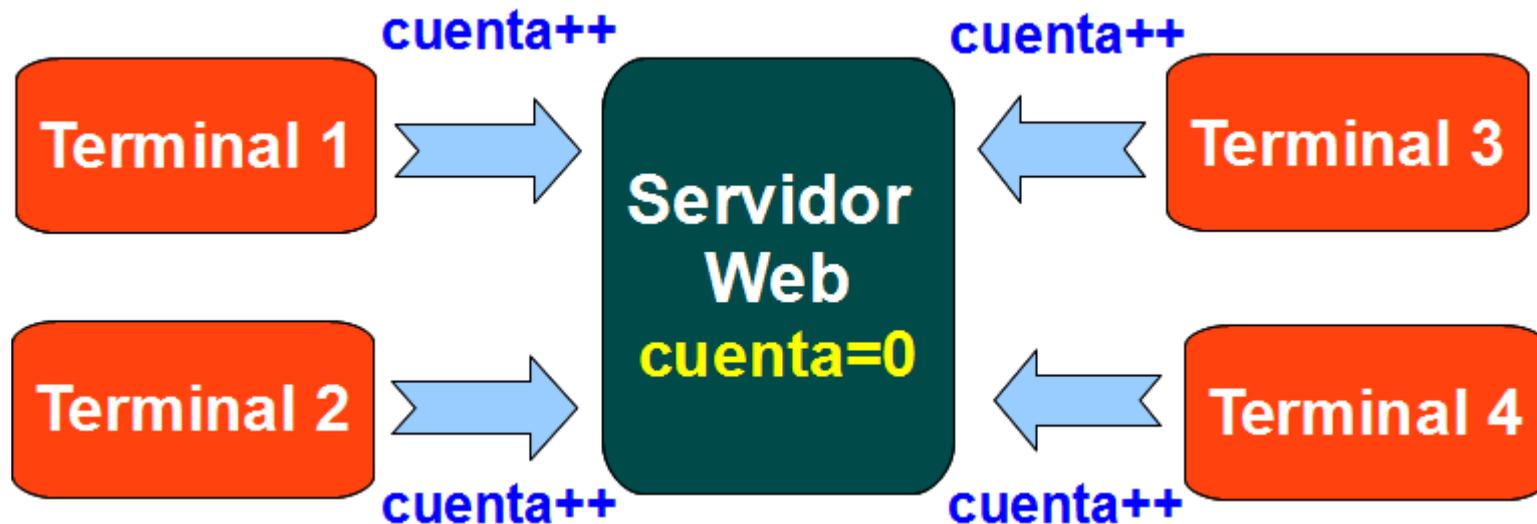


Sincronizar método

Uso de monitores

Enunciado del problema

- Se trata de contabilizar los accesos actuales a un servidor Web. Simularemos este hecho mediante un **recurso compartido**, la variable **cuenta** que se incrementa en cada acceso nuevo al Servidor.
- Cada Terminal es un hilo.
- Supondremos que en el momento de partida hay 0 accesos.



Clases que creamos

- **ServidorWeb.** Clase que implementa el recurso correspondiente al servidor y que contiene el método `incrementaCuenta()` que se encarga de incrementar en 1 el contador `cuenta` que controla el número de accesos al servidor. Establecemos como `synchronized` el método `incrementaCuenta()` que en este caso es el recurso que compartirán los diferentes hilos.
- **Hilo_Terminal.** Clase que implementa al Terminal que accede al Servidor. Simula que acceden 10 personas desde el terminal y por tanto invoca al método `incrementaCuenta()` en cada acceso.
- **Main.** Clase que crea los hilos y los inicia.
 - Se partirá de 0 accesos
 - Se simula que acceden 10 personas desde cada terminal
 - Se simula que se accede desde 4 terminales

Método synchronized

- Se establece como synchronized el método `incrementaCuenta()` que es el que contiene la variable compartida.

```
public class ServidorWeb {
    //clase que simula los accesos a un servidor
    private int cuenta;
    public ServidorWeb() {
        cuenta = 0;
    }
    public synchronized void incrementaCuenta() {
        //método sincronizado (monitor)
        System.out.println("hilo " + Thread.currentThread().getName()
            + "----- Entra en Método");
        //muestra le hilo que entra en el método
        cuenta++;
        System.out.println(cuenta + " accesos");
        //cuenta cada acceso al servidor y muestra el número de accesos
    }
}
```

Método sincronizado

Resultado de ejecución

- Con synchronized

```
Salida - SincronizaMetodo (run)
run:
hilo Thread-0----- Entra en Servidor
1 accesos
hilo Thread-1----- Entra en Servidor
2 accesos
hilo Thread-0----- Entra en Servidor
3 accesos
hilo Thread-1----- Entra en Servidor
4 accesos
hilo Thread-0----- Entra en Servidor
5 accesos
hilo Thread-1----- Entra en Servidor
6 accesos
hilo Thread-1----- Entra en Servidor
7 accesos
hilo Thread-1----- Entra en Servidor
8 accesos
hilo Thread-1----- Entra en Servidor
9 accesos
```

- **Resultado esperado y correcto** (Los accesos se van incrementando de de 1 en 1)

- Sin synchronized

```
Salida - SincronizaMetodo (run)
run:
hilo Thread-0----- Entra en Servidor
hilo Thread-3----- Entra en Servidor
hilo Thread-1----- Entra en Servidor
hilo Thread-2----- Entra en Servidor
3 accesos
2 accesos
1 accesos
hilo Thread-3----- Entra en Servidor
hilo Thread-1----- Entra en Servidor
4 accesos
6 accesos
5 accesos
hilo Thread-0----- Entra en Servidor
hilo Thread-3----- Entra en Servidor
hilo Thread-1----- Entra en Servidor
9 accesos
hilo Thread-2----- Entra en Servidor
hilo Thread-1----- Entra en Servidor
```

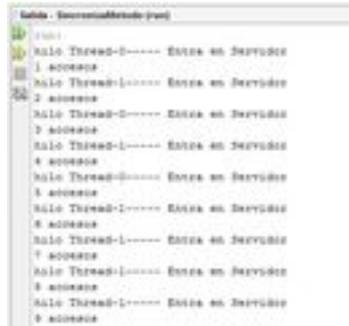
- **Resultado no esperado e incorrecto** (Los accesos no se van incrementando de de 1 en 1)

Credenciales

Imagen	Datos de licencia
	Autoría: Isabel M. Cruz Granados Licencia: Uso educativo-no comercial. Procedencia: Elaboración propia.
Imagen	Datos de licencia
	Autoría: Isabel M. Cruz Granados Licencia: Uso educativo-no comercial. Procedencia: Captura de pantalla del editor de código del programa NetBeans, propiedad Sun Microsystems, bajo licencia GNU GPL v2.

Credenciales (continuación)

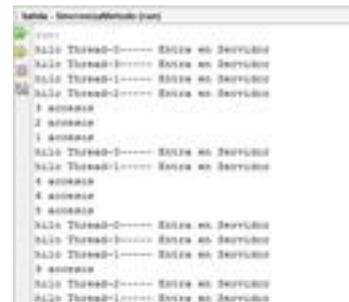
Imagen



Datos de licencia

Autoría: Isabel M. Cruz Granados
Licencia: Uso educativo-no comercial.
Procedencia: Captura de pantalla de la Salida del programa NetBeans, propiedad Sun Microsystems, bajo licencia GNU GPL v2.

Imagen



Datos de licencia

Autoría: Isabel M. Cruz Granados
Licencia: Uso educativo-no comercial.
Procedencia: Captura de pantalla de la Salida del programa NetBeans, propiedad Sun Microsystems, bajo licencia GNU GPL v2.