

## Taller 2 - Sincronización de Threads - Exclusión mutua

El propósito de este taller es entender cómo funciona la exclusión mutua y realizar algunos ejemplos prácticos. De manera específica, en este taller vamos a realizar la búsqueda del valor máximo global en una matriz, así como una simulación de una pasarela en la cual las personas pueden caminar en un solo sentido. Con ello en mente tenemos:

### Ejercicio 1 – Búsqueda concurrente del valor máximo en una matriz

Realice un programa en Java tal que dada una matriz numérica busque de manera concurrente el valor máximo presente en ella. Para ello, guíese con el diseño de la solución presente en las diapositivas de clase que incluye las clases **Máximo, Identificar y T**.

### Ejercicio 2 – La Pasarela

Desarrolle un programa que simule el comportamiento de personas en una pasarela estrecha en la que solo cabe una persona a lo ancho. Las personas pueden circular en cualquiera de las dos direcciones. Por lo tanto, en un momento dado, solo pueden estar caminando en la pasarela personas que van en la misma dirección.

### Ideas y fragmentos de código que se pueden usar en la solución del taller:

Considere una solución con tres clases: **Pasarela.java, Persona.java y Main.java**. Para modelar el comportamiento de las personas en la pasarela, vamos a usar threads. Cada persona desfila sobre la pasarela mediante la secuencia de las siguientes tres actividades: (1) entrar a la pasarela cualquiera de las dos direcciones (de derecha a izquierda o de izquierda a derecha); (2) caminar; y (3) salir.

En la clase Pasarela incluya los tres métodos: **entrar(direccion)**, **caminar()** y **salir()**. Cada persona tiene un identificador numérico y una referencia a la pasarela.

La dirección en la que una persona va a desfilarse sobre la pasarela puede ser un valor entero al azar entre 0 y 1, donde 0 puede representar la dirección de derecha a izquierda y 1 la dirección de izquierda a derecha.

En todo momento se debe llevar la cuenta del número de personas que va caminando sobre la pasarela, tanto de derecha a izquierda como de izquierda a derecha, por lo que se sugiere usar dos contadores. Sin embargo, hay diferentes soluciones para este problema.

Al entrar a la pasarela, si hay personas en la pasarela en dirección contraria, la persona debe esperar (el thread debe esperar). Al ser notificado, puede continuar, es decir, incrementa en 1 la cantidad de personas en la pasarela en esa dirección.

Caminar, puede ser representado como una espera de unos pocos segundos. Salir, se puede representar como el decremento de la cantidad de personas que están caminando en la pasarela y notificando a los demás threads para que puedan reanudar la ejecución.

Nota:

- Observe que las personas *entran, caminan y salen*. Solo que algunas de *ellas quedan esperando en la entrada para poder continuar*.
- Además, *observe que una persona se puede quedar esperando indefinidamente*; si al llegar hay personas caminando en la dirección contraria y siguen llegando personas en esa misma dirección. *Este problema se conoce como inanición*.