Estructura y Representación de Datos

Prof. Tatiana Ilabaca

Primer semestre 2021

Módulo 3
Estructuras de datos dinámicas

Introducción

Objetivos

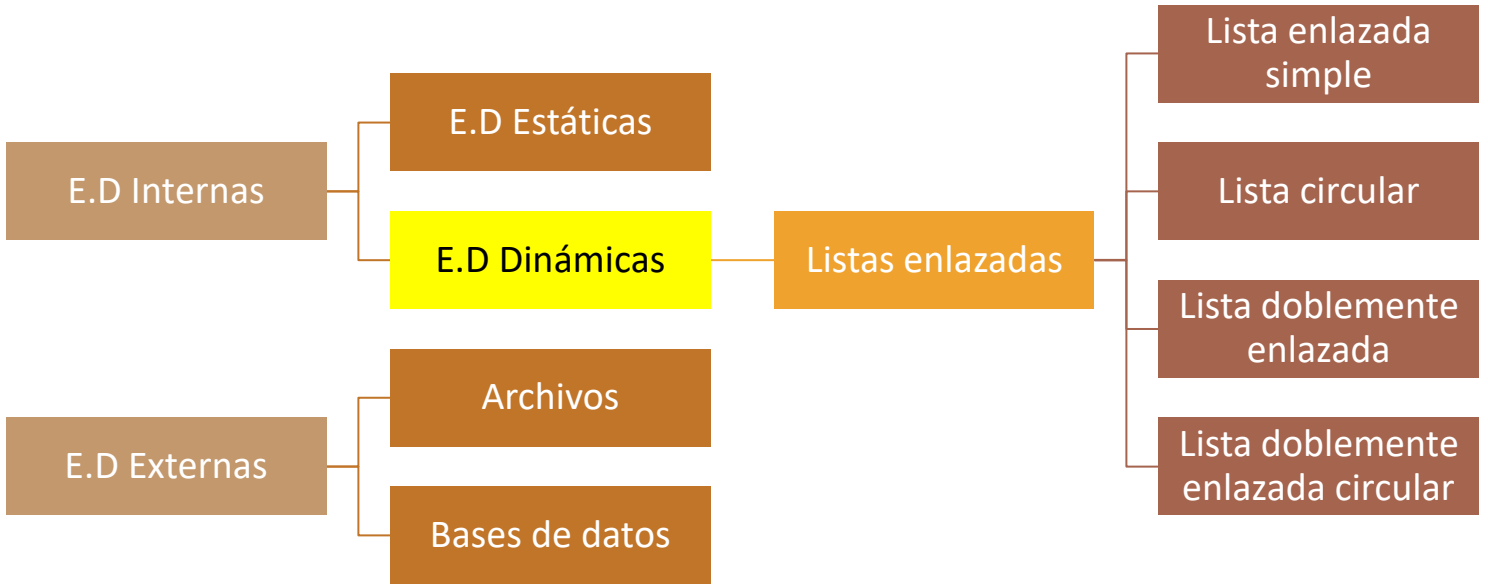
Lección 1

- Conocer las características de una estructura dinámica
- Conocer el concepto de puntero
- Conocer los operadores asociados a los punteros

Estructuras de datos dinámicas

Introducción

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



tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

Características

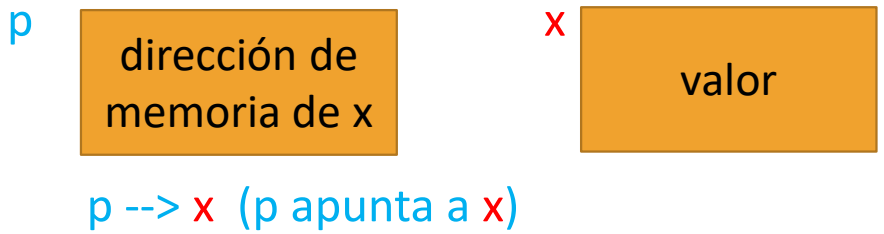
- Poseen una cantidad variable de elementos
- La cantidad de espacio asignado en memoria es variable
- El tamaño de la estructura puede variar en tiempo de ejecución, ya sea aumentando o disminuyendo
- Corresponden a una colección de elementos de un mismo tipo, estructuras homogéneas
- Se forman enlazando nodos (estructura que contiene un conjunto de valores y uno o más punteros)

tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

Puntero

- Variable estática cuyo valor corresponde a una dirección de memoria



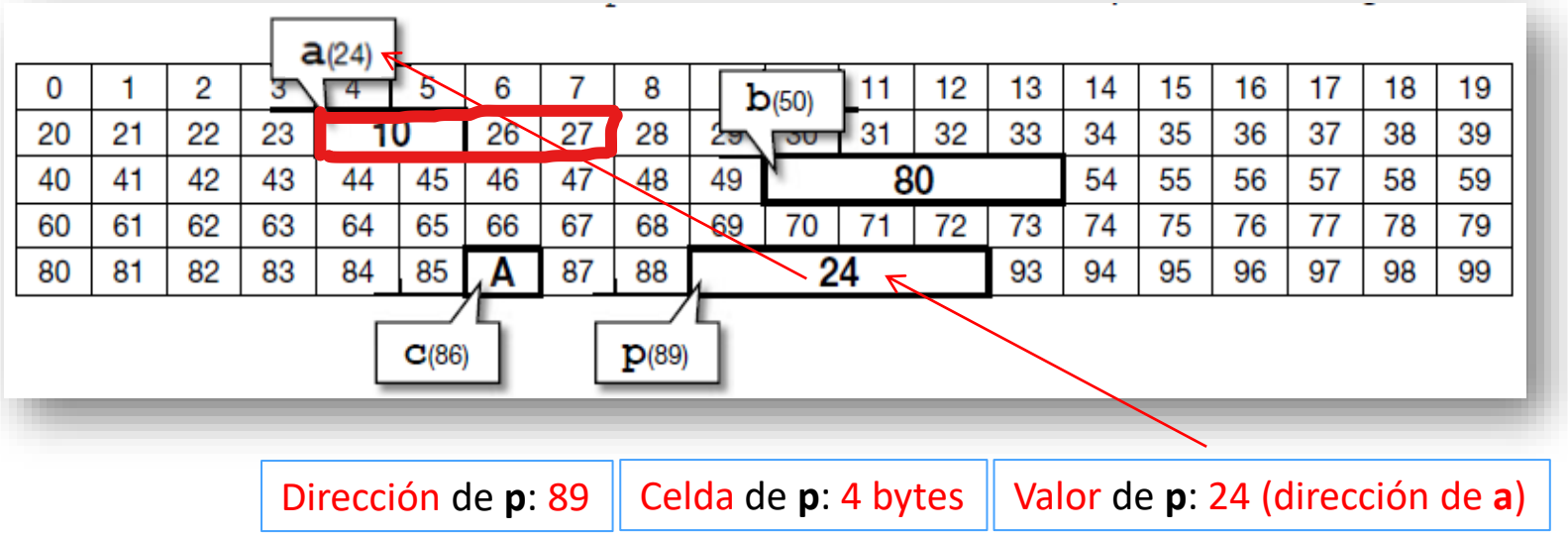
- Se representan en celdas de 4 bytes
- Inicialmente no posee valor, sin dirección (“no apunta a otra variable”)
- En el lenguaje C se utiliza NULL para indicar que la variable no contiene una dirección

tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

Declaración de un puntero

- Sea la declaración de variable: `int* p; // 4 bytes`
- Considerando la declaración de la variable p, una representación en memoria podría ser:



Dirección de **p**: 89 | Celda de **p**: 4 bytes | Valor de **p**: 24 (dirección de **a**)

tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

Declaración de un puntero

- ¿Cuál es la diferencia?

```
int* p;

int  *p;
```

```
int* p, q, r;    //TODAS las variables son de tipo puntero

int  *p, q, r;   //SOLO p es una variable de tipo puntero
```


tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

Operador de dirección - &

- Al anteponer el operador & al identificador de una variable, se obtiene su dirección de memoria

```
void main()
{

    int a = 10;
    int* p;

    p = &a; //Asignación de la dirección de memoria de la variable a
            //al puntero p
}
```

Como la variable **a** es de tipo **int**, entonces el tipo de dato de **p** es **int***.

tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

Operador de indirección - *

- Es un operador con doble función:
  - Para declarar una variable de tipo puntero

```
int* p;
```

  - Para acceder al espacio de memoria que el puntero direcciona, es decir, al contenido almacenado en dicha dirección

```
printf("El valor almacenado en la direccion asignada a p es: %i",*p);
```


tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

Operador flecha - -->

- Permite hacer referencia directa a los campos de una estructura que es accedida a través de un puntero
- Ejemplo:

```
typedef struct Dispositivo
{
    int  nroSerie;
    char marca[30];
    char sisOperativo[15]; //1:Android, 2: iOS, 3: Windows phone
    int  tamPantalla;
}Producto;

//Declaración de variables
Producto  disp1={111, "Samsung", "Android",7};
Producto* disp2=NULL;

printf("Marca en disp1: %s",disp1.marca);

printf("Marca en disp2: %s",disp2->marca);
```

Agregar, entre ambas salidas:

```
disp2=&disp1;
```


tatiana.llabaca.upla@gmail.com - 012021

Actividad

- Implementar el siguiente código

```
void main()
{
    int a = 10;
    int* p, q;  //AMBAS variables son de tipo puntero.
                //Cada variable puede almacenar una única dirección de memoria

    p = &a;     //Asignación de la dirección de memoria de la variable a al puntero p

    q = &p;     //Asignación de la dirección de memoria de la variable p al puntero q
}
```

- A continuación, agregar:

```
//Las direcciones varían de un equipo a otro
printf("El valor de a es: %i",a);
printf("\nLa direccion de a es: %i",p);

printf("\nEl valor de p es: %i",p);
printf("\nLa direccion de p es: %i",q);

printf("\nEl valor de q es: %i",q);
printf("\nLa direccion de q es: %i",&q);
```


tatiana.llabaca.upla@gmail.com - 012021