

Portafolio fumadores

los objetos de condicion utilizados son:

sem estanquero: se utiliza para indicar cuando es estanquero se puede producir un nuevo ingrediente, cuando un fumador coge el ingrediente producido, se desbloquea al estanquero para que genere un nuevo ingrediente.

sem fumadores[]: es utilizada para indiicar a los fumadores cuando se puede fumar, cuando el estanquero genera un ingrediente, se desbloque el fumadore correspondiente para hacer la accion de fumar.

Codigo fuente:

```
import java.util.Random;
import monitor.*;

class Estanco extends AbstractMonitor
{
    private int ingre;
    private Condition sem_estanquero=makeCondition();
    private Condition[] sem_fumadores=new Condition[3];

    public Estanco(){
        this.ingre=-1;
        for(int i=0;i<3;i++){
            sem_fumadores[i]=makeCondition();
        }
    }

    public void obtenerIngrediente(int milIngrediente){
        enter();
        if(milIngrediente!=ingre){
            sem_fumadores[milIngrediente].await();
        }
        System.out.println("El fumador " + milIngrediente + " esta fumando con
ingrediente " + milIngrediente);
        ingre=-1;
        sem_estanquero.signal();
        leave();
    }

    public void ponerIngrediente(int ingrediente){
        enter();
        ingre=ingrediente;
        System.out.println("El estanquero ha generado el ingrediente " + ingre);
        sem_fumadores[ingre].signal();
        leave();
    }
}
```

```

    }

    public void esperarRecogidaIngrediente(){
        enter();
        if(ingre!=-1){
            sem_estanquero.await();
        }
        leave();
    }
}

class Fumador implements Runnable
{
    private Estanco estanco;
    private int miIngrediente;
    public Thread thr;

    public Fumador(int p_miIngrediente, Estanco p_miEstanco){
        miIngrediente=p_miIngrediente;
        estanco=p_miEstanco;
        thr=new Thread(this,"fumador "+ miIngrediente);
    }

    public void run(){
        while(true){
            estanco.obtenerIngrediente(miIngrediente);
            aux.dormir_max(2000);
        }
    }
}

```

```

class Estanquero implements Runnable
{
    public Thread thr;
    private Estanco estanco;

    public Estanquero(Estanco mEstanquero){
        estanco=mEstanquero;
        thr=new Thread(this,"estanquero");
    }

    public void run(){
        int ingrediente;
        while(true){
            ingrediente=(int) (Math.random()*3.0);
            estanco.ponerIngrediente(ingrediente);
            estanco.esperarRecogidaIngrediente();
        }
    }
}

```

```

class Fumadores

```

```

{
    public static void main(String[] args){
        Estanco estanco=new Estanco();
        Estanquero estanquero=new Estanquero(estanco);
        Fumador[] fumadores=new Fumador[3];

        fumadores[0]=new Fumador(0,estanco);
        fumadores[1]=new Fumador(1,estanco);
        fumadores[2]=new Fumador(2,estanco);

        estanquero.thr.start();
        fumadores[0].thr.start();
        fumadores[1].thr.start();
        fumadores[2].thr.start();
    }
}

```

Salida del programa:

```

yang@yang-VirtualBox:~/Escritorio/p2$ java Fumadores
El estanquero ha generado el ingrediente 2
El fumador 2 esta fumando con ingrediente 2
El estanquero ha generado el ingrediente 0
El fumador 0 esta fumando con ingrediente 0
El estanquero ha generado el ingrediente 2
El fumador 2 esta fumando con ingrediente 2
El estanquero ha generado el ingrediente 2
El fumador 2 esta fumando con ingrediente 2
El estanquero ha generado el ingrediente 1
El fumador 1 esta fumando con ingrediente 1
El estanquero ha generado el ingrediente 0
El fumador 0 esta fumando con ingrediente 0
El estanquero ha generado el ingrediente 1
El fumador 1 esta fumando con ingrediente 1
El estanquero ha generado el ingrediente 1
El fumador 1 esta fumando con ingrediente 1
El estanquero ha generado el ingrediente 1

```