



Escuela Superior de
Tlahuelilpan

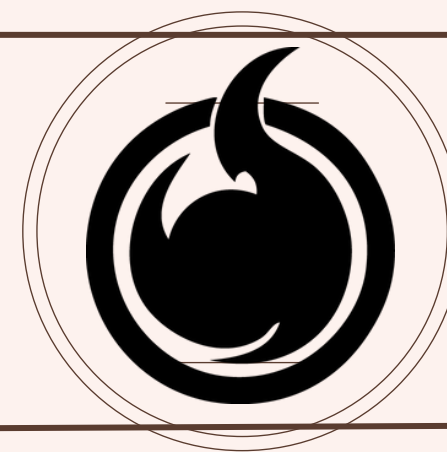


PABLO AXEL FLORES BAUTISTA

Lenguajes de programación

Licenciatura en ingeniería de software

Grupo 301. Periodo: Julio-Diciembre 2022



NÚMERO DE CUENTA: 385309

Práctica 2 Java.

1.- ¿Cuáles son las ventajas de utilizar la programación concurrente?

- Hacer un mejor uso de los procesadores de computadora.
- Velocidad de respuesta más rápida.
- Más fácil de desarrollar.

2.- ¿Cuáles son las desventajas de utilizar la programación concurrente?

- Aumentar el cambio de contexto, lo que afecta la velocidad de ejecución.
- Se produce un interbloqueo, lo que hace que el sistema no esté disponible.
- Las limitaciones de recursos afectan el rendimiento.

3.- ¿Son iguales la programación paralela y la concurrente? Explica ¿por qué?

No, porque no es la misma forma en que se realizan las tareas. Cuando se ejecutan tareas de forma concurrente a estas se les asigna un periodo de tiempo antes de cambiar de tarea, en dicho periodo se determinará si una tarea se inicie, continúe, o se complete la tarea, en contraparte, si ejecutamos tareas en paralelo, las tareas se realizarán de forma simultánea, comenzarán y finalizarán sin interrupciones.

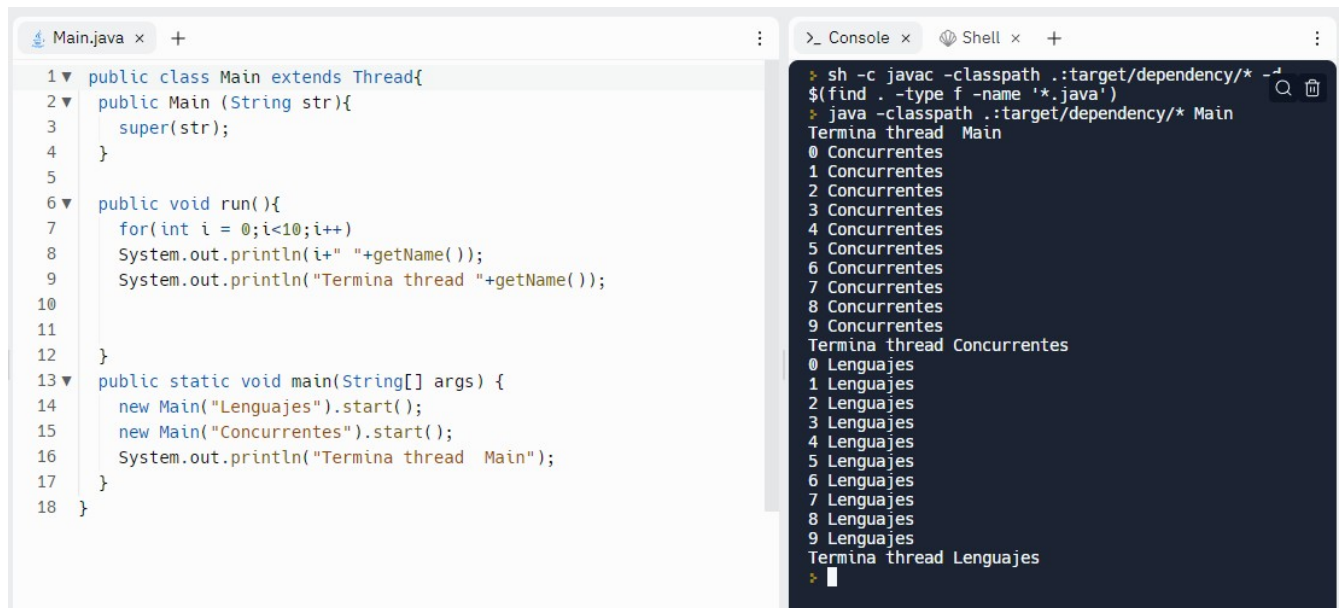
4.- ¿Cuáles son las diferencias entre la programación secuencial y la programación concurrente?

La programación secuencial (también llamada estructura secuencial), es un programa que hace que una instrucción siga a otra instrucción después de que esta haya finalizado.

La programación concurrente consiste en utilizar técnicas de programación que hagan que múltiples instrucciones de un programa funcionen al mismo tiempo.

En pocas palabras, la programación concurrente logra que múltiples instrucciones se ejecuten a la vez, logrando así mayor velocidad en la ejecución de nuestros programas.

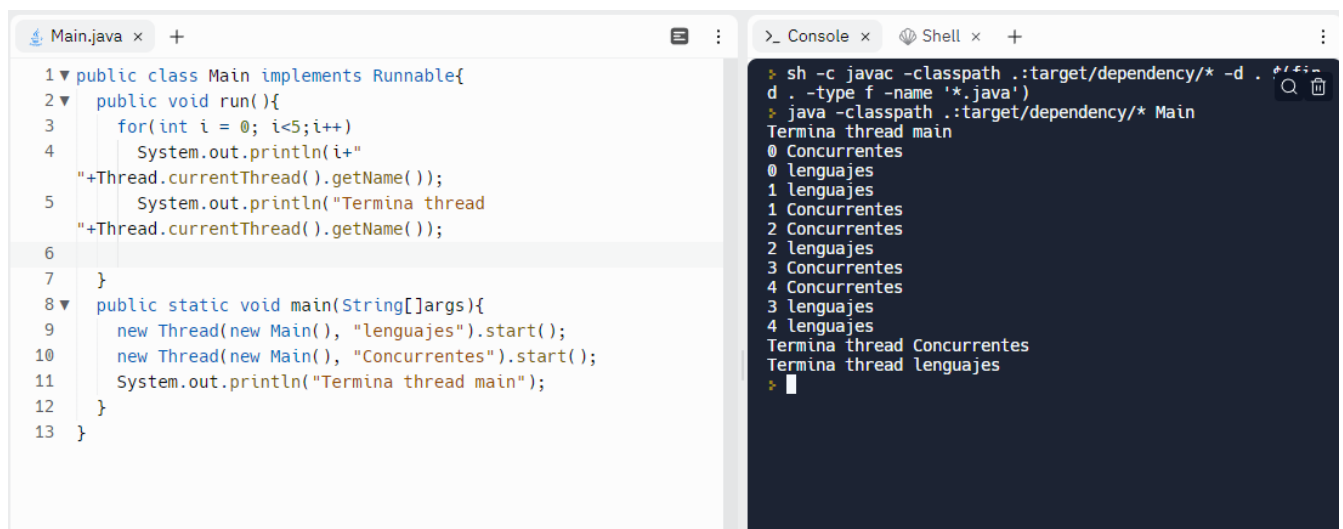
Evidencias de funcionamiento



The screenshot shows an IDE with two panels. The left panel displays the code for `Main.java`, which extends `Thread`. The right panel shows the console output of the program.

```
1 public class Main extends Thread{
2     public Main (String str){
3         super(str);
4     }
5
6     public void run(){
7         for(int i = 0; i<10; i++){
8             System.out.println(i+" "+getName());
9             System.out.println("Termina thread "+getName());
10        }
11    }
12
13    public static void main(String[] args) {
14        new Main("Lenguajes").start();
15        new Main("Concurrentes").start();
16        System.out.println("Termina thread Main");
17    }
18 }
```

```
>_ Console x Shell x +
> sh -c javac -classpath ./target/dependency/* -d . $(find . -type f -name '*.java')
> java -classpath ./target/dependency/* Main
Termina thread Main
0 Concurrentes
1 Concurrentes
2 Concurrentes
3 Concurrentes
4 Concurrentes
5 Concurrentes
6 Concurrentes
7 Concurrentes
8 Concurrentes
9 Concurrentes
Termina thread Concurrentes
0 Lenguajes
1 Lenguajes
2 Lenguajes
3 Lenguajes
4 Lenguajes
5 Lenguajes
6 Lenguajes
7 Lenguajes
8 Lenguajes
9 Lenguajes
Termina thread Lenguajes
```



The screenshot shows an IDE with two panels. The left panel displays the code for `Main.java`, which implements `Runnable`. The right panel shows the console output of the program.

```
1 public class Main implements Runnable{
2     public void run(){
3         for(int i = 0; i<5; i++){
4             System.out.println(i+"
5             "+Thread.currentThread().getName());
6             System.out.println("Termina thread
7             "+Thread.currentThread().getName());
8         }
9     }
10    public static void main(String[] args){
11        new Thread(new Main(), "Lenguajes").start();
12        new Thread(new Main(), "Concurrentes").start();
13        System.out.println("Termina thread main");
14    }
15 }
```

```
>_ Console x Shell x +
> sh -c javac -classpath ./target/dependency/* -d . $(find . -type f -name '*.java')
> java -classpath ./target/dependency/* Main
Termina thread main
0 Concurrentes
0 Lenguajes
1 Lenguajes
1 Concurrentes
2 Concurrentes
2 Lenguajes
3 Concurrentes
4 Concurrentes
3 Lenguajes
4 Lenguajes
Termina thread Concurrentes
Termina thread Lenguajes
```

Referencias

Ventajas y desventajas de la programación concurrente - programador clic. (s. f.). Recuperado 17 de octubre de 2022, de <https://programmerclick.com/article/67691919779/>

Programación secuencial VS Programación concurrente. (2021, 5 octubre). Recuperado 17 de octubre de 2022, de <https://tech.madirex.com/2021/10/programacion-secuencial-vs-programacion.html>