

Unidad II

Arreglos.

2.1 Unidimensional.

Un arreglo unidimensional es un tipo de datos estructurado que está formado por una colección finita y ordenada de datos del mismo tipo. Es la estructura natural para modelar listas de elementos iguales. Los datos que se guarden en los arreglos todos deben ser del mismo tipo.

El tipo de acceso a los arreglos unidimensionales es el acceso directo, es decir, podemos acceder a cualquier elemento del arreglo sin tener que consultar a elementos anteriores o posteriores, esto mediante el uso de un índice para cada elemento del arreglo que nos da su posición relativa.

Para implementar arreglos unidimensionales se debe reservar espacio en memoria.

Los arreglos nos permiten hacer un conjunto de operaciones para manipular los datos guardados en ellos, estas operaciones son: ordenar, buscar, insertar, eliminar, modificar entre otras.

REPRESENTACION EN MEMORIA: Los arreglos guardan en memoria la cantidad de espacios que se le indican en la declaración.

Ejemplo: sea el siguiente arreglo unidimensional donde se va a guardar 5 datos de tipo entero (integer)

x : array[1..5] of integer

En memoria la computadora guarda 5 espacios: esto quiere decir en el arreglo X en la posición 1 guarda 34, en el arreglo X en la posición 2 guarda 22, así hasta la última posición del arreglo X posición 5 guarda 72.

X [1]

34

X [2]

22

X [3]

39

X [4]

63

X [5]

72

Declaración de los arreglos unidimensionales: Se declara el tipo del arreglo, con la palabra reservada TYPE, luego se declara la variable de tipo arreglo, esto se hace en el bloque de declaración de variables palabra reservada VAR.

Type

nombre_arreglo= array[x..y]of tipo_dato;

Ejemplos:

```
salarios= array [1...x] of real;
```

```
nombre_trabajador=array[1..x] of string;
```

Nota: El tipo de dato del arreglo puede ser: integer,real, boolean, string etc.

Variables tipo array

Var

```
sueldo:salarios;
```

```
nombre:nombre_trabajador;
```

Una vez declarados los arreglos procedemos a cargar información en ellos, para esto usamos estructuras repetitivas la mas recomendable por su facilidad es el for, una vez cargada la información se puede hacer cualquier operación de calculo y por último mostramos la información de los arreglos usando de igual manera la estructura repetitiva for.

Carga de un arreglo, la letra i es el índice o subíndice del arreglo e indica la posición del arreglo:

```
for i:=1 to num do
```

```
begin
```

```
readln(notas[i]);
```

```
end;
```

Mostrar la información de un arreglo, la letra i es el índice o subíndice del arreglo e indica la posición del arreglo:

```
for i:=1 to num do
```

```
begin
```

```
writeln(notas[i]:3:1);
```

```
end;
```

Se han hechos programas que suman una cantidad de números usando dos variables, una para leer cada número y otra para acumular la suma. Este enfoque tiene la desventaja de que se pierden los valores de los sumandos. El uso de arreglos permite calcular la suma de los números con una cantidad mínima de código y a la vez conservar cada valor, como muestra el siguiente programa completo:

```
Program SumaN;  
  
Uses  
  
Crt;  
  
Const  
  
n = 5;  
  
Var  
  
nums: Array[1..n] Of Integer;  
  
s, i: Integer;  
  
Begin  
  
For i:=1 To n Do  
  
Begin  
  
Write('Escriba el número: ');  
  
ReadLn(nums[i]);  
  
s := s + nums[i];  
  
End;  
  
WriteLn('La suma es: ', s);  
  
End.
```

2.2 Multidimensional.

El escalamiento multidimensional (MDS) es una forma de visualizar el nivel de similitud de los casos individuales de un conjunto de datos. Se refiere a un conjunto de técnicas de ordenación relacionados utilizados en la visualización de la información, en particular, para visualizar la información contenida en una matriz de distancia. Un algoritmo de MDS tiene como objetivo colocar cada objeto en el espacio N-dimensional de tal manera que las distancias entre objetos se conservan, así como sea posible. Cada objeto se asigna coordenadas en cada

uno de los N dimensiones. El número de dimensiones de una parcela MDS N puede ser superior a 2 y se especifica a priori. La elección de $N = 2$ optimiza las ubicaciones de los objetos de un diagrama de dispersión de dos dimensiones