

Sistemas Distribuidos

Tarea # 3

Profesora: Erika Rosas

Ayudantes: Catherin Vargas catherin.vargas.13@sansano.usm.cl

Alfredo Silva alfredo.silva.13@sansano.usm.cl

26 de Noviembre de 2018

1. Objetivos

- Implementar y utilizar algoritmos de coordinación en un sistema distribuido.
- Implementar y coordinar diferentes componentes de un sistema.

2. Tarea

La tarea consiste en desarrollar una aplicación distribuida, específicamente emular los registros que se realizan dentro de un hospital con respecto a los pacientes y sus fichas. Para esto se propone la utilización de diferentes máquinas, donde cada una de ellas tendrá distintos procesos que serán doctores, enfermeros y paramédicos.

2.1. Arquitectura

Dentro de todo el sistema debe existir un único archivo con los registros de los pacientes, cada paciente tiene una sección con sus datos personales, piense en esta sección como una ficha de ingreso del paciente, es decir, debe tener como mínimo el nombre, rut, edad, alergias y enfermedades preexistentes; luego debe tener una sección con los procedimientos asignados/completados, otra con los exámenes que deben realizarse/realizados y una última sección con los remedios recetados/suministrados. Dicho archivo debe ser leído por cada una de las maquinas, **cada máquina tendrá sus propios doctores, enfermeros y paramédicos.**

Los doctores, enfermeros, paramédicos con sus datos, serán leídos desde un archivo disponible localmente en cada máquina.

- **Los Doctores:** acceden a las fichas de los pacientes con acceso total de edición, excepto por lo datos personales, que quedan fijos al crearse el archivo. Se debe permitir que sólo un doctor acceda a una ficha a la vez, para lograr esto y evitar que exista conflicto entre dos o más doctores, tendrá que implementar el algoritmo del matón (**The Bully Algorithm**), para poder elegir un coordinador, el que será elegido dadas las características de años de experiencia y años de estudio. Dicho coordinador será el que de los permisos de acceso a los otros doctores, enfermeros y paramédicos que estén solicitando acceso a una ficha.
- **Los Enfermeros:** pueden acceder a las fichas de los pacientes, pero sólo pueden editar las secciones de procedimientos completados y remedios suministrados, además sólo podrán acceder cuando no existan doctores u otros enfermeros accediendo simultáneamente a los registros.
- **Los Paramédicos:** sólo pueden editar la sección de exámenes realizados de cada paciente, y podrán hacerlo únicamente cuando no hayan doctores o enfermeros utilizando la ficha.

Se debe crear 4 archivos log con el registro de quién hizo qué modificación y el *timestamp* de cada evento, uno en cada máquina.

Desde cada máquina se intentan hacer escrituras al archivo, que pueden ser de manera concurrente. Cada vez que esto ocurre debe ocurrir uno de los siguientes procedimientos:

1. se envían los cambios al coordinador actual y este propaga a las réplicas para mantener consistencia completa (consistencia secuencial).
2. cada máquina localmente hace el cambio, y envía (propaga) el append de información hacia los logs de las otras máquinas. Para lograr esto tendrán que hacer algún protocolo descentralizado de exclusión mutua para que los logs mantengan el orden de información que se genera de manera concurrente.

Compare ambos métodos en cuanto a: número de mensajes, latencia y mantención de la consistencia en el sistema.

Debe tener en cuenta que cada doctor, enfermero y paramédico tendrá una lista o cola de las fichas a las que desea acceder, dicha lista se obtiene de un archivo local, por lo que deberá coordinar el orden de los accesos, no deben existir colisiones y todos deben cumplir con la cola de pacientes en el archivo.

A continuación se muestra, a grandes rasgos, lo que el sistema debe hacer:

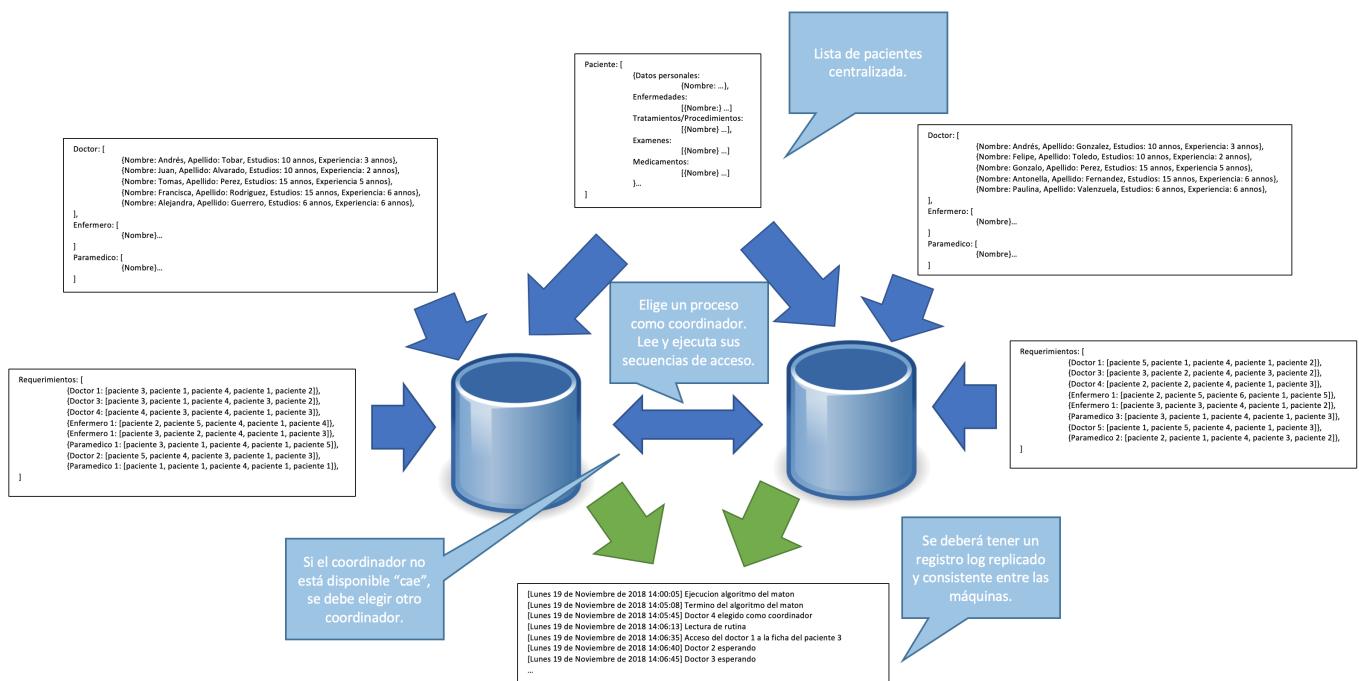


Figura 1: Estructura general de lo solicitado, por temas de espacio sólo se muestran 2 máquinas, considere que para la tarea debe utilizar 4.

2.2. Restricciones

- Cada componente debe estar programado en Java.
- Los archivos que contienen información deben estar en formato JSON.
- El archivo *log* debe tener formato Common Log Format, adaptado a este ejercicio.

3. Consideraciones

- El sistema a desarrollar debe utilizar 4 máquinas virtuales, las cuales será entregadas a cada grupo a través de moodle.
- Cada doctor, enfermero y paramédico es un proceso independiente dentro de cada máquina.

- Se realizará una ayudantía con el fin de presentar y resolver dudas de la tarea.
- Consultas sobre la tarea se deben realizar en moodle o enviar un correo a **catherin.vargas.13@sansano.usm.cl** o **alfredo.silva.13@sansano.usm.cl**

4. Reglas de entrega

- La tarea se realiza en **grupos de 2 personas**.
- La fecha de entrega es el día **16 de Diciembre**.
- Debe entregar todos los archivos fuente necesarios para la correcta ejecución. Recuerde que el código debe estar indentado, comentado, sin warnings y sin errores. Se descontará si no cumple con todo.
- Debe entregar un MAKEFILE o similar para cada uno de los componentes que utilizará.
- Debe entregar un README con nombre y rol de cada integrante, además de la información necesaria para ejecutar los archivos. Se piden instrucciones resumidas, no un informe o testamento.
- Cada hora o fracción de atraso, se penalizará con un descuento de **1 punto**.
- Copias se calificarán con **nota 0**.

5. Rúbrica

2 = Totalmente Logrado (o con mínimos detalles), Muy Bien, Destacable

1 = Parcialmente Logrado. Se aceptan variaciones +/- 0.5 por completitud intermedia

0 = NO logrado o su avance es tan menor que no amerita reconocimiento

Detalle	Porcentaje de ponderación
Doctores y Coordinación	35 %
Ejecución de Enfermeros	15 %
Ejecución de Paramédicos	15 %
Registros Logs (consistentes)	20 %
Comparación de algoritmos	5 %
MAKEFILE y README	5 %
Orden, indentación y comentarios	5 %

- Se aplicará un descuento de 5 puntos al total de la nota por cada Warning, Error o Problema de ejecución.