PROGRAMACIÓN AVANZADA

Prueba de Evaluación de Laboratorio

Práctica Final – Junio 2019

Publicado el 03 de junio 2019

Simulación de un Parque Infantil

Se pretende simular el comportamiento de unos **niños** que juegan en un parque infantil. En el parque hay unos **columpios**, con tres plazas individuales; un **tobogán**, al que suben de uno en uno y un **tiovivo** con cinco asientos. Los niños que entran en el parque están continuamente yendo de un juego a otro, eligiendo al azar a cuál ir cada vez. Cuando un niño llega a los **columpios**, mira a ver si hay alguna plaza libre y, en ese caso, la ocupa y se columpia durante un rato; si no la hay, se queda esperando. En el **tobogán**, los niños se suben, se deslizan y se van. Mientras el tobogán está ocupado, los niños que llegan esperan.

Cuando se cierra el parque, los niños terminan de montar en la última atracción en la que estuviesen montando o haciendo cola para montar, y ya no irán a ninguna más, terminando su actividad.

Un **vigilante,** que estará vigilando a distancia, podrá echar del tobogán a los niños de más de 7 años que se hayan subido al mismo.

Un **supervisor** será el que tendrá encomendadas las tareas de observar y de cerrar el parque.

Queremos crear una aplicación en Java Concurrente que simule el comportamiento del sistema (colas de espera, tiempos, mecanismos de sincronización, etc.) que permita también controlarlo desde otras aplicaciones Java Distribuidas, ejecutándose en el mismo equipo o en equipos diferentes.

Para ello hay que construir un modelo en el que estén representados:

1. Los *niños* (modelados como hilos que hereden de Thread), de los que se crearán 50, numerados del 1 al 50. En todo momento deberá poder conocerse en qué atracción están subidos o si están haciendo cola en ella.
2. El *tobogán*, en el que sólo podrá haber un niño montado en cada instante, y el resto estará esperando a que dicho niño termine.
3. Los *columpios*, en el que podrán estar montados un máximo de 3 niños, y el resto estará esperando a que dejen un asiento libre.
4. El *tiovivo*, en el que se pueden montar 5 niños. El tiovivo sólo se pone en marcha cuando se han subido los cinco niños. Entonces, da vueltas durante 5 segundos y los niños que había montados lo abandonan para volver a empezar el ciclo.

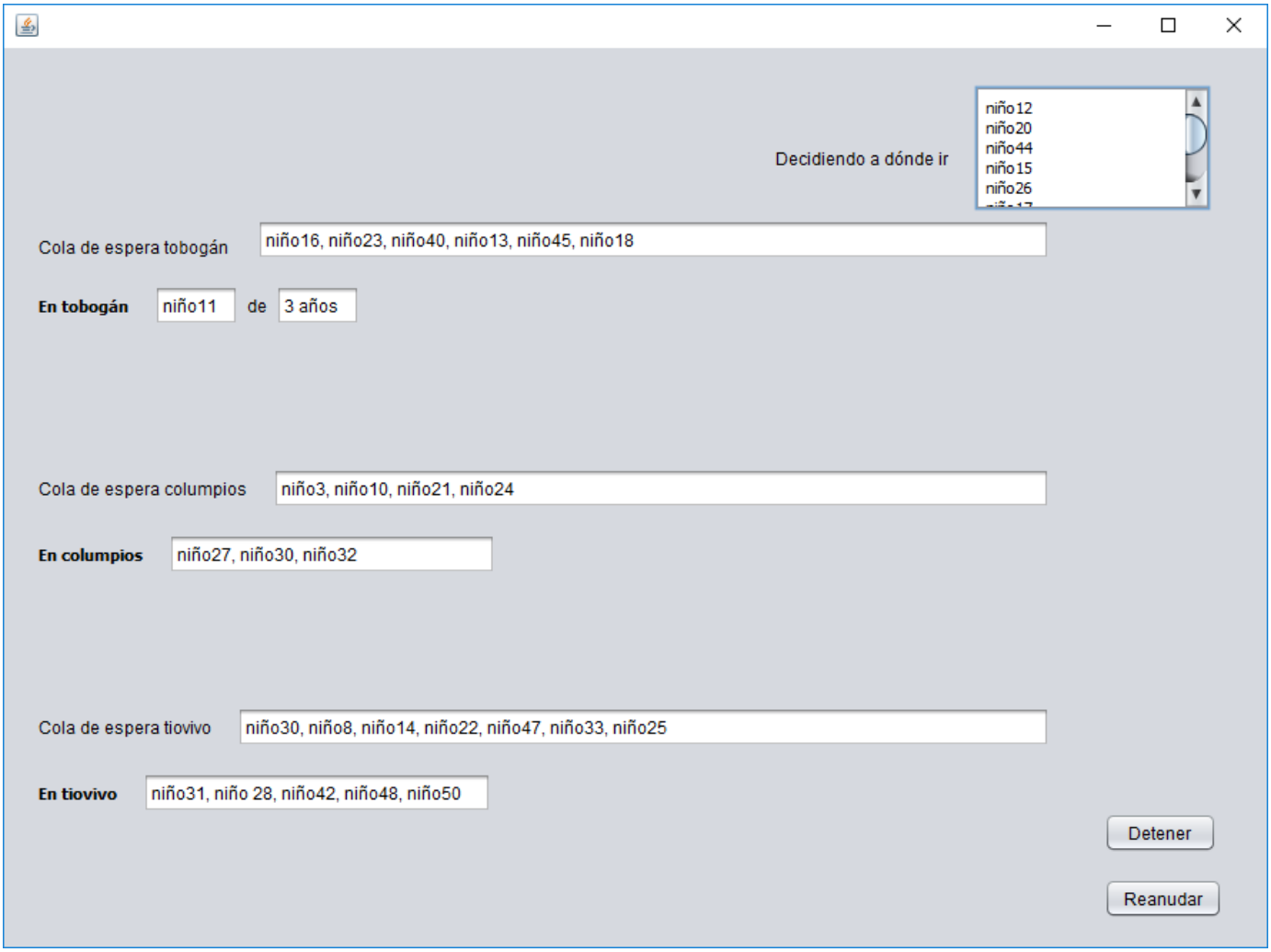
Para cada uno de ellos, la aplicación deberá mostrar dos listas: la de los niños usando la atracción, y la de los niños esperando a hacerlo.

El sistema estará compuesto por tres programas que se comunicarán entre sí mediante sockets y RMI, como se indica a continuación.

El módulo **ParqueInfantil** (“servidor”) deberá tener una interfaz gráfica que permita ver en una única ventana la situación de cada uno de los 50 niños.Este módulo incluirá también dos botones que permitan

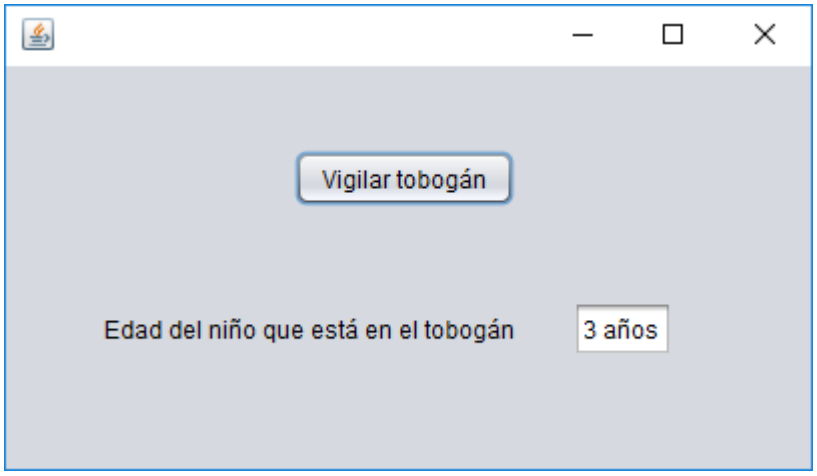
detener y **reanudar** la actividad general, con objeto de poder hacer comprobaciones sobre el funcionamiento del sistema.

De modo orientativo, la interfaz que debería tener el servidor sería parecida a la siguiente:



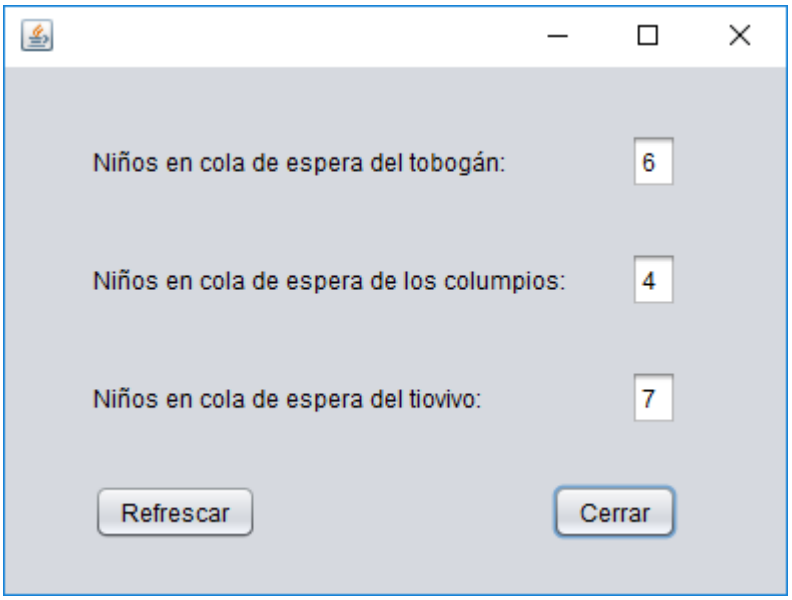
Un módulo **Vigilante** (“cliente”) que también deberá tener una interfaz gráfica, y tendrá un botón para comprobar la edad del niño que se está subiendo en el tobogán e interrumpirle si tiene más de 7 años, obligándole a bajarse sin deslizar, y un cuadro de texto en el que se podrá ver la edad del niño. El módulo **Vigilante** se comunicará con el servidor (**ParqueInfantil**) a través de **RMI**.

Una posible interfaz del vigilante sería la siguiente:



Otro módulo, **SupervisorParque** ("cliente"), será el encargado de visualizar el número de niños que hay en la cola de cada atracción en un momento determinado. Dispondrá de una ventana con tres cuadros de texto en los que se muestren dichos datos, y dos botones: uno "Refrescar", para refrescar la información de los datos; y otro "Cerrar", para cerrar el parque. El módulo **SupervisorParque** se comunicará con el servidor (**ParqueInfantil**) a través de **Sockets**.

En la siguiente imagen se muestra una posible interfaz del módulo SupervisorParque:



Funcionamiento general:

1. El programa principal irá creando cada uno de los 50 niños, asignándoles su identificación y una edad, entre 3 y 12 años, que será igual a 2 + la cifra de las unidades de su número de identificación. Por ejemplo, el niño número 13 tendrá 5 años (2+3), el niño 40 tendrá 2 años (2+0), etc.. La tasa de creación será de un niño cada 0,2 segundos.
2. Cada niño esperará un tiempo aleatorio entre 0,2 y 2,0 segundos para pensar en qué atracción montar, y elegirá una al azar. Se irá a montar en ella, haciendo cola si está ocupada. Cuando termina de montar, vuelve a realizar este mismo ciclo, hasta que se cierre el parque, en cuyo caso, termina el proceso.
3. El tiempo que pasa un niño montando en los columpios será aleatorio, oscilando entre 0,2 y 2,0 segundos. En el tobogán, subir le lleva un tiempo fijo de 1,2 segundos y deslizarse 0,5 segundos.

En el tiovivo, cada niño tiene que esperar el tiempo que haga falta hasta que se completen los cinco puestos. En ese momento se pone en marcha y da vueltas durante 5 segundos, tras los cuales los niños lo abandonan.

4. Cuando el sistema esté **detenido** por haberse pulsado el botón correspondiente, los identificadores de cada uno de los niños se visualizarán en el campo de texto correspondiente a la situación en que se encuentren en ese momento.

Módulo Vigilante:

5. Tendrá a su disposición un método remoto ofrecido, mediante RMI de Java, por el módulo servidor, que devolverá como resultado la edad del niño que ocupa el tobogán y que interrumpirá a éste, si se está subiendo y tiene más de 7 años.

Módulo SupervisorParque:

6. Se comunicará con el servidor mediante un socket en el puerto 2222, por el que enviará la petición de los datos sobre los totales de niños que hay en la cola de cada atracción, o la petición de cierre del parque.

Condiciones de entrega

1. *La práctica se realizará preferiblemente de forma **individual**, aunque **opcionalmente** se podrá entregar **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 se aceptarán trabajos enviados pasada la fecha límite de entrega.*
2. *Si la práctica es realizada por una pareja, **sólo uno de los integrantes deberá subirla** al aula virtual, indicando el nombre de ambos alumnos.*
3. ***La memoria deberá incluir, como anexo, el código fuente** del programa. Si esto no fuera así, la práctica no podrá ser aprobada.*
4. *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.*
5. *Para aprobar, es condición necesaria que todos los programas funcionen correctamente y de acuerdo a las especificaciones indicadas en los enunciados.*
6. *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.*

7. *En la portada de la memoria deberán figurar los datos siguientes:*
    a. ***Grado en Ingeniería _____ (****Informática o de Computadores****)***
    b. ***Curso 2018/2019 – Convocatoria Extraordinaria***
    c. ***DNI – Apellidos, Nombre***
8. *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.*
9. *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.*
10. *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.*
11. *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.*