

Laboratorio de Programación Avanzada
PROGRAMACIÓN CONCURRENTE Y DISTRIBUIDA
Práctica de evaluación Junio 2009

ENUNCIADO.

Publicado el 20 de Mayo de 2009

Ciudad Deportiva

Queremos construir un modelo en el que representar una Ciudad Deportiva en la que se encuentran varias instalaciones con diferentes características (recursos compartidos) a los que acuden deportistas (procesos concurrentes o threads) identificados mediante un número. Los deportistas llegan a la Ciudad Deportiva y utilizan todas y cada una de sus instalaciones. Cuando lo han hecho, abandonan el complejo (y termina la ejecución del hilo).

El sistema deberá estar dividido en dos partes distribuidas comunicadas entre sí por RMI.

El módulo **CiudadDeportiva** (“servidor”) deberá tener una interfaz gráfica que permita ver en una única ventana el estado de todas sus instalaciones y la situación de cada uno de los deportistas. Incluirá también dos botones que permitan **detener** y **reanudar** la actividad, con objeto de poder hacer comprobaciones sobre el funcionamiento del sistema.

El módulo **Control** (“cliente”) también deberá tener una interfaz gráfica que permita visualizar los deportistas que están jugando al tenis y los que hacen cola a la entrada de la sala de musculación en un momento determinado. Un botón permitirá actualizar dicha información. También incluirá otros dos botones: uno para **cerrar** la sala de musculación y otro para volverla a **abrir**.

Funcionamiento:

1. El número total de deportistas que acuden a la Ciudad Deportiva será de 49. Cada uno de ellos estará identificado por un número entre el 51 y el 99. Irá llegando uno cada 0,1 segundos.
2. Dentro de la Ciudad Deportiva hay cuatro instalaciones: **PistaTenis**, **SalaMusculacion**, **CampoGolf** y **Piscina**.
3. En la **PistaTenis**, solo puede haber jugando dos personas (una contra otra). Si entra en la pista un deportista y está solo, espera a que llegue otro. Cuando esto ocurre, se ponen a jugar un partido que puede durar entre 0,5 y 1 seg. Mientras están jugando, todos los deportistas que quieran echar un partido de tenis, tendrán que hacer cola a la entrada de la pista. Cuando los que estaban jugando terminen, se abrirá la puerta de la pista y entrarán nuevos deportistas respetándose el orden de llegada a la cola.
4. En la **SalaMusculacion** hay aparatos para 15 personas. Una vez alcanzado su límite de capacidad, los deportistas que quieran entrar deberán esperar su turno. Cada deportista se está en la sala entre 0,8 y 1,5 segundos. Cuando termina, sale y deja el sitio a otro.

5. El **CampoGolf** tiene solamente 4 hoyos y se juega de forma individual. Si llega un deportista y el hoyo 1 está libre, comienza a jugar. Si no, se pone a la cola. Cuando el deportista termina de hacerse un hoyo, mira a ver si el siguiente ya está libre. En caso afirmativo se pone a jugarlo abandonando el hoyo ya jugado. Si en el hoyo siguiente todavía hay un jugador, se espera en el hoyo recién hecho sin dejarlo libre. Hacerse un hoyo dura entre 0,4 y 0,6 segundos (mas lo que haya que esperar para pasar al hoyo siguiente). Cuando se termina el hoyo 4, se abandona el campo de golf. En la pantalla tenemos que poder ver qué deportista está jugando cada hoyo.
6. En la **Piscina**, los deportistas pueden estar hasta 5 segundos y no hay limitación en su capacidad. Cuando los deportistas llegan a la Ciudad Deportiva, van directos a la piscina. Luego siguen usando el resto de las instalaciones en el orden que deseen (cada uno elegirá dónde ir de forma aleatoria).
7. Cada deportista deberá usar todas y cada una de las instalaciones de la Ciudad Deportiva una sola vez.
8. Al salir de la Ciudad Deportiva, cada deportista escribirá un mensaje en la consola (con `System.out.println`) indicando que ya ha terminado y el número que tenía asignado.

Módulo de Control:

9. Tendrá a su disposición cuatro métodos remotos ofrecidos, mediante RMI de Java, por el módulo servidor **CiudadDeportiva**:
 - Un método **jugandoTenis** para ver qué deportistas están jugando un partido de tenis en un momento determinado.
 - Un método **colaMusculacion** nos devolverá la lista de deportistas que están esperando a entrar en la SalaMusculacion en un momento determinado.
 - Un método **cerrarSalaMusculacion** que hará que cierre sus puertas dicha instalación. Los deportistas que estuviesen esperando en la cola, seguirán esperando a que se vuelva a abrir. A los que ya estuviesen dentro, se les permitirá terminar tranquilamente sus ejercicios antes de marcharse a la calle.
 - Un método **abrirSalaMusculacion** para reanudar la actividad normal de esta instalación.
10. El módulo de control deberá poder arrancarse en un ordenador diferente del que ejecuta el módulo CiudadDeportiva. Ofrecerá una interfaz gráfica con dos campos de texto para representar, en uno, a los dos deportistas que están jugando al tenis y en el otro, la lista de deportistas que están esperando a entrar en la sala de musculación.
11. Además, tendrá tres botones con el texto: “*cerrarSalaMusculacion*”, “*abrirSalaMusculacion*” y “*refrescarDatos*” (para actualizar los datos mostrados en la pantalla).

Condiciones de entrega

1. La práctica se realizará en grupos de 2 personas como máximo.
2. Es condición necesaria para aprobar la práctica, que todos los programas funcionen correctamente y de acuerdo a las especificaciones indicadas en los enunciados.
3. Durante la defensa de la práctica, el profesor podrá solicitar la modificación de cualquier parámetro que interviene en el sistema, como número de hilos, tiempos de permanencia en los locales, etc... Los alumnos que no sepan introducir adecuadamente las modificaciones solicitadas serán evaluados como SUSPENSO, independientemente del contenido y calidad de la práctica entregada.
4. Deberá entregarse un CD-ROM debidamente etiquetado y conteniendo una carpeta - cuyo nombre será el DNI de uno de los integrantes del grupo - en la que deberán incluirse un fichero:
 - **CiudadDeportiva.doc o CiudadDeportiva.pdf**, con la memoria de la práctica, como se describe en el apartado siguiente

y dos carpetas:

- **Fuentes**, con todos los archivos .java
- **Clases**, con los archivos .class.

También se puede incluir una carpeta con el contenido completo del proyecto NetBeans utilizado en el desarrollo de la práctica.

Para mostrar el funcionamiento de los programas, será necesario instalarlos en alguno de los ordenadores del laboratorio o traerlos instalados en un ordenador portátil.

5. Un documento impreso con la memoria explicativa de la práctica (análisis de alto nivel y diseño del sistema) y que contendrá, entre otros, la relación entre clases (jerarquías) con su descripción (atributos y métodos) así como los posibles diagramas que aclaren el análisis y diseño de la aplicación. También deberá contener una descripción de la interfaz implementada y un manual de usuario. El código fuente no se incluirá en la memoria de la práctica
6. En la portada de la memoria, se incluirá:

Ingeniería Técnica en Informática de _____”
Laboratorio de Programación Avanzada. Junio 2009
Práctica de P. Concurrente y Distribuida: CIUDAD DEPORTIVA
DNI – Apellidos, Nombre
DNI – Apellidos, Nombre (*si se hace entre 2*)

7. El día oficial del examen de la asignatura es el lunes 29 de Junio de 2009 y se realizará en el laboratorio NL4. Antes de esa fecha, cada profesor pondrá en la puerta del laboratorio, una hoja de reservas de horas para la defensa de la práctica, con objeto de evitar esperas innecesarias.
8. La documentación deberá imprimirse a dos caras y no deberá extenderse más de 40 páginas (20 hojas A4). La calidad de la documentación – presentación, estructura, contenido, redacción – será un elemento básico en la evaluación de la práctica, representando el 30% de la calificación final.
9. En caso de aprobado, la calificación obtenida en esta práctica **se mantendrá exclusivamente hasta la convocatoria de Septiembre de 2009.**