

Salidas:

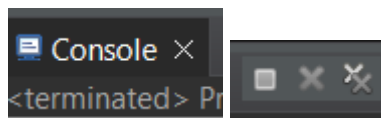
```
Insertado 10 | tamaño: 1
Productor 1, produce 10
Extraído 10 | tamaño: 0
Consumidor 0, consume 10
Insertado 14 | tamaño: 1
Productor 5, produce 14
Insertado 13 | tamaño: 2
Productor 4, produce 13
Insertado 12 | tamaño: 3
Productor 3, produce 12
Extraído 14 | tamaño: 2
Consumidor 0, consume 14
Extraído 13 | tamaño: 1
Consumidor 0, consume 13
Extraído 12 | tamaño: 0
Consumidor 0, consume 12
Insertado 11 | tamaño: 1
Productor 2, produce 11
Extraído 11 | tamaño: 0
Consumidor 0, consume 11
FIN del programa
```

```
Insertado 10 | tamaño: 1
Extraído 10 | tamaño: 0
Productor 1, produce 10
Insertado 14 | tamaño: 1
Productor 5, produce 14
Consumidor 0, consume 10
Insertado 13 | tamaño: 2
Productor 4, produce 13
Insertado 12 | tamaño: 3
Productor 3, produce 12
Extraído 14 | tamaño: 2
Consumidor 0, consume 14
Extraído 13 | tamaño: 1
Consumidor 0, consume 13
Extraído 12 | tamaño: 0
Consumidor 0, consume 12
Insertado 11 | tamaño: 1
Productor 2, produce 11
Extraído 11 | tamaño: 0
Consumidor 0, consume 11
FIN del programa
```

-Da resultados parecidos pero nunca consume sin haber producido antes, porque cuando funciona un hilo, bloquea al otro con `wait()`, y el productor desbloquea al consumidor con `notify()` cada vez que ha producido.

-No los inserta en orden, pero si los extrae en orden.

Además, consume el primero que se ha introducido, y respeta el método FIFO.



Finaliza correctamente porque termina los bucles del Productor y del Consumidor.