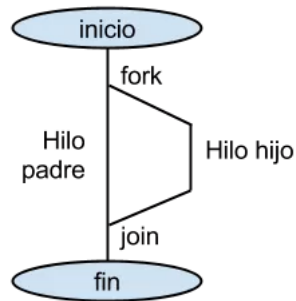


Hilos



La programación multi hilo es una técnica necesaria para aprovechar el poder de computo de las nuevas tecnologías y para diseñar software más modular y multi tarea. Sin la programación con hilos no podrías estar viendo un video mientras se descarga, por poner un ejemplo.

La clase Thread

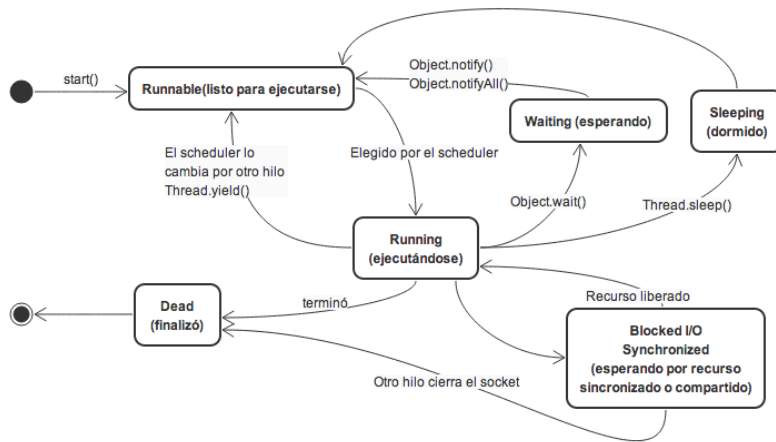
La clase Thread en java es la que nos permite definir hilos en nuestro programa. La clase Thread implementa la clase Runnable. Cualquier clase que extienda de la clase Thread debe de implementar el método run()

Ejemplo:

```
public class Hilo extends Thread {
    String mensaje;
    public Hilo(String mensaje){
        this.mensaje = mensaje;
    }

    public void run () {
        System.out.println(mensaje);
    }
}
```

En el ejemplo se crea la clase Hilo que extiende de la clase Thread, posee un constructor y el método obligatorio run(). Lo único que va hacer este hilo es recibir un mensaje e imprimirlo en pantalla cuando se ejecute.



Estados de la clase Thread

Para crear hilos usando la clase Hilo definida anteriormente, tan solo se debe instanciar un objeto de ese tipo y llamar a su método run(), automáticamente se creará un proceso para atender esa porción de código y se ejecutará a la par del programa principal.

```

public class Main {
    public static void main(String args[]){
        Hilo hilo1 = new Hilo("Hola desde hilo 1");
        hilo1.run();
    }
}
  
```

Los hilos son porciones de código ejecutándose a la par del programa principal, pero si el hilo del programa padre termina, los hijos también lo harán aunque no hayan acabado su trabajo. Si se quiere evitar ese inconveniente es necesario que el hilo padre espere a los hijos antes de terminarse; para eso usamos el método Thread.join()

```

public class Main {
    public static void main(String args[]){
        Hilo hilo1 = new Hilo("Hola desde hilo 1");
        hilo1.run();
    }
}

```

Los hilos son porciones de código ejecutándose a la par del programa principal, pero si el hilo del programa padre termina, los hijos también lo harán aunque no hayan acabado su trabajo. Si se quiere evitar ese inconveniente es necesario que el hilo padre espere a los hijos antes de terminarse; para eso usamos el método `Thread.join()`

```

public class Main {
    public static void main(String args[]){
        Hilo hilo1 = new Hilo("Hola desde hilo 1");
        hilo1.run();
        try {
            hilo1.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

A esta forma de trabajar con hilos se le conoce como el modelo fork/join(separa y une)

<https://www.taringa.net/posts/hazlo-tu-mismo/16291610/Reloj-digital-con-fecha-en-java-clase-paso-a-paso.html>