

## Tarea 1

### 1.- Descripción de la tarea

#### 2.- Actividad 1. Multiprocesos

#### 3.- Actividad 2. Multihilos

### 2.- Información de interés

### 3.- Evaluación de la tarea

### Anexo: Licencia de recursos

Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](#)

# 1.- Descripción de la tarea

## Caso práctico

**Ada** confía plenamente en el trabajo que están realizado **Ana** y **Juan** en el proyecto de programación de un sistema de control climático de invernaderos.

Ada sabe que la programación **multihilo** y la programación **multiproceso** permiten ofrecer una solución potente y elegante para el proyecto, pero les recuerda que este tipo de programación debe planificarse cuidadosamente, ya que de lo contrario, esta técnica de programación puede producir errores inesperados difíciles de corregir.

Así que, en primer lugar, se debe analizar que solución es más apropiada para la resolución de su problema (programación multiproceso o multihilo) y, en segundo lugar, realizar una buena planificación y diseño.

## ¿Qué te pedimos que hagas?

Tendrás que llevar a cabo la **implementación de un sistema compuesto de varios ejercicios**. Se trabajarán con **múltiples procesos**, a los que se tendrán que **llamar a ejecución y compartir información** y otros que **trabajen con recursos compartido y, además, de forma sincronizada** utilizando hilos en Java. Para ello, tendrás que implementar y hacer funcionar las clases que se describen en los siguientes apartados.



## Tarea 1

### 1.- Descripción de la tarea

#### 2.- Actividad 1. Multiprocesos

#### 3.- Actividad 2. Multihilos

### 2.- Información de interés

### 3.- Evaluación de la tarea

### Anexo: Licencia de recursos

Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](#)

## 2.- Actividad 1. Multiprocesos

La actividad 1 se compone en 4 partes

- Primera parte. Implementa una aplicación, llamada 'cadenas', que genere cadenas formadas por caracteres del alfabeto (a-z). La cantidad de cadenas será indicada por el usuario al ejecutar la aplicación y su longitud será aleatoria (máximo 10 caracteres). Las cadenas se deben escribir mediante tubería establecida con la aplicación 'frecuencia'.
- Segunda parte. Implementa una aplicación, llamada 'frecuencia', que dada una cadena de texto recibida a través de la tubería obtenga la frecuencia absoluta de cada una de las vocales, es decir, la cantidad de veces que aparece cada vocal. En el caso de que una de las vocales no aparezca la frecuencia será 0. Por ejemplo: para la palabra "caramelo" el resultado debe ser 2 1 0 1 0
- Tercera parte. Para generar palabras parecidas al español y que no salgan combinaciones extrañas de muchas consonantes seguidas: podéis utilizar un array de sílabas de dos o tres letras que siempre contengan una vocal:

```
String[] SILABAS = {
    "ba", "be", "bi", "bo", "bu", "ca", "ce", "ci", "co", "cu",
}
```

Cuanto más completo creéis el array, más combinaciones posibles. Lo normal es que se generen cadenas que no tengan ningún significado pero que si tengan un parecido al español.

- Cuarta parte. Realiza un pequeño manual (tipo "¿Cómo se hace?") utilizando un procesador de textos en el que indiques, con pequeñas explicaciones y capturas, cómo has probado la ejecución de las aplicaciones que has implementado en este ejercicio. Entre las pruebas que hayas realizado, debes incluir una captura del resultado final donde debe tener el siguiente aspecto:

```
Cadena: caramelo --> 2 1 0 1 0
Cadena: cotimu --> 0 0 1 1 1
Cadena: tomate --> 1 1 0 1 0
```

En una captura de una ejecución de la prueba es **OBLIGATORIO** tener como fondo la plataforma con el perfil del alumno/a;

## Tarea 1

### 1.- Descripción de la tarea

#### 2.- Actividad 1. Multiprocesos

#### 3.- Actividad 2. Multihilos

### 2.- Información de interés

### 3.- Evaluación de la tarea

### Anexo: Licencia de recursos

Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](#)

## 3.- Actividad 2. Multihilos

En esta segunda actividad debemos crear un programa con dos hilos productores y un hilo consumidor.

Los hilos productores se encargarán de almacenar valores numéricos en un array, mientras que el hilo consumidor sacará elementos del array de uno a uno para ir sumándolos en una variable a la que llamaremos 'Suma'. El array tendrá una capacidad para almacenar 10 valores. Los valores introducidos por los productores serán valores aleatorios entre 0 y 10.

El programa tendrá que tener en cuenta lo siguiente:

- ✓ Se debe impedir la superposición de operaciones sobre secciones críticas.
- ✓ No se podrá consumir si el array está vacío o producir si está lleno.
- ✓ Si el array está lleno, deberán detenerse los hilos productores (instrucción wait) hasta que el hilo consumidor termine de leer el array. Cuando esto suceda, será este hilo quien vuelva a activarlos.
- ✓ Simularemos un tiempo de producción y de consumo en cada hilo. Es decir, los hilos deberán descansar después de producir (10 milisegundos) y después de consumir (150 milisegundos).
- ✓ Se establecerá el valor máximo a 100, para la suma de los elementos sacados del array por el hilo consumidor, de forma que si se alcanza este valor máximo se finalizarían los hilos creados y se visualizará el resultado obtenido.
- ✓ Tanto los hilos productores, como el hilo consumidor, comprobarán si se ha alcanzado el valor máximo para dar por finalizada su tarea.
- ✓ Se debe registrar en un fichero la siguiente información por cada acción: **Hilo ejecutado, operación realizada, el actual estado del vector y el valor de la variable 'Suma' en ese momento.**
  - Hilo productor 1, introduce el valor 6 en la posición 1, Array = [6], Suma=0
  - Hilo productor 2, introduce el valor 1 en la posición 2, Array = [6, 1], Suma=0
  - Hilo productor 1, introduce el valor 4 en la posición 3, Array = [6, 1, 4], Suma=0
  - Hilo consumidor, saca el valor 4 de la posición 3, Array = [6, 1, 4], Suma=4
  - Hilo consumidor, saca el valor 1 de la posición 2, Array = [6, 1, 4], Suma=5
  - ...
  - Hilo consumidor, saca el valor 8 de la posición 7, Array = [6, 5, 0, 5, 6], Suma=107
  - Resultado final: 107

Al igual que en el ejercicio anterior debes realizar un pequeño manual (tipo "¿Cómo se hace?"), utilizando un procesador de textos en el que indiques con pequeñas explicaciones y capturas, la forma en que has realizado la tarea.

En alguna capturas es **OBLIGATORIO** tener como fondo la plataforma con el perfil del alumno/a;