

## Act 2.5.- Sumador - Restador sincronizado y alternativo

### Main

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Contador c = new Contador(c: 100);  
  
        Sumador suma = new Sumador(c);  
        Thread resta = new Thread(new Restador(c));  
  
        suma.start();  
        resta.start();  
        |  
        try{  
            suma.join();  
            resta.join();  
        } catch (InterruptedException ex) {  
            System.out.println(x: "Error Hilos");  
        }  
  
        System.out.println("El valor final de c es " + c.valor());  
    }  
  
}
```

### Contador

```
public class Contador {  
  
    private int c = 0;  
  
    public Contador(int c) {  
        this.c = c;  
    }  
  
    public synchronized void sumarSync() throws InterruptedException {  
        while (c % 2 != 0) {  
            wait();  
        }  
        c++;  
        System.out.printf(format: "%sSumador %d\n", args: "\u001B[31m", args: c);  
        notify();  
    }  
  
    public synchronized void restarSync() throws InterruptedException {  
        while (c % 2 == 0) {  
            wait();  
        }  
        c--;  
        System.out.printf(format: "%sRestador %d\n", args: "\u001B[32m", args: c);  
        notify();  
    }  
  
    public int valor() {  
        return c;  
    }  
  
}
```

## Restador

```
public class Restador implements Runnable {

    private final Contador c;

    public Restador(Contador c) {
        this.c = c;
    }

    @Override
    public void run() {

        for(int i = 0; i < 300; i++){
            try{
                c.restarSync();
                Thread.sleep((long) (Math.random() * 100 + 50));
            } catch (InterruptedException ex) {
                System.out.println(x: "ERROR");
            }
        }
    }
}
```

## Sumador

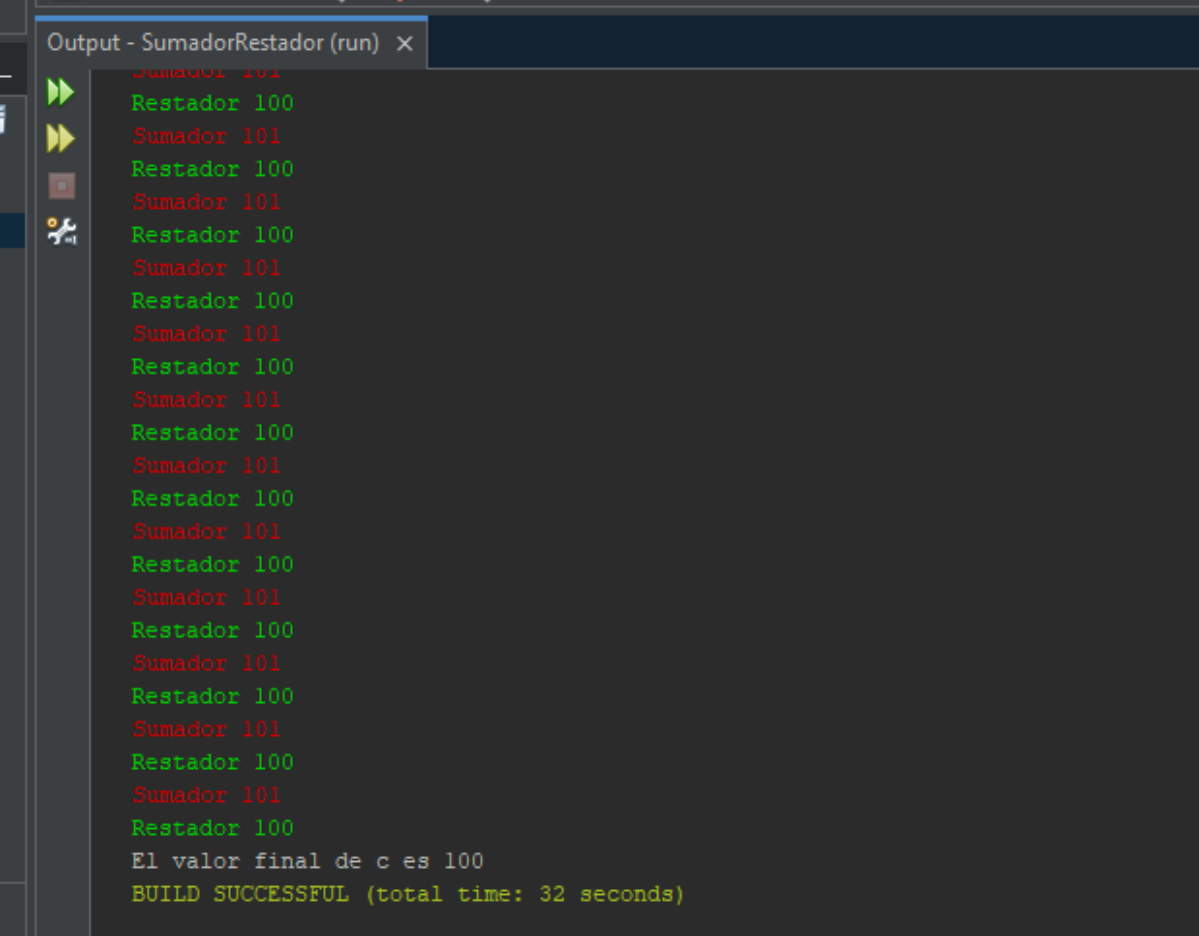
```
public class Sumador extends Thread{

    private final Contador c;

    public Sumador(Contador c){
        this.c=c;
    }

    @Override
    public void run(){
        for(int i = 0; i < 300; i++){
            try{
                c.sumarSync();
                Thread.sleep((long) (Math.random() * 100 + 50));
            } catch (InterruptedException ex) {
                System.out.println(x: "ERROR");
            }
        }
    }
}
```

## Resultado



```
Output - SumadorRestador (run) x
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
Sumador 101
Restador 100
El valor final de c es 100
BUILD SUCCESSFUL (total time: 32 seconds)
```