

```
package clases;
```

```
/*
```

NO salió bien. no salió dinámico, ya que al poner el while en true se me morían los hilos

```
*/
```

```
public class ClasePrincipal {
```

```
    public static void main(String[] args){
```

```
        int n=0,aleatorio;
```

```
        gato hilo1=new gato();
```

```
        gato hilo4=new gato();
```

```
        gato hilo11=new gato();
```

```
        // Thread hilo2=new Thread(new raton());
```

```
        raton hilo2=new raton();
```

```
        raton hilo5=new raton();
```

```
        raton hilo6=new raton();
```

```
        comer hilo3=new comer();
```

```
        persona hilo7=new persona();
```

```
        persona hilo10=new persona();
```

```
        nopasanada hilo8=new nopasanada();
```

```
        comeraton hilo9=new comeraton();
```

```
    /*
```

no supe como administrar cada gato en forma de hilo

```
//generar 5 gatos
```

```
hilo1.ValordeCondicion(5)
```

```
*/
```

```
while(n<5){
```

```
//metodo para arracar hilos
```

```
aleatorio =(int) (Math.random()*5);
```

```
if(aleatorio==1){
```

```
    hilo1.start();
```

```
    hilo7.start();
```

```
    hilo8.start();
```

```
try{
```

```
hilo1.sleep(1000);
```

```
}catch(InterruptedException e){
```

```
    System.out.println("error "+ e);
```

```
}
```

```
}
```

```
if(aleatorio==2){
```

```
    hilo4.start();
```

```
    hilo3.start();
```

```
//bloquear por tiempo definido 1s es 1000
```

```
try{
```

```
hilo3.sleep(1000);  
}catch(InterruptedException e){  
    System.out.println("error "+ e);  
}  
  
}
```

```
if(aleatorio==3){  
    hilo11.start();  
    hilo10.start();  
    hilo9.start();
```

```
    try{  
        hilo9.sleep(2000);  
    }catch(InterruptedException e){  
        System.out.println("error "+ e);  
    }  
}
```

```
n++;  
  
System.out.println("NO pasa nada"+" "+n);  
  
}///fin while  
  
//////////fin//////////
```

```

    }

}

////////////////////////////////////

public class comer extends Thread{

    @Override

    public void run (){

        System.out.println("Tengo hambre...comiendo...." + " "+getName());

    }

}

////////////////////////////////////

public class comeraton extends Thread {

    @Override

    public void run (){

        System.out.println("hay un raton...come raton!!!"+" "+getName());

    }

}

////////////////////////////////////

public class gato extends Thread{

    //int valor;

    @Override

    public void run (){

        System.out.println("soy un gato"+" "+getName());

        /*

        metodo

        try{

            gato.sleep(1000)

```

```

    }catch(InterruptedException e){
        System.out.println("error " + e);
    }
    */
    /*
    for(int i=1; i<5;i++){
        System.out.println("soy un gato"+" "+i);
    }
    public void ValordeCondicion(int valor1){
        this.valor=valor1;
    }
    */
    }
}

////////////////////////////////////

public class nopasanada extends Thread {

    @Override
    public void run (){
        System.out.println("gato ronronea"+" "+getName());
    }

}

////////////////////////////////////

public class persona extends Thread {

```

```

@Override

public void run (){

System.out.println("Dueño de gato"+" "+getName());

}

}

////////////////////////////////////

public class raton extends Thread{

@Override

public void run (){

System.out.println("soy un raton"+" "+getName());

/*

metodo

try{

gato.sleep(1000)

}catch(InterruptedException e){

System.out.println("error "+ e);

}

*/

}

}

```