*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*

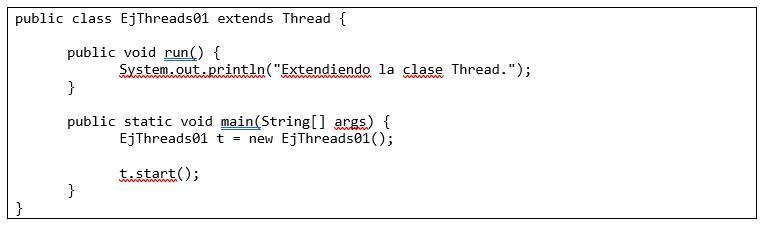


**Taller 1 – Threads**

El propósito de este taller es entender la forma como se crean los threads para implementar aplicaciones concurrentes. En este taller implementaremos el concepto en los lenguajes Java y Python. Comencemos por dar un vistazo a la implementación en el lenguaje Java. En Java existen dos formas de realizar la implementación de los threads, bien sea mediante la extensión de la clase **Thread** o la implementación de la interface **Runnable**.

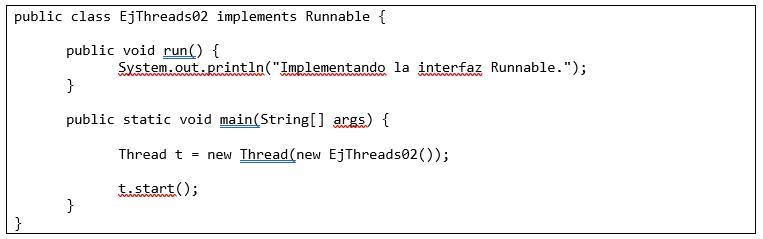
**Ejemplo 1: Creación de threads como extensión de la clase Thread**

Escriba el código de la clase **EjThreads01**.



**Ejemplo 2: Creación de threads como implementación de la interface Runnable**

Escriba el código de la clase **EjThreads02**.



***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

1

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



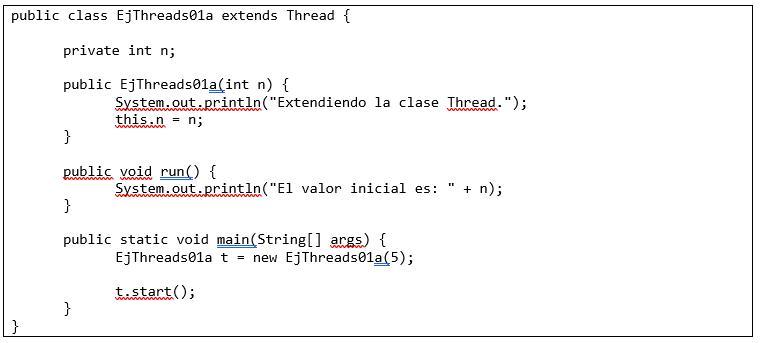
**Responda:**

1. Complete la siguiente tabla, con respecto a la creación de threads usando la extensión de la clase **Thread** y la implementación de la interface **Runnable**.

|  |  |
| --- | --- |
| **Se parecen** | **Se diferencian** |
| * Ambos hacen lo mismo | * Uno extiende la clase thread, mientras que el otro implementa Runnable * Su inicializaciónp |

**Ejemplo 3: Creación de threads con un valor inicial – como extensión de la clase Thread**

Escriba el código de la clase **EjThreads01a**.



**Ejemplo 4: Creación de threads con un valor inicial – como implementación de la interface Runnable**

Escriba el código de la clase **EjThreads02a**.

***Carlos Eduardo Gómez Montoya***

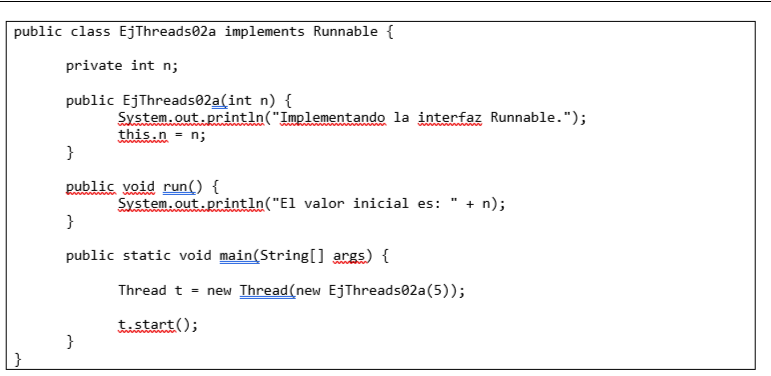
***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

2

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

3

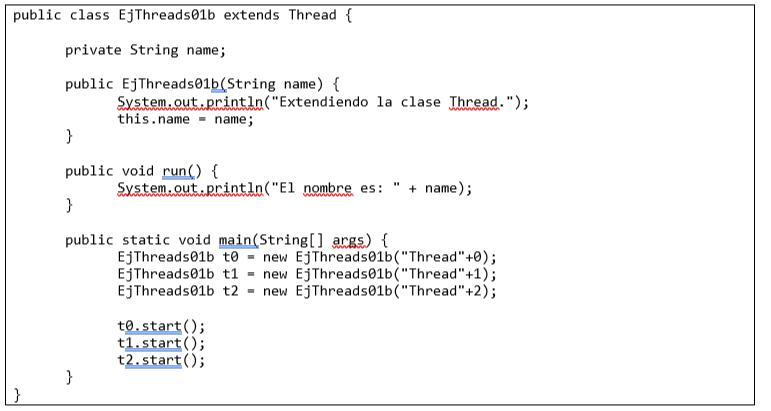
*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



**Ejemplo 5: Creación de threads con nombre – como extensión de la clase Thread**

Escriba el código de la clase **EjThreads01b**.



**Ejemplo 6: Creación de threads con nombre – como implementación de la interface Runnable**

Escriba el código de la clase **EjThreads02b**.

***Carlos Eduardo Gómez Montoya***

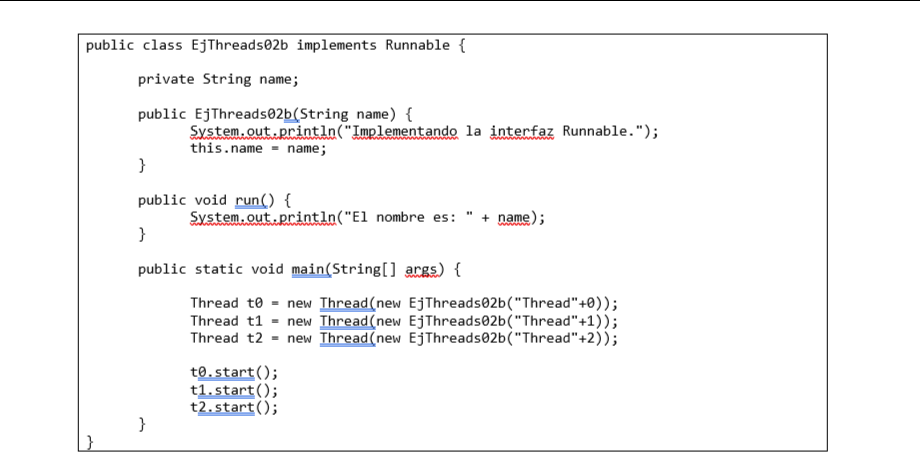
***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

4

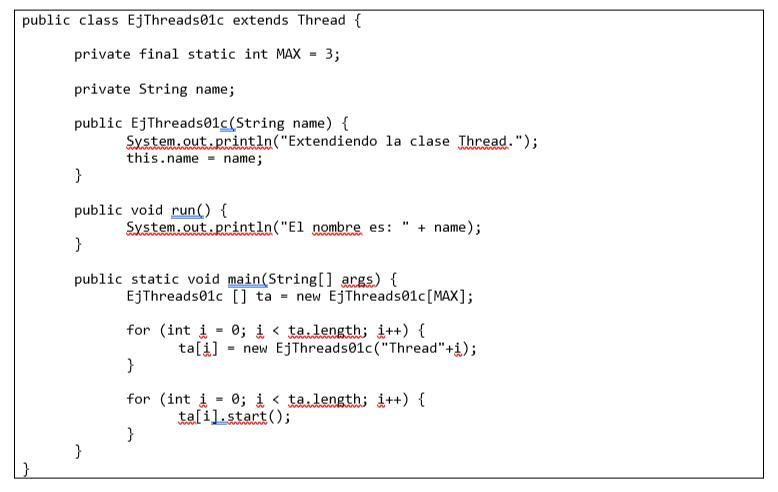
*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



**Ejemplo 7: Creación de threads con nombre usando ciclos – como extensión de la clase Thread**

Escriba el código de la clase **EjThreads01c**.



***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

5

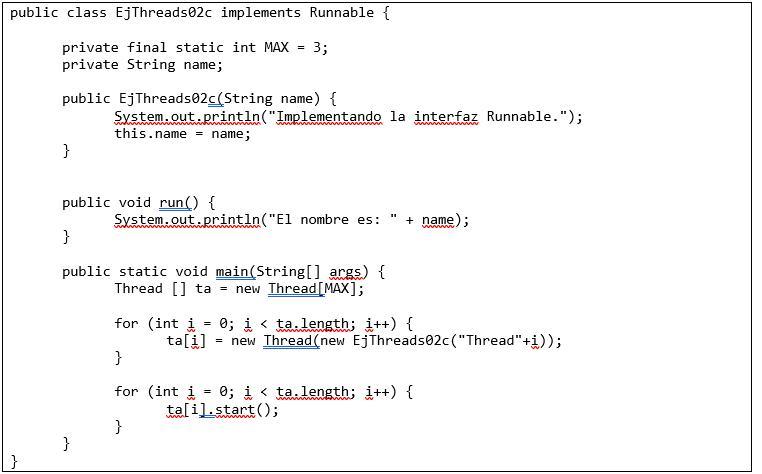
*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



**Ejemplo 8: Creación de threads con nombre usando ciclos – como implementación de la interface Runnable**

Escriba el código de la clase **EjThreads02c**.



**Ejemplo 9: Creación de threads con nombre usando ciclos – como extensión de la clase Thread**

Escriba el código de la clase **EjThreads01d**.

***Carlos Eduardo Gómez Montoya***

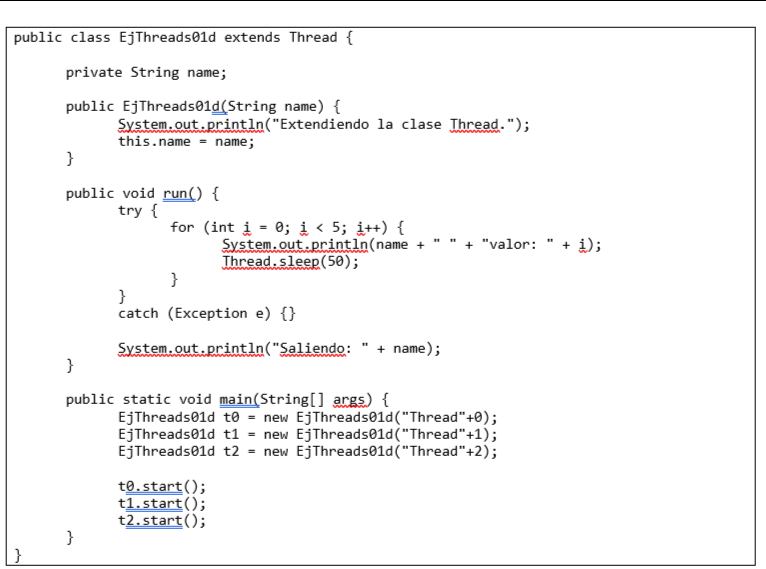
***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

6

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



**Ejemplo 10: Creación de threads con nombre usando ciclos – como implementación de la interface Runnable**

Escriba el código de la clase **EjThreads02d**.

***Carlos Eduardo Gómez Montoya***

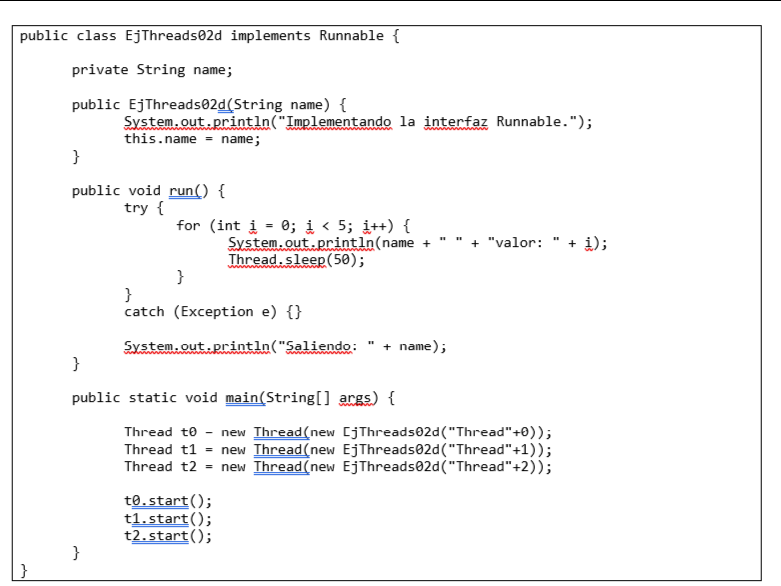
***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

7

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



A continuación, exploremos la implementación de los threads en Python. Existen dos maneras de crear threads en este lenguaje. La primera consiste en crear un objeto thread y enviar como referencia el proceso a invocar, lo anterior se expresa pasando un objeto que sea invocable (por ejemplo, una referencia a una función). La segunda opción consiste en escribir una clase que herede de la clase **Thread,** esta se encuentra disponible en el paquete **threading**. Luego se sobrescribe el método **run()** igual como lo hacemos en Java.

**Ejemplo 11: Creación de threads en Python utilizando referencias a objetos invocables**

***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

8

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



Como podemos observar la función se referencia en el argumento **target.** Los argumentos de la función se envian en una tupla mediante el parametro **args.**

**Ejemplo 12: Creación de threads en Python utilizando clases**

***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

9

*Ingeniería de Sistemas y Computación*

*ISIS 2203 Infraestructura Computacional*



**Ejercicio**

Desarrolle un programa en Java que tenga dos threads. El primer thread imprime los números pares entre 1 y un límite superior recibido por entrada estándar. El segundo thread imprime los números impares entre el mismo rango. Después de imprimir un número en la consola, el thread debe dormir por una cantidad de milisegundos especificada desde la creación del thread. Haga dos versiones de este programa, una utilizando la extensión de la clase **Thread** y otra mediante la implementación de la interface **Runnable** o alguna de las dos implementaciones existentes en Python.

***Carlos Eduardo Gómez Montoya***

***Jairo Emilio Bautista Mora***

***Geovanny Andrés González Rodríguez***

10