Legajo: …… Nombre y Apellido: ……..

Programación de Objetos Distribuidos
Parcialito - RMI - Callbacks.

Se desea implementar un **servicio de subastas** utilizando servicios asincrónicos.

Para este servicio una subasta es un objeto que contiene:

- Un título: que funciona como identificador de la misma
- Una oferta mínima (**MinimoEsperado**): este es el mínimo valor que se cree que vale el objeto de la subasta.
- Una oferta esperada (**ValorEsperado**): este el valor que se cree que vale el objeto de la subasta. La subasta del objeto sigue abierta hasta que se logre igualar o superar este valor. Debe ser superior a la oferta mínima

Así una objeto subasta podría ser creado con los siguientes datos: “1-2-3-Minions”, 100, 150. lo cual indica que no se acepta menos de 100$ y se cierra la subasta cuando se alcance o supere los 150$.

Una subasta puede encontrarse en 2 estados:

1.1) Abierta: todavía no hubo ofertas de valores iguales o superiores a **ValorEsperado**

1.2) Cerrada: lo opuesto a la abierta. Si se intenta seguir ofertando sobre una subasta cerrada, se considera un error.

2) Una oferta puede tener los siguientes resultados:

2.1) Ganadora: Si la oferta es la primera que supera el **ValorEsperado.**

2.2) Aceptada: cuando la subasta estaba abierta, el valor de la oferta no sea inferior a **MinimoEsperado** y no haya otra oferta anterior por arriba de este valor.

2.3) Rechazada: cuando la subasta está abierta pero el valor de la oferta o bien no supera la oferta candidata hasta ahora mejor (si es que existe) o tiene un valor inferior al valor **MinimoEsperado**.

2.4) Error: cuando la oferta no se podía hacer (subasta cerrada o inexistente).

Legajo: …… Nombre y Apellido: ……..

El servicio de subasta se ofrece a partir de la siguiente interfaz

- ***SubastaService***:
    - ***listarSubastas***: retorna el listado de los títulos de las subastas abiertas (ver item 1.1)
    - ***ofertar***: Recibe la oferta para una subasta y un handle del cliente que oferto y de acuerdo al resultado de la oferta:
        - Si es ganadora (ver item 2.1): se guarda monto y handle y se notifica a quien ofertó mediante ***notificarSubastaGanada***. Si había una oferta aceptada previa le notifica mediante ***notificarOfertaSuperada.***
        - Si la oferta es aceptada (ver item 2.2) se guarda monto y handle y se notifica a quien ofertó mediante ***notificarOfertaAceptada***. Si había una oferta aceptada previa le notifica mediante ***notificarOfertaSuperada.***
        - Si la oferta es rechazada (ver item 2.3) se realiza la notificación ***notificarOfertaRechazada****,* a quien corresponda***.***
        - En casos de error (ver item 2.4) se notifica al que llama mediante a ***notificarError***.

- ***SubastaClientHandle***: es la interfaz remota del cliente para recibir notificaciones de las operaciones del servidor. Ofrece:
    - ***notificarOfertaSuperada***: Para un cliente que iba ganando una subasta se le notifica que llegó una oferta que superó la suya (indicando el valor de esa nueva oferta).
    - ***notificarOfertaAceptada***: Le indica al cliente que su oferta para la subasta fue aceptada y por ahora va ganando.
    - ***notificarOfertaRechazada***: Le indica al cliente que su oferta para la subasta fue rechazada.
    - ***notificarSubastaGanada***: Le indica al cliente que ha ganado la subasta.
    - ***notificarError***: El servidor notifica los errores al ofertar por medio de este método. Envía un código numérico y un texto explicando el error.

Legajo: …… Nombre y Apellido: ……..

Consideraciones:

- El sistema no tiene usuarios ni autenticación.
- El servicio debe ser implementado utilizando **RMI**, debe correr en un entorno thread safe y usuarios concurrentes. Además de un cliente que pueda ejecutarse en otra computadora física y reciba notificaciones del servidor a través de callbacks.
- Se debe implementar:
    - el servant del servicio según las especificaciones dadas
    - el server que inicie el servicio y publique al servant.
    - Un objeto que implemente SubastaClientHandle imprimiendo en la consola los mensajes recibidos por el servidor.
    - Un cliente se comunique con el server y haga algunos llamados de ejemplo para probar las condiciones del servicio.
- Se pueden agregar otras clases al servidor y al cliente de considerarlo necesario.
- Las interfaces del servicio y del client handler **deben tener javadoc completo y cualquier otro comentario en el resto de las clases que crean pertinente es bienvenido.**
- **No hace falta persistir el estado de las subastas de una publicación a otra del servicio.**
- **El servidor se inicializa con un número fijo de subastas y no se pueden agregar ni borrar subastas.**

Condiciones del exámen

El exámen se realiza en grupo de hasta 2 alumnos y el tiempo para la entrega es de 1 hora y 30 minutos.

Se entrega al equipo:

Bajar de Sakai – Material Didactico – Ejercicios- Clase05-parcialito.tar

El paquete contiene:

- SubastaService.java con la interfaz del servicio.
- SubastaClientHandle.java con la interfaz del handler del cliente

Legajo: …… Nombre y Apellido: ……..

El equipo debe entregar
Un paquete tar con el nombre parcialito-05.tar con el siguiente contenido:

- Un archivo de texto **integrantes.txt** con el nombre, apellido y legajo de los integrantes.
- Un documento llamado **tutorial** explicando:
    - Asunciones y decisiones de diseño realizadas.
    - Otros casos de prueba que se podrían realizar
    - detalle de cuáles son las mínimas clases/interfaces que deben estar del lado del cliente y cuáles las interfaces/clases que deben estar de lado del servidor para que funcione;
    - explicite las terminales de JVM que hay que abrir y cuáles son las variables de entorno que hay que setear y los comando que hay que ejecutar para poner en funcionamiento el server y el cliente, y que ambos se puedan conectar.
- El código fuente del proyecto

Este paquete debe ser subido por uno de los integrantes del equipo a sakai, dentro de la materia en su carpeta personal. Se debe indicar en esta hoja en la carpeta de quien se subió:

……...