OpenWebinars

HILOS Y SU CICLO DE VIDA

CLASE THREAD

 Cada hilo de ejecución de nuestras aplicaciones se asocia a una instancia de <u>Thread</u>.

Sus métodos básicos son:

- public void run(): contiene el código que queremos ejecutar en el hilo. NO se debe invocar nunca directamente.
- public void start(): lanza la ejecución del hilo.

CLASE THREAD

```
public class PrimoThread extends Thread {
          //propiedades y constructor
          public void run() {
             long n = minimo;
  Código
             while(!testPrimalidad(n)) {
  que se
               System.out.printf("%d no es primo %n", n);
ejecutará
               ++n;
 en otro
    hilo
             System.out.printf("El número primo es %d %n", n);
          public static boolean testPrimalidad(long n) {
             //cuerpo del método
```

EJECUCIÓN DE UN THREAD

```
public class Ejemplo02Primo {
            public static void main(String[] args) {
               PrimoThread pt01 = new PrimoThread(1234567);
               pt01.start();
               PrimoThread pt02 = new PrimoThread(23456789);
  El método
               pt02.start();
  run() no se
               PrimoThread pt03 = new PrimoThread(34567891);
     invoca
               pt03.start();
directamente
```

La ejecución de los 3 hilos se solapará

CICLO DE VIDA DE UN THREAD

The Thread constructor is called to create a new instance of the Thread class The thread can become The start method is blocked for various reasons New called to designate the and will not run again until it is thread as runnable returned to the Runnable state **Blocked** The Java thread scheduler runs the thread as the processor Runnable becomes available Waiting The thread ends when the run method terminates If the thread calls the wait method, it is put into the Terminated Waiting state and will remain there until another thread calls the notify or notifyAll method.

CICLO DE VIDA DE UN THREAD

- 1. Se llama al constructor de Thread para crear el nuevo hilo.
- 2. Se llama al método **start** para designarlo como ejecutable
- El planificador lo ejecuta en cuanto el procesador está disponible.

CICLO DE VIDA DE UN THREAD

- 4. El hilo puede pasar a **bloqueado** por diferentes razones, y no vuelve hasta que pasa de nuevo a **ejectuable.**
- 5. Si se utiliza el método wait, se pone en estado de espera, y permanece ahí hasta que se ejecuta notify o notifyAll.
- 6. El hilo termina cuando finaliza la ejecución de su método **run**.

PAUSAR LA EJECUCIÓN DE UN THREAD

- Usando el métodoThread.sleep(long milis).
- ► Es un método estático
- Pausa el hilo que actualmente está ejecutandose.