

SD	Sistemas Distribuidos
15/16	Práctica no guiada
	Control Estaciones Meteorológicas 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 opcionales, a elegir **obligatoriamente** una de ellas:

- UDDI
- Seguridad

El estudiante deberá implementar cualquiera de las dos partes, la que él o ella escoja, sobre la práctica. Solo una de ellas puntuará en la nota final de la práctica, aunque si el estudiante implementa ambas partes, será tenido en cuenta en la nota final de la asignatura en caso de necesidad.

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 organización que se dedica a la monitorización de variables climáticas en estaciones repartidas por un amplio territorio, para ofrecer así el estado meteorológico del país.

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 estaciones meteorológicas serán expuestas como servicios web que podrán ser accesibles desde una aplicación consumidora de servicios web. Este consumidor será una aplicación de escritorio .NET a la que se le indicará dónde encontrar los servicios (IP:PUERTO) y podrá acceder a ellos para consumirlos, tal y como muestra este esquema:

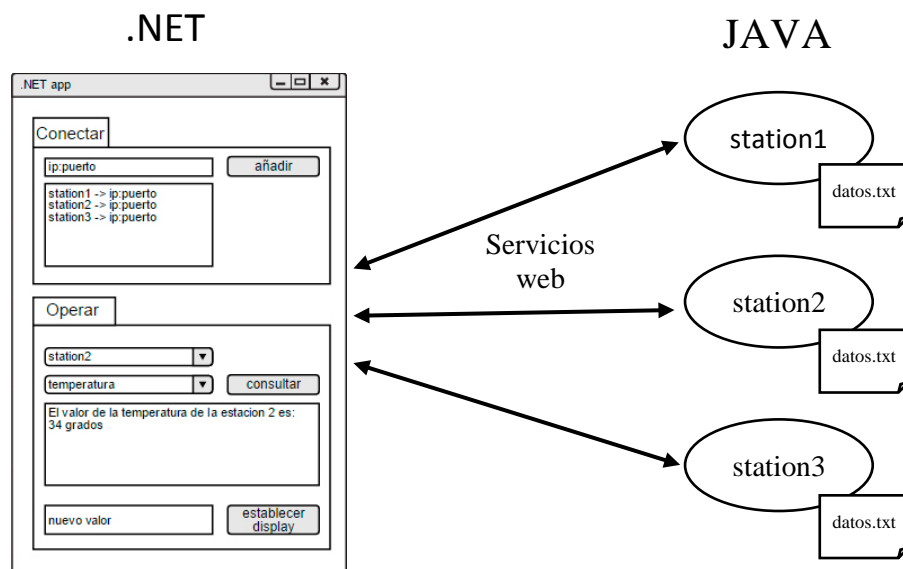


Imagen 1. Esquema conceptual del Sistema software, interconexiones entre los componentes y plataformas.

Nota: la imagen 1 muestra un mockup de la posible aplicación en .NET, en ningún caso es una especificación de interfaz, el estudiante debe diseñar e implementar su aplicación conforme a las especificaciones detalladas.

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 es el desarrollo continúa siendo la distribuir la información de las estaciones meteorológicas, cuyas variables siguen siendo las mismas que en la pasada práctica, con las mismas características:

Sensores	Temperatura: obtiene un valor de temperatura de entre -30°C y 50 °C Humedad: obtiene un valor de entre 0 y 100 Luminosidad: obtiene un valor de entre 0 y 800
Actuadores	Pantalla LCD: pantalla que muestra mensajes, tamaño de 150 caracteres

Tabla 1. Sensores y actuadores de cada estación meteorológica

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 estaciones meteorológicas, sino que el cliente, en este caso la aplicación .NET puede acceder y consumir directamente los servicios de consulta del estado de sensores y actuadores, y los servicios de actualización en este caso del display directamente.

Aplicación cliente .NET

Esta aplicación actúa como consumidor de los servicios ofrecidos por las estaciones meteorológicas 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.

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 “station”, estación meteorológica, es decir, debe ser capaz de conectarse a 1 o n servicios de este tipo, y consultar cualquier de sus variables o actualizar su display. La siguiente imagen muestra un ejemplo de cómo debe funcionar la aplicación .NET.

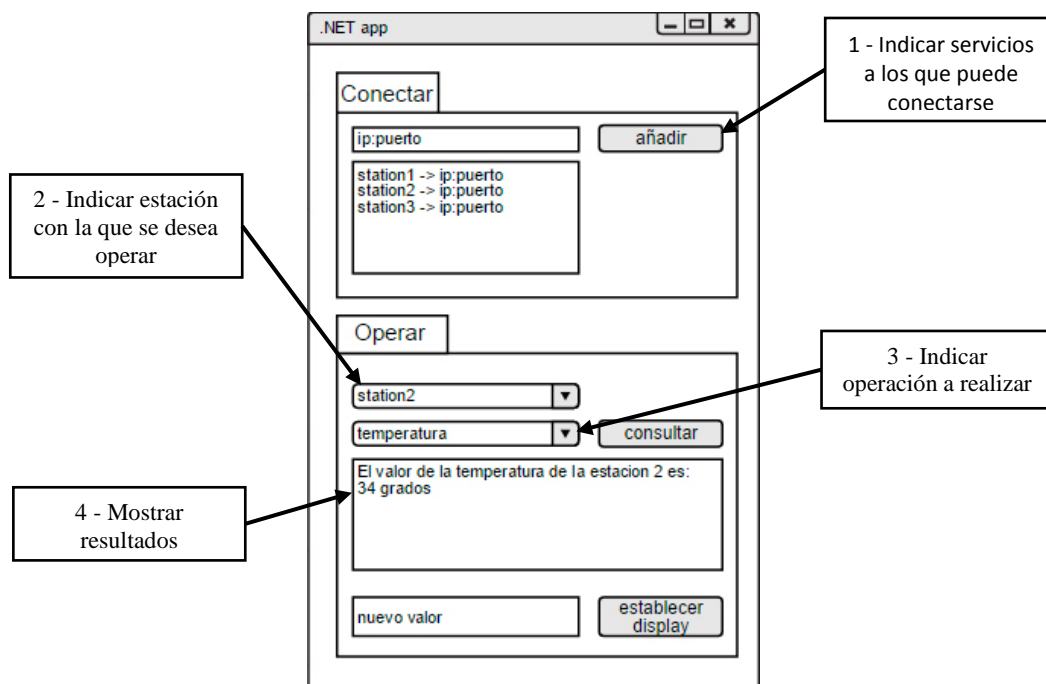


Imagen 2. Acciones a realizar en la aplicación .NET

Servicios station

Al igual que ocurría en RMI, cada estación meteorológica 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 estación meteorológica en un archivo de datos.

Se podrán desplegar tantas estaciones meteorológicas como se deseen, conteniendo cada una de ellas los servicios necesarios para interactuar con los datos y poder así cumplir con las especificaciones.

Parte opcional 1 – UDDI

Esta parte opcional es **obligatoria** si no se elige realizar la “Parte opcional 2: Seguridad” que se describe más adelante, en este documento.

La utilización de un registry 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 acceso.

Se pide al estudiante que implemente la utilización de un registro UDDI en el sistema, de forma que todos los servicios serán registrados previamente en el registro UDDI, y la aplicación .NET solo tenga que acceder al registro UDDI para conocer la ubicación y acceso de los servicios. La siguiente imagen muestra cómo ha de funcionar la aplicación .NET en caso de que se utilice UDDI.

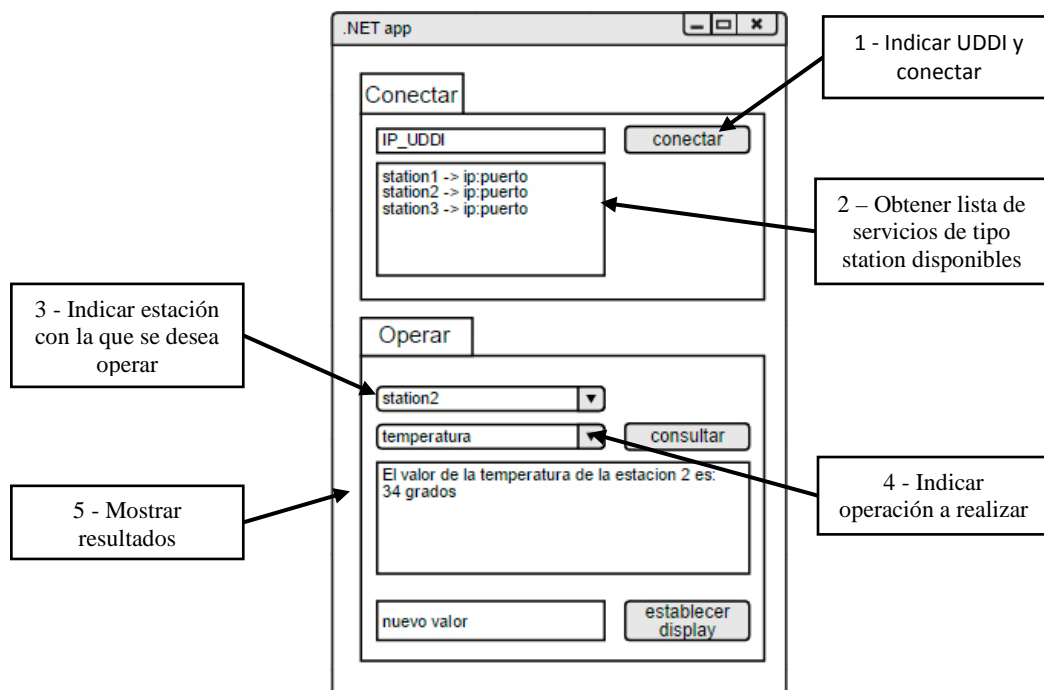


Imagen 3. Acciones a realizar en la aplicación .NET cuando se utiliza UDDI

Como puede apreciarse en la imagen 3, al utilizar el registro UDDI ya no es necesario indicar una por una cada una de las estaciones a las que es posible conectarse, ya que a través de las funciones de consulta, se puede obtener la lista de los servicios tipo “station” que se encuentran desplegados.

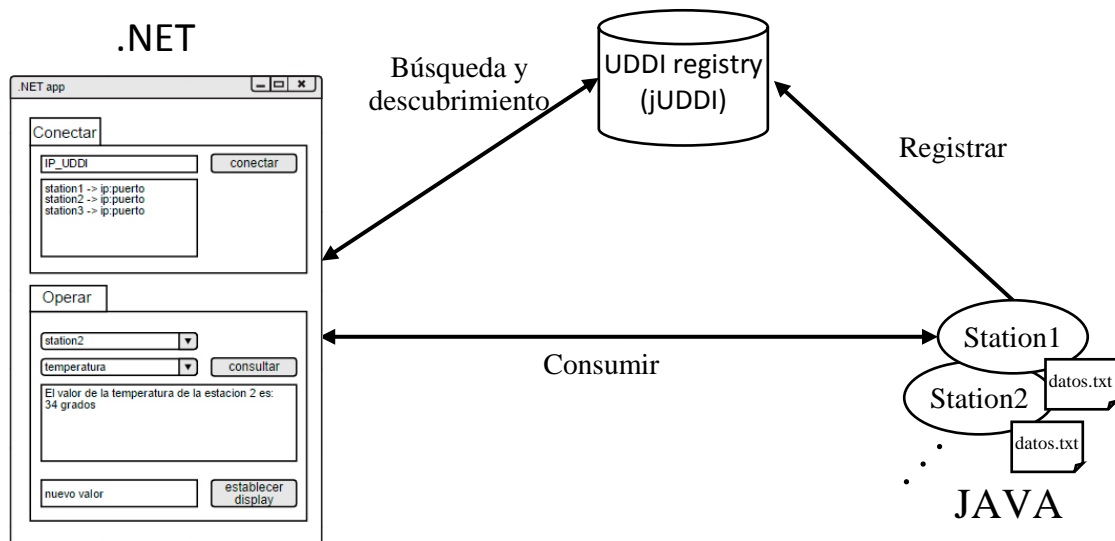


Imagen 4. Esquema conceptual del sistema desplegado utilizando UDDI

La imagen 4 muestra un esquema del funcionamiento del sistema cuando se hace uso del registro UDDI. En este caso, las estaciones meteorológicas primero deben registrarse en el registro UDDI para poder ser accedidas desde la aplicación .NET. La aplicación consultará en el registro UDDI que estaciones han sido desplegadas y luego ya accederá a cada una de ellas para consumir sus servicios.

Parte opcional 2 – Seguridad

Esta parte opcional es **obligatoria** si no se ha elegido realizar la “Parte opcional 1: UDDI” descrita anteriormente.

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). 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 de 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 como los servicios 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).

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 meteorológico en cada uno, siguiendo el esquema:

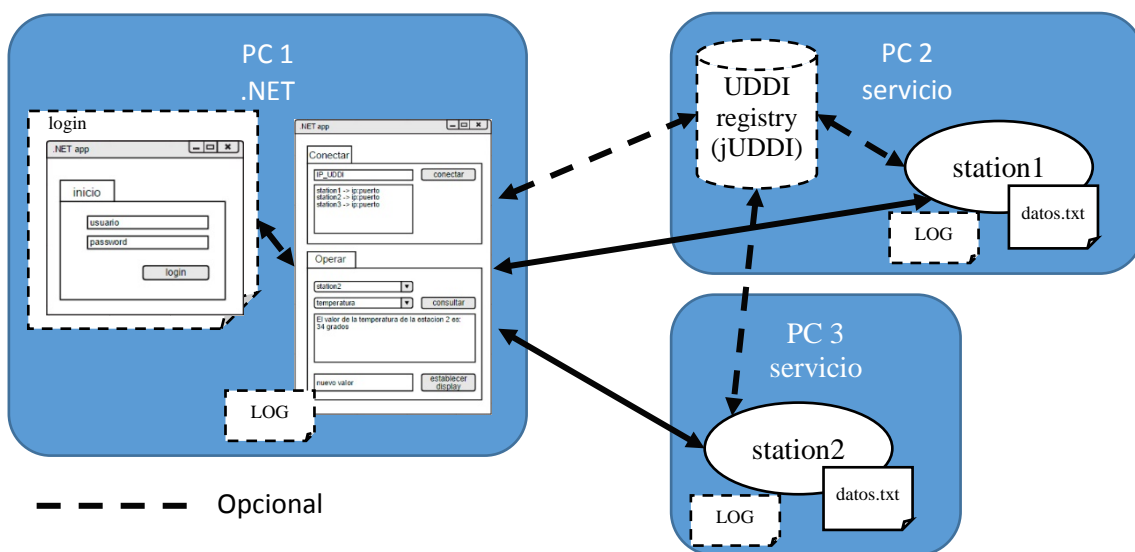


Imagen 5. Escenario físico mínimo para el despliegue de la práctica.

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

El escenario muestra que la parte UDDI es opcional, por lo que podrá ser incluida o no en el despliegue dependiendo de si el estudiante ha decidido implementarla. En caso de existir, los servicios deberán registrarse primero de 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, de existir en la aplicación .NET deberá haber una primera interfaz de login y existir un log en cada uno de los elementos que compone el sistema.

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 meteorológicos 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.