

Estructuras de Datos

Prof. Tatiana Ilabaca
Primer semestre 2022



Módulo 2
Estructuras de datos estáticas

Arreglos

Objetivos

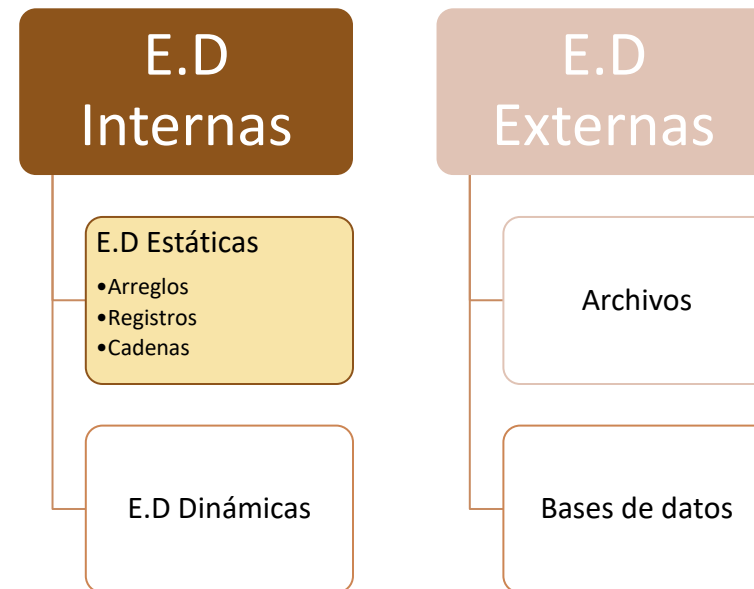
Lección 1

- Conocer las características de una estructura de datos estática
- Conocer las características de los arreglos
- Implementar arreglos unidimensionales (VECTORES)
- Implementar arreglos bidimensionales (MATRICES)
- Implementar arreglos multidimensionales (CUBOS)

Estructuras de datos estáticas

Introducción

- Las estructuras de datos según el **tipo de memoria** donde residen se clasifican en:
 - Estructuras internas (**memoria principal**)
 - Estructuras externas (**memoria secundaria** – soporte externo)



Estructuras de datos estáticas

Características

- La cantidad de espacio asignado en memoria es **fijo**
- Poseen una **cantidad fija** de elementos
- El tamaño de la estructura:
 - queda determinado con la declaración de la estructura en el programa
 - **no puede variar en tiempo de ejecución**
- La reorganización de sus elementos puede resultar muy costosa

Arreglos

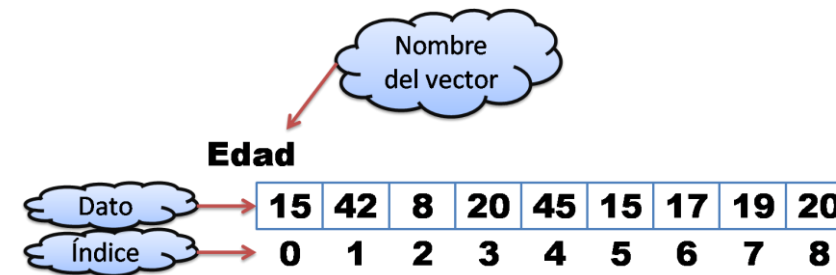
Características

- Colección **ordenada** de elementos; cada elemento tiene una posición
- Estructuras **homogéneas**; sus elementos son del mismo tipo de datos
- Permiten referirse a una colección de elementos mediante un **mismo nombre**.
- Son **referenciados**; el acceso a cada elemento se realiza a través de uno o más **índices**
- El número de índices depende de las **dimensiones** del arreglo

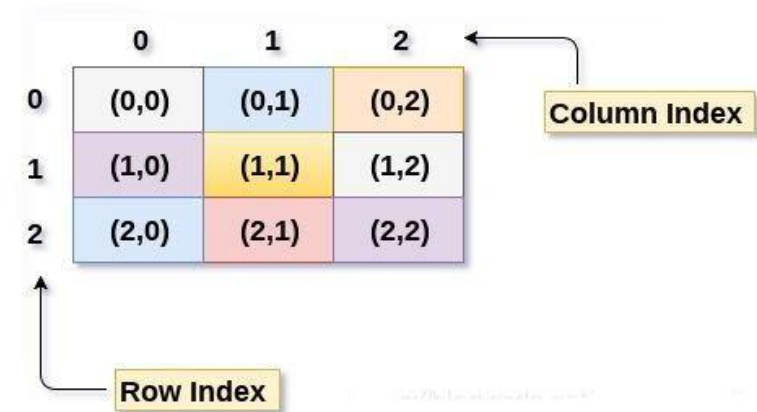
Arreglos

Características

- Los arreglos más comunes son:
 - Arreglo unidimensional: **Vector**



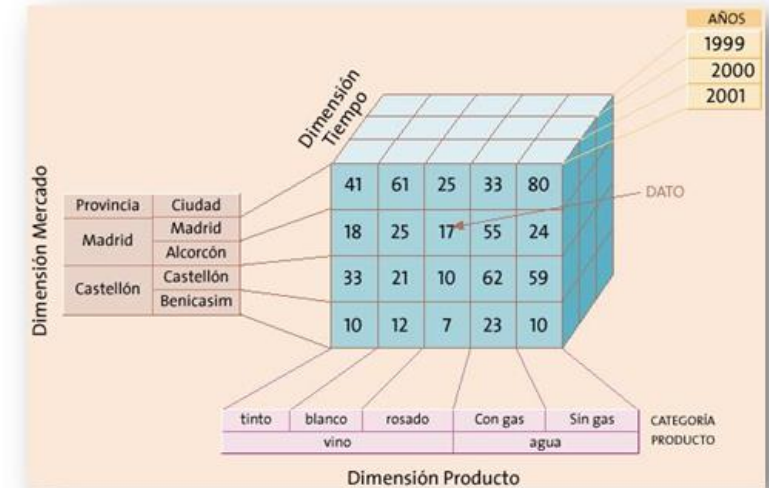
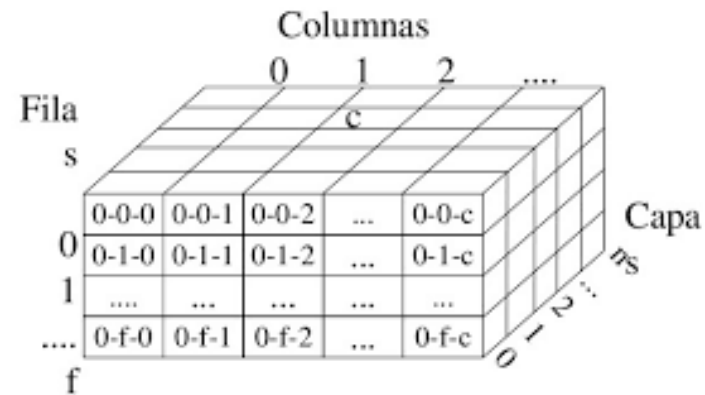
- Arreglo bidimensional: **Matriz**



Arreglos

Características

- Los arreglos más comunes son:
 - Arreglo tridimensional: **Cubo (Arreglo multidimensional)**



Arreglos

Características

- Un arreglo representa a un conjunto de **celdas de memorias consecutivas** y su identificador corresponde la dirección de memoria del primer elemento
- Por ejemplo: Sea el vector **a**, con capacidad para 5 elementos e inicializado con los valores 1, 2 y 3

```
short int a[5]={1,2,3};
```

a es la dirección de memoria de la primera celda

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99

Vectores

Características

- Arreglo unidimensional
- Utiliza un índice para referenciar los elementos
- El índice debe ser de tipo entero
- El valor del índice puede estar dado por un número, una variable o una expresión

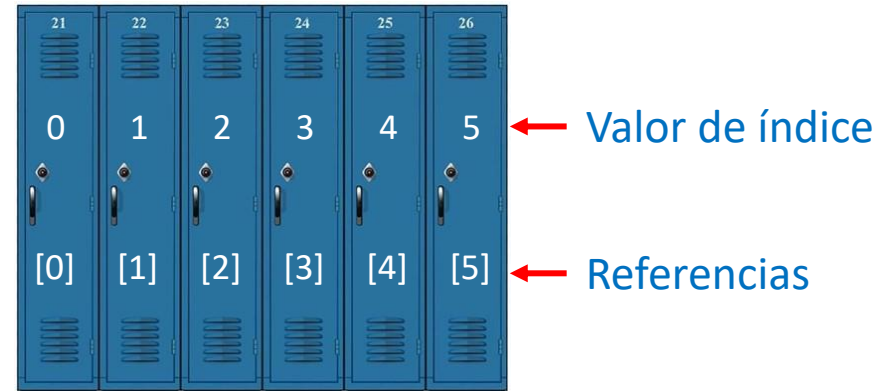
```
int vector1[10];  
float vector2[5];  
char vector3[20];
```

```
int i=0;
```

```
vector1[1]=7;  
vector2[i]=3.5;  
vector3[i+1]='A';
```

Vectores

Características

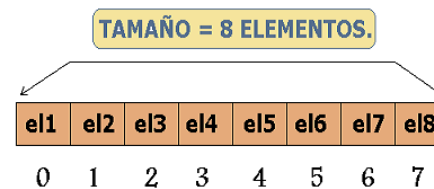


Declaración en C

```
tipo nombre[tamaño];
```

//Ejemplo

```
int locker[6];
```



```
float calificaciones[8];
```

Vectores

Operaciones

- Inicialización
- Asignación
- Eliminación*
- Modificación
- Búsqueda
- Ordenamiento

Matrices

Características

- Arreglo bidimensional, formado por **filas** y **columnas**
- Utiliza **dos índices** para referenciar los elementos, uno para las filas y otro para las columnas

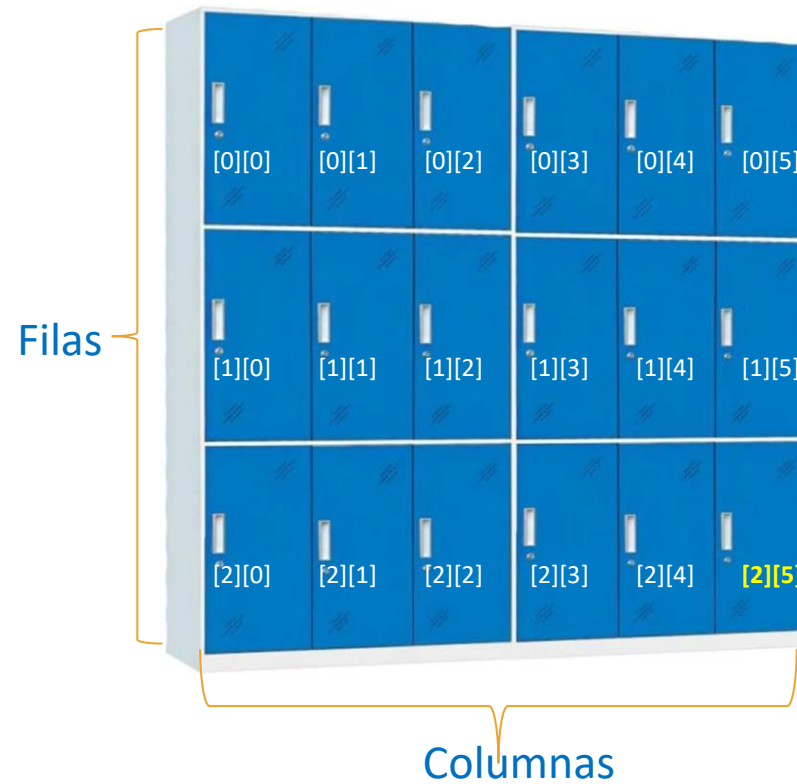
```
int puntaje[30][5];      //Filas: alumnos, Columnas: pregunta
int medicionUV[3][14]    //Filas: tipo, Columnas: hora (07:00 a 20:00 hrs.)
float nota[20][5]        //Filas: alumnos, Columnas: tipo de evaluación
int i;
int j;

i=0;
j=0;

puntaje[7][3]=15;
nota[15][0]=6.5;
medicionUV[i][j]=700;
puntaje[10][j+1]=8;
```

Matrices

Características



Referencias (o “coordenadas”)

Declaración en C

```
tipo nombre[cantFilas][cantColumnas];
```

//Ejemplo

```
int locker[3][6];
```

Matrices

Características

$$\begin{pmatrix} 2 & 1 & 5 & -1 & 8 \\ -1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 10 & 11 & 13 \end{pmatrix}$$

Matriz de orden 3x5

```
int matriz[3][5];
```

Matrices

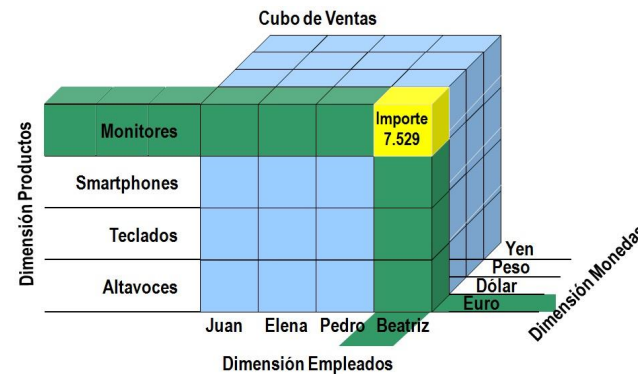
Operaciones

- Inicialización
- Asignación
- Eliminación*
- Modificación
- Búsqueda
- Ordenamiento

Cubos

Características

- Arreglo tridimensional (multidimensional), formado por **planos**, **filas** y **columnas**.
- Utiliza **tres índices** para referenciar los elementos; uno para los planos, uno para las filas y otro para las columnas

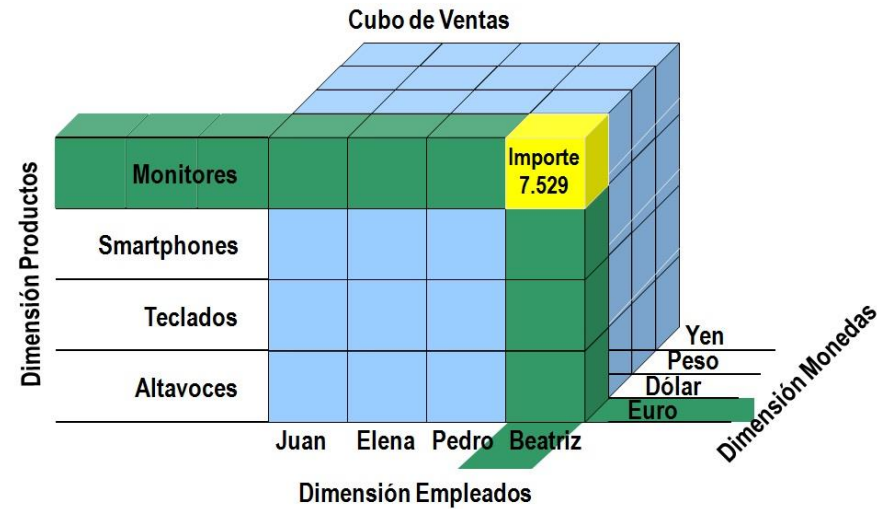


//Declaración en C

```
tipo nombre[cantPlanos][cantFilas][cantColumnas];
```


Cubos

Características



```
//Ejemplo
//Filas: Productos
//Columnas: Empleados
//Planos: Tipo de moneda
```

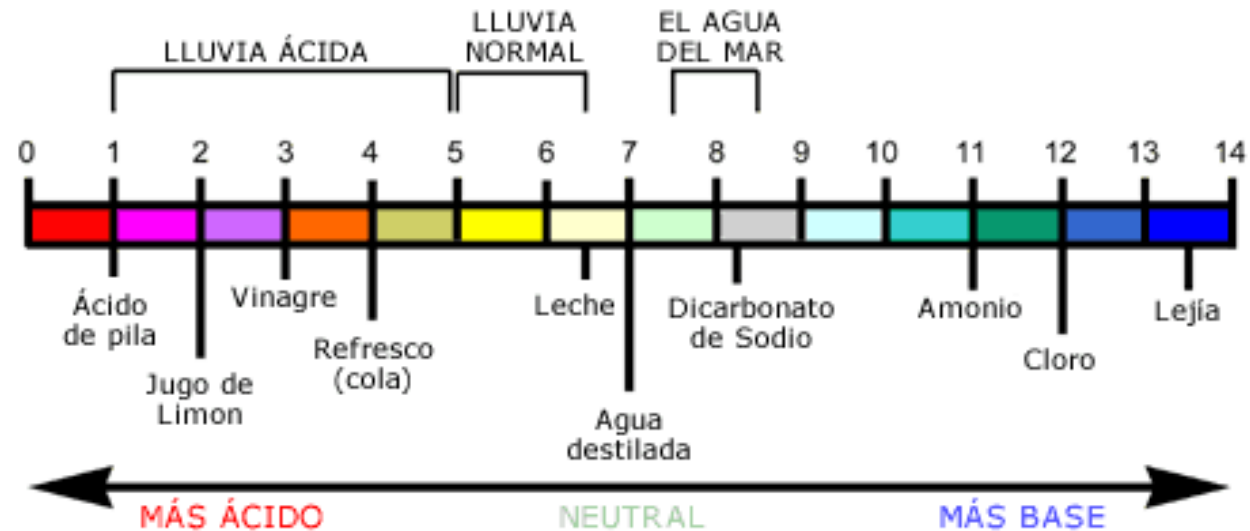
```
float ventas[4][4][4];
```

```
//Referencia del Importe
//Plano: 0 - Fila: 0 - Columna: 3
```

```
ventas[0][0][3];
```

Actividad 1

- Implementar un programa que permita ingresar los valores correspondientes a 10 mediciones de PH, los almacene en un vector y calcule el promedio



Actividad 2

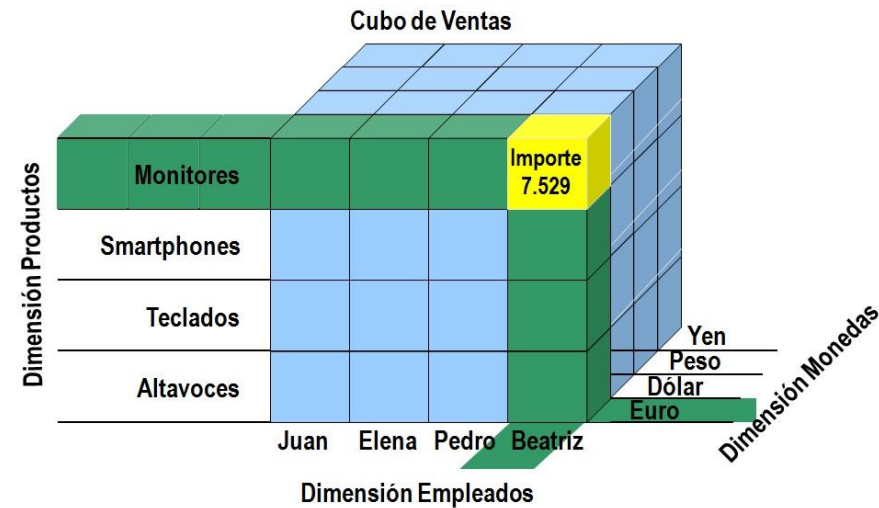
- Implementar un programa que permita sumar dos matrices de orden nxn
- Por ejemplo:

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 3 & 0 & 0 \\ 5 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

$$A + B = \begin{pmatrix} 2+1 & 0+0 & 1+1 \\ 3+1 & 0+2 & 0+1 \\ 5+1 & 1+1 & 1+0 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 2 \\ 4 & 2 & 1 \\ 6 & 2 & 1 \end{pmatrix}$$

Actividad 3

- Implementar el ingreso de datos al cubo siguiente. Considera sólo las monedas Dólar y Peso.
- Para completar los valores en pesos, haz la conversión de acuerdo al valor del dólar actual.



Estudiar



Estudiar para el Martes 19.04

- Texto: **Estructura de Datos**. Proyecto LATIn
- Capítulo 3: Algoritmos de búsqueda
 - Introducción
 - Búsqueda secuencial
 - Búsqueda binaria
 - Búsqueda Hash (hasta colisiones). Métodos 1 y 2.
- **No considerar complejidad**