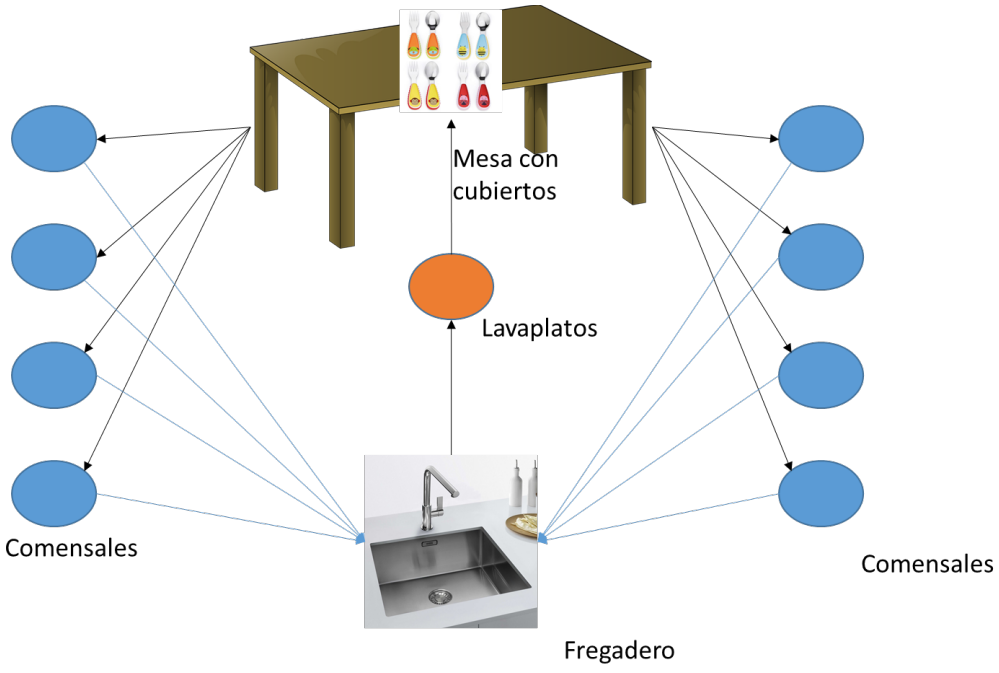
Universidad de los Andes
Departamento de Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional

Caso 1
Manejo de la Concurrencia

Queremos implementar una variación al problema de los filósofos.:

- En esta versión, los cubiertos están en el centro de la mesa y son de dos tipos (tenedor y cuchillo, por ejemplo). Tenemos `numComensales` en la mesa y para poder comer, un comensal necesita un cubierto de cada tipo.
- En la mesa hay `numCubiertosT1` cubiertos de tipo 1 y `numCubiertosT2` cubiertos de tipo 2. Al principio están todos en el centro de la mesa.
- La cena consiste de `numPlatos` platos. Después de cada plato los cubiertos deben ser cambiados. Para evitar que alguien termine mucho antes que los otros, a la mitad de la cena, todos se esperan entre ellos.
- Para cambiar los cubiertos el comensal debe pasarlos, a través de un fregadero de tamaño `tamFregadero`, a un lavaplatos de cubiertos quien los lavará y los volverá a dejar en el centro de la mesa. Una vez el comensal ha entregado los cubiertos en el fregadero, podrá intentar tomar nuevos cubiertos del centro de la mesa.



Objetivo

Diseñar un mecanismo de comunicación para implementar la arquitectura descrita. Para este caso, los comensales y el lavaplatos serán *threads* en la misma máquina (en realidad debería ser un sistema distribuido; este es solo un prototipo).

El proyecto debe ser realizado en java, usando *threads*. Para la sincronización solo pueden usar directamente las funcionalidades básicas de Java: `synchronized, wait, notify, notifyAll, sleep, yield, join` y las `CyclicBarrier`.

Funcionamiento

Cada *thread* comensal implementa un ciclo para comer todos sus platos atendiendo las reglas descritas anteriormente: necesita 2 cubiertos para comer, cambiar los cubiertos después de cada plato y, por razones de etiqueta, debe esperar a que todos lleguen a la mitad del número de platos antes de avanzar.

El *thread* lavaplatos es un loop infinito que recoge cubiertos del fregadero, los lava (una tarea que dura un valor aleatorio de 1 y 2 segundos) y los vuelve a poner en la mesa.

Las reglas de acceso a los cubiertos son las siguientes:

- Solo se puede tomar un cubierto a la vez. Si no puede tomar el primero, debe esperar (espera pasiva) a que haya un cubierto libre de ese tipo.
- Si después de tener un cubierto no puedo acceder al segundo, el comensal debe liberar el que ya tiene y esperar (espera pasiva) a que se libere un cubierto del tipo que no pudo tomar para volver a intentar tomar los dos cubiertos.
- El fregadero para pasar los cubiertos para lavarlos funciona en espera semi-activa (yield), tanto para los comensales como para el lavaplatos.
- En la mesa siempre hay espacio para nuevos cubiertos (una vez lavados).

Comer un plato toma un tiempo aleatorio entre 3 y 5 segundos. Cada comensal debe tardar por lo menos ese tiempo antes de pasar al siguiente plato. Para esto, el thread del comensal genera un número entero aleatorio entre 1 y 3 antes de comenzar a comer un nuevo plato.

El número de platos, cubiertos de cada tipo y tamaño del fregadero para comunicar con el lavaplatos deben estar en un archivo, el cual debe ser procesado por el `main` del programa.

Se debe definir la manera de evidenciar que el programa funciona correctamente. La salida debe ser clara para que se puede validar fácilmente la correcta operación del sistema.

La configuración para la ejecución debe estar almacenada en un archivo de propiedades. Un ejemplo de dicho archivo sería:

```
concurrencia.numComensales = 6
concurrencia.numCubiertosT1= 7
concurrencia.numCubiertosT2= 10
concurrencia.numPlatos = 8
concurrencia.tamFregadero = 5
```

Condiciones de entrega

- En un archivo .zip entregar el código fuente del programa, y un documento Word explicando el diseño y funcionamiento del programa, así como la validación realizada. En particular, para cada pareja de objetos que interactúan, explique cómo se realiza la sincronización, así como el funcionamiento global del sistema. El nombre del archivo debe ser: caso1_*login1*_*login2*.zip
- El trabajo se realiza en grupos de máximo **2** personas de la misma sección. No debe haber consultas entre grupos.
- El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes).

- Se puede solicitar una sustentación a cualquier miembro del grupo sobre cualquier parte del trabajo. Dicha sustentación puede afectar la nota de todos los miembros.
- El proyecto debe ser entregado por BloqueNeón por uno solo de los integrantes del grupo. **Al comienzo del documento Word, deben estar los nombres y carnés de los integrantes del grupo**. Si un integrante no aparece en el documento entregado, el grupo podrá informarlo posteriormente, sin embargo, habrá una penalización: la calificación asignada será distribuida (dividida de forma equitativa) entre los integrantes del grupo.
- **Se debe entregar por BNe a más tardar el martes 07 de septiembre a las 23:55 (p.m.)**