




```
int main()  
{  
    return 0;  
}
```



Colas

Programación III.

Objetivos

- 
- Conocer el modelo del tipo abstracto de datos cola
 - Elaborar ejercicios de aplicación de colas

Colas

- Una cola (fila) es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción push se realiza por un extremo y la operación de extracción pop por el otro. También se le llama estructura FIFO (First In First Out), debido a que el primer elemento en entrar también será el primero en salir.

Implementación de Colas

- ❑ Las colas pueden implementarse mediante arrays, mediante punteros o listas enlazadas.
- ❑ Una cola puede estar vacía o llena. Si un programa intenta sacar elementos de una lista vacía se producirá un “underflow”(desbordamiento negativo), si se intenta agregar un elemento a una lista llena(arrays) entonces se producirá un error llamado desbordamiento(overflow).

Operaciones básicas de las colas

- ❑ Push: Añadir un elemento al final de la cola.
- ❑ Pop: Leer y eliminar un elemento del principio de la cola.

Push en una cola vacía

- ❑ Partiremos de que ya tenemos el nodo a insertar y, por supuesto un puntero que apunte a él, además de los punteros, primero, ultimo que valdrán nulo.

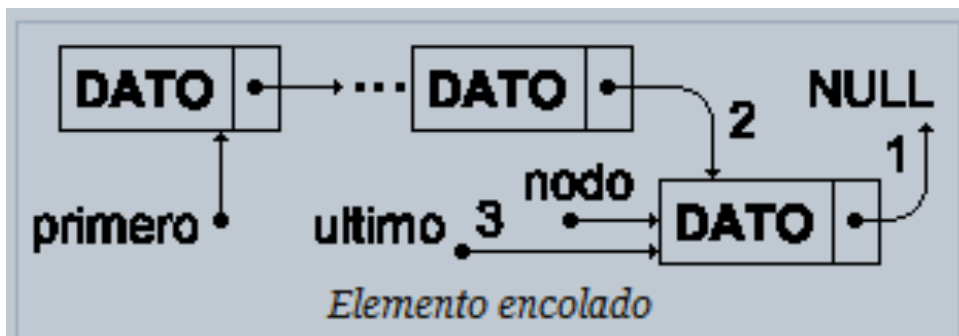
El proceso es muy simple, bastará con que:

- ❑ nodo->siguiente apunte a NULL.
- ❑ Que el puntero primero apunte a nodo
- ❑ Que el puntero ultimo tambien apunte a nodo



Push en una cola no vacía

- ❑ De nuevo partiremos de un nodo a insertar, con puntero que apunte a el, y de una cola, en este caso, al no estar vacía, los punteros primero y ultimo no serán nulos:
- ❑ Hacemos que nodo->siguiente apunte a NULL
- ❑ Después que último->siguiente apunte a nodo
- ❑ Y actualizamos último, haciendo que apunte a nodo



Push en una cola (Caso general)

- ❑ Hacemos que nodo->siguiente apunte a NULL
- ❑ Si ultimo no es NULL, hacemos que ultimo->siguiente apunte a nodo
- ❑ Y actualizamos ultimo, haciendo que apunte a nodo
- ❑ Si primero es NULL, significa que la cola estaba vacía, así que haremos que primero también apunte a nodo

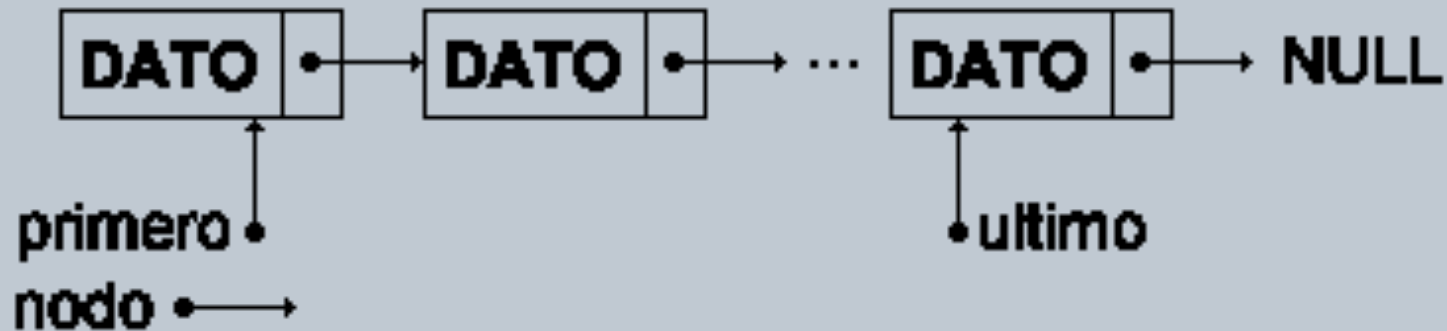
Pop, leer y eliminar un elemento

También existen dos casos, que la cola tenga un solo elemento o que tenga mas de uno

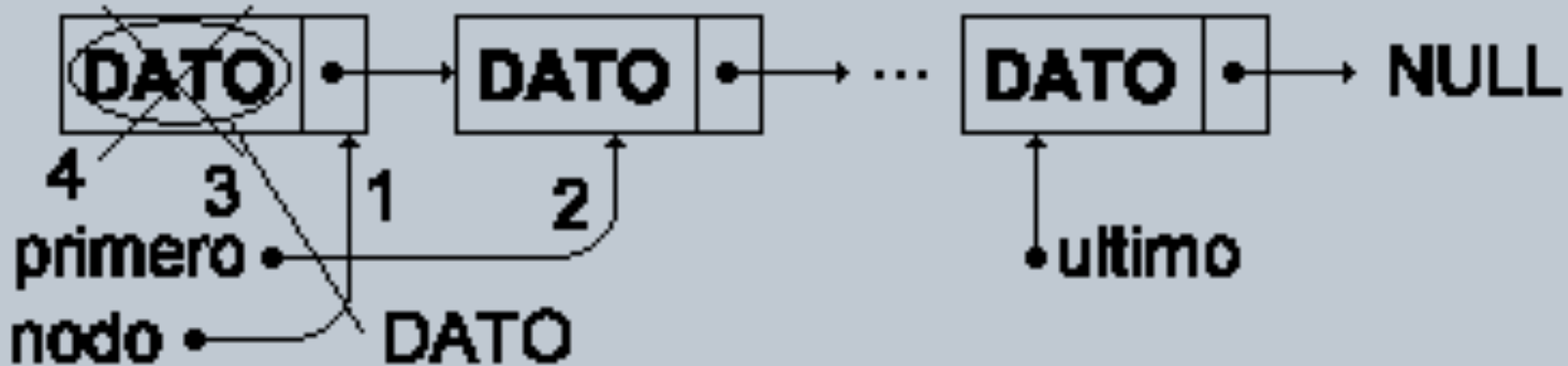
Primer caso

- ❑ Usaremos un puntero a un nodo auxiliar
- ❑ Hacemos que nodo apunte al primer elemento de la cola, es decir a primero
- ❑ Asignamos a primero la dirección del segundo nodo de la cola : primero->siguiente
- ❑ Guardamos el contenido de nodo para devolverlo como retorno, recordar que la operación de lectura en colas implica tambien borrar
- ❑ Se libera la memoria asignada al primer nodo, el que se quiere eliminar

Pop, leer y eliminar un elemento



Cola con más de un elemento



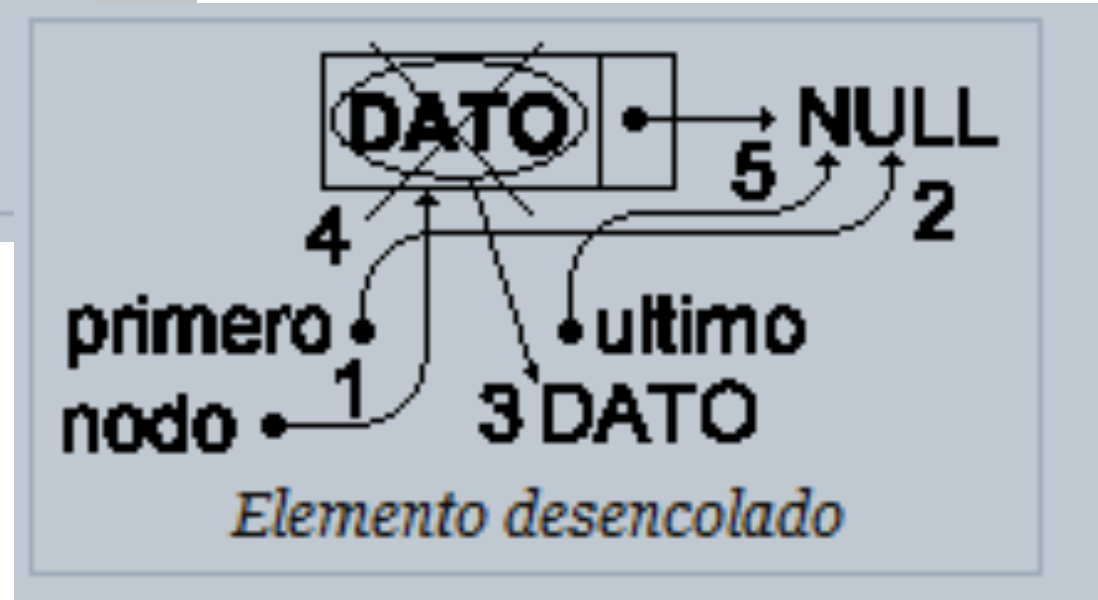
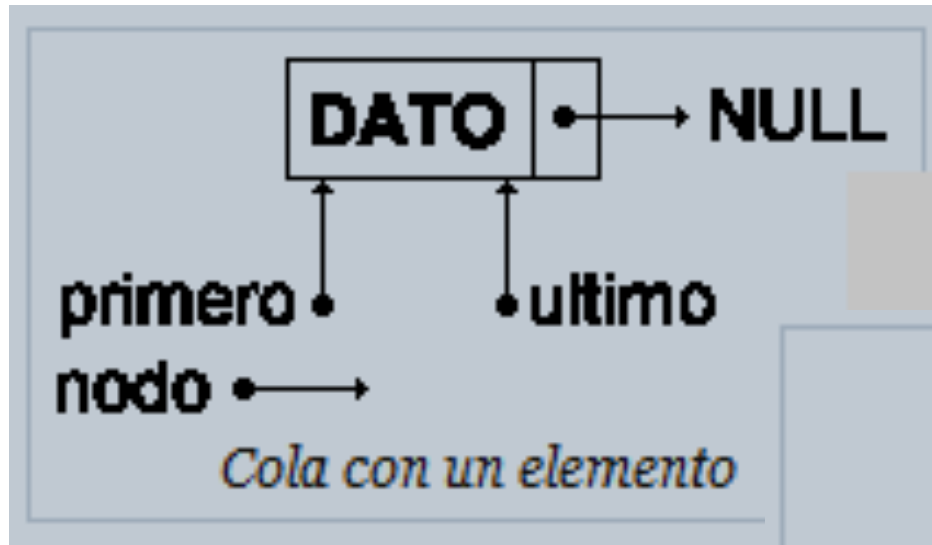
Elemento desencolado

Pop, leer y eliminar un elemento

Segundo caso

- ❑ También se necesita un puntero a un nodo auxiliar, y hacemos que nodo apunte al primer elemento de la cola.
- ❑ Asignamos NULL a primero que es la dirección del segundo nodo teórico de la cola: primero->siguiente
- ❑ Guardamos el contenido del nodo, para devolverlo como retorno.
- ❑ Se libera la memoria asignada al primer nodo
- ❑ Hacemos que ultimo apunte a NULL

Pop, leer y eliminar un elemento



Pop, leer y eliminar (Caso general)

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

Definición de la Cola

```
#ifndef NODO_H_INCLUDED
#define NODO_H_INCLUDED
class nodo{
private:
    int valor;
    nodo *siguiente;
public:
    nodo(int v, nodo *sig = NULL) {
        valor = v;
        siguiente = sig;
    }
    friend class cola;
};
typedef nodo *pnodo;
#endif // NODO H INCLUDED
```

Insertar un elemento en la cola

```
void Push(int v) {
    pnode nuevo;
    nuevo = new nodo(v);
    if (ultimo)
        ultimo->siguiente = nuevo;
    ultimo = nuevo;
    if (!primero)
        primero = nuevo;
}
```

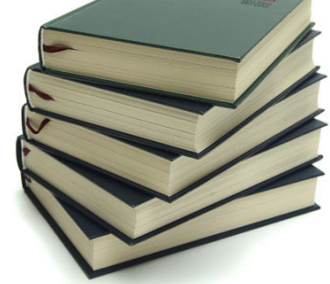
Operación pop

```
int Pop() {
    pnode nodo;
    int v;
    nodo = primero;
    if (!nodo)
        return 0;
    primero = nodo->siguiente;
    v = nodo->valor;
    delete nodo;
    if (!primero)
        ultimo = NULL;
    return v;
}
```




```
int main()  
{  
    return 0;  
}
```





Bibliografía

- <http://c.conclase.net/edd/?cap=002#inicio>
- [http://es.wikipedia.org/wiki/Cola_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Cola_(inform%C3%A1tica))