

## Problema 2: Barbero dormilón

Procesos donde se realizan múltiples procesos:

En el caso del problema del barbero los procesos paralelos que se realizan vienen siendo aquellos donde al llegar un nuevo cliente se crea inicia el proceso de iniciar corte y en dado caso el barbero ya esté realizando el corte, el cliente procede a esperar hasta que el barbero quede libre

```
public class Cliente extends Thread{
    private Barbero barbero;
    private int turn;

    public Cliente(int clienumber, Barbero barber) {
        this.turn = clienumber;
        this.barbero = barber;
    }

    @Override
    public void run() {
        try {
            barbero.inicioCorte(turn);
            sleep(1000);
            barbero.finCorte(turn);
        } catch (InterruptedException ex) {
            ex.printStackTrace();
        }
    }
}

public synchronized void inicioCorte(int numClie) {
    while (isOcupado()) {
        System.out.println("Barbero atendiendo al cliente: " + numClie + ", no puede atender a alguien más. Espere");
        try {
            // con la finalidad de que se pueda representar la concurrencia el hilo bloquea el proceso de las demás hasta que se habilite por
            wait();
        } catch (InterruptedException ex) {
            Logger.getLogger(Barbero.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    this.setOcupado(true);
    System.out.println("El barbero atiende al cliente: " + numClie);
}

public synchronized void finCorte(int numClie) {
    this.setOcupado(false);
    System.out.println("El barbero ha terminado de atender al cliente: " + numClie);
    //libera cualquier posible bloqueo que se realice por medio de otros hilos
    notify();
}
```

El único caso que se presento dónde podría existir algún tipo de bloqueo seria en el caso de que al realizar el corte se quede en espera dejando bloqueado el proceso de finalización de corte, lo cual haría que se impida liberar este hilo para así poder proseguir con los demás cortes.

```
public boolean ocupado;  
public boolean[] sillas = new boolean[50];
```

Las variables que se compartirían dentro de los procesos, serían las variable de estado del barbero, junto con la de las sillas ocupadas.