

## Práctica 1

El objetivo consiste en implementar un sistema que permita vaciar un contenedor. Este sistema presenta un problema de sección crítica, es importante identificarla y utilizar el algoritmo de Peterson para la solución.

El sistema se compone de dos brazos robots y un contenedor con piezas, compartido por ambos. Cada brazo tiene como propósito tomar una determinada cantidad de piezas del contenedor, de manera que entre los dos brazos robots logren vaciar el contenedor. Los brazos pueden tomar la misma o diferente cantidad de piezas.

En un principio, el contenedor cuenta con 50 piezas y con las siguientes restricciones:

- Un brazo sólo puede tomar una pieza cada vez que accede al contenedor.
- Para evitar colisiones entre los brazos, el acceso al contenedor debe ser en exclusión mutua, es decir, los dos brazos no pueden descargar piezas del contenedor simultáneamente.

Para implementar el sistema se deberán crear tres clases: **Contenedor**, **Brazo** y **Sistema**.

La clase **Contenedor** representa el recurso compartido por los dos brazos y es el objeto del proceso de vaciado. El constructor de la clase debe permitir asignar un identificador único al contenedor y configurar la cantidad de piezas que contiene inicialmente. Además, debe ofrecer como parte de su interfaz el método *descargarUnaPieza()*. Este método deberá decrementar en una unidad la cantidad de piezas existentes en el contenedor.

La clase **Brazo** implementa una abstracción del brazo robot. Cada brazo es un proceso en el sistema concurrente, por lo que, esta clase deberá implementar la interfaz **Runnable**. El constructor de la clase tendrá tres parámetros de entrada: un identificador único asignado al brazo, el número de piezas que deben ser tomadas por el brazo durante su actividad y, finalmente, el contenedor sobre el que debe trabajar. Esta clase únicamente deberá implementar el método *run()*, responsable de tomar del contenedor compartido el número de piezas indicado.

Finalmente, la clase **Sistema** es la aplicación que configura y ejecuta el sistema. Contendrá el método *main()* donde serán declarados los elementos que constituyen el sistema: un contenedor de 50 piezas y dos brazos robots que tendrán como propósito vaciarlo. Inicialmente, cada brazo puede estar configurado para tomar 25 piezas del contenedor, aunque este parámetro puede ser posteriormente modificado. Para comprobar el funcionamiento de los brazos cuando tienen que tomar un número diferente de piezas. ¿Qué podría suceder en este último caso?

Un aspecto importante de la implementación del sistema es poder visualizar y analizar su comportamiento. Con este propósito, cada vez que un brazo toma una pieza del contenedor deberá escribir un mensaje que muestre su identificador único e indique el número de piezas tomadas hasta ese momento.

A continuación se muestra el comienzo de la ejecución:

Brazo A descarga pieza 1 de 50 del contenedor...

Brazo B descarga pieza 1 de 50 del contenedor...

Brazo A descarga pieza 2 de 50 del contenedor...

Brazo B descarga pieza 2 de 50 del contenedor...

Brazo A descarga pieza 3 de 50 del contenedor...