

```
1 import java.util.concurrent.*;
2
3 class SistemaNavegacion implements Callable<String> {
4     @Override
5     public String call() throws Exception {
6         Thread.sleep(1200);
7         return "🚀 Navegación: trayectoria corregida con éxito.";
8     }
9 }
10
11 class SistemaSoporteVital implements Callable<String> {
12     @Override
13     public String call() throws Exception {
14         Thread.sleep(900);
15         return "🩺 Soporte vital: presión y oxígeno dentro de parámetros normales.";
16     }
17 }
18
19 class SistemaControlTermico implements Callable<String> {
20     @Override
21     public String call() throws Exception {
22         Thread.sleep(1000);
23         return "🔥 Control térmico: temperatura estable (22°C).";
24     }
25 }
```



```

27 class SistemaComunicaciones implements Callable<String> {
28     @Override
29     public String call() throws Exception {
30         Thread.sleep(700);
31         return "📡 Comunicaciones: enlace con estación terrestre establecido.";
32     }
33 }
34
35 public class SimulacionMisionEspacial {
36     public static void main(String[] args) throws Exception {
37         System.out.println("🚀 Simulación de misión espacial iniciada...\n");
38
39         ExecutorService executor = Executors.newFixedThreadPool(4);
40
41         Future<String> futuroNavegacion = executor.submit(new SistemaNavegacion());
42         Future<String> futuroSoporteVital = executor.submit(new SistemaSoporteVital());
43         Future<String> futuroControlTermico = executor.submit(new SistemaControlTermico());
44         Future<String> futuroComunicaciones = executor.submit(new SistemaComunicaciones());
45
46         // Mostrar resultados (el orden puede variar)
47         System.out.println(futuroComunicaciones.get());
48         System.out.println(futuroSoporteVital.get());
49         System.out.println(futuroControlTermico.get());
50         System.out.println(futuroNavegacion.get());
51     }

```

Solo se admiten operaciones con lápiz

```
33 }
34
35 public class SimulacionMisionEspacial {
36     public static void main(String[] args) throws Exception {
37         System.out.println("🚀 Simulación de misión espacial iniciada...\n");
38
39         ExecutorService executor = Executors.newFixedThreadPool(4);
40
41         Future<String> futuroNavegacion = executor.submit(new SistemaNavegacion());
42         Future<String> futuroSoporteVital = executor.submit(new SistemaSoporteVital());
43         Future<String> futuroControlTermico = executor.submit(new SistemaControlTermico());
44         Future<String> futuroComunicaciones = executor.submit(new SistemaComunicaciones());
45
46         // Mostrar resultados (el orden puede variar)
47         System.out.println(futuroComunicaciones.get());
48         System.out.println(futuroSoporteVital.get());
49         System.out.println(futuroControlTermico.get());
50         System.out.println(futuroNavegacion.get());
51
52         executor.shutdown();
53
54         System.out.println("\n✅ Todos los sistemas reportan estado operativo.");
55     }
56 }
57
```



≡ Console



🚀 Simulación de misión espacial iniciada...

📡 Comunicaciones: enlace con estación terrestre establecido.
🩺 Soporte vital: presión y oxígeno dentro de parámetros normales.
🔥 Control térmico: temperatura estable (22°C).
🧭 Navegación: trayectoria corregida con éxito.

✅ Todos los sistemas reportan estado operativo.

