

# Ejercicio 4 - Sistemas distribuidos

---

Manuel Panichelli LU 072/18

## Enunciado

Una comunidad virtual está formada por más de 4000 estudiantes de Ciencias de la Computación, distribuidos geográficamente por todo el mundo. El único medio con el que cuentan para comunicarse es una red privada, mediante la cual pueden enviarse mensajes punto a punto (cada mensaje solo tiene un receptor y un destinatario).

Este grupo de personas decidió buscar cuanto antes un nuevo servidor que les funcione como repositorio. Todos tienen la tarea de explorar distintas opciones de servidores candidatos. Tres días después, tienen que migrar a un servidor elegido; el objetivo es elegir el que tenga las mejores características.

1. ¿Qué problema de los sistemas distribuidos se evidencia en este caso?
2. ¿Qué ocurre cuando existen dos o más servidores con las mejores características?
3. Proponga un esquema que solucione los problemas de los puntos 1 y 2.

## Resolucion

1. Este es el problema de eleccion de lider, "*una serie de procesos elije a uno como lider para realizar alguna tarea*". En este caso, el lider seria el servidor con las mejores características.

Para esto seria necesario poder compararlas entre si, entonces supongo que se puede establecer una relacion de orden total entre las características. Esto resulta natural, ya que si las personas los iban a comparar entre si, tiene que haber una forma de decidir si uno o es mejor que el otro. Pero puede haber empates.

2. Se generaria un *empate*, el cual habria que romper de alguna forma *deterministica* (i.e sin tirar un random, al menos no sin seed) ya que todos deben obtener el mismo resultado.
3. El esquema/algorithm propuesto es el visto en la teorica de chapa, paso a describirlo a grandes razgos (el resto *leanlo ustedes™*)

### Requisitos previos

- Organizar a los nodos (computadoras) en un anillo (estilo *token ring*) Esto es necesario ya que los nodos solamente pueden mandar mensajes punto a punto. Cada nodo puede comunicarse con el que esta "a su izquierda" (con el que esta conectado)
- Cada nodo mantiene un estado que dice si es el lider o no.
- Supongo que cada nodo tambien tiene un process ID que es unico. Esto se podria generarse tambien con un [UUID](#) o algo así, que a fines practicos seria unico pero en la teoria podria dar colisiones, lo cual seria un problema.

Ademas, se puede establecer una relacion de orden total entre esos PIDs.

### Algoritmo

- Un nodo comienza circulando (enviando al proceso al que esta conectado) las características de su servidor
- Cuando cada nodo recibe las características de un servidor, las compara con las del suyo, y hace circular el mejor de los dos.

En caso de haber dos con las mismas características, se rompe el empate comparando por el PID del nodo. (de esta forma se soluciona el problema del punto 2)

- Cuando los mensajes dieron toda la vuelta del anillo, ya sabemos cual es el servidor con las mejores características.
- Hacemos circular una ultima ronda de mensajes con la eleccion de mejor servidor para que todos sepan.

De esta forma, se puede elegir el mejor servidor de forma distribuida.