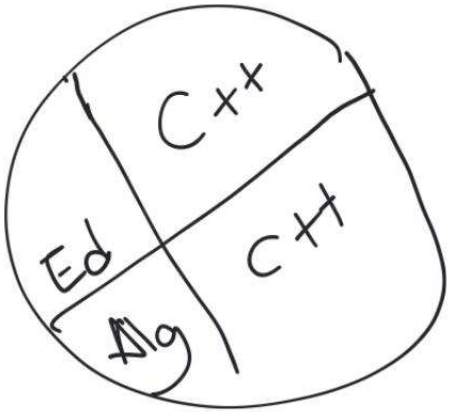
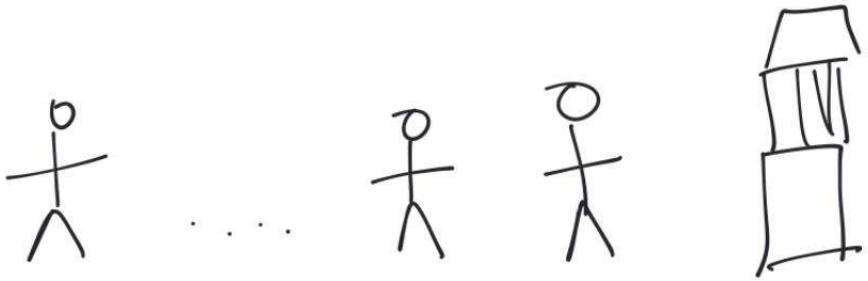
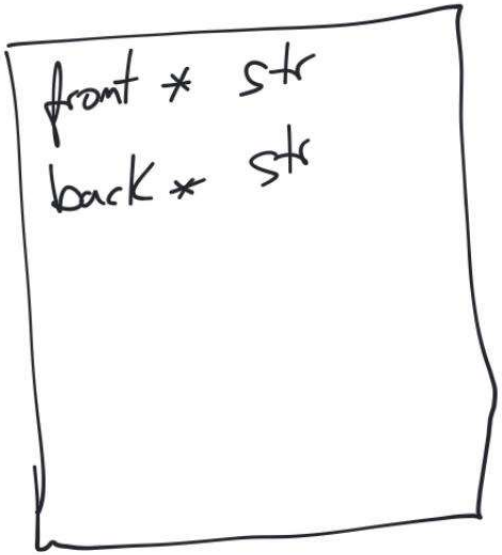
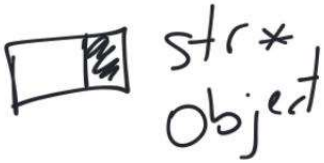
¿dudas?
Sobre la asignatura
Templates
Cola
revisitar pila
Ejercicios


C++
C++
Ed
Alg

Cola Fifo





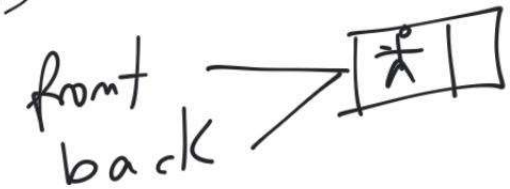


make empty()

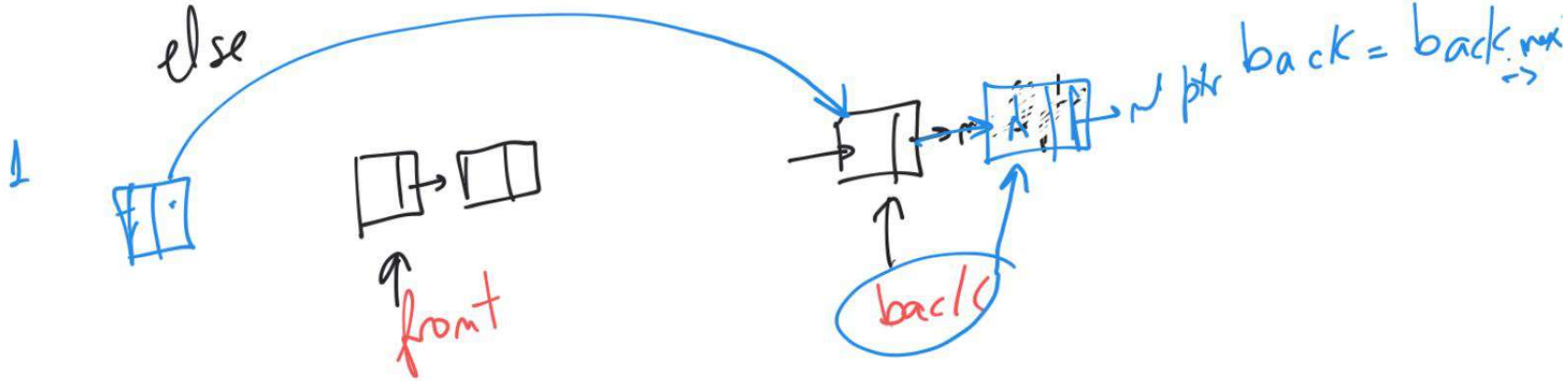
front = back = nullptr

llega Alguien enqueue

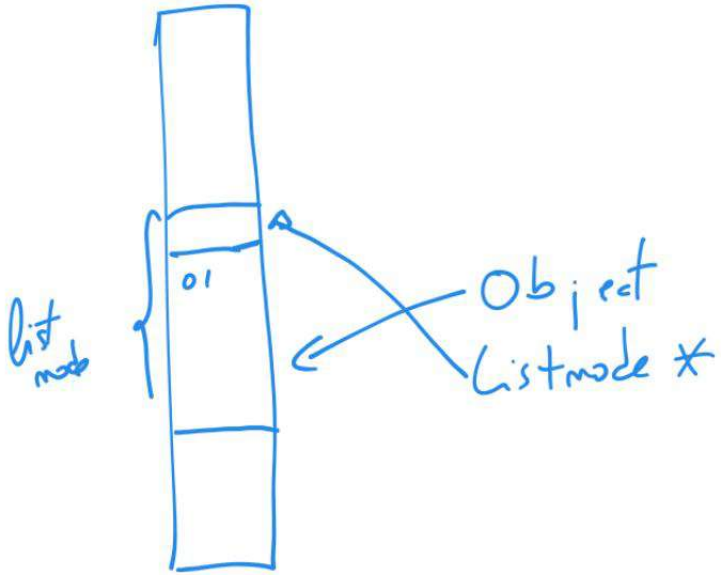
if (empty)



else



```
// Insert x into the queue.
template <class Object>
void Queue<Object>::enqueue( const Object & x )
{
    if( isEmpty( ) )
        back = front = new ListNode( x );
    else
        back = back->next = new ListNode( x );
}
```


list node
01
Object
Listnode *

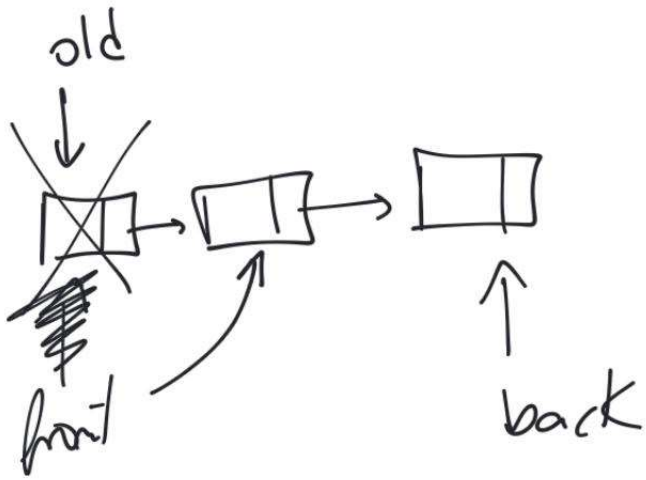
```
private:
  struct ListNode
  {
      Object    element;
      ListNode *next;

      ListNode( const Object & theElement, ListNode * n = NULL )
        : element( theElement ), next( n ) { }
  };
```

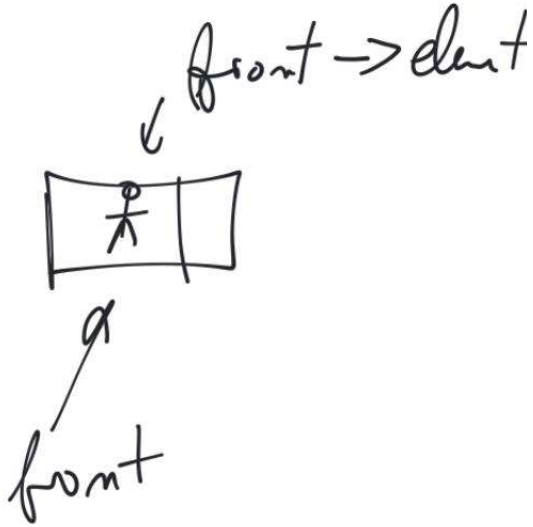
```
// Return and remove the least recently inserted item from
// the queue. Throw UnderflowException if empty.
template <class Object>
Object Queue<Object>::dequeue( )
{
    Object frontItem = getFront( );

    ListNode *old = front;
    front = front->next;
    delete old;

    return frontItem;
}
```



```
// Return the least recently inserted item in the queue
// or throw UnderflowException if empty.
template <class Object>
const Object & Queue<Object>::getFront( ) const
{
    if( isEmpty( ) )
        throw UnderflowException( );
    return front->element;
}
```



```
// Test if the queue is logically empty.
// Return true if empty, false, otherwise.
template <class Object>
bool Queue<Object>::isEmpty( ) const
{
    return front == NULL;
}
```

```
// Make the queue logically empty.
template <class Object>
void Queue<Object>::makeEmpty( )
{
    while( !isEmpty( ) )
        dequeue( );
}
```

stackInt

struct

int element

*

```cpp
template <class T>
class className {
private:
    T var;
    ... .. ...
public:
    T functionName(T arg);
    ... .. ...
};
```

```
className<int> classObject;
className<float> classObject;
className<string> classObject;
```

```cpp
// Class template
template <class T>
class Number {
private:
    // Variable of type T
    T num;

public:
    Number(T n) : num(n) {}   // constructor

    T getNum() {
        return num;
    }
};
```

```
int main() {

    // create object with int type
    Number<int> numberInt(7);

    // create object with double type
    Number<double> numberDouble(7.7);

    cout << "int Number = " << numberInt.getNum() << endl;
    cout << "double Number = " << numberDouble.getNum() << endl;

    return 0;
}
```

infija $2 + 3$

post fija $2 \; 3 \; +$

pre fija $sim(\pi)$

1ª $\boxed{2}$

2ª $\boxed{3}$ $\boxed{2}$

3ª $+$

$3 \to a$

$2 \to b$

$2 \quad 3 \to$ desapilado

$+$

$\boxed{5} \leftarrow$ apilar(5)

$2 + 3 * (7 + 2)$

$2 \quad 7 \quad + \quad 3 \quad * \quad 2 \quad +$

$a = 2$
$b = 27$

29

$a \leftarrow 7$
$b \leftarrow 2$

$+ \quad 9$

$a\ 3$
$b\ 9$

$\times$

27

$\cancel{7} \quad \cancel{3} \quad \cancel{2}$
$\cancel{2} \quad \cancel{9} \quad \cancel{27}$

$1\ 2\ 3\ 4\ 7\ 25\ +\ -\ *$

Crear una calculadora de notación postfija

Gin ?

1er Gin ≠ .

2º Gin ¿0-9? {+- + *} .

3er 0-9 → apila

4º + - * /

desapila → a

desapila → b

a+b → apila

5º done!

1 2 3 7 4 25 7 + 2 − * /

2 +

25+7

32

4

4

7

3

2

1