

# Programación de Servicios y Procesos

---

UT2: Actividad de colas con multihilos

Alexis López Briongos Dam2t

09/11/2023

**Índice**

Clase Cola .....	2
Clase Productor .....	3
Clase Consumidor .....	4
Clase Principal y Resultado.....	5

## Clase Cola

- Esta clase es la que se dedicará a gestionar las entradas y salidas de la clase Productor y Consumidor.

```
public class Cola {
    private String cadena;
    private boolean disponible = false;

    public synchronized String get() {
        while(disponible==false) {
            try {
                wait();
            } catch (InterruptedException e) {
                System.err.println("Error de hilo.");
            }
        }
        disponible=false;
        notify();
        return cadena;
    }

    public synchronized void put(String cadenaProductor) {
        while (disponible==true) {
            try {
                wait();
            } catch (Exception e) {
                System.err.println("Error de hilo.");
            }
        }
        disponible = true;
        cadena = cadenaProductor;
        notify();
    }
}
```

- Si el hilo consumidor utiliza el método get() entra en el método sincronizado, después en el bucle y se queda pausado hasta que el hilo productor introduzca un valor, lo despierte y ya pueda retornar el valor y despertar al hilo productor para poder permitirle que vuelva a meter otro valor en el método put(String valor).
- Si el productor utiliza el método put(String valor) entra en el método sincronizado, obvia el bucle, establece el valor a true, iguala el valor de la cadena a el valor introducido por el productor y realiza un notify(), por si el hilo consumidor está dormido pueda despertarlo.
- Si el hilo productor intenta entrar de nuevo al método put al estar el valor en true se quedara pausado hasta que el hilo consumidor llame al método get() y despierte al hilo productor.

## Clase Productor

```
public class Productor extends Thread {
    private Cola cola;
    private String cadena;

    public Productor(Cola c, String cadenaInput) {
        cola = c;
        this.cadena = cadenaInput;
    }

    @Override
    public void run() {
        for (int i = 0; i < 10; i++) {
            if (cadena == " PING ") {
                cadena = " PONG ";
            } else {
                cadena = " PING ";
            }
            cola.put(cadena);
        }

        try {
            sleep(100);
        } catch (Exception e) {
            System.err.println("Error de hilo.");
        }
    }
}
```

- En esta clase es la que se dedicará a introducir los valores que el consumidor va a recibir mediante la cola.
- Obtenemos el valor introducido al crear el hilo productor, y en el método run() introducimos el valor introducido en el constructor del hilo cambiándole el valor según el valor introducido , le agregamos un sleep para que realice la operación cada 100 milisegundos.

## Clase Consumidor

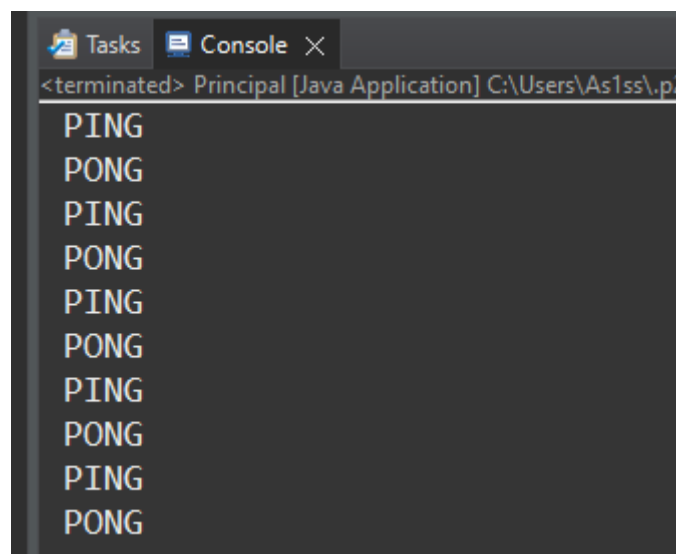
```
public class Consumidor extends Thread {  
    private Cola cola;  
    private String cadena;  
  
    public Consumidor(Cola c) {  
        this.colas = c;  
        cadena = "";  
    }  
  
    @Override  
    public void run() {  
        for (int i = 0; i < 10; i++) {  
            this.cadena = cola.get();  
            System.out.println(cadena);  
        }  
    }  
}
```

- Esta clase es la que recibirá los valores del productor mediante la cola y los mostrará por consola.
- Inicializamos la clase cola y el string cadena, mediante un bucle for obtenemos el valor de la cola del string, lo igualamos a nuestra variable cadena y mostramos por consola dicho valor.

## Clase Principal y Resultado

```
public class Principal {  
    public static void main(String[] args) {  
        Cola cola = new Cola();  
  
        Productor p = new Productor(cola, "PING");  
        Consumidor c = new Consumidor(cola);  
  
        c.start();  
        p.start();  
    }  
}
```

- Instanciamos una clase Cola e inicializamos dicha clase.
- Instanciamos una clase Productor y Consumidor y los inicializamos.
- Ejecutamos ambos hilos.



```
<terminated> Principal [Java Application] C:\Users\As1ss\p...  
PING  
PONG  
PING  
PONG  
PING  
PONG  
PING  
PONG  
PING  
PONG
```

- Resultado de la ejecución de ambos hilos.