

SERVICIO WEB DEL SISTEMA DE CONTROL DE TECNOLOGÍAS INFORMÁTICA Y COMUNICACIONES

**Trabajo para optar por el Título de Ingeniero en Ciencias
Informáticas.**

Autor: Humberto Salazar Pacios

Tutor: Cp. Ing. Frank David Garcés Presa

Cotutor: Ing. Alberto E. Ruiz Romero

AGRADECIMIENTOS

A mis padres, Nereida y Humberto, que han sido el ejemplo que he seguido siempre, quienes me dieron la vida y me han apoyado siempre en mis decisiones.

A mi hermano Raúl que fue mi primer amigo en la vida y quien me mostró el camino de la informática y su esposa Lisbet que también ha sido como una hermana para mí.

A los otros cuatro del equipo de Primavera (Jorge, Jose, Fabián y Eric) porque si faltara uno no hubiéramos podido hacer ni la mitad de lo que hemos hecho.

A Lisandra, mi hermanita en la UCI, por cargar siempre conmigo, por la confianza que me ha brindado, por sus consejos, por su sinceridad y por ser capaz de resistir las palabras más duras que puedan salir de mi boca.

A los profes: Lombillo, Dosagües y Renier porque cada uno puso su esfuerzo en función de mi formación y fueron el mejor apoyo para salir adelante.

A los compañeros del grupo 3, porque fueron mi familia en la escuela.

A la gente de MININT que siempre estuvo apoyándome, interesándose por todo y regañándome cuando fue necesario.

A los muchachos de conTics (Liu, Daliana, Reinier, Yerandy, Losada, Luis) por todas las madrugadas que dedicamos en las vacaciones para hacer el sistema y especialmente a mi tutor Frank porque a pesar de estar lejos de su casa trabajó más que todos nosotros juntos y no permitió que nos rindiéramos.

A todos los integrantes de la lista infinita de los que pusieron su granito de arena.

A todos, muchas gracias.

RESUMEN

El presente trabajo documenta todo el proceso de investigación y desarrollo seguido para la implementación de la capa de Servicios Web del Sistema de Control de Tecnologías de Informática y Comunicaciones (conTics).

En el mismo se hace un estudio sobre el estado actual del sistema y la necesidad de expandir sus funcionalidades con vistas al posible surgimiento de nuevas aplicaciones que, para el funcionamiento de algunos de sus procesos internos, puedan necesitar la información contenida en la base de datos del Sistema conTics.

INDICE

INTRODUCCIÓN	1
CAPÍTULO 1	4
Fundamentación teórica	4
Introducción.....	5
1.1 Conceptos generales asociados.....	6
1.1.1 Aplicación Informática.....	6
1.1.2 Servicio.....	7
1.1.3 Servicio Web.....	7
1.1.4 Solución distribuida	8
1.1.5 Base de datos	9
1.1.6 Sistema de Gestión de Base de Datos (SGBD).....	10
1.1.7 Extensible Mark-up Language (XML) – Lenguaje Extensible de Marcas.....	10
1.2 Tendencias y tecnologías	11
1.3 Tecnología a utilizar	12
1.3.1 Microsoft .NET Framework 3.5.....	13
1.3.2 Windows Communication Foundation.	13
1.3.3 Lenguaje de Implementación C#.....	13
1.3.4 Servidor Web. Internet Information Services (ISS)	13
1.3.5 Sistema de Gestión de Base de Datos Relacional. ORACLE	14
1.4 Entorno de Desarrollo Integrado. Microsoft Visual Studio Team System 2008	15
1.5 Metodologías de desarrollo.....	16
1.5.1 Rational Unified Process (RUP)	16
1.5.2 Programación Extrema (XP).....	17

1.5.3	Web Service Implementation Methodology (WSIM) – XP	18
1.6	Selección de la metodología de desarrollo a utilizar	19
	Conclusiones del capítulo	20
CAPÍTULO 2.....	21	
Características de la solución propuesta.....	21	
Fase de requerimientos y análisis	21	
	Introducción.....	22
2.1	Descripción del entorno donde trabajará el sistema.....	23
2.2	Propuesta del sistema.....	24
2.2.1	Términos comunes asociados al entorno.....	24
2.2.2	Eventos principales del entorno	25
2.3	Descripción del sistema propuesto.....	25
2.3.1	Descripción del personal relacionado con el sistema	25
2.3.2	Listado de requerimientos del producto	26
2.4	Historias de usuario	27
2.4.1	Sobre las Historias de usuario.....	33
2.5	Planificación	33
2.6	Plataforma tecnológica como marco de trabajo para la implementación.	34
2.7	Propuesta de especificación de la arquitectura de software.....	35
2.7.1	Definición de la granularidad de los Servicios Web.....	35
2.7.2	Servicios Web reutilizables	36
	Conclusiones del capítulo	37
CAPÍTULO 3.....	38	
Construcción de la solución propuesta.....	38	

Fase de diseño, implementación y pruebas	38
Introducción.....	39
3.1 Diseño del sistema.....	40
3.1.1 Principios de la orientación a servicios	40
3.1.2 Contratos de un servicio.....	41
3.1.3 Librerías de servicios.....	46
3.1.4 Acceso a la base de datos.....	46
3.1.5 Tarjetas CRC	46
3.1.6 Esquema del sistema.....	52
3.2 Implementación del sistema	54
3.3 Pruebas del sistema.....	55
Conclusiones del capítulo	61
CONCLUSIONES	62
RECOMENDACIONES	63
REFERENCIAS BIBLIOGRÁFICAS	64

INTRODUCCIÓN

El Ministerio del Interior de la República de Cuba (MININT), como institución, maneja una amplia y variada gama de medios tecnológicos necesarios para el desarrollo de las tareas investigativas y operativas que forman parte de su hacer cotidiano.

En los últimos años, como parte de la Tarea 20x50 (Veinte Tareas de Informatización por el 50 Aniversario del Triunfo de la Revolución), se han ido modernizando la mayor parte de éstos y se han adquirido un gran número de nuevos medios, con el objetivo de informatizar y automatizar en mayor medida los procesos dentro de la institución para optimizarlos.

Es de vital importancia para el Ministerio del Interior tener un estricto control sobre la existencia, estado y uso de estos medios, por lo que se decidió, con la colaboración de los Cadetes Insertados del Ministerio del Interior en la Universidad de las Ciencias Informáticas (UCI) desarrollar una aplicación para gestionar toda la información referente a los mismos.

A raíz de lo anteriormente expresado surge el Sistema de Control de Tecnologías de Informática y Comunicaciones (conTics), el cual estaría basado en una aplicación Web con una base de datos que contendría toda la información referente a la ubicación, asignación, uso, estado técnico y datos técnicos de cada uno de los medios y servicios tecnológicos en posesión del Ministerio del Interior.

El Sistema conTics brinda todas las funcionalidades necesarias para la gestión de toda esta información, pero al ser una aplicación Web, el acceso a la información gestionada por la misma solamente es posible utilizando un navegador web instalado en una estación de trabajo (computadora) cliente. Dadas las características del Ministerio del Interior, el gran volumen y diversidad de la información contenida en la base de datos de la aplicación, así como la posible necesidad futura de que otras aplicaciones se nutran de esta, el modelo actual para el acceso a este sistema es limitado.

Por tanto, **el problema a resolver** con esta investigación está en **extender las posibilidades de acceso a la información gestionada por el Sistema conTics.**

Se toma como **idea a defender** que la implementación de una capa de Servicios Web para la aplicación conTics expandirá sus capacidades, haciendo posible el acceso a la información y a las funcionalidades

de la misma desde otros sistemas que requieran como parte de su funcionamiento la gestión total o parcial de un determinado conjunto de medios técnicos.

Posterior al análisis de esta situación se hace imprescindible la búsqueda de una solución a la misma teniendo como **objeto de estudio** de la investigación el desarrollo e implantación de Servicios Web en la intranet del Ministerio del Interior, tomando como **campo de acción** el proceso de gestión de la información referente a los medios técnicos al servicio del Ministerio del Interior.

Para lograr una solución al problema planteado el **objetivo general** de este trabajo investigativo queda definido de la siguiente manera: Desarrollar un Servicio Web que permita centralizar, estandarizar y ofrecer información detallada de las Tics que se explotan en el Ministerio del Interior.

Para cumplir el objetivo general y darle solución a la situación problemática planteada, se proponen las **tareas investigativas** siguientes:

- ✓ Analizar y estudiar el funcionamiento de los Servicios Web.
- ✓ Hacer un estudio sobre las soluciones existentes.
- ✓ Investigar sobre la aplicación de una metodología de desarrollo para la implementación de Servicios Web.
- ✓ Investigar sobre las tecnologías disponibles para el desarrollo y despliegue de Servicios Web.
- ✓ Hacer un estudio sobre las posibles aplicaciones que pudieran beneficiarse con el consumo del Servicio Web de conTics.
- ✓ Hacer un levantamiento de las funcionalidades a implementar como Servicios.
- ✓ Diseñar e implementar todos los Servicios propuestos utilizando metodología seleccionada.

La presente investigación está estructurada por 4 capítulos cuya descripción será expuesta brevemente a continuación.

Capítulo 1: Fundamentación Teórica. Incluye un conjunto de elementos teóricos que sustentan la investigación, así como un estudio del estado del arte y una descripción de las soluciones existentes. Además se hace un profundo análisis de las tendencias y tecnologías actuales, con el objetivo de seleccionar los componentes y herramientas a utilizar para la solución del problema planteado.

Capítulo 2: Características de la Solución Propuesta. (Fase de **Requerimientos** y **Análisis**) En este capítulo se definen y describen las características de la solución propuesta. Se hace un estudio sobre las características del entorno donde trabajará el sistema así como del flujo actual de procesos involucrados en el campo de acción y de la forma de ejecución de los mismos, explicando las causas que originan la situación problemática y las consecuencias de la misma.

Se hace un análisis comparativo entre soluciones actuales y la solución propuesta indicando las ventajas de la misma.

Se definen además el listado de requerimientos del sistema así como las historias de usuario propuestas por el cliente en base a sus necesidades y se define el plan de iteraciones.

Capítulo 3: Construcción de la solución propuesta. (Fase de **Diseño**, **Implementación** y **Pruebas**). En el presente capítulo se incluyen temas relacionados con los principios de la orientación a servicios, algunos conceptos fundamentales para la comprensión de la estructura del sistema así como las Pruebas de Aceptación que garantizan que cada Historia de Usuario fue implementada correctamente.

CAPÍTULO 1

Fundamentación teórica

Introducción.

Este capítulo contiene un estudio del arte sobre la implementación y explotación de Servicios Web en la Universidad de las Ciencias Informáticas, el Ministerio del Interior, Cuba y el resto del mundo. Además se exponen de manera concreta un grupo de definiciones de términos, necesarias para la total comprensión de la investigación así como un estudio comparativo sobre las plataformas y herramientas más utilizadas para desarrollo y despliegue de Servicios Web.

1.1 Conceptos generales asociados.

1.1.1 Aplicación Informática

Componente lógico de un computador digital que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido.

Según el propósito con el que haya sido diseñada, una aplicación informática puede ser clasificada en varias categorías, tales como: procesadores de texto, editores de imágenes, reproductores de multimedia, gestores de bases de datos, etc.

En dependencia de su entorno de ejecución las aplicaciones informáticas se pueden dividir en dos grandes grupos:

1.1.1.1 Aplicaciones de Escritorio

Son aquellas que se instalan en el sistema operativo del usuario y almacenan sus datos en ficheros locales.

Su difusión es más lenta ya que a la salida de una nueva versión es necesario que el usuario la instale en el sistema o actualice la ya existente.

Dependen en gran medida de la plataforma (Sistema Operativo) del usuario y es relativamente complejo portarlas a plataformas diferentes.

Su capacidad de respuesta es muy rápida y posibilitan un alto nivel de interactividad.

1.1.1.2 Aplicaciones Web

Son aquellas que se encuentran publicadas en servidores, el acceso a las mismas se hace a través de un navegador Web (Web Browser) y dependen de este último para su visualización. Generalmente almacenan su información en servidores de bases de datos.

Son de rápida difusión y posibilitan que un gran número de usuarios interactúe directamente con la misma. Pueden ser usadas por cualquier cliente independientemente de la plataforma (Sistema Operativo) del mismo.

Una de sus desventajas es la poca capacidad de interactividad que permiten así como los prolongados tiempos de respuesta. Aunque en los últimos tiempos con la explotación de tecnologías como AJAX¹ y Flash prácticamente ha sido eliminada esta desventaja.

1.1.2 Servicio

Un servicio es un recurso abstracto que representa la capacidad de realizar tareas que forman una funcionalidad coherente desde el punto de vista de entidades proveedoras y entidades solicitantes. (1)

Según Lamb, Hair y McDaniel, "un servicio es el resultado de la aplicación de esfuerzos humanos o mecánicos a personas u objetos. Los servicios se refieren a un hecho, un desempeño o un esfuerzo que no es posible poseer físicamente. (2)

O sea, un servicio es una acción o un conjunto de acciones dirigidas a satisfacer alguna necesidad, resultando en un bien intangible.

Según se define en el glosario de términos de la ITIL² del CMPP³, "un servicio informático es aquel servicio que se suministra a través de un sistema de Tecnología de la Información". (3)

1.1.3 Servicio Web

Un Servicio Web es un sistema de software diseñado para soportar interacción máquina a máquina sobre una red. Posee una interfaz descrita en un formato compresible para las máquinas (específicamente WDSL⁴). (1)

Los Servicios Web (Web Service) permiten a las aplicaciones compartir información e incluso invocar funciones de otras aplicaciones independientes, sin importar el sistema operativo, plataforma en la que se ejecuten o incluso el dispositivo que se utilice para acceder a ellos. Los Servicios Web aunque son independientes pueden tener comunicación entre ellos y formar grupos de colaboración para poder realizar una tarea específica. (4)

Las soluciones orientadas a servicios se caracterizan esencialmente por:

¹ Asynchronous JavaScript And XML

² Information Technology Infrastructure Library

³ Center for Maximum Public Performance

⁴ Web Service Description Language

- ✓ **Los servicios son aislados.** Cambios internos en la implementación de un servicio no implica cambiar, reiniciar o rehacer otros servicios.
- ✓ **Los servicios son independientes de ubicación.** Un servicio es accedido de la misma forma, así se encuentre en la misma PC o en una ubicación remota.
- ✓ **Los servicios son neutrales en cuanto a protocolo, transporte y formato.** Los detalles de comunicación entre dos partes son flexibles y deben poder modificarse o adicionarse sin complicaciones en el servicio.
- ✓ **Los servicios son independientes de plataforma e implementación.** Un servicio no tiene por qué conocer detalles sobre la plataforma de ejecución de otro servicio para poder interactuar con él. Los esquemas de contrato intercambiados serán suficientes. (5)

En resumen, son aplicaciones que permiten el intercambio de datos en formato XML⁵. Estos datos pueden ser consumidos por cualquier aplicación, independientemente de las tecnologías usadas tanto para la implementación del servicio como de la aplicación cliente, esto los hace multiplataforma y muy portables. Constituyen la base para la creación de soluciones distribuidas.

1.1.4 Solución distribuida

Hoy en día es casi imposible pensar en una aplicación que carezca de comunicación con otras, por lo que se impone llevar la comunicación a un plano significativo en el modelo de programación.

En una solución orientada a servicios se intercambian mensajes en lugar de invocar métodos y pasarse objetos. Esto permite que ambos puntos de la comunicación sean implementados en plataformas diferentes entre sí. En la práctica para crear soluciones distribuidas los desarrolladores se basan en servicios orientados a mensajes e implementados aplicando los principios de programación orientada a objetos dando lugar a una combinación perfecta.

Una solución distribuida, para su funcionamiento, está compuesta fundamentalmente por dos tipos de programas que, según el papel que desempeñen en la solución, se pueden catalogar en:

⁵ Extended Markup Language

- ✓ **Clientes.** Programas que inician mensajes consumiendo servicios. Un cliente envía un mensaje a un servicio para lograr ejecutar alguna funcionalidad. Una vez que el servicio recibe el mensaje, se lleva a cabo alguna acción. Un cliente puede esperar o no por la respuesta de un servicio. (5)
- ✓ **Servicios.** Programa que reacciona ante mensajes, es decir, recibe mensajes de los clientes y entonces realiza alguna acción dependiendo de la información recibida. Estas acciones pueden implicar más intercambio de mensajes para responder al cliente que envió el mensaje inicial. (5)

Los mensajes son la base de la comunicación de las soluciones distribuidas. Estos mensajes tienen una estructura compuesta por dos elementos fundamentales:

- ✓ **Cabecera:** contiene un conjunto de elementos denominados encabezados cuya información puede ser utilizada por la infraestructura de comunicación y los desarrolladores.
- ✓ **Cuerpo:** contiene la información a transportar y es consumido por los clientes o servicios creados por un desarrollador.

1.1.5 Base de datos

Se define una Base de Datos (BD) como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados para su posterior uso, en el cual se acceden a ellos mediante un conjunto de programas que manipulan esta información.

Cada base de datos se compone de una o más tablas que guardan un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queremos guardar en la tabla, cada fila de la tabla conforma un registro. (6)

Entre las principales características de los sistemas de base de datos podemos mencionar:

- ✓ Independencia lógica y física de los datos.
- ✓ Redundancia mínima.

- ✓ Acceso concurrente por parte de múltiples usuarios.
- ✓ Integridad de los datos.
- ✓ Consultas complejas optimizadas.
- ✓ Seguridad de acceso y auditoría.
- ✓ Respaldo y recuperación.
- ✓ Acceso a través de lenguajes de programación estándar.

1.1.6 Sistema de Gestión de Base de Datos (SGBD)

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la BD, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. (6)

1.1.7 Extensible Mark-up Language (XML) – Lenguaje Extensible de Marcas

XML es un lenguaje de etiquetas o tags (similar en sintaxis a HTML) que permite escribir documentos de texto que expresan datos y no aspecto visual. No impone un conjunto de etiquetas, sino sólo unas pocas normas de uso. Se puede decir que XML es la base para toda la tecnología entorno al desarrollo y despliegue de Servicios Web. Ejemplos de aplicación en esta rama son:

- ✓ SOAP (Simple Object Access Protocol)
- ✓ WSDL
- ✓ UDDI

Se puede utilizar para describir la estructura de los datos y sus campos de aplicación son innumerables, entre los más destacados se encuentran:

- ✓ Integración de aplicaciones heterogéneas
- ✓ Configuración de aplicaciones

- ✓ Bases de datos.

1.2 Tendencias y tecnologías

En el campo de implementación y despliegue de Servicios Web se destacan como fundamentales un conjunto de tecnologías que colaboran dando como resultado el funcionamiento de un Servicio Web.

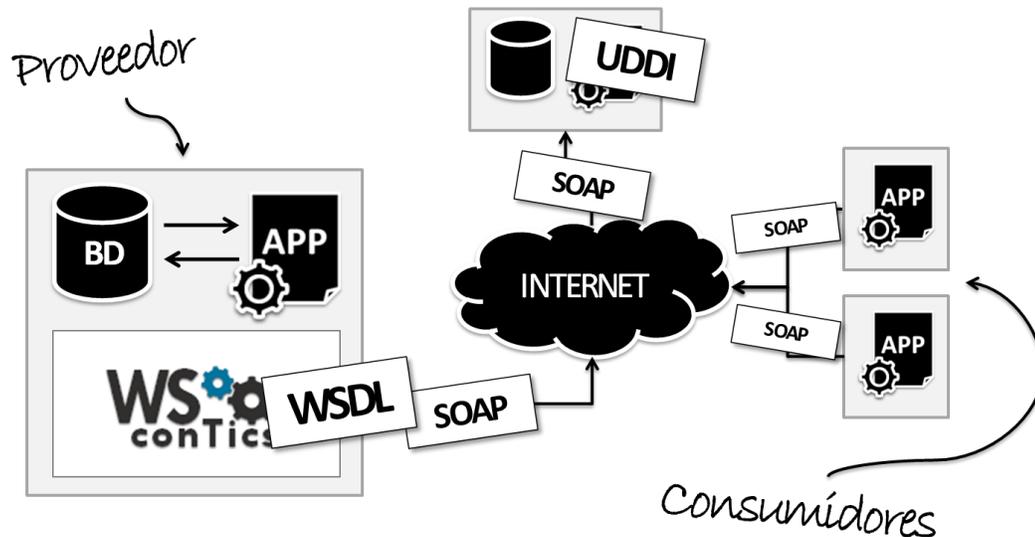


Ilustración 1: Esquema de Arquitectura de Servicios Web

Como se muestra en la Ilustración 1, todos los elementos que interactúan a través de la red en la arquitectura de Servicios Web se comunican empleando **SOAP** (Protocolo basado en XML) para envío de mensajes empleando comunicación síncrona y asíncrona.

Por su parte cada Servicio Web debe representar una especie de “contrato” entre el proveedor y el consumidor del servicio. Este contrato se logra utilizando **WSDL** (Web Services Description Language) para especificar en XML las operaciones y tipos de datos del Servicio Web.

Finalmente toda la información referente al Servicio Web es registrada en un catálogo que proporciona operaciones de registro y búsqueda de los mismos. Este catálogo es conocido como **UDDI** (Universal Description, Discovery and Integration of Web Services).

Estas tecnologías para su funcionamiento se apoyan en:

- ✓ **Sistemas de Gestión de Base de Datos** para almacenar la información.
- ✓ **Lenguajes de programación** para la lógica de funcionamiento del Servicio Web
- ✓ **Servidor Web** para lograr la disponibilidad del Servicio Web sobre el protocolo estándar HTTP.

Las tendencias actuales para el caso de los servidores y lenguajes de programación se pueden clasificar en tres grupos.

- ✓ **Alternativa libre** (Servidor Apache + PHP)
- ✓ **Alternativa Microsoft** (Servidor Internet Information Services + ASP.NET)
- ✓ **Alternativa Java** (Servidor Apache Tomcat + Java)

Para el caso de los Sistemas de Gestión de Bases de Datos es posible encontrarlos combinados con cualquiera de los tres grupos, los principales exponentes en el mercado actual son:

- ✓ PostgreSQL
- ✓ Oracle
- ✓ Microsoft SQL Server

1.3 Tecnología a utilizar

Luego del análisis sobre las tendencias y tecnologías más utilizadas y probadas para el desarrollo e implementación de Servicios Web a niveles profesionales se procede a hacer una selección de la tecnología a utilizar para darle cumplimiento al objetivo general de la investigación.

Producto de la necesidad de integración del resultado final de la investigación con el sistema conTics y con la finalidad de reutilizar componentes del mismo para lograr de manera óptima la extensión de sus funcionalidades a través de la capa de Servicios Web se ha decidido emplear tecnología 100% compatible con la utilizada en la implementación y despliegue del mismo.

1.3.1 Microsoft .NET Framework 3.5

Este marco de trabajo permite la creación rápida de aplicaciones conectadas que ofrecen experiencias de usuario fuera de lo común al proporcionar bloques de creación (software prefabricado) para resolver tareas de programación comunes. Las aplicaciones conectadas creadas modelan los procesos empresariales de forma eficaz y facilitan la integración de sistemas en entornos heterogéneos.

Al combinar Visual Studio y .NET Framework, se reduce la necesidad de código común irrelevante, lo que acorta el tiempo de desarrollo y permite que los desarrolladores se concentren en la solución de los problemas de la empresa.

En la versión 3.5 de este producto se han realizado mejoras en áreas de características, como la biblioteca de clases base, Windows Workflow Foundation, Windows Communication Foundation, Windows Presentation Foundation y Windows CardSpace. (7)

1.3.2 Windows Communication Foundation.

Windows Communication Foundation (WCF) es la nueva plataforma de Microsoft para la creación de servicios distribuidos. En esencia su avanzada infraestructura y modelo de programación permite crear, configurar y distribuir servicios en la red. (5)

1.3.3 Lenguaje de Implementación C#

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi).

1.3.4 Servidor Web. Internet Information Services (ISS)

Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son:

- ✓ **FTP:** File Transfer Protocol - Protocolo Transferencias Archivos

- ✓ **SMTP:** Simple Mail Transfer Protocol - Protocolo Simple de Transferencia de Correo
- ✓ **NNTP:** Network News Transport Protocol - Protocolo para la Transferencia de Noticias en Red
- ✓ **HTTP/HTTPS:** Hypertext Transfer Protocol - Protocolo de Transferencia de Hipertexto

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET

1.3.5 Sistema de Gestión de Base de Datos Relacional. ORACLE

Oracle es un Sistema de Gestión de Base de Datos, destinado fundamentalmente a servidores empresariales, mercado que domina casi totalmente debido a sus reconocidas prestaciones y al ser considerado uno de los sistemas de gestión de bases de datos más completos.

Entre sus características principales, resaltan las siguientes:

- ✓ **Soporte de transacciones:** Capacidad de ejecutar un conjunto de órdenes formando una unidad de trabajo, es decir, en forma indivisible o atómica.
- ✓ **Escalabilidad:** Habilidad para manejar el crecimiento continuo de trabajo y estar preparado para hacerse más grande sin afectar su funcionamiento.
- ✓ **Estabilidad:** Posee una muy reducida tasa de fallos en su funcionamiento.
- ✓ **Soporte multiplataforma:** Compatibilidad con varias plataformas (Sistemas Operativos) explotadas actualmente a nivel mundial (Microsoft Windows, Linux, Unix).
- ✓ **Soporte de Triggers (Disparadores):** es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).
- ✓ **Soporte de Procedimientos Almacenados:** Programa (o procedimiento) el cual es almacenado físicamente en una base de datos. Al ser ejecutados directamente en el motor de bases de datos, posee acceso directo a los datos que necesita manipular y solo necesita enviar sus resultados de

regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes.

- ✓ **Soporte de Índices:** Estructura de datos que mejora la velocidad de las operaciones, permitiendo un rápido acceso a los registros de una tabla.

Básicamente es una herramienta cliente/servidor para la gestión de bases de datos. Posee escalabilidad para soportar un gran número de usuarios y cargas de trabajo de alto volumen de transacciones, el optimizador de consultas determina dinámicamente los caminos de acceso y métodos de enlace más eficientes para cada consulta. Además soporta efectivamente ambientes mixtos de carga de trabajo caracterizados por actividades simultáneas de consulta y actualización. (8)

Posee lectura de multiversión sin bloqueo, siempre proporciona a los usuarios resultados consistentes, al mismo tiempo que no penaliza al rendimiento en la actividad concurrente de actualización. También garantiza la autenticidad apropiada de los usuarios y la privacidad e integridad de los datos, y permiten manejar la asignación de privilegios y monitorear las operaciones de la base de datos a lo largo de toda la empresa. (8)

1.4 Entorno de Desarrollo Integrado. Microsoft Visual Studio Team System 2008

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Ofrece herramientas de desarrollo avanzadas, características de depuración, funcionalidad de base de datos y características innovadoras para crear rápidamente las aplicaciones de vanguardia del futuro en una gran variedad de plataformas. (7)

Incluye mejoras, como los diseñadores visuales, para el desarrollo más rápido con .NET Framework 3.5, mejoras sustanciales de las herramientas de desarrollo web y mejoras del lenguaje que aceleran el desarrollo con todo tipo de datos. Visual Studio 2008 ofrece a los desarrolladores toda la compatibilidad necesaria con herramientas y marcos para crear aplicaciones web atractivas, expresivas y compatibles con AJAX. (7)

1.5 Metodologías de desarrollo

1.5.1 Rational Unified Process (RUP)

Es la metodología de desarrollo más popular y una de las más utilizadas a nivel mundial en grandes proyectos por su robustez. Muchas metodologías de surgimiento posterior son derivadas o adaptaciones de esta metodología tradicional.

RUP es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. Permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Cuenta con las mejoras prácticas del modelo de desarrollo de un software en particular. (9)

- ✓ Desarrollo de software de forma iterativa.
- ✓ Manejo de requerimientos.
- ✓ Utiliza arquitectura basada en componentes.
- ✓ Modela el software de forma visual, usando UML.
- ✓ Verifica la calidad del software.
- ✓ Controla los cambios.
- ✓ Dirige las tareas de cada desarrollador por separado y del equipo como un todo.
- ✓ Especifica los artefactos que deben desarrollarse en cada fase de desarrollo del software.

Entre las principales características que identifica a RUP, es posible enumerar las siguientes:

- ✓ **Guiado por Casos de uso:** Los cuales sirven para describir el comportamiento del sistema y elaborar los casos de prueba con los que se comprueba que el sistema desarrollado hace lo que el cliente quiere.

- ✓ **Centrado en la arquitectura:** La arquitectura del sistema es la columna vertebral de todo el desarrollo del mismo. Cada iteración gira en torno a la misma, fortaleciendo y corrigiendo sus características.
- ✓ **Iterativo e incremental:** En cada ciclo de iteración se produce una nueva versión del software.

Utiliza UML como lenguaje de modelado y cuenta con varias fases de trabajo en las cuales se desarrolla una serie de flujos fundamentales del desarrollo del proyecto. (10)

Una de las principales deficiencias que cuenta RUP es que constituye una metodología pesada en la cual se genera documentación en exceso, además se considera una metodología aplicable a proyectos de larga duración. Además debe tenerse en cuenta que es muy importante respetar la planificación que se realiza en cada una de sus fases y flujos de trabajos, lo que implica que al aparecer un cambio inmediato se realice una solicitud de cambio para realizar el mismo y así respetar la planificación realizada.

1.5.2 Programación Extrema (XP)

La más popular entre las llamadas “metodologías ágiles”. Debe su popularidad entre los desarrolladores a la escasa documentación que requiere.

A continuación se enumeran sus características más distintivas:

- ✓ **Retroalimentación con el cliente:** Conceptualmente, al menos uno de los miembros del equipo de trabajo del proyecto, es un cliente. Esto propicia una constante interacción del mismo con el producto en desarrollo.
- ✓ **Cortas iteraciones:** En cada iteración, se obtiene un producto listo para entregar y que tiene valor para el cliente. La entrega continua de resultados compromete a ambas partes en la evolución del proyecto e indirectamente influye de forma positiva en la calidad del producto final.
- ✓ **Muy flexible a cambios:** Una de las características más reconocidas de XP. La constante retroalimentación con los clientes permite prever futuros cambios y evita llegar a momentos que paralizan el desarrollo del producto. (11)

Cuenta con una serie de prácticas destinadas a aumentar la productividad del equipo de trabajo como la programación en pares, reuniones diarias y planes de entrega acorto plazo, por mencionar algunos. Al igual que RUP, también utiliza UML como lenguaje de modelado.

XP es una aplicable a proyecto de corta duración, en el cual efectuar un cambio es más importante que respetar la planificación realizada. Una de las principales ventajas que con las que cuenta XP que la hace superior a RUP en este trabajo investigativo es que cuenta con una Metodología de Implementación de Web Service llamada WSIN, que facilita la construcción de los mismos.

1.5.3 Web Service Implementation Methodology (WSIM) – XP

Es el resultado de la aplicación práctica del “**OASIS Framework for Web Services Implementation (FWSI)**” a la metodología XP.

El FWSI tiene como objetivo facilitar la implementación de Servicios Web robustos a partir de la definición de una metodología práctica y extensible basada en un proceso de implementación y elementos funcionales comunes que los desarrolladores puedan adoptar para crear sistemas de Servicios Web de alta calidad sin re-inventarlos para cada implementación. (12)

Adicionalmente los Servicios Web deben ser desplegados rápidamente y se necesita que se integren en armonía con los nuevos colaboradores y clientes del negocio.

En el desarrollo de los mismos se necesita que el cliente asigne prioridades, defina el dominio del problema y tome decisiones claves, porque es el cliente quien comprende completamente el proceso que el software está intentando emular. (13)

Básicamente se hace una adaptación a la metodología XP para ajustarla al proceso de desarrollo, implementación y despliegue de Servicios Web.

La metodología XP define las siguientes fases:

- ✓ Planificación.
- ✓ Diseño.
- ✓ Codificación y Pruebas.

Estas fases también se aplican al proceso de implementación y desarrollo de Servicios Web.

Fases de XP	Fases de WSIM - XP
Planificación	Análisis y Requerimientos
Diseño	Diseño
Codificación	Codificación
Pruebas	Pruebas
Entrega	Despliegue

Tabla 1: Asociación entre fases definidas por XP y WSIM (13)

Los Servicios Web, a diferencia del software tradicional no están concebidos para ser consumidos por usuarios. Son piezas de la lógica del negocio, las cuales, poseen interfaces que los programadores aprovechan como parte de la implementación de nuevas aplicaciones o sistemas.

1.6 Selección de la metodología de desarrollo a utilizar

Luego del estudio realizado sobre las metodologías de desarrollo más difundidas a nivel internacional se decidió aplicar la metodología OASIS Web Service Implementation Methodology por ser una adaptación ideal de la metodología ágil XP al marco de la solución propuesta teniendo en cuenta los siguientes factores:

- ✓ Se requiere un corto plazo de tiempo para la culminación del producto, como máximo 4 meses.
- ✓ El equipo de trabajo es pequeño. En este caso de un solo desarrollador y el cliente.
- ✓ El cliente conoce ampliamente el funcionamiento del proceso del negocio.
- ✓ El cliente forma parte activa del equipo de desarrollo.

Conclusiones del capítulo

Para el desarrollo de la solución propuesta se utilizarán tecnologías compatibles con las empleadas en el desarrollo de conTics. El conjunto de tecnologías seleccionadas sería el siguiente:

- ✓ Lenguaje de Implementción - ASP.NET utilizando C# y el Framework .NET 3.5.
- ✓ Servidor HTTP – Internet Information Services, plataforma de despliegue por excelencia para aplicaciones web basadas en ASP.NET.
- ✓ Sistema gestor de Base de Datos – Oracle
- ✓ Entorno de desarrollo – Microsoft Visual Studio Team System 2008, entorno de desarrollo por excelencia para la plataforma .NET, incluye el .NET Framework 3.5.
- ✓ Metodología de desarrollo - OASIS Web Service Implementation Methodology por ser una adaptación ideal de la robusta metodología RUP al marco de la solución propuesta.

CAPÍTULO 2

**Características de la solución propuesta
Fase de requerimientos y análisis**

Introducción

En este capítulo se definen y describen las características de la solución propuesta. Se hace un estudio sobre las características del entorno donde trabajará el sistema así como del flujo actual de procesos involucrados en el campo de acción y de la forma de ejecución de los mismos, explicando las causas que originan la situación problemática y las consecuencias de las mismas.

Se hace un análisis comparativo entre soluciones actuales y la solución propuesta indicando las ventajas de la misma.

Según propone la metodología empleada, en la fase de **Requerimientos** y **Análisis** se determina la necesidad de implementación de servicios Web, se definen los requerimientos que deben cumplir los mismos según las necesidades del cliente, las cuales se describen en forma de historias de usuario. Se realiza una selección de la plataforma tecnológica como marco de trabajo para la implementación. Se define una propuesta de estructura arquitectónica para el Servicio Web. Se define la granularidad de los servicios agrupándolos según su funcionalidad en Servicios Web. Se identifican Servicios Web reusables y se identifican las interfaces de contrato para los nuevos servicios. Finalmente se confecciona el plan de entrega del producto.

2.1 Descripción del entorno donde trabajará el sistema

La solución propuesta formará parte del Sistema conTics el cual contiene un gran volumen de información referente al estado, ubicación, datos técnicos y de explotación de todos los medios técnicos en posesión del MININT.

Con el objetivo de automatizar los procesos de gestión de interrupciones y reparaciones se están desarrollando una serie de sistemas que para su funcionamiento pleno, necesitan estar vinculados al proceso de gestión de la información referente a estos medios.

Actualmente sólo es posible acceder a esta información a través de la interfaz Web del Sistema conTics ya que el mismo no brinda posibilidades de acceso desde otros sistemas.

Esta problemática es común en el Ministerio del Interior el cual se caracteriza por tener en explotación un gran número de sistemas que manejan elevados volúmenes de información, los cuales, están implementados sobre tecnologías y plataformas muy diversas por lo que se hace extremadamente complejo que compartan funcionalidades o se comuniquen entre sí.

Partiendo del ejemplo de un sistema de gestión **A** con una base de datos que contiene información necesaria para el funcionamiento total o parcial de un sistema **B**. Las alternativas que se aplican actualmente para satisfacer las necesidades de acceso a la información entre sistemas son:

- ✓ Acceder directamente a la base de datos del sistema **A** desde el sistema **B**.
- ✓ Incluir en el sistema **B** una copia total o parcial de la base de datos del sistema **A**.

Estas soluciones no son óptimas ya que, en el primer caso, se violan completamente las políticas de seguridad y restricciones de acceso a la información implementadas por el sistema original, se pone en riesgo la integridad de los datos al ser manejados libremente por un sistema externo, no se aprovechan las funcionalidades ya implementadas por el sistema de gestión y en segundo caso, además de presentar los inconvenientes ya mencionados, en un momento dado la versión de los datos contenidos en la copia de la base de datos no estará en correspondencia con los datos actuales y será necesaria la actualización sistemática de los mismos.

2.2 Propuesta del sistema

La propuesta del sistema consiste en la implementación de una capa de Servicios Web integrada al Sistema conTics del MININT que permita a otros sistemas informáticos obtener información actualizada imprescindible para el funcionamiento de los mismos. Sin importar la tecnología o plataforma de implementación empleada en los mismos.

La capa de servicios debe servir como interfaz de comunicación con otros sistemas brindando la posibilidad de acceso a una serie de operaciones.

Estas operaciones serán únicamente de lectura de información y deben respetar las restricciones de acceso definidas por conTics.

Se identifican como posibles consumidores de estos servicios los siguientes sistemas:

- ✓ Sistema de reparaciones y talleres.
- ✓ Sistema de gestión de interrupciones.
- ✓ Sistema de control de uso de computadoras conectadas a la red.

Se debe tener en cuenta el posible surgimiento de nuevos sistemas que requieran de esta funcionalidad.

Esta solución elimina totalmente los inconvenientes descritos anteriormente y expande las posibilidades de interacción con el Sistema de Control de Tecnologías de Informática y Comunicaciones del Ministerio del Interior.

2.2.1 Términos comunes asociados al entorno

Asociados directamente al proceso de gestión de la información referente a los medios técnicos al servicio del Ministerio del Interior se encuentran una serie de términos cuyos conceptos se exponen a continuación.

- ✓ **Medio Técnico:** elemento o dispositivo tecnológico que tiene asociado una serie de datos técnicos. Compone o define una entidad determinada.

- ✓ **Entidad:** estructura compuesta por uno o más medios técnicos de la cual es posible identificar datos de ubicación, asignación, uso y explotación.

Para facilitar la comprensión de estos términos se puede partir de los ejemplos concretos planteados a continuación.

- ✓ **Ejemplo 1.** Para el caso de una PC de escritorio se identifican una serie de medios técnicos (teclado, mouse, monitor, disco duro, etc.) los cuales conforman a la entidad Microcomputadora.
- ✓ **Ejemplo 2.** Para el caso de un teléfono, se define una entidad Teléfono que está compuesta únicamente por el medio técnico que la define (Teléfono).

2.2.2 Eventos principales del entorno

- ✓ **Solicitud de información:** Mediante este proceso, los sistemas externos o consumidores del servicio solicitan una determinada información imprescindible o complementaria para completar alguna de sus operaciones.

2.3 Descripción del sistema propuesto.

El sistema para su correcto funcionamiento y aceptación por parte del cliente debe cumplir una serie de especificaciones, las cuales han sido establecidas por el cliente en su papel dentro del equipo de trabajo como especialista en el funcionamiento del negocio.

2.3.1 Descripción del personal relacionado con el sistema.

Dada la naturaleza de la aplicación se toma como personal relacionado con el sistema a otros sistemas externos. Ya que a diferencia de las soluciones de software tradicionales, los Servicios Web están pensados para ser consumidos por otras aplicaciones y no por usuarios que interactúen directamente con los mismos.

Personal Relacionado	Justificación
Sistema Externo	En todos los casos, las operaciones serán iniciadas por los sistemas que necesiten obtener alguna información como parte del desarrollo de un determinado proceso interno de los mismos.

Tabla 2: Descripción del personal relacionado con el sistema

2.3.2 Listado de requerimientos del producto

La metodología propone el levantamiento de una serie de requisitos, que constituyen las características del producto y definen el funcionamiento natural del negocio.

R.1 La capa de Servicios Web debe permitir, dado el sello de un medio técnico, obtener todos los datos asociados a la entidad de la cual forma parte incluyendo al resto de los medios técnicos que la compongan.

R.2 La capa de Servicios Web debe permitir, dado el sello de un medio técnico, obtener todos los datos asociados al mismo.

R.3 La capa de Servicios Web debe permitir verificar el estado de un determinado sello.

R.3.1 Verificar si un sello ha sido editado.

R.3.2 Verificar si un sello existe.

R.3.3 Verificar si un sello ha sido asignado.

R.4 La capa de Servicios Web debe permitir obtener un listado que contenga todas las definiciones de entidades registradas en el sistema.

R.5 La capa de Servicios Web debe permitir, dado un nombre de una definición de entidad, obtener todos los datos asociados a la definición de la misma así como de los medios técnicos que la componen así como indicar en cada caso si el medio técnico es obligatorio, opcional o si define a la entidad.

R.6 La capa de Servicios Web debe permitir, dado un nombre de definición de medio técnico, obtener los datos de explotación definidos para el mismo.

R.7 La capa de Servicios Web debe permitir obtener todos los datos asociados a una entidad partir del número de identificador raíz de la misma.

R.8 La capa de Servicios Web debe permitir, dado el identificador de la entidad, obtener los datos históricos de la misma.

R.9 La capa de Servicios Web debe permitir, dado el identificador de la entidad, obtener los datos de ubicación de la misma.

R.10 La capa de Servicios Web debe permitir, dado el nombre de un medio técnico, obtener un listado de las marcas nombradas para el mismo.

R.11 La capa de Servicios Web debe permitir, dado el sello de un medio técnico, obtener un listado con las marcas registradas para ese tipo de medio.

R.12 La capa de Servicios Web debe permitir, dado el nombre y la marca del medio técnico, obtener un listado de los modelos normalizados para los medios de ese tipo y esa marca.

R.13 La capa de Servicios Web debe permitir, dado el nombre de una entidad, obtener un listado de los usos nombrados para la misma agrupándolos por sus categorías.

R.14 Garantizar el acceso a las funcionalidades de la capa de Servicios Web desde cualquier plataforma o lenguaje de implementación.

R.15 El sistema debe ser capaz de brindar la información con el criterio de búsqueda indicado por el usuario.

R.16 La eficiencia del producto estará condicionada por la velocidad de transmisión de la red y su nivel de congestión.

R.17 El sistema debe tener una elevada capacidad de respuesta y retornar los resultados en el menor tiempo posible.

R.18 El sistema debe estar disponible las 24 horas del día y los 7 días de la semana.

R.19 El acceso a la información brindada por la capa de Servicios Web debe respetar las políticas de acceso por roles definidas por conTics.

2.4 Historias de usuario

La metodología seleccionada para el proceso de desarrollo de la solución emplea las Historias de Usuario (HU) para describir y organizar las ideas del cliente.

Estas historias describen cómo el sistema debe trabajar y están escritas a un alto nivel con ayuda del cliente y la estructura de las mismas es la siguiente:

- ✓ **Nombre:** Nombre descriptivo de la Historia de Usuario.
- ✓ **Prioridad:** Grado de prioridad que le asigna el cliente a la Historia de Usuario en dependencia de sus necesidades. Los valores que puede tomar son (Alta, Media o Baja)
- ✓ **Complejidad:** Grado de complejidad que le asigna el equipo de desarrollo a la Historia de Usuario luego de analizarla. (Alta, Media o Baja)
- ✓ **Estimación:** Unidades de tiempo estimadas por el equipo de desarrollo para darle cumplimiento a la Historia de Usuario.
- ✓ **Iteración:** Número de la iteración en la cual será implementada la Historia de Usuario.
- ✓ **Descripción:** Descripción simple que brinda el cliente sobre lo que debe hacer la funcionalidad en cuestión.

Adicionalmente a cada Historia de Usuario se le asigna un número para facilitar su identificación por parte del equipo de desarrollo y se le puede incluir alguna información adicional que pueda ser útil para la comprensión de la misma.

Historia de Usuario				Número: 1
Nombre:	Obtener datos de la entidad			
Prioridad:	Alta	Complejidad:	Alta	Estimación: 4 Iteración: 2
Descripción:				
Dado el sello de un medio técnico, obtener todos los datos asociados a la entidad de la cual forma parte incluyendo al resto de los medios técnicos que la compongan.				
Información Adicional:				
Da cumplimiento al requisito R1 . Se debe tener en cuenta el nivel de acceso del sistema que solicita la información.				

Tabla 3: Descripción de la historia de usuario: Obtener datos de la entidad

Historia de Usuario				Número: 2
Nombre:	Obtener datos del medio técnico			
Prioridad:	Alta	Complejidad:	Alta	Estimación: 4 Iteración: 2
Descripción:				
Dado el sello de un medio técnico, obtener todos los datos asociados al mismo.				
Información Adicional:				
Da cumplimiento al requisito R2. Se debe tener en cuenta el nivel de acceso del sistema que solicita la información.				

Tabla 4: Descripción de la historia de usuario: Obtener datos del medio técnico

Historia de Usuario				Número: 3
Nombre:	Verificar si el sello existe			
Prioridad:	Alta	Complejidad:	Baja	Estimación: 1 Iteración: 1
Descripción:				
Comprobar si un determinado sello existe en el sistema.				
Información Adicional:				
Da cumplimiento al requisito R3.2.				

Tabla 5: Descripción de la historia de usuario: Verificar si el sello existe

Historia de Usuario				Número: 4
Nombre:	Verificar si el sello ha sido editado			
Prioridad:	Alta	Complejidad:	Baja	Estimación: 1 Iteración: 1
Descripción:				
Comprobar si un determinado sello ha sido editado.				
Información Adicional:				
Da cumplimiento al requisito R3.1.				

Tabla 6: Descripción de la historia de usuario: Verificar si el sello existe

Historia de Usuario				Número: 5
Nombre:	Verificar si el sello ha sido asignado			
Prioridad:	Alta	Complejidad:	Baja	Estimación: 1 Iteración: 1
Descripción:				
Comprobar si un determinado sello ha sido asignado.				
Información Adicional:				
Da cumplimiento al requisito R3.3.				

Tabla 7: Descripción de la historia de usuario: Verificar si el sello ha sido asignado

Historia de Usuario				Número: 6
Nombre:	Obtener listado de definiciones de entidad			
Prioridad:	Alta	Complejidad:	Baja	Estimación: 1 Iteración: 1
Descripción:				
Obtener un listado que contenga todas las definiciones de entidades registradas en el sistema.				
Información Adicional:				
Da cumplimiento al requisito R4.				

Tabla 8: Descripción de la historia de usuario: Obtener listado de definiciones de entidad

Historia de Usuario				Número: 7
Nombre:	Obtener los datos asociados a una definición de entidad			
Prioridad:	Alta	Complejidad:	Alta	Estimación: 4 Iteración: 2
Descripción:				
Dado un nombre de una definición de entidad, obtener todos los datos asociados a la definición de la misma así como de los medios técnicos que la componen así como indicar en cada caso si el medio técnico es obligatorio, opcional o si define a la entidad.				
Información Adicional:				
Da cumplimiento al requisito R5.				

Tabla 9: Descripción de la historia de usuario: Obtener los datos asociados a una definición de entidad

Historia de Usuario				Número: 8
Nombre:	Obtener los datos asociados a la definición de datos de explotación de un tipo de MT.			
Prioridad:	Media	Complejidad:	Media	Estimación: 2 Iteración: 2
Descripción:				
Dado un nombre de definición de medio técnico, obtener los datos de explotación definidos para el mismo.				
Información Adicional:				
Da cumplimiento al requisito R6.				

Tabla 10: Descripción de la historia de usuario: Obtener los datos asociados a la definición de datos de exp. de un tipo de MT

Historia de Usuario				Número: 9
Nombre:	Obtener datos de la entidad por id			
Prioridad:	Alta	Complejidad:	Alta	Estimación: 4 Iteración: 2
Descripción:				
Obtener todos los datos asociados a una entidad partir del número de identificador raíz de la misma.				
Información Adicional:				
Da cumplimiento al requisito R7. Se debe tener en cuenta el nivel de acceso del sistema que solicita la información.				

Tabla 11: Descripción de la historia de usuario: Obtener datos de la entidad por id

Historia de Usuario				Número: 10
Nombre:	Obtener datos históricos de la entidad			
Prioridad:	Baja	Complejidad:	Alta	Estimación: 4 Iteración: 3
Descripción:				
Dado el identificador de la entidad, obtener los datos históricos de la misma				
Información Adicional:				
Da cumplimiento al requisito R8. Se debe tener en cuenta el nivel de acceso del sistema que solicita la información.				

Tabla 12: Descripción de la historia de usuario: Obtener datos históricos de la entidad

Historia de Usuario				Número: 11
Nombre:	Obtener datos de ubicación de la entidad			
Prioridad:	Baja	Complejidad:	Media	Estimación: 2 Iteración: 3
Descripción:				
Dado el identificador de la entidad, obtener los datos de ubicación de la misma				
Información Adicional:				
Da cumplimiento al requisito R9 . Se debe tener en cuenta el nivel de acceso del sistema que solicita la información.				

Tabla 13: Descripción de la historia de usuario: Obtener datos de ubicación de la entidad

Historia de Usuario				Número: 12
Nombre:	Obtener marcas nombradas para un tipo de medio técnico			
Prioridad:	Baja	Complejidad:	Baja	Estimación: 1 Iteración: 3
Descripción:				
Dado el nombre de un medio técnico, obtener un listado de las marcas nombradas para el mismo.				
Información Adicional:				
Da cumplimiento a los requisitos R10, R11 .				

Tabla 14: Descripción de la historia de usuario: Obtener marcas nombradas para un tipo de medio técnico

Historia de Usuario				Número: 13
Nombre:	Obtener modelos nombrados para una marca y tipo de medio técnico			
Prioridad:	Baja	Complejidad:	Baja	Estimación: 1 Iteración: 3
Descripción:				
Dado el nombre y la marca del medio técnico, obtener un listado de los modelos normalizados para los medios de ese tipo y esa marca.				
Información Adicional:				
Da cumplimiento al requisito R12 .				

Tabla 15: Descripción de la historia de usuario: Obtener modelos nombrados para una marca y tipo de medio técnico

Historia de Usuario				Número: 14
Nombre:	Obtener listado de usos para un tipo de entidad			
Prioridad:	Alta	Complejidad:	Media	Estimación: 2 Iteración: 1
Descripción:				
Dado el nombre de una entidad, obtener un listado de los usos nomencrados para la misma indicando sus categorías.				
Información Adicional:				
Da cumplimiento al requisito R13.				

Tabla 16: Descripción de la historia de usuario: Obtener listado de usos para un tipo de entidad

2.4.1 Sobre las Historias de usuario

Para la confección del modelo de las Historias de Usuario se aplicaron una serie de convenciones con el objetivo de adaptarlas a la situación real, las cuales aparecen enumeradas y descritas a continuación.

- ✓ **Omitir el campo de “Usuario” en las fichas:** Este campo se utiliza cuando con el sistema interactúan más de un tipo de usuario. La metodología seleccionada, permite que sea omitido puesto que, al tratarse de Servicios Web, estos siempre serán iniciados por un Sistema Externo y todas sus funcionalidades están disponibles por igual.
- ✓ **Unidades de estimación:** Dadas las características de las funcionalidades a implementar en el desarrollo de Servicios Web, se ha tomado como unidad mínima de estimación una sesión de trabajo, equivalente a 8 horas generalmente distribuidas en los intervalos de 08:00 – 13:00 y de 14:00 – 17:00. Aunque estos intervalos pueden ser ajustados a conveniencia del equipo de desarrollo. Llevando este tiempo a semanas quedaría de la siguiente forma: **1 Semana = 5 sesiones** de trabajo = **40 Horas**.

2.5 Planificación

Durante esta fase se estima el tiempo total que tomará la implementación de la solución tomando como base la estimación de tiempo asignada a cada una de las historias de usuario. Posteriormente se procede a distribuir las entre varias iteraciones ordenándolas por prioridad y agrupándolas en paquetes de 20 sesiones de trabajo.

De esta manera se garantiza que al menos una vez al mes el cliente tendrá un prototipo del sistema con un porcentaje de funcionalidades listas para ser probadas y tomar decisiones en cuanto a cambios que puedan requerirse para mantener el camino correcto rumbo a la meta propuesta.

Iteración	Historias de Usuario	Tiempo
1	<ul style="list-style-type: none"> 3. Verificar si el sello existe 4. Verificar si el sello ha sido editado 5. Verificar si el sello ha sido asignado 6. Obtener listado de definiciones de entidad 14. Obtener listado de usos para un tipo de entidad 	2 semanas
2	<ul style="list-style-type: none"> 1. Obtener datos de la entidad 2. Obtener datos del medio técnico 7. Obtener los datos asociados a una definición de entidad 8. Obtener los datos asociados a la definición de datos de explotación de un tipo de MT 9. Obtener datos de la entidad por id 	4 semanas
3	<ul style="list-style-type: none"> 10. Obtener datos históricos de la entidad 11. Obtener datos de ubicación de la entidad 12. Obtener marcas nombradas para un tipo de medio técnico 13. Obtener modelos nombrados para una marca y tipo de medio técnico 	4 semanas

Tabla 17: Distribución de Historias de Usuario por iteración.

2.6 Plataforma tecnológica como marco de trabajo para la implementación.

La metodología WSIM basada en XP propone que se involucre al cliente en la selección de la plataforma tecnológica a aplicar en la solución, mostrándole los diversos lenguajes y frameworks que se pueden emplear, describiendo los beneficios de cada uno.

Se plantea que estas decisiones afectan el ciclo de vida de los Servicios Web en materia de costos de mantenimiento y otras cuestiones reales del negocio por lo que no se debe asumir nada y hacer que el cliente tome parte en las mismas.

En este caso, tal y como se explicó en el Capítulo 1, sección 1.3, en la elección de la plataforma tecnológica debe tener en cuenta la tecnología empleada en el desarrollo y despliegue de sistema conTics

por cuestiones elementales de compatibilidad, integración y aprovechamiento de los recursos existentes puestos a disposición para la explotación del mismo.

Por estas cuestiones, como se explica en el mencionado capítulo se decide conjuntamente con el cliente emplear la plataforma .NET de Microsoft, específicamente la versión 3.5 del .NET Framework por incluir mejoras en cuanto a la facilidad de implementación de Servicios Web bajo la filosofía de Windows Communication Foundation. Por consiguiente, el servidor Web a emplear será Internet Information Service por ser mejor caso de integración con el .NET Framework.

2.7 Propuesta de especificación de la arquitectura de software

Una vez identificados los objetivos del cliente se procede a identificar las opciones en cuanto a tipos de contenidos a ser ofrecidos como Servicios Web.

Se toman todas las opciones en las que el cliente se muestra interesado y el próximo paso es evolucionar estas opciones a posibles Servicios Web. Definiendo así una arquitectura de alto nivel identificando los componentes que exponen funcionalidades como Servicios Web.

2.7.1 Definición de la granularidad de los Servicios Web

A partir de las historias de usuario suministradas por el cliente es posible identificar grupos de servicios y funcionalidades como Servicios Web.

Básicamente se han identificado 3 grupos importantes de funcionalidades según su tipo que pueden ser agrupadas y contenidas dentro de Servicios Web.

Estos tres grupos son:

- ✓ **Verificaciones de sellos:** Todas sus funcionalidades se basan en verificar el estado de un determinado número de sello. Contiene las historias de usuario: **3, 4 y 5**. Son completamente públicos ya que no brindan información específica sobre ningún medio técnico o entidad.
- ✓ **Obtención de datos generales:** Todas sus funcionalidades están dirigidas a la recuperación de datos nombrados o definiciones. Son completamente públicos ya que no brindan información específica sobre ningún medio técnico o entidad. Contiene las historias de usuario: **6, 7, 8, 12, 13 y 14**.

- ✓ **Obtención de datos específicos:** El objetivo de cada una de sus funcionalidades es devolver los datos registrados sobre una entidad o medio técnico específico. Debe tener en cuenta el nivel de acceso a la información según los privilegios del sistema que accede previa autenticación. Contiene las historias de usuario: **1, 2, 9, 10 y 11.**

2.7.2 Servicios Web reutilizables

La metodología WSIM plantea que, como parte del proceso de definición de las especificaciones de arquitectura de software, se realice un estudio de los posibles Servicios Web existentes que brinden algunas de las funcionalidades solicitadas por el cliente, con el objetivo de re-utilizarlos como parte de la solución propuesta. En este caso no existe ningún servicio aprovechable por lo tanto queda descartada esta posibilidad.

Conclusiones del capítulo

En este capítulo han quedado estrictamente plasmadas las características que debe tener la solución propuesta, obtenidas a partir del entendimiento del funcionamiento de los flujos de trabajo asociados al proceso de gestión de la información referente a los medios tecnológicos empleados en el Ministerio del Interior.

Han quedado definidas las funcionalidades necesarias para satisfacer las necesidades y solucionar los problemas detectados en el funcionamiento del mismo y expandir las posibilidades del Sistema de Control de Tecnologías de Informática y Comunicaciones del Ministerio del Interior.

También se han definido los requerimientos e Historias de Usuario a tener en cuenta para el correcto desarrollo de la etapa de diseño e implementación.

CAPÍTULO 3

**Construcción de la solución propuesta
Fase de diseño, implementación y pruebas**

Introducción.

La metodología WSIM propone para esta fase tareas tales como, la refinación de los requerimientos e historias de usuario propuestas por el cliente, adaptación de los Servicios Web reutilizables en caso de existir alguno, refinar las interfaces de los nuevos Servicios Web, diseñar los servicios, programación del funcionamiento interno de los servicios, creación de códigos de ejemplo de consumidores de los servicios y la realización de pruebas unitarias.

El principal artefacto resultante de esta fase consiste en código fuente de la aplicación, ya que la metodología no recomienda el uso de diagramas UML con el objetivo de minimizar el tiempo de desarrollo del producto. De cualquier manera, no se prohíbe el uso de aquellos diagramas que el equipo de desarrollo estime conveniente usar para lograr un mayor entendimiento interno.

En el presente capítulo se incluyen temas relacionados con los principios de la orientación a servicios, algunos conceptos fundamentales para la comprensión de la estructura del sistema así como las Pruebas de Aceptación que garantizan que cada Historia de Usuario fue implementada correctamente.

3.1 Diseño del sistema

Para facilitar la comprensión de la concepción del sistema, se impone describir brevemente los principios de la orientación a servicios utilizando Windows Communication Foundation y algunos conceptos relacionados.

Con el objetivo de agilizar el trabajo, la metodología WSIM, hereda de XP la recomendación de utilizar sencillos esquemas descritos en pizarras y técnicas como las tarjetas CRC (Clase, Responsabilidad, Colaborador) en lugar de diagramas de clases UML para describir el diseño del software.

3.1.1 Principios de la orientación a servicios

En las aplicaciones distribuidas orientadas a servicios, en la práctica, lo que se hace es basarse en servicios orientados a mensajes e implementar estos servicios usando orientación a objetos.

Esta filosofía sigue 4 principios fundamentales:

- ✓ **Las fronteras de un servicio son explícitas.** Los servicios interactúan compartiendo mensajes entre diferentes ubicaciones, estas ubicaciones trazan fronteras formales y explícitas con el objetivo de preservar la flexibilidad en cuanto a cómo son implementados y distribuidos.
- ✓ **Cada servicio es autónomo.** Debe ser posible cambiar un servicio sin interrumpir el sistema completo. Los servicios deben ser distribuidos de forma independiente, de manera que los servicios no fallen cuando partes del sistema completo no están disponibles. Los cambios en la implementación de un servicio no requieren cambios en los clientes porque estos se encuentran suavemente acoplados.
- ✓ **Los servicios comparten esquemas y contratos, no clases ni tipos.** En su marcada diferencia con la orientación a objetos, un servicio no comparte clases ni tipos, sino esquemas y contratos. Esto se debe a que un servicio no conoce el ambiente de ejecución de otro servicio o de un cliente y la información que ambos intercambian debe ser neutral de lo contrario se vería limitada la interoperabilidad. Los contratos y esquemas de un servicio deben permanecer estables todo el tiempo de vida del mismo.

- ✓ **La compatibilidad se basa en políticas.** Los servicios no interactúan de forma arbitraria. Cada servicio provee una descripción de sus funcionalidades y requerimientos.

3.1.2 Contratos de un servicio.

Los contratos son, en el caso de Windows Communication Foundation, son **Interfaces** o **Clases** que detallan cuáles operaciones ofrece el servicio.

La forma más rápida de definir un contrato no es escribiendo un WSDL con su complejo código de marcas. En este caso lo ideal es programarlos en C# y WCF se encarga de el resto. Los tipos de contratos básicos necesarios para la puesta a punto de un Servicio Web son los siguientes.

- ✓ **Contrato de servicio.** Este contrato describe las funcionalidades del servicio. Cada contrato de servicio ofrece un conjunto de operaciones denominadas operaciones de servicio. En términos de C#, un contrato es un tipo (clase o interfaz) en el que los métodos son las operaciones del servicio.
- ✓ **Contrato de datos.** Las operaciones de un servicio aceptan tipos de datos complejos, es decir, estructuras además de los conocidos tipos simples que son multiplataforma por naturaleza (cadenas, enteros, booleanos, etc.). Estas estructuras se definen a través de los contratos de datos.

Por ejemplo, en el caso real de la implementación de la solución propuesta se han definido tres Servicios Web con varias funcionalidades. Para cada uno de estos servicios se define un contrato de servicio o de interfaz en el cual se describen los nombres de las funcionalidades que brinda el mismo así como las políticas en cuanto a datos requeridos o parámetros y datos de respuesta.

En la mayoría de los casos se trabaja con tipos básicos como: booleanos, enteros y cadenas. Algunas de las funcionalidades deber retornar tipos de datos complejos definidos específicamente para esta solución, como es el caso, por ejemplo, de la funcionalidad “**Obtener datos de la entidad**” es necesario devolver un objeto **Entidad** compuesto por varios objetos **MedioTecnico**. Teniendo en cuenta que este tipo de datos fue definido en C# y el objetivo fundamental del servicio es que pueda ser consumido desde cualquier plataforma o lenguaje de implementación, el hecho de utilizar en el contrato un tipo de dato complejo podría ser una seria limitante.

Gracias a los contratos de datos de Windows Communication Foundation (WCF) es posible usar cómodamente estas estructuras complejas sin temor alguno.

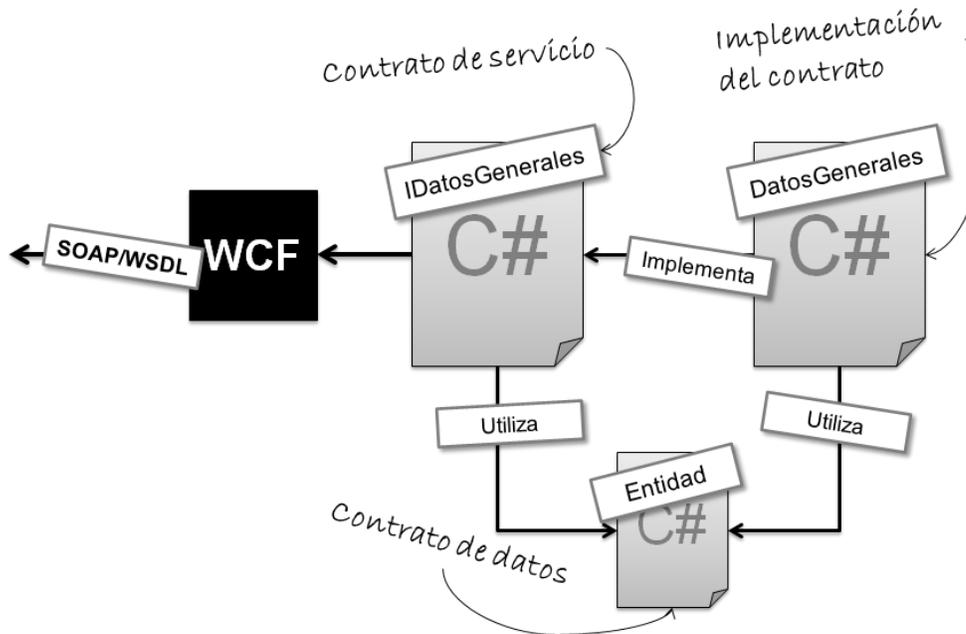


Ilustración 2: Relación entre contrato de servicio y contrato de datos en WCF

Como se puede apreciar en la **Ilustración 2**, el contrato del servicio se define como una interfaz de C# que es implementada por una clase donde se incluye toda la lógica de las funcionalidades, en ambos casos se utilizan los tipos de datos complejos definidos en clases como contratos de datos.

Finalmente Windows Communication Foundation se encarga de convertir los datos complejos en estructuras de datos básicas que pueden ser interpretadas por cualquier tipo de tecnología y transforma la interfaz del contrato del servicio en un documento WSDL estándar.

La metodología WSIM plantea que se deben especificar los contratos de interfaz para cada uno de los servicios definidos. Estos contratos, una vez definidos, no deben variar puesto que cualquier cambio en los contratos afectaría el funcionamiento de los consumidores del servicio.

3.1.2.1 Contrato de interfaz para los nuevos servicios

Según quedó definido en el acápite 2.7.1 existen tres grupos de funcionalidades que darán lugar a Servicios Web. Las interfaces de contrato para cada uno de estos servicios son las siguientes.

Verificación de Sello		
Nombre: Verificar si el sello existe		Historia de Usuario: 3
Función: VerificarExistenciaSello		Retorno: bool
Tipo	Nombre	Descripción:
string	sello	Número de sello que se desea verificar.
Nombre: Verificar si el sello ha sido editado		Historia de Usuario: 4
Función: VerificarEdicionSello		Retorno: bool
Tipo	Nombre	Descripción:
string	sello	Número de sello que se desea verificar.
Nombre: Verificar si el sello ha sido asignado		Historia de Usuario: 5
Función: VerificarAsignacionSello		Retorno: bool
Tipo	Nombre	Descripción:
string	sello	Número de sello que se desea verificar.

Tabla 18: Definición de interfaces contrato del Servicio Web: Verificación de sello

Datos Generales			
Nombre: Obtener listado de definiciones de entidad		Historia de Usuario: 6	
Función: ListadoDefinicionEntidad		Retorno: string[]	
Tipo	Nombre	Descripción:	
-	-	-	
Nombre: Obtener los datos asociados a una definición de entidad		Historia de Usuario: 7	
Función: DatosDefinicionEntidad		Retorno: EntidadDef	
Tipo	Nombre	Descripción:	
string	tipo	Nombre del tipo de entidad de la cual se desean obtener datos de definición.	
Nombre: Obtener los datos asociados a la definición de datos de explotación de un tipo de MT		Historia de Usuario: 8	
Función: ListadoDatosExplotacion		Retorno: DatoExplotacionDef []	
Tipo	Nombre	Descripción:	
string	tipo	Nombre del tipo de entidad de la cual se desea obtener la definición de datos de explotación.	
Nombre: Obtener marcas nomencladas para un tipo de medio técnico.		Historia de Usuario: 12	
Función: ListadoMarcasNomecladas		Retorno: string[]	
Tipo	Nombre	Descripción:	
string	tipo	Nombre del tipo de entidad de la cual se desea obtener el listado de marcas definidas.	
Nombre: Obtener modelos nomenclados para una marca y tipo de medio técnico.		Historia de Usuario: 13	
Función: ListadoModelosNomeclados		Retorno: bool	
Tipo	Nombre	Descripción:	
string	tipo	Nombre del tipo de entidad.	
string	modelo	Modelo del cual se desea obtener el listado de marcas definidas.	
Nombre: Obtener listado de usos para un tipo de entidad		Historia de Usuario: 14	
Función: ListadoUsosNomenclados		Retorno: Uso[]	
Tipo	Nombre	Descripción:	
string	tipo	Nombre del tipo de entidad de la que se desea conocer los usos.	

Tabla 19: Definición de interfaces contrato del Servicio Web: Datos Generales

Datos Específicos			
Nombre: Obtener datos de la entidad		Historia de Usuario: 1	
Función: DatosEntidadSello		Retorno: Entidad	
Tipo	Nombre	Descripción:	
string	sello	Sello de uno de los medios técnicos de la entidad de la que se desea obtener información.	
Nombre: Obtener los datos asociados a una definición de entidad		Historia de Usuario: 2	
Función: DatosMedioTecnico		Retorno: MedioTecnico	
Tipo	Nombre	Descripción:	
string	sello	Sello del medio técnico del que se desea obtener información.	
Nombre: Obtener datos de la entidad por id		Historia de Usuario: 9	
Función: DatosEntidadId		Retorno: Entidad	
Tipo	Nombre	Descripción:	
int	id	Id raíz de la entidad de la que se desea obtener información.	
Nombre: Obtener datos históricos de la entidad.		Historia de Usuario: 10	
Función: DatosHistoricos		Retorno: Entidad[]	
Tipo	Nombre	Descripción:	
int	id	Id raíz de la entidad de la que se desea obtener información.	
Nombre: Obtener datos de ubicación de la entidad		Historia de Usuario: 11	
Función: DatosUbicacion		Retorno: Ubicacion	
Tipo	Nombre	Descripción:	
int	id	Id raíz de la entidad de la que se desea obtener información.	

Tabla 20: Definición de interfaces contrato del Servicio Web: Datos Específicos

3.1.2.2 Definición de los contratos de datos

Como se van a implementar tres Servicios Web que utilizarán contratos de datos comunes y accederán a una misma fuente de datos, se tomó la decisión de crear una librería común para los contratos de datos.

3.1.3 Librerías de servicios.

La plataforma WCF permite implementar servicios con independencia del soporte de difusión que se empleará para los mismos.

En el caso de los Servicios Web tradicionales de .NET, éstos dependían totalmente de un servidor HTTP/HTTPS (Internet Information Service) para su difusión y debían ser implementados sobre ASP. Los servicios de WCF pueden ser difundidos sin emplear un servidor HTTP y pueden estar soportados sobre otros protocolos como TCP.

Gracias a este nivel de independencia, para facilitar su portabilidad es posible crear las llamadas **Librerías de Servicios de Windows Communication Foundation** que dan como resultado un fichero DLL que, como se representa en la **Ilustración 3**, contiene el la interfaz de contrato del servicio, así como su implementación y puede ser incluido en cualquier proyecto de .NET.

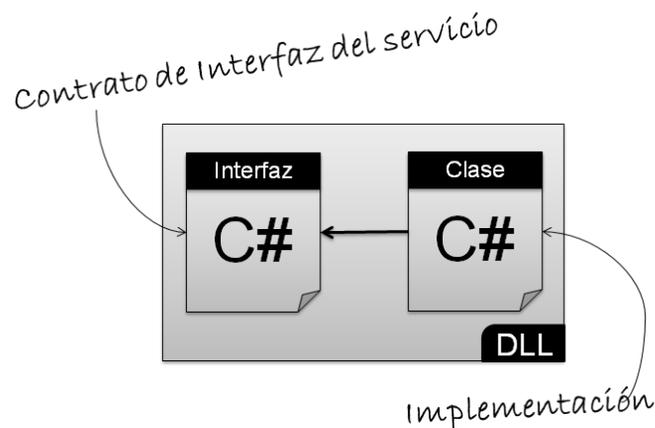


Ilustración 3: Representación de una librería WCF

3.1.4 Acceso a la base de datos.

Se decidió crear una librería que contenga todas las utilidades necesarias para el acceso a la base de datos del Sistema de Control de Tecnologías de Informática y Comunicaciones.

3.1.5 Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad, Colaborador) son una herramienta de reflexión en el diseño de software orientado a objetos. Fueron propuestas por Ward Cunningham y Kent Beck. Se utilizan normalmente cuando primero se determinan las clases que se necesitan y cómo van a interactuar y después se implementa la solución.

Las partes que componen una tarjeta CRC son las siguientes:

- ✓ **Clase:** Nombre de la clase con que se está modelando.
- ✓ **Súper Clase:** Nombre de la clase padre en la herencia.
- ✓ **Sub Clase(s):** Nombre de las clases hijas en la herencia.
- ✓ **Responsabilidades:** Descripción de alto nivel del propósito de la clase.
- ✓ **Colaboraciones:** Indica con cuáles otras clases se requiere relación para cumplir con la responsabilidad.

A continuación, las tarjetas CRC agrupadas por librerías.

3.1.5.1 Librería de acceso a datos (CTDBA)

Tarjeta CRC	
Clase: DBA Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Gestionar la conexión a la fuente de datos. ✓ Brindar las funcionalidades de lectura de datos.	-

Tabla 21: Tarjeta CRC de la clase DBA

3.1.5.2 Librería de contratos de datos (DataContracts)

Tarjeta CRC	
Clase: MedioTecnico Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir contrato de datos para el tipo de objeto Medio Técnico	-

Tabla 22: Tarjeta CRC de la clase MedioTecnico

Tarjeta CRC	
Clase: Uso Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir contrato de datos para el tipo de objeto Uso.	-

Tabla 23: Tarjeta CRC de la clase Uso

Tarjeta CRC	
Clase: Ubicacion Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir contrato de datos para el tipo de objeto Ubicación.	-

Tabla 24: Tarjeta CRC de la clase Ubicacion

Tarjeta CRC	
Clase: DatoExplotacion Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir contrato de datos para el tipo de objeto Dato de Explotación.	-

Tabla 25: Tarjeta CRC de la clase DatoExplotacion

Tarjeta CRC	
Clase: Entidad Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir contrato de datos para el tipo de objeto Entidad	✓ MedioTecnico ✓ DatoExplotacion ✓ Ubicacion ✓ Uso

Tabla 26: Tarjeta CRC de la clase Entidad

Tarjeta CRC	
Clase: MedioTecnicoDef Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir contrato de datos para la definición de Medio técnico.	-

Tabla 27: Tarjeta CRC de la clase MedioTecnicoDef

Tarjeta CRC	
Clase: DatoExplotacionDef Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir contrato de datos para la definición de Dato de Explotación	-

Tabla 28: Tarjeta CRC de la clase DatoExplotacionDef

Tarjeta CRC	
Clase: EntidadDef Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir contrato de datos para la definición de Entidad	✓ MedioTecnicoDef ✓ DatoExplotacionDef

Tabla 29: Tarjeta CRC de la clase EntidadDef

3.1.5.3 Librería del Servicio Web Verificar Sello (VerificarSelloLib)

Tarjeta CRC	
Clase: IVerificarSello	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir el contrato de interfaz para el Servicio Web Verificar Sello	✓ VerificarSello

Tabla 30: Tarjeta CRC de la clase IVerificarSello

Tarjeta CRC	
Clase: VerificarSello	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> ✓ Verificar si un sello existe ✓ Verificar si un sello ha sido editado ✓ Verificar si un sello ha sido asignado 	<ul style="list-style-type: none"> ✓ DBA ✓ IVerificarSello

Tabla 31: Tarjeta CRC de la clase VerificarSello

3.1.5.4 Librería del Servicio Web Datos Generales (DatosGeneralesLib)

Tarjeta CRC	
Clase: IDatosGenerales Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Definir el contrato de interfaz para el Servicio Web Datos Generales.	✓ DatosGenerales

Tabla 32: Tarjeta CRC de la clase IDatosGenerales

Tarjeta CRC	
Clase: DatosGenerales Súper Clase: - Sub Clase(s): -	
Responsabilidades	Colaboraciones
✓ Listar definiciones de entidad. ✓ Obtener datos de definición de entidad. ✓ Listar datos de explotación. ✓ Listar marcas nomencladas. ✓ Listar modelos nomenclados. ✓ Listar usos nomenclados.	✓ DBA ✓ EntidadDef ✓ DatoExplotacionDef ✓ Uso

Tabla 33: Tarjeta CRC de la clase DatosGenerales

3.1.5.5 Librería del Servicio Web Datos Específicos (DatosEspecificosLib)

Tarjeta CRC	
Clase: IDatosEspecificos	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> ✓ Definir el contrato de interfaz para el Servicio Web Datos Específicos 	<ul style="list-style-type: none"> ✓ DatosEspecificos

Tabla 34: Tarjeta CRC de la clase IDatosEspecificos

Tarjeta CRC	
Clase: DatosEspecificos	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> ✓ Obtener los datos de una entidad por su sello. ✓ Obtener los datos de un medio técnico por su sello. ✓ Obtener los datos de una entidad por su identificador. ✓ Listar los datos históricos de una entidad. ✓ Obtener los datos de ubicación de una entidad. ✓ Iniciar sesión. ✓ Cerrar sesión. 	<ul style="list-style-type: none"> ✓ DBA ✓ Entidad ✓ MedioTecnico ✓ Ubicacion

Tabla 35: Tarjeta CRC de la clase DatosEspecificos

3.1.6 Esquema del sistema.

Para mejor entendimiento de la estructura del sistema completo se muestra el siguiente esquema donde se representan cada una de las librerías implementadas y la relación entre ellas.

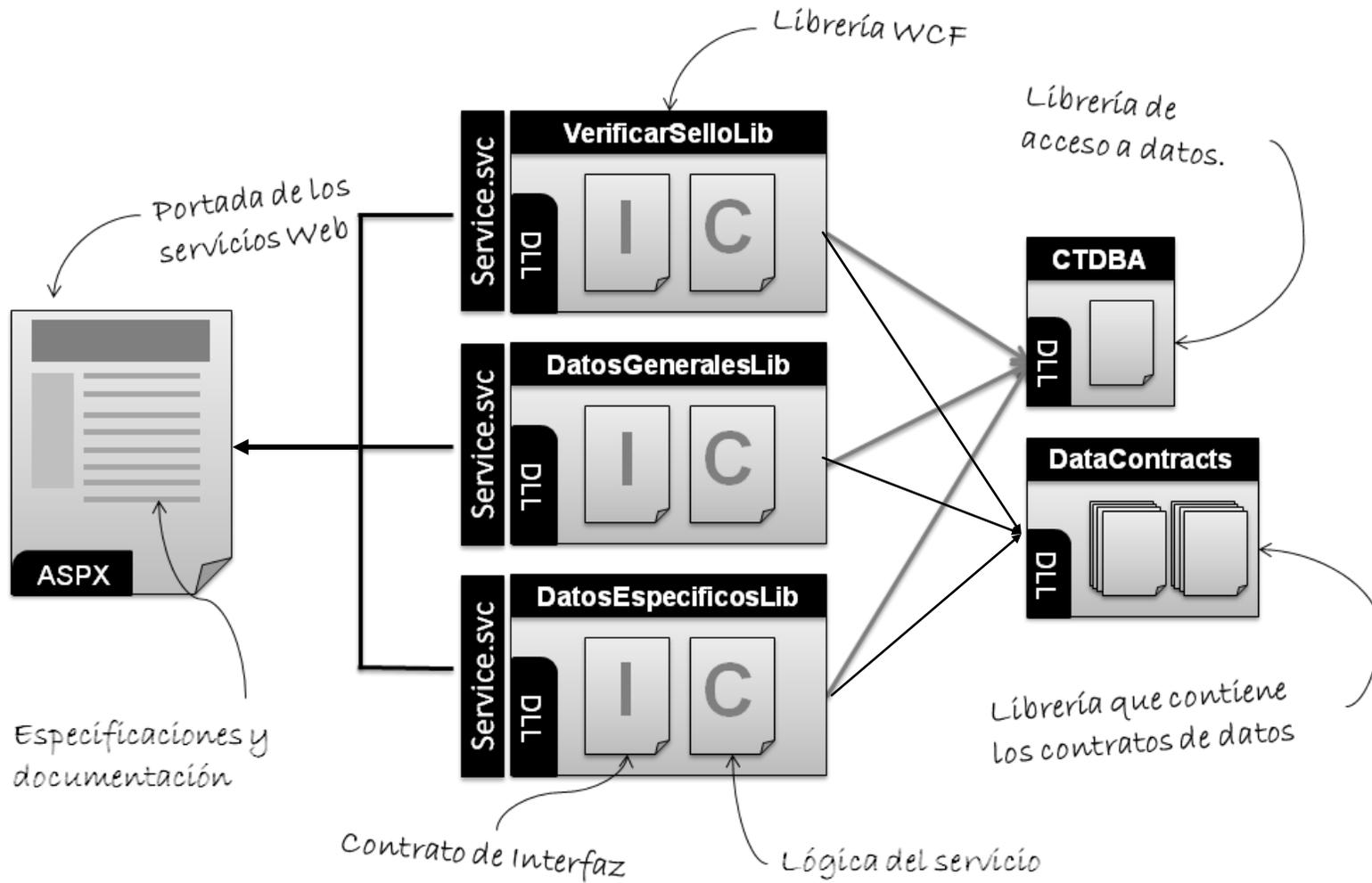


Ilustración 4: Estructura del sistema.

3.2 Implementación del sistema

En esta fase se genera todo el código fuente necesario para satisfacer las Historias de Usuario definidas para la solución y se describen todas las tareas realizadas en cada iteración.

1^{era} Iteración

El objetivo de esta iteración es desarrollar las Historias de Usuario 3, 4, 5, 6 y 14, éstas son: “Verificar si el sello existe”, “Verificar si el sello ha sido editado”, “Verificar si el sello ha sido asignado”, “Obtener listado de definiciones de entidad” y “Obtener listado de usos para un tipo de entidad” respectivamente. Al final de esta iteración se deben tener, las funcionalidades propuestas en cada HU, las implementaciones de los contratos de datos requeridos para el funcionamiento de las mismas y la librería de acceso a datos.

2^{da} Iteración

El objetivo de esta iteración es desarrollar las Historias de Usuario 1, 2, 7, 8 y 9, éstas son: “Obtener datos de la entidad”, “Obtener datos del medio técnico”, “Obtener los datos asociados a una definición de entidad”, “Obtener los datos asociados a la definición de datos de explotación de un tipo de MT” y “Obtener datos de la entidad por id” respectivamente. Al final de esta iteración se deben tener, las funcionalidades propuestas en cada HU y las implementaciones de los contratos de datos requeridos para el funcionamiento de las mismas.

3^{era} Iteración

El objetivo de esta iteración es desarrollar las Historias de Usuario 10, 11, 12 y 13, éstas son: “Obtener datos históricos de la entidad”, “Obtener datos de ubicación de la entidad”, “Obtener marcas nombradas para un tipo de medio técnico” y “Obtener modelos nombrados para una marca y tipo de medio técnico” respectivamente. Al final de esta iteración se deben tener, las funcionalidades propuestas en cada HU y las implementaciones de los contratos de datos requeridos para el funcionamiento de las mismas. Adicionalmente se deben generar los códigos de ejemplo para el consumo de todos los servicios a modo de documentación.

3.3 Pruebas del sistema

La metodología Oasis WSIM al igual que XP divide las pruebas del sistema en dos grupos:

- ✓ **Pruebas Unitarias.** Son diseñadas por los programadores, se encargan de las verificaciones en el código. Son utilizadas para comprobar si el sistema se comporta de la manera esperada.
- ✓ **Pruebas de Aceptación.** Se diseñan en base a las Historias de Usuario definidas por el cliente como parte del equipo de desarrollo con el objetivo de evaluar al final de cada iteración si se alcanzaron las metas propuestas por el cliente para cada funcionalidad.

Las pruebas de aceptación tienen más peso que las unitarias ya que constituyen un indicador de la satisfacción del cliente con la solución además de marcar el final de una iteración y el comienzo de la siguiente. Se recomienda que el cliente sea quien diseñe estas pruebas o que al menos participe de manera activa en el proceso.

Es posible establecer cierta similitud entre las pruebas de aceptación de WSIM y las pruebas de caja negra de RUP ya que ambas representan las salidas esperadas del sistema.

La garantía de calidad es una parte esencial en el proceso de XP. La realización de este tipo de pruebas y la publicación de los resultados debe ser los más rápido posibles, para que los desarrolladores puedan realizar con la mayor rapidez los cambios que sean necesarios.

Caso de prueba de aceptación	Código: HU01-P01
Nombre: Obtener datos de la entidad.	
Descripción: Prueba para la funcionalidad de obtener los datos de una entidad a partir del sello de uno de los medios técnicos que la componen.	
Condiciones: El sistema cliente debe estar autenticado.	
Entrada: Se envía el número de sello de un medio técnico.	
Resultado: Debe retornar la entidad a la cual pertenece el medio técnico especificado.	
Evaluación: Prueba satisfactoria	

Tabla 36: Caso de prueba de aceptación: HU01 -P01

Caso de prueba de aceptación	Código: HU02-P01
Nombre: Obtener datos del medio técnico.	
Descripción: Prueba para la funcionalidad de obtener los datos de un medio técnico a partir de su sello.	
Condiciones: El sistema cliente debe estar autenticado.	
Entrada: Se envía el número de sello de un medio técnico.	
Resultado: Debe retornar los datos asociados al medio técnico con el sello especificado.	
Evaluación: Prueba satisfactoria	

Tabla 37: Caso de prueba de aceptación: HU02-P01

Caso de prueba de aceptación	Código: HU03-P01
Nombre: Verificar si un sello existe.	
Descripción: Prueba para la funcionalidad de verificar la existencia de un sello determinado.	
Condiciones: Ninguna.	
Entrada: Se envía el número de sello a verificar. Se utilizará un sello existente.	
Resultado: Debe retornar verdadero.	
Evaluación: Prueba satisfactoria	

Tabla 38: Caso de prueba de aceptación: HU03-P01

Caso de prueba de aceptación	Código: HU03-P02
Nombre: Verificar si un sello existe.	
Descripción: Prueba para la funcionalidad de verificar la existencia de un sello determinado.	
Condiciones: Ninguna.	
Entrada: Se envía el número de sello a verificar. Se utilizará un sello que no exista.	
Resultado: Debe retornar falso.	
Evaluación: Prueba satisfactoria	

Tabla 39: Caso de prueba de aceptación: HU03-P02

Caso de prueba de aceptación		Código: HU04-P01
Nombre:	Verificar si un sello ha sido editado.	
Descripción:	Prueba para la funcionalidad de verificar si un número de sello determinado ha sido editado.	
Condiciones:	Ninguna.	
Entrada:	Se envía el número de sello de un medio técnico.	
Resultado:	Debe retornar verdadero si el sello fue editado.	
Evaluación:	Prueba satisfactoria	

Tabla 40: Caso de prueba de aceptación: HU04-P01

Caso de prueba de aceptación		Código: HU05-P01
Nombre:	Verificar si un sello ha sido asignado.	
Descripción:	Prueba para la funcionalidad de verificar si un determinado número de sello ha sido asignado a algún medio técnico.	
Condiciones:	Ninguna.	
Entrada:	Se envía el número de sello de un medio técnico.	
Resultado:	Debe retornar verdadero si el sello ha sido asignado.	
Evaluación:	Prueba satisfactoria	

Tabla 41: Caso de prueba de aceptación: HU05-P01

Caso de prueba de aceptación		Código: HU06-P01
Nombre:	Obtener listado de definiciones de entidad.	
Descripción:	Prueba para la funcionalidad de obtener un listado con los nombres de todas las definiciones de entidad existentes.	
Condiciones:	Ninguna.	
Entrada:	Ninguna.	
Resultado:	Debe retornar un listado con los nombres de las definiciones de entidad existentes.	
Evaluación:	Prueba satisfactoria	

Tabla 42: Caso de prueba de aceptación: HU06-P01

Caso de prueba de aceptación		Código: HU07-P01
Nombre:	Obtener los datos asociados a una definición de entidad	
Descripción:	Prueba para la funcionalidad de obtener los datos asociados a una definición de entidad y de los medios asociados a la misma a partir del nombre de la entidad.	
Condiciones:	Ninguna.	
Entrada:	Se envía el nombre de la entidad.	
Resultado:	Debe retornar un objeto de definición de entidad.	
Evaluación:	Prueba satisfactoria	

Tabla 43: Caso de prueba de aceptación: HU07-P01

Caso de prueba de aceptación		Código: HU08-P01
Nombre:	Obtener los datos asociados a la definición de datos de explotación de un tipo de Medio Técnico.	
Descripción:	Prueba para la funcionalidad de obtener los datos de explotación definidos para un Medio Técnico a partir de su nombre.	
Condiciones:	Ninguna.	
Entrada:	Se envía nombre del Medio Técnico.	
Resultado:	Debe retornar un listado de definiciones de los datos de explotación asociados al Medio Técnico.	
Evaluación:	Prueba satisfactoria	

Tabla 44: Caso de prueba de aceptación: HU08-P01

Caso de prueba de aceptación		Código: HU09-P01
Nombre:	Obtener datos de la entidad por id	
Descripción:	Prueba para la funcionalidad de obtener todos los datos asociados a una entidad partir del número de identificador raíz de la misma.	
Condiciones:	El sistema cliente debe estar autenticado.	
Entrada:	Se envía el identificador raíz de la entidad.	
Resultado:	Debe retornar la entidad registrada bajo el identificador raíz dado.	
Evaluación:	Prueba satisfactoria	

Tabla 45: Caso de prueba de aceptación: HU09-P01

Caso de prueba de aceptación		Código: HU10-P01
Nombre:	Obtener datos históricos de la entidad	
Descripción:	Prueba para la funcionalidad de obtener los datos históricos de una entidad a partir de su identificador raíz.	
Condiciones:	El sistema cliente debe estar autenticado.	
Entrada:	Se envía el identificador raíz de la entidad.	
Resultado:	Debe retornar un listado histórico de la entidad registrada bajo el identificador dado.	
Evaluación:	Prueba satisfactoria	

Tabla 46: Caso de prueba de aceptación: HU10-P01

Caso de prueba de aceptación		Código: HU11-P01
Nombre:	Obtener datos de ubicación de la entidad	
Descripción:	Prueba para la funcionalidad de obtener los de ubicación de una entidad.	
Condiciones:	El sistema cliente debe estar autenticado.	
Entrada:	Se envía el identificador raíz de la entidad.	
Resultado:	Debe retornar los datos de ubicación de la entidad registrada bajo el identificador dado.	
Evaluación:	Prueba satisfactoria	

Tabla 47: Caso de prueba de aceptación: HU11-P01

Caso de prueba de aceptación		Código: HU12-P01
Nombre:	Obtener marcas nomencladas para un tipo de medio técnico	
Descripción:	Prueba para la funcionalidad de obtener un listado de marcas nomencladas para un tipo de medio técnico.	
Condiciones:	Ninguna.	
Entrada:	Se envía el nombre del Medio Técnico.	
Resultado:	Debe retornar un listado de marcas.	
Evaluación:	Prueba satisfactoria	

Tabla 48: Caso de prueba de aceptación: HU12-P01

Caso de prueba de aceptación		Código: HU13-P01
Nombre:	Obtener modelos nombrados para una marca y tipo de medio técnico	
Descripción:	Prueba para la funcionalidad de obtener un listado de los modelos normalizados para los medios de un tipo y marca específicos.	
Condiciones:	Ninguna.	
Entrada:	Se envía el tipo de medio y la marca.	
Resultado:	Debe retornar un listado de modelos.	
Evaluación:	Prueba satisfactoria	

Tabla 49: Caso de prueba de aceptación: HU13-P01

Caso de prueba de aceptación		Código: HU14-P01
Nombre:	Obtener listado de usos para un tipo de entidad	
Descripción:	Prueba para la funcionalidad de obtener un listado de usos para un tipo de entidad	
Condiciones:	Ninguna.	
Entrada:	Se envía el tipo de entidad.	
Resultado:	Debe retornar un listado de usos asociados a la definición de la entidad dada.	
Evaluación:	Prueba satisfactoria	

Tabla 50: Caso de prueba de aceptación: HU14-P01

Conclusiones del capítulo

En este capítulo se describieron los principios de la orientación a servicios así como algunos conceptos específicos a tener en cuenta para la solución.

Quedaron definidos los contratos de interfaz y de datos de cada uno de los Servicios a implementar.

Empleando tarjetas CRC, se describieron todas las clases del sistema agrupándolas por librerías y finalmente representándolas en un esquema que describe la estructura del mismo.

Se hizo una descripción de los pasos a seguir para la implementación en cada una de las iteraciones y finalmente quedaron documentadas las pruebas de aceptación que validan como finalizadas las Historias de Usuario propuestas por el cliente.

CONCLUSIONES

A lo largo de la investigación se han ido describiendo todos los pasos seguidos para darle cumplimiento al objetivo de desarrollar un Servicio Web que permita centralizar, estandarizar y ofrecer información detallada de las Tics que se explotan en el Ministerio del Interior.

Se analizó la base teórica del diseño y desarrollo de Servicios Web así como el impacto que tienen estos para el Ministerio del Interior en la integración de las aplicaciones que automatizan los diferentes flujos de trabajos.

Quedaron definidas las funcionalidades que garantizan el intercambio de información asociada al control de las Tics entre el Sistema conTics y el resto de las aplicaciones que lo necesiten.

Finalmente se desarrolló la solución propuesta siguiendo los pasos recomendados por la metodología Oasis WSIM en su versión basada en XP dando como resultado una capa de Servicios Web completamente funcional que permite centralizar, estandarizar y ofrecer información detallada de de las Tecnologías de Informática y Comunicaciones que se explotan en el Ministerio del Interior.

RECOMENDACIONES

A partir de los resultados o beneficios que proporciona este trabajo de diploma, se proponen las siguientes recomendaciones.

- ✓ Que este material sea tomado como fuente de consulta por el personal que vaya a desarrollar un sistema similar.
- ✓ Implementar un catálogo UDDI en la red del Ministerio del Interior para facilitar el descubrimiento y explotación de los Servicios Web que se implementen dentro de la misma.
- ✓ Hacer un levantamiento de los procesos a automatizar dentro del Ministerio del Interior que puedan aprovechar las funcionalidades ofrecidas por el Servicio Web del Sistema de Control de Tecnologías de Informática y Comunicaciones.

REFERENCIAS BIBLIOGRÁFICAS

1. **World Wide Web Consortium.** Web Services Glossary. *W3C Working Group Note*. [En línea] 11 de Febrero de 2004. [Citado el: 29 de Enero de 2009.] <http://www.w3c.org/TR/ws-gloss/>.
2. **Charles, L., H. Joseph, and M. Carl.** *Marketing Internacional - Séptima Edición*. 2004. 666.
3. **Center for Maximum Public Performance.** Information Technology Infrastructure Library Glossary. [En línea] 2006. [Citado el: 2 de Febrero de 2009.] <http://www.cmpn.net/ITIL/ITILConf.aspx>.
4. **Cristalab.** Cristalab. *Web Services mediante WebServiceConnector de Flash*. [En línea] 17 de Diciembre de 2004. [Citado el: 12 de Febrero de 2009.] <http://www.cristalab.com/tutoriales/web-services-mediante-webserviceconnector-de-flash-c77/>.
5. **Katrib, Miguel, y otros.** *Visual Studio .NET 2008 desafía todos los retos...* Ciudad de la Habana : Editorial Capitán San Luis, 2008. 978-959-211-329-9.
6. **Pérez Valdés, Damián.** Maestros del Web. [En línea] 26 de Octubre de 2007. [Citado el: 18 de Febrero de 2009.] <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>.
7. **Microsoft Corporation.** Microsoft Developer Network. [En línea] Microsoft, 2008. [Citado el: 18 de Febrero de 2009.] <http://msdn.microsoft.com/es-ar/vstudio/products/bb931331.aspx>.
8. **Alvarez, S.** Desarrolloweb. *Sistema gestor de base de datos*. [En línea] 2006. [Citado el: 13 de Febrero de 2009.] <http://www.desarrolloweb.com/faq/504.php>.
9. **Pressman, R.S.** *Ingeniería del software. Un enfoque práctico*. 2002.
10. **Kruchten, P.** *A Software Development Process for a Team of One*. 2002.
11. **Shore and Warden.** *The Art of Agile Development - Primera Edición*. s.l. : O'Reilly, 2007.
12. **OASIS.** OASIS. *Web Service Implementation Methodology – Rational Unified Process (Example)*. [Online] 2006. [Cited: Febrero 18, 2009.] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=fws.

13. —. FWSI IM Case Example using XP. *Working Draft 03*. 2005.