

SD	Sistemas Distribuidos
17/18	Práctica no guiada
	Control de invernaderos Servicios Web

Preámbulo

El objetivo de esta práctica es afianzar las nociones de comunicación mediante la tecnología de Servicios Web revisados durante las prácticas guiadas de la asignatura y del paradigma de computación distribuida SOA estudiada en las sesiones de teoría (temas 1 al 3).

Como se ha indicado en clase, aunque ambos paradigmas han surgido como dos propuestas independientes, hoy en día se conciben como dos caras de la misma moneda:

- SOA desde el punto de vista conceptual y metodológico, es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos, ideado para incluir facilidad y flexibilidad e integración.
- Servicios Web desde el punto de vista tecnológico utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Esta práctica supone la ampliación de la práctica no guiada basada en Sockets y RMI, de forma que junto a la práctica anterior dará como resultado el desarrollo completo de un escenario similar a los que se producen hoy en día en los entornos empresariales, poniendo el foco en el uso e integración de distintos paradigmas de comunicación susceptibles de ser utilizados.

Se proponen dos partes funcionales nuevas:

- La introducción del registro UDDI
- Contemplar aspectos de seguridad

Especificación

En esta práctica se abordará el mismo problema planteado en la práctica de Sockets y RMI pero siguiendo un enfoque B2B frente al B2C que se reflejaba en la práctica anterior. De nuevo vamos a simular que somos una empresa que se dedica al control de plazas de aparcamiento mediante el uso de dispositivos sensores instalados en el techo de las plazas de parking.

Se podrá usar tanto el enfoque top-down como el bottom-up siempre y cuando se cumplan los parámetros de interoperabilidad con otros lenguajes. Uno de los objetivos es la reusabilidad de la propuesta, refiriéndonos en este caso a la reusabilidad del documento WSDL (parte abstracta y enlace).

En este caso, las funcionalidades a los dispositivos de sensorización (sondas) y actuación (actuadores) serán expuestas como Servicios Web que podrán ser accesibles desde una

aplicación consumidora de dichos servicios. Este consumidor será una aplicación de escritorio .NET que accederá a los servicios usando el registro UDDI.

El consumidor en .NET debe implementar las mismas funcionalidades que las realizadas en HTML sobre el navegador en la práctica 3.

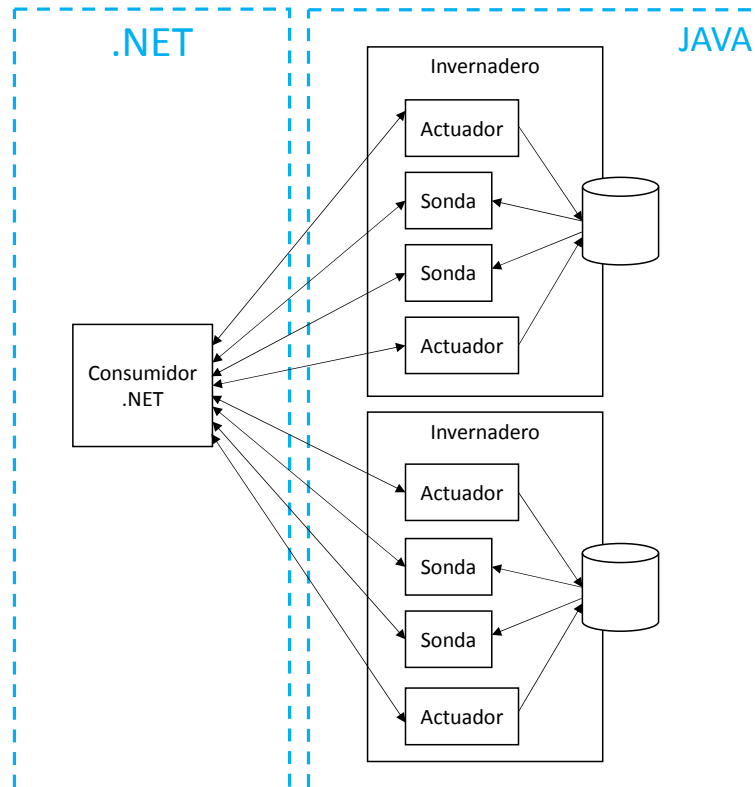


Imagen 1. Esquema conceptual del Sistema software, interconexiones entre los componentes y plataformas.

Si en la práctica 3 veíamos como el sistema a construir requería de la integración de varias tecnologías (RMI y sockets), en esta práctica se hará uso de un único paradigma tecnológico, los Servicios Web, pero esta vez entre diferentes plataformas como son JAVA y .NET, precisamente para demostrar la capacidad de integración de este paradigma.

El objetivo en el desarrollo continúa siendo distribuir la información de las sondas de los invernaderos y modificar el comportamiento de los actuadores para resolver las cuestiones previamente planteadas, cuyas variables son las mismas que en la práctica anterior, con las mismas características.

Como puede verse en la imagen 1, una gran diferencia respecto a la práctica anterior es que, en este escenario, **no hay un elemento controlador que hace de intermediario** entre la aplicación cliente y las sondas de sensorización y los actuadores, sino que el cliente, en este caso la aplicación .NET, accede y consume directamente los servicios de consulta del estado de sensores y servicios sobre actuadores.

Servicios Sonda

Al igual que ocurría en RMI, cada sonda de sensorización o actuador será implementada como un servicio que será puesto a disposición de la aplicación .NET (o cualquier otra aplicación que desee consumirlo). Los Servicios Web no tienen persistencia, por eso es necesario almacenar los valores de cada sonda en algún sistema de persistencia, igual que en la práctica anterior con un pequeño archivo de texto plano es suficiente. De la misma forma los actuadores accederán a esa persistencia para cambiar el valor. En este caso deberemos incluir un servicio que permita acceder a él para distribuir sondas y actuadores.

Se podrán desplegar tantas sondas de sensorización y actuadores como se deseen o necesiten, conteniendo cada una de ellas los servicios necesarios para interactuar con los datos y poder así cumplir con las especificaciones.

Las sondas y actuadores deben venir especificadas mediante un contrato WSDL común (parte abstracta y el binding) y sólo diferenciarse en la parte de la localización física del servicio. Lo mismo sucederá para los actuadores.

Sin embargo, cada sonda y actuador debe ser identificado de forma única para que pueda ser consumido desde el consumidor .NET.

El contrato WSDL debe exponer las mismas funcionalidades que se implementaron en la práctica anterior.

Nota: es importante destacar que las sondas y los actuadores se ofrecen como servicios atómicos, autónomos y desacoplados. Si fuera necesaria lógica de negocio que relacionase operaciones que implicasen a varios sensores y actuadores se podría optar por dos estrategias. Una primera y más acoplada, donde dicha lógica de control se implementaría en el consumidor .NET. Esta solución es menos escalable, reusable y modular, además de dependiente. Otra solución sería encapsular dicha lógica de control en un Servicio Web externo y consumirlo desde la aplicación .NET. Ésta estaría más alineada con un enfoque SOA y de Microservicios. Para la práctica se debe realizar la primera opción.

Aplicación cliente .NET

Esta aplicación actúa como consumidor de los servicios ofrecidos por las sondas y los actuadores de los invernaderos y por tanto deberá implementar la lógica necesaria para ser capaz de acceder a un determinado servicio, consultar una variable o actualizar un valor y mostrarnos el resultado. Es decir, el consumidor debe implementar la misma lógica de cliente generada en la práctica 3 como código HTML pero en uno de los lenguajes de la plataforma .NET.

Es decir, se deben incluir las funcionalidades de monitorización y gestión de los invernaderos incluyendo:

- Configuración de UDDI y endpoints
- Descubrimiento de sondas y actuadores
- Listar sondas y actuadores
- Visualizar estado de los invernaderos
 - o Datos de las sondas

- Modificar parámetros a través de actuadores

Como la parte abstracta y el binding del contrato WSDL de las sondas y los actuadores son comunes entre ellos, debemos tener información de la ubicación física de cada sonda y actuador. Dicha información estará disponible en el registro UDDI, y deberá ser obtenida en tiempo de ejecución a partir de la búsqueda de la sonda o del actuador.

Se debe dar la posibilidad, en caso de fallo del registro UDDI, de introducir las direcciones de acceso (el endpoint) de la sonda y del actuador de forma manual.

La aplicación debe ser capaz de recibir la información necesaria sobre los servicios a los que puede acceder. Todos los servicios a los que es capaz de acceder son de tipo “sonda” o “actuador”, es decir, debe ser capaz de conectarse a 1 o n servicios de este tipo, y consultar cualquiera de sus variables o actualizar el valor de una variable de entorno.

Parte UDDI

La utilización de un registro UDDI permite desacoplar aún más la aplicación consumidora de los servicios, ya que es este registro el responsable de mantener un catálogo de los servicios desplegados y su localización.

Se pide al estudiante que implemente la utilización de un registro UDDI en el sistema a través del servicio disponible en la nube utilizado en la práctica guiada, de forma que todos los servicios sonda y actuadores serán registrados previamente en el registro UDDI, y la aplicación .NET sólo tendrá que acceder al registro UDDI para conocer la ubicación y acceso a los servicios.

La información de acceso a los servicios puede ser registrada usando la interfaz de usuario que proporciona el servicio en la nube o a través de llamadas SOAP mediante SOAPUI.

Es necesario indicar en la aplicación .NET si se opta por el uso de UDDI, para lo cual habrá que poder configurar sus endpoints de acceso a los servicios, o si por el contrario, usamos la forma sin UDDI, donde se introducen los endpoints de los servicios de forma manual.

Las sondas y actuadores primero deben registrarse en el registro UDDI para poder ser accedidas desde la aplicación .NET. La aplicación consultará en el registro UDDI que sondas y actuadores han sido desplegadas y luego ya accederá a cada una de ellas para consumir sus servicios.

Parte Seguridad

Se plantea la necesidad de incorporar mecanismos de seguridad y auditoría. Al tratarse de un escenario B2B, y dado que los servicios web están accesibles para cualquier usuario, se requiere incorporar mecanismos de seguridad en las comunicaciones y control del uso del sistema, de forma que se asegure la privacidad de las comunicaciones entre los objetos y también la auditoría del uso que se da al sistema, desarrollando mecanismos que permitan analizar los accesos realizados, en caso de ser necesario.

Para garantizar la **privacidad** de las comunicaciones: se debe implementar un mecanismo de identificación y autenticación (login y password) y de cifrado de las comunicaciones entre los

servicios web y las aplicaciones-cliente con algún algoritmo criptográfico simétrico o asimétrico (se podrán utilizar librerías existentes o extensiones de SOAP como WS-Security). En lo que respecta a la gestión de usuarios solo se pide que se pueda hacer login en la aplicación .NET, no siendo necesario desarrollar un sistema de gestión de usuarios, (el estudiante podrá dar alta manualmente a una serie de usuarios en su base de datos y solo utilizando el sistema de login de la aplicación con uno de esos usuarios podrá accederse al sistema).

En cuanto a los mecanismos de **auditoría** se pide al estudiante que sea capaz de generar un archivo de “log” en el que se almacenen las acciones que van ocurriendo en cada extremo de la comunicación (es decir, tanto la aplicación .NET como los servicios sondas JAVA deberán tener recopiladas las acciones realizadas sobre el objeto). Este “log” deberá contener al menos los campos siguientes: quién (realiza la operación correspondiente → según login), dónde (desde que ip se solicita la operación), cuándo (tiempo UTC de la operación), qué (operación realizada y sobre qué elemento).

Los mismos aspectos de seguridad deben ser aplicados tanto a las sondas como a los actuadores.

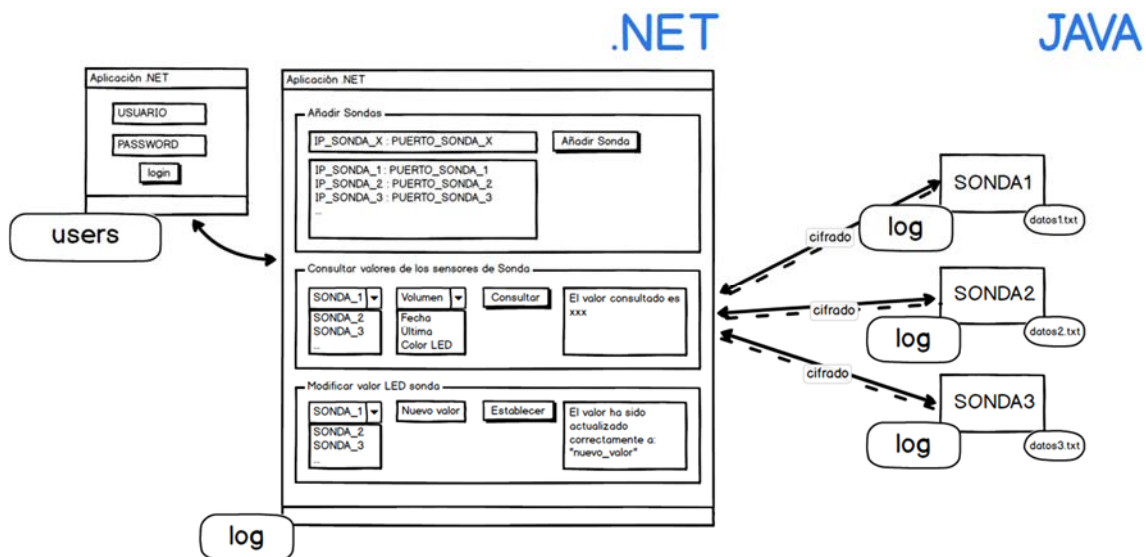


Imagen 1. Esquema conceptual del sistema desplegado utilizando mecanismos de seguridad

Guía mínima de despliegue

Para la correcta evaluación de la práctica es necesario comprobar que la aplicación distribuida solicitada es desplegada (puesta en marcha) en un entorno verdaderamente distribuido. Es por ello que para su prueba es necesario al menos 3 PCs distintos en los que se desplegarán los componentes solicitados (pueden utilizarse de nuevo máquinas virtuales como en la práctica anterior). Se ha de desplegar en uno de los PCs la aplicación .NET, y en los otros dos PCs al menos 1 servicio sonda en cada uno y los correspondientes actuadores. La imagen 2 y 3 muestran el escenario de despliegue para la práctica, teniendo en cuenta que el registro UDDI ya estará desplegado en la nube.

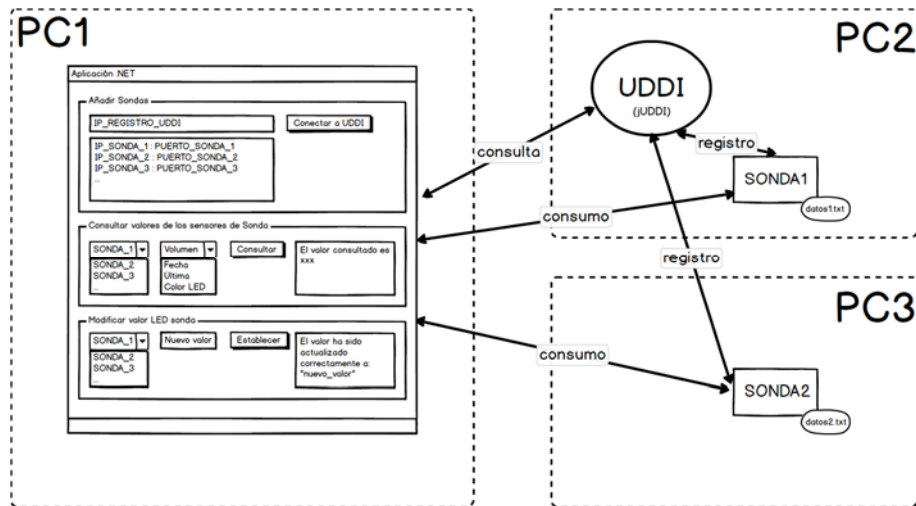


Imagen 2. Escenario de despliegue parte UDDI.

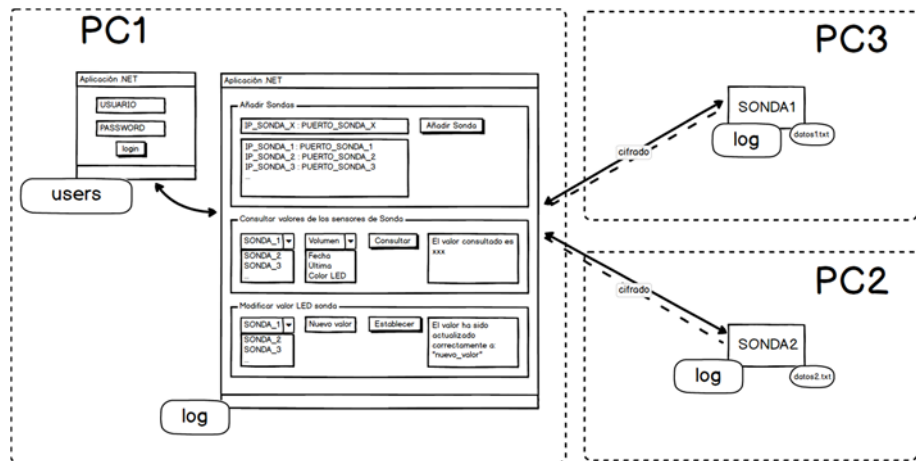


Imagen 3. Escenario de despliegue parte SEGURIDAD.

Como puede verse en el escenario de despliegue, al menos cada máquina de servicio sonda debe estar en un PC distinto, de forma que el escenario nuevamente lo componen al menos 3 PCs (entornos de ejecución real o virtual) diferentes. Las aplicaciones deben ser arrancadas desde los archivos compilados y ejecutables (no deben ser iniciados utilizando entornos de desarrollo de .NET, NetBeans, etc.).

De la misma forma que las sondas, los actuadores también deberían poder ser distribuidos en diferentes PC's.

En el caso de UDDI, los servicios deberán registrarse primero para poder ser consumidos por la aplicación .NET, que utilizará el registro UDDI para conocer la ubicación de los servicios a consumir.

Igualmente ocurre con la parte de seguridad y auditoría. Deberá haber una primera interfaz de login y existir un log en cada uno de los elementos que compone el sistema además del cifrado de la comunicación.

Entregables y evaluación

La evaluación de la práctica se realizará en los laboratorios. El estudiante debe desplegar, él mismo, la práctica que resuelve el enunciado anterior. Debe desplegar un sistema completo, es decir, la aplicación .NET y al menos dos servicios sonda y los actuadores correspondientes en dos máquinas distintas, todos ellos interconectados entre sí. **Este es el requisito indispensable para poder realizar la corrección.** Además deben poderse evaluar positiva o negativamente todos los apartados que aparecen en la Guía de corrección (ver documento de guía de corrección de la práctica de servicios web). Cada uno de los apartados puntúa de forma variable, por tanto cada apartado no implementado o que no pueda comprobarse su correcto funcionamiento no podrá ser tenido en cuenta y por tanto no puntuará. El estudiante deberá presentar para la evaluación el documento “Guía de corrección” cumplimentado para que el profesor pueda validar los apartados implementados.

El estudiante deberá entregar, además, por tutoría virtual a su profesor de prácticas una memoria de prácticas, un documento donde se detalle la siguiente información. El formato es libre pero debe ser un documento ordenado y debidamente formateado, cuidando la redacción y ortografía.

- Portada con el nombre, apellidos y DNI del estudiante, año académico y el título de la práctica.
- Una informe donde se indique el nombre de los componentes software desarrollados y una descripción de cada uno de ellos (aplicación .NET, servicios, documentos WSDL...).
- El detalle, paso a paso, de una guía de despliegue de la aplicación, que deberá ser la misma que utilice cuando haga la corrección de la práctica.
- Capturas de pantalla que muestren la aplicación en marcha.

Cada profesor de prácticas podrá solicitar a los estudiantes cualquier otra evidencia que considere adecuada para poder formalizar la evaluación al igual que podrá hacer cualquier pregunta sobre el código que implementa la práctica.