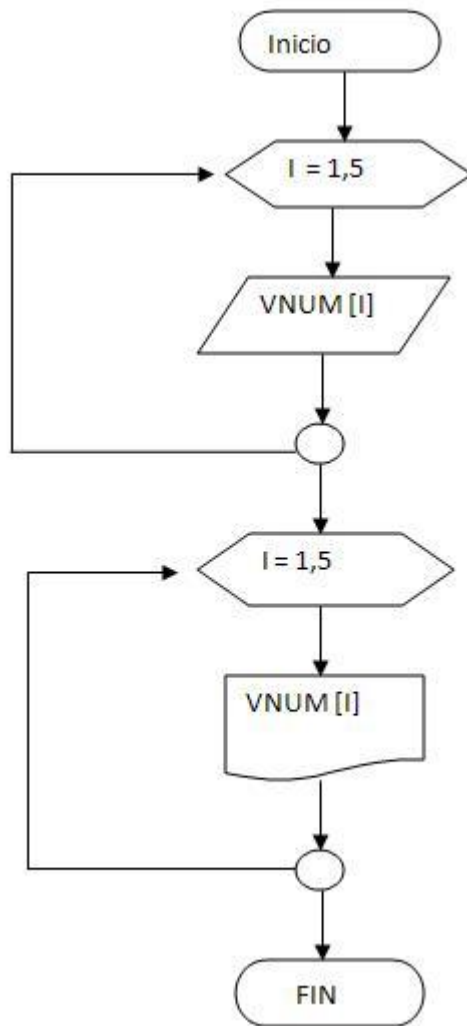


imagen4 ejemplo1

d. Prueba de escritorio

Esta parte de la prueba muestra la lectura o llenado del vector.



I	VNUM[I]
1	20
2	45
3	67 0
4	100
5	12

imagen5 ejemplo1 prueba escritorio

Esta parte de la prueba muestra la escritura o impresión de los datos encontrados en el vector.

I	VNUM[I]
1	20
2	45
3	67
4	100
5	12

imagen6 ejemplo1 prueba escritorio parte2

e. Algoritmo

Inicio

VNUM de 5 elementos

Para I = 1 hasta 5 haga

Lea VNUM [I]

Fin para

Para I = 1 hasta 5 haga

Escriba VNUM [I]

Fin para

Fin algoritmo



2. Llenar un vector de 5 casillas con las calificaciones de un estudiante del curso de programación y posteriormente calcular e imprimir su promedio.

- a. Análisis de la solución

Lo primero que se debe hacer es llenar el vector con las cinco calificaciones, para posteriormente para posteriormente tomar la nota de la casilla e irla sumando en un acumulador y luego sumar el promedio.

- b. Definición de variables y nombres

VNOTA nombre del vector o arreglo unidimensional, que guardara las notas de un estudiante

P nombre del subíndice con el cual se recorrerá el arreglo.

ACN es donde se suman o cumulan las notas encontradas, en cada casilla del vector.

PRO sirve para calcular e imprimir el promedio de las notas del estudiante.

c. Diagrama de flujo

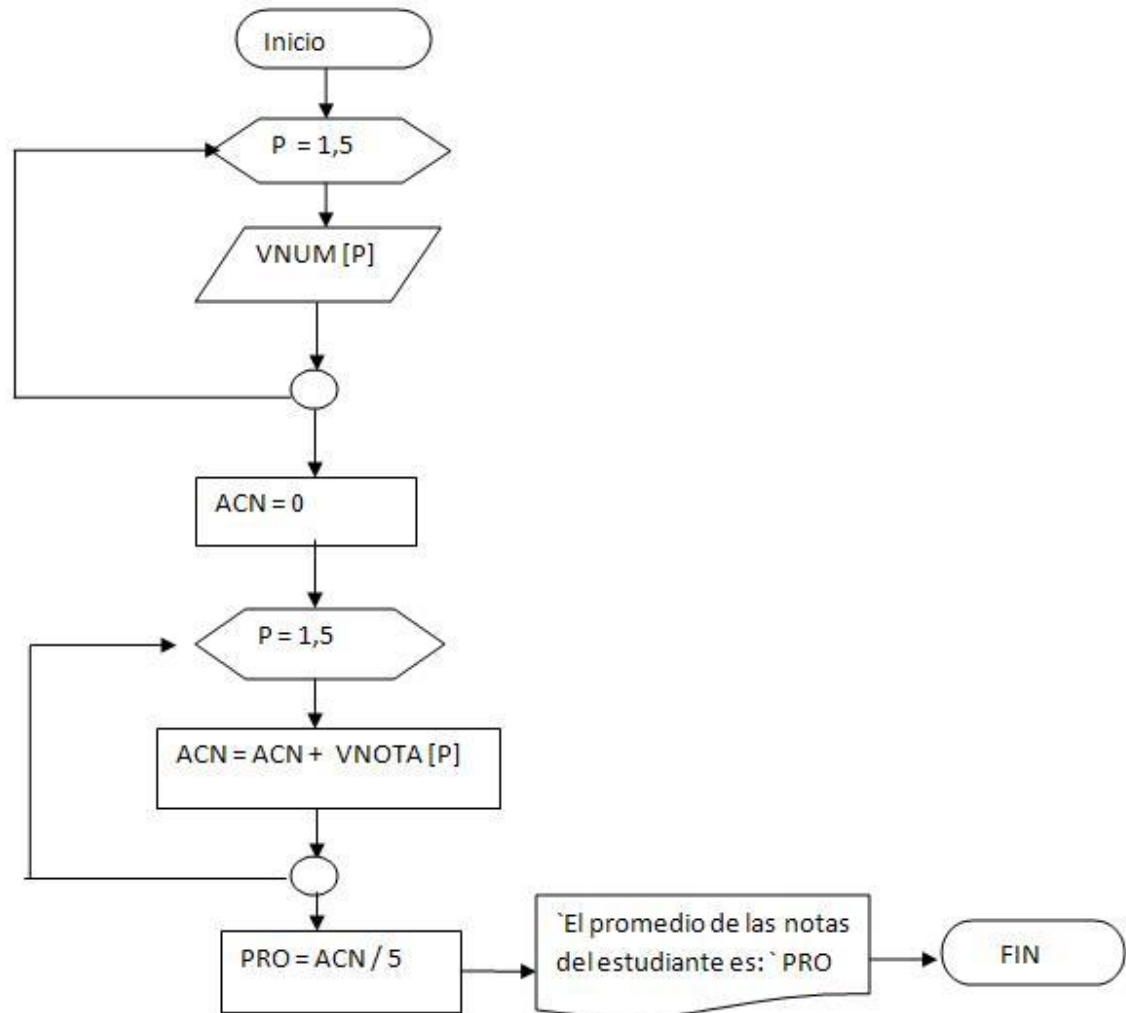


imagen7 diagrama ejemplo 2



d. Prueba escrito

Aquí se llena el vector

P	VNOT[P]
1	4.2
2	1.5
3	5.0
4	3.9
5	0.8

Lo encontrado en cada casilla del vector, se va sumando en el acumulador con el fin de calcular el promedio, cuando se hayan tomado todas las notas del vector así

P	ACCN	PRO
0		
1	4.2	3.08
2	1.5	
3	5.0	
4	3.9	
5	0.8	

imagen8.prueba de escrito ejemplo 2

e. Algoritmo

Inicio

VNOTA de 5 elementos

ACN = 0

Para P = 1 hasta 5 haga

Lea VNOTA[P]

Fin para

PRO = ACN / 5

Escriba `El promedio de notas del estudiante es:`, PRO

Fin algoritmo



Desarrollar un algoritmo y diagrama de flujo el cual, reciba como entrada un arreglo bidimensional de orden $m \times n$ con valores de tipo entero y muestre el arreglo bidimensional:

Algoritmo

Inicio

declaración de variables:

i, j, m, n : entero

mat: Arreglo[m, n] entero

Leer m, n

$i = 1$

Repetir con i desde 1 hasta m

$j = 1$

5.1 repetir con j desde 1 hasta n

Leer mat[i, j]

$j = j + 1$

Fin del ciclo del paso 5.1

$i = i + 1$

Fin del ciclo del paso 5

$i = 1$

Repetir con i desde 1 hasta m

$j = 1$

Repetir con j desde 1 hasta n

Imprimir mat[i, j]

$j = j + 1$

Fin del ciclo del paso 8.1

$i = i + 1$

Fin del ciclo del paso 8

Fin

Diagrama de flujo

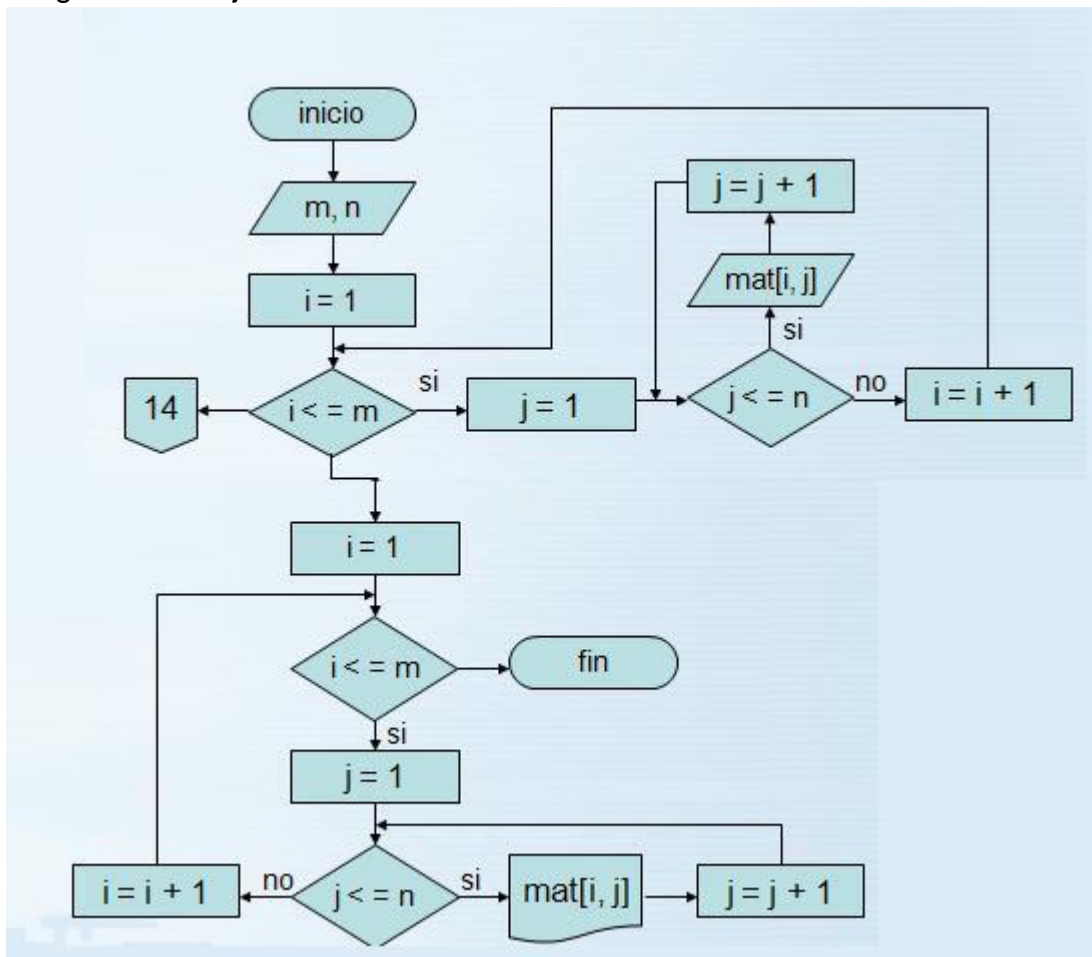


imagen9. Ejemplo arreglos



3. Registros

Cundo se habla de arreglos se dice que se trata de una colección de datos y que con ello se puede solucionar un gran número de problemas. Pero en muchas oportunidades se necesitan estructuras que permitan almacenar diferentes tipos e datos bajo un mismo nombre. Considerando que en una universidad tiene por cada estudiante los siguientes datos:

- Nombre (cadena de caracteres)
- Dirección(cadena de caracteres)
- Matricula (entero)
- Carrera(cadena e caracteres)

Si se requiere almacenar estos datos no sería posible utilizar un arreglo, ya que sus componentes deben ser todos del mismo tipo. La estructura que puede guardar esta información es la que se conoce con el nombre de registro. Cada campo se identifica con un nombre único (el identificador de campo)

3.1 Definición de registros

Un registro es un dato estructurado donde cada uno de sus componentes se denomina campo. Los campos de un registro pueden ser todos de diferentes tipos. Por lo tanto también podrán ser registros o arreglos.

```
iden_registro = REGISTRO
    id_campo1: tipo1
    id_campo2: tipo2
    .....
    .....
    Id_campon: tipon
fin
```

Como un registro es un dato estructurado no puede accesarse directamente como un todo, si no que debe especificarse que elemento (campo) del registro interesa.



3.2. Diferencias con arreglos

Las tres diferencias sustanciales con la estructura de datos tipo arreglo son:

- Un arreglo puede almacenar N elementos del mismo tipo, mientras que un registro puede almacenar N elementos de diferentes tipos.
- Los componentes de un arreglo se accesan por medio de índices, mientras que los componentes de un registro se accesan por medio de su nombre, el cual es único.
- En un arreglo es posible establecer un orden entre sus elementos, mientras que en un registro no existe orden entre los campos.