



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
IC	2003-1		Programación Orientada a Objetos I

PRÁCTICA No.	LABORATORIO DE		DURACIÓN (HORA)
	NOMBRE DE LA PRÁCTICA	Hilos	2

1 INTRODUCCIÓN

Los hilos (Threads) proporcionan a un programa, una manera de realizar varias tareas de manera concurrente. Un **proceso** es un programa ejecutándose de forma independiente y con un espacio propio de memoria. Un Sistema Operativo multitarea es capaz de ejecutar más de un **proceso** simultáneamente. Un **thread** o **hilo** es un **flujo secuencial simple** dentro de un **proceso**. Un único **proceso** puede tener varios **hilos** ejecutándose.

2 OBJETIVO (COMPETENCIA)

Conocer las ventajas de la programación multihilos y aplicarla en la resolución de problemas prácticos.

Formuló	Revisó	Aprobó	Autorizó
M.C. Jorge Eduardo Ibarra Esquer	M.C. Gloria Etelbina Chávez Valenzuela		
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formato para prácticas de laboratorio

3 FUNDAMENTO

Se presentan varias situaciones en las cuales tiene sentido utilizar hilos dentro de un programa en Java. Algunos programas deben ser capaces de realizar varias actividades a la vez que siguen respondiendo a entrada de datos adicional por parte del usuario. Por ejemplo, un navegador web debe tener la capacidad de responder a las acciones del usuario mientras descarga un documento o reproduce un sonido.

Si un programa modela objetos del mundo real que presentan un comportamiento autónomo e independiente, entonces la opción es utilizar un hilo separado para cada uno de esos objetos.

A pesar de existir muchas ventajas al utilizar hilos, también hay algunas razones por las cuales limitar su uso. Si la lógica de un programa es enteramente secuencial, donde el fin de una operación provoca el inicio de la siguiente y se debe esperar a que esta termine de ejecutarse para continuar con el programa, entonces un solo hilo puede implementar la secuencia completa. Utilizar hilos en una situación como la anterior resultaría en un programa más complejo y sin beneficios adicionales.

Cuando varios hilos comparten objetos, estos deben utilizar mecanismos de sincronización para coordinar el acceso de los diferentes hilos a sus atributos y, de esta forma, mantener un estado consistente. Estos mecanismos agregan complejidad a los programas y, si no se tiene el cuidado adecuado, pueden volverse una fuente de fallos.

La clase `java.lang.Thread`

La clase `Thread` del paquete `java.lang` crea y controla hilos en programas escritos en Java. La ejecución de código en Java siempre se encuentra bajo el control de un objeto `Thread`.

En **Java** hay dos formas de crear nuevos **hilos**. La primera de ellas consiste en crear una nueva clase que herede de la clase **`java.lang.Thread`** y sobrecargar el método **`run()`** de dicha clase. El segundo método consiste en declarar una clase que implemente la interfaz **`java.lang.Runnable`**, la cual declarará el método **`run()`**; posteriormente se crea un objeto de tipo **`Thread`** pasándole como argumento al constructor el objeto creado de la nueva clase (la que implementa la interfaz **`Runnable`**).

En cualquiera de los dos casos, lo que se requiere es sobrescribir el método **`public void run()`**, el cual es el equivalente al **`main()`** para un hilo. Para crear el hilo y ejecutar el método **`run()`** se llama al método **`start()`**.

La clase que se muestra a continuación ejemplifica algunos de los conceptos del trabajo con hilos en Java.

Formuló	Revisó	Aprobó	Autorizó
M.C. Jorge Eduardo Ibarra Esquer	M.C. Gloria Etelbina Chávez Valenzuela		
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

3 FUNDAMENTO

```
public class Corredor extends Thread{

    private long tiempo=-1;
    private int miLugar=0;
    private static boolean empieza=false;
    private static int lugar=1;

    public Corredor(long tiempo, String nombre) {
        this.tiempo=tiempo;
        setName(nombre);
    }

    public void run(){
        while(!empieza);
        try{
            this.sleep(tiempo);
        }
        catch(InterruptedException e){
            e.printStackTrace();
        }
        finaliza();
    }

    private synchronized void finaliza(){
        miLugar=lugar;
        lugar++;
    }

    public static void setEmpieza(boolean b){
        empieza=b;
    }

    public void resultado(){
        System.out.println(getName()+" - "+miLugar);
    }

    public static void main(String[] args) {
        Corredor c1=new Corredor(10000,"Juanelo");
        Corredor c2=new Corredor(12000,"Sancho");
        Corredor c3=new Corredor(9800,"Ludovico");
        Corredor c4=new Corredor(9850,"Clodomiro");
```

Formuló	Revisó	Aprobó	Autorizó
M.C. Jorge Eduardo Ibarra Esquer	M.C. Gloria Etelbina Chávez Valenzuela		
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

3 FUNDAMENTO

```

c1.start();
c2.start();
c3.start();
c4.start();

Corredor.setEmpieza(true);
try{
    c1.join();
    c2.join();
    c3.join();
    c4.join();
}
catch(InterruptedException e){
    e.printStackTrace();
}

c1.resultado();
c2.resultado();
c3.resultado();
c4.resultado();
}
}

```

4 PROCEDIMIENTO (DESCRIPCIÓN)

A EQUIPO NECESARIO	MATERIAL DE APOYO
Computadoras con una versión reciente del JDK	

Formuló	Revisó	Aprobó	Autorizó
M.C. Jorge Eduardo Ibarra Esquer	M.C. Gloria Etelbina Chávez Valenzuela		
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

4 PROCEDIMIENTO (DESCRIPCIÓN)

B DESARROLLO DE LA PRÁCTICA

1. Investiga lo siguiente:
 Funcionamiento de los métodos ***setName()***, ***join()*** y ***sleep()*** de la clase Thread
 Utilidad del modificador ***synchronized***
2. Escribe el código de ejemplo que se muestra en la sección de fundamentos.
3. Antes de ejecutarlo, ¿cuál crees que será el resultado?
4. En el método ***main()***, modifica los valores que se envían al constructor de los objetos de la clase Corredor. Observa los resultados.
5. Modifica la clase para que en lugar de extender a Thread, la ejecución de los hilos se haga implementando la interfaz Runnable

Formuló	Revisó	Aprobó	Autorizó
M.C. Jorge Eduardo Ibarra Esquer	M.C. Gloria Etelbina Chávez Valenzuela		
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

C CÁLCULOS Y REPORTE

5 RESULTADOS Y CONCLUSIONES

6 ANEXOS

Formuló	Revisó	Aprobó	Autorizó
M.C. Jorge Eduardo Ibarra Esquer	M.C. Gloria Etelbina Chávez Valenzuela		
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad