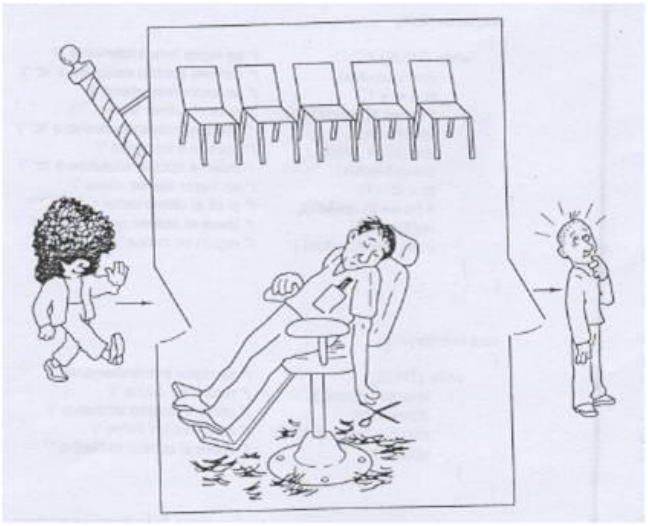
LABORATORIO 11:
SEMAFOROS: PROBLEMA DEL BARBERO DORMILÓN

Problema planteado por Dijkstra, es un problema clásico de comunicación y sincronización entre procesos, que enuncia lo siguiente:

Una barbería tiene una sala de espera con n sillas y la habitación del barbero tienen un único sillón. Si no hay clientes, entonces el barbero se duerme, si un cliente entra en la barbería y todas las sillas están ocupadas entonces el cliente se marcha, si el barbero está ocupado, pero hay sillas disponibles, entonces el cliente se sienta en una de ellas. Si el barbero está durmiendo, entonces el cliente lo despierta.



- **Llega un primer cliente** y entra en la barbería con **sillas desocupadas**, el primer cliente encuentra al barbero dormido y trata de despertarlo, si sigue dormido y hay sillas disponibles, **el cliente se sienta** a esperar su turno en una silla (el cliente también esperaría si el barbero atiende a otro cliente).
- El barbero despierta y atiende al primer cliente.
- **Llega un segundo cliente** y entra en la barbería con **sillas aun desocupadas**, el segundo cliente observa que el barbero está atendiendo al primer cliente, por lo tanto, el segundo **cliente se sienta** en una de las sillas a esperar.
- Si llegan más clientes mientras el barbero corta el cabello de un cliente, se van a sentar en las **sillas desocupadas**.
- **Llega un nuevo cliente** a la barbería y observa que todas las sillas están ocupadas entonces el nuevo cliente se marcha de la barbería.
- El nuevo cliente regresaría después cuando ya haya terminado el barbero de atender un cliente, o cuando haya al menos una **silla desocupada.**

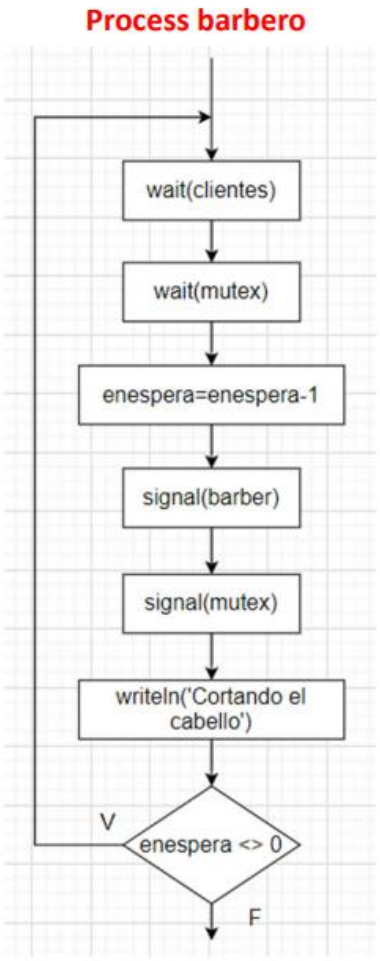
Programando problema de barbero dormilón

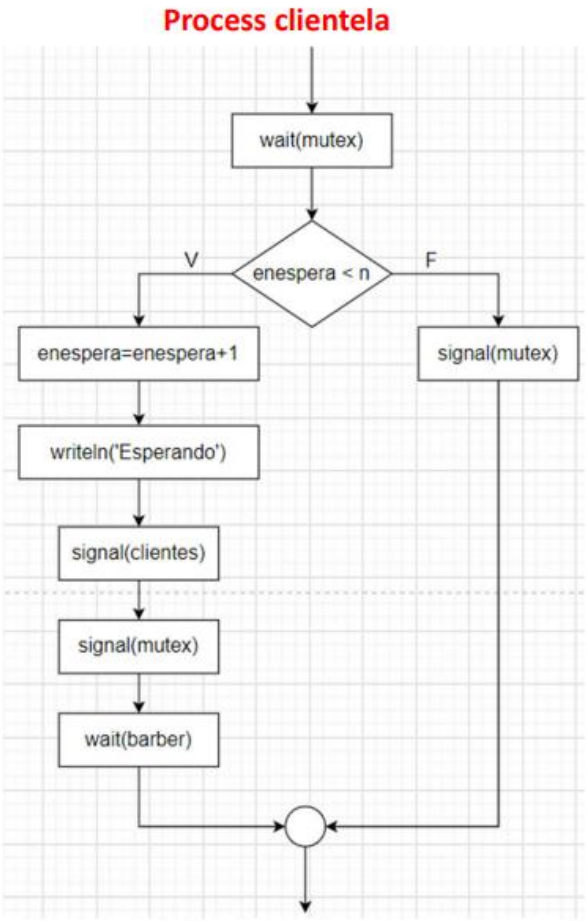
Condiciones:

- Una sala de espera con N sillas
- Una sala del barbero con una silla para cortar el cabello al cliente

Variables:

- Mutex: semáforo, bloquea a un cliente si la silla del barbero está ocupada, inicializado a 1.
- Clientes: semáforo, bloquea a los clientes que llegan y no encuentran sillas libres, inicializado a 0.
- Barber: semáforo, bloquea al barbero cuando no hay clientes, inicializado a 0. •
- N=5 sillas en la sala de espera
- Enespera=0






```
program barbero;
var mutex,clientes,barber :semaphore;
var n,enespera : integer;
process type barbero;
begin
      repeat
      while enespera <> 0   do
      begin
            wait(clientes);
            wait(mutex);
            enespera:=enespera-1;
            signal(barber);
            signal(mutex);
            writeln(' Barbero cortando el cabello ');
            end
      forever
end;
process type clientela;
begin
      repeat
            wait(mutex);
            if (enespera < n) then
            begin
                  enespera:=enespera+1;

                  write('Esperando');
```

```
                signal(clientes);
                signal(mutex);
                wait(barber);
            end
            else
            begin

                signal(mutex);
            end;

      forever
end;

VAR   BR: array[0..1] OF barbero;
var   C: array[0..4] of clientela;
begin

      initial(mutex,1);
      initial(clientes,0);
      initial(barber,0);
      enespera:=0;
      n:=5;
      cobegin
            C[0];
            C[1];
            C[2];
            C[3];
            C[4];
            BR[1];
      coend
end.
```

- Pruebe la solución del algoritmo de solución del Barbero Dormilon de Pascal FC a C++ con el IDE Codeblocks u otro de preferencia

```
#include <iostream>
#include <thread>
#include <semaphore.h>

using namespace std;

const int NUM_SILLAS = 5;

sem_t sem_sillas;
sem_t sem_barbero;
sem_t sem_cliente;
sem_t sem_corteTerminado;
```