

Eduardo Rojo Sánchez



Servicios y Procesos

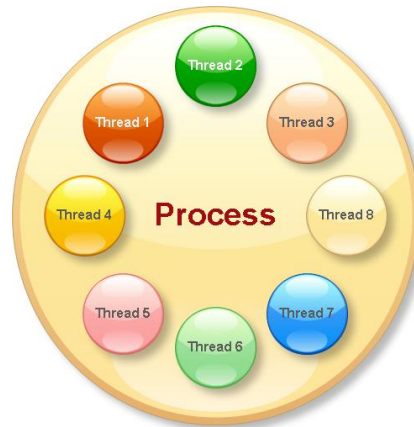
2º CFGS DAM



2. Ejercicio Hilo.

Objetivo

- Crear **dos thread** que saque por pantalla 10 veces el nombre del *thread creado*.





Package Explorer

Hilo1
Hilo2

Crea otro proyecto.

Problems Declaration Console LogCat Devices

<terminated> SimpleHilo [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (21 de sept. de 2015 13:02:48)

proceso.java

*DosHilo.java

Declaro 2 variables.

objeto

```
// Declaro dos variable de la clase proceso.  
proceso miHilo;
```

```
// Creo un objeto de la clase proceso.  
// Le doy al hilo el nombre de Hilo1.  
miHilo = new proceso("Hilo1");
```

```
// Llamo al método start(), heredado de la super clase thread,  
// que se encarga de llamar al metodo run().  
miHilo.start();
```

```
}
```

proceso.java

DosHilo.java

```
1 package hilo;
2
3 public class DosHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9         // Declaro dos variable de la clase proceso.
10        proceso miHilo;
11        proceso segundoHilo;
12
13
14        // Creo un objeto de la clase proceso.
15        // Le doy al hilo el nombre de Hilo1.
16        miHilo = new proceso("Hilo1");
17
18        // Llamo al método start(), heredado de la super clase thread,
19        // que se encarga de llamar al metodo run().
20        miHilo.start();
21
22    }
23
24 }
```

```
1 package hilo;
2
3 public class DosHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9
10
11
12
13
14         // Creo un objeto de la clase proceso.
15         // Le doy al hilo el nombre de Hilo1.
16         miHilo = new proceso("Hilo1");
17         segundoHilo = new proceso("Hilo2");
18
19
20         // Llamo al método start(), heredado de la super clase thread,
21         // que se encarga de llamar al metodo run().
22         miHilo.start();
23
24     }
25 }
```

Crear otro objeto y lo llamo Hilo2.

proceso.java

DosHilo.java

```
3 public class DosHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9         // Declaro dos variable de la clase proceso.
10        proceso miHilo;
11
12
13
14
15
16
17        segundoHilo = new proceso("Hilo2");
18
19
20        // Llamo al método start(), heredado de la super clase thread,
21        // que se encarga de llamar al metodo run().
22        miHilo.start();
23        segundoHilo.start();
24    }
25
26 }
27
```

Llamo al método **Start()**.

proceso.java

DosHilo.java

```
3 public class DosHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9         // Declaro dos variable de la clase proceso.
10        proceso miHilo;
11
12
13
14
15
16
17        segundoHilo = new proceso("Hilo2");
18
19
20        // Llamo al método start(), heredado de la super clase thread,
21        // que se encarga de llamar al metodo run().
22        miHilo.start();
23        segundoHilo.start();
24    }
25
26 }
27
```

Se ejecutarán en
paralelo.

Haz clic en
Run.

Package Explorer

Hilo1
Hilo2
src
hilo
DosHilo.java
proceso.java
JRE System Library [JavaSE-1.8]

proceso.java DosHilo.java

```
1 public class DosHilo
2 {
3
4     public static void main(String[] args) {
5         // Para poner en marcha este nuevo thread se debe crear un objeto
6         // de la clase proceso que acabamos de contruir.
7
8         // Declaro dos variable de la clase proceso.
9         proceso miHilo;
10        proceso segundoHilo;
11
12
13        // Creo un objeto de la clase proceso.
14        // Le doy al hilo el nombre de Hilo1.
15        miHilo = new proceso("Hilo1");
16        segundoHilo = new proceso("Hilo2");
17
18
19        // Llamo al método start(), heredado de la super clase thread,
20        // que se encarga de llamar al metodo run().
21        miHilo.start();
22        segundoHilo.start();
23    }
24 }
25
26 }
27
```

Problems Declaration Console LogCat Devices

<terminated> SimpleHilo [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (21 de sept. de 2015 13:02:48)

No se ve que se este
ejecutando
simultáneamente.

[illegible]



Package Explorer

> Hilo1
> Hilo2

proceso.java

DosHilo.java

```
public class DosHilo {  
  
    public static void main(String[] args) {  
        // Para poner en marcha este nuevo thread se debe crear un objeto  
        // de la clase proceso que acabamos de contruir.  
  
        // Declaro dos variable de la clase proceso.
```

Lo ejecuto de nuevo.

Este es el thread : Hilo1

Este es el thread : Hilo2

Este es el thread : Hilo2

Este es el thread : Hilo2

Este es el thread : Hilo2

Este es el thread : Hilo2

Este es el thread : Hilo2

Este es el thread : Hilo2

Este es el thread : Hilo2

Este es el thread : Hilo2

Este es el thread : Hilo1

Este es el thread : Hilo1

Este es el thread : Hilo1

Este es el thread : Hilo1

Este es el thread : Hilo1

Este es el thread : Hilo1

Este es el thread : Hilo1

Este es el thread : Hilo1

Ves que se ejecuta el proceso 1, luego el proceso 2 y luego el proceso 1 de nuevo. No hay orden, es simultaneo.

Funcionan para aplicaciones gráficas.

[illegible]

Si creas una aplicación gráfica es necesario realizar acciones en diferentes procesos.