

# Guarderia-Junio2017Semaforos.pdf



**TTronc0**



**Programación de Sistemas y Concurrencia**



**2º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingeniería Informática  
Universidad de Málaga**

```

package Junio2017Sem;

import java.util.concurrent.Semaphore;

public class Guarderia {

    private int numBebes = 0;
    private int numAdultos = 0;
    private int quiereSalirAdulto = 0;
    private int quiereSalirBebe = 0;

    private int entraBebe = 0;

    private Semaphore mutex = new Semaphore(1);
    private Semaphore bebe = new Semaphore(1);
    private Semaphore adulto = new Semaphore(1);

    private Semaphore esperaBebe = new Semaphore(0);
    private Semaphore esperaAdulto = new Semaphore(0);

    /**
     * Un bebe que quiere entrar en la guarderia llama a este
metodo. Debe esperar
     * hasta que sea seguro entrar, es decir, hasta que cuado
entre haya, al menos,
     * 1 adulto por cada 3 bebes
     */
    public void entraBebe(int id) throws InterruptedException {

        System.out.println("El bebe " + id + " espera para
entrar");
        bebe.acquire();
        mutex.acquire();

        if ((numBebes + 1) > (3 * numAdultos)) {
            entraBebe = 1;

            bebe.release();
            mutex.release();
            esperaBebe.acquire();

            bebe.acquire();
            mutex.acquire();
        }

        numBebes++;
        System.out.println("El bebe " + id + " entra en la
guarderia. " + "Hay " + numBebes + " bebes y " + numAdultos
            + " adultos");

        mutex.release();
    }
}

```

```

        bebe.release();
    }

    /**
     * Un bebe que quiere irse de la guarderia llama a este
metodo *
     */
    public void saleBebe(int id) throws InterruptedException {

        System.out.println("El bebe " + id + " quiere salir");
        bebe.acquire();
        mutex.acquire();

        if ((numBebes - 1) > (3 * numAdultos)) {
            quiereSalirBebe = 1;

            mutex.release();
            esperaBebe.acquire();
            mutex.acquire();
        }

        numBebes--;
        System.out.println(
            "El bebe " + id + " sale de la guarderia. Hay
" + numBebes + " bebes y " + numAdultos + " adultos");

        mutex.release();
        bebe.release();
    }

    /**
     * Un adulto que quiere entrar en la guarderia llama a este
metodo *
     */
    public void entraAdulto(int id) throws InterruptedException {

        adulto.acquire();
        mutex.acquire();

        numAdultos++;
        System.out.println(
            "El adulto " + id + " entra en la guarderia.
Hay " + numBebes + " bebes y " + numAdultos + " adultos");

        if (entraBebe == 1 && (numBebes <= (3 * numAdultos))) {
            esperaBebe.release();
            entraBebe = 0;
        } else if (quiereSalirBebe == 1) {
            esperaBebe.release();
            quiereSalirBebe = 0;
        } else if (quiereSalirAdulto == 1) {

```

```

        esperaAdulto.release();
        quiereSalirAdulto = 0;
    }

    mutex.release();
    adulto.release();
}

/**
 * Un adulto que quiere irse de la guarderia llama a este
metodo. Debe esperar
 * hasta que sea seguro salir, es decir, hasta que cuando se
vaya haya, al
 * menos, 1 adulto por cada 3 bebes
 *
 */
public void saleAdulto(int id) throws InterruptedException {

    System.out.println("El adulto " + id + " solicita salir
de la guarderia");
    adulto.acquire();
    mutex.acquire();

    if (numBebes > (3 * (numAdultos - 1))) {
        System.out.println("Todavia no puedo salir");
        quiereSalirAdulto = 1;

        adulto.release();
        mutex.release();
        esperaAdulto.acquire();

        adulto.acquire();
        mutex.acquire();
    }

    numAdultos--;
    System.out.println(
        "El adulto " + id + " sale de la guarderia.
Hay " + numBebes + " bebes y " + numAdultos + " adultos");

    mutex.release();
    adulto.release();
}
}

```