

EJERCICIO TARRO DE GALLETAS.

Un tarro de galletas, y 'N' niños accediendo al bote y cogiendo una galleta, cuando se vacía, los niños avisan a la madre para que reponga el tarro con galletas.

```
package TarroGalletas;

import java.util.concurrent.Semaphore;
import java.util.*;
class missem {
    public static int galletas=0;
    public static Semaphore nohaygalletas= new Semaphore (0);
    public static Semaphore haygalletas= new Semaphore (0);
    public static Semaphore mutex= new Semaphore (1);
}
class nino extends Thread {

    nino(int i){
        this.setName("nino"+i);
    }
    public void run () {
        try{
            while(true) { //Los ninos siempre estarán comiendo galletas
                //Aquí evitamos que dos procesos comprueben el número de
                galletas a la vez, ya que podría darse una falsa lectura
                sleep(1000);
                System.out.println("soy el " + this.getName() + " entrando");
                missem.mutex.acquire();
                System.out.println("soy el " + this.getName() + " paso
mutex");

                if(missem.galletas == 0) { //Si no hay galletas...
                    missem.nohaygalletas.release(); //Desbloquea a la madre
                    para que rellene
                    missem.haygalletas.acquire(); //Espera a que la madre
                    rellene para continuar
                    System.out.println("soy el " + this.getName() + " y Me
                    como una galleta. Quedan " + missem.galletas);

                    missem.galletas--; //coge galleta

                }else{
                    System.out.println("soy el " + this.getName() + " y Me
                    como una galleta. Quedan "+ missem.galletas);

                    missem.galletas--; //una galleta menos

                }
                missem.mutex.release();
            }
        } catch (InterruptedException e) {e.printStackTrace();}
    }
}
```

```

class mama extends Thread {

    mama(){
        this.setName("mama");
    }
    public void run () {
        try{
            while(true) {//La madre siempre esperando para reponer

                missem.nohaygalletas.acquire(); //Aquí se bloquea por que
el semáforo "NoHayGalletas" está a 0
                System.out.println("soy " + this.getName() + " paso
nohaygalletas. mutex vale" + missem.mutex.getQueueLength());
                //missem.mutex.acquire();

                System.out.println("soy " + this.getName() + " y relleno con
10 galletas.");
                missem.galletas=10; //Relleno el tarro de galletas
                //missem.mutex.release(); //El semáforo mutex impide que 2
procesos comprueben la variable "galletas" a la vez
                missem.haygalletas.release(); //Aviso a los niños de que
hay galletas poniendo a 1 el semáforo "HayGalletas"
            }
        } catch (InterruptedException e) {e.printStackTrace();}
    }
}

public class TarroGalletas {

    private static nino a_ninos[];

    public static void main(String[] args) {

        int num_ninos= 40;

        a_ninos=new nino[num_ninos];

        System.out.println( "voy a lanzar los niños...");
        //Creando y lanzando niños
        for (int i=0;i<num_ninos;i++){
            a_ninos[i]= new nino(i+1);
            a_ninos[i].start();
        }
        //Creando y Lanzando a la Madre

        System.out.println(" Voy a lanzar a la mama...");
        mama mimama=new mama();
        mimama.start();
    }
}

```