

Estructura y Representación de Datos

Prof. Tatiana Ilabaca
Primer semestre 2021



Módulo 2
Estructuras de datos estáticas

Arreglos

Objetivos

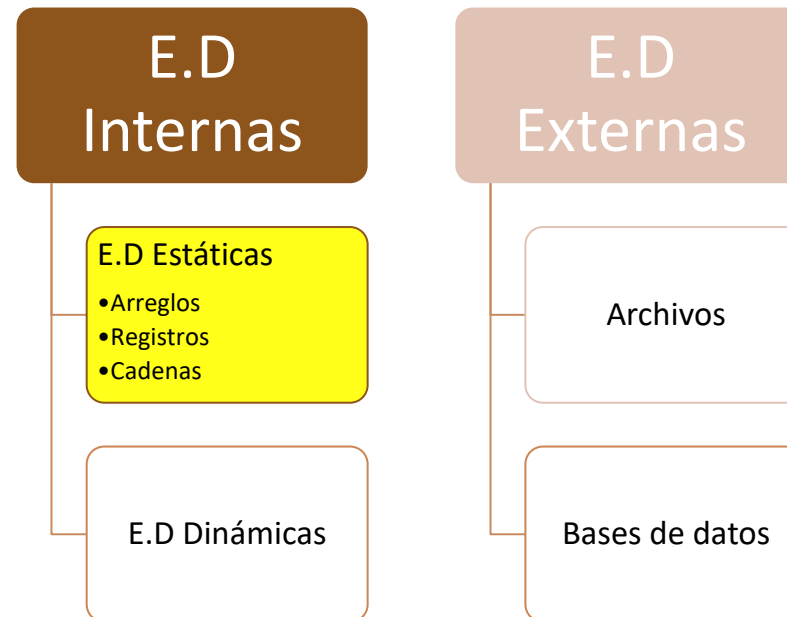
Lección 1

- Conocer las características de una estructura estática
- Conocer las características de los arreglos
- Conocer las características de los arreglos unidimensionales
- Aplicar e implementar arreglos unidimensionales (VECTORES)

Estructuras de datos estáticas

Introducción

- Las estructuras de datos, según donde residían, se clasificaban en:
 - Estructuras internas (memoria principal)
 - Estructuras externas (soporte externo)



Estructuras de datos estáticas

Características

- Poseen una **cantidad fija** de elementos
- La cantidad de espacio asignado en memoria es fijo
- 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
- Son 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
- Los arreglos más comunes son:
 - Arreglo unidimensional: **Vector**
 - Arreglo bidimensional: **Matriz**
 - Arreglo tridimensional: **Cubo**

Arreglos

Características

- Un arreglo representa a un conjunto de **celdas de memorias consecutivas** y su identificador es la dirección de memoria del primer elemento

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

Diagram illustrating memory layout for the array `a`. The array starts at address 68 (labeled `a(68)`) and contains the values 1, 2, and 3. The memory addresses 68, 69, and 70 are highlighted, corresponding to the values 1, 2, and 3 respectively. The array is declared as `short int a[5]`, indicating it can hold 5 elements of type `short int`.

Arreglos

Vectores (Arreglos unidimensionales)



Referencias

Arreglo
Unidimensional

Finito: Contiene N elementos.
Homogéneo: del mismo tipo.
Ordenado: con una posición.
Referenciado: con un índice.

PRIMERO	SEGUNDO	TERCERO	CUARTO	QUINTO	SEXTO	SÉPTIMO	...	ENÉSIMO
C_1	C_2	C_3	C_4	C_5	C_6	C_7	...	C_N
i_0	i_1	i_2	i_3	i_4	i_5	i_6	...	C_{N-1}

Declaración en C

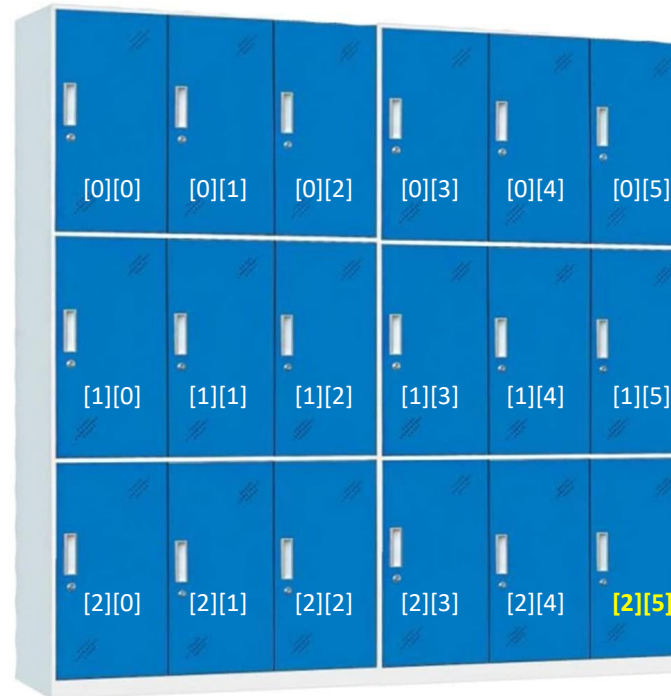
`tipo nombre[tamaño];`

//Ejemplo

`int locker[6];`

Arreglos

Matrices (Arreglos bidimensionales)



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

Matriz de orden 3x5

Referencias

Declaración en C

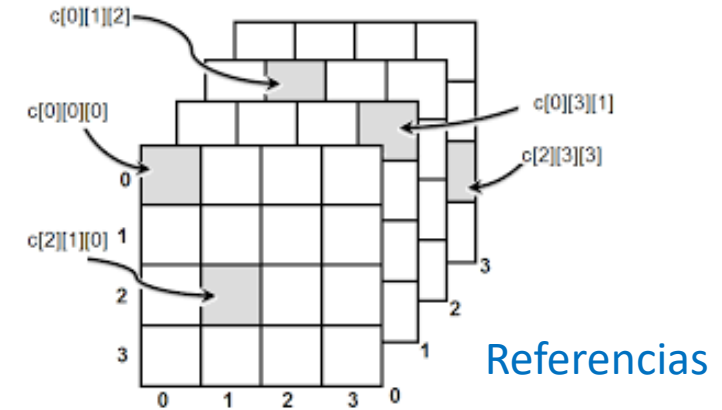
```
tipo nombre[cantFilas][cantColumnas];
```

//Ejemplo

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


Arreglos

Cubos (Arreglos tridimensionales)



//Declaración en C

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

*//Ejemplo
//Filas: Tipo vacuna
//Columnas: Rango etareo
//Planos: Región*

```
int vacunacion[2][10][16];
```

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

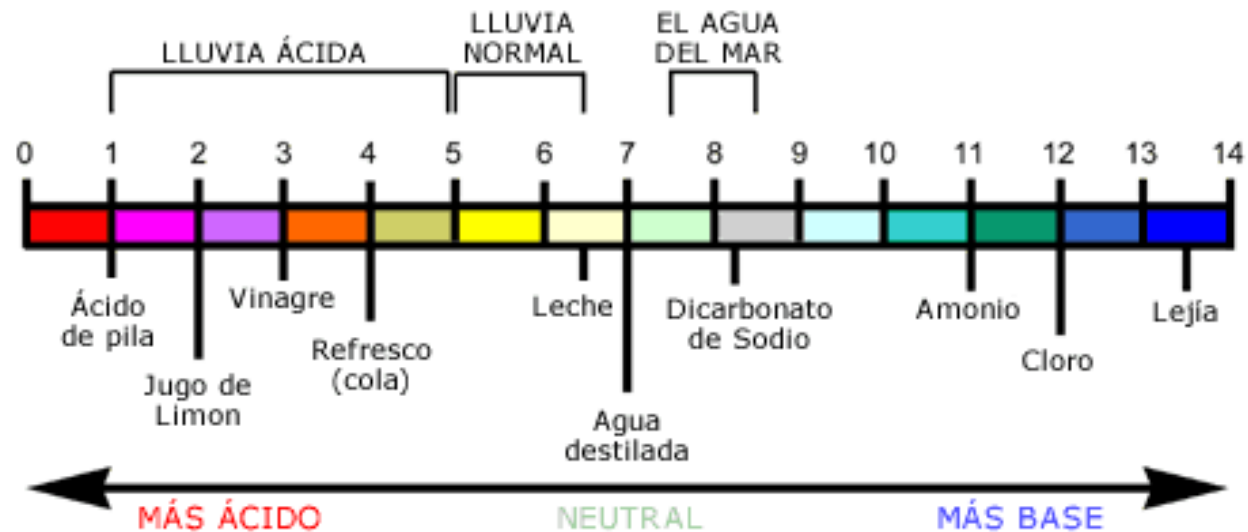
Operaciones

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



Actividad

- Implementar un programa que permita ingresar los valores correspondientes a 10 mediciones de PH, los almacene en un vector, calcule el promedio y obtenga el valor de la medición más ácida.



Estudiar



Estudiar para el Lunes 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**



Actividad

Desarrollar la **Guía de Práctica 3** publicada en el Aula Virtual.

