# Estructura y Representación de Datos

Prof. Tatiana Ilabaca

Primer semestre 2021


## Módulo 3
Estructuras de datos dinámicas

Nodo

# Objetivos

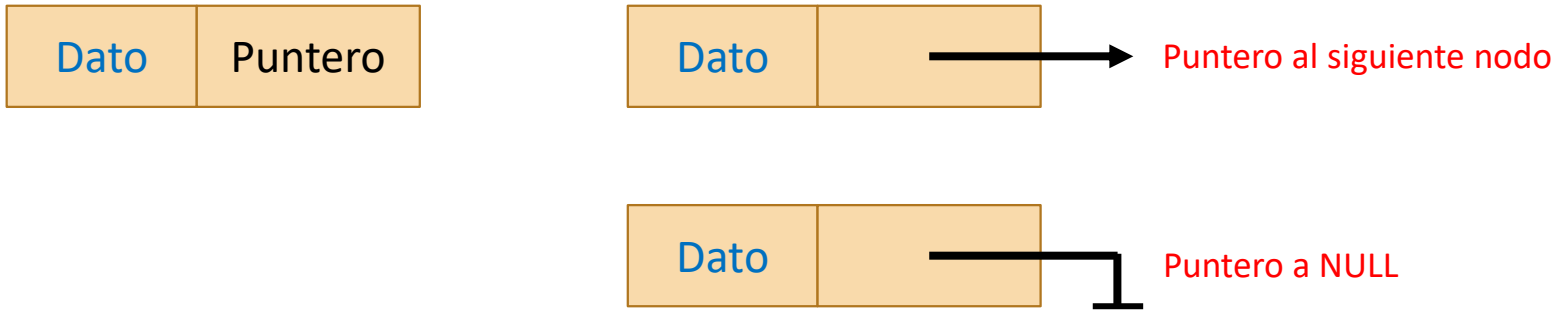
## Lección 2

- Conocer las características de un Nodo
- Definir un Nodo
- Implementar las operaciones básicas asociadas a los Nodos: creación, eliminación y acceso

# Estructuras de datos dinámicas

## Nodo

- Representa a un conjunto de uno o más valores, más un puntero referenciando al siguiente elemento de la colección
- Es una estructura heterogénea, ya que puede contener campos de distinto tipo de dato
- Esquema básico:





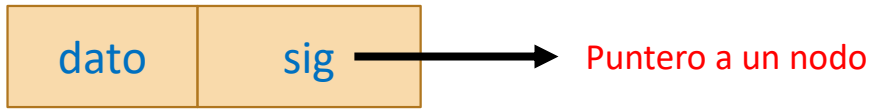

tatiana.llabaca.upla@gmail.com - 012021

# Estructuras de datos dinámicas

## Definición de un nodo en C

- Definición: código que lo implementa

```
typedef struct Nodo
{
    int dato;           //Valor a almacenar en el nodo
    struct Nodo* sig;   //Dirección de un nodo con la misma estructura
}Nodo;
```






tatiana.llabaca.upla@gmail.com - 012021

# Estructuras de datos dinámicas

## Creación de un nodo en C – Función malloc()

- Permite direccionar n bytes de memoria consecutivos
- La memoria gestionada con malloc permanece durante toda la ejecución del programa y trasciende a la función que la invocó
- Requiere la biblioteca <stdlib.h>

```
Nodo* nodo=NULL;

nodo=(Nodo*)malloc(sizeof(Nodo));
```

Dirección de acuerdo al tipo de nodo

Solicitud de memoria

Cantidad de bytes de acuerdo a la estructura del nodo

Operaciones explícitas de tipos en C: Operador cast
Forma general: (tipo_de_dato) expresión



tatiana.llabaca.upla@gmail.com - 012021

# Estructuras de datos dinámicas

## Creación de un nodo en C – Ejemplo

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

typedef struct Nodo
{
    int dato;
    struct Nodo* sig;
}Nodo;

void main()
{
    Nodo* nodo=NULL;

    printf("Direccion de la variable nodo antes de malloc: %p",&nodo);
    printf("\nValor de la variable nodo antes de malloc: %p",nodo);

    nodo=(Nodo*)malloc(sizeof(Nodo));

    printf("\nDireccion de la variable nodo despues de malloc: %p",&nodo);
    printf("\nValor de la variable nodo despues de malloc: %p",nodo);

    //Asignación de valores al nodo creado
    nodo->dato=27;
    nodo->sig=NULL;
}
```



tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

# Eliminación de un nodo en C – Función free()

- Consiste en liberar explícitamente la memoria que se encuentra direccionada a través de un puntero
- Requiere la biblioteca <stdlib.h>

```
free(nodo);

nodo=NULL;
```



tatiana.llabaca.upla@gmail.com - 012021

Estructuras de datos dinámicas

# Eliminación de un nodo en C – Ejemplo

- Al programa anterior, agrega el siguiente código:

```
printf("\nValor de la variable nodo antes de free: %p",nodo);

free(nodo);

printf("\nValor de la variable nodo despues de free: %p",nodo);

nodo=NULL; //Nunca olvidar!

printf("\nValor de la variable nodo despues de NULL: %p",nodo);
```



tatiana.llabaca.upla@gmail.com - 012021

# Actividad

- Desarrolla la Guía de Práctica 05 publicada en eAula



tatiana.llabaca.upla@gmail.com - 012021