

## **DESCRIPCIÓN Y DESCUBRIMIENTO DE SERVICIOS WEB CON UDDI**

### UDDI: REGISTRO DE SERVICIOS WEB

UDDI es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que éstas ofrecen. A través de UDDI, se puede publicar y descubrir información de una empresa y de sus servicios. Se puede utilizar sistemas taxonómicos estándar para clasificar estos datos y poder encontrarlos posteriormente en función de la categorización. Lo más importante es que UDDI contiene información sobre las interfaces técnicas de los servicios de una empresa. A través de un conjunto de llamadas a API XML basadas en SOAP, se puede interactuar con UDDI tanto en tiempo de diseño como de ejecución para descubrir datos técnicos de los servicios que permitan invocarlos y utilizarlos. De este modo, UDDI sirve como infraestructura para una colección de software basado en servicios Web.

¿Por qué UDDI? ¿Por qué resulta necesario un registro de este tipo? Teniendo en cuenta que existe una colección de software de miles (quizás millones) de servicios Web, se nos plantean varias cuestiones difíciles:

¿Cómo se descubren los servicios Web?

¿Cómo se categoriza la información de forma coherente?

¿Cómo repercute esto en la localización?

¿Cómo afecta a las tecnologías de propietario? ¿Cómo se puede garantizar la interoperabilidad del mecanismo de descubrimiento?

¿Cómo se puede interactuar en tiempo de ejecución con este mecanismo de descubrimiento cuando mi aplicación depende de un servicio Web?

La iniciativa UDDI surgió como respuesta a estas preguntas. Varias empresas, incluidas Microsoft, IBM, Sun, Oracle, Compaq, Hewlett Packard, Intel, SAP y unas trescientas más (para obtener un listado completo, consulte UDDI: Community [en inglés]), unieron sus esfuerzos para desarrollar una especificación basada en estándares abiertos y tecnologías no propietarias que permitiera resolver los retos anteriores. El resultado, cuya versión beta se lanzó en diciembre de 2000 y estaba en producción en mayo de 2001, fue un registro empresarial global alojado por varios nodos de operadores en el que los usuarios podían realizar búsquedas y publicaciones sin coste alguno.

A partir de la creación de esta infraestructura para servicios Web, los datos sobre estos servicios se pueden encontrar de forma sistemática y confiable en una capacidad universal totalmente independiente de proveedores. Se pueden llevar a cabo búsquedas categóricas precisas utilizando sistemas de identificación y taxonómicos extensibles. La integración de UDDI en tiempo de ejecución se puede incorporar a las aplicaciones. Como resultado, se fomenta el desarrollo de un entorno de software de servicios Web.

## **FUNCIONAMIENTO**

La información de UDDI se aloja en nodos de operador, empresas que se han comprometido a ejecutar un nodo público conforme a la especificación que rige el consorcio UDDI.org. En la actualidad existen dos nodos públicos que se ajustan a la versión 1 de la especificación UDDI: Microsoft aloja uno e IBM el otro. Hewlett Packard se ha comprometido a alojar un nodo bajo la versión 2 de la especificación. Los operadores del host deben replicar datos entre ellos a través de un canal seguro, para conseguir la redundancia de la información en el registro UDDI. Se pueden publicar los datos en un nodo y descubrirlos en otro tras la réplica. Actualmente, la réplica se produce cada 24 horas. En el futuro, este intervalo entre réplicas se reducirá, ya que habrá más aplicaciones que dependan de los datos de UDDI.

Resulta importante observar que no existen requisitos de propietario respecto al modo en que el operador del host implementa su nodo. El nodo sólo se debe ajustar a la especificación UDDI. El nodo de Microsoft (<http://uddi.microsoft.com/default.aspx> [en inglés]), por ejemplo, se ha escrito por completo en C# y se ejecuta en producción en tiempo de ejecución en lenguaje común .NET Beta 2. El código de base se beneficia claramente de la compatibilidad nativa con SOAP y de la serialización que ofrecen las clases de sistema .NET. En el lado del servidor, el nodo del operador Microsoft utiliza Microsoft® SQL Server 2000 como almacén de datos. Creo que basta con mencionar que IBM utiliza tecnologías diferentes para ejecutar su nodo, ¿verdad?. No obstante, los dos nodos se comportan exactamente igual, ya que se ajustan al mismo conjunto de llamadas a API XML basadas en SOAP. Las herramientas de los clientes pueden interoperar con ambos nodos sin problemas. Por eso, el nodo público UDDI constituye un claro ejemplo de que el modelo de servicios Web XML funciona en entornos heterogéneos.

El próximo paso para comprender la iniciativa UDDI consiste en ver qué datos se almacenan en UDDI y cómo se estructuran. UDDI es relativamente ligero; se ha diseñado como registro, no como depósito. La diferencia, aunque sutil, resulta esencial. Un registro redirige al usuario a recursos, mientras que un depósito sólo almacena información. El registro Microsoft® Windows® puede servir de ejemplo: contiene las configuraciones y parámetros básicos pero, en última instancia, su función es la de dirigir la aplicación a un recurso o binario. Buscar un componente COM basándonos en su Id. de programa nos conducirá a un Id. de clase, que a su vez nos dirigirá a la ubicación del binario.

UDDI se comporta de forma similar: como el registro de Windows, se basa en identificadores únicos globales (GUID) para garantizar la capacidad de búsquedas y determinar la ubicación de

recursos. En última instancia, las consultas a UDDI conducen a una interfaz (un archivo .WSDL, .XSD, .DTD, etc.) o a una implementación (como un archivo .ASMX o .ASP) ubicadas en otro servidor. Por tanto, UDDI puede responder a este tipo de preguntas:

"¿Qué interfaces de servicios Web basadas en WSDL se han publicado y establecido para un sector determinado?"

"¿Qué empresas han escrito una implementación basada en una de estas interfaces?"

"¿Qué servicios Web, categorizados de algún modo, se ofrecen actualmente?"

"¿Qué servicios Web ofrece una empresa determinada?"

"¿Con quién se debe poner en contacto el usuario para utilizar los servicios Web de una empresa?"

"¿Cuáles son los detalles de implementación de un servicio Web concreto?"

### **WSDL y UDDI**

WSDL se ha convertido en una pieza clave de la pila de protocolos de los servicios Web. Por eso, es importante saber cómo colaboran UDDI y WSDL y por qué la idea de interfaces frente implementaciones forma parte de cada protocolo. WSDL y UDDI se diseñaron para diferenciar claramente los metadatos abstractos y las implementaciones concretas. Para entender cómo funcionan WSDL y UDDI resulta esencial comprender las consecuencias de esta división.

Por ejemplo, WSDL distingue claramente los mensajes de los puertos: los mensajes (la sintaxis y semántica que necesita un servicio Web) son siempre abstractos, mientras que los puertos (las direcciones de red en las que se invoca al servicio Web) son siempre concretos. No es necesario que un archivo WSDL incluya información sobre el puerto. Un archivo WSDL puede contener simplemente información abstracta de interfaz, sin facilitar datos de implementación concretos, y ser válido. De este modo, los archivos WSDL se separan de las implementaciones.

Una de las consecuencias más interesantes de esto es que pueden existir varias implementaciones de una única interfaz WSDL. Este diseño permite que sistemas dispares escriban implementaciones de la misma interfaz, para garantizar así la comunicación entre ellos. Si tres empresas diferentes implementan el mismo archivo WSDL y una parte del software de cliente crea el código auxiliar/proxy a partir de esa interfaz, dicho software se podrá comunicar con las tres implementaciones con el mismo código de base, cambiando simplemente el punto de acceso.

UDDI establece una distinción similar entre la abstracción y la implementación con el concepto de tModels. La estructura tModel, abreviatura de "Technology Model" (modelo de tecnología), representa huellas digitales técnicas, interfaces y tipos abstractos de metadatos. El resultado de los tModels son las plantillas de enlace, que son la implementación concreta de uno o más tModels. Dentro de una plantilla de enlace se registra el punto de acceso de una implementación particular de un tModel. Del mismo modo que el esquema de WSDL permite separar la interfaz y la implementación, UDDI ofrece un mecanismo que permite publicar por separado los tModels de las plantillas de enlace que hacen referencia a ellos. Por ejemplo, un grupo industrial o de

estándares publica la interfaz canónica para un sector particular y, a continuación, varias empresas escriben implementaciones de esta interfaz. Obviamente, cada una de estas implementaciones haría referencia al mismo tModel. Los archivos WSDL constituyen un ejemplo perfecto de tModel de UDDI.

### **Registro en UDDI**

La publicación en UDDI es un proceso relativamente sencillo. El primer paso consiste en determinar información básica sobre cómo definir la empresa y los servicios en UDDI. El siguiente paso, una vez determinada esta información, consiste en llevar a cabo el registro, ya sea mediante programación o a través de una interfaz de usuario basada en el Web. Por último, se debe probar la entrada para asegurar que se registró correctamente y que aparece tal y como se esperaba en diferentes tipos de búsquedas y herramientas.

#### **PRIMER PASO: Definir la entrada de UDDI**

Partiendo del modelo de datos descrito anteriormente, se debe recopilar cierta información importante antes de establecer una entrada de UDDI.

***Determine los tModels*** (archivos WSDL) que utilizan las implementaciones del servicio Web.

Al igual que sucede en el desarrollo de un componente COM, el servicio Web se ha desarrollado a partir de una interfaz existente o de una interfaz de diseño propio. En el caso de un servicio Web basado en una interfaz WSDL existente, deberá determinar si el archivo WSDL se ha registrado en UDDI. Si es así, deberá comprobar su nombre y tModelKey, que es el identificador GUID que generó UDDI cuando se produjo el registro.

Por el contrario, si el servicio Web se basa en un archivo WSDL que no se ha registrado en UDDI, deberá crear un nuevo tModel para representar esta interfaz. El nombre de este tModel debería tener un formato URI (identificador de recursos uniforme), como MyCompany-com:SampleWebService-interface:v1, y señalar a la ubicación del archivo WSDL.

Si su servicio Web es un servicio de Microsoft® Visual Studio® .NET, se puede generar una descripción WSDL utilizando una cadena de consulta desde el archivo .ASMX (como <http://www.mycompany.com/SampleWebService.asmx?wsdl>). No obstante, el archivo WSDL generado por Visual Studio .NET se relaciona estrechamente con el punto de acceso para la invocación del servicio Web, lo cual puede no resultar adecuado cuando la interfaz del servicio tiene varias implementaciones. Esto no supondrá ningún problema si su intención es que el archivo WSDL sólo tenga una implementación.

Determine el nombre de la empresa y una breve descripción de la misma en varios idiomas, si es necesario, así como los contactos principales para los servicios Web que ofrece.

UDDI es compatible con el espacio de nombre xml:lang, lo que permite a las empresas ofrecer su descripción en varios idiomas. Asimismo, UDDI permite enumerar los contactos, incluyendo datos como el correo electrónico, el teléfono y la dirección. Esta lista de contactos muestra los recursos de una empresa con los que se puede poner en contacto en relación con los servicios Web ofrecidos. Por ejemplo, si un usuario desea comenzar a utilizar el servicio Web deberá ponerse en contacto con el responsable de relaciones comerciales correspondiente pero, ¿cómo puede llegar a saber quién es? ¿Existe algún contacto para obtener asistencia técnica a la hora de utilizar los servicios Web de la empresa? También se debería incluir en la lista a esta persona.

### **Determine los servicios Web que la empresa ofrece a través de UDDI.**

A continuación, se debe determinar los servicios Web a registrar con la empresa en el nodo público UDDI. ¿Existen varios puntos de acceso para este servicio? ¿Es preciso que los clientes conozcan otros parámetros y otra información para utilizar el servicio Web?

Resulta importante destacar que no todo el mundo puede obtener acceso a un servicio Web porque éste se haya registrado en UDDI. A una entrada de registro UDDI le pueden acompañar medidas de seguridad, autorización y autenticación. No basta que el usuario sepa que existe un servicio Web para que pueda invocarlo. Puede existir una comunicación fuera de banda entre empresas antes de permitir el acceso a un servicio Web.

### **SEGUNDO PASO: Registrar la entrada de UDDI**

Una vez finalizada la tarea de definición, el siguiente paso consiste en registrar la empresa. Deberá obtener una cuenta con un registro UDDI. Esta operación no se puede realizar mediante programación, ya que deberá mostrar su conformidad con una declaración de condiciones de uso. El nodo de Microsoft utiliza Passport para la autenticación, así que deberá adquirir una cuenta de Passport (<http://www.passport.com/Consumer/default.asp>) para continuar con el registro.

En este punto se ofrecen dos opciones:

- 1.- Se puede utilizar la interfaz de usuario Web del nodo de Microsoft o realizar el registro mediante programación dirigiendo al propio nodo las llamadas a API de SOAP. Si no piensa modificar la entrada o ésta es relativamente simple, bastará con la interfaz de usuario Web. No obstante, si pretende actualizar la entrada con frecuencia, o bien, ésta es más compleja, resulta recomendable realizar el proceso de registro con secuencias de comandos, utilizando el SDK de Microsoft UDDI. Además, la interfaz de usuario de Microsoft no está localizada en otros idiomas, así que se deberá registrar mediante programación para disfrutar esa característica de la API de UDDI.

- 2.- Uso de la interfaz de usuario Web de Microsoft Registrar un servicio utilizando la interfaz de usuario Web de Microsoft es un proceso relativamente intuitivo. Para comenzar, diríjase a la página de administración

(<https://uddi.microsoft.com/register.aspx>). Una vez iniciada la sesión, aparecerá la opción para registrar tModels y empresas. Antes de continuar, debe saber lo siguiente:

- Asegúrese de registrar el archivo WSDL como un tModel antes de registrar el servicio, ya que el tModel será necesario en una etapa posterior del proceso.
- Al registrar un documento WSDL como un tModel, debería clasificarlo utilizando el sistema taxonómico UDDI Types. Como mínimo, se debería clasificar el archivo WSDL como "Specification for a Web Service" (wsdlSpec).

De este modo, se garantiza que el tModel se categoriza conforme al documento con las prácticas de uso más adecuadas "Using WSDL in a UDDI Registry" (en inglés). Es esencial clasificar de algún modo el tModel, ya que puede contener referencias a documentos que no sean archivos WSDL. Existen herramientas, como Visual Studio .NET, que se basan en estas clasificaciones para reducir el conjunto de resultados de las consultas realizadas.

- Una vez haya registrado la interfaz WSDL como un tModel, deberá agregar el nombre de una empresa junto con la información de contacto y clasificación pertinente. Podrá agregar todas las clasificaciones que considere oportunas.

Al continuar agregando los servicios Web que desea exponer a través de UDDI. Deberá agregar un enlace para cada servicio, ya que éstos puede tener varias implementaciones. En cada enlace, se proporcionará el punto de acceso al servicio Web, por ejemplo, <<http://www.mycompany.com/SampleWebService.asmx>>.

Cada enlace debe crear una referencia a las interfaces con las que es compatible, lo que se conoce en la UI de Microsoft como "specification signatures" (firmas de especificación). Una firma de especificación es el tModel que contiene la interfaz WSDL. La IU de Microsoft ofrecerá una pantalla para buscar el tModel en función de su URN. Este tModel puede ser el que registró en el primer paso o el tModel de un archivo WSDL registrado por un tercero.

Por último, puede agregar a la ubicación http un documento general sobre esa instancia concreta del servicio Web, así como los parámetros de la instancia que considere oportunos.

Registrar mediante programación con el SDK de Microsoft UDDI .NET

La otra opción de registro consiste en realizar esta operación mediante programación. Con el SDK de Microsoft UDDI este proceso resulta muy sencillo. Deberá obtener una cuenta de UDDI utilizando la interfaz de usuario Web pero, antes de finalizar esta tarea, podrá seguir con el resto del proceso a través de una secuencia de comandos. Primero, descargue e instale el SDK de UDDI desde <http://msdn.microsoft.com/webservices/downloads/default.aspx>. A continuación, cree una nueva aplicación de consola C# utilizando Visual Studio .NET. Agregue una referencia a la dll del SDK de Microsoft UDDI, que se instalará de forma predeterminada en C:\Archivos de programa\Microsoft UDDI SDK\VS7\Microsoft.Uddi.Sdk.dll. A continuación, agregue al principio del código algunas referencias a espacios de nombres.

```
using Microsoft.Uddi;  
using Microsoft.Uddi.Binding;  
using Microsoft.Uddi.Business;  
using Microsoft.Uddi.Service;  
using Microsoft.Uddi.ServiceType;
```

Agregue el siguiente código a la función estática void Main (string[] args):

```
//Puede ejecutar inicialmente este programa en el registro de prueba en  
//https://test.uddi.microsoft.com/publish  
Publish.Url = "https://uddi.microsoft.com/publish";  
Publish.User = "su cuenta";  
Publish.Password = "*****";
```

Este código establecerá la autenticación de la cuenta. A continuación, agregue el siguiente código para publicar los archivos WSDL como un tModel:

```
//Crear tModel  
SaveTModel stm = new SaveTModel();  
stm.TModels.Add();  
stm.TModels[0].Name = "Insertar URN";  
stm.TModels[0].Descriptions.Add("es", "Insertar descripción");  
stm.TModels[0].OverviewDoc.OverviewURL = "Insertar URL del WSDL";  
//La siguiente línea es necesaria para la correcta categorización del tModel  
stm.TModels[0].CategoryBag.Add  
( "uddi-org:types",  
"wsdlSpec",  
"uuid:c1acf26d-9672-4404-9d70-39b756e62ab4" );  
  
string sTModelKey = "";  
  
//Enviar a UDDI  
try  
{  
    TModelDetail tmd = stm.Send();  
    sTModelKey = tmd.TModels[0].TModelKey;  
}  
catch (UddiException ue)  
{  
    Console.WriteLine ( ue.Message );  
    return;  
}  
catch (Exception e)  
{  
    Console.WriteLine ( e.Message );  
    return;  
}  
}
```

Una vez guardado correctamente, UDDI generará un nuevo y único tModelKey, que se utilizará más adelante para el enlace del servicio Web. A continuación, cree la entrada de la empresa:

```

//Crear Business
SaveBusiness sb = new SaveBusiness();
sb.BusinessEntities.Add();
sb.BusinessEntities[0].Name = "Insertar nombre de la empresa";
sb.BusinessEntities[0].Descriptions.Add("es","Insertar descripción");

//Crear BusinessService
sb.BusinessEntities[0].BusinessServices.Add();
sb.BusinessEntities[0].BusinessServices[0].Name = "Insertar nombre del servicio";
sb.BusinessEntities[0].BusinessServices[0].Descriptions.
Add("es","Insertar descripción del servicio");

//Crear BindingTemplate
sb.BusinessEntities[0].BusinessServices[0].BindingTemplates.Add();
sb.BusinessEntities[0].BusinessServices[0].BindingTemplates[0].
Description.Add("es","Insertar descripción del enlace");
sb.BusinessEntities[0].BusinessServices[0].BindingTemplates[0].
AccessPoint.Text = "Insertar punto de acceso";
sb.BusinessEntities[0].BusinessServices[0].BindingTemplates[0].
AccessPoint.URLType = Microsoft.Uddi.Api.URLTypeEnum.Http;

//Crear tModelInstanceInfo
sb.BusinessEntities[0].BusinessServices[0].BindingTemplates[0].
TModelInstanceDetail.TModelInstanceInfos.Add();
sb.BusinessEntities[0].BusinessServices[0].BindingTemplates[0].
TModelInstanceDetail.TModelInstanceInfos[0].Descriptions.
Add("es","Insertar descripción");
//Utilizar la cadena tModelKey anterior
sb.BusinessEntities[0].BusinessServices[0].BindingTemplates[0].
TModelInstanceDetail.TModelInstanceInfos[0].TModelKey = sTModelKey;

//Enviar a UDDI
try
{
    BusinessDetail bd = sb.Send();
    //mostrar xml
    Console.WriteLine ( bd );
}
catch (UddiException ue)
{
    Console.WriteLine ( ue.Message );
    return;
}
catch (Exception e)
{
    Console.WriteLine ( e.Message );
    return;
}
}

```

La definición WSDL y la información sobre la empresa ya se han guardado en UDDI. Podrá editar estas entradas siempre que lo desee introduciendo la clave correspondiente.

### **TERCER PASO: Buscar la entrada en UDDI**



Es recomendable realizar tres comprobaciones una vez registrada la entrada en UDDI. En primer lugar, utilizando la interfaz de usuario Web de Microsoft, busque la empresa por su nombre y categorizaciones para verla entre los conjuntos de resultados devueltos. En segundo lugar, abra Visual Studio .NET y asegúrese de que aparece en el cuadro de diálogo "Agregar referencia Web". Si no aparece, se puede deber a que el tModel no se categorizó correctamente utilizando la taxonomía `uddi-org:types` descrita anteriormente. Podrá agregar el servicio Web al proyecto y generar el código proxy basado en el archivo WSDL. Por último, transcurridas 24 horas, la entrada se replicará al nodo de IBM y podrá buscarla con su IU en <https://www-3.ibm.com/services/uddi/protect/find> (en inglés).

#### Registrarse en UDDI

Nunca se había registrado Cold Rooster en el sitio Microsoft UDDI (en inglés), así que lo primero que debíamos hacer era crear una cuenta en este sitio. Para registrar una cuenta es necesario iniciar sesión en Passport. Una vez iniciada la sesión, se puede configurar como dirección de correo electrónico de contacto en UDDI la dirección vinculada a la cuenta de Passport u otra diferente.

En la pantalla de registro se solicita otra información de contacto, como el nombre del que se registra, un número de teléfono y una dirección de correo. Para finalizar la operación de registro, es preciso aceptar un acuerdo con las condiciones de uso (consulte las condiciones de uso [en inglés]).

Una vez registrada la información de contacto y aceptadas las condiciones de uso, el sitio UDDI le enviará un correo electrónico para confirmar la dirección. Haga clic en el vínculo que aparece en este mensaje para administrar la cuenta de UDDI. La administración de una cuenta implica, por ejemplo, el registro de datos sobre la empresa, la publicación de tModels y la edición de información del registro. La figura 1 muestra la página de administración.



Figura 1. Página de administración de UDDI

#### Agregar la empresa a UDDI

Una vez establecida nuestra cuenta en el sitio Microsoft UDDI, el siguiente paso consistía en agregar Cold Rooster Consulting al registro UDDI. La empresa se podía registrar a través de la página Web o mediante la API de UDDI utilizando el SDK de .NET o el SDK de COM. Decidí utilizar la IU, ya que nuestra intención era registrar la información sólo una vez. Si más adelante observáramos que se cambian partes de la interfaz con regularidad, automatizaríamos este proceso para reducir el riesgo de errores. Una vez agregada la empresa al registro, los usuarios podrán encontrarnos a partir de nuestra actividad y el tipo de servicios Web que ofrecemos. Para agregar Cold Rooster Consulting al registro UDDI, seleccionamos el vínculo "Add a new business" (Agregar nueva empresa) de la página de administración (figura 1). La primera página solicita el nombre y la descripción de la empresa. Para Cold Rooster, escribí:

Name (Nombre): Cold Rooster Consulting

Description (Descripción): empresa ficticia utilizada por el equipo MSDN Architectural Samples

Una vez registrada la empresa en UDDI, se podía agregar más información:

Contacts (Contactos): personas con las que el usuario se puede poner en contacto para obtener ayuda sobre aspectos relacionados con los servicios ofrecidos por la empresa. Agregamos a la lista de contactos varios miembros del equipo MSDN Architectural Samples.

Services (Services): los tModels (archivos WSDL) que expone la empresa. Agregamos los tModels Account, Logon y Report a la entrada de UDDI.

Identifiers (Identificadores): datos exclusivos de esta empresa, tales como un número de registro mercantil. No rellenamos este campo ya que Cold Rooster no dispone de este tipo de datos.

Business classifications (Clasificaciones de la empresa): éstas identifican la ubicación de la empresa, así como la actividad a la que se dedica. Cold Rooster Consulting se encuentra en el estado de Washington, en Estados Unidos.

Discovery URLs (URL de descubrimiento): ofrece una ubicación en la que se puede encontrar información sobre la empresa.

Cumplimentaremos los puntos más simples primero y guardaremos los servicios para el final.

#### Agregar contactos a UDDI

Es muy sencillo. Tan sólo hay que rellenar los campos con información general sobre las personas de contacto, como si se tratase de un formulario de dirección corriente. A partir de las notas de descripción y uso, los usuarios potenciales consultan estos datos para ponerse en contacto con la empresa a la hora de conseguir la licencia para el servicio Web, obtener soporte técnico u otros asuntos relacionados. La figura 2 muestra el formulario con mi información de contacto.

The image shows a web form titled "Contact details" with a blue header. Below the header, there is a blue box containing the text: "Enter the name, description ( who this person is ) and usage notes ( under what circumstances you should establish contact ) For this contact." The form consists of several input fields and buttons:

- Name \***: Input field containing "Scott Seely".
- Description**: Input field containing "SDE".
- Usage notes**: Input field (empty).
- Email**: Input field containing "xxxxx@microsoft.com". To its right is a "Usage notes" field containing "Work e-mail" and an "Additional email" button.
- Phone**: Input field containing "425-555-1212". To its right is a "Usage notes" field containing "Not his real phone #" and an "Additional phone" button.
- Address**: Input field containing "One Microsoft Way" and "Redmond, WA 98052". To its right is a "Usage notes" field containing "Work Address" and an "Additional address" button.

At the bottom of the form, there is a legend: "\* Required Field". At the very bottom, there are two buttons: "Continue" and "Cancel".

Figura 2. Página con los datos de contacto

#### Clasificar la empresa

La mayoría de las empresas se pueden clasificar en función de la actividad a la que se dedican.

Existen varias taxonomías disponibles para la clasificación UDDI de entidades. Se pueden utilizar las siguientes:

- North American Industry Classification System (NAICS-1997)

- Universal Standard Products and Services Codes (UNSPSC-7.03)
- ISO 3166 Geographic Taxonomy
- Standard Industrial Classification (SIC-1987)
- GeoWeb Geographic Classification
- UDDI Types Taxonomy
- Registramos Cold Rooster bajo todas estas taxonomías, excepto UDDI Types, ya que este sistema taxonómico se utiliza para clasificar los tModels y la información sobre el servicio.
- Para saber qué vamos a registrar, tenemos que conocer a qué se dedica Cold Rooster y dónde se ubica. Cold Rooster Consulting tiene su sede en Redmond, Washington (EE.UU.), y ofrece servicios basados en proyectos y servicios adicionales de asesoramiento informático al personal. Asimismo, se especializa en el desarrollo basado en Windows e Internet. Partiendo de esta información, debíamos clasificar la empresa convenientemente en cada uno de los seis esquemas de clasificación. La tabla 1 muestra cómo se clasificó la empresa en cada esquema.

Esquema de clasificación	Clasificación
NAICS	<ul style="list-style-type: none"> <li>• 541511: Custom Computer Programming Services</li> <li>• 541512: Computer Systems Design Services</li> </ul>
UNSPSC	<ul style="list-style-type: none"> <li>• 81.11.16.07.00: Programming for C or C++</li> <li>• 81.11.16.03.00: Programming for HTML</li> <li>• 81.11.16.01.00: Programming for Microsoft® Visual Basic®</li> <li>• 81.11.16.12.00: Programming or Proprietary Languages (nosotros utilizamos también C#)</li> <li>• 81.11.21.06.00: Application Service Providers (nosotros alojamos servicios Web)</li> <li>• 81.11.21.03.00: World Wide Web (WWW) site design services</li> </ul>
ISO 3166	<ul style="list-style-type: none"> <li>• US-WA (Washington, USA, World)</li> </ul>
SIC	<ul style="list-style-type: none"> <li>• 7371: Computer programming services</li> <li>• 7372: Information retrieval services</li> </ul>
GeoWeb Geographic Classification	<ul style="list-style-type: none"> <li>• 518816 (Redmond, Washington, USA, North America, World)</li> </ul>

Tabla 1. Clasificaciones UDDI de ejemplo

La empresa ya se ha clasificado por completo. El siguiente paso consiste en agregar los tres tModels.

### **Agregar los tModels**

Para aquéllos que no leyeron el artículo de la semana pasada, un tModel es un modelo de tipos. En los servicios Web, tModel es generalmente sinónimo de archivo WSDL. Un tModel define los tipos que utiliza el servicio Web, al tiempo que incluye las definiciones de operación y mensaje de dicho servicio. Partiendo de un tModel concreto, se puede saber qué operaciones de servicio Web realiza la entidad que implementa dicho tModel y cómo obtener acceso a ellas. Los archivos WSDL se registran como tModels porque se pueden llegar a realizar varias implementaciones de estos últimos.

El servicio Web Favoritos del lado del servidor se compone de tres servicios: Logon, Account y Report. El servicio Web Logon permite al titular de licencia iniciar la sesión y obtener un token. Con este token, el titular de licencia puede obtener acceso a otros métodos de los servicios Web Account y Report. Para agregar los archivos WSDL, deberá distribuirlos en un servidor al que se pueda obtener acceso a través de Internet. Para este proceso se debe utilizar también la clasificación UDDI.

Para agregar un tModel, basta con seleccionar el vínculo "Add a new tModel" en la página de administración (figura 1). A continuación, se debe registrar información básica sobre el tModel: el nombre, una descripción y una ubicación para el documento WSDL. Una vez facilitados estos datos, se clasifica el tModel como un documento WSDL y se publica la información en el registro UDDI. Es muy sencillo. La figura 3 muestra los datos registrados para el tModel Logon.

**tModel Detail**

Please provide a name and brief description of this tModel. It is recommended that the tModel name be structured as a Uniform Resource Name (URN).

Service name: \*

Description:

\* Required fields.

**Overview Document**

An overview document provides further, detailed information about this tModel such as a specification for the XML documents that drive services using this type. The location will typically be a web address and you may enter a brief description of the document that is available at that location. For example, WSDL files can be registered as tModels; the overview document would then be the URL where that WSDL file can be found.

Document location:

Description:

Figura 3. Detalles del tModel Logon

En la pantalla siguiente, podrá agregar las clasificaciones del servicio y los identificadores de la empresa. Estos datos pueden coincidir con la información registrada sobre la empresa. Sólo se utiliza el sistema UDDI porque se trata de la clasificación de un servicio. Deberá seleccionar varios vínculos para especificar esta información:

- These types are used for tModels (tipos que se utilizan con los tModels)
- Specification for a Web Service (especificación de un servicio Web)
- Specification for a Web Service described in WSDL (especificación de un servicio Web descrito en WSDL)

Al registrar el servicio Favoritos de Cold Rooster, seguí estos pasos para los archivos Logon.WSDL, Account.WSDL y Report.WSDL. Con esta operación finaliza el proceso de registro en UDDI y se puede encontrar la información disponible en el registro.

#### Defining the Services (Definir los servicios)

Una vez determinados los tModels, se deben agregar los servicios para manifestar su existencia. Para ello, vuelva a la página de administración en la que ya aparecerá la empresa enumerada bajo el vínculo "Add a new business" (figura 1). Seleccione su empresa y desplácese hasta "Services". Haga clic en "Add a Service". Deberá completar los datos del servicio en la primera página. Para el servicio Web Logon, introduje la siguiente información:

Name (Nombre): Logon

Description (Descripción): valida los titulares de licencia y proporciona un token de acceso

A continuación, volví a clasificar el servicio como "Specification for a Web Service described in WSDL" y lo vinculé al tModel Logon registrado. En "Define a new binding", rellené los campos del siguiente modo:

Access point (Punto de acceso): <https://Coldrooster.com/SSF/Logon.asp>

URL type (Tipo de dirección URL): http

Description (Descripción): extremo del servicio Web Logon de Cold Rooster Consulting

Para finalizar, debía asociar este servicio con el servicio Web Favoritos: tModel Logon. En Specification Signatures, seleccione "Add specification signature". Para buscar el modelo por su nombre, escriba "Favorites Web Service" y se mostrarán los tres tModels guardados para este servicio. Seleccione "Favorites Web Service: Logon" y presione "Continue". Aparecerá una página Web en la que se deberá editar información adicional sobre el extremo. Yo rellené los campos de este modo:

- Edit specification signature (Editar firma de especificación); Description (Descripción): implementación de Cold Rooster del servicio Web Favoritos: tModel Logon.
- Instance details (Datos de instancia); Parameters (Parámetros): <http://msdn.microsoft.com/library/?url=/library/en-us/dncold/html/ssfapiref.asp?frame=true>
- Instance details (Datos de instancia); Description (Descripción): documento de referencia a API
- Overview document (Documento general); Document location (Ubicación del documento): <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncold/html/ssf1sec.asp>
- Overview document (Documento general); Description (Descripción): información de seguridad sobre Favoritos del lado del servidor

Una vez completados estos campos, volví a seleccionar "Continue". Aparecía una página en la que se podía revisar toda la información introducida sobre el servicio Logon antes de volver a la vista principal con los datos de la empresa y publicar todo en el registro UDDI. Repetí todo el proceso para los servicios Account y Report.

Buscar la información

A fecha de hoy (8 de octubre de 2001), sólo aparece una empresa en Redmond de acuerdo con el sistema GeoWeb Taxonomy: Cold Rooster Consulting. Realicé otra búsqueda, esta vez con la clasificación ISO 3166, y aparecieron nueve empresas. Las búsquedas con el resto de clasificaciones devolvieron otros resultados. En algunos esquemas de clasificación se produjo el doloroso resultado de varias páginas de empresas.

Como los datos ya se pueden descubrir, la gente interesada en un servicio Web que almacene direcciones URL favoritas acudirá, en teoría, a la solución de Cold Rooster para utilizarlo. A los usuarios de Microsoft® Visual Studio® .NET les resultará muy sencillo el uso de UDDI para buscar un servicio Web y agregarlo a su proyecto.

La primera vez que se utiliza el cuadro de diálogo Add Web Reference (ProjectAdd Web Reference), se ofrece la posibilidad de buscar una referencia Web (denominada también archivo WSDL) en el servidor de Microsoft UDDI. El extremo utilizado para Visual Studio es <http://uddi.microsoft.com/visualstudio/>. Le indico al extremo que busque todas las empresas cuyo nombre comience por "cold" y sólo encuentra los tres servicios Web que registré: Account, Logon y Report. Para agregar una referencia Web al servicio Logon, basta con hacer clic en "Logon" para expandir el nodo y, a continuación, seleccionar "Favorites Web Service: Logon" para ver el WSDL. Llegados a este punto, hago clic en "Add Reference" y ya puedo iniciar sesión en el servicio Web Favoritos.

Si ha seguido todos los pasos pero no dispone de una licencia de Favoritos, vaya a la consola de administración del servicio Favoritos (en inglés) y suscríbase para obtener una. Recibirá la contraseña en un plazo de 15 a 30 minutos. El código utilizado para conectar con el servicio Web Logon con Visual Basic es muy sencillo:

```
Sub Main()  
    Dim svc As New com.coldrooster.www.Logon()  
    System.Console.WriteLine(svc.Logon("Nombre de titular de licencia", "Contraseña"))  
    System.Console.WriteLine("Presione Return para salir")  
    System.Console.ReadLine()  
    svc.Dispose()  
End Sub
```

Este código describe el token GUID que se utiliza para obtener acceso al resto de métodos disponibles para el servicio Web Favoritos. El código de los otros métodos se escribe igualmente de forma sencilla.