

Laboratorio de Programación Avanzada - Curso 2010-2011

PROGRAMACIÓN CONCURRENTE Y DISTRIBUIDA. Práctica 2

Publicado: 3 de Noviembre de 2010

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. Los que tienen un número de identificación impar son hombres y los que lo tienen par, mujeres. Irán 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 del mismo sexo (una contra otra). Si la pista está vacía, entra la primera persona que llegue y espera dentro de la pista hasta que llegue otra de su mismo sexo. (Cuando haya una persona dentro de la pista, las personas de sexo opuesto se quedarán en la puerta esperando). Cuando se juntan dos, 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. Una vez terminado el partido, abandonarán la pista y se abrirá la puerta para que entre el/la deportista que lleve más tiempo esperando en la puerta y se comienza de nuevo el ciclo.
4. En la **SalaMusculacion** hay 15 aparatos, 9 de ellos para hombres y 6 para mujeres. Una vez alcanzado su límite de capacidad, los/las 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/a** deportista y el hoyo 1 está libre, comienza a jugar. Si no, se pone a la cola. Cuando el/la 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/las deportistas pueden estar hasta 5 segundos y no hay limitación en su capacidad. Cuando los/las 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).

Requisitos generales para la evaluación del Laboratorio de Programación Avanzada.

Curso 2010-2011

Publicado: 3-Noviembre-2010

1. **El examen será el día designado por la Escuela para el examen de Laboratorio de Programación Avanzada de la convocatoria correspondiente.** Constará de dos partes:
 - a. Realización in situ y con tiempo limitado, de un programa Java y de un programa CAML similares a los propuestos como ejercicios en las sesiones del laboratorio.
 - b. Entrega del disco y memoria impresa de las dos aplicaciones prácticas realizada por el estudiante durante el curso.
2. Tanto para la realización de la primera parte del examen, como para la defensa de las prácticas desarrollada, el estudiante deberá venir provisto de un ordenador personal portátil con el software necesario.
3. El mismo día del examen, el profesor citará a cada estudiante, (en esa fecha o en fechas posteriores dependiendo del número de presentados) para la presentación y defensa de las prácticas entregadas.
4. La realización de las aplicaciones a presentar como práctica será individual. Cada estudiante deberá hacer las dos prácticas que le hayan sido asignadas, de forma individualizada, en el mes de Noviembre. Las dos prácticas asignadas serán las mismas para las tres convocatorias (dos ordinarias y una extraordinaria) del curso 2010-2011 exclusivamente.
5. La aplicación Funcional deberá hacerse utilizando los lenguajes CAML o F#.
6. La aplicación Concurrente/Distribuida deberá realizarse utilizando NetBeans y tendrá una interfaz gráfica. Estará escrita en lenguaje Java, compatible con Java SE y no deberán utilizarse clases o métodos obsoletos (“deprecated”).
7. La comunicación Cliente/Servidor de los programas de la aplicación distribuida se hará mediante RMI.
8. Con el objetivo de verificar la autoría de las prácticas, el profesor solicitará al diseñador la implementación de código no recogido en el enunciado o la modificación de algún aspecto de la aplicación. El estudiante que no sepa introducir adecuadamente las modificaciones solicitadas será evaluado como SUSPENSO, independientemente del contenido y calidad de la práctica entregada.
9. La aplicación deberá presentar una interfaz gráfica de usuario, intuitiva y fácil de utilizar por los distintos usuarios de la misma.
10. Los programas fuente deberán ser fáciles de mantener por cualquier programador experto en Java. Deberán estar bien estructurados y contendrán los comentarios necesarios para un fácil seguimiento del código. El sistema debe ser robusto frente a fallos y no “abortar” por circunstancias imprevistas.
11. La falta de alguno de los elementos entregables, supone la imposibilidad de presentarse al examen.

Entregables

En papel

Cada estudiante deberá confeccionar una memoria de cada aplicación desarrollada. La documentación deberá redactarse para que pueda ser leída por un hipotético técnico informático que tuviese que evaluar la aplicación y recomendar a una empresa que la adquiriera o no. Deberá imprimirse a doble cara y no ocupar más de 40 páginas.

La memoria de Programación FUNCIONAL deberá incluir:

- **Portada** En la portada de la memoria, se incluirá:
Ingeniería Técnica en Informática de _____”
Laboratorio de Programación Avanzada. Curso 2010/11
Práctica Funcional N: _____
Autor: DNI – Apellidos, Nombre
- **Análisis de alto nivel.** Para cada uno de los apartados de que consta la práctica, se describa el análisis de alto nivel efectuado sobre el problema a resolver, todas las funciones auxiliares que se vayan definiendo y la función principal. En todos los casos, se incluirá el código CAML de cada función y el tipo resultante. De cada función se explicará su funcionamiento y además, cuando se considere conveniente - para una mejor comprensión del funcionamiento de alguna función en particular, - se incluirá algún ejemplo de la aplicación de esa función sobre algún argumento interesante.

La memoria de Programación CONCURRENTE y DISTRIBUIDA deberá incluir:

- **Portada** En la portada de la memoria, se incluirá:
Ingeniería Técnica en Informática de _____”
Laboratorio de Programación Avanzada. Curso 2010/11
Práctica Concurrente N: _____
Autor: DNI – Apellidos, Nombre
- **Análisis de alto nivel.** Una explicación general de cómo se ha desarrollado la aplicación y qué elementos principales intervienen en la misma. Para realizarlo se construye el Modelo de Objetos Conceptual o Modelo de Análisis mediante un *diagrama de clases sencillo*, identificando las clases de negocio. No se imprimirá código fuente en Java.
- **Manual de usuario.** En este apartado se explicará como interactúa un usuario con la aplicación. Es conveniente capturar las pantallas de la aplicación (se hace con las teclas Alt + Impr Pant y luego se pegan en el texto) y hacer un seguimiento de las distintas opciones.

La calidad de la documentación – presentación, estructura, contenido – será un elemento básico en la evaluación de la práctica, representando el 30% de la calificación final.

En CD o DVD

Acompañando a la memoria del proyecto, se incluirá un CD o DVD – libre de virus y de errores –, etiquetado con los nombres de los autores y que contendrá:

- Ficheros Word, RTF o PDF, con las dos memorias impresas.
- Los ficheros .ML (con el programa CAML) y .EXE (con el ejecutable).
- El proyecto en NetBeans completo de la aplicación concurrente (guardando la carpeta del proyecto es suficiente).
- Los archivos necesarios para arrancar el sistema simulando un estado no inicial de funcionamiento, que permita comprobar que se cumplen las especificaciones formuladas.