PROGRAMACIÓN AVANZADA

Práctica de Laboratorio

Convocatoria Extraordinaria – Septiembre 2020

Publicado el 02 de julio de 2020

Simulación del funcionamiento de tres ascensores de un hospital

Parte 1: Programación Concurrente

Se pretende simular el comportamiento de **tres ascensores** existentes en un hospital de 21 plantas (20 plantas más la planta baja), y de las personas que los utilizan.

La implementación del sistema considerará a los ascensores como un recurso compartido que quieren usar concurrentemente varios hilos, que son las personas. Cada ascensor se representará mediante el número de personas que lleva en su interior y su estado: parado (P), bajando (B), subiendo (S), estropeado (E).

El comportamiento del sistema a tener en cuenta es:

- Las personas deben ser modeladas como hilos.
- Cada ascensor deberá ser modelado como un hilo independiente.
- Un ascensor tiene capacidad para albergar a un máximo de 8 personas a la vez.
- Por motivos de ahorro energético, solamente hay dos ascensores funcionando en cada momento.
- Los ascensores que están en funcionamiento están parados mientras nadie los esté utilizando.
- Cada persona es identificada con una “P” más un número (id) único. Ejemplo: P1, P2, P3, etc.
- Las personas se ubicarán inicialmente en las diferentes plantas de manera aleatoria, de tal forma que no todas las personas comiencen en la planta baja. Su destino (la planta a la que se dirigen), también será aleatorio.

- Las personas irán llegando a los ascensores de forma escalonada (no todas a la vez), con un ritmo aleatorio entre ellas de mínimo 0.5 segundos y máximo 2 segundos.
- Cuando una persona quiere utilizar el ascensor debe pulsar el botón del piso en el que se encuentra. Estas pulsaciones quedan registradas hasta que el ascensor las atiende. En cada piso pueden estar varias personas esperando al ascensor.
- En cada planta existe un único botón para llamar a cualquiera de los ascensores que estén en funcionamiento en ese momento.
- Cuando una persona sube al ascensor, selecciona el botón del piso de destino. Estas pulsaciones quedan registradas hasta que el ascensor las atiende.
- Cuando el ascensor llega a un piso, todas las personas con destino en ese piso se bajan; y si hay una persona esperando para subir en ese piso, ésta decidirá subir al ascensor si la dirección del ascensor (subida o bajada) es la misma que debe tomar ella. Una vez dentro del ascensor, pulsará su destino.
- El ascensor sigue la política de no cambiar de sentido cuando ha iniciado el movimiento, hasta que no quedan peticiones que atender en ese sentido, tanto del interior de la cabina como del exterior.
- El ascensor consume 0.5 segundos en subir o bajar un piso.
- Cada cierto tiempo (aleatorio entre 20 y 30 segundos), uno de los ascensores que están funcionando se estropea. En el caso de estropearse uno de los ascensores, el tercer ascensor entrará en funcionamiento. Los ascensores estarán estropeados durante un tiempo aleatorio entre 10 y 15 segundos. Una vez ambos ascensores vuelven a estar operativos, el tercer ascensor se detendrá.
- Cada vez que se estropea el ascensor, las personas que estén dentro se bajan en la planta en la que se ha estropeado y cambian de ascensor (tienen que volver a pulsar el botón de llamada del ascensor). Por simplificar, el ascensor nunca se estropeará entre dos plantas, sino en una planta determinada.
- Cuando un ascensor se estropea, quedará parado en esa planta hasta que esté reparado y vuelva a estar en funcionamiento porque el otro se haya estropeado. Es decir, cuando vuelva a entrar en funcionamiento, lo hará desde la planta en la que se estropeó.

Para seguir la evolución del sistema será preciso que se imprima el estado cada vez que el ascensor cambie de posición. El estado del sistema tendrá la siguiente forma:

Piso	Asc.1	Asc.2	Asc.3	Botón pulsado:	Destinos Asc.1	Destinos Asc.3
20	|	|	|	No	-	-
19	|	|	|	--Sí	-	-
18	|	E	|	No	-	-
17	|	|	|	--Sí	-	-
16	|	|	|	No	-	-
15	|	|	|	No	- P1	-

```
14          |       |       |           No                -           -
13          |       |       |           No                -           -
12          |       |       |           --Sí              -           -
11          |       |       |           No                -           -
10          |       |       |           No                - P2, P3    -
9           |       |      S#0          No                -           -
8           |       |       |           --Sí              -           -
7           |       |       |           No                -           -
6           |       |       |           No                -           -
5           |       |       |           No                -           -
4           |       |       |           --Sí              -           -
3           |       |       |           --Sí              -           -
2           |       |       |           No                -           -
1          B#4      |       |           No                -           -
0           |       |       |           No                - P4        -
```

Esta descripción será **mostrada mediante impresiones por pantalla** con System.out.println(“”). El significado del estado anterior es que el ascensor 1 está en la planta 1, bajando (“B”) con 4 personas dentro (“#4”), que en las plantas 3, 4, 8, 12, 17 y 19 se ha pulsado el botón de llamada al ascensor y que las personas que están en el ascensor 1 quieren ir a las plantas 0, 10 (a ésta dos personas) y 15. Por su parte, el ascensor 2 está estropeado en la planta 18 y el ascensor 3 está parado en la planta 9 sin ningún usuario.

El comportamiento de las personas se generará aleatoriamente mediante las funciones random de Java, y **todo el comportamiento del sistema se guardará en un fichero de texto** (además de mostrarse en pantalla) de forma que sea sencillo analizar lo sucedido. En dicho fichero, entre cada muestra del estado de los ascensores, se incluirán líneas que representen cada acción de pulsación que se ha producido por las personas que llaman al ascensor o que están dentro de los mismos, así como los momentos en los que un ascensor se estropea y la gente que estaba dentro se baja del mismo.

El sistema ejecutará 5000 movimientos de ascensor (en total entre ambos ascensores) y creará 6000 personas **de forma escalonada y aleatoria**, no todas a la vez. Una vez terminados los 5000 movimientos, se mostrará el estado final de los ascensores en funcionamiento (personas en su interior), sus destinos, la situación de cada ascensor (estado y planta en la que se encuentra) y las personas que no fueron atendidas. Esta situación también se grabará en el fichero.

El nombre del fichero será “evoluciónAscensor.txt”.

Parte 2: Programación Distribuida

Basándose en el programa creado en la Parte 1, realizar las modificaciones necesarias para incluir dos nuevos módulos (utilizando programación concurrente distribuida): un **módulo vigilante** y un **módulo evacuación**. Estos dos nuevos módulos tendrán las siguientes características:

Módulo vigilante

Este módulo permitirá consultar de forma remota el estado de los dos ascensores del hospital, mostrando su estado de forma automática con una periodicidad de 1 segundo. La información relativa a la situación de cada ascensor mostrada será: piso en el que se encuentra el ascensor, cuántas personas se encuentran en su interior, así como sus destinos y las personas que se encuentran esperando en cada planta.

El módulo vigilante, al ser un módulo de sólo consulta (no permitirá operaciones de modificación en los ascensores) deberá permitir la conexión de varios clientes a la vez, por lo que se deberá implementar un mecanismo de atención de varias consultas simultáneas.

La interfaz de este módulo, a elección del alumno, puede ser realizada mediante interfaz de consola (System.out.println) o mediante una interfaz gráfica utilizando jFrame de Java.

Módulo evacuación

Este módulo contendrá un único botón, el cuál una vez se pulse permitirá iniciar un proceso de evacuación del hospital de la siguiente manera: dejarán de crearse nuevas personas y, automáticamente, la planta de destino de todas las personas que estén esperando a un ascensor será la planta baja, para abandonar el hospital. Ambos ascensores empezarán a funcionar a la vez, recogiendo a todas las personas de cada planta y llevándolas a la planta baja (con la limitación de máximo 8 personas de capacidad dentro de cada ascensor). Si, debido a las limitaciones de ocupación de cada ascensor, en una única bajada no fuera posible evacuar a todas las personas, se repetirá el proceso hasta que todas estén evacuadas.

Una vez se hayan desalojado todas las personas del hospital, los ascensores se detendrán y se dará por finalizada la ejecución.

Consideraciones generales

Se deben desarrollar un total de 3 programas:

- Un servidor, cuyo código base será el programa desarrollado en la Parte 1, ampliado con la funcionalidad correspondiente para dar soporte al módulo de programación distribuida.
- Desarrollar un programa cliente que realice la consulta a través del módulo vigilante, y permita mostrar por pantalla el estado de los ascensores con los datos recibidos.
- Desarrollar un programa cliente que permita interactuar a través del módulo evacuación (utilizando jFrames de Java), en el que exista un botón con el que se podrá iniciar un proceso de evacuación del hospital con las condiciones descritas anteriormente.

Se podrán utilizar todos los mecanismos vistos en clase para resolver todos los problemas de comunicación y sincronización que se plantean en este enunciado. No obstante, se deben utilizar los mecanismos de sincronización y comunicación que resuelvan el problema **de la forma más eficiente y óptima** posible.

Condiciones de entrega

1. *La práctica se realizará (**opcionalmente**) **por parejas** y deberá ser entregada antes de la fecha indicada en el Aula Virtual, a través de la tarea correspondiente, mediante la subida de dos archivos: la memoria de la práctica en formato PDF o DOC y el proyecto Netbeans completo, comprimido como **ZIP** (no utilizar extensión .rar). No se aceptarán trabajos enviados pasada la fecha límite de entrega.*
2. *No se aceptará ninguna práctica que no contenga un proyecto de NetBeans. La utilización de cualquier otro IDE distinto supondrá la no aceptación de la práctica.*
3. *Si la práctica es realizada por una pareja, **sólo uno de los integrantes deberá subirla** al aula virtual, indicando el nombre de ambos alumnos.*
4. ***La memoria deberá incluir, como anexo, el código fuente** del programa. Si esto no fuera así, la práctica no podrá ser aprobada.*
5. *La entrega fuera del plazo indicado en el Aula Virtual supondrá una reducción en la calificación final, siendo del 25% si se entrega el día siguiente a la fecha límite, o del 50% si se entrega dentro de los dos días siguientes. La entrega más allá de esos dos días no será admitida bajo ninguna circunstancia.*
6. ***Ambas partes** (Parte 1 y Parte 2) de la práctica de laboratorio **se deberán entregar juntas** (es decir, en un único proyecto y una única memoria), ya que la Parte 2 se construye sobre la Parte 1.*
7. *Para aprobar, es condición necesaria que todos los programas funcionen correctamente y de acuerdo a las especificaciones indicadas en los enunciados.*
8. *Para aprobar, se debe desarrollar la solución haciendo uso de buenas prácticas de programación. Por ejemplo, es necesario que todos los nombres de las clases comiencen por una letra mayúscula y todos los nombres de atributos y métodos comiencen por una letra minúscula; los atributos deberán ser privados, y sólo se podrá acceder a ellos mediante métodos getter y setter.*
9. *En la portada de la memoria deberán figurar los datos siguientes:*
    a. ***Grado en Ingeniería [Informática / de Computadores]***
    b. ***Curso 2019/2020 – Convocatoria Extraordinaria***
    c. ***DNI – Apellidos, Nombre***
10. *La memoria explicativa de la práctica realizada deberá incluir, en el orden siguiente: 1) un análisis de alto nivel; 2) diseño general del sistema y de las herramientas de sincronización utilizados; 3) las clases principales que intervienen con su descripción (atributos y métodos); 4) un diagrama de clases que muestren cómo están relacionadas; y 5) el código fuente, como anexo.*

11. *Dicha documentación, exceptuando el código, no deberá extenderse más de 20 páginas. La calidad de la documentación – presentación, estructura, contenido, redacción – será un elemento básico en la evaluación de la práctica.*

12. *Para la defensa de la práctica, si el profesor de laboratorio así lo estimara necesario, deberá presentarse una copia en papel de la memoria, impresa por las dos caras y grapada. Este documento podrá ser utilizado por el estudiante como base para responder a las cuestiones que se le planteen en el ejercicio escrito sobre la realización de la aplicación.*

13. *Para mostrar el funcionamiento de los programas, es conveniente que cada estudiante utilice su propio ordenador portátil, en previsión de posibles problemas al instalarlos en alguno de los ordenadores del laboratorio.*