producer

```
WHILE (an empty slot) is true
        synchronized (queue) {
                While (queue size ==max size) is true
                        Wait until consumer consume an item
                End While
                Add next produced to the buffer
                queue. Notify
                }
ENDWHILE
```

1. first producer cheek if there is at least one empty slot.

2. Then it decrements the empty queue because, there will now be one less empty slot, since the producer is going to insert data in one of those slots.

3. then synchronized the buffer, so that the consumer cannot access the buffer until producer completes its operation

4. After performing the insert operation, the value of full is incremented because the producer has just filled a slot in the buffer.

5.finally waking up threads that are waiting for access this object

Consumer

```
WHILE (one full slot) is true
        synchronized (queue) {
                While (queue size ==0) is true
                        Wait until producer produce an item
                End While
                remove an item from buffer
                queue. Notify
                }
ENDWHILE
```

1. first consumer cheek if there is at least one full slot in the buffer.

2. decrements the full queue because the number of occupied slots will be decreased by one, after the consumer completes its operation

3. the consumer synchronized the buffer

4. the consumer completes the removal operation so that the data from one of the full slots is removed.

5. the empty semaphore is incremented by 1, because the consumer has just removed data from an occupied slot, thus making it empty.

6.finally waking up threads that are waiting for access this object