

Tópicos Especiales En Telemática

Reto 4: Aplicación distribuida basado en webservices

Presentado por:

John Jairo Tangarife Vélez

Docente:

Edwin Montoya

Universidad EAFIT Mayo del 2013

Contenidos

1Análisis de la aplicación.....	1
1.1Descripción de la aplicación.....	1
1.2Consideraciones iniciales.....	1
2Diseño Global del sistema y arquitectura.....	2
2.1Arquitectura seleccionada.....	2
2.2Diseño Global del sistema.....	3
2.3Descripción de componentes.....	3
3Especificaciones de diseño.....	4
3.1Diseño Detallado.....	4
3.2Descripción de las entidades.....	4

1 Análisis de la aplicación

1.1 Descripción de la aplicación

Calculadora distribuida que permite hacer operaciones remotas como un servicio que se consumen de diferentes servidores que los ponen a disposición. Tiene dos servidores WsCalc2Server que se encarga de las operaciones de restar y dividir y WsCalc1Server que se encarga de las operaciones de multiplicar y sumar.

Se implementa una aplicación cliente llamada RobotCalc que por medio de un archivo xml calcserver.xml hace automáticamente las operaciones descritas por el archivo y en el mismo escribe los resultados.

1.2 Consideraciones iniciales

- Se podrá acceder a los servicios prestados por los servidores WsCalc2Server y WsCalc1Server a través de una página web, haciendo uso de un servlet.
- El servlet leerá de un archivo xml calcserver.xml el URL de los WSDL de los servidores con el siguiente formato:

<urls>

```

        <url>http://localhost:8080/WsCalc1Server/wsCalc1?WSDL</url>
        <url>http://localhost:8080/WsCalc2Server/wsCalc2?WSDL</url>
    </urls>

```

- Existirá una aplicación de escritorio RobotCalc que a partir de una serie de instrucciones en un xml calcserver.xml hace en los servidores las diferentes operaciones y escribe las respuestas en este mismo documento. El documento calcserver.xml tendrá el siguiente formato.

```

<?xml version="1.0" encoding="UTF-8" standalone="no">
<webservices>
    <webservice>
        <url>http://localhost:8080/WsCalc1Server/wsCalc1?
        WSDL</url>
        <method>sumar</method>
        <params>
            <p1>5</p1>
            <p2>20</p2>
        </params>
        <result>25</result>
    </webservice>
    <webservice>
        <url>http://localhost:8080/WsCalc2Server/wsCalc2?WSDL</url>
        <method>restar</method>
        <params>
            <p1>-5</p1>
            <p2>20</p2>
        </params>
        <result>-25</result>
    </webservice>
</webservices>

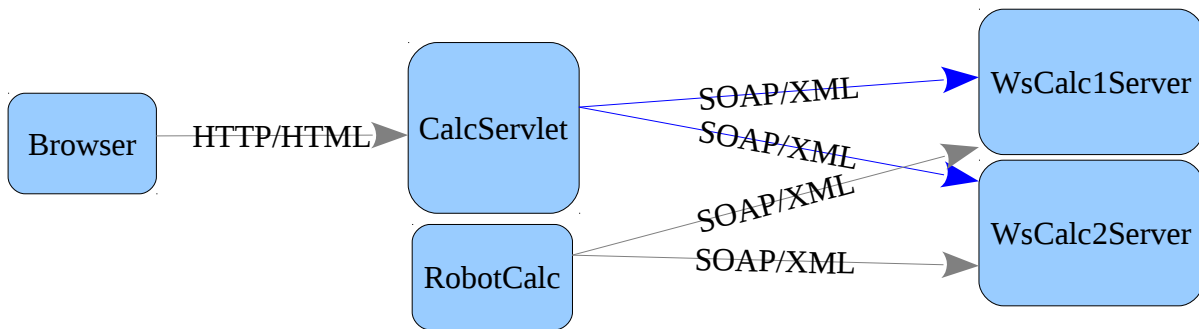
```

2 Diseño Global del sistema y arquitectura

2.1 Arquitectura seleccionada

El sistema tendrá una arquitectura cliente/servidor, con dos tipos de clientes: RobotCalc que es una aplicación de escritorio y Calc.html que es una página web que por medio de un servlet consume los servicios. Habrán también dos tipos de servidores: WsCalc2Server que hará las operaciones de restar y dividir y WsCalc1Server que hará las operaciones de sumar y multiplicar. Estos dos servidores expondrán sus métodos como servicios.

2.2 Diseño Global del sistema

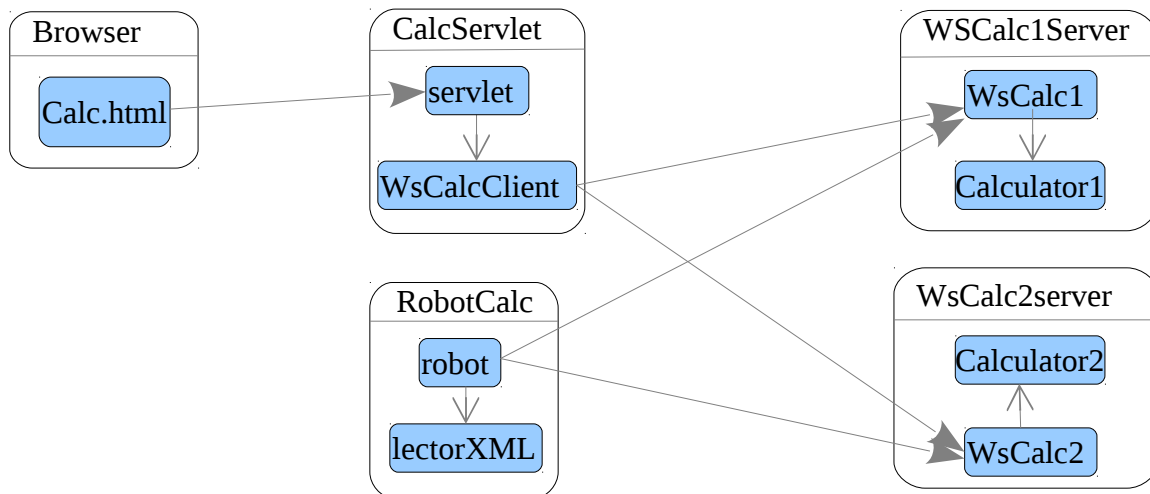


2.3 Descripción de componentes

- Browser: Página web desde donde se accede al servlet CalcServlet. Es la vista o interfaz donde la persona que desee usar los servicios de calculadora ingresa los parámetros.
- CalcServlet: Servlet que permite “consumir” los servicios de calculadora expuestos por WsCalc1server y WsCalc2Server y muestra a la persona el resultado de la operación que solicitó hacer.
- RobotCalc: Es una aplicación de escritorio que recibe un archivo xml con varias operaciones y sus respectivos parámetros, y escribe en este mismo archivo los resultados de estas operaciones.
- WsCalc1Server: Servidor que expone los métodos multiplicar y sumar y realiza internamente estas operaciones.
- WsCalc2Server: Servidor que expone los métodos dividir y restar y realiza internamente estas operaciones.

3 Especificaciones de diseño

3.1 Diseño Detallado



3.2 Descripción de las entidades

- **Calc.html:** Página web que consta de un formulario que por medio del método GET envía al servlet los parámetros y la operación a realizar.
- **servlet:** Entidad que recibe los parámetros del usuario y se encarga de mostrar al usuario el resultado del calculo por medio de una página dinámica. Hace uso del WsCalcClient para enviar las operaciones al servidor y recibir los resultados.
- **WsCalcClient:** entidad que hace las funciones de proxy al permitir la conexión con uno o varios servidores y consumir los servicios que estos ponen a disposición. Tiene los métodos sumar(int, int), restar(int, int), multiplicar(int, int) y dividir(int, int) que son una invocación de estos servicios en los diferentes servidores; de esta manera es transparente para las entidades que lo usan de donde se consume la operación.
- **Robot:** entidad que retoma los datos leídos por el lectorXML, los envía a los servidores para ser procesados y retoma las respuestas de estos procesos para luego enviarlas de nuevo al LectorXML para que escriba los resultados en el documento xml. Tiene los métodos sumar(int, int), restar(int, int), multiplicar(int, int) y dividir(int, int) que son una invocación de estos servicios en los diferentes servidores.
- **LectorXML:** entidad que se encarga de leer los parámetros y operaciones del documento xml y también se encarga de escribir en este mismo documento los resultados de estas operaciones. Tiene los siguientes métodos:
 - traerInstrucciones(rutaXml)
 - guardarResultados(arregloResultados)

- **WsCalc1** esta entidad es un servicio que expone los métodos sumar(int, int) y multiplicar(int, int) utilizando Calculator1 para realizar estas operaciones.
- **WsCalc2** esta entidad es un servicio que expone los métodos restar(int, int) y dividir(int, int) utilizando Calculator2 para realizar estas operaciones.
- **Calculator1:** entidad que realiza las operaciones de sumar dos enteros y multiplicar dos enteros.
- **Calculator2:** entidad que realiza las operaciones de restar dos enteros y dividir dos enteros.