

# **Vectores**

Programación digital I  
Escuela de Sistemas  
Facultad de Ingeniería  
Gilberto Diaz

## Tipos de Datos:

- **Simple:** son las unidades mínimas para almacenar un valor, ejemplo: enteros, caracteres, reales.
- **Compuestos:** almacenan uno o más valores de uno o diversos tipos. Estos se utilizan cuando es necesario realizar el procesamiento de múltiples datos que tienen características comunes.  
Ejemplo:
  - Un conjunto de valores enteros

# Vectores o Arreglos

- Representan un conjunto de valores donde todos comparten el mismo nombre. Cada valor es referenciado por uno o más subíndices

numeros      tamaño = 8

11	0	43	54	-3	8	1	23
0	1	2	3	4	5	6	7

# Tipos de Arreglos

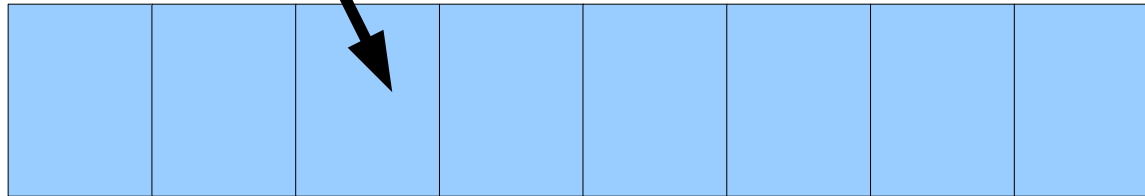
Los arreglos se clasifican según su dimensión:

- **Vectores:** son arreglos unidimensionales (un sólo subíndice)
- **Matrices:** son arreglos de dos dimensiones (dos subíndices)
- **Multidimensionales:** son arreglos de tres o más dimensiones

# Tipos de Arreglos

- Vectores:

numeros[2]

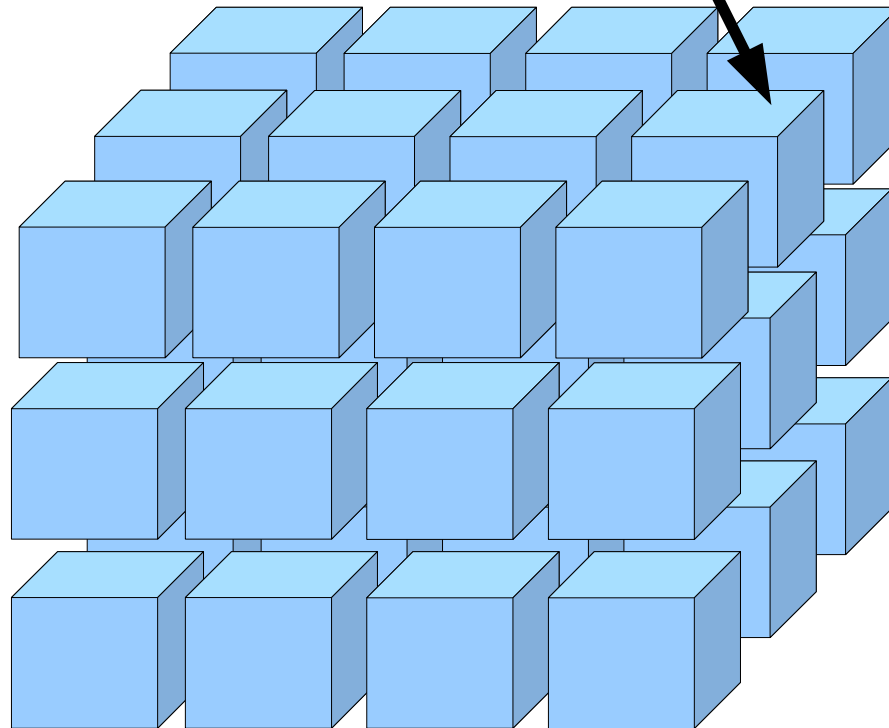




# Tipos de Arreglos

- Matrices:

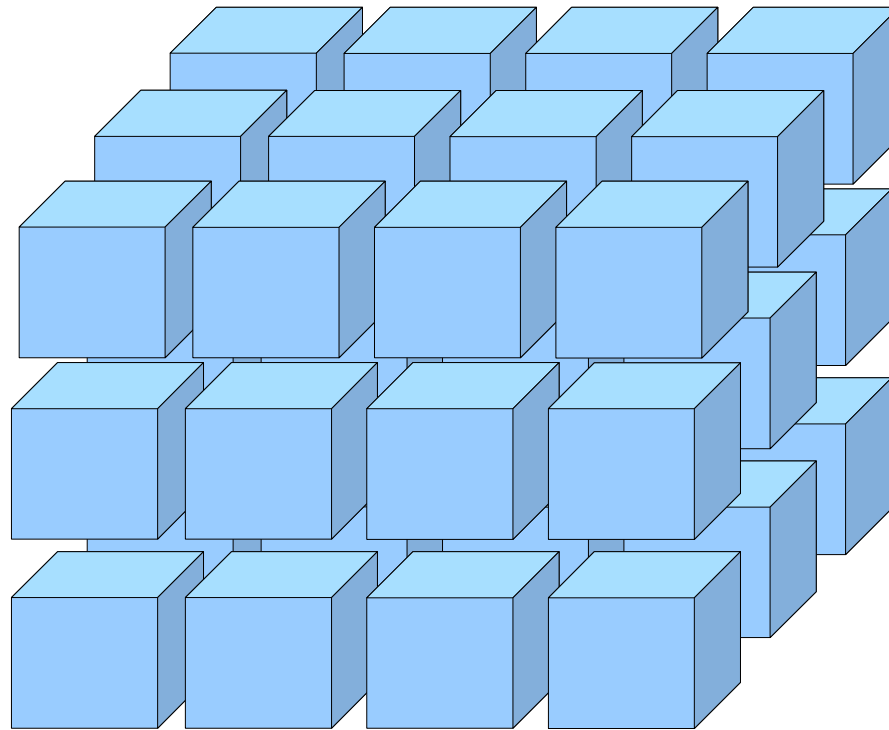
`coordenadas3d[0][3][1]`



# Tipos de Arreglos

- Matrices:

`coordenadas3d[1][0][2] ?`

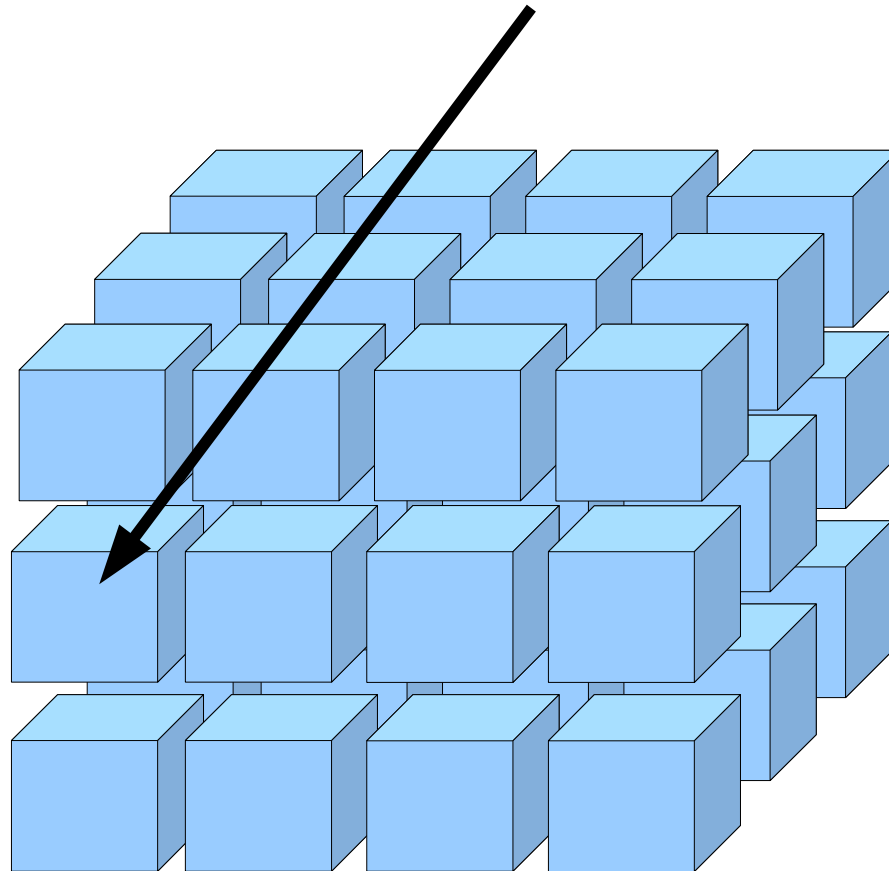




# Tipos de Arreglos

- Matrices:

`coordenadas3d[1][0][2]`



# Vectores

Es un grupo de localidades consecutivas del mismo tamaño, que tienen el mismo nombre y tipo

Cada localidad almacena un elemento del vector

En C los elementos de un vector están numerados desde 0 hasta  $N-1$ . Algunos otros lenguajes comienzan en 1

# Notación algorítmica

TipoDeDato NombreDelVector[númeroElementos]

ejemplo:

entero notas[43]

caracter codigo[4]

## Notación en lenguaje C

TipoDeDato NombreDelVector[númeroElementos];

ejemplo:

```
int notas[43];
```

```
char codigo[4];
```

## Ejemplo en algoritmo

TAM: 10

MAXCAR: 24

NHORAS: 24

NFIL: 2

Entero v[TAM]

Caracter palabra[MAXCAR]

Real temp[NHORAS]

Entero m[NFIL]

## Ejemplo en lenguaje C

```
#define TAM 10
#define MAXCAR 80
#define NHORAS 24
#define NFIL 2

int v[TAM];
char palabra[MAXCAR];
double temp[NHORAS];
int m[NFIL];
```

## **Acceso a los elementos de un Vector**

Para manejar cualquiera de los elementos de un vector se utilizan los subíndices. Por ejemplo para asignar un valor al primer elemento de un vector se utiliza la siguiente sentencia:

```
numeros[0] = 20;
```

## Acceso a los elementos de un Vector

El valor del subíndice puede variar entre 0 y  $N-1$ , siendo  $N$  el tamaño del vector, es decir, el número de elementos que tiene el vector.

El subíndice puede ser construido utilizando expresiones aritméticas enteras. Por ejemplo:

```
notas[x-y+1] = 18;
```

En general para tener acceso al  $i$ -ésimo elemento de un vector se utiliza:

```
vector[i-1]
```



## Acceso a los elementos de un Vector

El subíndice también puede estar conformado por contadores, este caso se puede dar cuando los encontramos dentro de lazos. Por ejemplo:

```
for (i=0; i<N; i++){  
    notas[i] = 18;  
}
```

```
for (i=0; i<N; i++){  
    scanf("%i", num[i]);  
}
```

# Inicialización de un vector

Podemos inicializar los vectores con valores constantes:

```
#define N 100
//Declaración del vector y del subíndice
int vector[N], i;

for (i = 0; i < N; i++)
    vector[i] = 0; // Inicialización de cada
                  // elemento del vector en 0
```

## Inicialización de un vector

Podemos inicializar los vectores con otro tipo de instrucciones

```
int a[4] = {0};
```

```
a[0] = a[1] = a[2] = a[3] = 0
```

```
float n[5] = {0.25, 0.50, 0.75, 1.00, 1.25};
```

```
n[0] = 0.25    n[1] = 0.50    n[2] = 0.75  
n[3] = 1.00    n[4] = 1.25
```

## Inicialización de un vector

Podemos inicializar los vectores con otro tipo de instrucciones

```
char cdn[8] = {'S', 'l', ' ', 'a', 'l', 'l', 'i', ','};
```

```
cdn[0] = 'S'   cdn[1] = 'l'   cdn[2] = ' '  
cdn[3] = 'a'   cdn[4] = 'l'   cdn[5] = 'l'  
cdn[6] = 'i'   cdn[7] = ','
```

## Resumen

Todos los elementos de un vector deben ser del mismo tipo.

Cada elemento de un vector se almacena en una localidad de memoria separada.

Cada elemento de un vector es accedido a través de un subíndice único. El rango del subíndice varía de 0 a  $n-1$ , siendo  $n$  el número de elementos del vector.

Un vector recibe un nombre, independientemente del número de elementos que contenga.

## Resumen

Antes de usar un vector, éste debe ser declarado como cualquier otra variable.

Para acceder un elemento de un vector se usa el nombre, seguido de su posición dentro del vector entre corchetes: nombre\_vector[subíndice].h

## Ejercicios

- Escribir un programa en C que lea una línea de texto, la almacene en un vector y la escriba al revés. La longitud de la línea no será especificada (terminará al pulsar la tecla Enter), pero se supone que no excederá de 80 caracteres.
- Escribir un programa para realizar el producto punto de dos vectores