



UPN, PASIÓN POR TRANSFORMAR VIDAS

UNIDAD 1: LISTAS, PILAS Y COLAS

SESIÓN 1: Listas enlazadas simples

Dr. Eric Gustavo Coronel Castillo
eric.coronel@upn.pe



**LA VIDA ES DISCIPLINA,
ORDEN Y SACRIFICIO.**

Robert Half

LOGRO DE LA UNIDAD 1



Al finalizar la unidad, el estudiante implementa algoritmos utilizando listas simples, listas dobles, listas circulares, pilas y colas, a través de control de versiones y el lenguaje C# con entorno gráfico, demostrando lógica y habilidad en la implementación de los algoritmos.

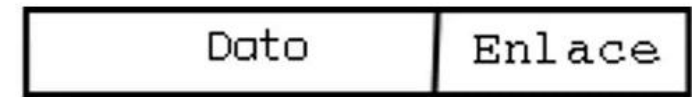


Al término de la sesión, el estudiante aprende algoritmos de listas enlazadas, como representarlos y usarlos con eficacia.

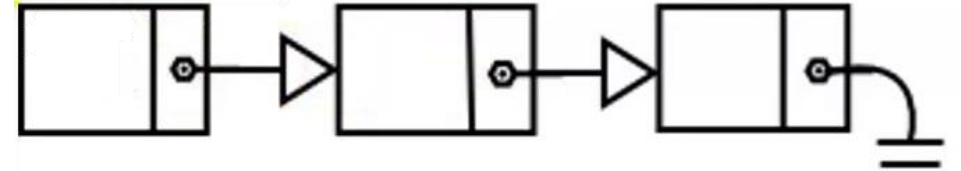
- Presentación del sílabo del curso.
- Listas enlazadas Simples: Operaciones de inserción, recorrido, eliminación.



1. ¿Qué es una lista enlazada?
2. ¿Qué es una estructura?
3. ¿Qué es un nodo?



Estructura de un nodo



Lista enlazada

REFLEXIONA



Una lista enlazada se compone de una serie de nodos enlazados mediante referencias.

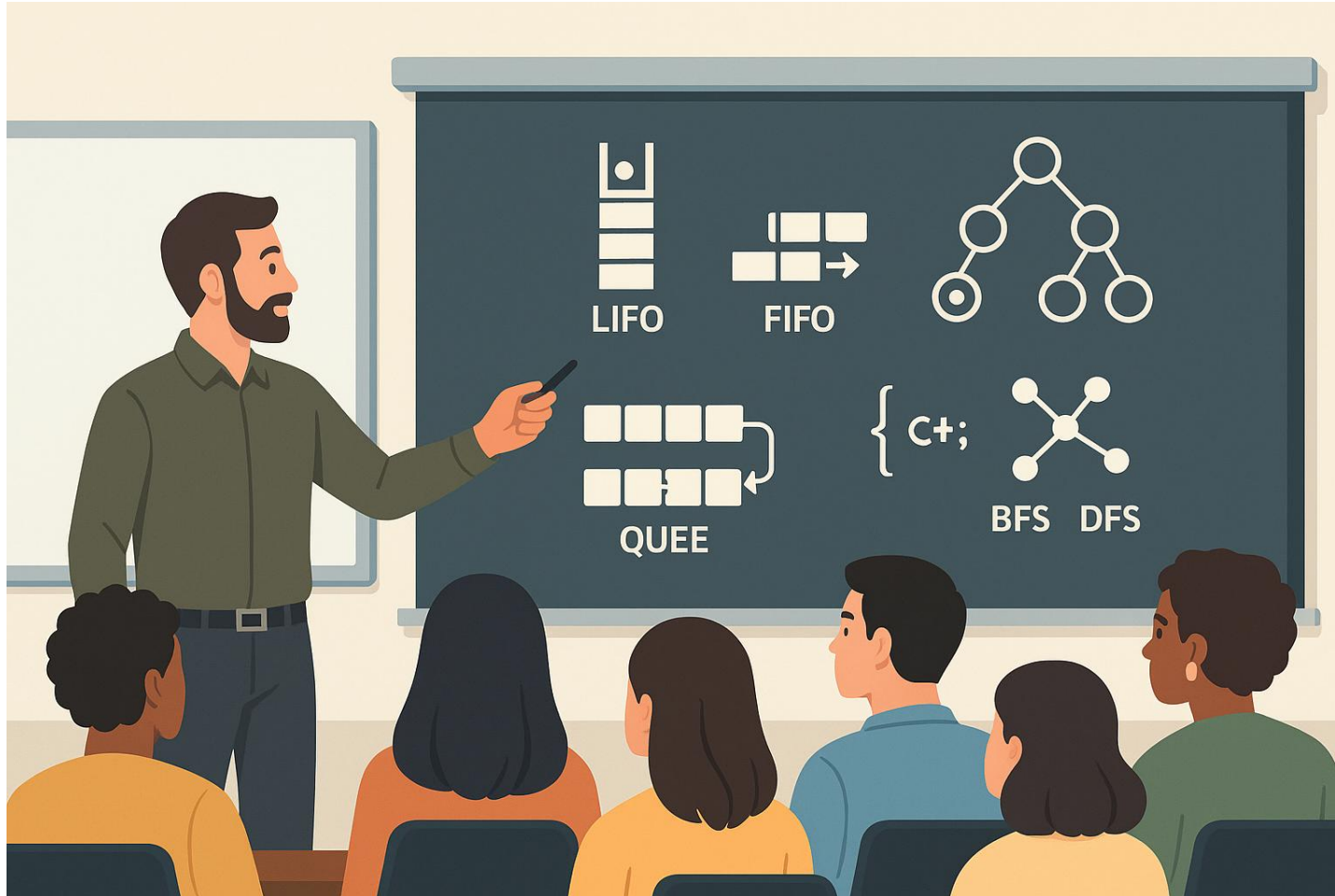


```
class Nodo
{
    int dato;
    Nodo enlace;
    public Nodo(int t)
    {
        dato = t;
        enlace = null;
    }
}
```

```
class Elemento
{
    //...;
}

class Nodo
{
    Elemento dato;
    Nodo enlace;
}
```

DESARROLLO





- El principal beneficio de las listas enlazadas respecto a los arreglos convencionales es que el orden de los elementos enlazados puede ser diferente al orden de almacenamiento en memoria o el disco, permitiendo que el orden de recorrido de la lista sea diferente al de almacenamiento.
- Las listas enlazadas permiten inserciones y eliminación de nodos en cualquier punto de la lista en tiempo constante, pero no permiten un acceso aleatorio.

CONCEPTOS

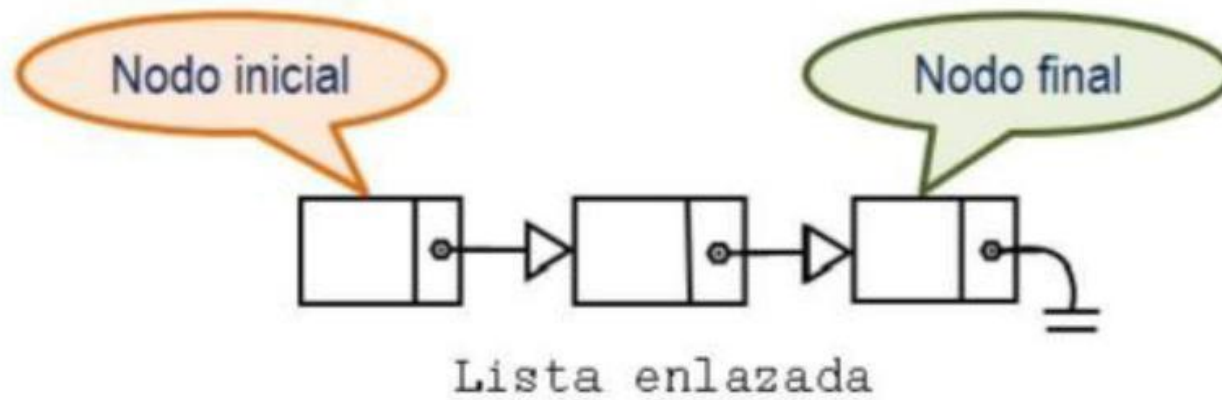


- **Lista enlazada:** Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias al nodo anterior y/o posterior.
- **Cabeza:** Al primer nodo de una lista enlazada
- **Fin:** Al último nodo de una lista enlazada.

¿QUÉ ES UNA LISTA ENLAZADA?



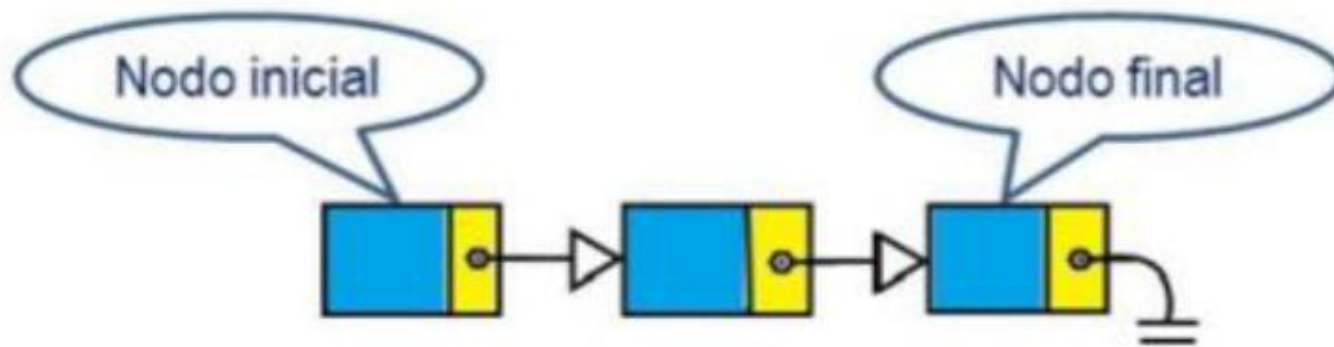
Una lista enlazada consta de un número de nodos con dos componentes(campos), un enlace al siguiente nodo de la lista y un valor, que puede ser de cualquier tipo.



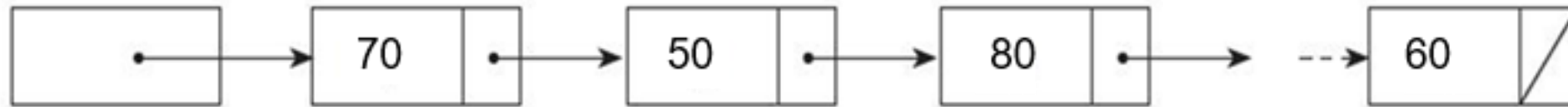
¿QUÉ ES UNA LISTA ENLAZADA?



Nodo: es la pieza fundamental de una lista enlazada, y se compone de dos partes: dato y enlace



¿QUÉ ES UNA LISTA ENLAZADA?



```
public class Nodo
{
    public int Dato { get; set; }
    public Nodo Siguiente { get; set; }

    public Nodo(int dato)
    {
        Dato = dato;
        Siguiente = null;
    }

    public Nodo(int dato, Nodo siguiente)
    {
        Dato = dato;
        Siguiente = siguiente;
    }
}
```

CLASIFICACIÓN DE LISTAS ENLAZADAS



Las listas se pueden dividir en cuatro categorías:

- Listas Simplemente Enlazadas
- Listas Doblemente Enlazadas
- Lista Circular Simplemente Enlazada
- Lista Circular Doblemente Enlazada

LISTAS SIMPLEMENTE ENLAZADAS



Cada nodo contiene un único enlace que conecta ese nodo al nodo siguiente o nodo sucesor. La lista es eficiente en recorridos directos (adelante).



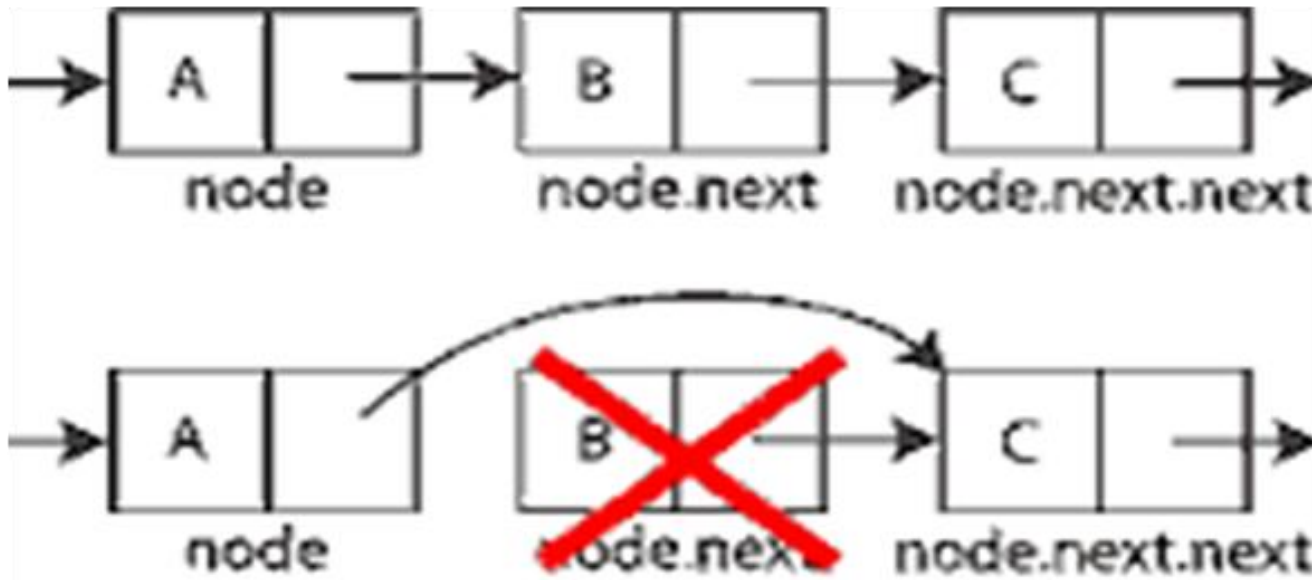
LISTAS SIMPLEMENTE ENLAZADAS



Tiene un enlace por nodo. Este enlace apunta al siguiente nodo en la lista, o al valor Nulo o lista Vacía, si es el último nodo.



LISTAS SIMPLEMENTE ENLAZADAS

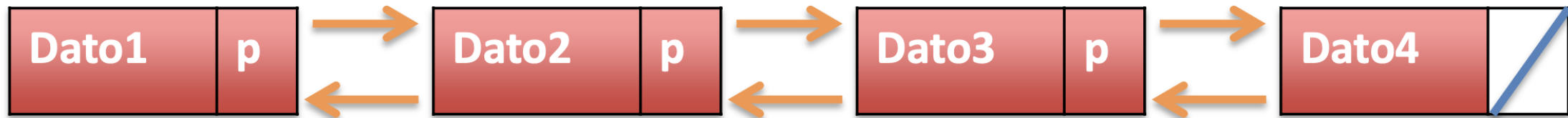


¿Qué es una lista enlazada?

LISTAS DOBLEMENTE ENLAZADAS



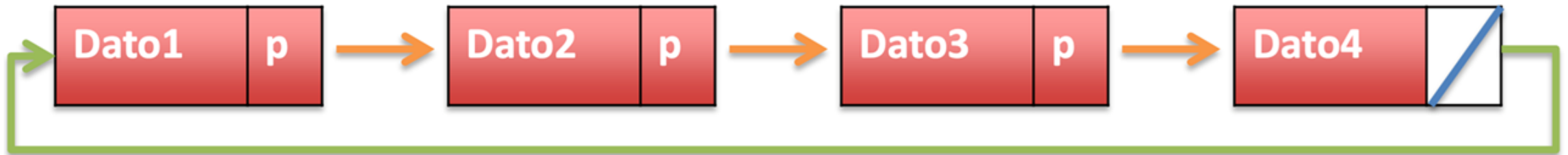
Cada nodo contiene dos enlaces, uno a su nodo predecesor y el otro a su nodo sucesor. La lista es eficiente tanto en recorrido directo (adelante) como en recorrido inverso (atrás).



LISTA CIRCULAR SIMPLEMENTE ENLAZADA



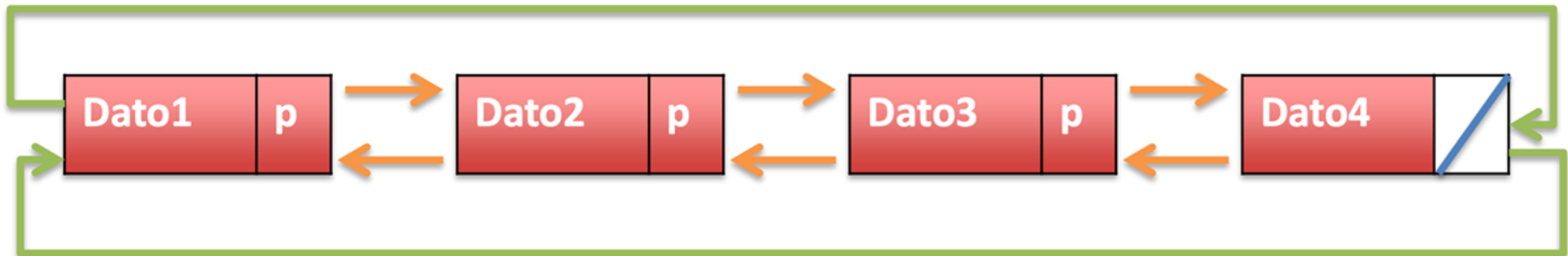
Una lista enlazada simplemente en la que el último elemento (cola) se enlaza al primer elemento (cabeza) de tal modo que la lista puede ser recorrida de modo circular (en anillo).



LISTA CIRCULAR DOBLEMENTE ENLAZADA



Una lista doblemente enlazada en la que el último elemento se enlaza al primer elemento y viceversa. Esta lista se puede recorrer de modo circular (en anillo) tanto en dirección directa (adelante) como inversa (atrás).



OPERACIONES EN LISTAS ENLAZADAS



- Insertar elementos en una lista enlazada
- Mostrar los elementos de una lista enlazada
- Buscar un elemento en una lista enlazada
- Eliminar un elemento en una lista enlazada

VENTAJAS SOBRE LOS ARREGLOS



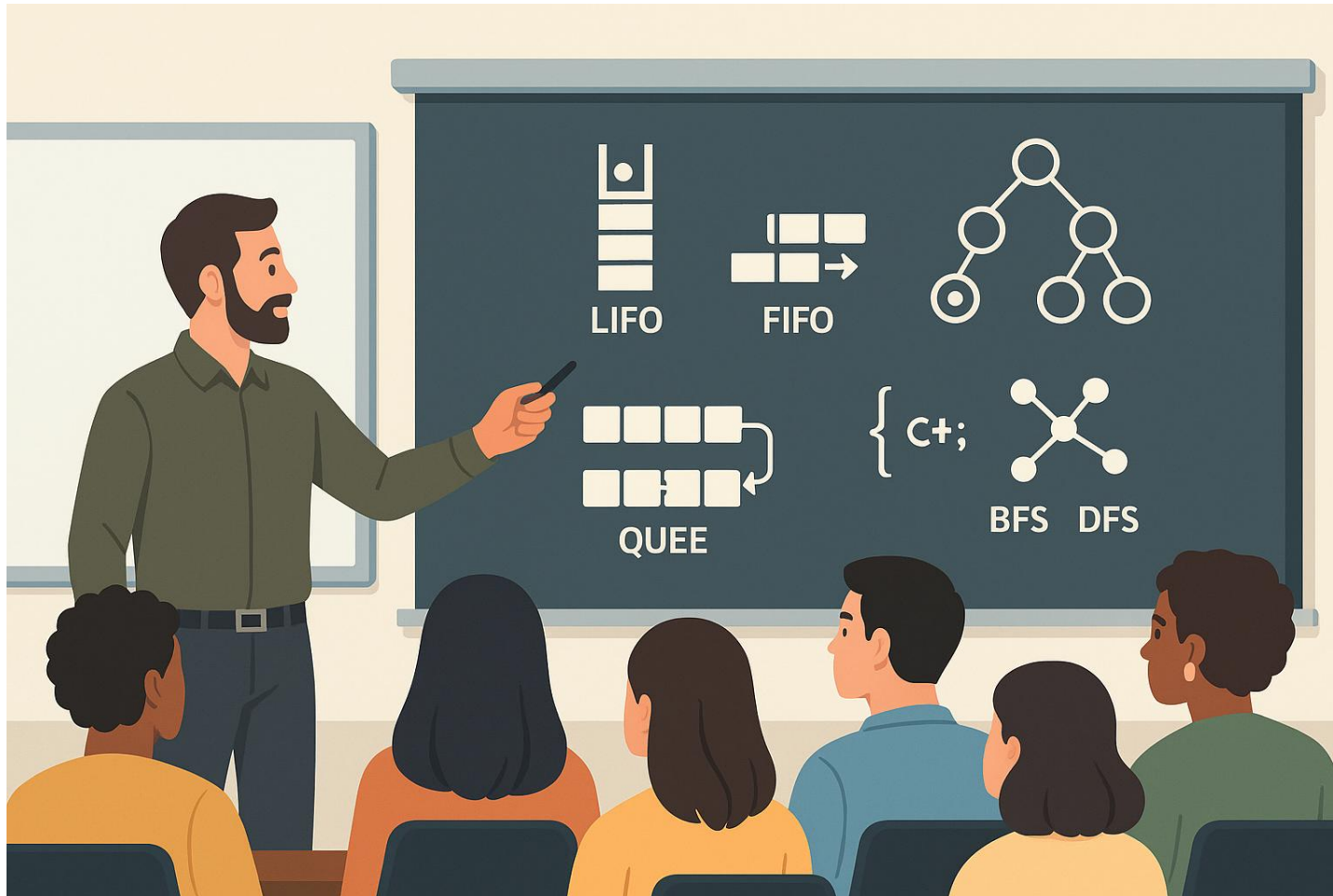
Los elementos se pueden insertar en una lista indefinidamente mientras que un arreglo tarde o temprano se llenará o necesitará ser redimensionado, una costosa operación que incluso puede no ser posible si la memoria se encuentra fragmentada.

DESVENTAJAS SOBRE LOS ARREGLOS



- Las listas son de acceso secuencial, y sólo puede ser recorridas en una dirección.
- El acceso secuencial es más lento en las listas.
- Se requiere de almacenamiento extra para las referencias
- Puede resultar lento asignar memoria para cada nuevo elemento.

LISTAS ENLAZADAS SIMPLES



BUSCAR UN ELEMENTO EN UNA LISTA



Para buscar un elemento en una lista, sólo hay que seguir 4 pasos:

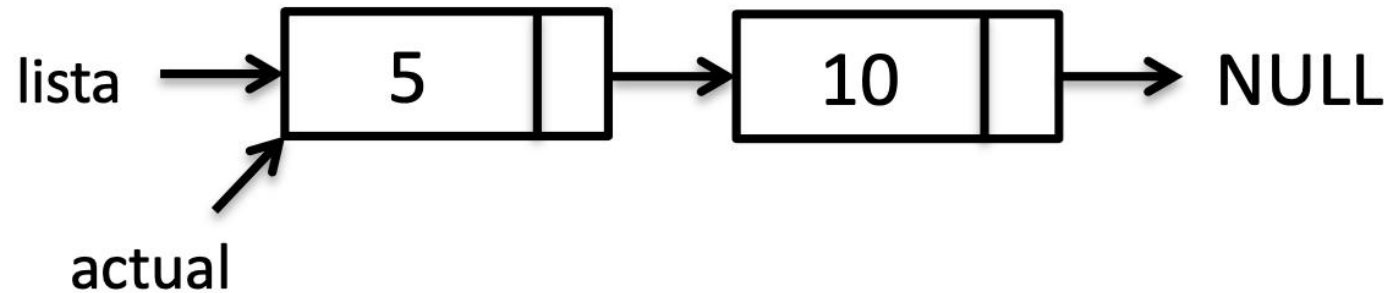
1. Crear un nuevo nodo(actual).
2. Igualar ese nuevo nodo(actual) a la lista.
3. Recorrer la lista.
4. Determinar si el elemento existe o no en la lista.

1. CREAR UN NUEVO NODO(ACTUAL).



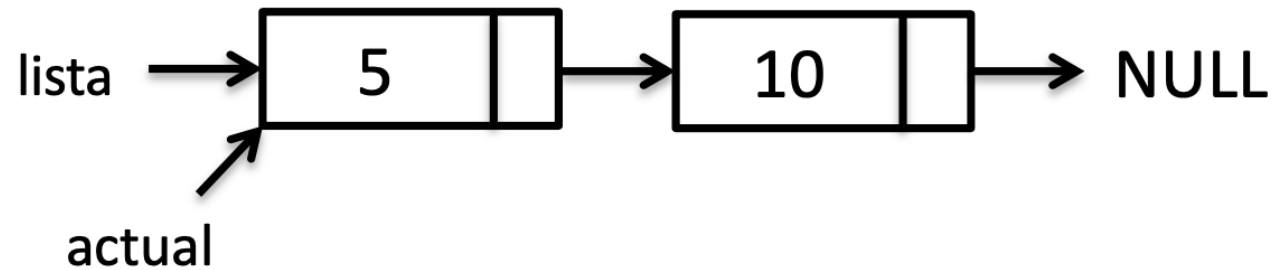
```
void buscarLista(Nodo *lista, int n){  
    Nodo *actual = new Nodo();  
}
```

2. IGUALAR ESE NUEVO NODO(ACTUAL) A LA LISTA



```
void buscarLista(Nodo *lista, int n){  
    Nodo *actual = new Nodo();  
    actual = lista;  
}
```

3. RECORRER LA LISTA

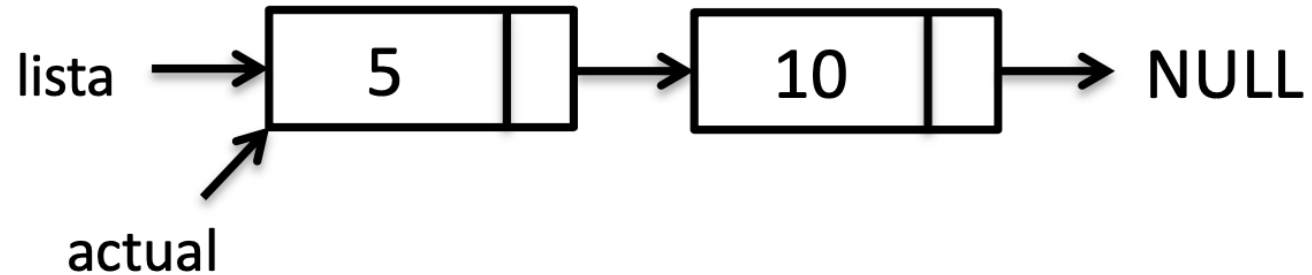


n = 10;

```
while((actual != NULL) && (actual->dato <= n)){
```

```
}
```

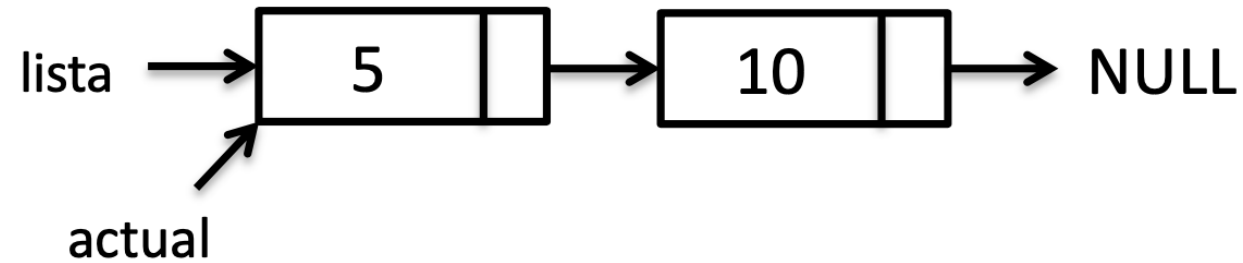
4. DETERMINAR SI EL ELEMENTO EXISTE O NO EN LA LISTA



n = 10;

```
while((actual != NULL) && (actual->dato <= n)){  
    if(actual->dato == n){  
        band = true;  
    }  
    actual = actual->siguiente;  
}
```

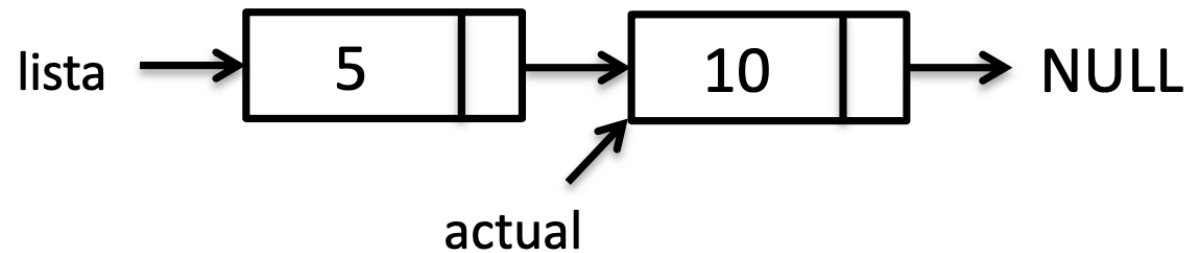
4. DETERMINAR SI EL ELEMENTO EXISTE O NO EN LA LISTA.



n = 12;

```
while((actual != NULL) && (actual->dato <= n)){  
    if(actual->dato == n){  
        band = true;  
    }  
    actual = actual->siguiente;  
}
```

4. DETERMINAR SI EL ELEMENTO EXISTE O NO EN LA LISTA.



n = 12;

```
while((actual != NULL) && (actual->dato <= n)){  
    if(actual->dato == n){  
        band = true;  
    }  
    actual = actual->siguiente;  
}
```

ACTIVIDAD EN CLASE



PROBLEMA



Implemente una lista de contactos que permita:

- Insertar un nuevo contacto
- Listar todos los contactos
- Buscar un contacto

De cada contacto se debe guardar su nombre y su celular.

CONCLUSIONES



- Las listas simplemente enlazadas crecen dinámicamente sin requerir memoria contigua y permiten inserciones y eliminaciones, pero añaden sobrecosto de punteros y no ofrecen acceso aleatorio.
- Por su acceso secuencial, son adecuadas como base de pilas, colas y listas de adyacencia en grafos, pero resultan poco eficientes cuando se necesita indexación directa frecuente.

¿PREGUNTAS O COMENTARIOS?





Cuestionario en la plataforma

Nivel	Rango
Nivel 4	17 – 20
Nivel 3	13 – 16
Nivel 2	9 – 12
Nivel 1	0 - 8

Comprobación del Logro

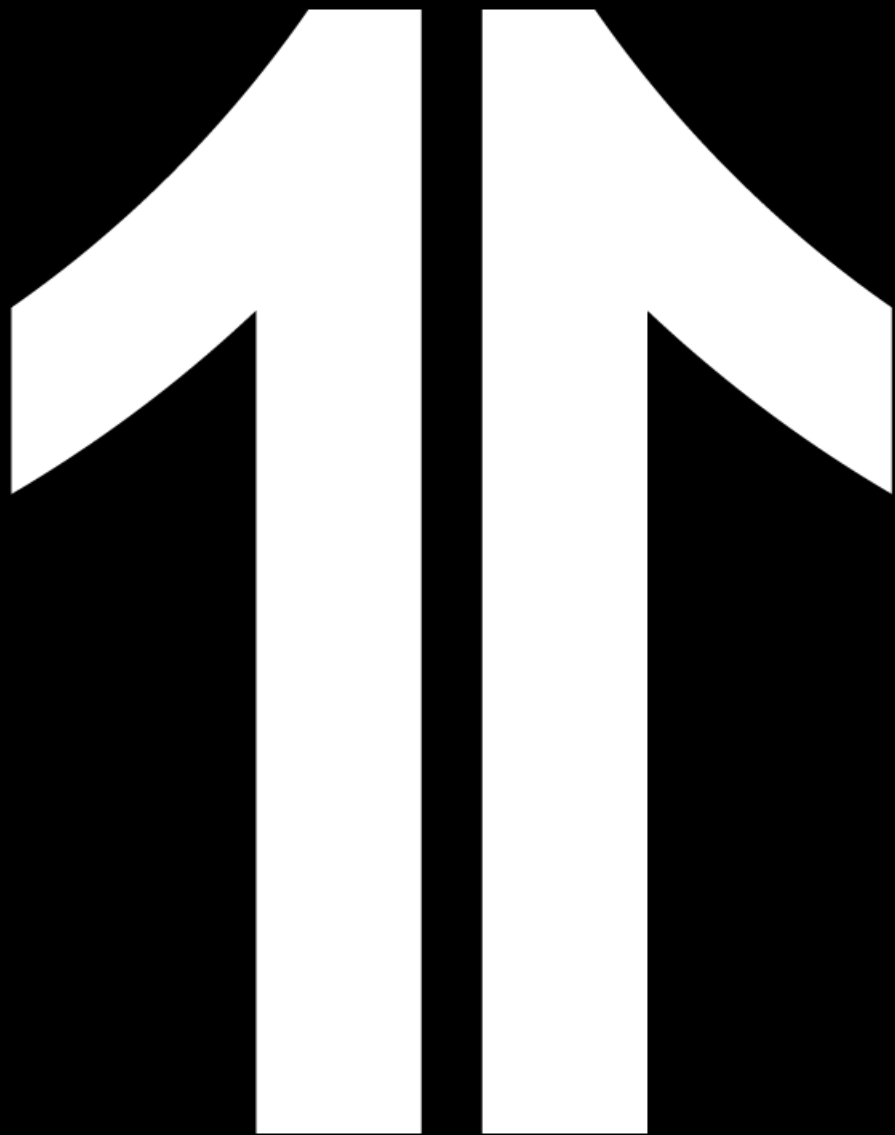


Criterio	Nivel 4 (Excelente)	Nivel 3 (Bueno)	Nivel 2 (Regular)	Nivel 1 (Deficiente)
Comprensión conceptual	Demuestra comprensión completa de los conceptos clave.	Comprende la mayoría de los conceptos clave.	Muestra comprensión parcial, con algunas confusiones.	No demuestra comprensión clara de los conceptos.
Aplicación de conocimientos	Aplica correctamente los conceptos en ejemplos o escenarios.	Aplica conceptos con algunos errores menores.	Aplica de forma superficial o con errores evidentes.	No logra aplicar los conceptos o lo hace incorrectamente.
Claridad y precisión en respuestas	Las respuestas son claras, bien estructuradas y precisas.	Las respuestas son mayormente claras, con leves imprecisiones.	Respuestas poco claras o con ideas mal organizadas.	Respuestas confusas, incompletas o incoherentes.
Cobertura del contenido esperado	Responde completamente las 5 preguntas, desarrollando bien cada una.	Responde correctamente 4 de las 5 preguntas.	Responde correctamente 2 o 3 preguntas.	Responde solo 1 pregunta correctamente o no responde.

BIBLIOGRAFIA REFERENCIAL



- Ceballos Sierra, F. Microsoft C#: Curso de Programación (2a.ed.) 2014
<https://elibronet.eu1.proxy.openathens.net/es/lc/upnorte/titulos/106417>
- Cesar Liza Avila; Estructura de datos con C/C++



GRACIAS

