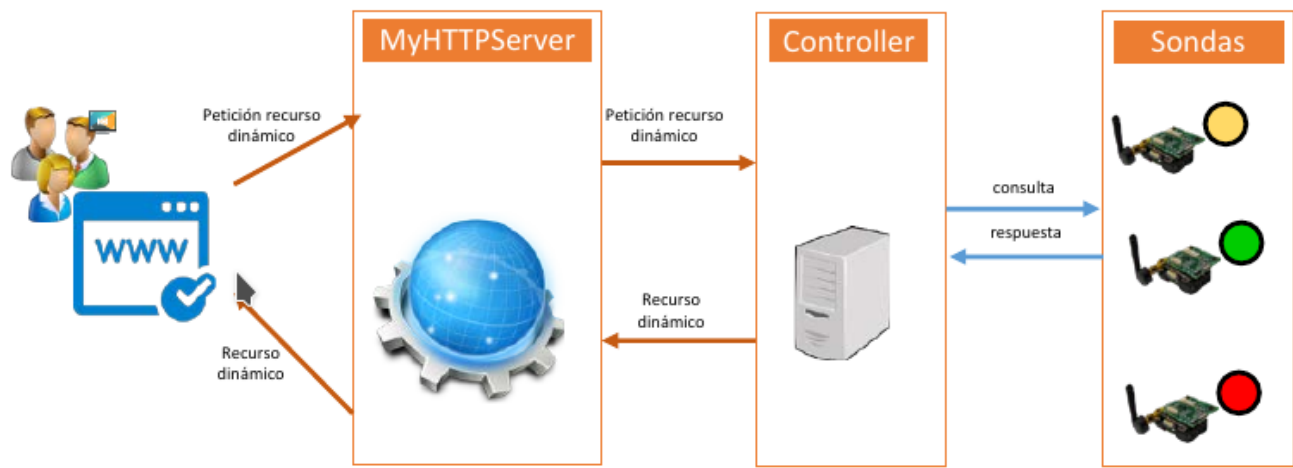


Sistemas Distribuidos

Práctica no guiada

Control de invernaderos Sockets y RMI



Alejandro Jesús Aliaga Hyder
48765284V

Curso 2017 / 2018

Grupo 3 Prácticas
Laboratorio L22
Viernes 13:00 – 15:00

Índice

- 1 Componentes software desarrollados
 - 1.1 Servidor http
 - 1.2 Controlador
 - 1.3 Sondas y Actuadores
- 2 Guía de despliegue
 - 2.1 Ejecución
 - 2.2 Asistentes
- 3 Páginas de la aplicación
 - 3.1 Páginas de inicio
 - 3.2 Páginas de resultados
 - 3.3 Páginas de error

1. Componentes software desarrollados

La aplicación consta de tres componentes software: servidor, controlador, RMI. Cada uno desempeña un papel distinto dentro de la aplicación, a continuación se explicará la función de cada componente y los archivos necesarios para su funcionamiento, entre los que encontramos: políticas de seguridad, archivos html...

1.1 Servidor http

Componente encargado de dar acceso al sistema de información de los invernaderos, se realiza a través de una aplicación web que permite al usuario, utilizando un navegador cualquiera, acceder al sistema y poder ver la información de cualquier sensor del invernadero.

Este componente implementa una versión reducida del protocolo HTTP para que sea posible interactuar a través del navegador.

El servidor es capaz de atender peticiones de forma concurrente, siendo parametrizable. Por tanto cada petición será atendida en un thread distinto.

Para poder llevar a cabo la concurrencia el servidor constará de dos archivos java, uno de ellos se encargará de mostrar un asistente y aceptar las peticiones enviadas por los clientes y el segundo archivo de código fuente de interpretar y responder a la petición, siendo este segundo el que se ejecutará en los distintos threads.

En el asistente consta de unos valores por defecto para cada parámetro, siendo configurables para cualquier ordenador. Los parámetros que encontremos en el asistente son: el puerto de escucha del servidor, el puerto de escucha del controlador, el host del controlador y el número de concurrencias.

El servidor, no genera de forma dinámica la información enviada en las respuestas, esto quiere decir que la información enviada por el servidor se encuentra previamente escrita en archivos html y siendo leídos estos archivos para su envío en la respuesta http. Por tanto la página principal del servidor y sus páginas de errores ya están escritas, cuando sea necesario obtener información de los invernaderos el servidor la recibirá del controlador y luego este se la enviará en cliente.

Los archivos que representan la información estática del servidor son: error404.html, error405.html, error409.html e index.html. Por tanto los recursos del componente servidor son accedidos cuando el usuario introduce una url con el siguiente forma:

`http://HostServidor:puertoServidor/recursoEstático.html`

1.2 Controlador

Para acceder a los recursos dinámicos, es necesario un componente que ejecute y obtenga los datos de las aplicaciones ejecutables, de esto se encarga el controlador.

La función del controlador es dar acceso a los diferentes módulos funcionales que nos ofrecen los dispositivos del invernadero (sonda y actuadores) y que son encapsulados a través de componentes de negocio distribuidos haciendo uso de RMI. Para ello el controlador será invocado por el servidor cuando la URL del recurso invocado cumpla el siguiente patrón:

`http://IPServer:PortServer/controladorSD/...`

Una vez que el servidor Web detecte este patrón, encapsulará los datos de la petición y se los pasará al controlador haciendo uso de sockets.

Además, a causa del uso de RMI para acceder a los objetos remotos, en este caso las sondas de los invernaderos, necesitamos un banco de registros con las sondas disponibles.

El controlador consta de 5 archivos java:

Tres de ellos denominados como Registro, se encargan de registrar un objeto en el banco de registros, para que los objetos remotos sean capaces de registrarse usándolo y lo que nos permite desde el controlador llamar a sus métodos para obtener su información, es decir los valores de los sensores.

Todo esto es necesario ya que debido a temas de seguridad solo se pueden registrar objetos y solo pueden registrar los objetos que se encuentren físicamente en el mismo ordenador que el banco de registro. Por ello le ofrecemos a los objetos que se encuentran en otro lugar un método que si se encuentra en el controller, a través de este se registran los objetos remotos.

Constan de tres archivos java:

-RegistroInterfaz.java: Este es el tipo de archivo que registramos, para que los objetos remotos sepan el método que deben usar para registrarse ellos mismos.

-Registro.java: implementa el método que usan los objetos remotos para registrarse.

-RegistroMain.java: se mantiene a la espera a que una sonda que quiera registrarse e invoca al método del objeto Registro.java para registrarla. Este archivo será el que se ejecute en el controller.

Por último, para poder leer y escribir del banco de registros, necesitamos liberar las opciones de seguridad de RMI. Esta se encuentra en el archivo registrar.policy, este será indicado con un flag en la ejecución de todos los archivos .java que lean, escriban o usen métodos de objetos registrados.

1.3 Sondas y actuadores

Para representar las sondas y actuadores, estas se encuentran en un tercer componente distribuido.

Las sondas encapsulan el comportamiento de un componente electrónico, representadas cada una con un fichero de texto, permitiendo leer y escribir del fichero. Este comportamiento será usado por el controlador para obtener los datos pedidos por el cliente. El formato del fichero es el siguiente:

```
ID=[numeroDelOrdenCreación]
Tipo=[Temperatura o Humedad]
Temperatura=[Valor de la temperatura]
Humedad=[Valor de la humedad]
Fecha=[Fecha creación del archivo]
```

Los actuadores, son representados mediante un archivo de texto al igual que las sondas uno para cada actuador. La encapsulación de los actuadores se encuentra también en el objeto sonda. De esta forma cada sonda siempre tendrá su propio actuador. Estos simulan el sistema de goteo y calefacción de un invernadero. Activándose en función de los valores de temperatura y humedad de las sondas. El fichero tendrá el siguiente formato:

```
ID=[númeroDelOrdenCreación]
Funcion=[humedad o temperatura]
Modo=[aumento o decremento]
```

2. Guía de despliegue

El orden de ejecución de los componentes será el mismo que el indicado en la explicación de la práctica. Aunque es posible variarlo este será el usado en la corrección:

- Iniciar el registro RMI
- Registrar sondas
- Iniciar el controlador
- Iniciar servidor

2.1 Ejecución

- Iniciar el registro RMI:

En una terminal, vamos a la carpeta o paquete donde se encuentren los archivos compilados (.class) y el archivo con la política de seguridad de rmi, ejecutamos:

```
$ rmiregistry [puertoRmi] -J-Djava.security.policy=registrar.policy
```

- Registrar sondas

Ejecutar el archivo que registra el objeto que usarán las sondas para registrarse y también se queda a la espera de nuevas sondas para registrarlas.

```
$ Java -cp . -Djava.rmi.server.hostname=HostController -J-Djava.security.policy=registrar.policy controller.RegistroMain
```

Y ahora empezamos a crear y registrar las sondas.

```
$ Java -cp . -Djava.rmi.server.hostname=HostSondas -J-Djava.security.policy=registrar.policy controller.SondaMain
```

- Iniciar el controlador

```
$ Java -J-Djava.security.policy=registrar.policy controller.Controller
```

- Iniciar servidor

```
$ Java minihttp.Server
```

2.2 Asistentes

A continuación pondré imágenes de los menús tras una ejecución de los componentes

```
*** Asistente RegistroMain ***
```

```
Puerto rmi [pd 1099]:  
registro obtenido  
REGISTRANDO
```

```
|registrado objeto de tipo Registro
```

```
Esperando nuevas sondas a registrar...  
Registered: sonda0  
Registered: sonda1
```

*** Asistente SondaMain ***

Host del controller [pd localhost]: 192.168.56.101
puerto del registro rmi [pd 1099]:

Registrando sonda 0
Tipo de sonda [humedad/temperatura]: humedad
Valor de la humedad]: 2
Intentando registrar sonda0 en el registro remoto

Registrando sonda 1
Tipo de sonda [humedad/temperatura]: temperatura
Valor de la temperatura]: 27
Intentando registrar sonda1 en el registro remoto

Registrando sonda 2
Tipo de sonda [humedad/temperatura]:

*** ASISTENTE CONTROLLER ***

>Puerto del controller [pd 8090]:
>Host minihttpserver [pd localhost]: 192.168.0.19
>Puerto minihttpserver [pd 8080]:
>Host rmi [pd localhost]:
>Port rmi [pd 1099]:

Datos a usar en el controller...
portcontroller: 8090
hosthttp: 192.168.0.19
porthttp: 8080
hostrmi: localhost
porRmi: 1099

>Abriendo controller...

*** ASISTENTE MINIHTTPSERVER ***

>Puerto server http [pd 8080]:
>Concurrencias [pd 6]:
>Puerto controller [pd 8090]:
>Host del controller [pd localhost] :
182.168.56.101

Datos a usar en el server...
puertoServer: 8080
concurrencias: 6
puertoController: 8090
hostController: 182.168.56.101

>Abriendo server...
>Escucho el puerto 8080

3. Páginas de la aplicación

Estas son la URLs que se pueden usar una vez desplegada la práctica:

=> En cuanto al contenido estático

HostServer:PortServer/

HostServer:PortServer/index.html

HostServer:PortServer/estáticoNoDefinido

=> En cuanto al contenido dinámico

HostServer:PortServer/controladorSD/

HostServer:PortServer/controladorSD/ObtenerTemperatura?sonda=NumSondaExistente

HostServer:PortServer/controladorSD/ObtenerTemperatura?sonda=NumNoSondaExistente

HostServer:PortServer/controladorSD/ObtenerTemperatura?sonda=NumSondaHumedad

HostServer:PortServer/controladorSD/ObtenerHumedad?sonda=NumSondaExistente

HostServer:PortServer/controladorSD/ObtenerHumedad?sonda=NumNoSondaExistente

HostServer:PortServer/controladorSD/ObtenerHumedad?sonda=NumSondaTemperatura

HostServer:PortServer/controladorSD/ObtenerAlgoRandom?sonda=NumSonda

HostServer:PortServer/controladorSD/sonda=NumSondaExiste

HostServer:PortServer/controladorSD/sonda=NumSondaNoExiste

3.1 Páginas de inicio



CENTRO DE CONTROL DE INVERNADEROS

Combinaciones url:

host:puerto

- /
- /index.html
- /recursoNoExistente

host:puerto/controladorSD

- /index
- /obtenerTemperatura?sonda=1
- /obtenerHumedad?sonda=1
- /sonda=1

[Página principal \(index.html\)](#)

[Listado sondas \(/controladorSD/\)](#)

3.2 Páginas de resultados



LISTADO DE SONDAS Y SUS LECTURAS

[Inicio](#)

SONDA 0

Tipo=humedad

Fecha=acordarse de hacer la fecha

Humedad=185

SONDA 1

Tipo=humedad

Fecha=acordarse de hacer la fecha

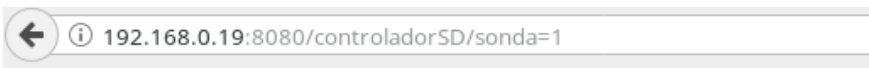
Humedad=43

SONDA 2

Tipo=temperatura

Fecha=acordarse de hacer la fecha

Temperatura=185



Datos del dispositivo sonda 1

SONDA 1

Tipo=humedad

Fecha=acordarse de hacer la fecha

Humedad=39

Temperatura=5



Dato encontrado

Para la sonda sonda2 obtenerTemperatura = 180



Datos del dispositivo sonda2

SONDA 2

Tipo=temperatura

Fecha=acordarse de hacer la fecha

Humedad=0

Temperatura=130

3.3 Páginas de error



Error 409

Error de conexión al controlador

[Inicio](#)



Error 404

Recurso estático inexistente

[Inicio](#)

The screenshot shows the HttpRequester application interface. The address bar displays the URL `192.168.0.19:8080/controladorSD`. The REQUEST section on the left has the URL field set to `http://192.168.0.19:8080`. The method dropdown is set to POST, and the Content Type is `application/json`. The RESPONSE section on the right shows the status `405 Not Found`. A large **Error 405** message is displayed in the response area, with the text `httprequest distinto de GET` and a link [Inicio](#).

192.168.0.19:8080/controladorSD

HttpRequester

REQUEST

URL `http://192.168.0.19:8080`

POST Submit GET POST PUT

New request Paste Request Authentication...

Content to Send Headers Parameters

Content Type: `application/json`

Content Options: Base64 Parameter Body

☒ Content ☐ File Browse...

RESPONSE

POST on `http://192.168.0.19:8080`

Status: 405 Not Found

Error 405

httprequest distinto de GET

[Inicio](#)