

## Bloque 3. Paradigmas de la Programación

### Tarea 3: Simular el problema de las cabinas de votación

#### A tener en cuenta:

- Usar únicamente locks explícitos y conditions.
- Es un productor-consumidor.
- Tendremos 4 clases: Main, 2 clases para el productor (cabinas de votación) , 1 para el consumidor (sistema central de recuento) y por último la clase compartida.
- Main: crea el objeto compartido, 2 productores y 2 consumidores, y los lanza .
- VCompartida: objeto compartido con operaciones de escribirVoto y leerVoto.
- Productor: genera una comanda y la inserta en el ArrayList.
- Consumidor: consume una comanda y la elimina del ArrayList.

#### Código Clase Main:

```
public class Main {

    public static void main(String[] args) {
        VCompartida vc = new VCompartida(20); //Clase común al productor y consumidor
        //2 Hilos para la clase Productora
        Cabina c1 = new Cabina("Voto 1", vc);
        Cabina c2 = new Cabina("Voto 2", vc);
        //1 Hilo para la clase Consumidora, el sistema central de recuento
        Sistema s1 = new Sistema(vc);

        c1.start();
        c2.start();
        s1.start();
    }
}
```

#### Código Clase Productor - Cabina:

```
public class Cabina extends Thread{
    private String id;
    private VCompartida vc;

    public Cabina(String id, VCompartida vc) {
        this.id = id;
        this.vc = vc;
    }

    public void run() {
        String voto;
        int i = 0;
        while (true) { // Tendremos infinitos votos
            try {
                voto = id + " ---->" + i;
                i++;
            }
        }
    }
}
```

```

        vc.escribirVoto(voto);
        sleep(500);
    } catch (InterruptedException e) {
    }
}
}
}
}
}

```

### **Código Clase Consumidor - Sistema:**

```

public class Sistema extends Thread {
    private VCompartida vc;

    public Sistema(VCompartida vc) {
        this.vc = vc;
    }

    public void run() {
        String voto;
        while (true) { // Tendremos infinitos votos
            try {
                voto = vc.leerVoto();
                System.out.println("El sistema prepara ->" + voto); //Imprimiremos 20 lecturas de temperatura
                sleep(400);
            } catch (InterruptedException e) {
            }
        }
    }
}
}

```

### **Código Clase VCompartida:**

```

import java.util.ArrayList;
import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;

public class VCompartida {
    private ArrayList<String> arl; //Array de comandas
    private int maximo = 0, in = 0, out = 0, numElem = 0;
    Lock cerrojo = new ReentrantLock();
    private Condition lleno = cerrojo.newCondition();
    private Condition vacio = cerrojo.newCondition();

    public VCompartida(int max) {
        this.maximo = max;
        this.arl = new ArrayList<>();
    }
}

```

```

public void escribirVoto(String voto) throws InterruptedException {
    cerrojo.lock();
    while (numElem == maximo) { //Si el ArrayList está lleno... Espera
        lleno.await();
    }
    try {
        //Vamos añadiendo los votos de la clase productor
        arl.add(voto);
        numElem++;
        in = (in + 1) % maximo;

        vacio.signal(); //El ArrayList ya no está vacío
    } finally {
        cerrojo.unlock();
    }
}

public String leerVoto() throws InterruptedException {
    cerrojo.lock();
    while (numElem == 0) { //Si el ArrayList está vacío... Espera
        vacio.await();
    }
    try {
        String voto = arl.get(0);
        arl.remove(0);
        numElem = numElem - 1;
        out = (out + 1) % maximo;
        lleno.signal(); //El ArrayList ya no está lleno
        return (voto);
    } finally {
        cerrojo.unlock();
    }
}
}

```

### Resultado:

```

El sistema prepara ->Voto 2 ---->6
El sistema prepara ->Voto 1 ---->6
El sistema prepara ->Voto 2 ---->7
El sistema prepara ->Voto 1 ---->7
El sistema prepara ->Voto 2 ---->8
El sistema prepara ->Voto 1 ---->8
El sistema prepara ->Voto 2 ---->9
El sistema prepara ->Voto 1 ---->9

```

El sistema prepara ->Voto 2 ---->10  
El sistema prepara ->Voto 1 ---->10  
El sistema prepara ->Voto 1 ---->11  
BUILD STOPPED (total time: 9 seconds)