

# Memoria PC: Práctica 3

---

## Miembros del grupo:

Carlos Giraldo  
Ricardo Pragnell

## Ejercicio 1: Versión equitativa:

¿Puede producirse la inanición?

Sí, puede producirse la situación en la que un filósofo hambriento no pueda obtener los tenedores ya que sus vecinos izquierdo y derecho alternativamente vayan adquiriendo con éxito los tenedores. De esta manera el filósofo hambriento podría no llegar a adquirir los dos tenedores necesarios para comer, produciéndose inanición.

Para resolver este problema llevamos a cabo las modificaciones indicadas en el enunciado, añadiendo el nuevo estado STARVING.

Es importante diferenciar cuándo se llama a la función *void prueba(int k, boolean firstTime)* ya que en una situación estamos adquiriendo los tenedores y en la otra liberándolos. Para ello hacemos uso de la variable booleana *firstTime*.

## Ejercicio 2: Versión equitativa y más eficiente:

Para mejorar la eficiencia de la aplicación seguimos la propuesta del enunciado: utilizar un array de objetos *DeNotificación* para despertar y dormir selectivamente a los monitores.

Además aplicamos dos optimizaciones adicionales:

- Notificar a un filósofo vecino únicamente si su estado es *EATING*.
- Emplear un *if* en vez de un *while* al hacer el *wait()* de *takeForks()*.