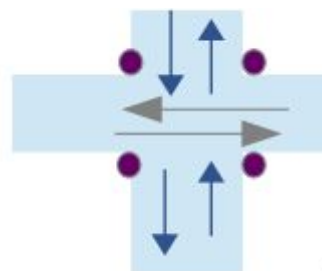
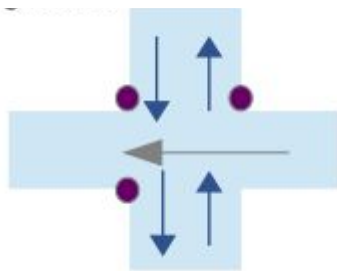




Programación Concurrente 2018 **Trabajo práctico N°2: Concurrencia**

1. Considere la intersección de dos calles muy transitadas. Debido a la alta incidencia de accidentes, se han instalado luces de tráfico con flecha de girar a la izquierda. Una configuración **segura** de las señales de luces es una en la cual dos autos no pueden colisionar en la intersección si no han pasado con luz roja. Considerando los 2 escenarios siguientes:



Una de las calles es de una mano (ejemplo Roca) y la otra doble mano (ejemplo Bouquet Roldán)

Las 2 calles son de doble mano, ejemplo (Fava y Gatica, en Nqn)

Resuelva: Caracterice todas las configuraciones de seguridad de señales de verde, rojo y flecha-izquierda-verde iluminado posibles, que garanticen que todos los conductores tendrán una oportunidad de avanzar.

Nota: Suponga que los conductores siempre obedecen las señales a través de semáforos encendidos y que los giros a la izquierda se hacen sólo cuando la correspondiente flecha de giro a la izquierda se ilumina.

2. Diseñar un programa en Java para mostrar las letras A, B y C. Para ello, utilizaremos 3 hilos, pero se quiere limitar la posible salida de caracteres por pantalla para que no se produzca de cualquier forma sino con la secuencia predefinidas: ABBCCC, por ejemplo: ABBCCABBBCC....

Se recomienda el uso de una variable compartida por los hilos que sirva para asignar turnos, de manera que cada hilo imprima su letra cuando sea su turno. De esta forma, cuando se crean los hilos, a cada uno se le asigna un identificador para que puedan comprobar cuando es su turno y que mientras no sea su turno, no haga nada.

3. En este ejemplo vamos a tener 3 clases: una clase Main, una clase CuentaBanco y una clase VerificarCuenta (Hilo).

```
public class Main {  
    public static void main(String[] args) {  
        VerificarCuenta vc = new VerificarCuenta();
```



```
        Thread Luis = new Thread(vc, "Luis");
        Thread Manuel = new Thread(vc, "Manuel");
        Luis.start();
        Manuel.start();
    }
}

public class CuentaBanco {
    private int balance = 50;
    public CuentaBanco() {
    }

    public int getBalance() {
        return balance;
    }

    public void retiroBancario(int retiro) {
        balance = balance - retiro;
    }
}

public class VerificarCuenta implements Runnable {
    private CuentaBanco cb = new CuentaBanco();

    private void HacerRetiro(int cantidad) throws InterruptedException {

        if (cb.getBalance() >= cantidad)
            System.out.println ( Thread.currentThread().getName() +
                " está realizando un retiro de: " + cantidad + ".");
            Thread.sleep(1000);
            cb.retiroBancario(cantidad);

            System.out.println(Thread.currentThread().getName() +
                ": Retiro realizado.");
            System.out.println(Thread.currentThread().getName() +
                ": Los fondos son de: " +
cb.getBalance());
        } else {
            System.out.println("No hay suficiente dinero en la
                cuenta para realizar el retiro
Sr." + Thread.currentThread().getName());
            System.out.println("Su saldo actual es de " +
                cb.getBalance());
            Thread.sleep(1000);
        }
    } // de hacer retiro

    public void run() {
        for (int i = 0; i <= 3; i++) {
```



```
try {  
    this.HacerRetiro(10);  
    if(cb.getBalance() < 0){  
        System.out.println("La cuenta está sobregirada.");  
    }  
} catch (InterruptedException ex) {  
    Logger.getLogger(VerificarCuenta.class.getName()).  
        log(Level.SEVERE, null, ex); }  
}  
}
```

Ejecute el código y comente el resultado. ¿Qué corrección debería realizar para mejorar la protección de los datos?.

4. Realice un programa en donde se muestre el comportamiento de dos hilos que acceden a un recurso compartido VALOR que se inicializa con 3, de la siguiente manera:

- La Tarea A (hilo 1) realiza TOTAL++.
- La tarea B (hilo 2) realiza TOTAL*=2.

Debe tener en cuenta que las operaciones de cada tarea son: tomar el valor TOTAL, operarlo y volverlo a guardar.

- Pruebe la ejecución varias veces e indique el valor resultante en cada una.
- ¿Qué puede concluir?
- Ahora modifique el programa utilizando alguna herramienta para sincronizar la ejecución.

5. Implemente la clase Auto, subclase de Vehículo con los atributos representativos de auto (patente, modelo, marca, kmFaltantesParaElService, etc). Defina el main en donde se detalle el movimiento del auto una determinada cantidad de km, hasta que llegue a la reserva, para después ir a cargar nafta y continúe andado. Considere ahora que tenemos varios autos funcionando a la vez. Cómo representaría la situación, realice un prueba del programa con 5 autos como mínimo. Considere que sólo se dispone de un surtidor que despacha combustible.

6. Retomar el ejercicio 6 del práctico 1 y modificarlo para considerar lo siguiente:

- de cada producto se tiene además el costo. Ahora la cajera debe calcular el valor total de la compra del cliente
- el supermercado colabora con un comedor infantil, y solicita la colaboración de sus clientes. Se cobra al cliente un 1% de su compra para destinarlo al comedor. Para



ello la donación del cliente es acumulada en el supermercado y entregada al comedor el primer día de cada mes.

7. En una tienda de mascotas están teniendo problemas para tener a todos sus hámster felices. Los hámster comparten una jaula en la que hay un plato con comida, una rueda para hacer ejercicio, y una hamaca en la que pueden descansar. Todos los hámsteres quieren comer del plato, correr en la rueda y luego descansar en la hamaca. Pero se encuentran con el inconveniente de que solo 1 de ellos puede comer del plato, solo uno puede correr en la rueda y solo 1 puede descansar en la hamaca.

a) Implemente un programa para simular la situación planteada, en donde todos los hámster puedan realizar todas las actividades.

Nota: considere que todas las actividades consumen cierto tiempo, por lo que para la simulación se sugiere asignar ese tiempo con “sleep()”

8. Dados los siguientes procesos:

considere que sem1, sem2, sem3, sem4 son semáforos binarios.

<pre>Proceso P1 adquirir(sem2); operaciones_P1; liberar(sem1); end</pre>	<pre>Proceso P2 adquirir(sem3); operaciones_P2; liberar(sem2); end</pre>
<pre>Proceso P3 adquirir(sem1); operaciones_P3; liberar(sem3); liberar(sem4); end</pre>	<pre>Proceso P4 adquirir(sem4); operaciones_P4; end</pre>
<pre>inicializa(sem1,0); inicializa(sem2,1); inicializa(sem3,0); inicializa(sem4,0);</pre>	

1. ¿Qué sucede si el semáforo sem2 se inicializa en 0?
2. ¿Qué sucede si el semáforo sem1 se inicializa en 1?

10. Implementar el pseudo-código correspondiente para lograr la sincronización de tres procesos (P1,P2,P3) de forma que se establezca el orden de ejecución P1, P3 y P2. Así primero se ejecuta P1 y cuando finaliza P1 se puede ejecutar P3 y cuando



Departamento de Programación
Facultad de Informática
Universidad Nacional del Comahue



finaliza P3 se puede ejecutar P2 y cuando finaliza P2 se puede ejecutar P1 y así sucesivamente.