

## semafori-prodcons-wrong

Perchè questa versione (suggerita da un vostro collega) del Produttore/Consumatore non è accettabile?

```
shared Object buffer = null;
Semaphore mutex = new Semaphore(1);
process Producer {
    while(true) {
        Object val = produce();
        for(;;) {
            mutex.P();
            if (buffer==null) break;
            mutex.V();
        }
        buffer=val;
        mutex.V()
    }
}
process Consumer{
    while(true) {
        Object val;
        for(;;) {
            mutex.P();
            if (buffer!=null) break;
            mutex.V();
        }
        val = buffer;
        buffer=null;
        mutex.V();
        consume(val);
    }
}
```

**Soluzione:**

Perchè in realtà utilizza busy waiting, che è proprio quello che volevamo evitare quando abbiamo introdotto il concetto di semaforo. La tecnica di busy waiting è contenuta nei cicli "for (;)" (cicli infiniti da cui si esce tramite break, non proprio una bella pratica): si continua ad entrare ed uscire dalla mutua esclusione testando il valore della variabile.