

Legajo:..... Nombre Apellido:.....

### Parcialito 31/08/2015

Se desea implementar un servicio remoto para una biblioteca virtual a la cual se le puede pedir prestado y devolver libros mediante la siguiente interfaz:

```
public interface BibliotecaService extends Remote {  
  
    Libro prestarLibro(String isbn) throws RemoteException;  
  
    void devolverLibro(Libro libro) throws RemoteException;  
  
    ArrayList<String> listarCatalogo() throws RemoteException;  
}
```

#### Requisitos Funcionales:

- La biblioteca tiene una cantidad fija libros disponibles cuyo ese stock no cambia mientras el servicio esté levantando.
- Cada libro está identificado por su ISBN (una cadena alfanumérica) debe poseer getters para este valor además del título (String), fecha de publicación (LocalDate) y autor que es un objeto que contiene getters para nombre (String) y apellido (String).
- La biblioteca puede recibir más de un ejemplar del mismo libro para prestar.
- El sistema no tiene usuarios, ni autenticación.
- Los métodos del servicio deben cumplir:
  - *prestarLibro*:
    - Retorna un Libro en caso de que esté disponible para prestar
    - Si el ISBN no es de un Libro válido se lo considera un error.
  - *devolverLibro*:
    - Si el libro fue prestado por el servicio queda inmediatamente disponible para prestarlo de nuevo.
    - En caso de que el libro no sea válido o no fue prestado se lo considera un error.
  - *listarCatalogo*:
    - Devuelve un listado de los libros que existen en la biblioteca con el formato ISBN-TITULO. Para la respuesta no tiene en cuenta si los libros están prestados o si tienen más de un ejemplar.
    - No tiene condiciones de error.

#### Requisitos No Funcionales:

- Se requiere un servicio remoto implementado utilizando **RMI** y debe soportar correr en un entorno con thread safe y usuarios concurrentes. Además de un cliente que pueda ejecutarse en otra computadora física
- La interfaz del servicio **no** puede ser modificada (en métodos y parámetros). Sí se pueden cambiar/agregar tipos de excepciones de considerarlo necesario.
- Se debe implementar el **servant** del servicio, el **server** que inicie el servicio y publique al servant y un **cliente** con invocaciones de ejemplo de varios casos de uso

Legajo:..... Nombre Apellido:.....

- Se pueden agregar clases al servidor y al cliente de considerarlo necesario.
- El diseño del servidor, estructuras de datos y el manejo de errores y demás casos no especificados queda a cargo del programador.
- La interfaz del servicio y las clases creadas deben tener javadoc completo y cualquier otro comentario que el programador crea conveniente es bienvenido.
- No hace falta persistir el estado de libros y préstamos de una publicación a otra del servicio.

## Condiciones del examen

El examen 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- **Clase04-parcialito.tar**

El paquete contiene:

- **BibliotecaService.java** un archivo con la interfaz a utilizar
- **libros.txt**: con un array para utilizar en la inicialización del servicio. **No se debe leer del archivo, simplemente copiar el array en su código.**

### El equipo debe entregar

#### Un paquete tar con el siguiente contenido:

- Un archivo de texto **integrantes.txt** con el nombre, apellido y legajo de los integrantes.
- Un documento llamado **diseño** explicando:
  - Asunciones y decisiones de diseño realizadas.
  - Comentario sobre los casos de prueba.
- Un documento llamado **tutorial** que debe indicar:
  - 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;
  - detalle los pasos para compilar el servicio cliente;
  - detalle los pasos para compilar el servicio ofrecido;
  - explicita las terminales de JVM que hay que abrir y cuáles son las variables de entorno que hay que setear y los comandos que hay que ejecutar para poner en funcionamiento la aplicación servidora.
  - ídem con el cliente.
- El código fuente del proyecto