

SD	Sistemas Distribuidos
13/14	Práctica 2
#2	Sockets y RMI: Sensores y controlador

Preámbulo

El objetivo de esta práctica es afianzar las nociones de comunicación mediante sockets y RMI revisadas durante la práctica guiada anterior de la asignatura y de la arquitectura cliente/servidor estudiada en las sesiones de teoría. También asentar las nociones sobre el funcionamiento de las tecnologías Web básicas y la introducción al concepto de middleware.

Los sockets son la base de las comunicaciones entre computadores y suponen el punto de enlace básico entre nuestra aplicación e Internet, utilizando bastante pocos recursos de red, pero requiriendo la interpretación del mensaje para actuar en consecuencia.

Por otro lado, la utilidad de RMI es similar a los sockets, aunque supone un nivel de abstracción superior que lo hace trabajar a nivel de objeto Java, con las ventajas e inconvenientes que eso conlleva: se transfieren objetos remotos al cliente, aunque los métodos se ejecutan en el servidor y eso hace que la comunicación sea más costosa, pero aumenta la productividad en el desarrollo de aplicaciones distribuidas.

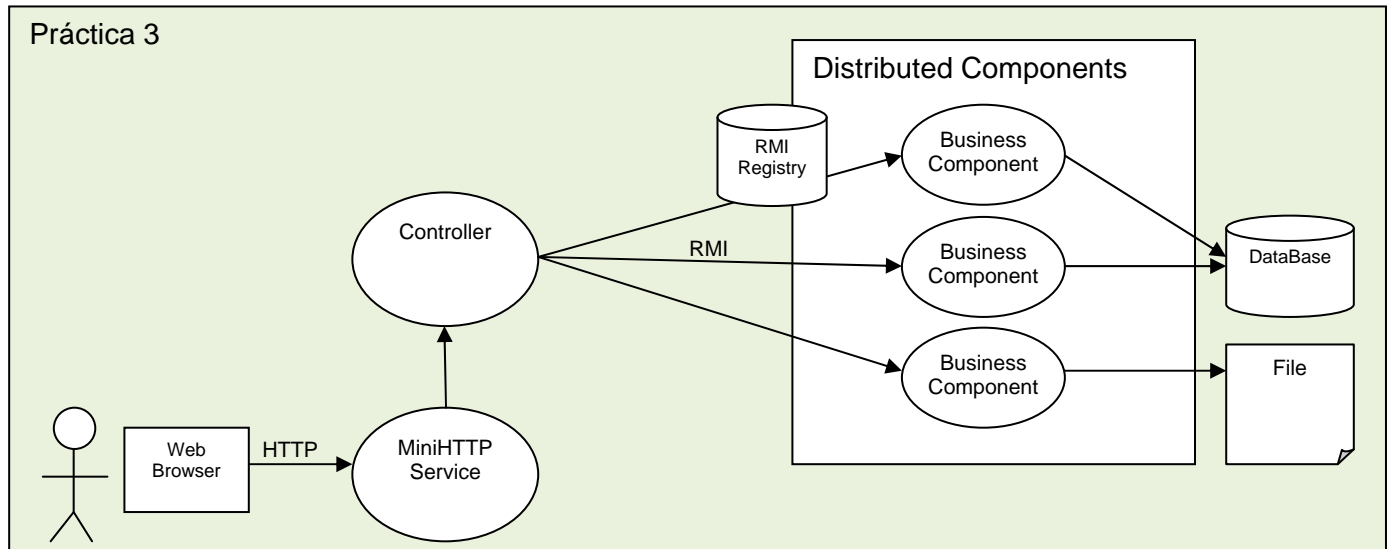
Este módulo de las prácticas pretende establecer el marco inicial de prácticas el cual se irá ampliando progresivamente a lo largo del cuatrimestre. En esta fase inicial se pretende que el alumnos conozcan los principios de la computación y comunicación distribuida como se mencionó anteriormente, planteando un esquema muy básico similar al de las plataformas middleware empresariales más utilizadas hoy en día, como J2EE o .NET.

Especificación

El objetivo de la práctica es el diseño de un sistema distribuido de gestión de invernaderos que habilita la monitorización y el control de diferentes sensores y actuadores a través de internet. Para ello, se pretende simular mediante procesos, un entorno de control de sensores de humedad, temperatura y riego que sea capaz de regular, mediante los correspondientes actuadores, el clima de una serie de invernaderos domotizados.

En la siguiente figura se puede ver la arquitectura general de la práctica donde aparecen los componentes mínimos que debe tener la aplicación:

- Servicio Web MiniHTTP
- Controlador
- Componentes de negocio distribuidos RMI



Servicio MiniHTTP

El acceso al sistema de control de invernadero se realizará a través de una aplicación Web implementada sobre el protocolo MiniHTTP, una versión reducida de HTTP estándar implementada por los alumnos. Se usará cualquier navegador estándar por los que sólo se debe implementar la parte del servidor HTTP, no la parte cliente.

El protocolo, que será implementado mediante el Api de Sockets en Java, debe cumplir las siguientes especificaciones:

Para empezar el servicio debe ser capaz de atender peticiones de manera concurrente, pudiendo parametrizar el número máximo de dichas peticiones en función de los recursos de nuestro servidor.

El servicio aceptará peticiones MiniHTTP y devolverá respuestas MiniHTTP.

Peticiones de entrada MiniHTTP

Como se va a usar un navegador estándar el servidor implementado deberá aceptar sólo las peticiones MiniHTTP. En caso contrario se deberá dar una respuesta de petición no soportada como se mostrará más adelante.

Cada petición MiniHTTP estará compuesta por tres partes como se muestra en la siguiente tabla siguiendo el estándar HTTP pero acotando las opciones:

Línea inicial	Define el método HTTP, la url de acceso al recurso y la versión del protocolo HTTP. Un salto de línea indica su fin y dónde comienza la cabecera del mensaje. Sintaxis: MétodoHTTP+EspacioEnBlanco+URIAccesoRecurso+EspacioEnBlanco+HTTPVersion+SaltoDeLínea Ejemplo: POST /sd/prueba.html HTTP/1.1	Métodos a implementar: GET
Cabecera	Directivas que indican características adicionales de la petición y que pueden influir en el tratamiento de dicha petición por parte del servidor. Cada directiva de cabecera estará separada por un salto de línea. Se debe incluir una línea en blanco antes del cuerpo del mensaje para separarlo.	Tipo de URI: Abs_path

Cuerpo	Información necesaria en algunos casos para realizar la operación definida en la línea inicial.	
--------	---	--

Línea inicial

Sólo se deben aceptar peticiones realizadas mediante el método GET. Aunque existen 4 tipos de URI de acceso a recursos, en el protocolo se implementará la más habitual (absolute path). En ella se indica la ruta absoluta para localizar el recurso. La dirección del servidor se indicará en la directiva de cabecera Host. Esta directiva es usada normalmente cuando el servidor soporta hosts virtuales o permite referenciar al servidor por varios nombres. En este caso, las rutas a los recursos pueden repetirse y deben ser diferenciadas por el nombre del servidor.

En nuestro caso, no será necesario el uso de la directiva Host.

Cabecera

Para simplificar la práctica y la implementación del servidor HTTP no se van a tener en cuenta las cabeceras de las peticiones.

Cuerpo del mensaje

Al implementar únicamente el método GET, este elemento deberá ignorarse.

Respuesta de salida MiniHTTP

La respuesta MiniHTTP está compuesta por tres partes:

Línea de estado	Indica el estado del servidor como respuesta a la petición realizada. Sintaxis: HTTPVerión+EspacioEnBlanco+CódigoDeEstado+EspacioEnBlanco+Descripción Estado+SaltoDeLínea Ejemplo: HTTP1.1 200 OK Los códigos de estados pueden ser consultados en: http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6
Cabecera	Directivas que indican características adicionales de la respuesta y que pueden influir en el tratamiento de dicha respuesta por parte del navegador. Cada directiva de cabecera estará separada por un salto de línea. Se debe incluir una línea en blanco antes del cuerpo del mensaje para separarlo.
Cuerpo	Contenido HTML que será mostrado al cliente a través del navegador.

Línea de estado

Utilizaremos al versión HTTP1.1.

Cuando el servidor reciba durante una petición algún método no soportado deberá devolver un código de estado "405" "Method Not Allowed".

Cabecera

La sintaxis es:

Directiva: valor
Directiva: valor
Directiva: valor

....

A continuación se describen las cabeceras que se deben incluir en la respuesta MiniHTTP.

Connection	Indica el tipo de conexión, si el servidor soporta conexiones persistentes o no. En este caso no se soportarán conexiones persistentes por lo que en todas las respuestas se enviará el valor close.
Content-Lenght	Longitud del contenido del cuerpo del mensaje. El tamaño es indicado en número de bytes contenidos.
Content-Type	Indica el tipo de información enviada en el cuerpo del mensaje y la codificación del mensaje (charset).
Server	Incluye información sobre el servidor.

Cuerpo del mensaje

Contendrá el código HTML que será mostrado por el navegador mostrando la información de los sensores o la descripción de la operación realizada por el actuador.

Ante cualquier duda se deberá seguir la especificación estándar de HTTP.

<http://www.w3.org/Protocols/rfc2616/rfc2616.html>

Controlador

El protocolo HTTP estándar únicamente permite el acceso a recursos estáticos a través de la red. Para acceder a recursos dinámicos, que pueden ser aplicaciones ejecutables, en el lado del servidor es necesaria la introducción de algún elemento que extienda y haga posible la invocación de aplicaciones ejecutables. En nuestro caso lo haremos a través de un elemento denominado controlador.

La función del controlador es dar acceso a los diferentes módulos funcionales que nos ofrecen los invernaderos (sensores y actuadores) y que son encapsulados a través de componentes de negocio distribuidos. Para ello, el controlador será invocado cada vez que la URL del recurso invocado cumpla un patrón previamente definido en el servidor como por ejemplo `http://www.sistemasdistribuidos.com/controladorSD`. Una vez que el servidor Web detecte este patrón, encapsulará los datos de la petición y se los pasará al controlador. Un ejemplo de encapsulación de datos podrías consistir en la siguiente tabla:

Objeto Request	Descripción	Ejemplo
		<code>http://www.sistemasdistribuidos.com/controladorSD/obtenerTemperatura?invernadero=1&sensor=1</code>
ResourcePath	Ruta al recurso a continuación de controladorSD	<code>obtenerTemperatura</code>
Parameters []	Colección de parámetros incluidos en la petición GET	<code>invernadero=1 sensor=1</code>
Headers []	Colección de directivas de cabecera que	

	soporta el servidor	
--	---------------------	--

A partir de la petición recibida, el controlador debe ser capaz de invocar al componente distribuido Java RMI correcto. Para ello deberá realizar una búsqueda en el registro con la información recibida, obtener el *stub* correspondiente e invocar al componente. Posteriormente, el controlador o un componente especializado, debe generar el contenido HTML con la información obtenida que se le pasará al servidor Web para que se lo presente al usuario a través del navegador.

Cada componente distribuido encapsulara la funcionalidad de los sensores y actuadores los cuales son descritos a continuación.

Sensores

Consiste en una serie de procesos que, según su naturaleza (sensor de humedad o sensor de temperatura) envía, ante una petición del modulo controlador, un conjunto de datos conteniendo los siguientes elementos:

- Identificador de sensor (asociado a un invernadero concreto)
- Tipo de sensor (temperatura o humedad)
- Valor de la lectura (humedad en el rango: 0%-100%; temperatura en el rango: -10º a 45º)
- Fecha y Hora de la lectura

Actuadores

Consiste en una serie de procesos que actuarán ante una petición del cliente para mantener los parámetros de humedad y temperatura dentro de los rangos adecuados. Se trata de la simulación de un sistema de climatización que podrá reducir/aumentar los niveles de humedad y temperatura del invernadero donde se encuentren.

- Función (humedad o temperatura)
- Posición (aumento o decremento)
- Tiempo

Habrà que generar las URLs y parámetros necesarios para acceder a dichas funcionalidades desde el controlador.

El operario que monitoriza los invernaderos debe cumplir con una serie de especificaciones de comportamiento:

- La humedad debe mantenerse entre el 40% y 75% en el invernadero correspondiente al sensor. Si baja de 40% se debe activar el goteo hasta que se regularice la situación. Si la humedad sube de 75% se debe activar el deshumidificador de ese invernadero.
- La temperatura debe mantenerse entre 25º C y 35º C en el invernadero correspondiente al sensor. Si baja de 25º C se debe activar el calefactor hasta que se regularice la situación. Si la temperatura sube de 35º C se debe activar el sistema de ventilación y la apertura de ventanas.

Enunciado

Se desarrollará mediante tecnologías sockets y RMI un entorno que cumpla con las especificaciones anteriores utilizando el Api de sockets de Java para la

implementación del servidor MiniHTTP y RMI para la invocación de los componentes de negocio que encapsulan el acceso a los sensores y actuadores.

Se podrán lanzar tantos sensores y actuadores como se quiera, aunque irán asociados a un invernadero concreto y un invernadero no tendrá más de un sensor o actuador del mismo tipo.

El lanzamiento de los sensores y actuadores se realizará desde la consola de comandos (no necesariamente en la misma máquina en la que se encuentran el resto de sensores, actuadores o el controlador), incluyendo como parámetro el número de identificador único del sensor o actuador (elegido por el usuario, se supone que no hay duplicados y no hace falta controlarlo en esta práctica), el valor medido por el sensor puede simularse o establecerse por código en un vector interno fijo, la fecha y hora se obtiene de la propia máquina donde se ejecuta. La frecuencia de sensorización se deja libre (evidentemente no deberá ser ni demasiado alta ni demasiado baja para que la experimentación permita ver los resultados correctamente).

Cuando el módulo controlador recibe la lectura de un sensor, generará el código HTML conteniendo el identificador del sensor, el valor y la hora. Además, si se detecta alguna situación de alerta de las contempladas en la especificación, deberá incluirse un mensaje de advertencia (goteo, calefactor,...).

La realización de la práctica es de carácter individual.