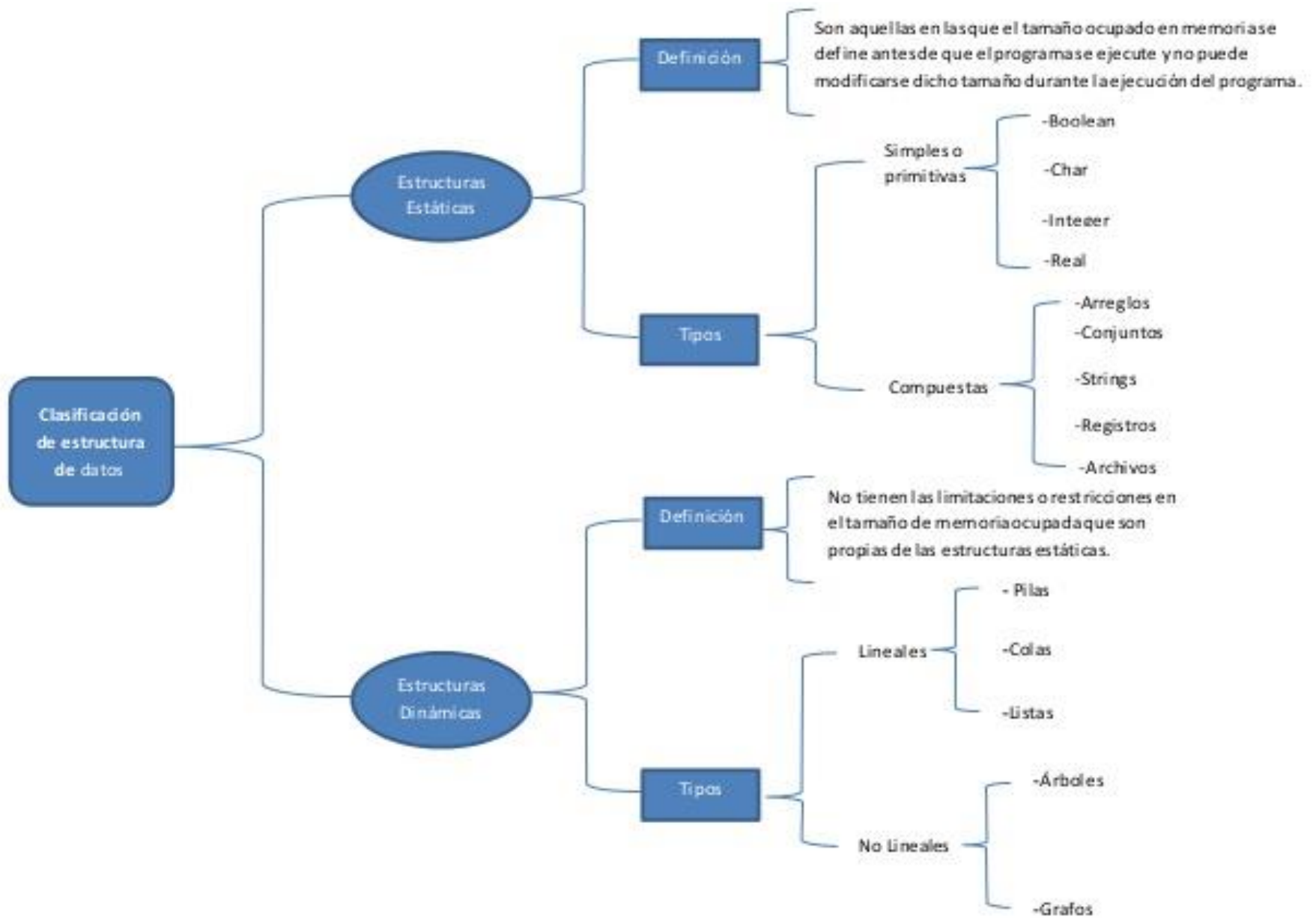


Laboratorio de Computación I

Estructuras de Datos

Es una colección de datos que se caracterizan por su organización y las operaciones que se definen en ella.



ARREGLOS - ARRAYS



Definición

Los arreglos son una colección de variables del mismo tipo que se referencian utilizando un nombre común.

5	67	98	34	22
a[0]	a[1]	a[2]	a[3]	a[4]

Definición

Se puede definir de una forma abstracta como “***un conjunto finito ordenado de elementos homogéneos***”.

Por **finito**, entendemos que hay un número específico de elementos en el arreglo; número que debe ser grande o pequeño pero debe existir.

Por **ordenado**, entendemos que los elementos están dispuestos de tal manera que hay un elemento cero, un elemento primero, un segundo, un tercero y así sucesivamente.

Por **homogéneo**, entendemos que todos los elementos del arreglo son del mismo tipo.

Características:

- También se conoce con el nombre de ARRAY o VECTOR.
- Un arreglo consta de posiciones de memoria contigua de igual tamaño consecutivas que tienen el mismo tipo de variable.
- La dirección más baja corresponde al primer elemento y la más alta al último. El límite inferior se fija siempre en 0 y el superior lo fija el programador .
- Un arreglo puede tener una o varias dimensiones .
- Para acceder a un elemento en particular de un arreglo se usa el nombre del arreglo seguido de un índice entre corchetes.
- La cantidad total de espacio que se usa por un arreglo depende de 2 cosas: El número de elementos en el arreglo y El tamaño del arreglo.

Declaración de un Arreglo Unidimensional

El formato para declarar un arreglo unidimensional es:

tipo nombre_arr [tamaño]

Por ejemplo, para definir un arreglo de diez valores enteros, se debe definir este arreglo de la siguiente manera:

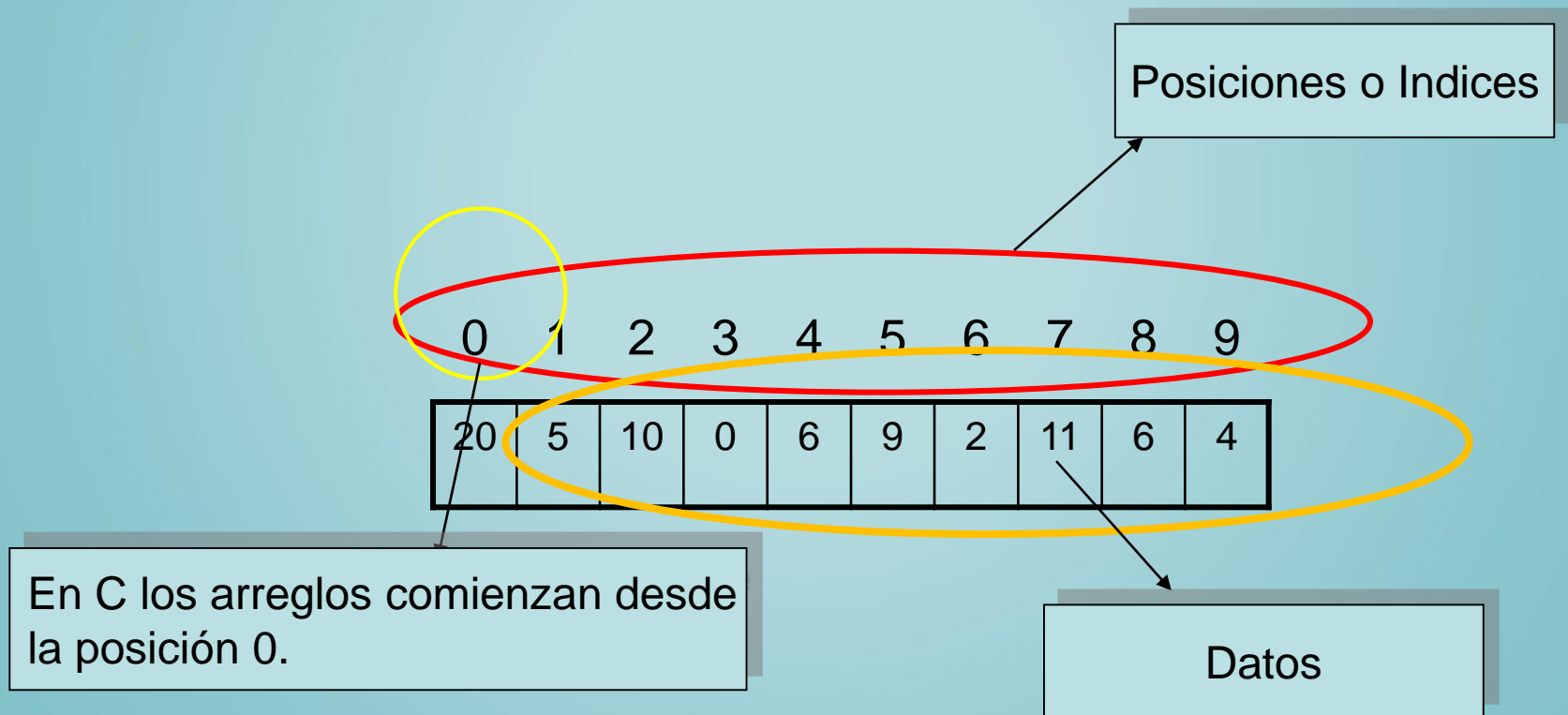
int elem[10];

Esta expresión es la declaración del arreglo. Donde int es el tipo de datos que almacena el arreglo, elem es el nombre del arreglo, y el número encerrado en los corchetes es el número de valores que contiene el arreglo.

En C, todos los arreglos usan cero como índice para el primer elemento. En el ejemplo se declara un arreglo de enteros con diez elementos desde elem[0] (primer elemento) hasta elem[9] (último elemento).

Declaración de un Arreglo Unidimensional

Ejemplo: considerando estas características antes nombradas, se puede conceptualizar un arreglo de 10 enteros de la siguiente forma grafica.



Operaciones con Arreglos

- Cargar un arreglo.
- Recorrer un arreglo.
- Buscar un elemento en particular.
- Acceder a un elemento en una posición determinada.
- Acceder a una posición determinada y mostrar su contenido.
- Insertar un nuevo elemento.
- Eliminar un elemento.
- Ordenar un arreglo

Operaciones con Arreglos

- **Cargar un arreglo.**
 - Mediante declaración del arreglo.
 - Usando una estructura repetitiva
- **Recorrer un arreglo para mostrar sus elementos.**

Forma de inicializar un arreglo es la siguiente:

tipo nombre_arreglo[tamaño] = {lista-valores};

Ejemplo:

int x[7] = {3,5,7,-2,0,1,4};

int y[30] = {2,4,5}; // el resto de los espacios vacíos es cero

int z[3] = {1,2,3,4} //esto es un desbordamiento ERROR

Lenguaje de Programación C

```
Int lista[9]= {0, 4, 78, 5, 32, 9, 77, 1, 23}  
Posición → 0  1  2  3   4  5  6  7  8 = (9 posiciones)
```

/*Utilizacion de arreglos*/

```
#include <stdio.h>  
#include <conio.h>
```

Cargar elementos en la declaración del arreglo

```
int main()  
{  
  
    int lista[9]= {0, 4, 78, 5, 32, 9, 77, 1, 23};  
    int i;  
  
    for(i = 0; i < 9; i++)  
        printf("Digito %d:%d\n",i,lista[i]);  
  
    return 0;  
    getch();  
}
```

Cargar elementos en el arreglo usando estructura repetitiva.

```
#include <stdio.h>
#include <conio.h>
```

```
int main()
{
```

```
    int lista[10];
    int i;
```

```
    for(i = 0; i < 10; i++)
    {
```

```
        printf("Ingresar el elemento %d:\n",i);
        scanf("%d",&lista[i]);
```

```
    }
```

```
    /*Mostrar los elementos del arreglo*/
```

```
    for(i = 0; i < 10; i++)
        printf("Elemento %d:%d\n",i+1,lista[i]);
```

```
    return 0;
    getch();
}
```


Operaciones con Arreglos

- **Buscar un elemento en particular.**
- **Acceder a un elemento en una posición determinada.**
- **Acceder a una posición determinada y mostrar su contenido.**

Usando Procedimientos en Arreglos

```
#include <stdio.h>
```

```
#define MAX 5
```

```
int vec[MAX];
```

```
void mostrar(void);
```

```
int main()
```

```
{
```

```
    int i;
```

```
    for(i=0;i<MAX;i++)
```

```
    {
```

```
        printf("Ingrese un numero entero");
```

```
        scanf("%d",&vec[i]);
```

```
    }
```

```
    mostrar();
```

```
    return 0;
```

```
}
```

```
void mostrar(void)
```

```
{
```

```
    int i;
```

```
    for(i=0;i<MAX;i++)
```

```
        printf("%d ",vec[i]);
```

```
}
```