UAEM | Universidad Autónoma del Estado de México

Ingeniería en Computación

"2015, Año del Bicentenario Luctuoso de José María Morelos y Pavón"

Unidad de Aprendizaje:

Estructura de Datos

Unidad de Competencia II:

Estructuras de Datos Lineales

M. en C. Edith Cristina Herrera Luna

Marzo 2015

ESTRUCTURAS DE DATOS

Propósito de la Unidad de Aprendizaje

- Conocer, analizar y aplicar estructuras de datos estáticas y dinámicas mediante programas para la solución de problemas informáticos.

Estructuras de Datos
INTRODUCCIÓN

- El estudio de las estructuras de datos, sin duda es uno de los más importantes dentro de las carreras relacionadas con la computación, ya que el conocimiento eficiente de las estructuras de datos suele ser imprescindible en la formación de los alumnos debido a la trascendencia que un aprendizaje teórico-práctico de las mismas supondrá para su carrera.

- En muchas ocasiones se necesitan estructuras que puedan cambiar de tamaño durante la ejecución del programa. Por supuesto, se pueden usar arreglos dinámicos, pero una vez creados, su tamaño también será fijo, y para hacer que crezcan o disminuyan de tamaño, deberán reconstruirse desde el principio.

- Las estructuras dinámicas permiten crear estructuras de datos que se adapten a las necesidades reales a las que suelen enfrentarse los programas. Pero no solo eso, también permitirá crear estructuras de datos muy flexibles, ya sea en cuanto al orden, la estructura interna o las relaciones entre los elementos que las componen.

Estructuras de Datos Lineales

Unidad de Competencia II

OBJETIVO:

- Desarrollar programas que incorporen las principales estructuras de datos lineales: Pila, Cola y Lista enlazada.

Estructuras de Datos Lineales
CONTENIDO

1. Pila – Stack
   - ¿Qué es una pila?
   - Representación Gráfica
   - Operaciones
   - Aplicaciones

2. Cola – Queue
   - ¿Qué es una cola?
   - Tipos y representación Gráfica
   - Operaciones
   - Aplicaciones

Estructuras de Datos Lineales
CONTENIDO

3. Lista – List
   - ¿Qué es un nodo?
   - ¿Qué es una lista?
   - Tipos y representación Gráfica
   - Operaciones
   - Aplicaciones

¿Qué es una Pila?

C. U. UAEM ZUMPANGO / ICO / ED / ECHL

¿Qué es una PILA?

- Una ***pila*** o ***stack*** es una estructura de datos lineal en la que los datos son agregados y eliminados únicamente por un extremo de la estructura.

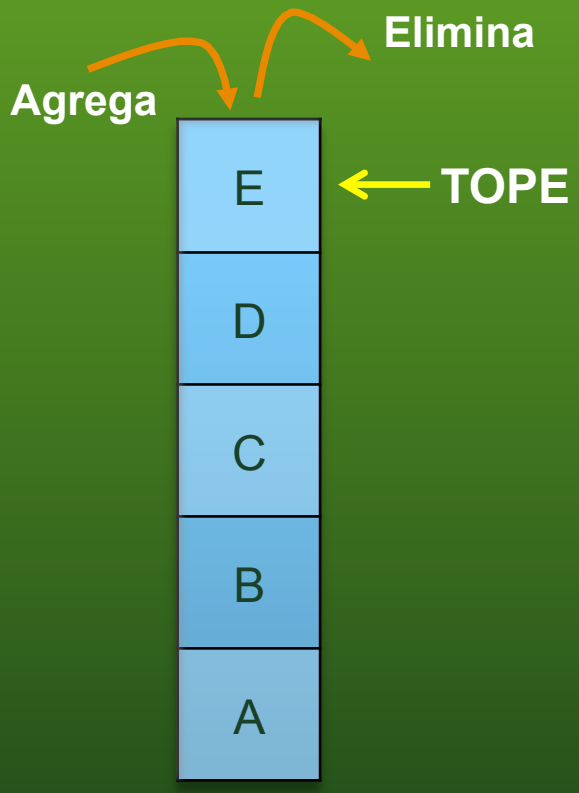
Son estructuras tipo **LIFO**:

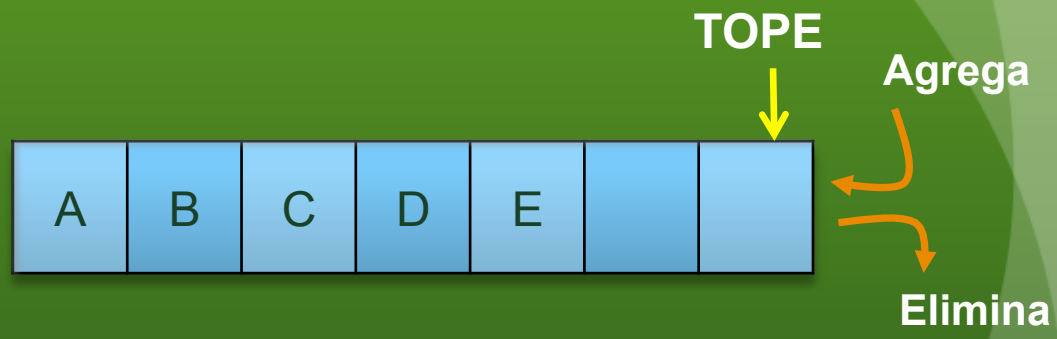
Last In, First Out. - “Ultimo en entrar, primero en salir”

Consta de un tope y no permite el uso de índices como un arreglo, aunque las pilas pueden ser implementadas a partir de ellos y de estructuras tipo lista.

Representación Gráfica

Una pila generalmente se representa gráficamente de forma vertical, sin embargo, el TOPE en la estructura es una señal para orientar su funcionamiento:





El TOPE (TOP) en una pila puede indicar:

- El ultimo dato agregado a la estructura
- El limite o borde del contenedor

Operaciones

Las operaciones permitidas sobre una pila son: agregar, eliminar, consultar el tope, pila vacía.

- **PUSH** - Apilar: Se agrega un dato al tope de la pila.

Ejemplo: En una pila de números enteros:

push(8) , push(50) , push (-14) , push (39)

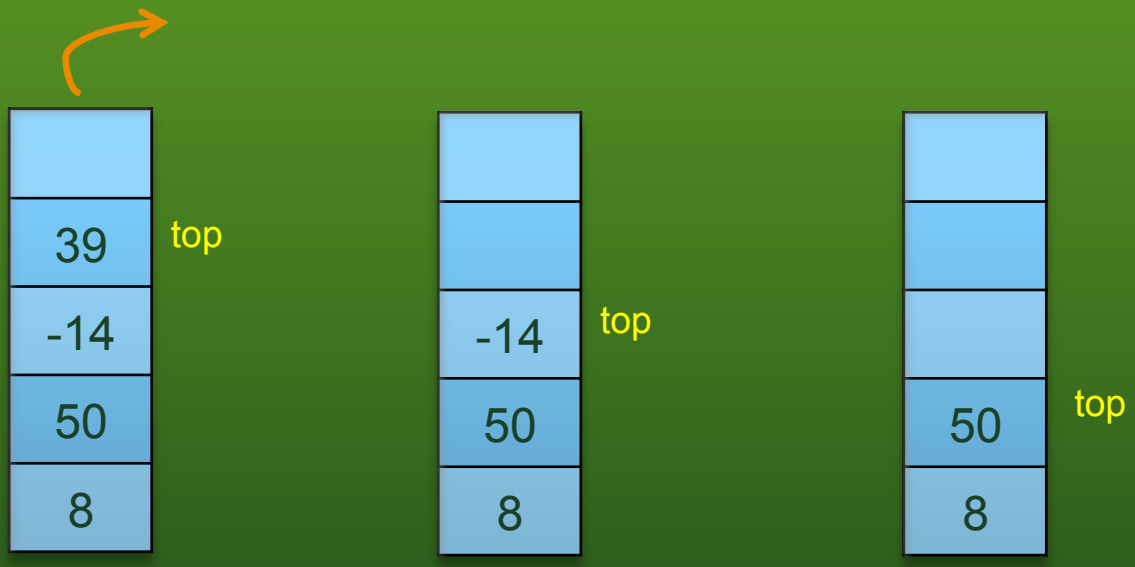
Operaciones

- **POP** - Des apilar: Se elimina el tope de la pila.

Ejemplo:

pop ( ) ,

pop ( )



Operaciones

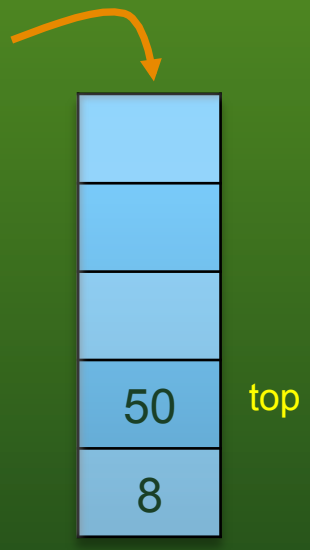
- **TOP** : Indica cual es el tope de la pila.

  No se producen cambios. (Consulta)

Ejemplo:

top ( )

El tope de la pila es 50



- **EMPTY** - Pila Vacía: Indica si la pila esta vacia

Ejemplo:

empty ( )

La pila esta vacia = false

Aplicaciones

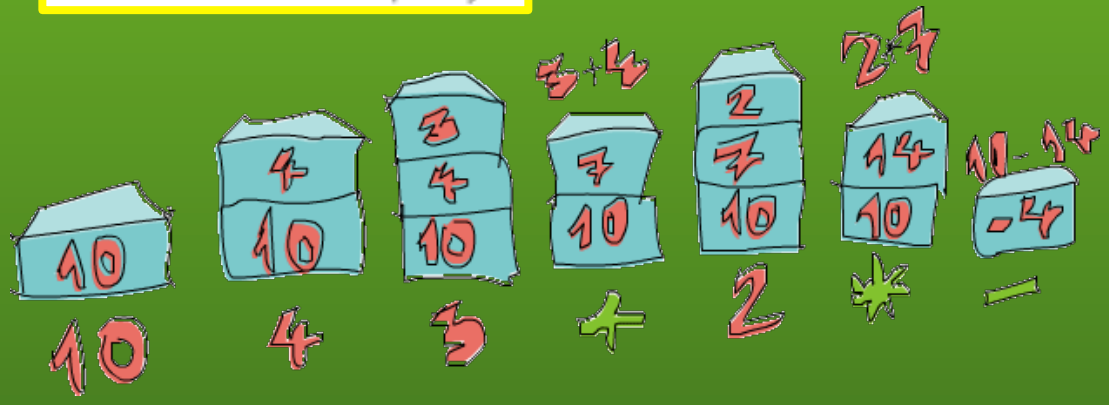
Las pilas se usan en:

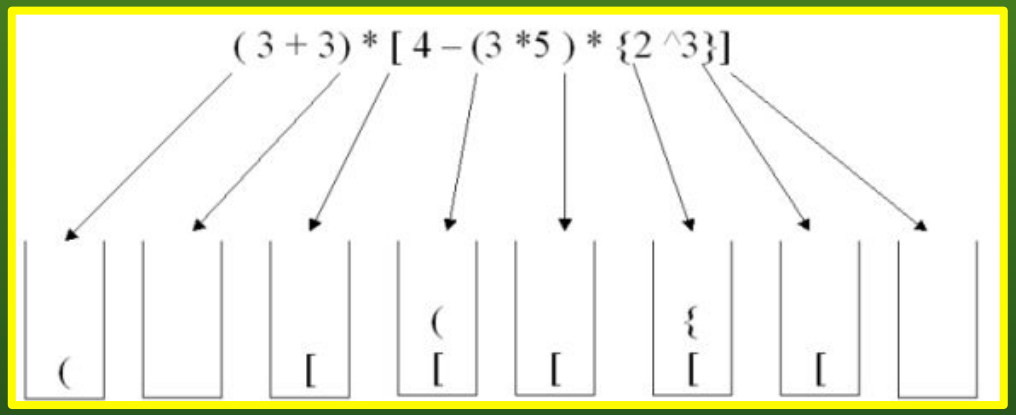
- Administración de llamadas a funciones.
- Equilibrio de paréntesis (corchetes y llaves) en expresiones.
- Pilas de recursividad.
- Equivalencias entre notaciones infijas, postfijas y prefijas.
- Historiales de cambios (deshacer).
- Torres de Hanoi, etc.

Una estructura tipo pila es util cuando el orden de los datos se necesita invertir, porque es el orden natural de su funcionamiento.

Aplicaciones

+ A B	Notación prefija
A + B	Notación infija
A B +	Notación postfija





¿Qué es una Cola?

C. U. UAEM ZUMPANGO / ICO / ED / ECHL

¿Qué es una COLA?

- Una ***cola*** o ***queue*** es una estructura de datos lineal en la que los datos son agregados por un extremo y eliminados por el extremo contrario.

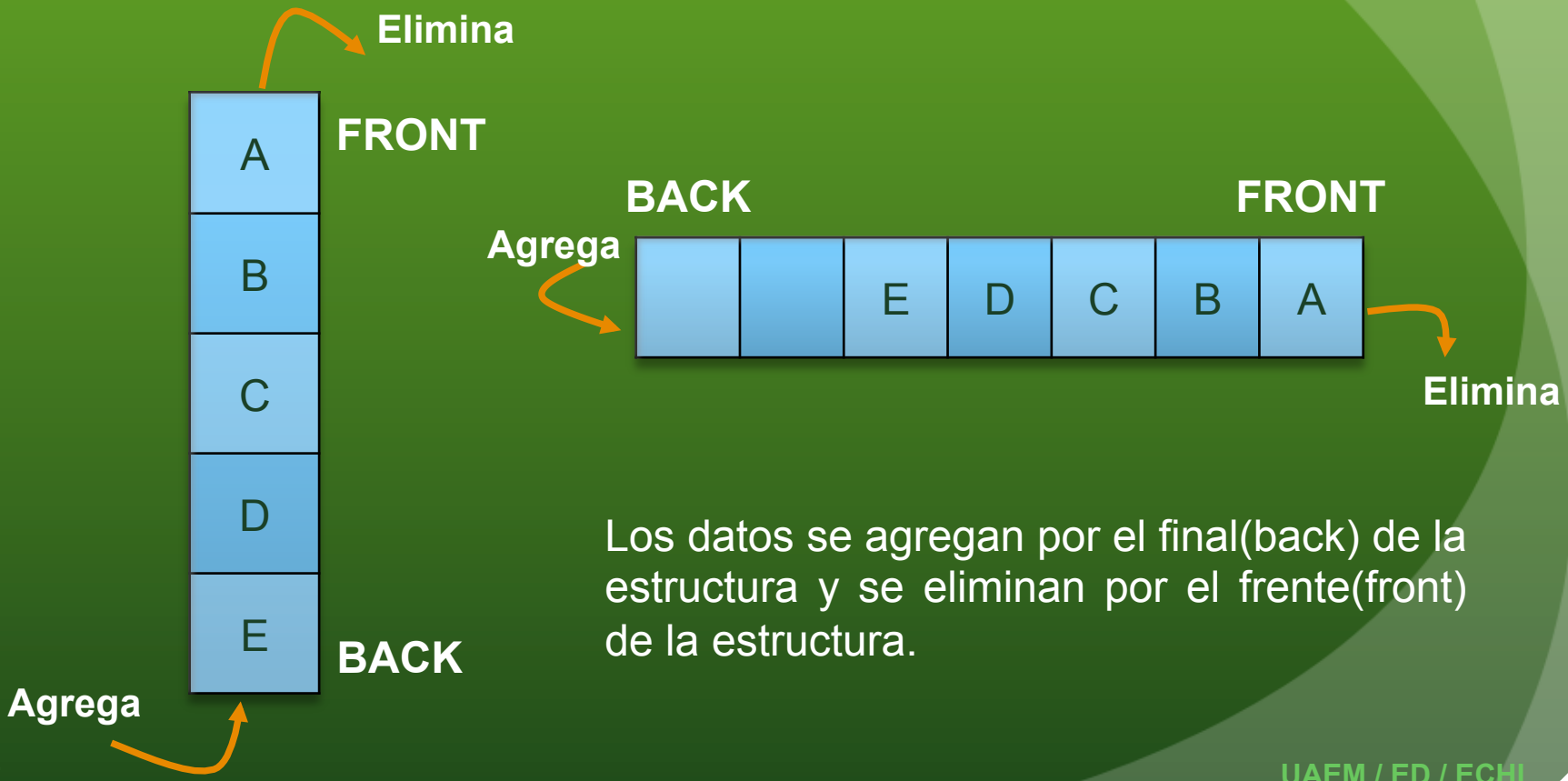
Son estructuras tipo **FIFO**:

First In, First Out. - “Primero en entrar, primero en salir”

Se tiene un extremo frontal y uno final (front / back), no permite el uso de índices como un arreglo, aunque las pilas pueden ser implementadas a partir de ellos y de estructuras tipo lista.

Representación Gráfica

Una cola generalmente se representa gráficamente de forma horizontal, sin embargo, el frente (front) de la estructura es una señal para orientar su funcionamiento:



Los datos se agregan por el final(back) de la estructura y se eliminan por el frente(front) de la estructura.

Tipos de Colas

Existen diferentes tipos de estructura cola:

- Cola simple — Queue
- Cola circular
- Cola de prioridad — Priority queue
- Bicola — Deque
  - Bicola de salida restringida
  - Bicola de entrada restringida

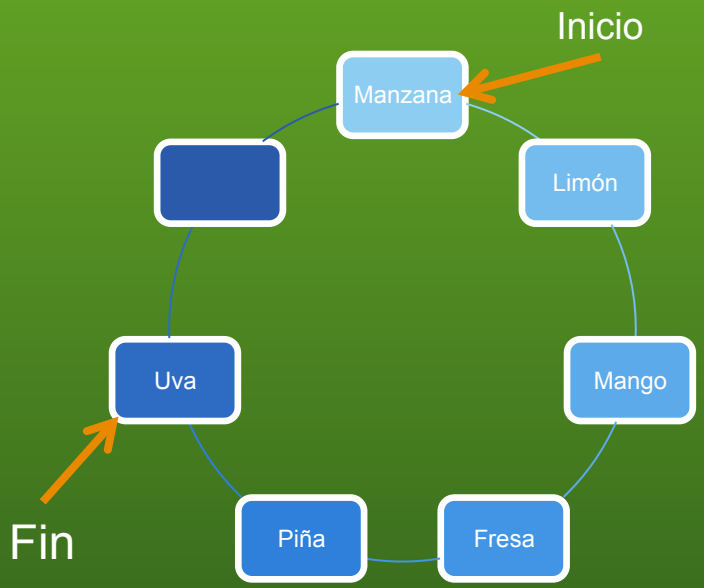
Cola Circular

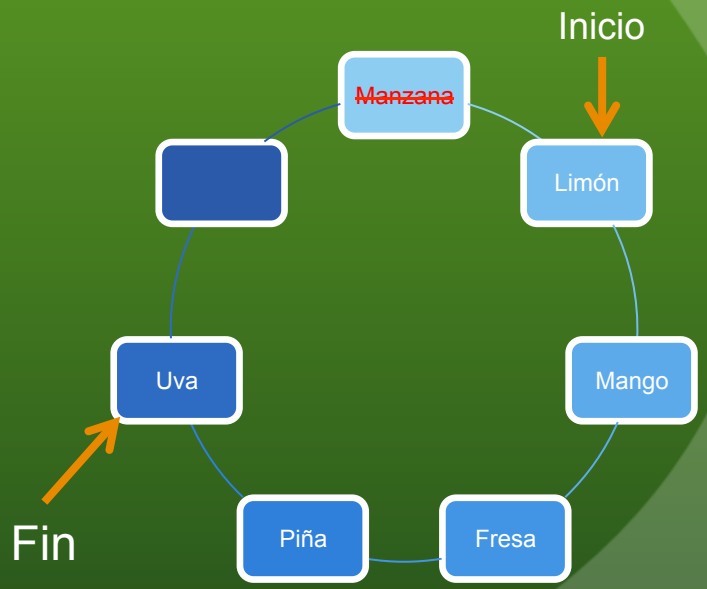
- El ultimo elemento de la estructura esta “ligado” o enlazado con el primer elemento de la misma.

- Generalmente se tiene un numero especifico de datos que se pueden almacenar en la estructura y apuntadores al inicio y final de la misma.

Cola Circular

Cuando un elemento se agrega se coloca al final de la estructura si hay espacio y se recorre el puntero final.





Cuando un elemento se elimina el puntero de inicio recorre la estructura por lo tanto libera un lugar.

Cola de Prioridad

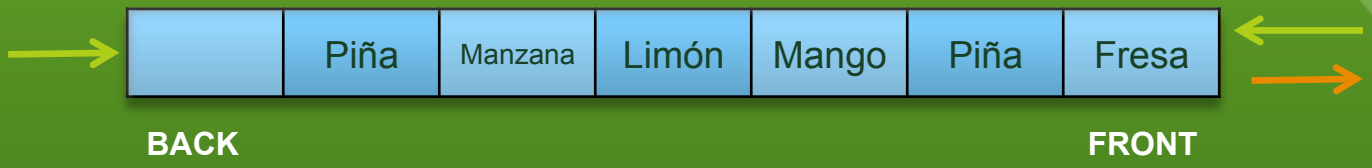
Los elementos se agregan al final de la estructura pero el elemento que sale o el primero es el de mayor importancia o prioridad.

Generalmente los elementos no están ordenados dentro de la estructura, sino que el primer elemento (y segundo en ocasiones) es el que tiene mayor prioridad.

Si existen dos o mas elementos con la misma prioridad se toman conforme se agregan a la estructura, como si fuera una cola simple.

Bicola

- Bicola de salida restringida
  - Se agregan elementos por ambos extremos de la estructura, pero se eliminan solo por el inicio (front).



- Bicola de entrada restringida
  - Se eliminan elementos por ambos extremos de la estructura, pero se agregan solo por el final (back).

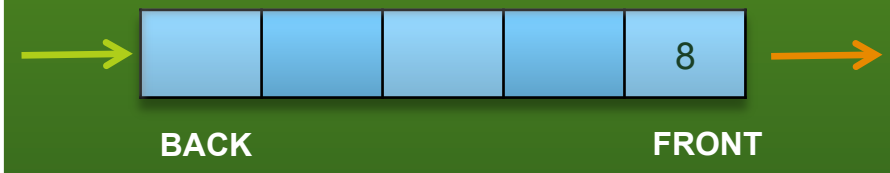
Operaciones

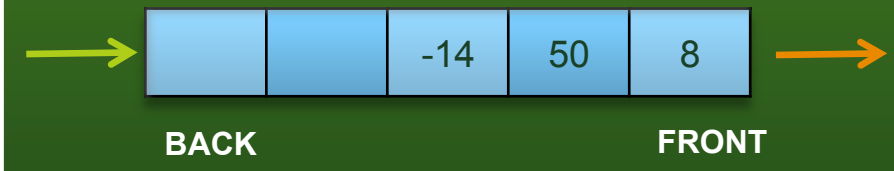
Las operaciones permitidas sobre una cola son: agregar, eliminar, consultar el primer y ultimo elemento, cola vacía.

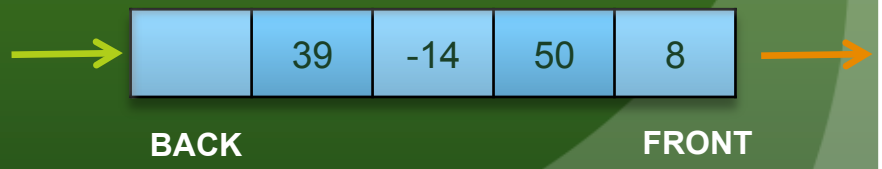
- **PUSH** - Agregar: Se agrega un dato al final de la cola.

Ejemplo: En una cola de números enteros:

push(8) , push(50) , push (-14) , push (39)



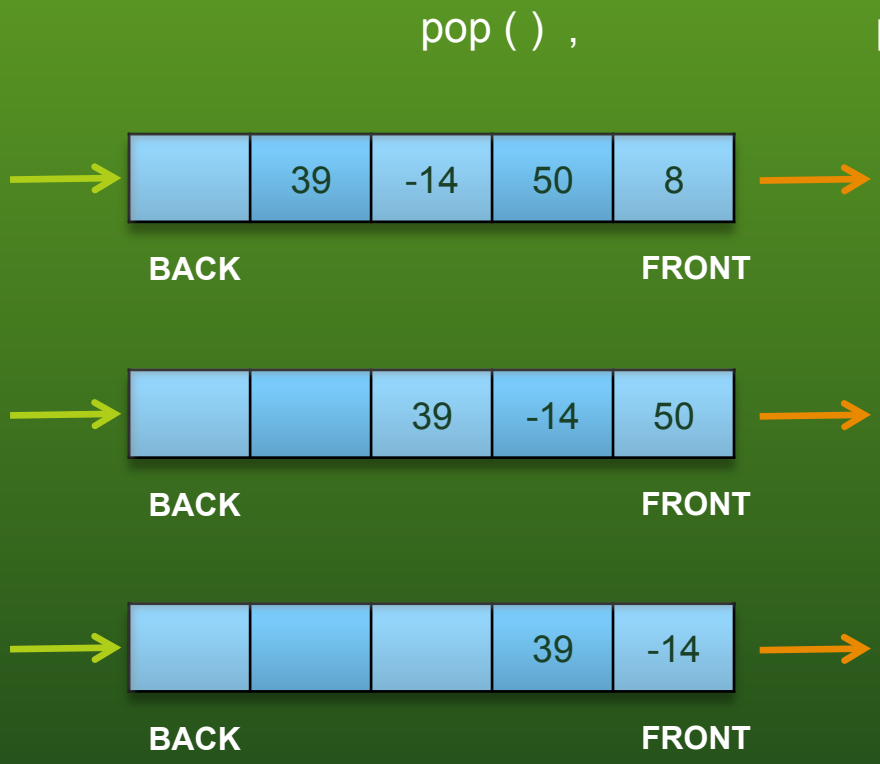




Operaciones

- **POP** - Quitar: Se elimina o quita el primer elemento de la estructura, el primero que se agrego.

Ejemplo:



Operaciones

- **FRONT** : Indica cual es el primer elemento en la estructura.
- **BACK** : Indica cual es el ultimo elemento en la estructura.

  No se producen cambios (Consulta).

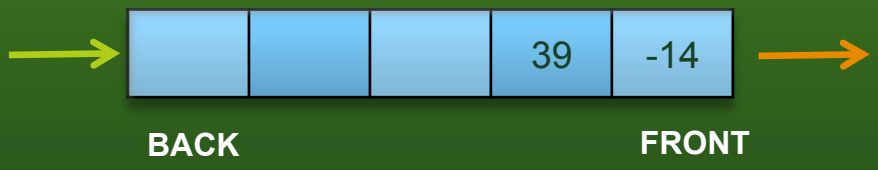
Ejemplo:

front ( )

El inicio de la cola es -14

back ( )

El final de la cola es 39



- **EMPTY** - Cola Vacía: Indica si la estructura no contiene elementos

Ejemplo:

empty ( )

La cola esta vacia = false

Aplicaciones

Las estructuras tipo cola se usan en:

- Organización de archivos de impresión (colas de impresión).
- Orden de datos.
- Aplicaciones sobre filas de elementos.
- Visores de datos, imágenes, archivos.
- Atención de elementos por prioridades, etc.

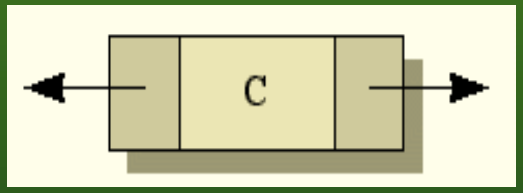
¿Qué es una Lista?

C. U. UAEM ZUMPANGO / ICO / ED / ECHL

¿Qué es un Nodo?

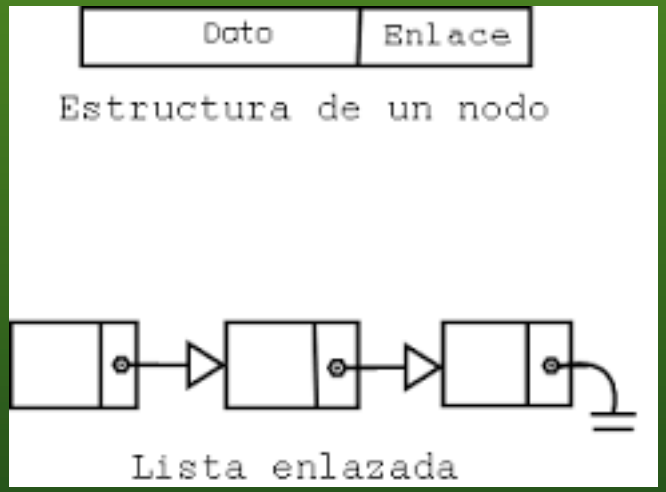
- Un nodo es una estructura que agrupa un conjunto de datos o campos arbitrarios, donde por lo menos un campo es de referencia o enlace a un nodo del mismo tipo (autorreferenciado).

Campo de Datos Campo de Enlace



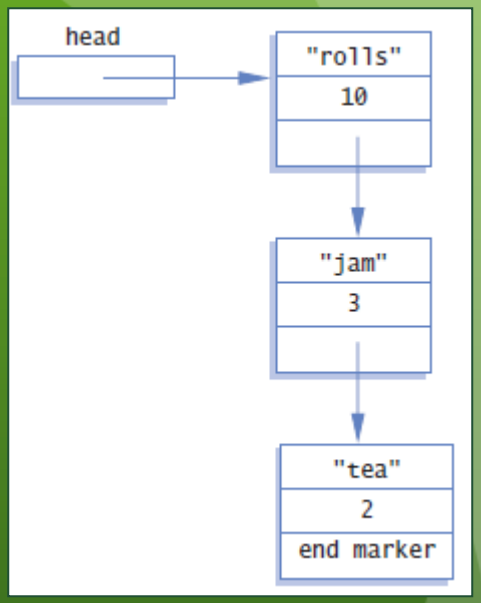
¿Qué es una Lista?

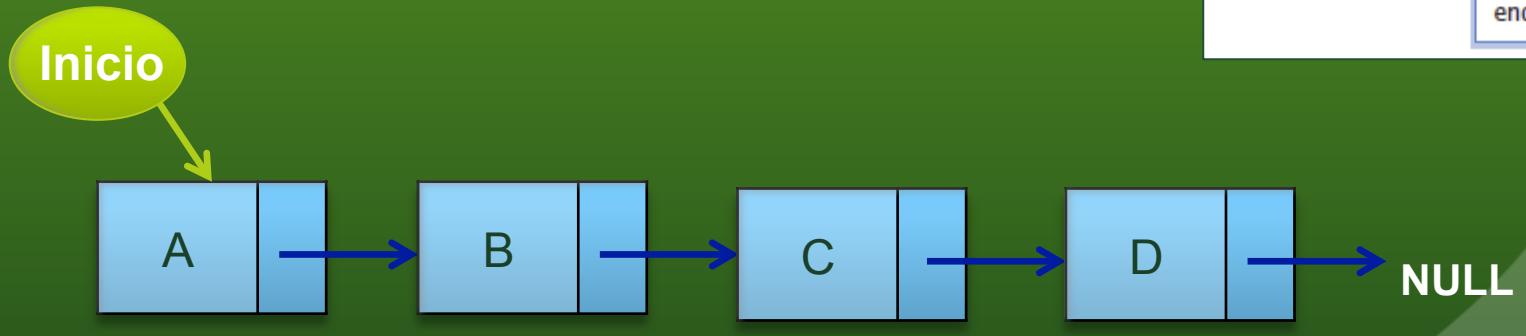
- Es una secuencia de nodos autorreferenciados con una o dos referencias al nodo anterior y/o siguiente.
- Una lista enlazada (ligada o abierta) es una de las estructuras de datos fundamentales y puede se usada para implementar otras estructuras de datos, como pilas y colas.



Representación Gráfica

Una lista generalmente se representa gráficamente de forma horizontal, cada nodo cuenta con una división marcada de los datos y los enlaces. Generalmente tiene un puntero especial al inicio de la lista, y en algunas ocasiones se tiene un puntero que indica el final de la estructura





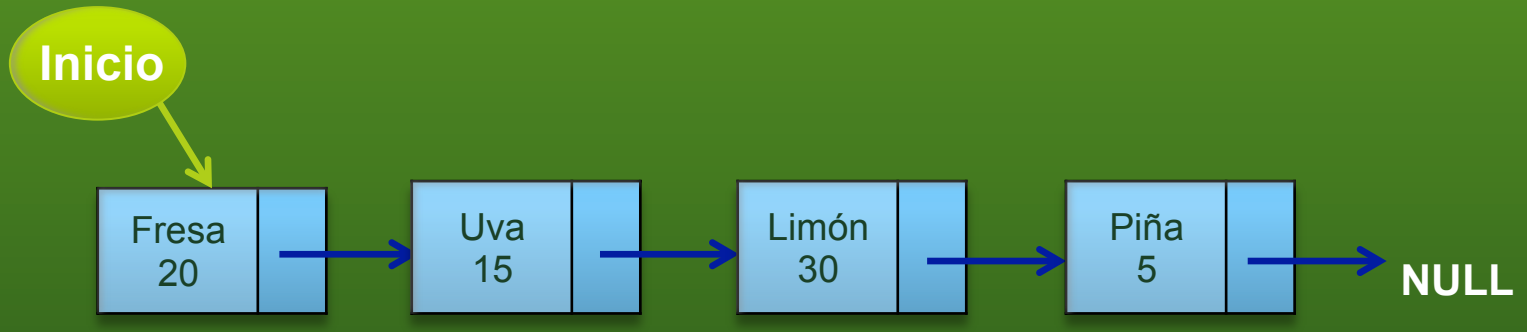
Tipos de Listas

Existen diferentes tipos de estructura lista:

- Lista simplemente enlazada
- Lista doblemente enlazada
- Lista circular simple
- Lista circular doble

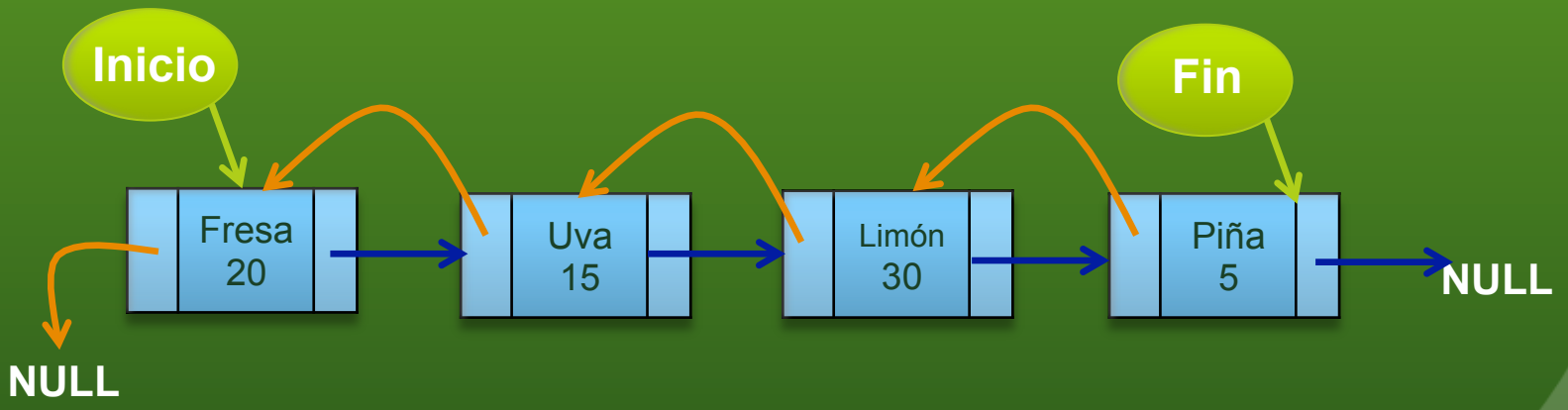
Lista simplemente enlazada

- Cada nodo de la estructura tiene un único campo de enlace que apunta al siguiente nodo en la lista.
- El ultimo nodo en la lista apunta NULL (vacio).



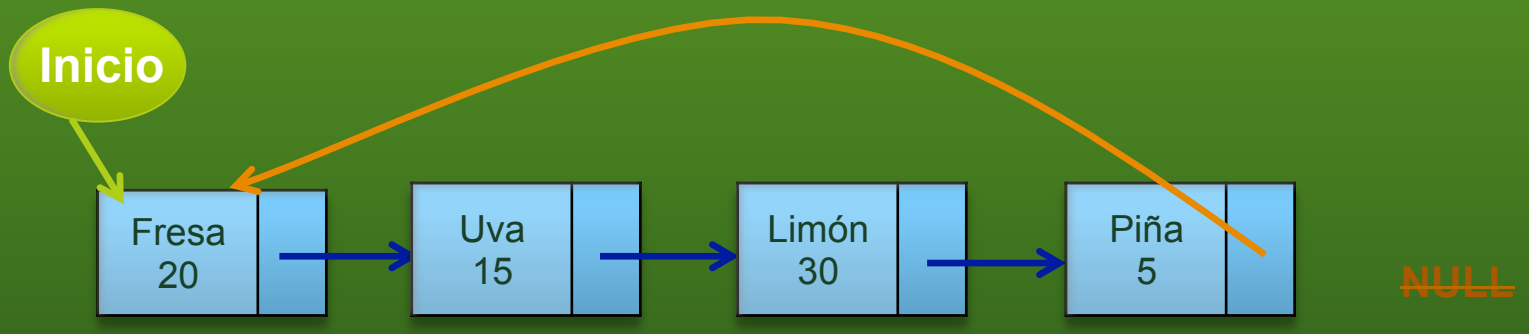
Lista doblemente enlazada

- Cada nodo de la estructura tiene un campo de enlace que apunta al siguiente nodo en la lista y un campo de enlace que apunta al nodo anterior de la lista.
- El ultimo y primer nodo en la lista apunta NULL.



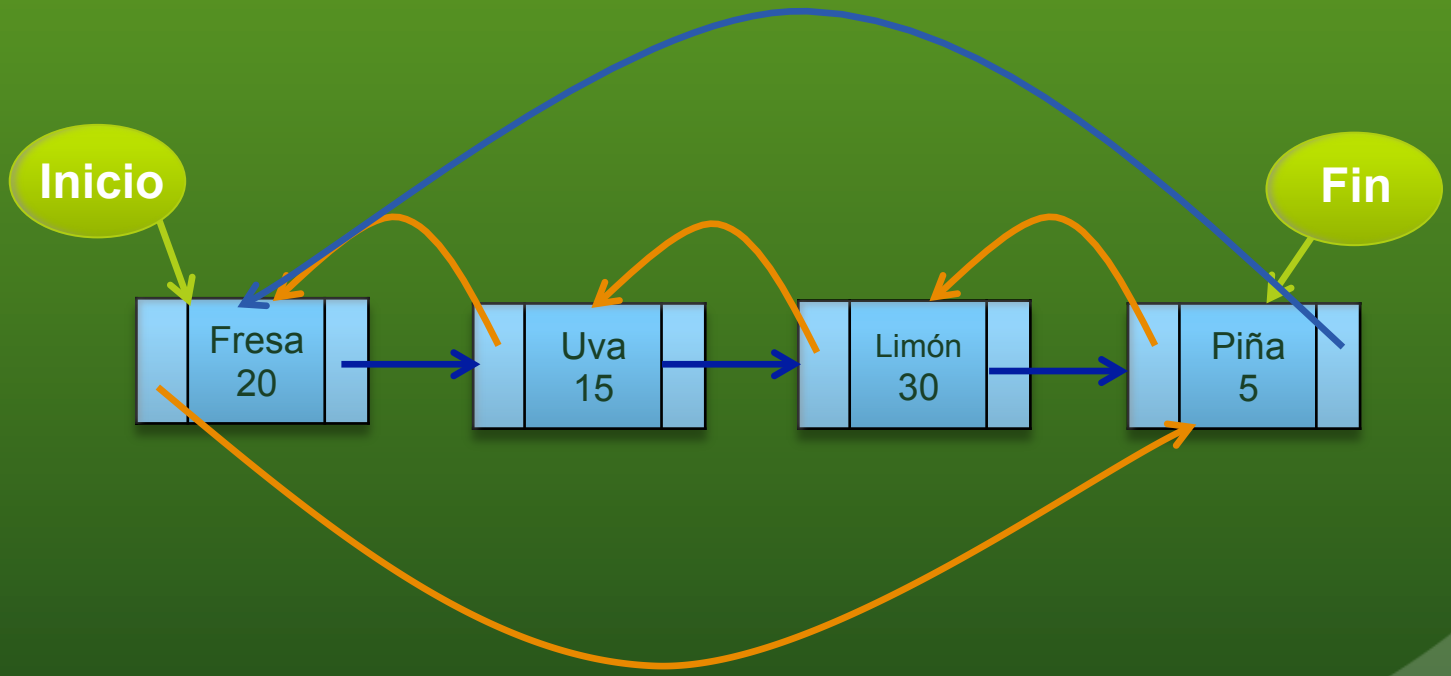
Lista circular simple

- Es una lista simple en la que el ultimo nodo apunta al primero en lugar de a NULL.



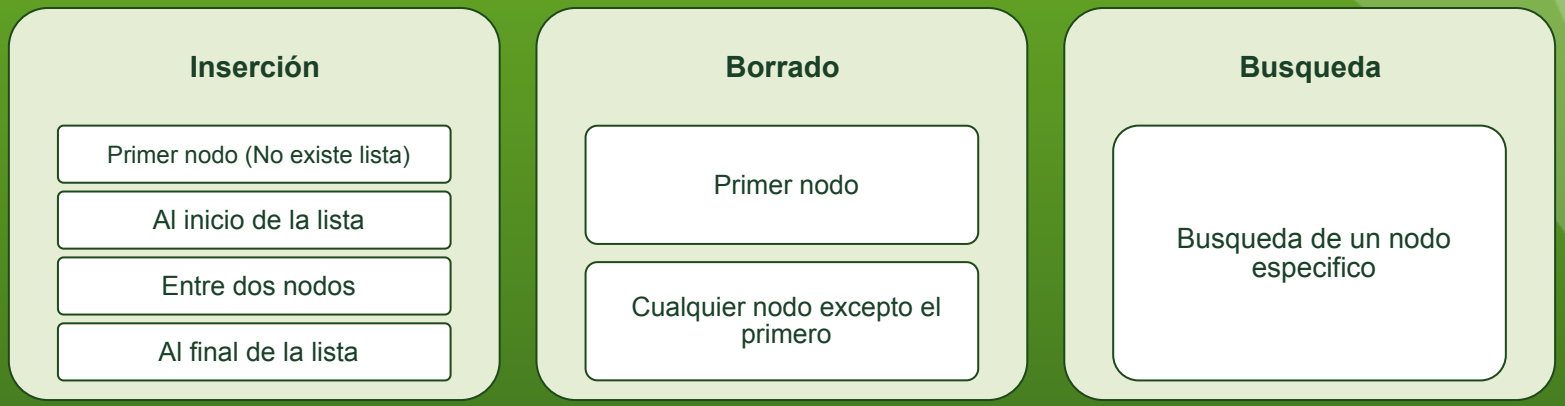
Lista circular doble

- Es una lista enlazada doble en la que el ultimo nodo apunta al primero y el primer nodo apunta al ultimo.



Operaciones

Las operaciones básicas en una lista son:

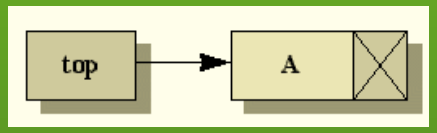


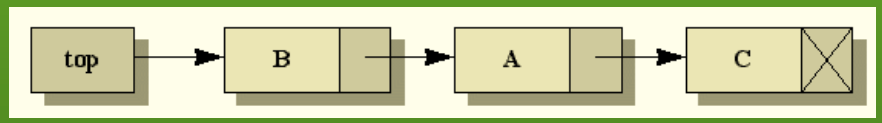
Las listas enlazadas permiten inserciones y eliminación de nodos en cualquier punto de la lista en tiempo constante (suponiendo que dicho punto está previamente identificado o localizado), pero no permiten un acceso aleatorio.

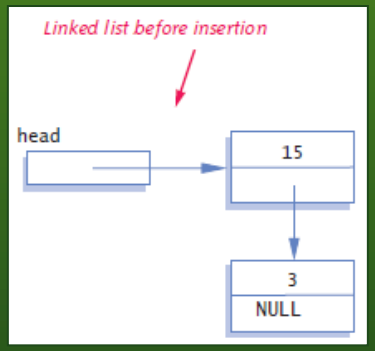
Operaciones

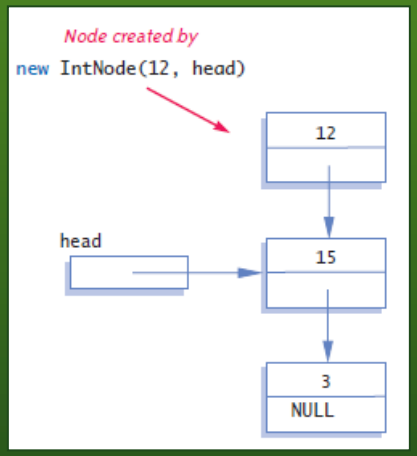
Insertar nodos:

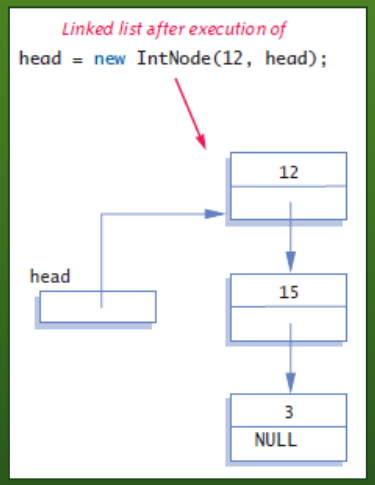
- **No existe lista**
- **Final de la lista**
- **Inicio de la lista**





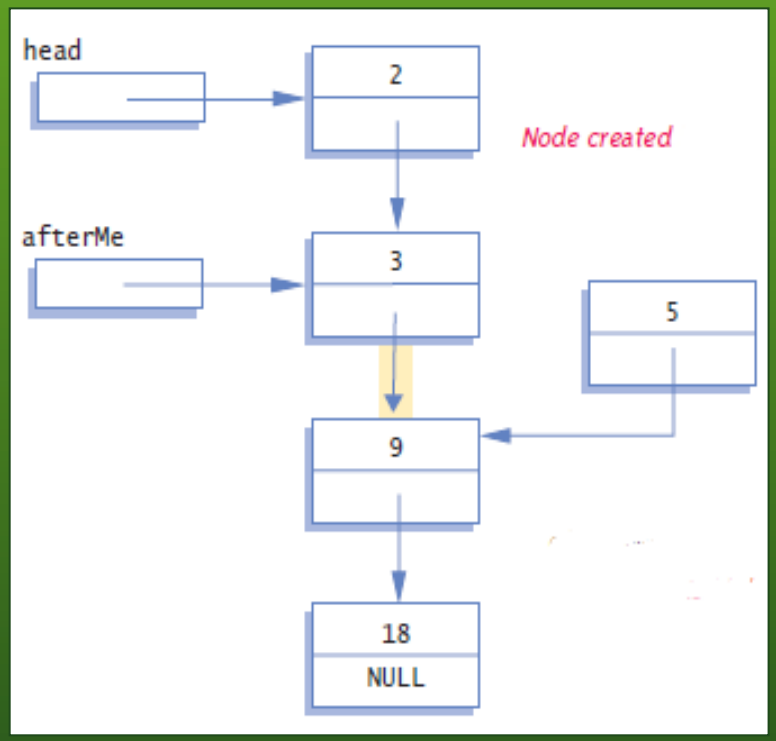


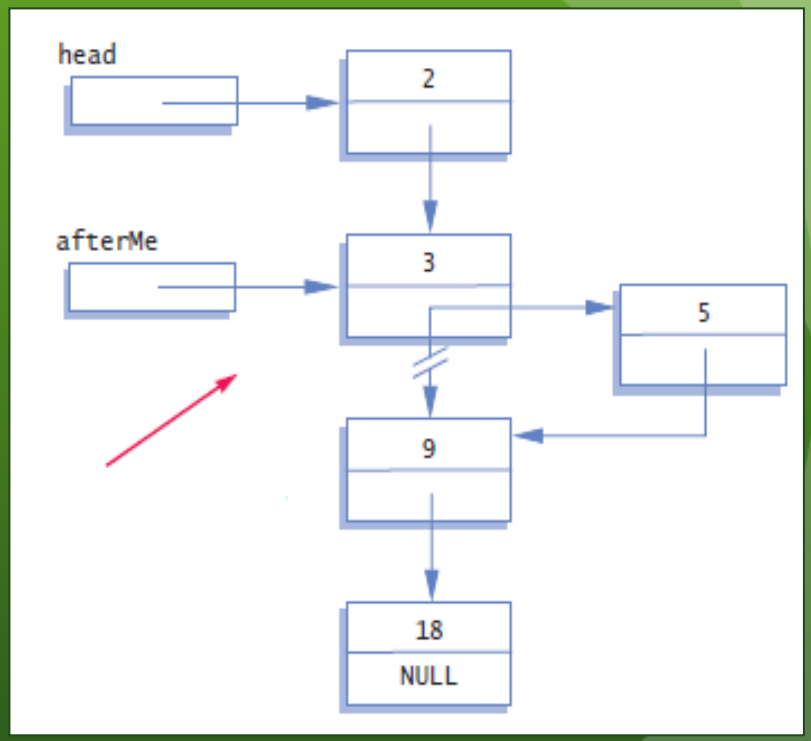




Operaciones

- Insertar entre dos nodos

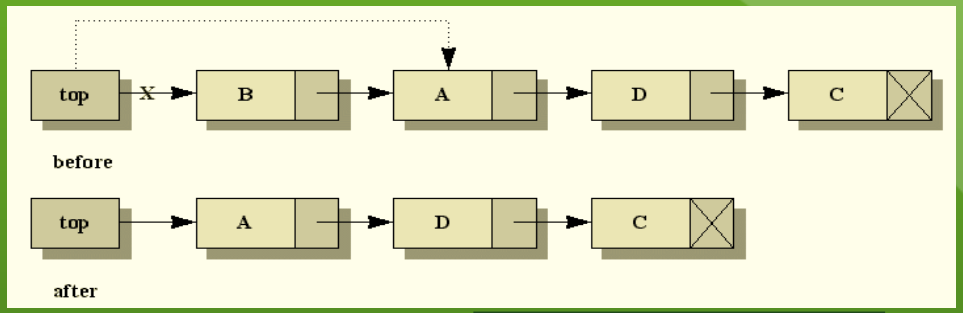




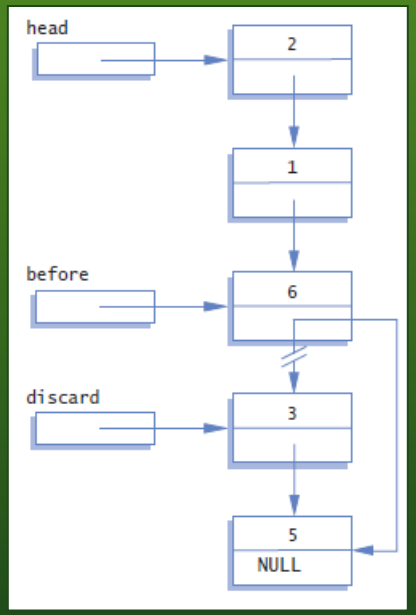
Operaciones

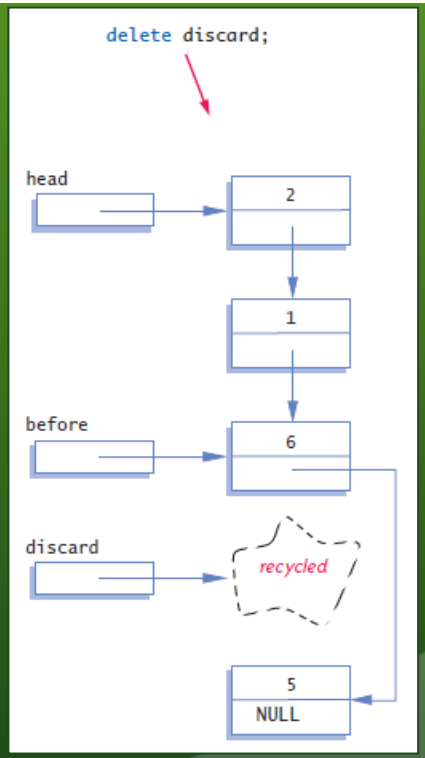
Borrar nodos:

- **Primer nodo**

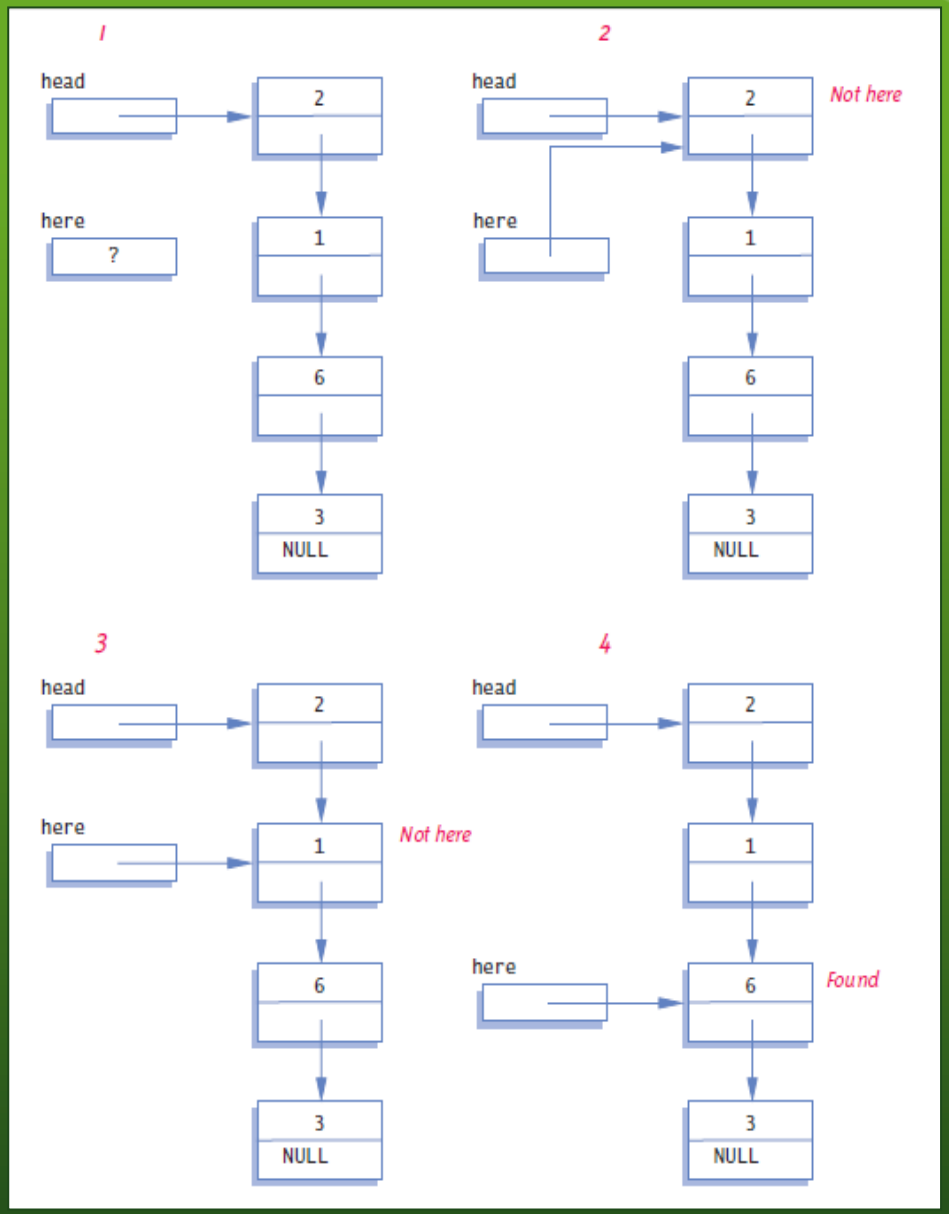


- **Cualquier nodo excepto primero**





Operaciones



- Buscar un nodo

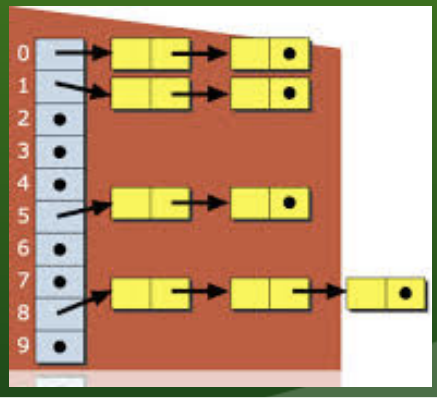
Listas y arreglos

- Las listas no requieren memoria entra para soportar la expansión, por tanto el rendimiento no se afecta.
- La inserción y borrado de elementos es más rápida que en los arreglos.
- Los elementos de los arreglos ocupan menos memoria que los nodos porque no requieren campos de enlace.
- Los arreglos ofrecen un acceso más rápido a los datos, mediante índices basados en enteros.
- Las listas enlazadas son más apropiadas cuando se trabaja con datos dinámicos. En otras palabras, inserciones y borrados con frecuencia. Por el contrario, los arreglos son más apropiados cuando los datos son estáticos (las inserciones y borrados son pocas).

Aplicaciones

Las estructuras tipo lista se usan:

- Para implementar otras estructuras.
- Organizar de datos.
- Visores de datos, imágenes, archivos, reproductores.
- Indexar u ordenar datos.
- Listar tareas, canciones, recordatorios, archivos, etc.



Referencias

- Koffman, Elliot y Wolfgang, Paul. (2008). Estructura de datos con C++. Objetos, abstracciones y diseño. McGraw-Hill.
- Cairó, Osvaldo y Guardati, Silvia. (2006). Estructuras de datos (3a. Edición). McGraw-Hill.
- Criado, Ma. Asunción. (2006). Programación en lenguajes estructurados. AlfaOmega Ra-Ma.
- Joyanes, Luis. (2008). Fundamentos de programación (4a Edición). McGraw-Hill.
- López, Leobardo. (2004). Programación estructurada. Un enfoque algorítmico (2a. Edición). AlfaOmega.
- Joyanes, Luis. M. Fernández, L: Sánchez, I. Zahonero. (2005). Estructuras de datos en C. McGraw-Hill. Schaum.
- Web:C Plus Plus: Biblioteca STL www.cplusplus.com/reference/stl .

GRACIAS

Continua Unidad de Competencia III:
Árboles

C. U. UAEM ZUMPANGO / ICO / ED / ECHL

Guía para el Profesor

- Las primeras diapositivas muestran el propósito, justificación y objetivos de la unidad de aprendizaje. Se presentan para que el alumno identifique dichos elementos.
- El contenido, conforme a la unidad de aprendizaje, maneja los temas de un menor a mayor grado de dificultad.
- Se recomienda que para cada tipo de estructura el alumno indique ejemplos relacionados a computación y de vida cotidiana donde se implemente la estructura.
- Realizar ejercicios en los que se pida dibujar la estructura y agregar o quitar elementos, mostrando cada paso en un dibujo para identificar que el alumno aprendió el funcionamiento de la estructura,
- Para el tema de pilas, pedir que resuelva el ejercicio de torres de Hanoi, con 5 discos y 3 torres. Se puede dibujar por pasos o usar monedas para visualizarlo.

Guía para el Profesor

- En ocasiones en conveniente iniciar con la estructura tipo lista, programarla y de ahí pasar a estructura tipo pila y cola, ya que para su implementación se pueden suprimir algunas operaciones ya programadas desde listas.

- El modelo de programación debe iniciar con la plantilla STL de C++ Standard Template Library para conocer el funcionamiento de las estructuras, posteriormente pedir que se programen usando arreglos y al final pedir que en base a estructuras (struct) el alumno defina sus propias pilas, colas y listas.

- Aunque puede ser eficiente un modelo orientado a objetos no es recomendable en este momento debido a que dicho paradigma se cubre en otra asignatura, y a menos que ya se haya cursado, implementar por medio de clases las estructuras implica que el alumno debe aprender el paradigma y el uso de las estructuras de datos, situación que podría complicarle su aprendizaje.