

Análisis de Algoritmos y Estructura de Datos

TDA lista enlazada 2

Prof. Violeta Chang C

Semestre 2 – 2023



TDA Lista Enlazada

- **Contenidos:**

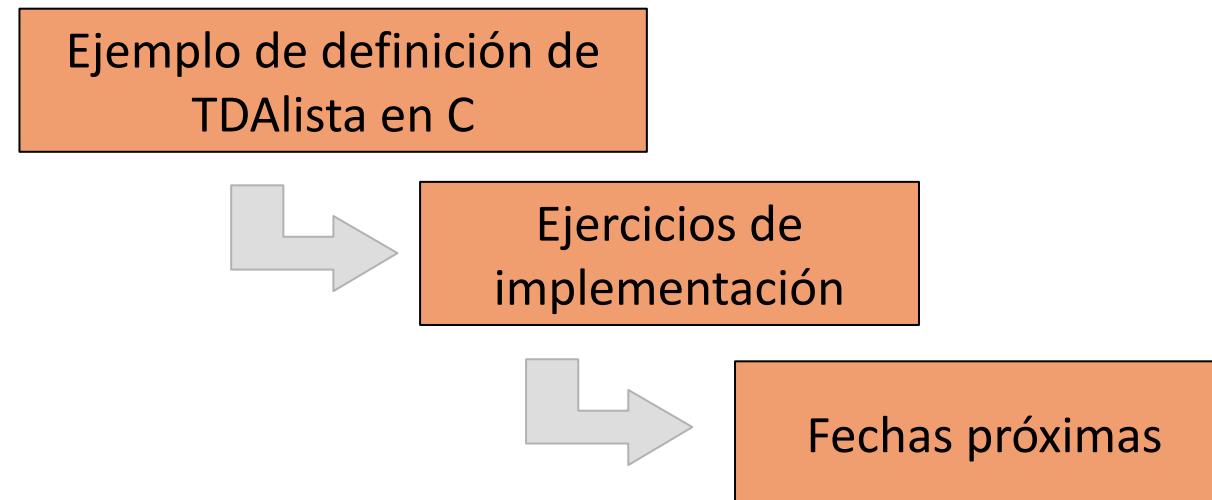
- Estructura de datos de TDA lista enlazada
- Operaciones de TDA lista enlazada

- **Objetivos:**

- Implementar con DESTREZA operaciones en TDA lista enlazada



Ruta de trabajo





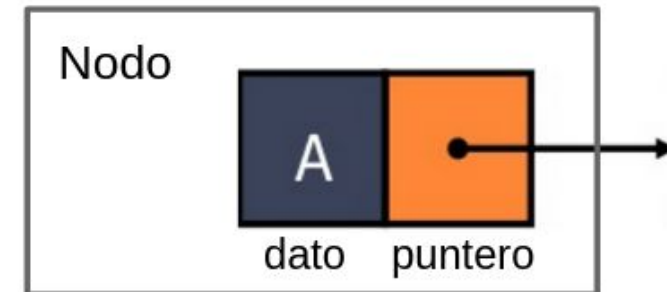
Especificación e implementación de TDA lista enlazada



Especificación de TDA lista doblemente enlazada

• Estructura de datos:

- Una lista enlazada (LE) es una secuencia de nodos conectados
- A una lista con 0 nodos se le conoce como **lista vacía**
- Cada nodo contiene:
 - Una parte de datos (cualquier tipo)
 - Un puntero al siguiente nodo de la lista
- **Cabeza**: puntero al primer nodo
- El último nodo apunta a **nulo**





Especificación de TDA lista enlazada

- **Operaciones:**

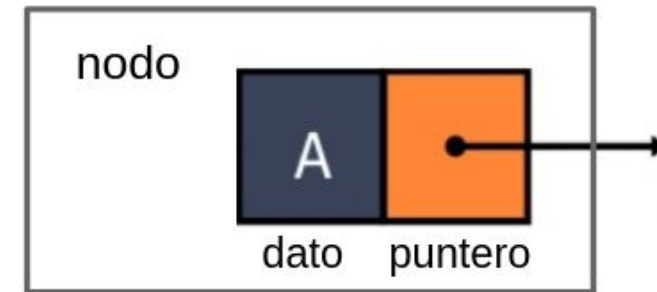
- **esListaVacía(L)**: determina si lista L está vacía o no
- **insertarNodo(L,dato)**: inserta nodo con *dato* en lista L
- **eliminarNodo(L,dato)**: elimina nodo con *dato* de lista L
- **buscarNodo(L,dato)**: busca nodo con *dato* en lista L
- **recorrerLista(L)**: muestra contenido de cada nodo de lista L



Implementación de estructura de datos de TDA lista enlazada

- La estructura de datos que representa un nodo de una lista enlazada simple es la siguiente:

```
typedef struct nodoGenerico  
{  
    int dato;  
    struct nodoGenerico* puntero;  
}nodo;
```

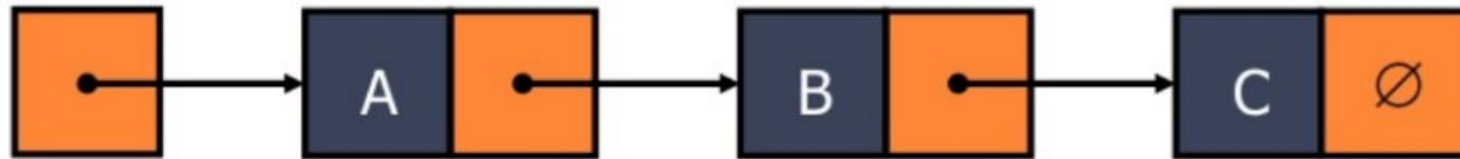




Implementación de estructura de datos de TDA lista enlazada

- La estructura de datos que representa una lista es la siguiente:

```
typedef nodo* TDAlista;
```



Actividades de implementación



Actividad 1 - individual

1. Implementar la siguiente función en **TDAlista.h** para eliminar todos los elementos de una lista enlazada simple. Después de **eliminar** cada elemento, lo que queda debe seguir siendo una lista enlazada.

```
void liberarLista(TDAlista* lista)
```

2. Evaluar las funciones creadas usando la lista de Actividad 1 de la semana5, generando secuencia de llamadas desde función *main()* en lab06-listaSimple.c para liberar dicha lista



Actividad 2 - individual

1. Implementar la siguiente función en **TDAlista.h** que devuelve una nueva lista enlazada *lista2*, tal que cada elemento de *lista2* sea la suma de sus vecinos en *lista1*

```
TDAlista sumaVecinos(TDAlista lista1)
```

2. Evaluar la función creada, generando secuencia de llamadas desde función *main()* en lab06-listaSimple.c para las siguientes listas:
 - 2, 3, 1, 4
 - 2, 3
 - 2



Actividad 3 - individual

1. Implementar la siguiente función en **TDAlista.h** que devuelve 1 en caso que lista1 y lista2 sean iguales, y 0 en caso contrario:

```
int compararListas(TDAlista lista1, TDAlista lista2);
```

2. Evaluar la función creada, generando secuencia de llamadas desde función *main()* en lab06-listaSimple.c para las siguientes listas:
 - lista1: 2, 3, 1, 4 – lista2: 2, 3, 1, 4
 - lista1: 2, 3, 1, 4 – lista2: 2, 3, 1
 - lista1: 2, 3, 1, 4 – lista2: 3, 1, 4



Actividad 4 - individual

1. Implementar la siguiente función en **TDAlista.h** que devuelve la lista con los elementos en orden invertido:

```
TDAlista listaInversa(TDAlista lista);
```

2. Evaluar las funciones creadas usando la lista de Actividad 1 de la semana5, generando secuencia de llamadas desde función *main()* en lab06-listaSimple.c



Actividad 5 - individual

1. Implementar la siguiente función en **TDAlista.h** que devuelve la lista modificada con los elementos en posición1 y posición2 intercambiados:

```
void elementosCambiados(TDAlista* lista, int posicion1, int posicion2);
```

2. Evaluar las funciones creadas usando la lista de Actividad 1 de la semana5, generando secuencia de llamadas desde función *main()* en lab06-listaSimple.c



Entrega de actividad de laboratorio

- Entrega obligatoria
- Subir todas las actividades de esta sesión en buzón de uVirtual, en único archivo **s6_apellido_nombre.zip**
- Se espera lab06-listaSimple.c y TDAlista.h (ambos modificados para responder a todas las actividades) comprimidos en archivo .zip
- Plazo: **hoy** dentro del horario de laboratorio de cada coordinación



Actividad de cierre



- Ir a [menti.com](https://www.menti.com) e ingresar código 5700 9171



Próximas fechas...

- Resumen de la semana:
 - TDA lista enlazada
 - TDA lista enlazada circular
 - TDA lista doblemente enlazada

U2 - S6

~~cátedra~~ – ~~refuerzo1~~ – ~~refuerzo2~~ – ~~lab1~~ – ~~lab2~~

- Próxima semana:
 - TDA pila
 - TDA cola

Octubre 2023						
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
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						

Noviembre 2023						
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Receso						
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		