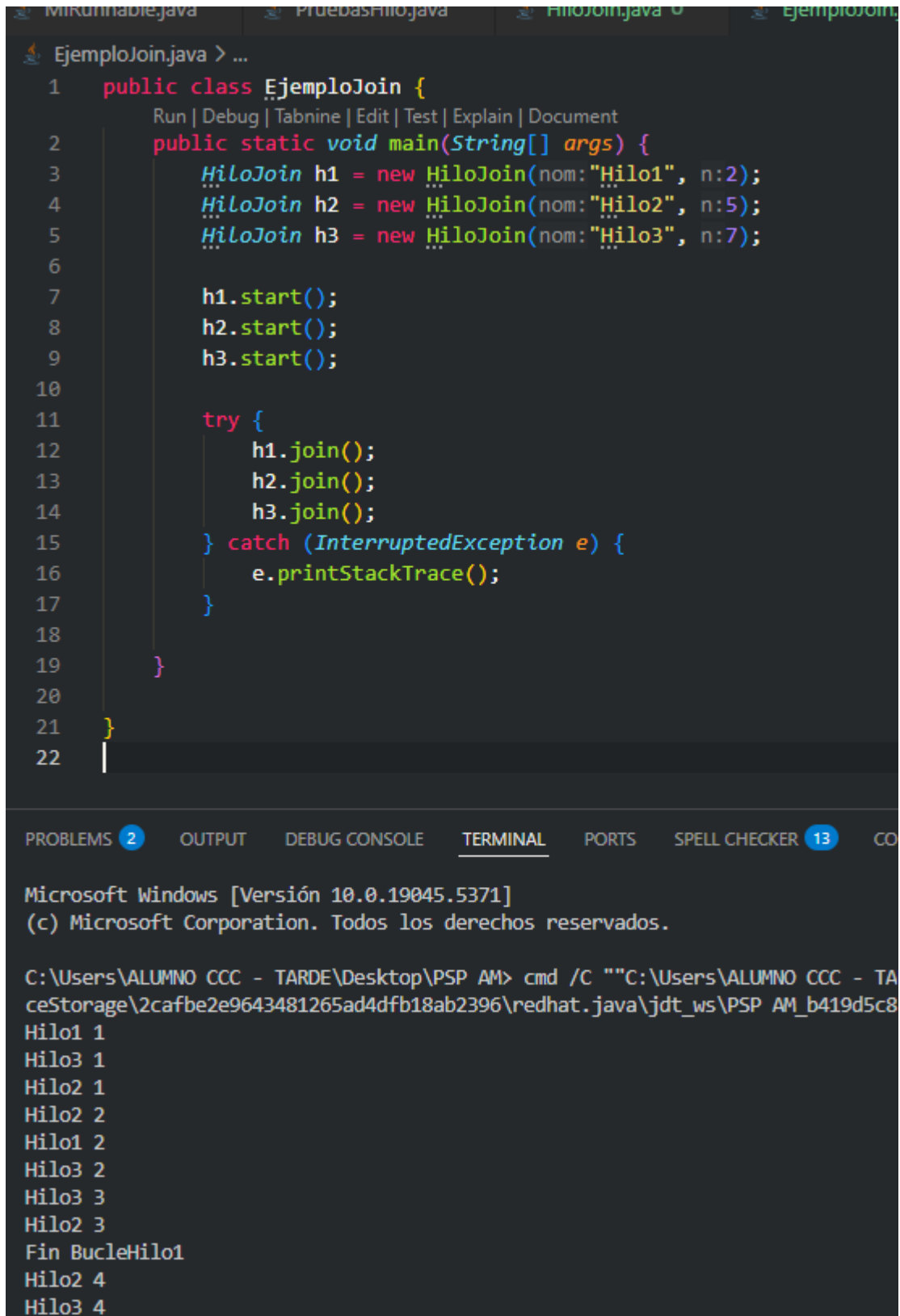


1- En primera instancia, tenemos el código que se ejecuta y comporta así :



The screenshot shows an IDE with a Java file named `EjemploJoin.java`. The code defines a `public class EjemploJoin` with a `main` method. Inside `main`, three `HiloJoin` objects are created: `h1` (nom: "Hilo1", n: 2), `h2` (nom: "Hilo2", n: 5), and `h3` (nom: "Hilo3", n: 7). Each object's `start()` method is called. Then, a `try` block calls `h1.join()`, `h2.join()`, and `h3.join()`. A `catch` block for `InterruptedException` prints the stack trace. The terminal output shows the execution of these threads, with `Hilo1` and `Hilo3` completing their loops before `Hilo2` finishes its loop and prints "Fin BucleHilo1".

```
1 public class EjemploJoin {
2     public static void main(String[] args) {
3         HiloJoin h1 = new HiloJoin(nom: "Hilo1", n: 2);
4         HiloJoin h2 = new HiloJoin(nom: "Hilo2", n: 5);
5         HiloJoin h3 = new HiloJoin(nom: "Hilo3", n: 7);
6
7         h1.start();
8         h2.start();
9         h3.start();
10
11        try {
12            h1.join();
13            h2.join();
14            h3.join();
15        } catch (InterruptedException e) {
16            e.printStackTrace();
17        }
18    }
19 }
20
21 }
22
```

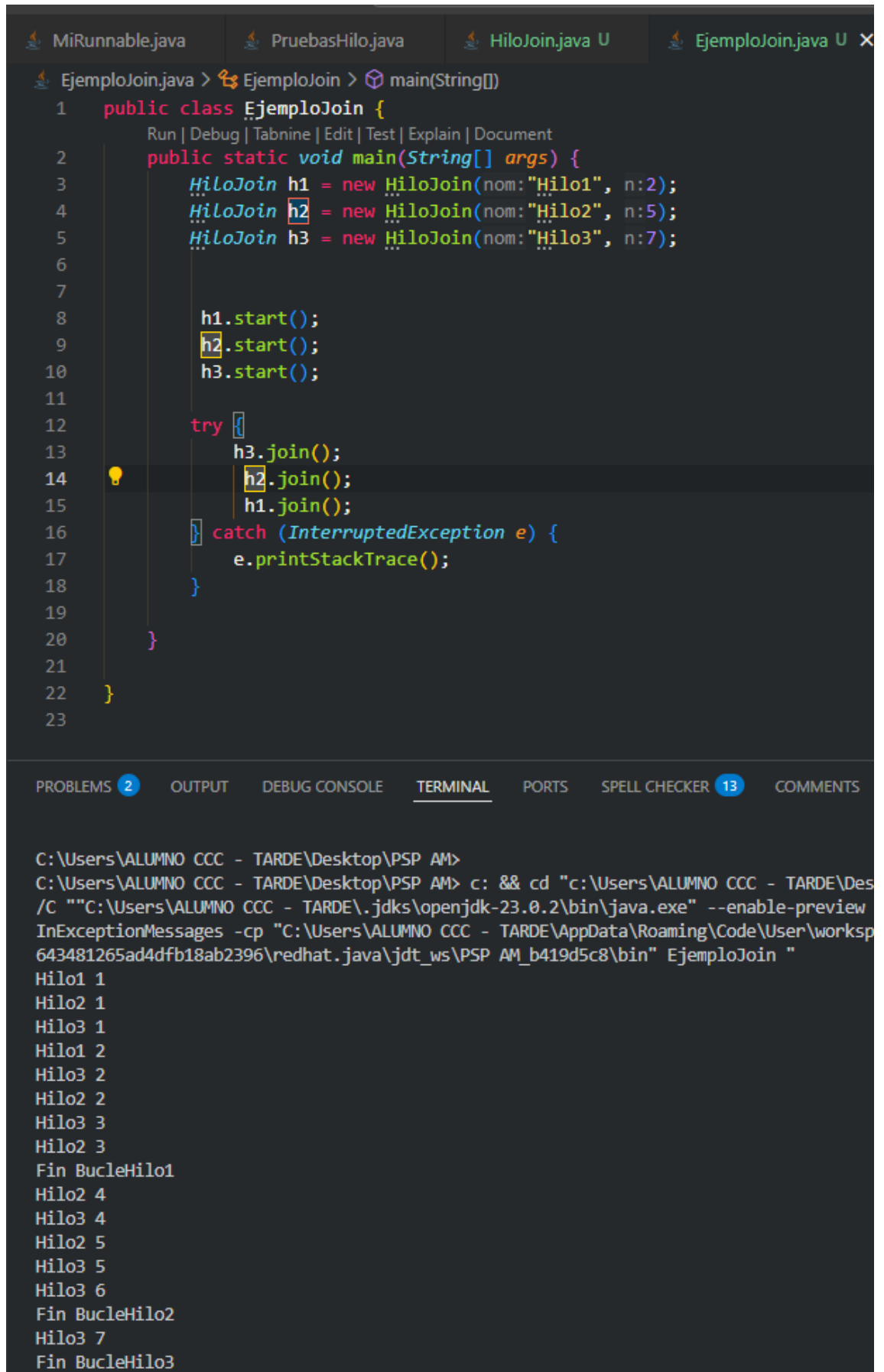
Microsoft Windows [Versión 10.0.19045.5371]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ALUMNO CCC - TARDE\Desktop\PSP AM> cmd /C ""C:\Users\ALUMNO CCC - TA
ceStorage\2cafbe2e9643481265ad4dfb18ab2396\redhat.java\jdt_ws\PSP AM_b419d5c8
Hilo1 1
Hilo3 1
Hilo2 1
Hilo2 2
Hilo1 2
Hilo3 2
Hilo3 3
Hilo2 3
Fin BucleHilo1
Hilo2 4
Hilo3 4

Donde se evidencia que el HILO1 se ejecuta (1-2) veces, el HILO2 se ejecuta (1-2-3-4-5) veces y por último el HILO3 (1-2-3-4-5-6-7), hasta finalizar cada hilo.

El uso de `join()` asegura que el hilo principal no continuará hasta que todos los hilos hayan terminado. Sin embargo, los hilos pueden ejecutarse en cualquier orden, ya que eso lo decide el sistema operativo.

2- Si cambiamos el orden de ejecución, se vería de la siguiente manera



The screenshot shows an IDE with the file `EjemploJoin.java` open. The code defines a `main` method that creates three `HiloJoin` objects (`h1`, `h2`, `h3`) with different parameters and starts them. It then uses a `try` block to join them in the order `h3`, `h2`, and `h1`. The terminal output shows the execution of these threads, with `h3` completing first, followed by `h2`, and finally `h1`.

```
1 public class EjemploJoin {
2     public static void main(String[] args) {
3         HiloJoin h1 = new HiloJoin(nom:"Hilo1", n:2);
4         HiloJoin h2 = new HiloJoin(nom:"Hilo2", n:5);
5         HiloJoin h3 = new HiloJoin(nom:"Hilo3", n:7);
6
7
8         h1.start();
9         h2.start();
10        h3.start();
11
12        try {
13            h3.join();
14            h2.join();
15            h1.join();
16        } catch (InterruptedException e) {
17            e.printStackTrace();
18        }
19
20    }
21
22 }
23
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 13 COMMENTS

```
C:\Users\ALUMNO CCC - TARDE\Desktop\PSP AM>
C:\Users\ALUMNO CCC - TARDE\Desktop\PSP AM> c: && cd "c:\Users\ALUMNO CCC - TARDE\Des
/C ""C:\Users\ALUMNO CCC - TARDE\.jdk\openjdk-23.0.2\bin\java.exe" --enable-preview
InExceptionMessages -cp "C:\Users\ALUMNO CCC - TARDE\AppData\Roaming\Code\User\worksp
643481265ad4dfb18ab2396\redhat.java\jdt_ws\PSP AM_b419d5c8\bin" EjemploJoin "
Hilo1 1
Hilo2 1
Hilo3 1
Hilo1 2
Hilo3 2
Hilo2 2
Hilo3 3
Hilo2 3
Fin BucleHilo1
Hilo2 4
Hilo3 4
Hilo2 5
Hilo3 5
Hilo3 6
Fin BucleHilo2
Hilo3 7
Fin BucleHilo3
```

En este caso, se ve un cambio, porque se están ejecutando los JOIN y hasta que estos no se dejen de ejecutar no se termina la secuencia de los hilos

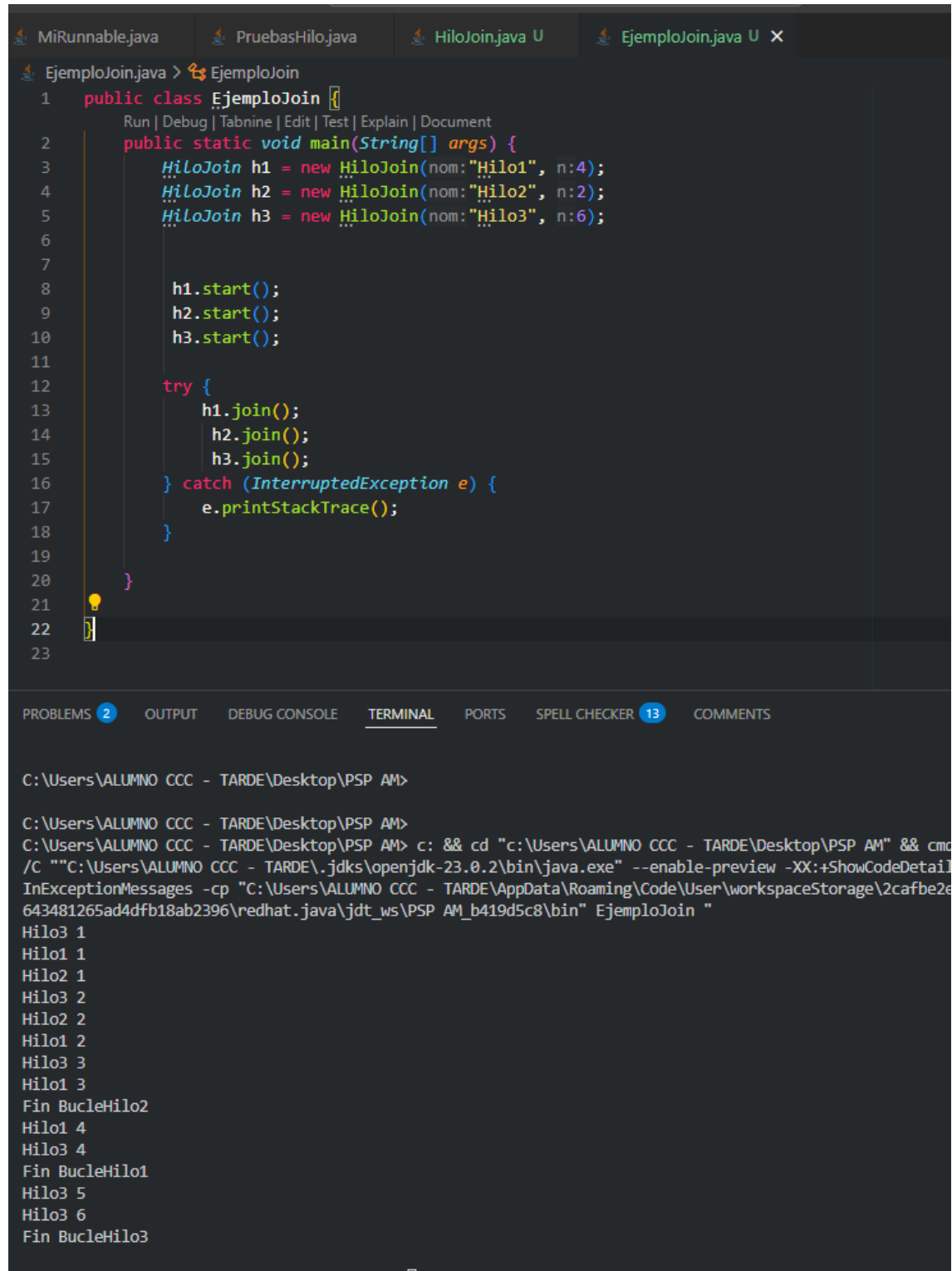
3- Tenemos los nombre de los hilos y veces así:

```
run | debug | runtime | edit | test | explain | document  
public static void main(String[] args) {  
    HiloJoin h1 = new HiloJoin(nom: "Hilo1", n: 2);  
    HiloJoin h2 = new HiloJoin(nom: "Hilo2", n: 5);  
    HiloJoin h3 = new HiloJoin(nom: "Hilo3", n: 7);  
}
```

Y si lo cambiamos por esto:

```
public static void main(String[] args) {  
    HiloJoin h1 = new HiloJoin(nom: "Hilo1", n: 4);  
    HiloJoin h2 = new HiloJoin(nom: "Hilo2", n: 2);  
    HiloJoin h3 = new HiloJoin(nom: "Hilo3", n: 6);  
}
```

Ahora el HILO1 ira de (1 a 4) HILO2 de (1 a 2) y por ultimo HILO3 (de 1 a 6), para tener un resultado así:



The screenshot shows an IDE with three tabs: `MiRunnable.java`, `PruebasHilo.java`, and `HiloJoin.java U`. The active tab is `EjemploJoin.java U`, which contains the following Java code:

```
1 public class EjemploJoin {
2     Run | Debug | Tabnine | Edit | Test | Explain | Document
3     public static void main(String[] args) {
4         HiloJoin h1 = new HiloJoin(nom: "Hilo1", n:4);
5         HiloJoin h2 = new HiloJoin(nom: "Hilo2", n:2);
6         HiloJoin h3 = new HiloJoin(nom: "Hilo3", n:6);
7
8         h1.start();
9         h2.start();
10        h3.start();
11
12        try {
13            h1.join();
14            h2.join();
15            h3.join();
16        } catch (InterruptedException e) {
17            e.printStackTrace();
18        }
19    }
20 }
21
22
23
```

The bottom panel shows the `TERMINAL` output:

```
C:\Users\ALUMNO CCC - TARDE\Desktop\PSP AM>
C:\Users\ALUMNO CCC - TARDE\Desktop\PSP AM>
C:\Users\ALUMNO CCC - TARDE\Desktop\PSP AM> c: && cd "c:\Users\ALUMNO CCC - TARDE\Desktop\PSP AM" && cmd
/c ""C:\Users\ALUMNO CCC - TARDE\.jdk\openjdk-23.0.2\bin\java.exe" --enable-preview -XX:+ShowCodeDetail
InExceptionMessages -cp "C:\Users\ALUMNO CCC - TARDE\AppData\Roaming\Code\User\workspaceStorage\2cafbe2e
643481265ad4dfb18ab2396\redhat.java\jdt_ws\PSP AM_b419d5c8\bin" EjemploJoin "
Hilo3 1
Hilo1 1
Hilo2 1
Hilo3 2
Hilo2 2
Hilo1 2
Hilo3 3
Hilo1 3
Fin BucleHilo2
Hilo1 4
Hilo3 4
Fin BucleHilo1
Hilo3 5
Hilo3 6
Fin BucleHilo3
```

Como el hilo 3 es el mas grande es el primero en ejecutarse, ya que el orden del JOIN el mismo sistema obliga a que este sea el primero en ejecutarse ya que el bucle de los demás es menor, entonces por ende este empieza primero para que pueda terminar y cumplir todo su ciclo.

