

PRODUCTOR-CONSUMIDOR

SOLUCIÓN CON SEMÁFOROS

- Barajas Cervantes Alfonso
- Cabello Figueroa Israel
- Cerritos Lira Carlos
- Franco López Benito Vicente

Instituto de Investigaciones
en Matemáticas Aplicadas y en Sistemas

1	La Situación	
2	Suposiciones	
3	El Problema	
4	Una Posible Solución	
5	Pongamonos Técnicos	
6	El Problema	
7	La Solución	
8	Ventajas y Desventajas	
9	Pseudo-código	

Existe una **empresa productora** de naranjas, las cuáles deben de ser entregadas en la central de abastos a un **comprador**. Para esto la empresa debe enviar su producción de naranjas en un **camión** de tamaño **finito**.



Se tienen las siguientes suposiciones:

- La empresa puede **producir** naranjas de forma **indefinida**, sin embargo, lo hace en intervalos irregulares (puede que un día haya mucha producción y otro día muy poca).
- El cliente en la central de abastos **siempre compra** las naranjas (si es que hay).
- El camión **viaja** de forma **instantánea** entre la empresa y la central.

Imaginemos los siguientes escenarios:

- Se producen **más** naranjas de las que caben en el camión, la producción restante se pudre.
- Se producen **menos** naranjas de las que son requeridas, entonces se pierde dinero.

Notemos que la región que debemos de cuidar es el camión, ya que es el recurso compartido entre la empresa y el consumidor.

Lo que podemos hacer es lo siguiente:

- Una vez **lleno** el camión, **parar** la producción de naranjas.
- Una vez el camión está **vacio**, enviar un **mensaje** de disculpa al consumidor, asegurándole que le informaremos de inmediato cuando haya más naranjas.

La situación antes descrita podemos identificar los siguientes elementos:

- **Productor:** produce información (empresa productora de naranjas).
- **Consumidor:** consume la información producida (comprador en central de abastos).
- **Canal de comunicación (buffer):** como su nombre lo dice, es la forma en la que se comunican los procesos, el productor introduce información al canal, mientras que el consumidor la tomá (camión donde se envían las naranjas).

Considerando que el buffer es de tamaño finito, tenemos los siguientes problemas:

- El buffer **se llena** y el productor intenta introducir información a esté.
- El buffer **está vacío** y el consumidor intenta tomar información.

Semáforo

Un semáforo es que da **acceso** o **bloquea** la entrada para un proceso, dependiendo del **estado de alguna variable** a la que apunte el semáforo, en este caso, nuestros procesos son el productor de naranjas y el consumidor, la variable que determina el acceso o bloqueo, es el **estado del camión**.

Como primer mecanismo de sincronización usaremos un semáforo o una señal que permita el correcto control del proceso compra-venta de naranjas.

Como tenemos que cuidar también las dos secciones críticas, otros dos semáforos, que en este ejemplo podrían ser dos vigilantes también



Ventajas

- La solución es **intuitiva**.
- Si tenemos N productores y M consumidores, la solución por semáforos se **generaliza** de forma sencilla.

Desventajas

- En el caso general, si cada productor solo puede introducir **tamaños discretos de información** al buffer, nuestra solución puede ocasionar un **deadlock** (cuando ninguno puede introducir información debido a que no hay suficiente espacio).

```
semaforo cajas_en_camion = 0;
semaforo espacios_vacios = TAMAÑO-CAMION;
empresa_naranjas()
{
    while (true)
    {
        caja_naranjas = producir_caja();
        down(espacios_vacios);
        #Si espacios_vacios = 0 va esperar hasta que
        #espacios_vacios > 0
        #Sección crítica
        subir_caja(caja_naranjas);
        #-----#
        up(cajas_en_camion);
    }
}

comprador_central()
{
    while (true)
    {
        down(cajas_en_camion);
        #Si cajas_en_camion = 0 va esperar hasta que
        #cajas_en_camion > 0
        #Sección crítica
        caja_naranjas = bajar_caja();
        #-----#
        up(espacios_vacios);
        vender_naranjas();
    }
}
```

```
1 import threading
2 N = 100
3 camion = []
4 en_camion = threading.Semaphore(0)
5 espacios_vacios = threading.Semaphore(N)
6
7 def producir_caja():
8     print("Una caja de naranjas producida!")
9     return 1
10
11 def empresa_naranjas():
12     while True:
13         caja_naranjas = producir_caja()
14         espacios_vacios.acquire()
15         camion.append(caja_naranjas)
16         en_camion.release()
17
18 def vender_naranjas(caja_naranjas):
19     print("Una caja de naranjas vendida!")
20
21 def comprador_central():
22     while True:
23         en_camion.acquire()
24         caja_naranjas = camion.pop(0)
25         espacios_vacios.release()
26         vender_naranjas(caja_naranjas)
27
28 empresa_naranjas_thread = threading.Thread(target=empresa_naranjas)
29 comprador_central_thread = threading.Thread(target=comprador_central)
30
31 empresa_naranjas_thread.start()
32 comprador_central_thread.start()
```