Laboratorio 12

Punteros, POO y Colas

1. Competencias

1.1. Competencias del curso

Conoce, comprende e implementa programas usando punteros y POO del lenguaje de programación C++.

1.2. Competencia del laboratorio

Conoce, comprende e implementa programas usando punteros y POO del lenguaje de programación C++.

2. Equipos y Materiales

- Un computador.
- IDE para C++.
- Compilador para C++.

3. Marco Teórico

3.1. Punteros con colas

Una cola es un tipo especial de lista enlazada en la que sólo se pueden insertar nodos en uno de los extremos de la lista y sólo se pueden eliminar nodos en el otro. Además, como sucede con las pilas, las escrituras de datos siempre son inserciones de nodos, y las lecturas siempre eliminan el nodo leído.

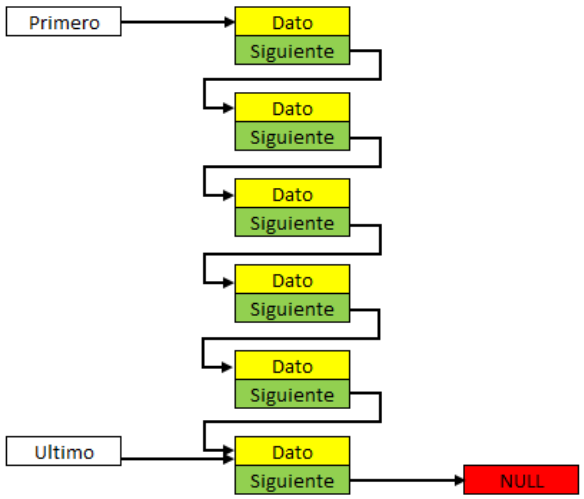
Este tipo de lista es conocido como lista FIFO (First In First Out), el primero en entrar es el primero en salir.

El ejemplo más común es una cola para comprar, por ejemplo, las entradas del cine. Los nuevos compradores sólo pueden colocarse al final de la cola, y sólo el primero de la cola puede comprar la entrada.

Los tipos que definiremos normalmente para manejar colas serán casi los mismos que para manejar listas y pilas

Es evidente, a la vista del gráfico, que una cola es una lista enlazada. Así que sigue siendo muy importante que nuestro programa nunca pierda el valor del puntero al primer elemento, igual que pasa con las listas enlazadas. Además, debido al funcionamiento de las colas, también deberemos mantener un puntero para el último elemento de la cola, que será el punto donde insertemos nuevos nodos.

Teniendo en cuenta que las lecturas y escrituras en una cola se hacen siempre en extremos distintos, lo más fácil será insertar nodos por el final, a continuación del nodo que no tiene nodo siguiente, y leerlos desde el principio, hay que recordar que leer un nodo implica eliminarlo de la cola.



3.2. Operaciones básicas con colas

De nuevo nos encontramos ante una estructura con muy pocas operaciones disponibles.

Las colas sólo permiten añadir y leer elementos:

- Añadir: Inserta un elemento al final de la cola.
- Leer: Lee y elimina un elemento del principio de la cola.

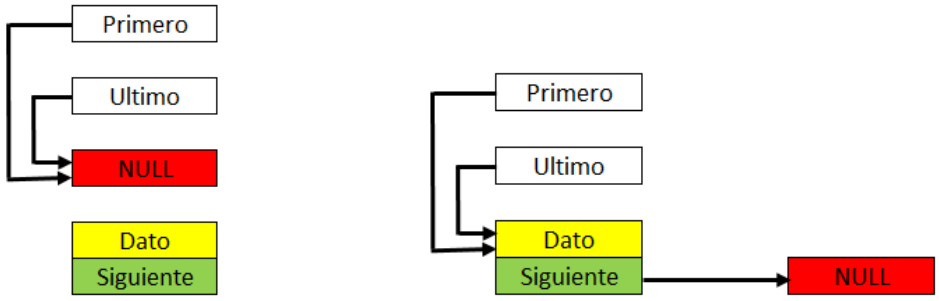
1. Añadir un elemento

a) Añadir un elemento en una cola vacía

Partiremos de que ya tenemos el nodo a insertar y, por supuesto un puntero que apunte a él, además los punteros que definen la cola, primero y ultimo que valdrán NULL:

El proceso es muy simple, bastará con que:

1. *nodo->siguiente apunte a NULL.*
2. *Y que los punteros primero y ultimo apunten a nodo.*

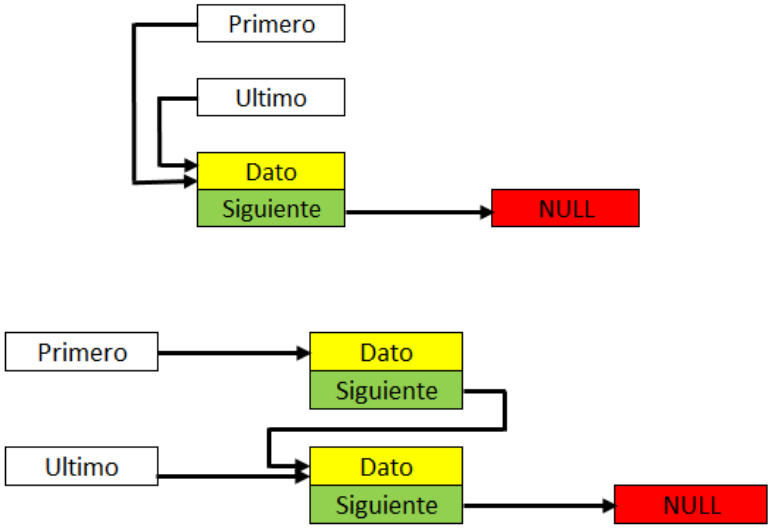


b) Añadir elementos en una cola no vacía

De nuevo partiremos de un nodo a insertar, con un puntero que apunte a él, y de una cola, en este caso, al no estar vacía, los punteros primero y último no serán nulos:

El proceso sigue siendo muy sencillo:

1. *Hacemos que nodo->siguiente apunte a NULL.*
2. *Después que Ultimo->siguiente apunte a nodo.*
3. *Y actualizamos Ultimo, haciendo que apunte a nodo.*

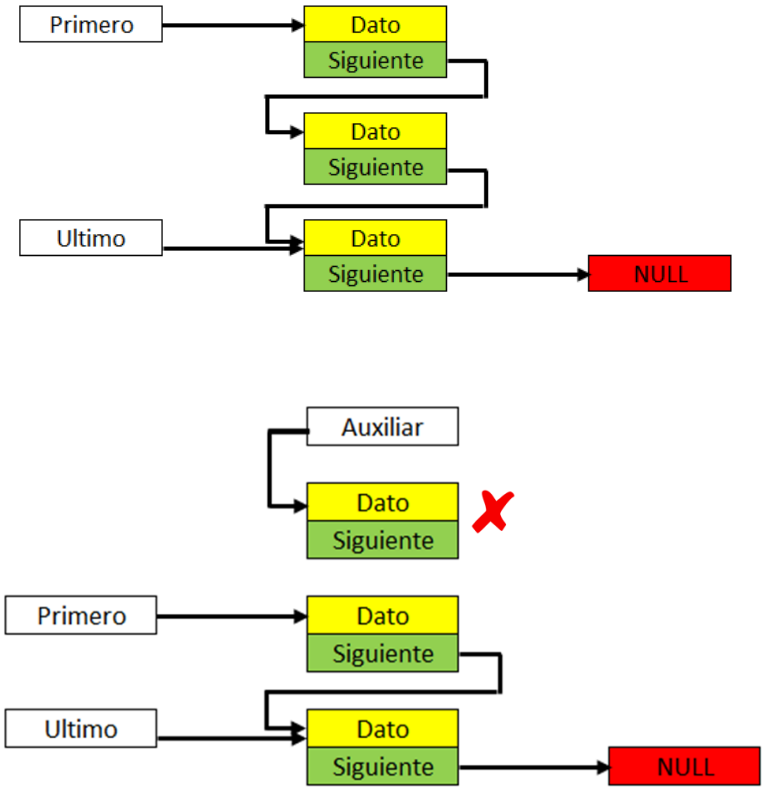


2. Leer un elemento de una cola

a) Leer un elemento en una cola con más de un elemento

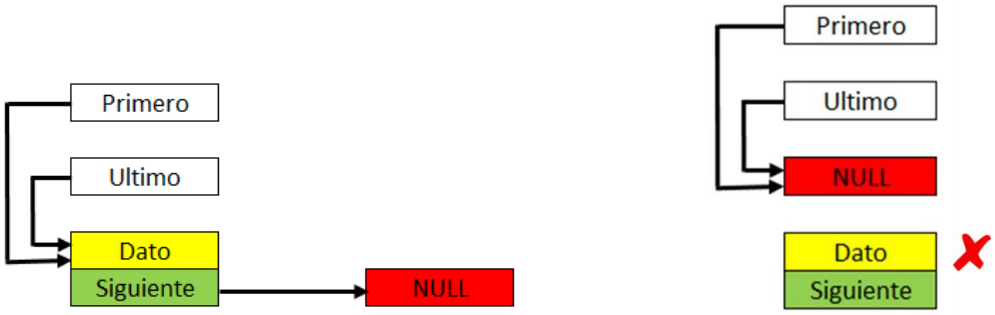
Usaremos un puntero a un nodo auxiliar:

1. Hacemos que Auxiliar apunte al primer elemento de la cola, es decir a primero.
2. Asignamos a primero la dirección del segundo nodo de la cola: primero- >siguiente.
3. Guardamos el contenido del Auxiliar para devolverlo como retorno, recuerda que la operación de lectura en colas implica también borrar.
4. Liberamos la memoria asignada al primer nodo, el que queremos eliminar.



También necesitamos un puntero a un nodo auxiliar:

1. Hacemos que Auxiliar apunte al primer elemento de la cola, es decir a primero.
2. Asignamos NULL a primero, que es la dirección del segundo nodo teórico de la cola: primero->siguiente.
3. Guardamos el contenido del Auxiliar para devolverlo como retorno, recuerda que la operación de lectura en colas implica también borrar.
4. Liberamos la memoria asignada al primer nodo, el que queremos eliminar.
5. Hacemos que ultimo apunte a NULL, ya que la lectura ha dejado la cola vacía.



3. Ejercicios

Resolver los siguientes ejercicios planteados:

1. Defina una Cola que permita insertar elementos utilizando clases.
2. Sobre el ejercicio anterior, adecue el programa para eliminar elementos de una Cola.
3. Implemente un algoritmo para buscar elementos de la Cola.
4. Escribir un programa que permita comparar las edades de diferentes elementos. Debe utilizar el formato de colas en clases. Se debe definir· inicialmente el número de elementos y valores de cada cola (pudiendo ser de diferentes tamaños ej. 2-3). Se evaluará el resultado de la comparación de los primeros elementos de las colas, realizada en un número de iteraciones 'n', en cada iteración se debe realizar el procedimiento de inserción y eliminación de elementos (rotando los elementos definidos inicialmente)

4. Entregables

Al final estudiante deberá:

1. Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
2. Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe de estar en formato PDF.
3. El nombre del archivo (comprimido como el documento PDF), será su LAB12_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.

   (Ejemplo: LAB12_GRUPO_A _2022123_PEDRO_VASQUEZ).
4. Debe remitir el documento ejecutable con el siguiente formato:

   LAB12_GRUPO_A/B/C_CUI_ EJECUTABLE_1erNOMBRE_1erAPELLIDO

   (Ejemplo: LAB12_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

En caso de encontrarse trabajos similares, los alumnos involucrados no tendrán evaluación y serán sujetos a sanción.