

Capítulo 7

Servicios Windows

Communication Foundation (WCF)

Objetivo

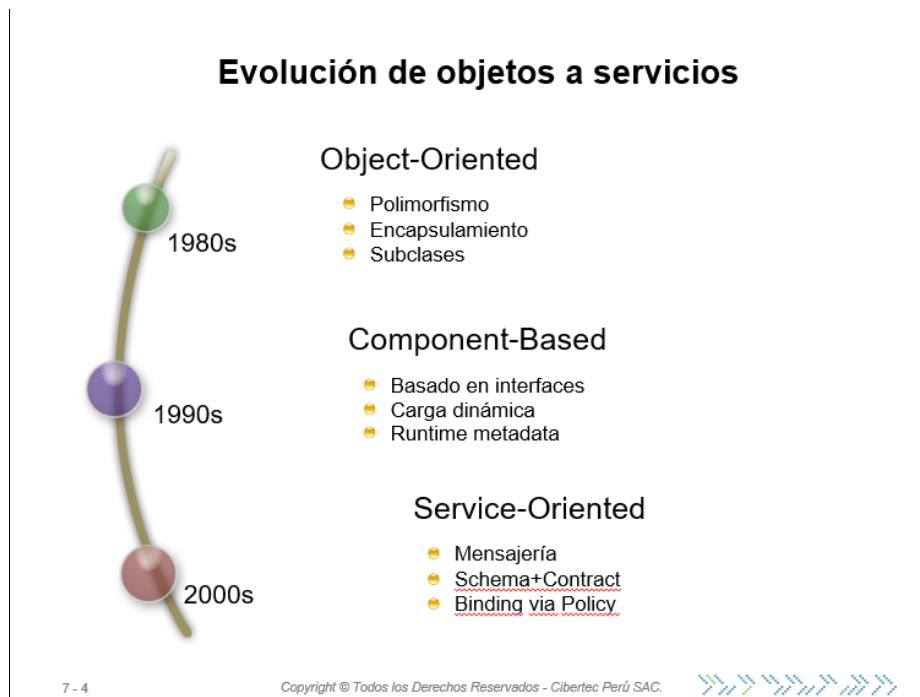
Al finalizar el capítulo, el alumno:

- Implementar servicios WCF en una arquitectura orientada a servicios.
- Definir e implementar contratos de datos y mensajes para servicios WCF.
- Usar múltiples endpoints con patrones de mensajes variados.

Temas

1. Evolución de objetos a servicios
2. ¿Qué es SOA?
3. ¿Qué es Windows Communication Foundation? (WCF)
4. Arquitectura de WCF
5. WCF Contracts
 - Service
 - Data
 - Message
6. Operadores Async/Await

1. Evolución de Objetos a Servicios



Arquitectura es un concepto indispensable entre las aplicaciones hoy en día, se refiere a la combinación e integración de múltiples ambientes o plataformas. [8] La integración y coordinación de procesos puede verse simplificada utilizando Arquitecturas de integración como SOA y Web Services. Desde la aparición de los computadores y el software se ha trabajado en diversos métodos para facilitar y mejorar la producción de código ejecutable y la creación de aplicaciones.

Como resultado se han ido consiguiendo diferentes técnicas y herramientas como ensambladores, compiladores, herramientas gráficas de desarrollo rápido de aplicaciones, herramientas de generación de código a partir de modelos y métodos basados en la aplicación de patrones, entre otros.

Actualmente las técnicas y métodos avanzados de desarrollo de software se basan en el uso de modelos abstractos a partir de los cuales es más sencillo, rápido y económico generar código, como el Desarrollo Dirigido por Modelos, más conocida como el estándar Model Driven Architecture (MDA). Es una especificación que representa una alternativa más para desarrollar software, en este caso a partir de modelos muy abstractos.


2. ¿Qué es SOA?

¿Qué es SOA?

- SOA no se deriva de una propuesta académica
- No hay technical reports de SOA en SEI (Software Engineering Institute)
- Service-oriented architecture fue descrita por primera vez por Gartner en 1996
 - ✓ SSA Research Note SPA-401-068, 12 de abril, “‘Service Oriented’ Architectures, Part 1” y SSA Research Note SPA-401-069, 12 de abril, “‘Service Oriented’ Architectures, Part 2”
- Web Services surgen con mayor fuerza hacia el 2000

7 - 5

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



SOA viene a ser un estilo arquitectónico de software, no es una tecnología determinada. Como es un estilo arquitectural, define a los servicios como unidades de partición para dar una respuesta a los requerimientos del negocio.

SOA proporciona una metodología y un marco de trabajo para mapear procesos de negocio a componentes de software en forma de servicios.

Tecnológicamente, estos servicios tienen la particularidad de ser:

- Encapsulados
- De bajo acoplamiento
- Reutilizables
- Sin información de estado (stateless)
- Localizables
- Independientes de una plataforma o lenguaje
- Basados en estándares y tecnologías ya utilizadas (SOAP, UDDI, WSDL, http, entre otros)

3. ¿Qué es Windows Communication Foundation? (WCF)

¿Qué es Windows Communication Foundation?

- Es un marco de trabajo para la creación de aplicaciones orientadas a servicios.
- Un extremo de servicio puede formar parte de un servicio disponible continuamente hospedado por IIS, o puede ser un servicio hospedado en una aplicación.
- Un extremo puede ser un cliente de un servicio que solicita datos de un extremo de servicio.
- Los mensajes pueden ser tan simples como un carácter o una palabra que se envía como XML, o tan complejos como una secuencia de datos binarios.

7 - 7

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



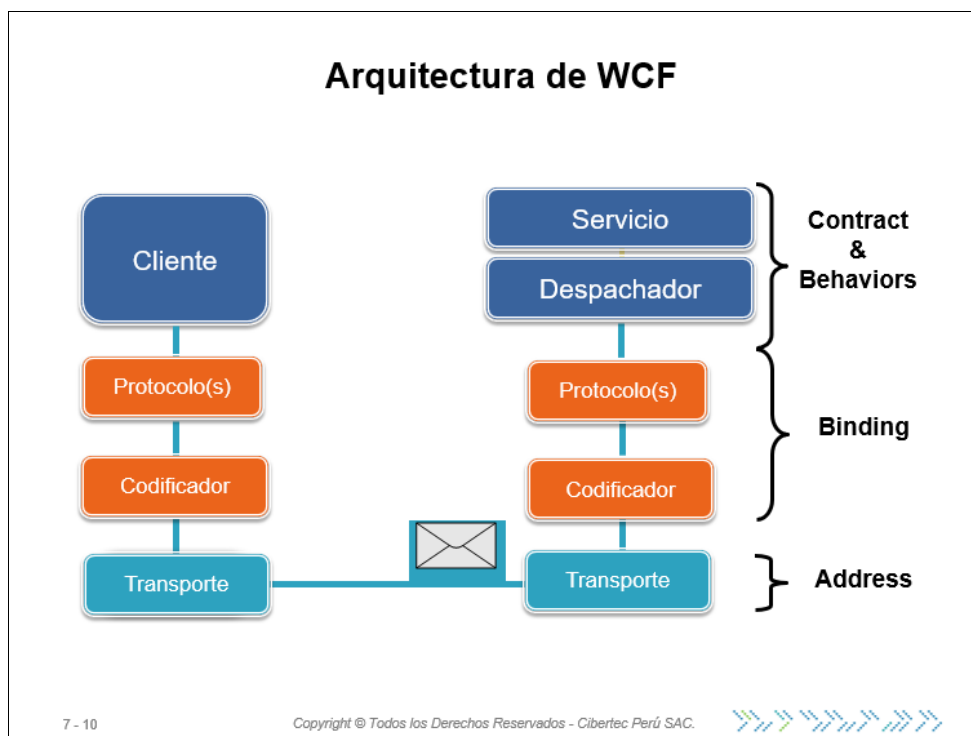
Windows Communication Foundation o WCF es la nueva plataforma de comunicación y mensajería que forma parte del ya conocido entorno de trabajo .NET (en su versión 3.0) La idea fundamental de WCF es la de permitir a las empresas desarrollar aplicaciones distribuidas basadas en la arquitectura orientada a servicios (SOA), donde éstas se puedan ejecutar, desde una máquina local hasta Internet, de manera simple y segura.

La clave fundamental es la sencillez del API de desarrollo y su fácil utilización dentro de la plataforma .NET Framework; permitiendo utilizar cualquier medio de conectividad, ya sea por XML, HTTP SOAP, etc.

El modelo de programación de WCF plantea una unificación de Servicios Web, .NET Remoting, Transacciones Distribuidas y mensajería MQ en un solo marco orientado a servicios distribuidos. Esto no es nada nuevo desde el punto de vista de SOA y de las tecnologías relacionadas, sólo facilita su desarrollo en comparación con otras plataformas que brindan las mismas posibilidades como es el caso de IBM u Oracle. Permitiendo tener ciclos de desarrollo más cortos sin la necesidad de contar con expertos en tecnologías de integración.

Lo que hay que entender de primera mano es que WCF surge dentro del marco de la iniciativa Windows Vista, dejando de lado la posibilidad de desarrollar aplicaciones para otras plataformas, no así la conectividad y la interoperabilidad que continúan siendo independientes de la tecnología y la plataforma gracias a los Servicios Web (SOAP) y demás estándares.

4. Arquitectura de WCF



Contratos y descripciones

Los contratos definen varios aspectos del sistema de mensajes. El contrato de datos describe cada parámetro que constituye cada mensaje que un servicio puede crear o utilizar. Los documentos de Lenguaje de definición de esquemas XML (XSD) definen los parámetros de mensaje, permitiendo a cualquier sistema que entienda XML procesar los documentos. El contrato del mensaje define partes específicas del mensaje utilizando los protocolos SOAP y permite el control más fino sobre las partes del mensaje, cuando la interoperabilidad exige tal precisión. El contrato de servicios especifica las firmas de método actuales del servicio y se distribuye como una interfaz en uno de los lenguajes de programación compatibles, como Visual Basic o Visual C#.

Las directivas y enlaces estipulan las condiciones exigidas para comunicarse con un servicio. Por ejemplo, el enlace debe especificar (como mínimo) el transporte utilizado (por ejemplo, HTTP o TCP) y una codificación. Las directivas incluyen los requisitos de seguridad y otras condiciones que se deben cumplir para comunicarse con un servicio.

Tiempo de ejecución de servicio

La capa del tiempo de ejecución del servicio contiene los comportamientos que solo se producen durante la operación actual del servicio, es decir, los comportamientos en tiempo de ejecución del servicio. La limitación de peticiones controla cuántos mensajes se procesan que puede variar si la demanda para el servicio crece a un límite preestablecido. Un comportamiento de error especifica lo que sucede cuando se produce un error interno en el servicio, por ejemplo, controlando qué información se comunica al cliente. (Demasiada información puede dar ventaja a un usuario malintencionado para

organizar un ataque.) El comportamiento de los metadatos rige cómo y si los metadatos se ponen a disposición del mundo externo. El comportamiento de la instancia especifica cuántas instancias del servicio se pueden ejecutar (por ejemplo, un singleton especifica solo una instancia para procesar todos los mensajes). El comportamiento de la transacción habilita la recuperación de operaciones de transacción si se produce un error. El comportamiento de la expedición es el control de cómo la infraestructura WCF procesa un mensaje.

La extensibilidad habilita la personalización de procesos en tiempo de ejecución. Por ejemplo, la inspección del mensaje es la facilidad para inspeccionar partes de un mensaje y la filtración de parámetros permite que se realicen acciones preestablecidas basándose en filtros que actúan en encabezados del mensaje.

Mensajería

La capa de la mensajería está compuesta por canales. Un canal es un componente que procesa un mensaje de alguna manera, por ejemplo, autenticando un mensaje. Un conjunto de canales también se conoce como una pila de canales. Los canales funcionan en los mensajes y encabezados del mensaje. Esto es diferente de la capa en tiempo de ejecución del servicio, que se ocupa principalmente de procesar el contenido de los cuerpos de los mensajes.

Hay dos tipos de canales: canales de transporte y canales de protocolo.

Los canales de transporte leen y escriben mensajes de la red (o algún otro punto de la comunicación con el mundo externo). Algunos transportes utilizan un codificador para convertir los mensajes (que se representan como conjuntos de información XMLs) hacia y desde la representación de la secuencia de bytes utilizada por la red. Son ejemplos de transportes HTTP, canalizaciones con nombre, TCP y MSMQ. Son ejemplos de codificaciones XML y binario optimizado.

Los canales de protocolo implementan protocolos de procesamiento de mensajes, a menudo leyendo o escribiendo encabezados adicionales en el mensaje. Los ejemplos de tales protocolos incluyen WS-Security y WS-Reliability.

La capa de la mensajería muestra los posibles formatos y modelos de intercambio de los datos. WS-Security es una implementación de la especificación WS-Security que habilita la seguridad en la capa del mensaje. El canal de mensajería WS-Reliable habilita la garantía de entrega del mensaje. Los codificadores presentan una variedad de codificaciones que se pueden utilizar para satisfacer las necesidades del mensaje. El canal HTTP especifica que el Protocolo de transporte de hipertexto se utiliza para la entrega del mensaje. El canal TCP especifica de manera similar el protocolo TCP. El canal de flujo de transacciones rige los modelos de mensajes de transacción. El canal de la canalización con nombre habilita la comunicación entre procesos. El canal de MSMQ habilita la interoperación con aplicaciones MSMQ.

5. WCF Contracts



Service Contract


- Define lo que puede hacer un servicio.
- Permite la interoperabilidad de un servicio.

Un contrato no es más ni menos que un acuerdo entre las partes involucradas, que define la manera en que se va a trabajar y con qué elementos. En este caso se especificará qué funciones se van a ofrecer, qué tipos de datos se van a recibir o enviar, y cómo van a ser enviados.

Data Contract: describe las estructuras de datos manejadas por las operaciones del servicio.

Message Contract: define el contenido de un mensaje, ya sea en la cabecera o en el cuerpo.

6. Operadores Async/Await




Operaciones asíncronas

- Orientada a aprovechar (principalmente en aplicaciones del lado cliente) los nuevos modelos de microprocesadores con múltiples núcleos.
- Este tipo de programación era relativamente compleja y diferente del modelo síncrono de programación (el modelo más habitual) de ahí que, no fuera tan habitual.
- La programación asíncrona es una necesidad debido a que tendremos algunos métodos que invocarlos a nuestras aplicaciones.

7 - 16

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



ASYNC Async es un modificador que se va a aplicar a aquellos métodos que van a contener las acciones a realizar de manera asíncrona.

El hecho de que un método sea cualificado con ASYNC no indica que se vaya a ejecutar asíncronamente, para ello necesita la segunda palabra clave.

AWAIT Await es un modificador de instrucción que va a indicar que dicha acción se tiene que realizar de manera asíncrona, por lo tanto, no se tendrán en cuenta las siguientes acciones hasta que dicha tarea haya finalizado. Cabe destacar que no bloquea el resto de acciones, sino que es como si definiera un CALLBACK para las líneas de código siguientes a AWAIT, y en el momento de finalizar el proceso asíncrono ejecutará dicho CALLBACK.

Cuando se habla de multitarea se puede referir a dos conceptos diferentes que hay que tener bien claros.

Multitarea de CPU o multithreading, que consiste en particional una tarea para aprovechar los diferentes núcleos que dispone la máquina. Este tipo de multitarea no es la que vamos a tratar en este artículo, y no es recomendable utilizarla en una máquina dedicada a dar servicio a traves de la red.

Multitarea asincronica IO, que consiste en no bloquear el procesador mientras se realizan tareas de entrada y salida de datos. Un tiempo en el que nuestra máquina no hace nada. Los casos típicos de estos bloqueos son el acceso a disco, consultas a bases de datos, comunicación con otras máquinas.

Otra forma muy sencilla y elegante de que se introdujo en el Framework 4.5 es mediante programación asíncrona con `async/await`. Esto nos permite liberar el hilo del pool para hacer ciertas tareas de entrada y salida y luego recuperarlo de una forma muy sencilla.

Basta con definir nuestro método de servicio como `async` y devolver un objeto `Task`. Exactamente igual que se definen cualquier otro tipo de métodos asíncronos.

```
[ServiceContract]
public interface IOperacioens
{
    [OperationContract]
    Task AddObjet(object x);

    [OperationContract]
    Task<int> CountObjets();}

public class OperacioensNetService : IOperacioen
{
    public async Task AddObjet(object x);
    {
    }

    public async Task<int> CountObjets()
    {
        ....
        return number;
    }
}
```

De esta forma tan simple tendremos creado nuestro método asíncrono para poder ser consumido remotamente.