

**APLICACION DE SUPERVISION DENTRO DEL MODELO DE
GESTION WBEM A TRAVES DE TERMINALES MOVILES WAP**



**JULIANA MARIA CITELLY MARTINEZ
ROGER EDUARDO CARVAJAL MUÑOZ**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERIA ELECTRONICA Y
TELECOMUNICACIONES
GRUPO DE NUEVAS TECNOLOGÍAS EN TELECOMUNICACIONES
POPAYAN
2001**

**APLICACION DE SUPERVISION DENTRO DEL MODELO DE
GESTION WBEM A TRAVES DE TERMINALES MOVILES WAP**



**JULIANA MARIA CITELLY MARTINEZ
ROGER EDUARDO CARVAJAL MUÑOZ**

**Monografía presentada como requisito
para optar al título de Ingeniero en
Electrónica y Telecomunicaciones**

DIRECTOR

ING. GUEFRY LEIDER AGREDO MENDEZ

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERIA ELECTRONICA Y
TELECOMUNICACIONES
GRUPO DE NUEVAS TECNOLOGÍAS EN TELECOMUNICACIONES
POPAYAN
2001**

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Nota de aceptación

Presidente del Jurado

Jurado

AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

Guefry Leider Agredo, Ingeniero Electrónico y Director del Trabajo de Grado, por su orientación en el desarrollo del trabajo con lo cual pudimos alcanzar los objetivos fijados, por la participación en la clarificación de inquietudes y motivación.

Giovanny López, Ingeniero Electrónico, por su ayuda en la concepción del proyecto.

Victor Quintero Florez, Ingeniero Electrónico y profesor de la Facultad, por su colaboración con el desarrollo de la monografía.

Alejandro Toledo, Ingeniero Electrónico, por su colaboración en la prestación de equipos de trabajo.

Diego Acosta, Ingeniero Electrónico, por su orientación con la cual pudimos optimizar nuestra aplicación y por facilitarnos los recursos con los cuales pudimos probar realmente nuestra aplicación.

Francisco Terán, Ingeniero Electrónico, por facilitarnos los recursos de la sala de telemática en donde pudimos desarrollar nuestra aplicación.

PREFACIO

El sector de las telecomunicaciones enfrenta en la actualidad rápidos y trascendentales cambios debido a factores tales como las nuevas tecnologías, la desregulación del mercado y mayores exigencias de los clientes en diversidad y calidad de los servicios. Esta situación representa para las diversas empresas que prestan servicios de telecomunicaciones grandes oportunidades pero de igual manera grandes retos.

Uno de los aspectos del negocio que es clave en la competitividad de estas empresas es la eficiente gestión de los servicios que prestan, de modo que obtengan la máxima satisfacción de sus clientes manteniendo márgenes de rentabilidad adecuados. Esta no es una tarea sencilla considerando que implica la gestión de una red de telecomunicaciones compuesta generalmente por equipos y sistemas de tecnologías y proveedores diversos, y además el intercambio de información con otras empresas en un escenario de múltiples servicios y múltiples proveedores.

La UIT elaboró a finales de los años ochenta un conjunto de normas denominadas TMN (Telecommunications Management Network) con el objeto de establecer una base conceptual, una arquitectura y unas guías que hicieran posible la construcción de una infraestructura para la gestión de redes de telecomunicaciones, mediante la interconexión de Sistemas de Operación y equipos de telecomunicación para el intercambio de información de gestión.

Los estándares de TMN abordan tres áreas fundamentales. La primera, una arquitectura con un modelo de niveles: negocio, servicios, red y elementos de red. La segunda, un modelo de información orientado a objetos que especifica nombres comunes y relaciones entre los ítems involucrados en un sistema de gestión. La tercera, un mecanismo de computación distribuida a través del cual se comunican y cooperan los componentes del sistema de gestión.

En tanto que el modelo de niveles TMN y el modelo de información son ampliamente aceptados y utilizados por las empresas operadoras de telecomunicaciones, no ha sucedido

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

igual con el sistema de computación distribuida propuesto. El desarrollo de éste, al igual que ha sucedido con otros estándares OSI, no ha avanzado mucho desde principios de los 90 mientras que nuevas tecnologías han entrado en escena ofreciendo soluciones efectivas a las necesidades de las aplicaciones distribuidas en entornos heterogéneos, y que cuentan además con un amplio respaldo entre fabricantes y usuarios.

Además se está planteando también a nivel mundial el tema de la Gestión Basada en la Web, que pretende aprovechar la amplia difusión de los navegadores como interfaz de usuario universal, para utilizarlos como interfaz para las aplicaciones de gestión.

El presente trabajo constituye el primer paso que da el Grupo de I+D en Nuevas Tecnologías en Telecomunicaciones en la línea de utilizar WMI y WAP en la implementación de aplicaciones de gestión de equipos. Su objetivo fundamental es desarrollar una aplicación de gestión a través de un dispositivo móvil WAP. Debido al gran potencial de WMI se podrá expandir el campo de gestión hacia otros dispositivos a través del uso de SNMP el cual es soportado por esta plataforma.

TABLA DE CONTENIDO

INTRODUCCION	1
1. GESTION EMPRESARIAL BASADA EN WEB - WBEM.....	3
1.1. Gestión Basada en Web (WBM).....	3
1.1.1 Definición	3
1.1.2 Estado del arte	3
1.1.3 Gestión basada en Web frente al modelo convencional de SNMP	4
1.1.3.1 Eficiencia	4
1.1.3.2 Escalabilidad	5
1.1.3.3 Nivel de abstracción de la funcionalidad e información	6
1.1.3.4 Nuevas funcionalidades	7
1.1.3.5 Costo.....	8
1.1.3.6 Interfaz de Usuario	8
1.1.3.7 Seguridad	8
1.2. La iniciativa WBEM	9
1.2.1. Contexto.....	9
1.2.1.1 Origen	9
1.2.1.2 DMTF	10
1.2.1.3 WBEM.....	10
1.2.2 El Modelo de Información Común (CIM)	11
1.2.2.1 Definición del CIM.....	11
1.2.2.2 Participantes.....	12
1.2.2.3 Estructura.....	12
1.2.3. Representación de CIM en XML	21
1.2.3.1 Introducción	21
1.2.3.2 Participantes.....	22
1.2.3.3 Traducción o mapeo de CIM en XML	23
1.2.3.4 Descripción de Elementos	24
1.2.4. CIM sobre HTTP.....	25
1.2.4.1 Introducción	25
1.2.4.2 Participantes.....	26
1.2.4.3 Sintaxis y Semántica de las Operaciones CIM	26
1.2.4.4 Encapsulado de las Operaciones CIM.....	28
1.2.4.5 Cabeceras Extendidas para CIM	29
2. MODELADO DE LA APLICACION DE SUPERVISION DENTRO DEL MODELO WBEM A TRAVES DE TERMINALES MOVILES WAP.	31
2.1 Introducción.....	31

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

2.2	Análisis de requerimientos	31
2.2.1	Definición y caracterización del sistema	31
2.2.2	Modelo del dominio del sistema	33
2.2.3	Definición del modelo de desarrollo específico	34
2.2.3.1	Modelos que describirán el sistema	34
2.2.3.2	Fundamentos metodológicos a utilizar	35
2.2.3.3	Modelo del proceso de desarrollo	35
2.2.4	Arbol de funciones.....	36
2.3	Análisis de la aplicación.....	39
2.3.1	Descripción de Casos de Uso Esenciales.....	39
2.3.1.1	Conexión Servidor.....	39
2.3.1.2	Verificación de usuario.....	41
2.3.1.3	Establecer conexión.....	44
2.3.1.4	Validación de usuario.....	47
2.3.1.5	Atender Solicitud.....	50
2.3.1.6	Seleccionar Datos.....	53
2.3.1.7	Enviar Datos.....	55
2.3.1.8	Generar páginas WML	56
2.3.1.9	Entregar Respuesta	58
2.4	Diseño de la aplicación.....	61
2.4.1	Clases de diseño	61
2.4.2	Diagrama subsistemas e interfaces.....	62
2.4.3	Diagrama de implantación	63
3.	CONCEPTOS FUNDAMENTALES PARA EL USO DE WMI	64
3.1	Introducción a WMI	64
3.2	Apreciación global de la arquitectura.....	64
3.3	Arquitectura de “WMI”	66
3.4	Aplicaciones de gestión.....	69
3.5	Proveedores “WMI”	72
3.6	Modelo de Información común (CIM)	74
3.7	WinMgmt.exe	75
3.7.1	Soporte para la Notificación de Evento	77
3.7.2	Soporte para el lenguaje de consulta	78
3.7.3	Soporte de seguridad	78
3.8	Descripción de los objetos “WMI” según el esquema WIN_32.....	79
3.8.1	SWbemLocator.....	79
3.8.1.1	SWbemLocator.ConnectServer.....	80
3.8.2	SWbemObjectSet.....	81
3.8.3	SWbemServices.....	82
3.8.3.1	SWbemServices.ExecQuery	84
3.8.4	SwbemObject	85

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

4. APLICACIONES Y ESQUEMAS DE ACCESO A LA INFORMACION DE GESTION.....	89
4.1 Aplicaciones.....	89
4.1.1 Soluciones de valor agregado.....	90
4.1.2 Aplicaciones con WMI	90
4.1.2.1 Hacer una lista de todos los servicios en el sistema.	91
4.1.2.2 Hacer una lista los servicios automáticos que están detenidos.	93
4.1.2.3 Hacer una lista de todas las particiones del disco con menos de 20 por ciento de espacio libre.	93
4.1.2.4 Estableciendo el retardo de arranque del sistema operativo.	94
4.1.2.5 Respaldando el registro de eventos.	96
4.1.2.6 Reiniciar una máquina remota.	98
4.1.2.7 Ejecutar el Bloc de Notas con WMI.....	99
4.1.2.8 Recolectando eventos del registro (log) de Windows NT en WMI.	100
4.1.2.9 Recolectando eventos sobre la utilización alta del CPU.	102
4.1.3 Avances de la tecnología inalámbrica	104
4.2 Esquemas de acceso a la información de gestión	105
4.2.1 WAP	105
4.2.2 Web.....	105
4.2.3 Visual Basic	106
 BIBLIOGRAFIA	 109
 GLOSARIO.....	 110
 ACRONIMOS	 113

LISTA DE FIGURAS

Figura 1.1 Ambiente de WBEM	11
Figura 1.2 Esquema UML de CIM	14
Figura 1.3 Area de Sistemas.	16
Figura 1.4 Area de Instrumentos.	17
Figura 1.5 Area de Redes	18
Figura 1.6 Area de Aplicación.	19
Figura 1.7 Area Física.	20
Figura 1.8 Representación de estadísticas	21
Figura 2.1 Entorno de aplicación	32
Figura 2.2 Dominio del sistema	34
Figura 2.3 Casos de uso.....	38
Figura 2.4 Clases de diseño	61
Figura 2.5 Diagrama de Subsistemas e Interfaces.....	62
Figura 2.6 Diagrama de Implantación.	63
Figura 3.1 Relación entre componentes WMI	68
Figura 3.2 Aplicaciones de gestión	71
Figura 4.1 Opciones de Arranque y Recuperación.	96
Figura 4.2 Interfaz Web.....	106
Figura 4.3 Interfaz Visual Basic	107

LISTA DE TABLAS

Tabla 1. Traducción de Tipos a elementos XML.....	28
Tabla 2. Cabeceras extendidas para CIM.	29
Tabla 3. Arbol de Funciones.....	37
Tabla 4. Caso de uso: Conexión Servidor	41
Tabla 5. Caso de uso: Verificación de Usuario.....	44
Tabla 6. Caso de uso: Establecer Conexión.....	47
Tabla 7. Caso de uso: Validación de Usuario.	50
Tabla 8. Caso de uso: Atender Solicitud.	53
Tabla 9. Caso de uso: Seleccionar Datos.....	55
Tabla 10. Caso de uso: Enviar Datos.	56
Tabla 11. Caso de uso: Generar páginas WML.	58
Tabla 12. Caso de uso: Entregar Respuesta.....	60
Tabla 13 Categorías de los Proveedores.....	74
Tabla 14 Método del objeto SWbemLocator.....	80
Tabla 15 Propiedad del objeto SWbemLocator	80
Tabla 16 Método del objeto SwbemObjectSet	82
Tabla 17 Propiedad del objeto SwbemObjectSet.....	82
Tabla 18 Métodos del objeto SWbemServices	84
Tabla 19 Propiedad del objeto SwbemServices	84
Tabla 20 Métodos del objeto SWbemObject.....	86
Tabla 21 Propiedades del objeto SWbemObject.....	87

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

INTRODUCCION

La gestión basada en web, es un nuevo paradigma que se va imponiendo cada vez más, puesto que el número de usuarios de la tecnología WWW aumenta cada día y por ello hay varias empresas dedicadas al desarrollo de productos que gestionan redes a través de internet.

En la actualidad, hay diferentes modelos que realizan este tipo de gestión, todos ellos están especificados en drafts de Internet. El modelo sobre el cual se va a desarrollar es WBEM (Web-Based Enterprise Management) más específicamente WMI (Windows Management Instrumentation). WMI es la plataforma de gestión WBEM de Microsoft Corporation, creada como integración y apoyo a la gestión de redes, equipos, sistemas operativos y periféricos. Siguiendo con la filosofía WBEM, WMI proporciona un acceso uniforme a todas las herramientas de gestión logrando una facilidad de uso y un aumento en el potencial en el desarrollo de estas herramientas.

La exigencia en comunicaciones que tienen las personas hoy en día, tienden a proporcionar una demanda en servicios de telecomunicaciones de datos que brinden un acceso seguro, rápido en cualquier momento y lugar, además de una excelente calidad. Las redes de comunicaciones inalámbricas brindan sus características para dar paso a la convergencia en servicios de comunicación de datos, y un ejemplo de ello es el estándar WAP, el cual da la oportunidad de que el cliente cuente con un gran número de servicios en su dispositivo portátil y que solamente se ven limitados por los proveedores de contenido de este tipo.

Hasta el momento la especificación WBEM (WMI) se enmarca dentro del ambiente de las conexiones cableadas, por lo cual la siguiente aplicación permitirá obtener un prototipo que muestre la posibilidad y las ventajas de poder cumplir, con una mayor propiedad, el cometido de gestión en cuestión a través de un terminal móvil soportándose en la plataforma del protocolo de aplicación inalámbrico – WAP.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Esta aplicación realiza un prototipo de gestión utilizando las tecnologías WBEM (WMI) y WAP el cual es una aproximación a los sistemas de gestión que se utilizarán en un futuro y que aún no se han implementado en Colombia.

El prototipo ofrece una interfaz interactiva soportada en formato web e integra software de Internet usado para gestión y conversión a formato WML para la presentación de la información acerca del dispositivo gestionado en un emulador de dispositivos móviles WAP.

La aplicación a desarrollar permitirá que el administrador pueda visualizar la información de gestión de un equipo remoto a través de un emulador de terminales WAP.

La siguiente monografía está estructurada en cuatro capítulos:

El primer capítulo introduce las tecnologías que conforman la base del desarrollo y su entendimiento es necesario para la asimilación del resto del documento. Se trata una introducción a WBEM, haciéndose énfasis solo en las características esenciales.

El segundo capítulo corresponde al capítulo central en el que se describe el diseño y el funcionamiento de la aplicación. En esta sección se encuentra el análisis de requerimientos, descripción de casos de uso e implementación, que se describen con la notación orientada a objetos definida por Rational Rose.

El tercer capítulo trata de la plataforma de gestión WMI en el cual se describe su funcionamiento y características así como los objetos utilizados para la supervisión en el desarrollo de la aplicación con su respectiva sintaxis.

El cuarto capítulo habla de los campos de aplicación de estas tecnologías (WMI y WAP) así como las aplicaciones adicionales desarrolladas en Visual Basic y a través de páginas Web.

1. GESTION EMPRESARIAL BASADA EN WEB - WBEM

1.1. Gestión Basada en Web (WBM)

1.1.1 Definición

Existen diferentes definiciones de lo que es gestión basada en Web, en cada RFC dedicado a los distintos modelos de este tipo de gestión viene una, pero se ha tomado como aplicación la tecnología World Wide Web a labores de gestión. Lo que significa la utilización del Protocolo de Transferencia de Hipertexto (HTTP-HiperText Transfer Protocol) entre servidores y clientes para el intercambio de información de gestión por medio del Lenguaje de Marcado de Hipertexto (HTML-HiperText Markup Language), texto-plano o código binario, así como la realización de operaciones de gestión tanto en máquinas como en redes.

1.1.2 Estado del arte

Este tipo de gestión se va imponiendo cada vez más, a medida que aumenta el número de usuarios de la tecnología de Red Mundial (WWW-World Wide Web) y por ello hay varias empresas dedicadas al desarrollo de productos que permitan gestionar redes a través de la Web permitiendo así la realización de la gestión en forma remota.

En la actualidad hay diferentes modelos que realizan este tipo de gestión, todos ellos están especificados en drafts de Internet. Los que se van a tratar son:

- Uso del Protocolo Simple de Gestión de Red (SNMP-Simple Network Management Protocol) y el Protocolo Común de Información de Gestión (CMIP-Common Management Information Protocol) sobre HTTP (Mediante formularios HTML o mapeando directamente en la URL)

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

- Gestión Empresarial Basada en Web (WBEM-Web-Based Enterprise Management)

1.1.3 Gestión basada en Web frente al modelo convencional de SNMP

Se analizarán las ventajas que WBEM presenta ante el modelo convencional SNMP, así como las limitaciones que tiene este tipo de gestión.

Como modelo convencional se ha elegido SNMPv1, ya que es el que mayor aceptación ha tenido y está siendo utilizado en la mayoría de los equipos actualmente.

Para realizar dicha comparación se analizarán distintos aspectos como: la eficiencia, escalabilidad, nivel de abstracción, expansibilidad, costo, interfaz de usuario, seguridad.

1.1.3.1 Eficiencia

SNMPv1

SNMP fue desarrollado para poder realizar la gestión de todos los equipos de una red TCP/IP, por lo cual el protocolo no implica que los equipos necesiten una alta capacidad de procesamiento. Debido a esto es necesario que el protocolo sea sencillo haciendo que SNMPv1 tenga algunas limitaciones.

Entre estas limitaciones está que no puede manejar grandes volúmenes de datos, por lo tanto para conseguir transferir grandes cantidades de datos es necesario realizar varias llamadas al agente para que este envíe la información requerida por el gestor. En un principio esto no suponía ningún tipo de problema, el problema surge cuando los retardos entre peticiones-respuestas son grandes, también se debe tener en cuenta el tiempo de procesamiento en el agente para interpretar y buscar la información requerida por el gestor. Si este tiempo es grande, puede que tanto las operaciones de gestión realizadas por el gestor, como las alarmas

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

generadas por los equipos gestionados no sean cursadas a tiempo, pudiendo tener graves consecuencias, como que se "caiga" la red, se estropee el equipo por sobrecalentamiento, etc. Es decir, que para que SNMPv1 funcione eficientemente, el retardo petición-respuesta entre el gestor y los equipos gestionados debe ser pequeño.

WBEM

Cuando se utiliza WBEM, lo que se puede hacer es utilizar un servidor HTTP que es capaz de recibir peticiones HTTP de gestión lo que permite que los equipos sean gestionados. Este servidor se coloca en la misma red que los agentes, así los retardos entre los gestores y los agentes son pequeños.

Cuando el gestor requiera información sobre un cierto parámetro, por ejemplo la temperatura, enviará una petición HTTP al servidor y lanzará dichas peticiones hacia los equipos gestionados. Las respuestas obtenidas serán enviadas directamente al Servidor.

La ineficiencia de HTTP viene dada por el propio protocolo, puesto que para la realización de cada petición-respuesta HTTP es necesaria establecer una nueva conexión TCP entre el servidor y el cliente.

1.1.3.2 Escalabilidad

SNMPv1

Los agentes SNMP no son capaces de llevar a cabo muchas peticiones SNMP simultáneamente, por lo que normalmente los equipos en los que están funcionando los agentes no son muy potentes. Este caso suele darse cuando un mismo agente está siendo gestionado por varios gestores.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

WBEM

Otra vez se encuentra con que el Servidor WBEM es una solución a este problema. Los gestores realizan las peticiones HTTP al servidor HTTP, si hay muchas peticiones será el servidor el que tenga mucha carga, pero no los agentes. Es decir, que el servidor debe estar en una máquina potente pero los agentes pueden seguir estando en máquinas sencillas y con menos capacidad de procesamiento. En caso de que las peticiones aumenten y se necesite un equipo más potente, solo hay que cambiar el servidor HTTP.

El servidor es el único que tiene control sobre el agente y por ello regula las peticiones-respuestas que le envía en cada momento. Si dos gestores piden la misma información en un período corto de tiempo, les envía a los dos la misma respuesta habiendo hecho una sola petición al agente.

1.1.3.3 Nivel de abstracción de la funcionalidad e información

SNMPv1

En la actualidad muchas LANs gestionadas mediante SNMP está utilizando una estación central, que gestiona a varios agentes. Esta estación debe ser una máquina bastante potente para poder controlar a todos los agentes. Si las plataformas de gestión SNMP fueran menos dependientes, este control centralizado se podría subdividir en varias máquinas, menos potentes, que realizarán diferentes tareas de gestión. Así como la posibilidad de utilizar herramientas de gestión de diferentes niveles, y que las de más alto nivel pudiesen utilizar las de más bajo nivel.

SNMP permite algo de gestión distribuida a través de agentes RMON.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

WBEM

A través del Servidor WBEM también se pueden realizar funciones de gestión distribuida. Aquí el cliente (gestor) hace peticiones HTTP, y el servidor puede realizar operaciones sencillas de gestión. El Servidor puede dar a los clientes información de distintos niveles de gestión, es decir, le puede proporcionar no solo datos del estado actual de un equipo, sino estadísticas del comportamiento de cierto(s) parámetro(s) a lo largo del tiempo, también le permite al gestor mediante una sola petición obtener grandes cantidades de datos.

En cuanto a gestión distribuida, WBEM no aporta grandes cambios frente a agentes RMON, aunque si hay algunas diferencias: HTTP permite diferentes tipos de información no solo SNMP. Además cualquier herramienta de gestión que utilice agentes RMON debe implementar también SNMP.

1.1.3.4 Nuevas funcionalidades

En muchas ocasiones adicionar nuevas funcionalidades a las herramientas ya existentes no es una tarea sencilla. Supone que hay que modificar el código fuente, y en muchos casos el programador requiere cambiar de lenguaje de programación.

Como se ha comentado el Servidor WBEM tiene una estructura que permite la realización de gestión distribuida. Las herramientas pueden ser más sencillas al requerir que realicen menos funciones, y por lo tanto será más fácil entender el código y añadir nuevas funcionalidades.

Además la utilización de HTTP, permite la utilización de CGI scripts, que pueden ser escritos en cualquier lenguaje de programación y el proceso que se comunica con el servidor a través del CGI también se puede programar en cualquier lenguaje. Ambos se pueden realizar mientras el servidor esté funcionando, y si hay otros CGI's funcionando seguirán haciéndolo independientemente del nuevo.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

1.1.3.5 Costo

Los modelos de gestión tradicional requieren un hardware bastante costoso para que funcionen correctamente. En el caso del Servidor WBEM, solo se necesita un PC medianamente potente. Además muchos vendedores de equipos incluyen en sus productos de hardware servidores HTTP que incluyen funcionalidades de gestión. Este tipo de gestión permite que el administrador pueda controlar remotamente los equipos a través de una página Web.

Para este tipo de gestión es importante tener una interfaz de usuario estandarizada, puesto que si cada fabricante suministra su propia interfaz de usuario el administrador tendría que aprender como funciona cada una, y este tipo de gestión no aportaría ninguna ventaja frente a los modelos tradicionales.

1.1.3.6 Interfaz de Usuario

Los sistemas de gestión tradicionales presentan una interfaz de usuario gráfica, ya que son más agradables y fáciles de manejar.

WBEM utiliza páginas Web como interfaz de usuario. Para ello puede utilizar HTML, applets de Java, php, etc. Esto permite tener unas interfaces de usuario que incluyan hipervínculos, formularios, tabla, etc.

1.1.3.7 Seguridad

SNMPv1

Sin duda el problema más importante de SNMP es la falta de seguridad, ya que solo se realiza autenticación de comunidad. La única forma de proteger nuestra red, es que no se pueda

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

utilizar SNMP desde afuera, es decir el control se ha de realizar desde el interior de la red. Por eso se dice que SNMPv1 es mejor para monitoreo que para control.

WBEM

El uso de HTTP soluciona este problema ya que en las respuestas HTTP solo se devuelve el estado del equipo, con lo que los clientes no pueden localizar los agentes por medio de esta respuesta HTTP.

Otra ventaja es que HTTP puede extenderse utilizando la Capa de Sockets de Seguridad (SSL-Security Sockets Layer), lo que permite tener más mecanismos de seguridad. El problema de HTTP es que utiliza una clave pública de encriptación, por lo que puede ser crackeada.

1.2. La iniciativa WBEM

1.2.1. Contexto

1.2.1.1 Origen

En Julio de 1996, el Grupo de Trabajo de Gestión Distribuida (DMTF- Distributed Management Task Force) puso de acuerdo a Sun, Microsoft y otros miembros en la aprobación del Modelo de Información Común (CIM-Common Information Model), un esquema que describe cómo construir un modelo orientado a información de objetos de gestión de red, que además de representar diferentes y múltiples objetos permite definir relaciones entre ellos. Este proyecto provenía del Esquema de Gestión HyperMedia (HMMS-HyperMedia Management Schema) de Microsoft, el cual a su vez era parte de otro más amplio denominado WBEM, también de Microsoft. WBEM estaba compuesto de un modelo de datos llamado Gestor de Objetos Hypermedia (HMOM-HyperMedia Object Manager) y un

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

protocolo basado en HTTP llamado Protocolo de Gestión Hipermedia (HMMP-HyperMedia Management Protocol) el cual fue rechazado en 1997 por el Grupo de Trabajo en Ingeniería de Internet (IETF-Internet Engineering Task Force).

1.2.1.2 DMTF

El DMTF es una organización industrial que lidera el desarrollo, adopción y unificación de estándares de gestión de red. Entre las empresas miembros que figuran en la especificación del CIM (versión 2.2), están: INTEL CORPORATION, MICROSOFT CORPORATION, DIGITAL EQUIPMENT CORPORATION, HEWLETT-PACKARD COMPANY, INTERNATIONAL BUSINESS MACHINES CORPORATION, NOVELL INC., SUN MICROSYSTEMS, INC., COMPAQ COMPUTER CORPORATION, DELL COMPUTER CORP., SYMANTEC, THE SANTA CRUZ OPERATION y NEC TECHNOLOGIES, INC.

1.2.1.3 WBEM

WBEM, del inglés Web-Based Enterprise Management, es una iniciativa basada en un conjunto de tecnologías estándares de gestión y de Internet desarrolladas para unificar la gestión de plataformas informáticas en entornos empresariales, a gran escala. WBEM proporciona a la industria el modo de desarrollar un conjunto integrado de herramientas de gestión basadas en estándares utilizando para ello tecnologías emergentes como CIM y el Lenguaje de Marcas Extendido (XML-eXtended Markup Language).

Pertenece a esta iniciativa el desarrollo del CIM por el DMTF con el apoyo de los grupos más importantes del sector.

La Figura 1.1 muestra el ambiente de WBEM:

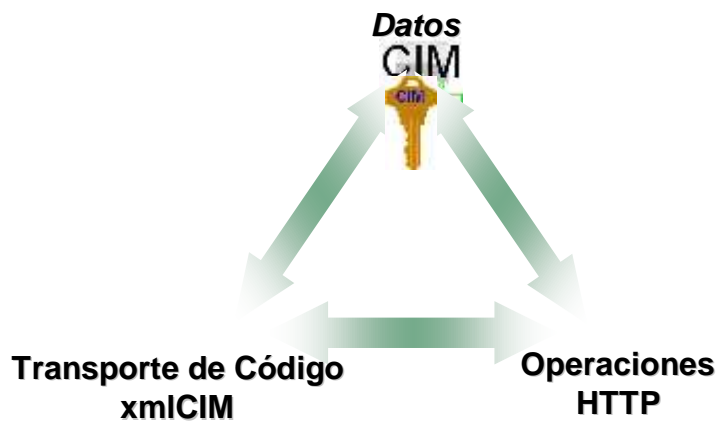


Figura 1.1 Ambiente de WBEM¹

1.2.2 El Modelo de Información Común (CIM)

1.2.2.1 Definición del CIM

El CIM es un modelo de datos genérico de un esquema de implementación neutral para describir información de gestión en un ambiente de red. CIM esta comprendido por una especificación y un esquema. La especificación define los detalles para la integración con otros modelos (por ejemplo las Bases de Información de Gestión (MIBs-Management Information Bases) de SNMP o el Formato de Información de Gestión (MIFs-Management Information Format) del DMTF) mientras que el esquema provee el modelo de descripción actual.²

El estándar CIM constituye un modelo de información para describir información de gestión y que se encuentra especificado en los siguientes documentos (disponibles en la web [DMTF] de forma gratuita):

¹ Fuente DMTF

² Fuente: Tutorial de CIM del DMTF

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

- Especificación de CIM v.2.2. (14 de Junio de 1999). Define los detalles de integración con otros modelos como las MIBs de SNMP.
- CIM Schema v2.2. Descripciones del modelo.
- XmlCIM v2.0 (codificación XML de CIM).
- Operaciones CIM sobre HTTP v1.0.

1.2.2.2 Participantes

Los participantes en el desarrollo del CIM (v.2.2) fueron las siguientes empresas:

- Compaq Computer Corporation
- Computer Associates Intl., Inc
- Hewlett-Packard Company
- Intel Corporation
- Microsoft Corporation
- Novell, Inc.
- Sun Microsystems, Inc.
- Tivoli Systems, Inc.

1.2.2.3 Estructura

El esquema de gestión proporcionado establece un marco conceptual común a nivel de topología fundamental tanto con respecto a la clasificación y asociación como con respecto a un conjunto básico de clases que pretenden establecer este marco común descriptivo del entorno gestionado.

El esquema CIM se estructura en tres niveles fundamentales y otro adicional para representación de datos estadísticos:

- Modelo Interno (Core Model)

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

- Modelo Común (Common Model)
 - Sistemas
 - Partes o Instrumentos (Devices)
 - Redes
 - Aplicaciones
 - Físico
- Esquemas de Extensión (Extension Schemas)
- Representación de Estadísticas

□ El Modelo Interno (Core Model)

En la figura 1.2 se puede observar el esquema en el Lenguaje Unificado de Modelamiento (UML-Unificated Model Language) del modelo interno del CIM. Las flechas azules denotan herencia, las líneas rojas denotan relaciones y las líneas verdes agregación.

Se puede observar cómo se define la clase CIM_ManagedSystemElement de la cual derivan dos tipos de elementos, los elementos físicos (CIM_PhysicalElement), que son los que ocupan un espacio físico y se rigen por las reglas de la física, y los elementos lógicos (CIM_LogicalElement) que son el resto. Además se puede ver la definición de asignaciones (CIM_Setting) y configuraciones (CIM_Configuration) como agregado de las anteriores. Ambas clases se asocian a los elementos gestionables.

En la mitad inferior se ve cómo los elementos físicos se pueden agrupar para formar un producto (CIM_Product) mientras que los lógicos derivan en sistemas (CIM_System), partes lógicas (como pudiera ser una partición de un disco duro físico) (CIM_LogicalDevice), servicios (CIM_Service) y puntos de acceso a servicios (CIM_ServiceAccessPoint).

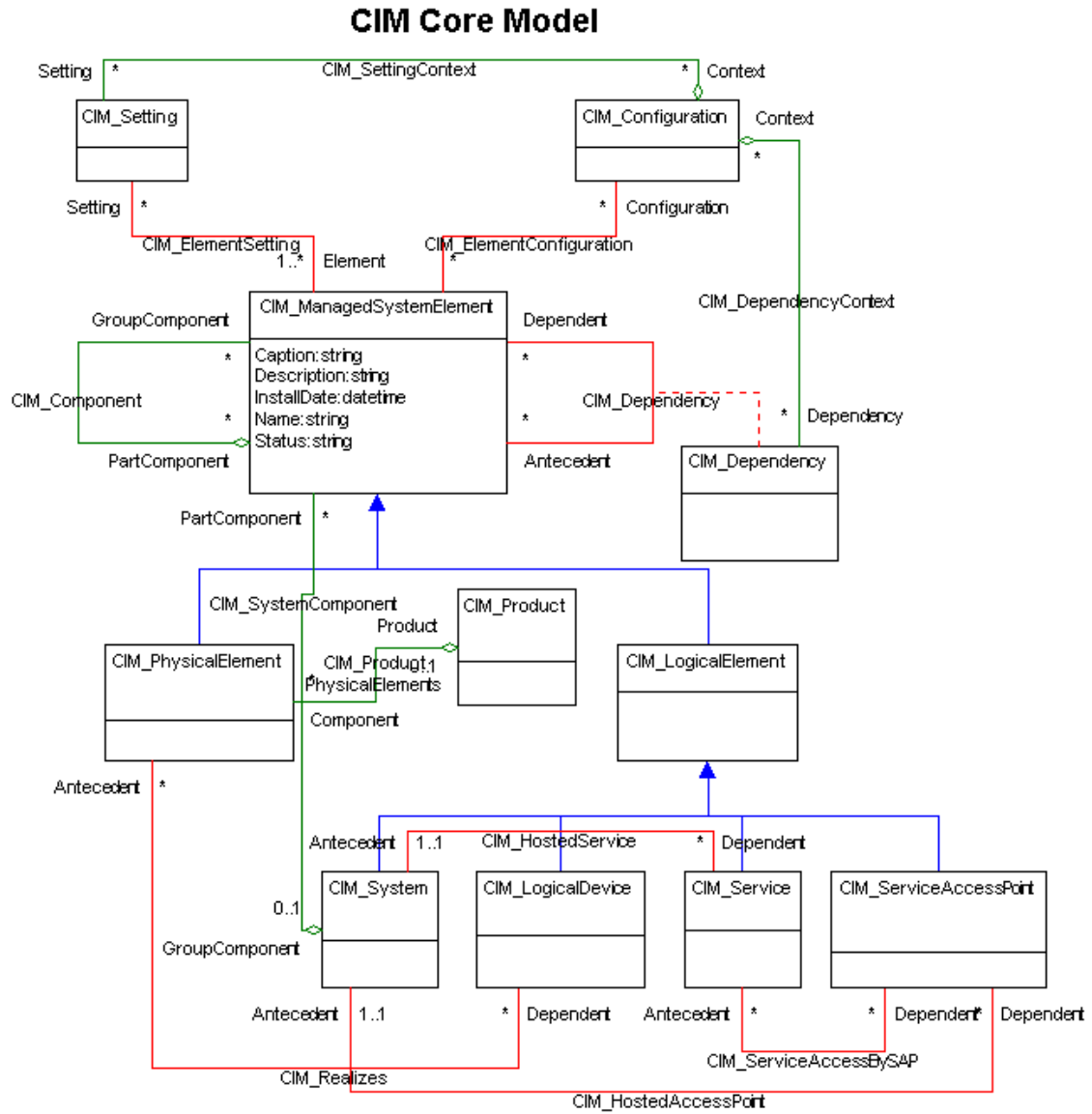


Figura 1.2 Esquema UML de CIM³

³ Fuente: Tutorial de CIM del DMTF

□ **El Modelo Común (Common Model)**

El Modelo Común es un modelo de información que captura nociones que son comunes a ciertas áreas particulares de gestión e independientes de una tecnología o implementación particular. Este modelo proporciona un conjunto de clases base para extenderlas ya en esquemas específicos de un área o tecnología. El Modelo Interior y el Modelo Común están expresados de manera conjunta en el Esquema CIM (CIM Schema) adjunto a la especificación.

Las áreas definidas en el modelo son las siguientes:

Sistemas: principalmente, modelan sistemas informáticos (computadores), sistemas de aplicaciones y sistemas de redes, como se observa en la Figura 1.3.

Common Model: Systems

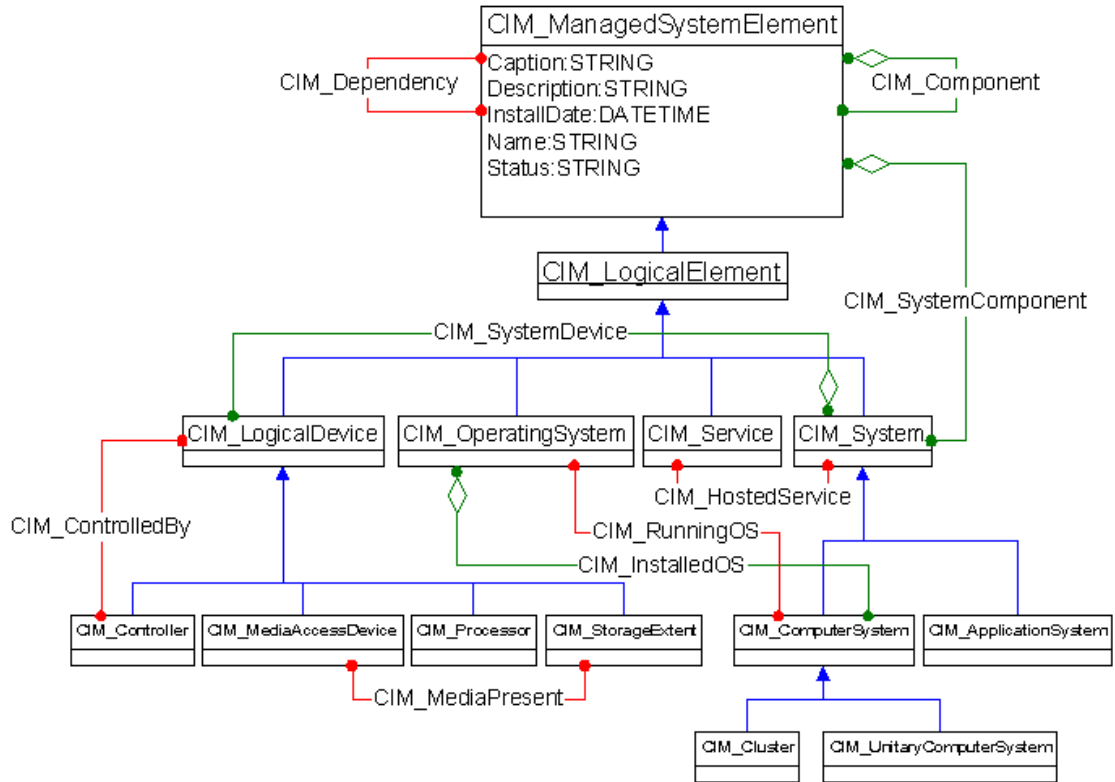


Figura 1.3 Area de Sistemas.⁴

Partes (Devices): En la Figura 1.4 se observa que las partes son una representación de unidades lógicas discretas de un sistema. Existe la tentación de asignarlas a componentes físicos, pero lo gestionado es la representación que el sistema operativo hace de él. Cada unidad proporciona una funcionalidad por sí misma. Derivan de esta clase el procesador, la fuente, la unidad de almacenamiento, etc.

⁴ Fuente: Tutorial de CIM del DMTF

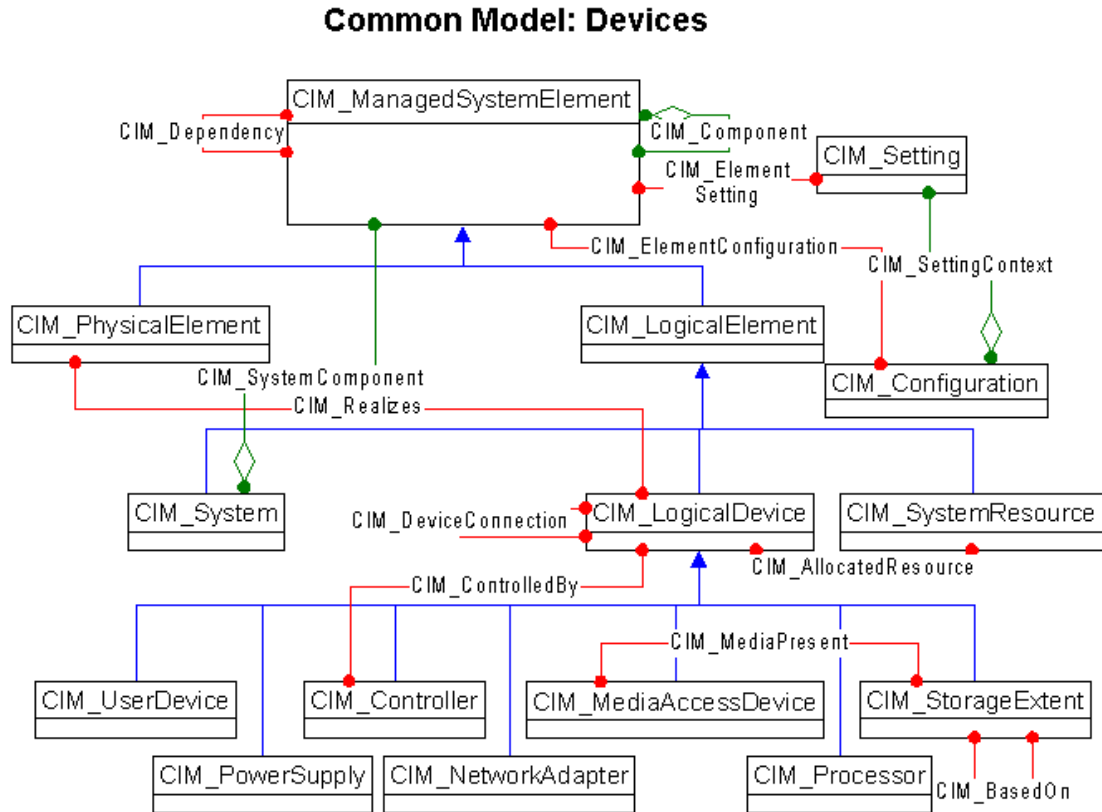


Figura 1.4 Area de Instrumentos.⁵

Redes: Representa características de un entorno en red como la topología, la conectividad, los protocolos y servicios necesarios para dar acceso a la red como se observa en la Figura 1.5.

⁵ Fuente: Tutorial de CIM del DMTF

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

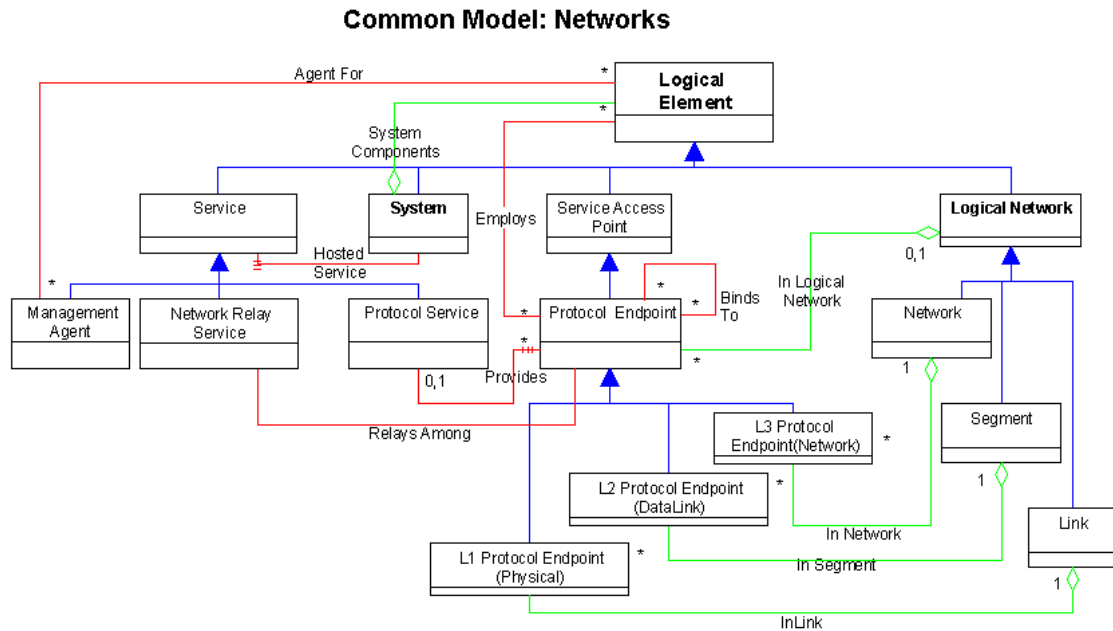


Figura 1.5 Area de Redes⁶

Aplicaciones: La Figura 1.6 muestra El Modelo de Gestión de Aplicaciones el cual es un modelo diseñado para describir los detalles requeridos para gestionar el software instalado. Una característica fundamental del modelo es la idea de ciclo de vida de la aplicación, que puede estar un uno de estos cuatro estados: disponible, instalable, ejecutable o ejecutándose.

⁶ Fuente: Tutorial de CIM del DMTF

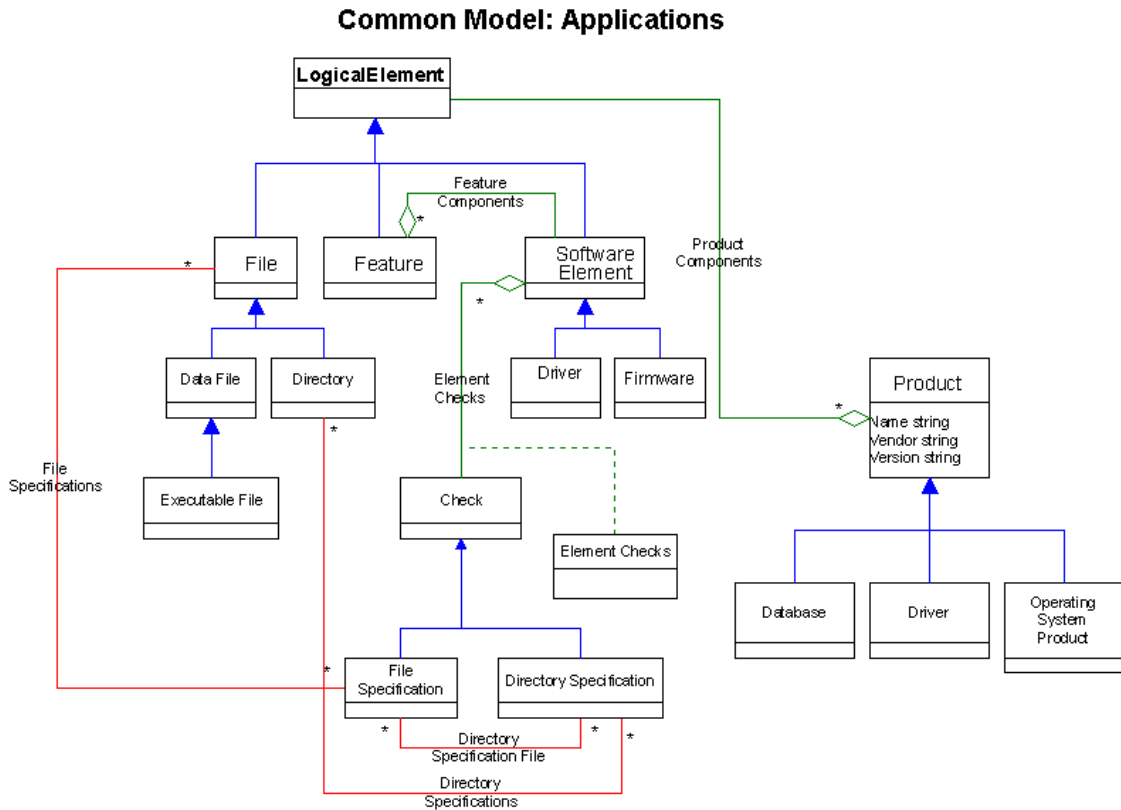


Figura 1.6 Area de Aplicación.⁷

Física: Proporciona el modelado del entorno físico real como se observa en la Figura 1.7. Sin embargo, la mayoría del entorno gestionado se representa por entidades lógicas más que por entidades físicas, ya que éstas son difíciles de controlar y difieren dramáticamente de sistema en sistema conforme avanza la tecnología.

⁷ Fuente: Tutorial de CIM del DMTF

Common Model: Physical

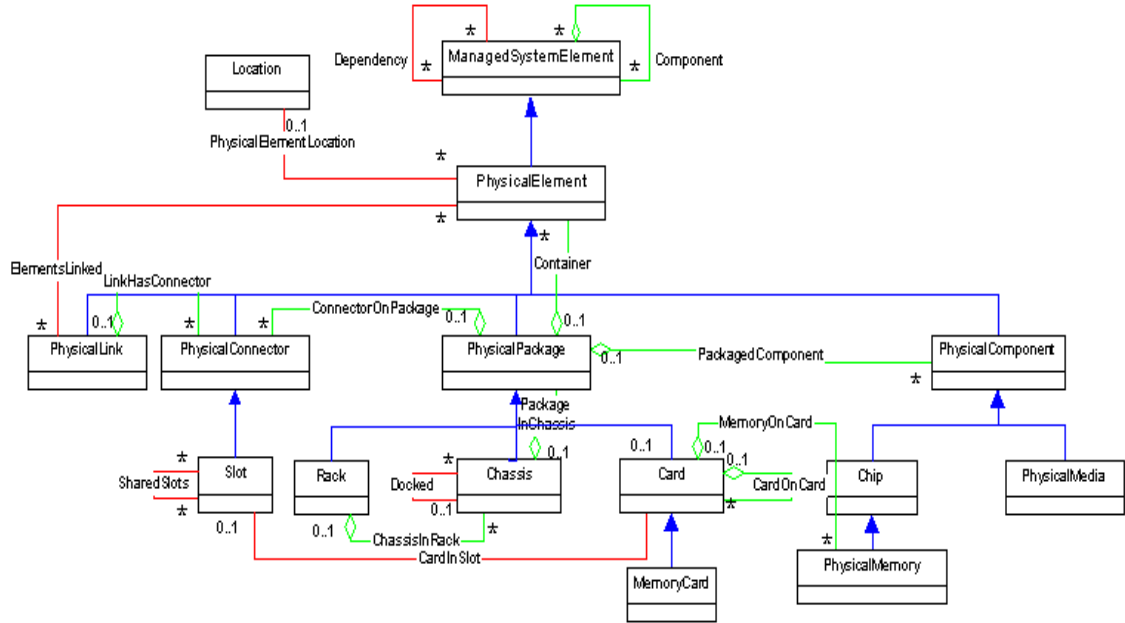


Figura 1.7 Area Física.⁸

❑ Esquemas de Extensión (Extension Schemas)

El esquema propuesto por CIM puede ser extendido para cubrir las necesidades particulares de un entorno de red, siendo este modelo enriquecido con clases adicionales que representan objetos gestionables de una tecnología determinada. Se crea entonces una extensión del modelo común de CIM que puede ser aprobada por el DMTF.

Para especificar un esquema se utiliza el Formato de Objetos Gestionados (MOF-Managed Object Format) que es un lenguaje creado por el DMTF basado en el Lenguaje para Definición de Interfaces (IDL-Interface Definition Language).

⁸ Fuente: Tutorial de CIM del DMTF

□ **Representación de Estadísticas**

El modelo CIM incluye la definición de una clase para manejar estadísticas que se relaciona con la clase genérica de elementos gestionables (CIM_ManagedSystemElement).

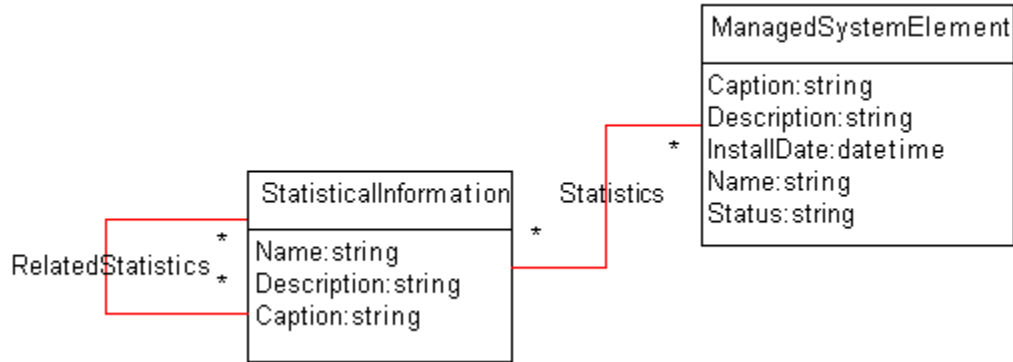


Figura 1.8 Representación Estadística.

1.2.3. Representación de CIM en XML

1.2.3.1 Introducción

El modelo CIM constituye una arquitectura orientada a objetos para almacenar información sobre sistemas gestionables. Era necesaria la representación de los elementos y mensajes CIM para su transmisión sobre HTTP mediante algún tipo de codificación, y se eligió XML por ser un estándar del W3C que permite representar información de forma extensible, potente, que además fue diseñado para su uso sobre HTTP (texto plano) y con una amplia aceptación.

Para permitir la interoperabilidad entre distintas implementaciones el DMTF redactó una especificación con título "Specification for the Representation of CIM in XML" a partir de la que se ha elaborado este apartado (en concreto, a partir de la versión 2.0, publicada el 20 de Julio de 1999).

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

La especificación intenta utilizar tecnologías ampliamente estandarizadas y se da prioridad a la completitud sobre la concisión, de forma que se disponga la información necesaria para ser transformados en otros formatos como MOF mediante tecnologías estandarizadas de representación de documentos XML como la Hoja de Estilo en Cascada (CSS-Cascading Style Sheet) o el Lenguaje de Hoja de Estilo Extensible (XSL-eXtensible Sheet Language). Sin embargo, CSS no es suficientemente potente y XSL está en desarrollo, por lo que no está estandarizada la hoja de estilo correspondiente.

Microsoft ha creado una herramienta de gestión llamada Instrumental para Gestión Windows (WMI-Windows Management Instrumentation), la cual está desarrollada para el sistema operativo Windows encontrándose incluida en las versiones más recientes. WMI se basa en el modelo CIM a través del cual la información de gestión es capturada, almacenada y enviada, si es necesario, a un equipo remoto.

Todas las operaciones con las cuales se accede a la información de gestión y la manera como ésta es almacenada y enviada se tratará más detalladamente en el siguiente capítulo.

1.2.3.2 Participantes

Las empresas que participaron en el comité del DMTF que elaboró esta especificación entre Julio de 1998 y Julio de 1999 (CIM TDC XML Sub-Committee) fueron las siguientes:

- Agranat Systems
- Hewlett-Packard Company
- IBM Corporation
- Microsoft Corporation
- Tivoli Systems, Inc.
- Customer Support Consortium
- Sun Microsystems, Inc.

- Intel Corporation

1.2.3.3 Traducción o mapeo de CIM en XML

En la especificación se explican dos modos posibles para traducir CIM a XML. Una posibilidad era especificar un esquema que describiera las clases CIM, y las instancias CIM se codificarían en XML siguiendo tal esquema. Sin embargo, a pesar de ser esta una solución intuitiva y simple, lo que define la especificación es un esquema que describe de manera general la noción de clase CIM o instancia. Entonces los nombres de elementos CIM se traducen a valores o atributos de elementos XML, y no a elementos XML. Las razones son cuatro:

- Sólo requiere una Definición de Tipo de Documento (DTD-Document Type Definition) estándar, y no un ilimitado número de ellos.
- Un DTD no permite una lista desordenada de elementos, lo que hubiese obligado en el otro modelo a fijar un orden arbitrario en las listas de propiedades, métodos y calificadores, o bien a definir un DTD que tuviese en cuenta explícitamente todos los órdenes posibles.
- En una traducción directa de instancias a elementos XML, el número de éstos hubiese hecho necesario el uso de dominios de nombres (namespaces) para replicar reglas con varios problemas asociados.
- Aunque una traducción del primer tipo hubiera permitido la validación de instancias XML sobre clases, esto sólo hubiera sido posible si la jerarquía entera hubiese sido unificada antes de pasar la clase CIM a esquema XML. De otra forma las propiedades estarían ausentes del DTD y la validación fallaría contra una instancia que incluyera valores de propiedades heredadas.

1.2.3.4 Descripción de Elementos

Tras describir en primer lugar nueve (9) entidades de parámetros utilizadas en el vocabulario del esquema, el DTD describe una serie de elementos que se comentan a continuación.

- **Elementos de Alto Nivel**

El elemento raíz de un documento XML es el elemento CIM. Este elemento puede contener un solo mensaje (MESSAGE) o una declaración (DECLARATION).

- **Elementos Declarativos**

El elemento DECLARATION a su vez puede contener los elementos DECLGROUP, DECLGROUP.WITHNAME, DECLGROUP.WITHPATH para declarar clases, instancias y calificadores con QUALIFIER.DECLARATION.

- **Elementos de Valores**

Los elementos de valores definen aquellos elementos del Schema que expresan el valor de los objetos CIM. El de más alto nivel es VALUE que puede englobar a VALUE.REFERENCE, VALUE.REFARRAY, VALUE.OBJECT, y otros.

- **Elementos de Nombrado y Localización**

Son aquellos que expresan el nombre y localización de objetos CIM, por ejemplo: NAMESPACEPATH, LOCALNAMESPACEPATH, HOST, NAMESPACE, CLASSPATH.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

□ Elementos de Definición de Objetos

Están relacionados con la expresión de la definición de objetos CIM (clases, instancias, propiedades, métodos y calificadores (qualifiers). Existen los elementos CLASS, INSTANCE, QUALIFIER, PROPERTY, METHOD, PARAMETER.

□ Elementos de Mensajes (Operaciones, Invocación de Métodos, Retornos)

Los elementos de mensajes son aquellos que sirven para crear mensajes CIM y luego ser enviados sobre HTTP. El de más alto nivel es el elemento MESSAGE que puede contener una petición, con SIMPLEREQ o MULTIREQ, o una respuesta, con SIMPLERSP o MULTIRSP. Las peticiones, a su vez, contienen invocaciones a métodos extrínsecos e intrínsecos con (I)METHODCALL, pasando parámetros con (I)PARAMVALUE, y las respuestas contienen (I)METHODRESPONSE. Por último, éstos contienen valores de retorno con (I)RETURNVALUE o errores con ERROR.

1.2.4. CIM sobre HTTP

1.2.4.1 Introducción

Pertenciente a la misma iniciativa WBEM y desarrollada con la colaboración de importantes corporaciones del sector de las telecomunicaciones, el DMTF publicó con fecha de 11 de Agosto de 1999 el documento "Specification for CIM Operations over HTTP Versión 1.0."

WMI realiza internamente las operaciones sobre HTTP por lo que en éste documento se tratará la codificación sobre HTTP de una manera general para el interés de todo aquel que quiera saber cómo se realiza esta operación.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

La traducción de operaciones CIM sobre HTTP permite a las implementaciones de CIM interoperar de una manera abierta y estandarizada. Para ello utiliza el CIM XML DTD que define el esquema XML para objetos y mensajes CIM.

La especificación está estructurada en cuatro secciones: la primera es una introducción breve, la segunda es la descripción de la sintaxis y semántica de las operaciones, la tercera describe el encapsulado de estas operaciones mediante XML y la cuarta profundiza en ciertos detalles del encapsulado y consideraciones de seguridad.

Siguiendo el mismo orden, se presenta a continuación cómo son las operaciones que se manejan y cómo se encapsulan para ser transmitidas como cuerpo de un paquete HTTP.

1.2.4.2 Participantes

Las empresas que participaron en el comité del DMTF que elaboró esta especificación durante el verano de 1999 (CIM TC XML Sub-Committee) coinciden con las participantes en la especificación para la representación de CIM en XML.

1.2.4.3 Sintaxis y Semántica de las Operaciones CIM

Las operaciones CIM sirven para preguntar o modificar la información de gestión modelada según el modelo CIM, es decir, en forma de clases. Ello se hace mediante documentos en formato XML que se validan con el CIM XML DTD. En XML se distingue entre documentos bien formados (podría existir un DTD para ellos porque siguen las reglas generales de construcción de un documento en XML) y los que además son válidos para un DTD determinado. En la especificación, además, se introduce una definición entre las dos anteriores, la del documento débilmente válido, que se aplica a aquellos documentos que, quitándoles los atributos o elementos que no aparecen en el DTD, se convierten en documentos válidos. Al permitir a los documentos este tipo de validación se hace posible la compatibilidad hacia atrás de versiones futuras enriquecidas.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Los documentos tienen como elemento raíz <CIM>. Dentro está el sub-elemento <MESSAGE> que a su vez incluye uno de los elementos siguientes: <SIMPLEREQ>, <MULTIREQ>, <SIMPLERSP>, <MULTIRSP>. Las operaciones CIM son invocaciones de métodos que pueden ser intrínsecos o extrínsecos según pertenezcan a la especificación comentada aquí o sean métodos pertenecientes a clases del esquema CIM, respectivamente.

□ **Métodos Intrínsecos**

Los métodos intrínsecos definidos en la especificación (versión 1.0) son: GetClass, GetInstance, DeleteClass, DeleteInstance, CreateClass, CreateInstance, ModifyClass, ModifyInstance, EnumerateClasses, EnumerateClassNames, EnumerateInstances, EnumerateInstanceNames, ExecQuery, Associators, AssociatorNames, References, ReferencesNames, GetProperty, SetProperty, GetQualifier, SetQualifier, DeleteQualifier y EnumerateQualifiers.

Este es un ejemplo de cómo define la especificación un método intrínseco:

```
GetClass <class> GetClass (
  [IN] <className> ClassName,
  [IN,OPTIONAL] boolean LocalOnly = true,
  [IN,OPTIONAL] boolean IncludeQualifiers = true,
  [IN,OPTIONAL] boolean IncludeClassOrigin = false,
  [IN,OPTIONAL,NULL] string PropertyList [] = NULL
)
```

Y a continuación la manera de traducir los tipos a elementos XML:

Tipo	Elemento XML
<object>	(VALUE.OBJECT VALUE.OBJECTWITHLOCALPATH VALUE.OBJECTWITHPATH)

<class>	CLASS
<instance>	INSTANCE
<className>	CLASSNAME
<namedInstance>	VALUE.NAMEDINSTANCE
<instanceName>	INSTANCENAME
<objectWithPath>	VALUE.OBJECTWITHPATH
<objectName>	(CLASSNAME INSTANCENAME)
<propertyValue>	(VALUE VALUE.ARRAY VALUE.REFERENCE)
<qualifierDecl>	QUALIFIER.DECLARATION

Tabla 1. Traducción de Tipos a elementos XML

□ **Métodos Extrínsecos**

Los servidores CIM deben soportar la llamada a los métodos extrínsecos que figuran en el esquema CIM que soporta. Si no, deben retornar el código de error CIM_ERR_NOT_SUPPORTED. Como éste, se definen en la especificación un total de 17 códigos de error.

1.2.4.4 Encapsulado de las Operaciones CIM

Las operaciones CIM se realizan utilizando el método POST o M-POST de HTTP, prefiriéndose este último. De hecho, la norma obliga a intentar primero la comunicación con M-POST y permite, como mecanismo de recuperación cuando el entorno no lo permite, el uso de POST.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

1.2.4.5 Cabeceras Extendidas para CIM

Cualquier petición de operación CIM o respuesta de operación CIM debe incluir la cabecera:

CIMOperation.

Además, en el caso de peticiones incluirán una y sólo una de estas tres:

CIMMethod

CIMObject

CIMBatch

Las respuestas pueden incluir las cabeceras:

CIMError (sólo en caso de error)

CIMVersion

Ejemplo:

Cabecera M-POST	Cabecera POST
M-POST /cimom HTTP/1.1 Man: http://www.dmtf.org./cim/mapping/http/v1.0 ; ns=23 23-CIMOperation: MethodCall	POST /cimom HTTP/1.1 CIMOperation: MethodCall ...

Tabla 2. Cabeceras extendidas para CIM.

Y en lo que respecta al cuerpo del mensaje, consiste en un documento XML parcialmente válido con el elemento raíz <CIM>, tal y como se explicaba anteriormente, que contiene un elemento <MESSAGE> que a su vez contiene uno de los cuatro elementos vistos más arriba. A partir de aquí, se codifica en XML según la especificación [DMTF-04] la operación CIM

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

de petición o de respuesta, con la traducción de tipos vista anteriormente, como en el ejemplo siguiente:

```
<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="87872" PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="GetClass">
<IRETURNVALUE>
<CLASS NAME="CIM_VideoBIOSElement"
SUPERCLASS="CIM_SoftwareElement">
</CLASS>
</IRETURNVALUE></IMETHODRESPONSE> </SIMPLERSP></MESSAGE></CIM>
```

En este capítulo se vieron las características y ventajas del estándar WBEM frente a los modelos de gestión tradicionales como SNMP, además se hizo referencia al Modelo de Información Común (CIM) sobre el lenguaje XML. A partir de estos conceptos se desarrollo un modelado de la aplicación siguiendo el esquema planteado por Rational Rose el cual se tratará en el capítulo siguiente.

2. MODELADO DE LA APLICACION DE SUPERVISION DENTRO DEL MODELO WBEM A TRAVES DE TERMINALES MOVILES WAP.

2.1 Introducción

Con el propósito de visualizar de una manera mas detallada los componentes y procesos que hacen parte de la aplicación, se escogió la metodología del Proceso Unificado de Rational (RUP-Rational Unified Process) que permite realizar un modelado de fácil entendimiento e independiente de el lenguaje de programación que se utilice para llevar a cabo la aplicación.

En este capítulo se consignan los resultados más significativos para el desarrollo del proyecto.

2.2 Análisis de requerimientos

2.2.1 Definición y caracterización del sistema

La aplicación ofrece una interfaz interactiva soportada en formato Web e integra software de Internet usado para gestión y conversión a formato WML para la presentación de la información acerca del dispositivo gestionado en un emulador de dispositivos móviles WAP.

El desarrollo trabaja la temática de gestión, específicamente la soportada en la tecnología WBEM (WMI) y WAP.

La aplicación de la que trata este documento permite que el administrador pueda visualizar la información de gestión de un equipo remoto a través de un emulador de terminales WAP.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Especificaciones Funcionales

El diagrama muestra el ambiente y los procesos posibles entre el administrador y el equipo a gestionar.

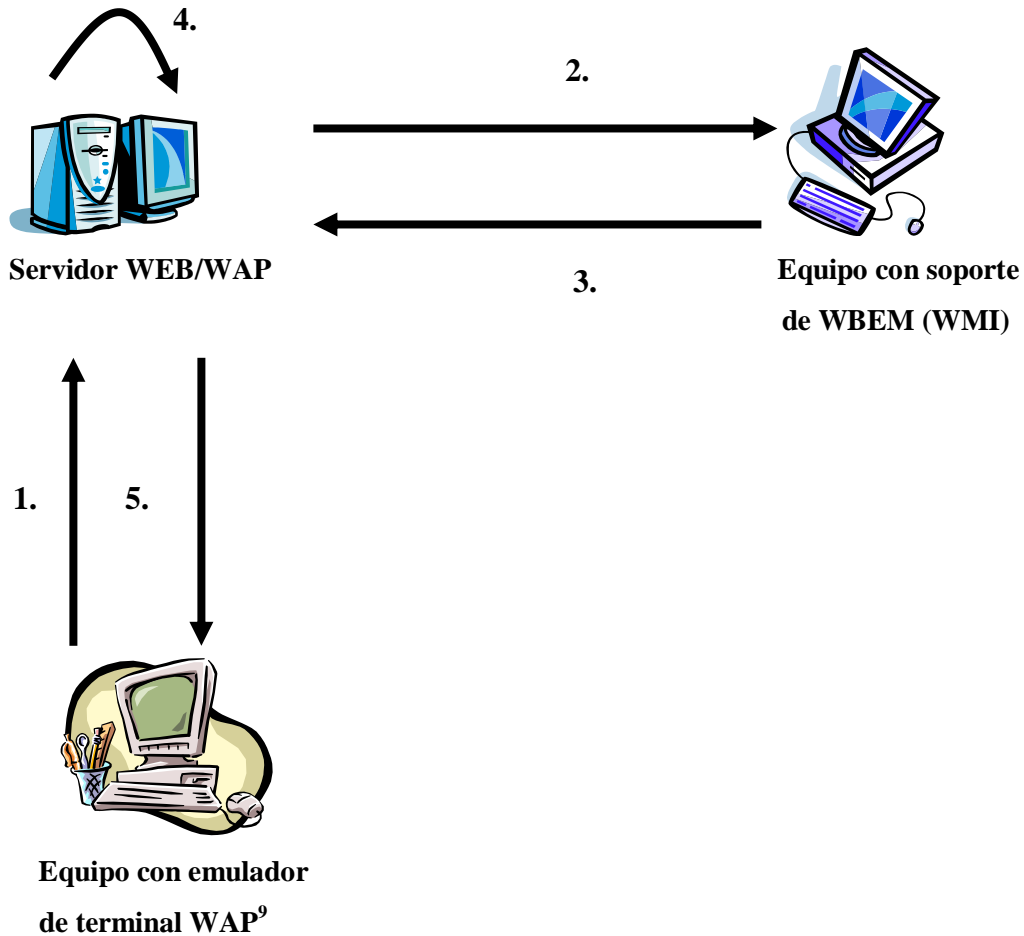


Figura 2.1 Entorno de aplicación

1. El equipo con emulador de terminal WAP, envía peticiones de supervisión al Servidor.

⁹ Aunque posteriormente se describe un actor como Terminal WAP (TW), este terminal no será real sino un emulador puesto que la disponibilidad de este terminal es muy limitada.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

2. El Servidor recibe estas peticiones y las envía al equipo que será supervisado (EG).
3. El equipo supervisado a través de WBEM(WMI), selecciona dicha información y la envía al servidor
4. El Servidor recibe esta información y la convierte a formato de cartas y barajas según el estándar WML.
5. El Servidor envía las páginas WML al equipo con emulador de terminal WAP (TW).

Para iniciar el servicio de supervisión del PC se requerirá una identificación de usuario, la cual esta almacenada en el repositorio WMI y es comparada con la contraseña enviada por el usuario, a través de su terminal WAP. Una vez validado el usuario, se da inicio al servicio de supervisión.

2.2.2. Modelo del dominio del sistema

La figura 2.2 muestra los actores que hacen parte de la aplicación y se nombran sus respectivas propiedades y métodos que describen el entorno de la aplicación.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

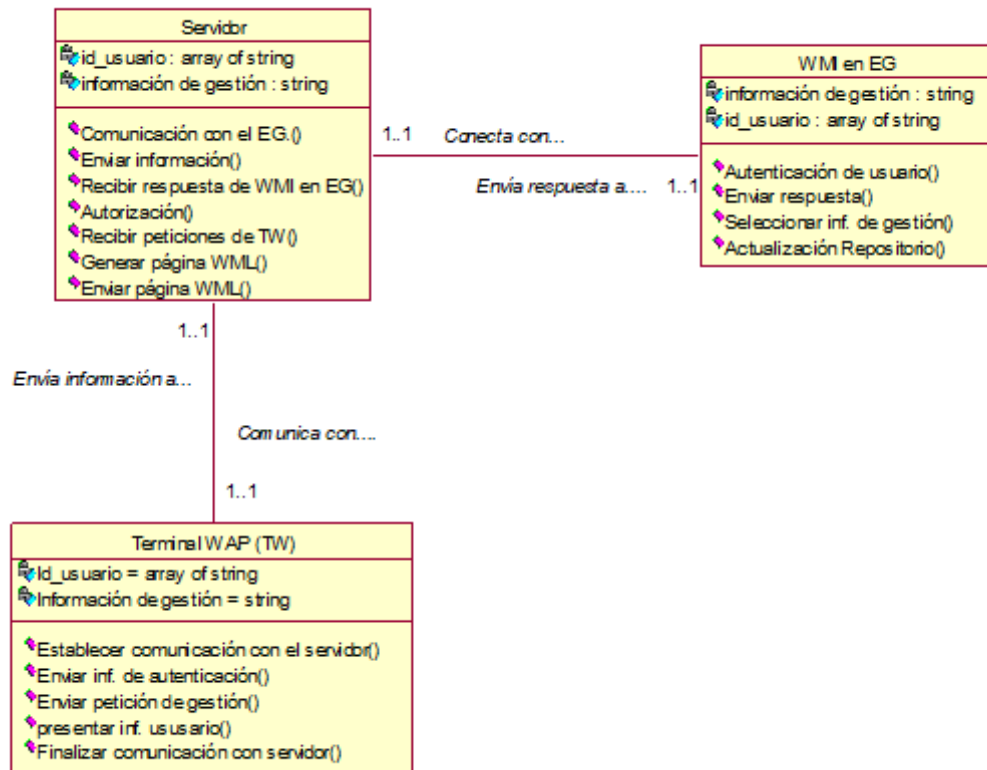


Figura 2.2 Dominio del sistema¹⁰

2.2.3 Definición del modelo de desarrollo específico

2.2.3.1 Modelos que describirán el sistema

A partir del RUP se generó el siguiente modelo de actividades para ser seguido durante el desarrollo del proyecto:

¹⁰ El término Repositorio se refiere a la base de datos de WMI en donde se encuentra almacenada toda la información de gestión.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

- *Captura de Requerimientos:* Realizar la identificación de los principales elementos que componen el problema a través del Modelo del Dominio, definir los principales términos utilizados en esta definición mediante un glosario, determinar las funciones que debe realizar el sistema en un árbol de funciones e identificar, priorizar y describir los casos de uso de alto nivel.
- *Análisis del software:* Realización del análisis de la arquitectura con el fin de encontrar los elementos del modelo (paquetes, clases, casos de uso ... etc.) que desempeñan actividades importantes dentro del sistema, además se estudian por separado cada uno de estos elementos realizando diagramas en los casos en que esto sea necesario, y se tienen los diagramas de colaboración de sistema y de estados.
- *Diseño del software:* Diseño de los subsistemas con sus interfaces, las clases de diseño , descripción de los casos de uso reales y de la arquitectura del sistema, además del modelo de implantación.

2.2.3.2 Fundamentos metodológicos a utilizar

Como modelo para el desarrollo del sistema se seguirá el RUP para el desarrollo de programas, instanciado para este caso particular según se ha descrito en el punto anterior.

2.2.3.3 Modelo del proceso de desarrollo

Este proceso se realizará basándose en el modelo en espiral planteado por el RUP y las fases que se seguirán son las siguientes:

- C0: Análisis de requerimientos.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

- C1: Análisis del software
- C2: Diseño del software

2.2.4. Arbol de funciones

De acuerdo a las entidades identificadas en el Modelo del Dominio, el árbol de funciones del sistema para el proceso de Autenticación es el siguiente:

ITEM	Función
1. Servidor	1.1 Conexión con el equipo gestionado 1.2 Autenticación de usuario <ul style="list-style-type: none"> 1.2.1 Pedir identificación 1.2.2 Recibir identificación 1.2.3 Enviar identificación a WMI en EG 1.2.4 Recibir respuesta de WMI en EG 1.3 Autorización 1.4 Recibir información del EG. 1.5 Recibir petición del TW 1.6 Generar página WML 1.7 Enviar página a TW
2. WMI en EG	2.1 Verificar id. Usuario 2.2 Enviar respuesta

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

	<p>2.3 Seleccionar información</p> <p>2.4 Actualizar repositorio</p>
3. Terminal WAP (TW)	<p>3.1 Establecer comunicación con el servidor</p> <p>3.2 Enviar información de autenticación de usuario</p> <p>3.3 Enviar petición de gestión</p> <p>3.4 Finalizar comunicación con el servidor</p> <p>3.5 Recibir información</p> <p>3.6 Desplegar información</p>

Tabla 3. Arbol de Funciones.

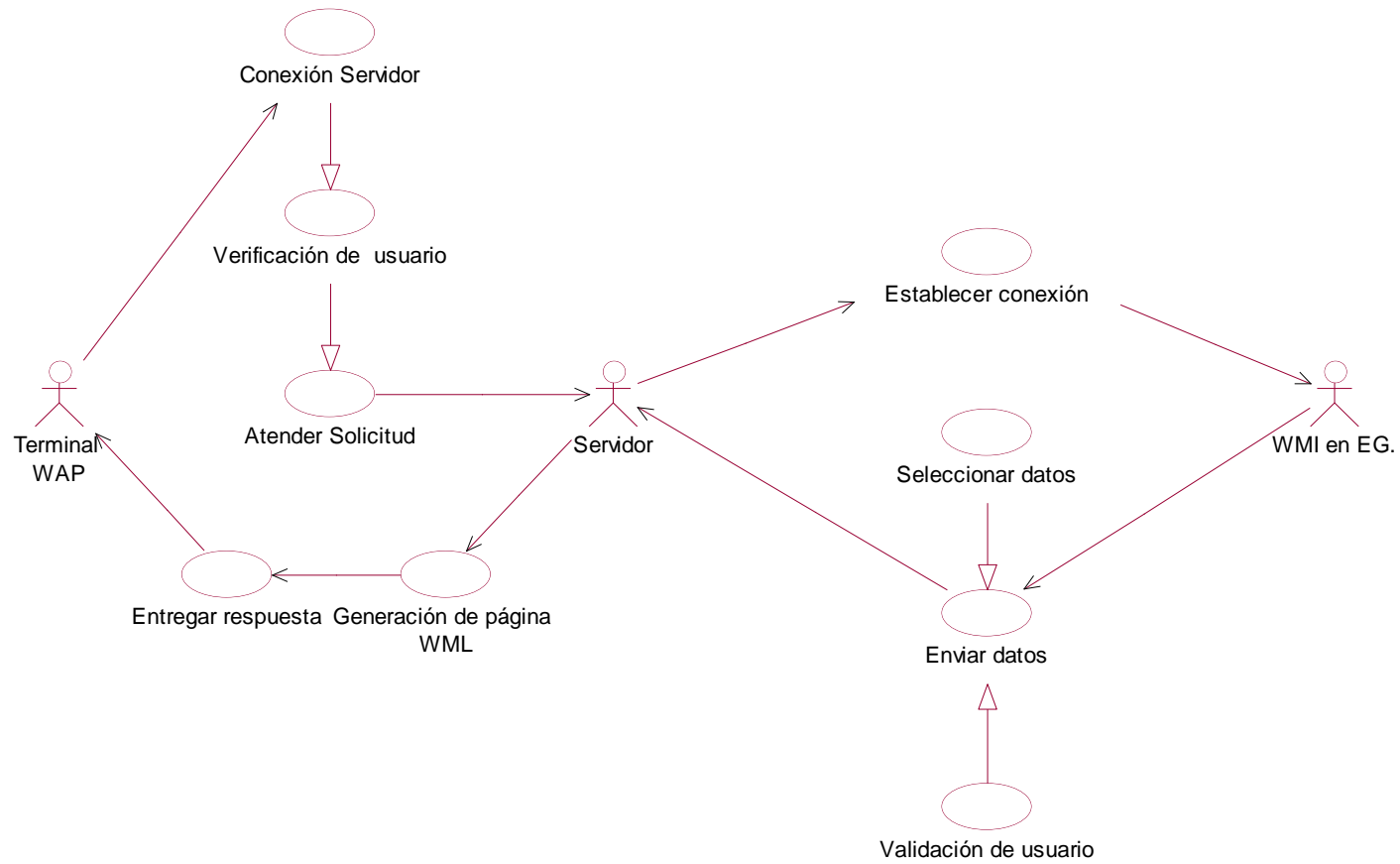


Figura 2.3 Casos de uso

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

2.3 Análisis de la aplicación

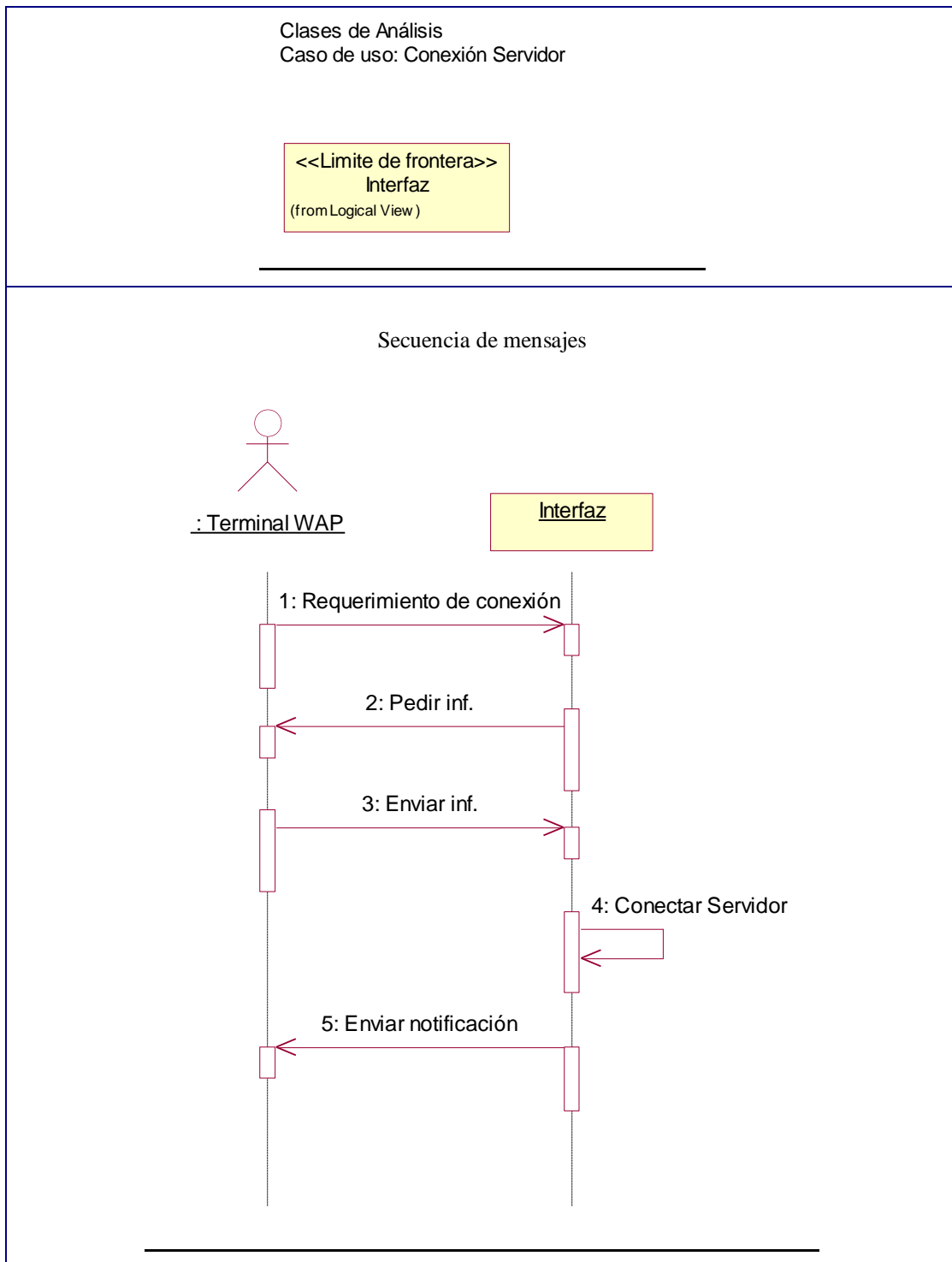
2.3.1 Descripción de Casos de Uso Esenciales

A continuación se hace la descripción de cada caso de uso de alto nivel, presente en el diagrama anterior que muestra los que se implementan en la aplicación.

2.3.1.1 Conexión Servidor

Caso de uso:	Conexión Servidor
Actores:	terminal WAP (Iniciador) y Servidor.
Propósito:	Establecer conexión con el Servidor.
Resumen:	El Terminal WAP pide comunicación con el servidor vía IP enviando el nombre o la dirección IP real del Servidor y la carpeta donde se encuentra la aplicación. Con esta información se establece una conexión con el Servidor indicado por el usuario.
Tipo:	Primario y esencial.
Referencias cruzadas	Funciones: R3.1
Pre-condiciones:	<ul style="list-style-type: none"> • El Servidor debe estar habilitado para establecer la conexión.
Flujo principal:	<ul style="list-style-type: none"> • Este caso de uso empieza cuando el TW inicia el procedimiento de conexión con el Servidor enviando la dirección IP o el nombre del Servidor con el cual se desea conectar. • Una vez entregada la información, se establece la conexión con el Servidor.
Flujos de Excepción	<ul style="list-style-type: none"> • Si el Servidor no esta habilitado para realizar la conexión se manda una notificación para reintentar la operación.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

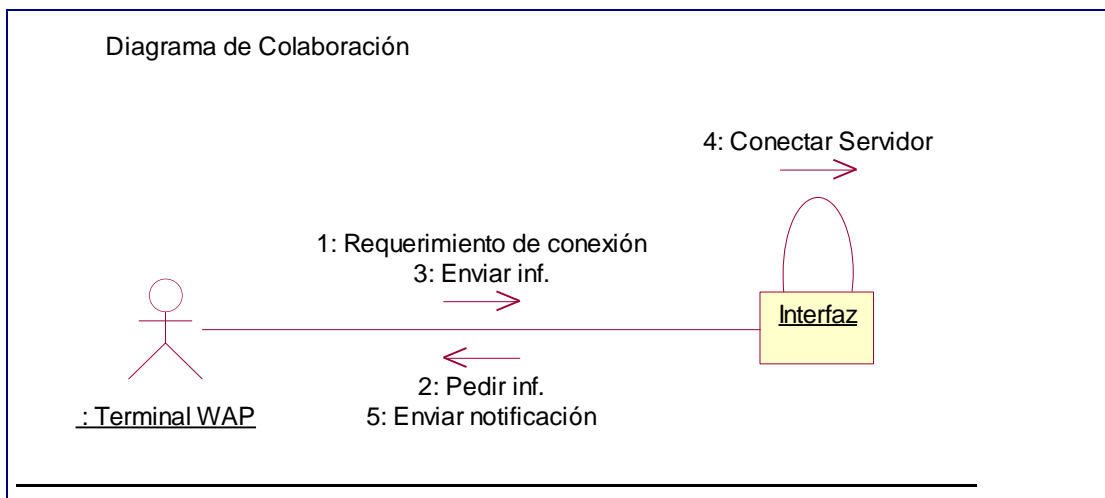
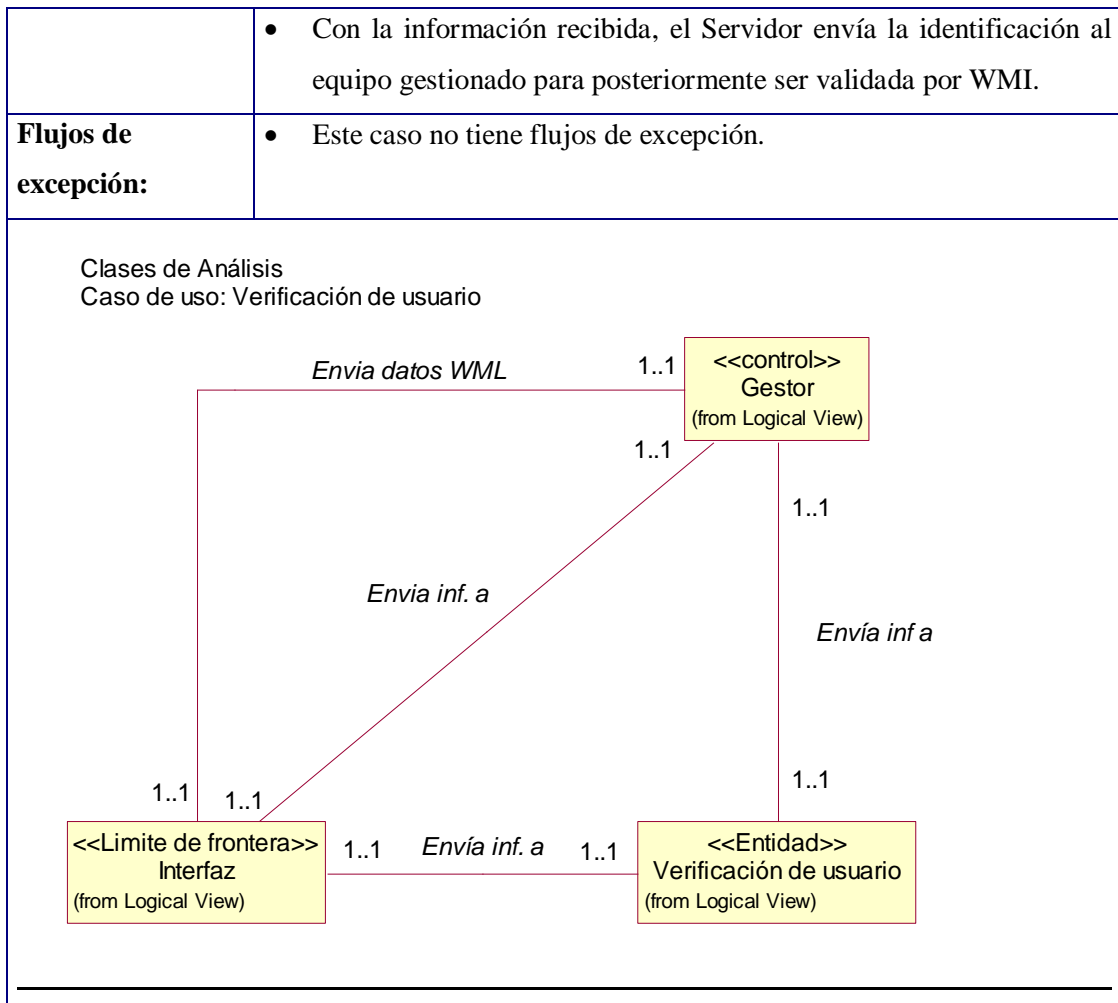


Tabla 4. Caso de uso: Conexión Servidor

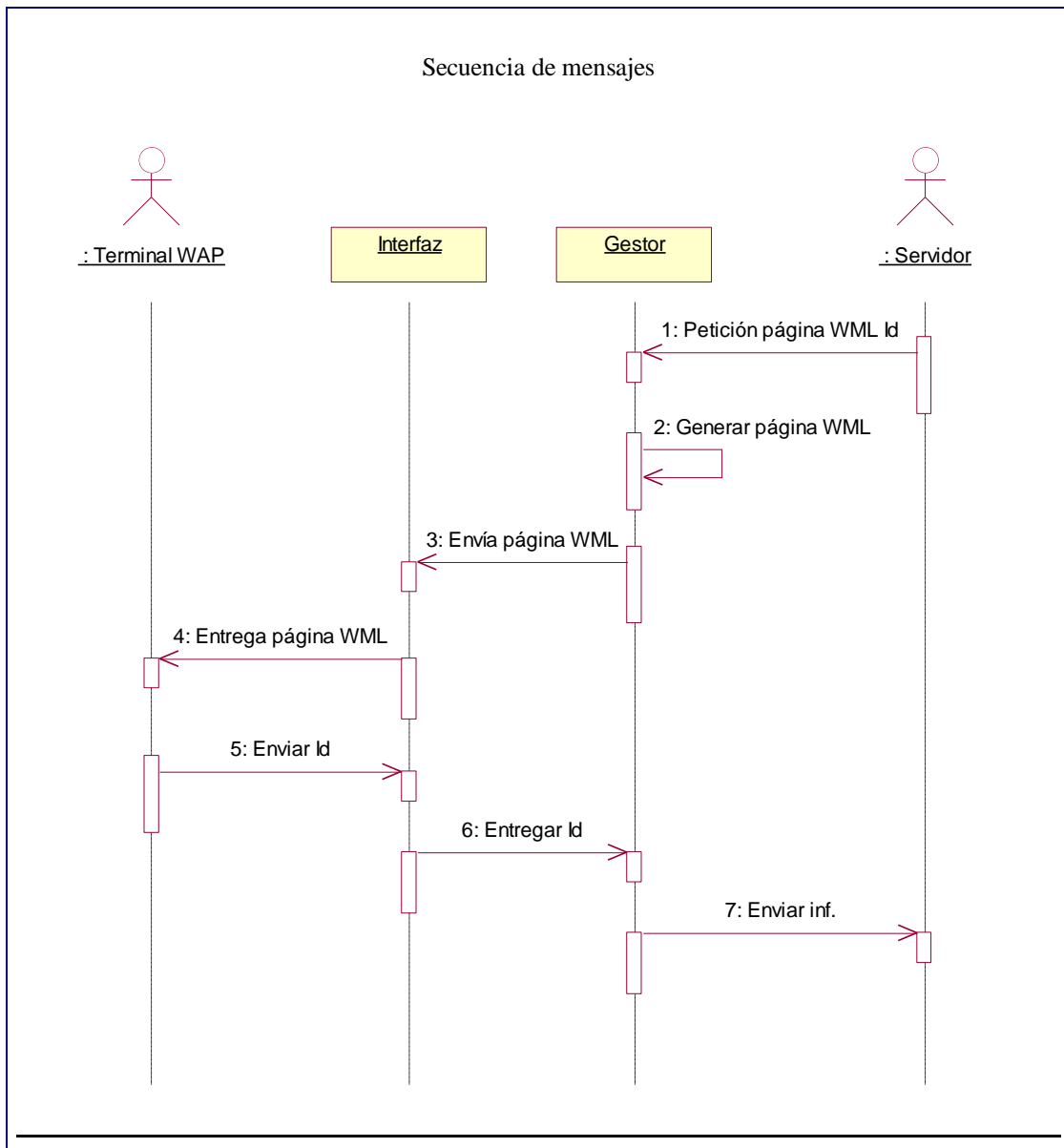
2.3.1.2 Verificación de usuario

Caso de uso: Verificación de usuario	
Actores:	Terminal WAP y Servidor. Este caso de uso es inicializado por el caso de uso Conexión Servidor.
Propósito:	Permitir la verificación del usuario para posteriormente enviar su autorización.
Resumen:	El usuario pide su autenticación al servidor el cual le solicita el nombre del equipo que desea supervisar, el nombre de usuario y la contraseña correspondientes a su sesión en el EG. Posteriormente el servidor envía la información al EG.
Tipo:	Primario.
Referencias cruzadas	Funciones: R1.2.1, R1.2.2, R1.5, R1.6, R3.1, R3.2, R3.5 y R3.6
Pre-condiciones:	<ul style="list-style-type: none"> Se debe haber establecido la conexión con el Servidor.
Flujo principal:	<ul style="list-style-type: none"> Este caso de uso empieza cuando el caso de uso Conexión Servidor se haya cumplido. El servidor solicita la identificación de usuario para verificarla.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

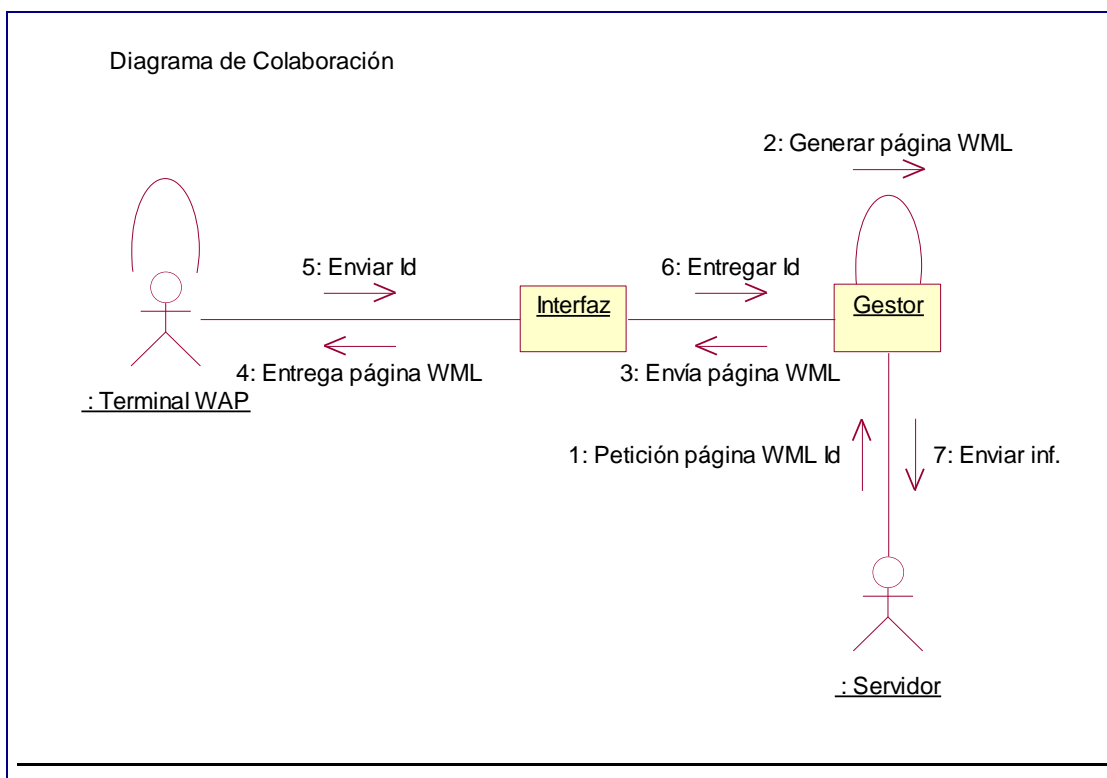


Tabla 5. Caso de uso: Verificación de Usuario.

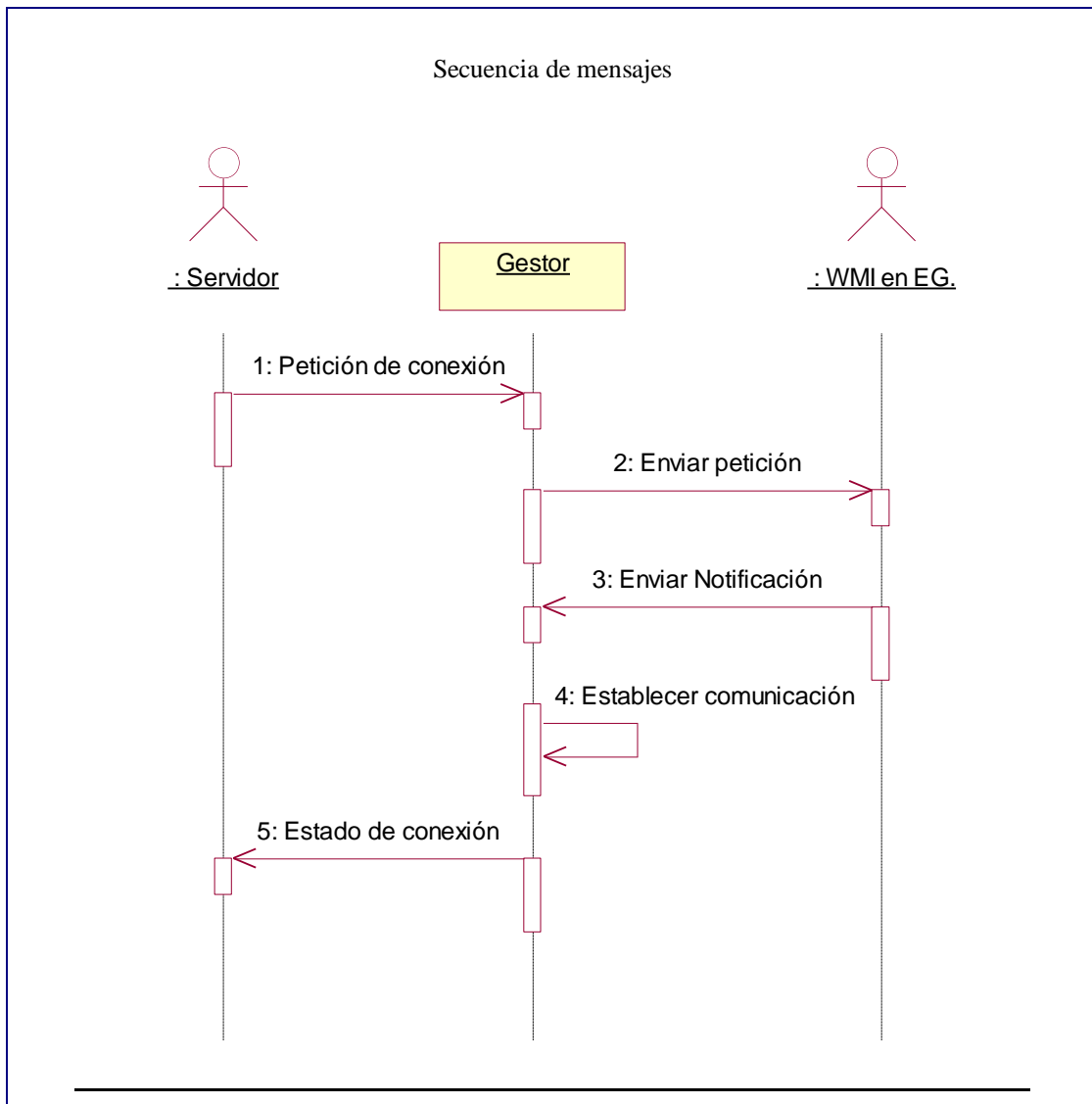
2.3.1.3 Establecer conexión

Caso de uso:	Establecer Conexión
Actores:	Servidor (Inicializador) y Equipo Gestionado.
Propósito:	Permitir al Servidor conectarse con el equipo gestionado.
Resumen:	El Servidor establece una conexión con el equipo gestionado.
Tipo:	Primario y esencial.
Referencias cruzadas	Funciones: R1.1
Pre-condiciones:	<ul style="list-style-type: none"> Este caso de uso no tiene pre-condiciones.
Flujo principal:	<ul style="list-style-type: none"> Este caso de uso empieza cuando en el Servidor se ejecuta la

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

	función con la cual se conecta al equipo gestionado a través de WMI.
Flujos de excepción:	Este caso de uso no tiene flujos de excepción.
<p>Clases de Análisis Caso de uso: Establecer conexión</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"><p><<control>> Gestor (from Logical View)</p></div> <hr style="width: 30%; margin: 10px auto;"/>	

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

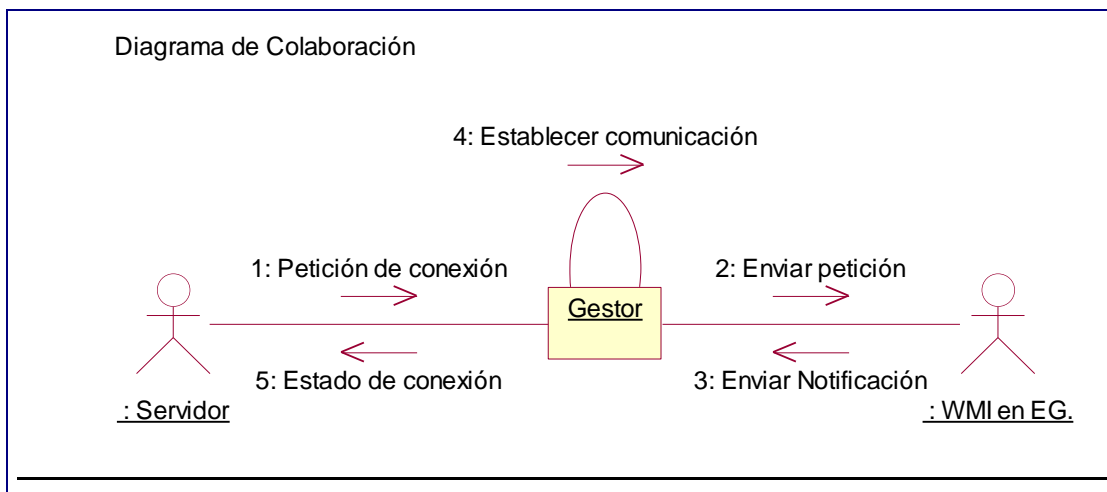
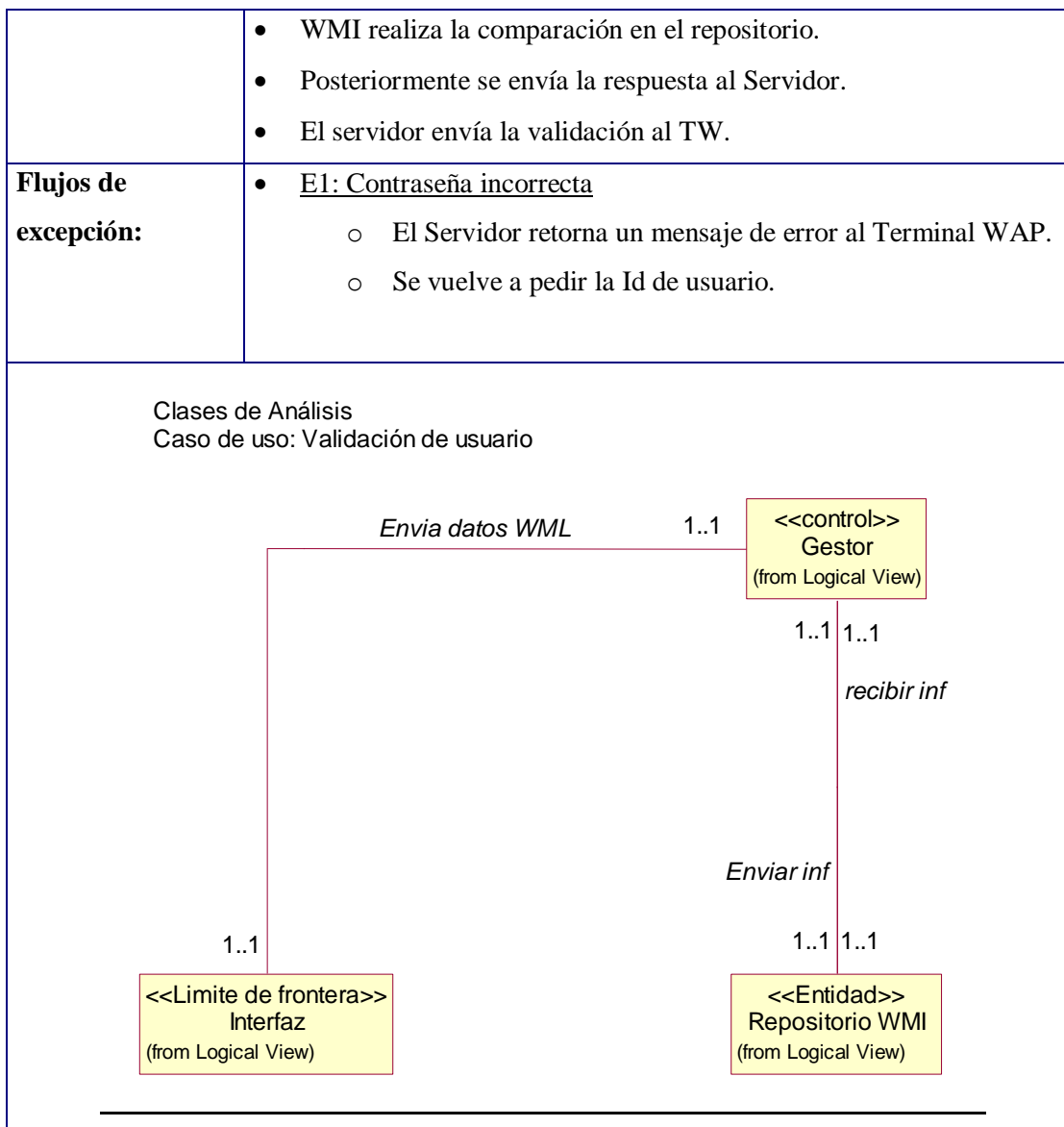


Tabla 6. Caso de uso: Establecer Conexión.

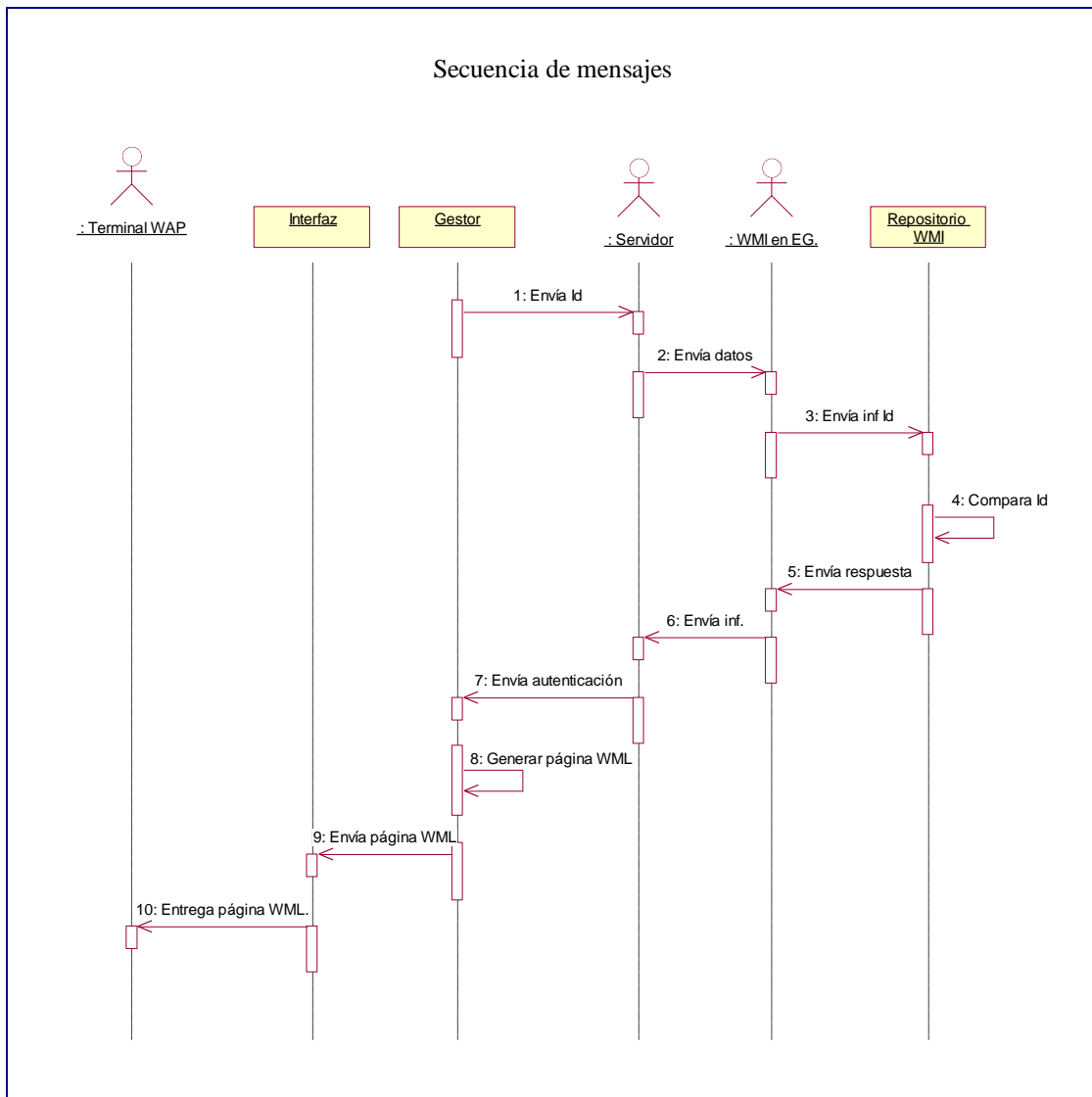
2.3.1.4 Validación de usuario.

Caso de uso:	Validación de usuario.
Actores:	Servidor (Iniciador), WMI en EG y TW
Propósito:	Permitir la validación del usuario para posteriormente enviar su autorización.
Resumen:	El servidor envía la id de usuario al EG el cual a través de WMI realiza la comparación con la almacenada en el repositorio y envía la notificación al servidor y se envía esta validación al TW.
Tipo:	Primario.
Referencias cruzadas	Funciones: R1.1, R1.2.3, R1.2.4, R1.3, R1.4, R1.6, R1.7, R2.1, R2.2 y R3.5.
Pre-condiciones:	<ul style="list-style-type: none"> • El sistema debe contar con la siguiente información: <ul style="list-style-type: none"> ○ Contraseña almacenada en el repositorio.
Flujo principal:	<ul style="list-style-type: none"> • Este caso de uso empieza cuando el Servidor envía la Id de usuario a WMI del EG

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

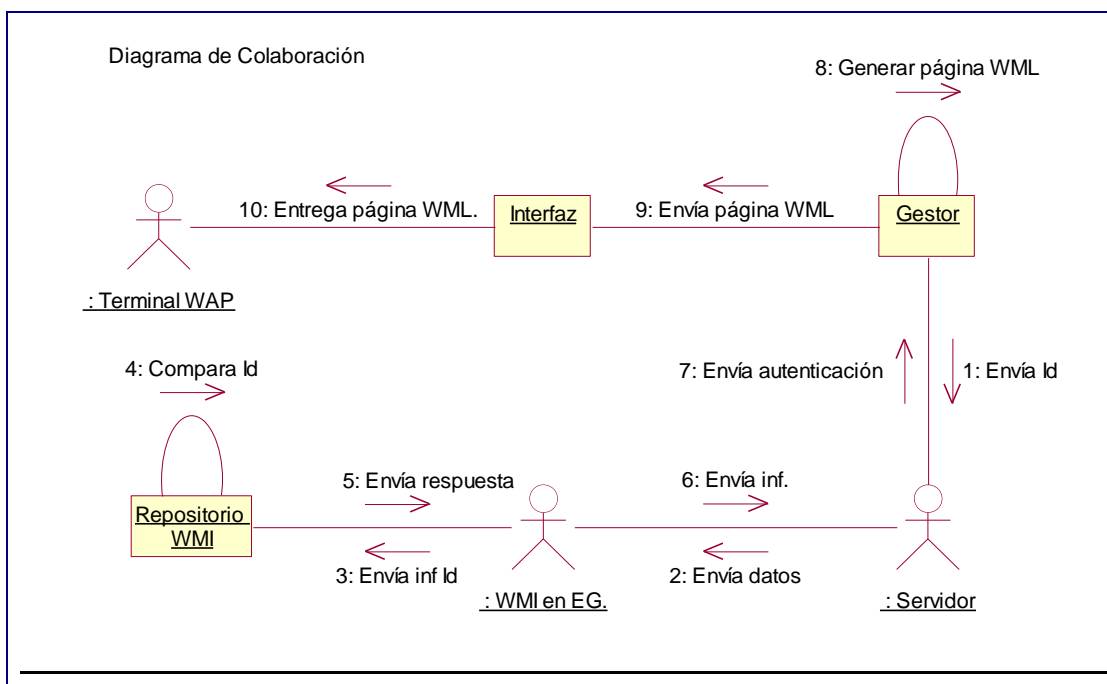
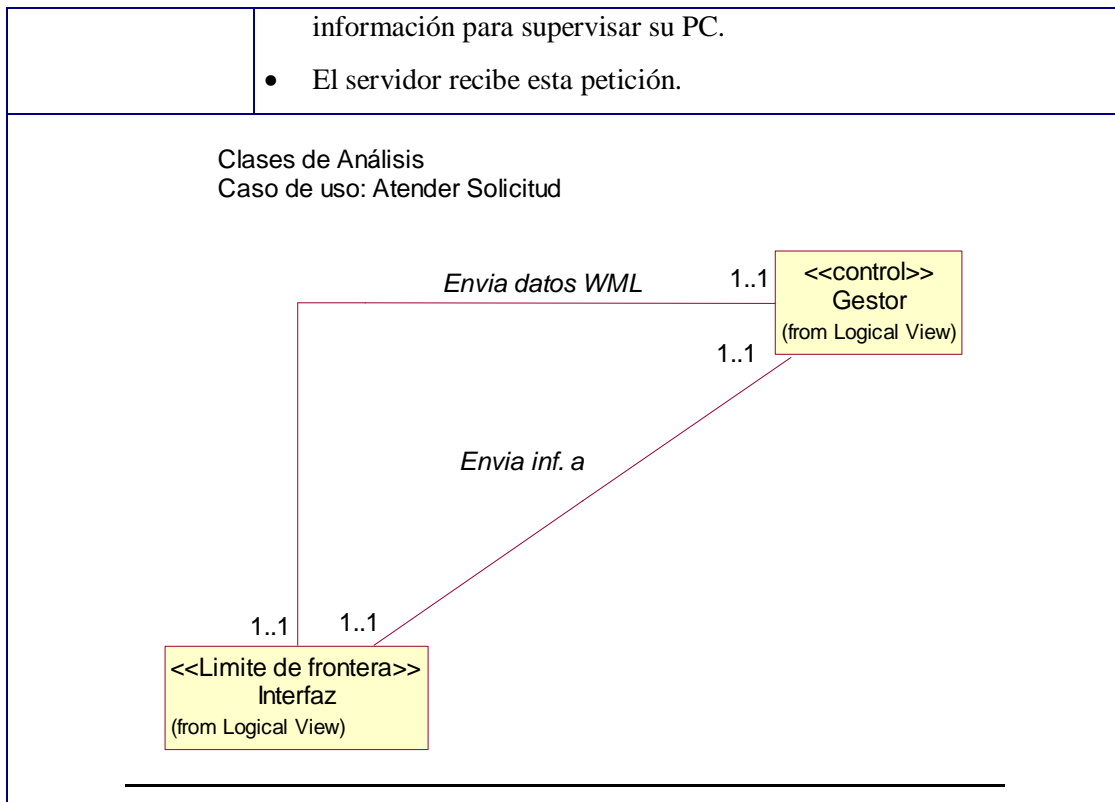


Tabla 7. Caso de uso: Validación de Usuario.

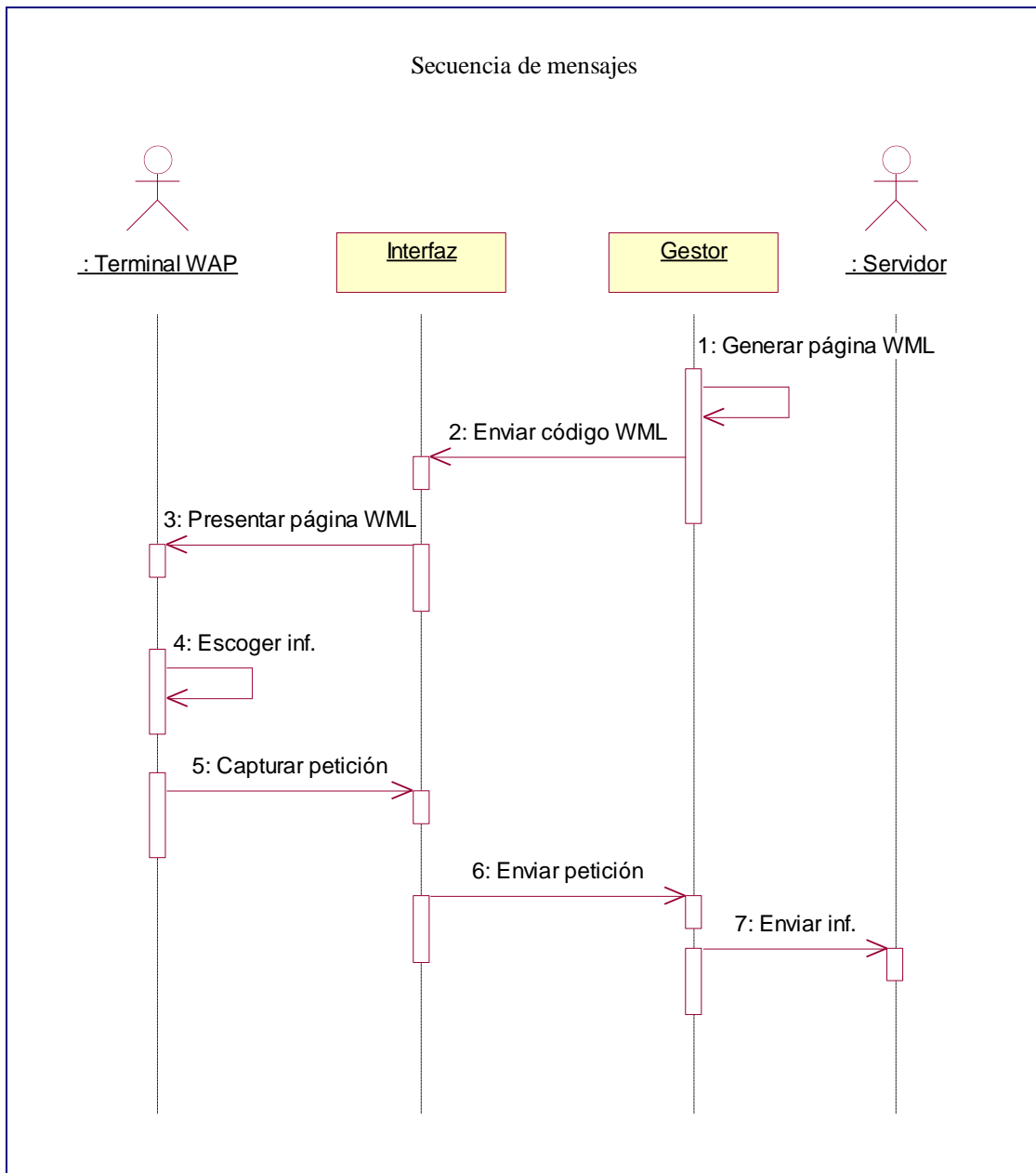
2.3.1.5 Atender Solicitud

Caso de uso:	Atender Solicitud
Actores:	Terminal WAP (Iniciador) y Servidor.
Propósito:	Iniciar el servicio de supervisión del PC
Resumen:	El usuario después de haber sido autorizado envía solicitudes para gestionar su PC. Luego el servidor recibe esta petición para posteriormente seleccionar la información del repositorio.
Tipo:	Primario.
Referencias cruzadas	Funciones: R1.5, R1.6, R1.7 y R3.3
Pre-condiciones:	<ul style="list-style-type: none"> El usuario debe haber sido autorizado para iniciar el servicio de gestión.
Flujo principal:	<ul style="list-style-type: none"> Este caso de uso empieza cuando el Terminal WAP solicita

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

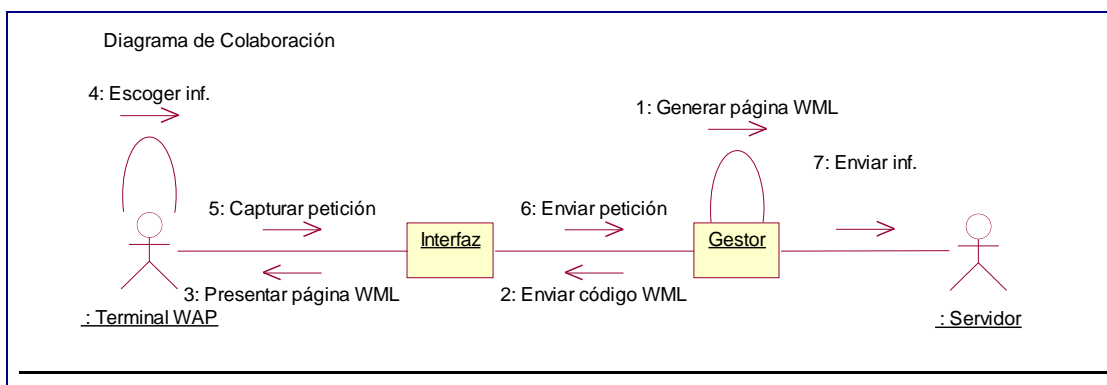
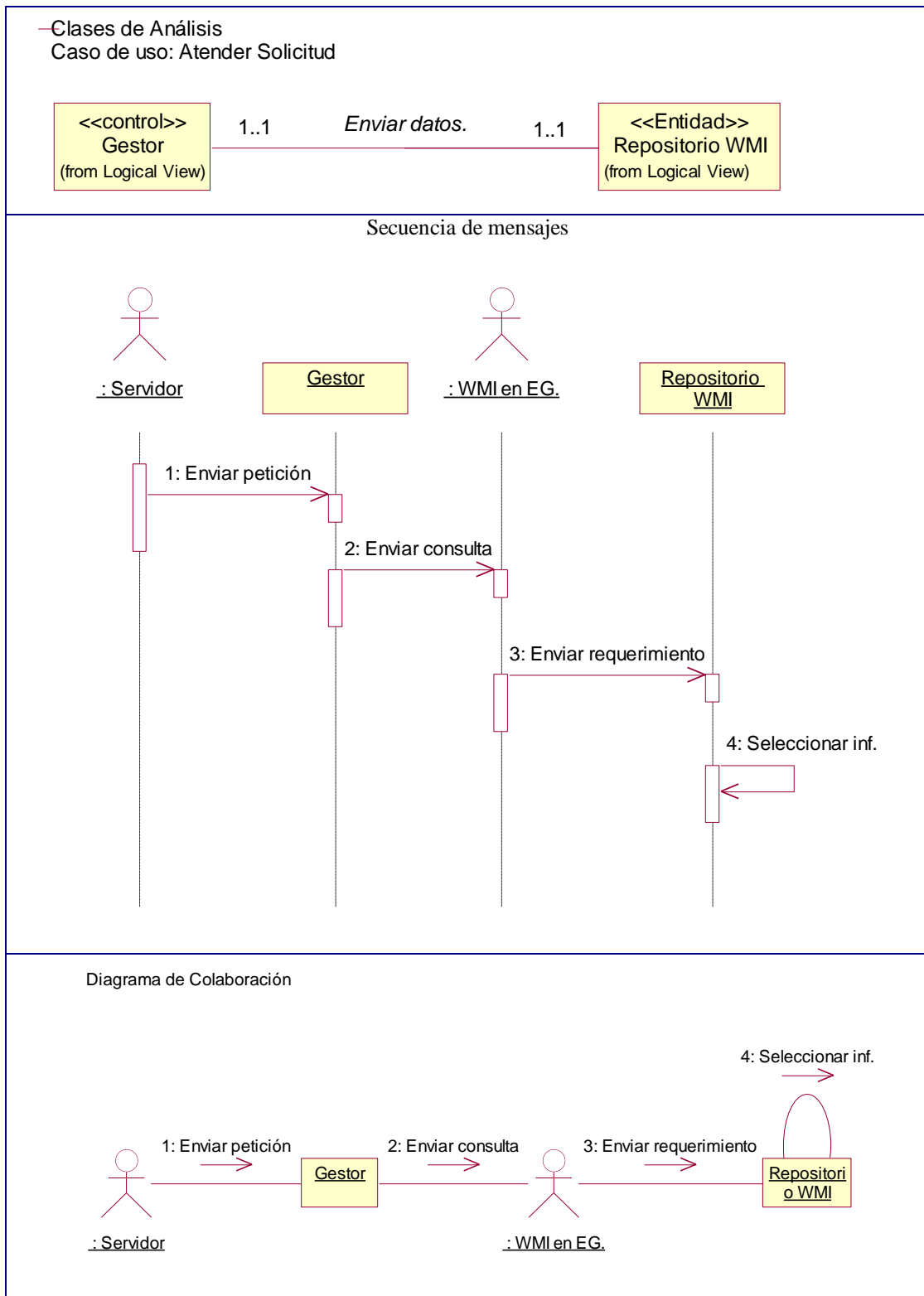


Tabla 8. Caso de uso: Atender Solicitud.

2.3.1.6 Seleccionar Datos

Caso de uso:	Seleccionar datos.
Actores:	Servidor (Inicializador) y WMI en EG.
Propósito:	Obtener la información requerida por el usuario sobre la supervisión de su PC.
Resumen:	Este caso de uso es activado por el Servidor, el cual envía una consulta para que WMI en el EG la selecciones del repositorio.
Tipo:	Primario.
Referencias cruzadas	Funciones: R2.3
Pre-condiciones:	<ul style="list-style-type: none"> • Se de haber establecido una conexión con WMI en el EG. • Se debe tener la petición del usuario en el Servidor.
Flujo principal:	<ul style="list-style-type: none"> • El caso de uso es inicializado por el Servidor el cual envía una consulta a WMI en el EG. • En base a esta petición WMI selecciona la información de supervisión solicitada por el usuario.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Tabla 9. Caso de uso: Seleccionar Datos.

2.3.1.7 Enviar Datos

Caso de uso: Enviar datos.	
Actores:	El Caso de uso Seleccionar Datos es el inicializador. Servidor y WMI en EG.
Propósito:	Enviar la información solicitada por el usuario al Servidor para luego presentársela al usuario.
Resumen:	Este caso de uso es activado por el caso de uso Seleccionar Datos, con el cual una vez obtenida la información de supervisión es enviada al Servidor para posteriormente presentársela al usuario en el TW.
Tipo:	Primario.
Referencias cruzadas	Funciones: R2.2
Pre-condiciones:	<ul style="list-style-type: none"> Se debe haber realizado la consulta a WMI en EG.
Flujo principal:	<ul style="list-style-type: none"> El caso de uso es inicializado por el caso de uso Seleccionar Datos, el cual seleccionó la información requerida por el usuario. Una vez seleccionada la información, el EG la envía al Servidor.
<p>–Clases de Análisis Caso de uso: Enviar Datos</p> <pre> classDiagram class Gestor["<<control>>\nGestor\n(from Logical View)"] class RepositorioWMI["<<Entidad>>\nRepositorio WMI\n(from Logical View)"] Gestor "1..1" -- "1..1" RepositorioWMI : Recibir datos </pre>	

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

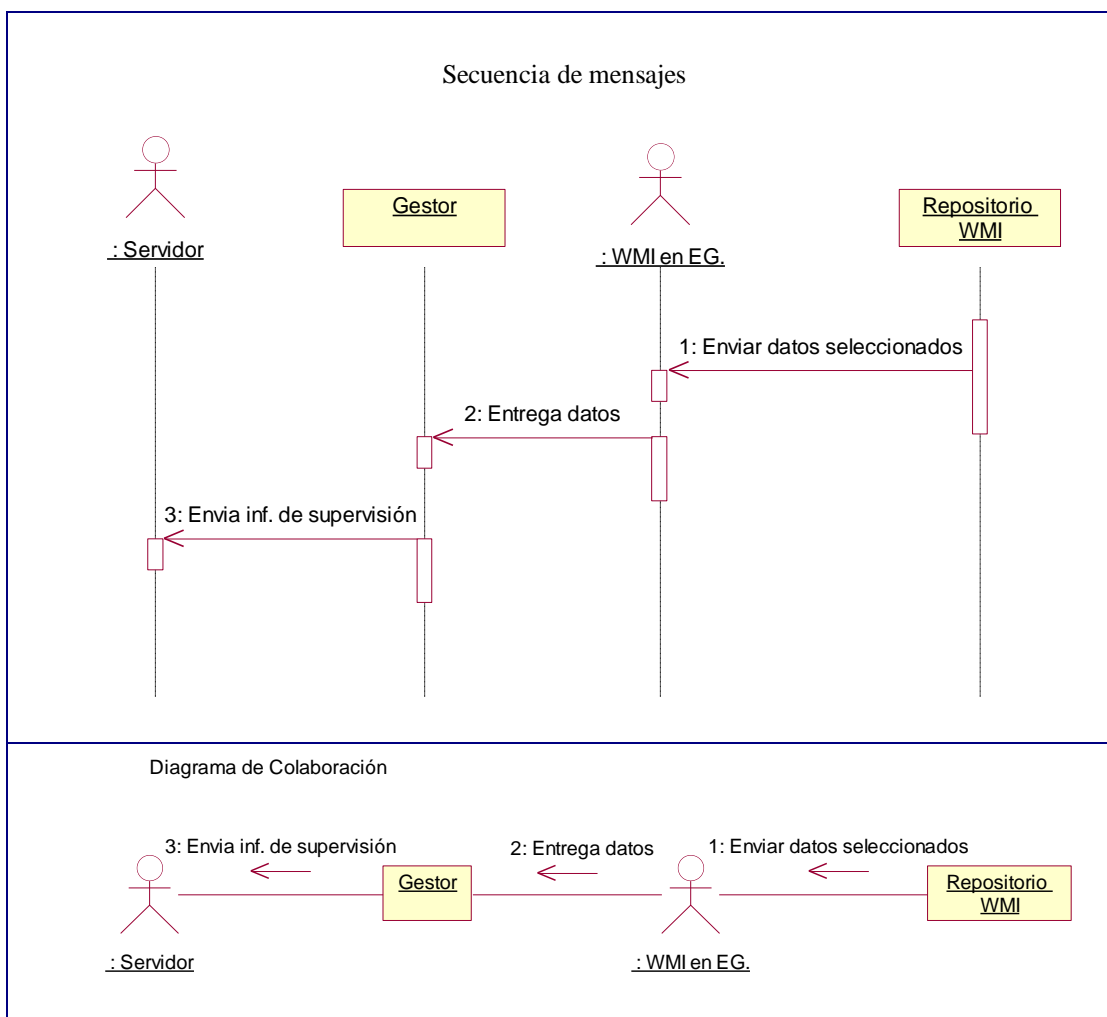


Tabla 10. Caso de uso: Enviar Datos.

2.3.1.8 Generar páginas WML

Caso de uso:	Generar páginas WML
Actores:	Servidor (Iniciador).
Propósito:	Convertir la información a un formato entendible por el terminal WAP.
Resumen:	El servidor después de tener la información solicitada por el usuario la convierte a formato WML para posteriormente entregarla al terminal.

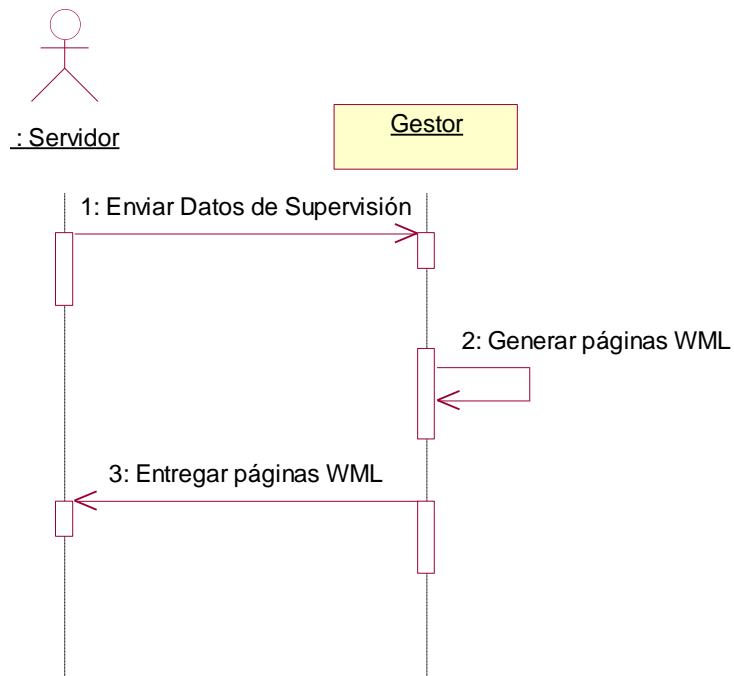
Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Tipo:	Primario.
Referencias cruzadas	Funciones: R1.6
Pre-condiciones:	<ul style="list-style-type: none"> El sistema debe contar con la siguiente información: <ul style="list-style-type: none"> Los datos solicitados por el usuario.
Flujo principal:	<ul style="list-style-type: none"> Este caso de uso empieza cuando el servidor obtiene la información del EG solicitado por el usuario para convertirla en un deck WML que sea entendible por el Terminal WAP.

Clases de Análisis
Caso de uso: Generar Deck

Gestor

Secuencia de mensajes



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

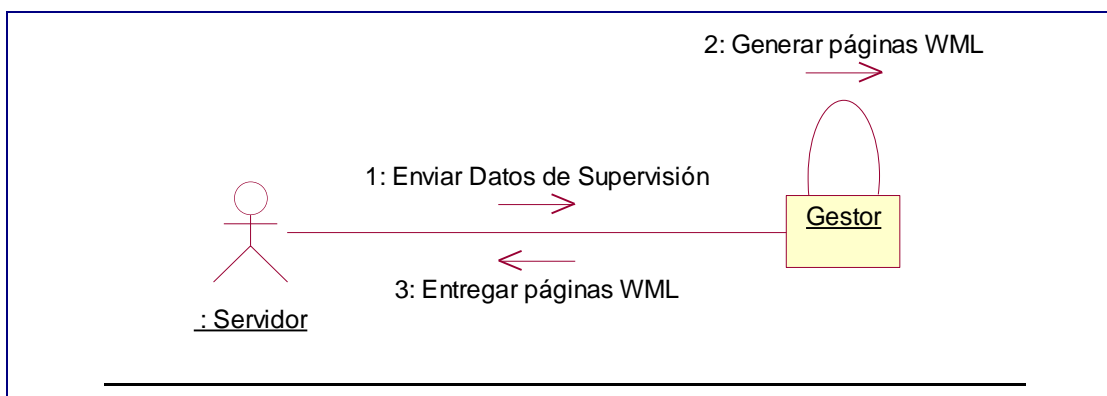
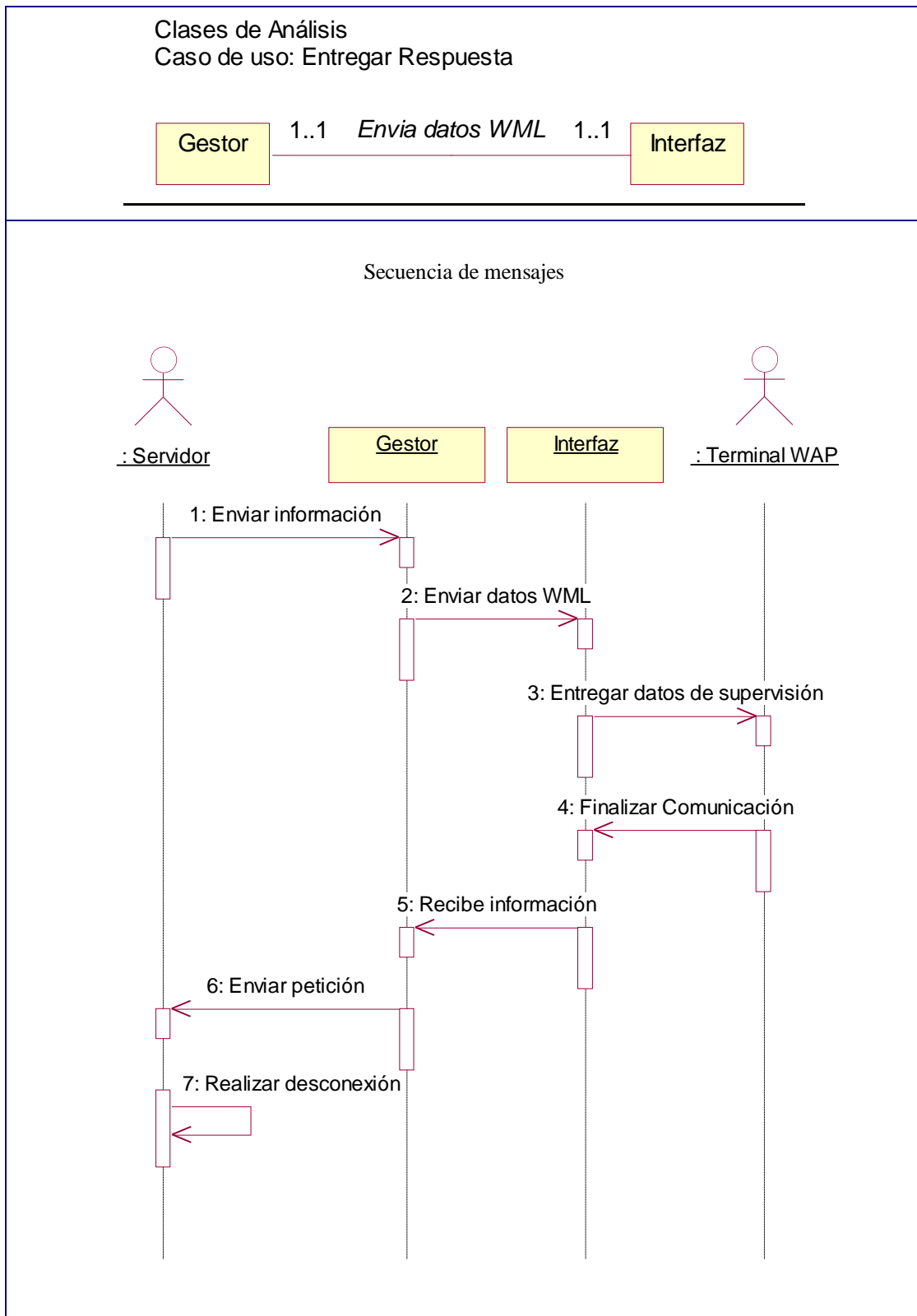


Tabla 11. Caso de uso: Generar páginas WML.

2.3.1.9 Entregar Respuesta

Caso de uso:	Entregar respuesta
Actores:	Servidor (Iniciador) y terminal WAP.
Propósito:	Entregar la respuesta a la solicitud requerida por el usuario.
Resumen:	El servidor después de tener la información en formato WML entrega la información al terminal WAP para que se visualizada por el usuario. También se encarga de finalizar la comunicación con el servidor.
Tipo:	Primario.
Referencias cruzadas	Funciones: R1.7, R3.4 y R3.5
Pre-condiciones:	<ul style="list-style-type: none"> Se debe tener la información en formato WML para poderla presentar al usuario.
Flujo principal:	<ul style="list-style-type: none"> Este caso de uso empieza cuando el servidor ha generado el deck de la información solicitada por el usuario y se dispone a entregarla Una vez entregada la información, si el usuario lo desea, se finaliza la comunicación con el servidor.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP



Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

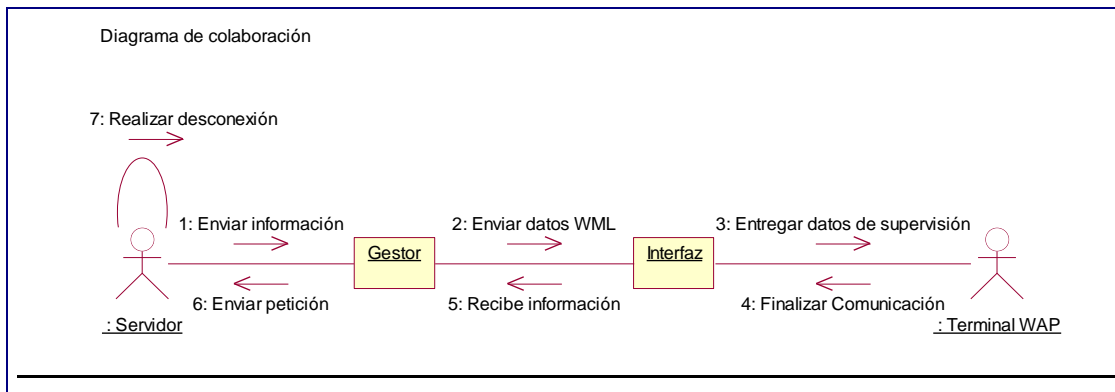


Tabla 12. Caso de uso: Entregar Respuesta.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

2.4 Diseño de la aplicación

2.4.1 Clases de diseño

A continuación se presentan los diagramas que implementan la aplicación.

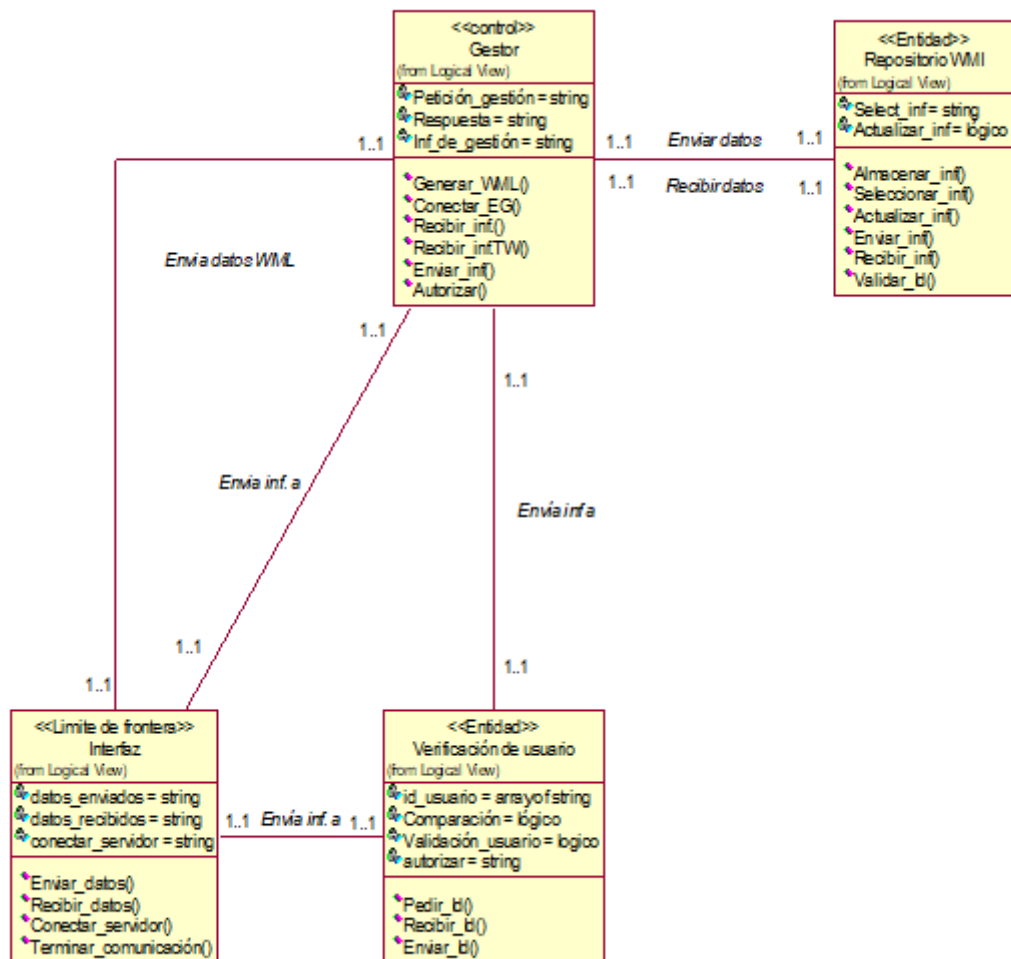


Figura 2.4 Clases de diseño

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

2.4.2 Diagrama subsistemas e interfaces

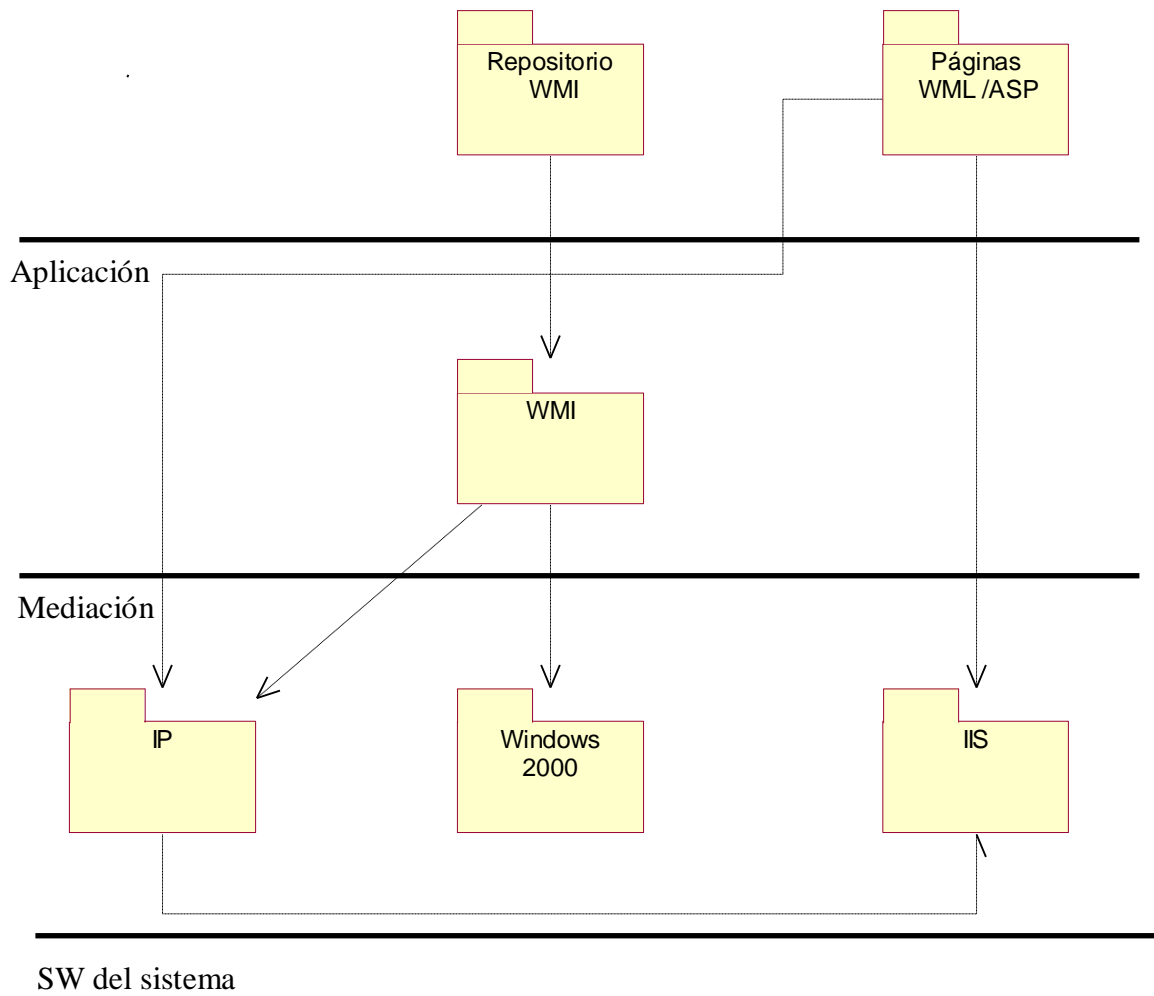


Figura 2.5 Diagrama de Subsistemas e Interfaces¹¹

¹¹ IIS (Internet Information Server)

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

2.4.3 Diagrama de implantación

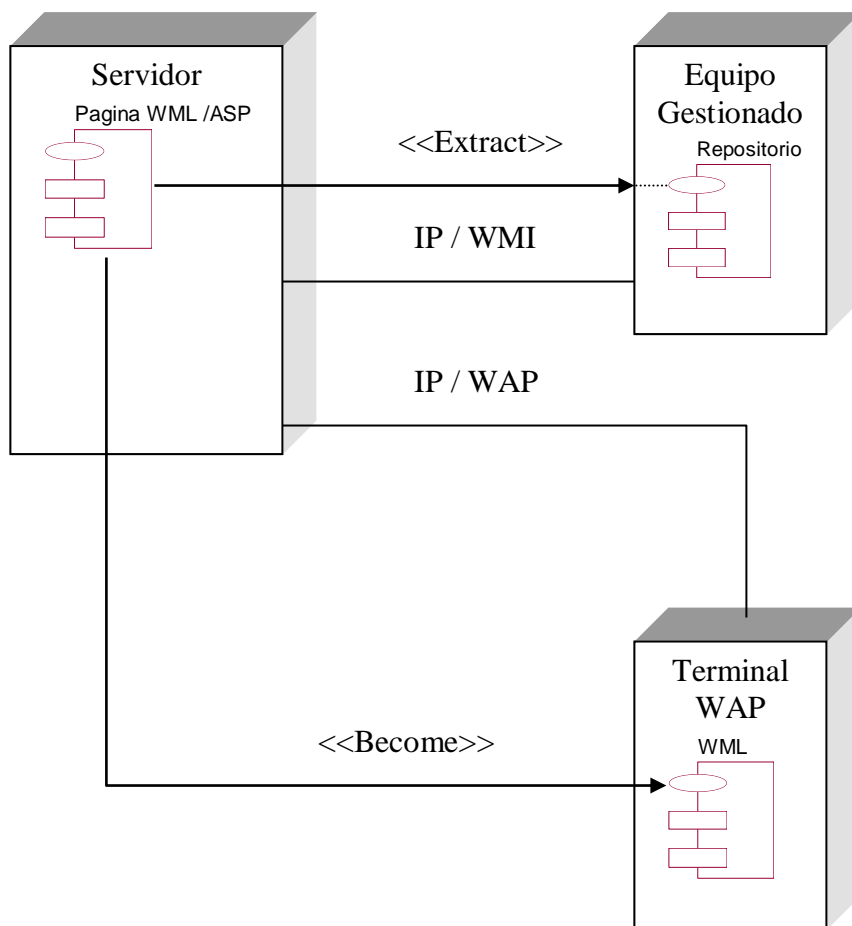


Figura 2.6 Diagrama de Implantación.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

3. CONCEPTOS FUNDAMENTALES PARA EL USO DE WMI

3.1 Introducción a WMI

El estándar WBEM proporciona acceso uniforme a información de gestión. WMI es la aplicación Microsoft de esta tecnología. Tal información de gestión incluye información sobre el estado de memoria del sistema, inventarios de aplicaciones del cliente actualmente instaladas, e información sobre estado del cliente. La tecnología WMI implementada por las plataformas Microsoft® Windows® permite representar los sistemas, aplicaciones, redes, y otros componentes gestionados utilizando el CIM diseñado por el DMTF. CIM puede modelar cualquier elemento en el ambiente gestionado sin tener en cuenta la localización de la fuente de datos.

Además para el modelamiento de datos, WMI ofrece un poderoso conjunto de servicios base que incluyen recuperación de información basada en consultas y notificación de evento. El acceso a estos servicios y a los datos de gestión se hacen con un solo Modelo de Objeto de Componente (COM- Component Object Model) para programar la interfaz.

El Conjunto de Desarrollo Software WMI (WMI SDK- WMI Software Development Kit) incluye las herramientas para crear aplicaciones del cliente y proveedores de “WMI”.¹²

3.2 Apreciación global de la arquitectura

La Gestión Empresarial Basada en Web es una iniciativa emprendida por el DMTF para proveer un estándar para la gestión de ambientes empresariales y la solución económica a sus

¹² Para mayor información, ver anexo C

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

necesidades de gestión. La iniciativa de la Gestión Empresarial Basada en Web abarca una multitud de tareas y va desde la configuración de una simple estación de trabajo hasta la gestión máxima de la empresa a través de múltiples plataformas. La parte central a la iniciativa es el “CIM”, un modelo de datos extensible para representar objetos que existen en ambientes de gestión típicos y el Formato de Objeto Gestionado, lenguaje para definir y guardar datos modelados.

“WMI” es una aplicación de la iniciativa “WBEM” para las plataformas de Microsoft® Windows® que extiende “CIM” para representar objetos que existen en ambientes de “WMI” y llevando a cabo una infraestructura de gestión para apoyar el lenguaje de “MOF” y una interfaz de programación común, “WMI” permite a las diversas aplicaciones manejar una variedad de componentes de la empresa transparentemente.

La infraestructura de “WMI” consta de los siguientes componentes:

- El software de “WMI” (Winmgmt.exe) el cual es un componente que proporciona a las aplicaciones un acceso uniforme para gestionar datos.
- El repositorio “CIM” que es una área central de almacenamiento para gestionar datos.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

3.3 Arquitectura de “WMI”

La arquitectura de la tecnología “WMI” consta de:

- Aplicaciones de gestión
- Objetos gestionados
- Proveedores
- Infraestructura de gestión la cual contiene el software de “WMI” (Winmgmt.exe) y el repositorio “CIM”.

Las aplicaciones de gestión son aplicaciones basadas en Microsoft® Windows® o servicios Microsoft® Windows NT®/Windows 2000 que procesan o despliegan datos de los objetos gestionados. Una aplicación de gestión puede realizar una variedad de tareas como medir desempeño, informar detenciones y correlacionar datos.

Los objetos gestionados son componentes lógicos o físicos de la empresa. Se modelan los objetos manejados utilizando el “CIM” y ellos son accedidos por aplicaciones de gestión a través de “WMI”. Un objeto gestionado puede ser cualquier componente de la empresa, desde un pequeño pedazo hardware como un cable, hasta una gran aplicación de software como un sistema de base de datos.

Los proveedores utilizan la Interfaz de Programación para Aplicaciones COM (COM API-Component Object Model Application Programming Interface) para proporcionar “WMI” con datos de los objetos gestionados, manejar demandas en nombre de las aplicaciones de gestión y para generar notificaciones de eventos. El “WMI SDK” proporciona varios proveedores normales, como un proveedor del registro, para acceder información desde el registro del sistema. Además el “WMI SDK” también proporciona un Proveedor de eventos Windows NT/Windows 2000 que permite a las aplicaciones recibir notificaciones de los eventos Windows NT/Windows 2000 y acceder la información guardada en el registro de eventos Windows NT/Windows 2000. Los vendedores que desarrollan para este ambiente, llamados

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

vendedores de terceras partes, pueden crear proveedores usuales para interactuar con objetos gestionados específicos de su ambiente.

La infraestructura de gestión consiste de “WMI” y el “Repositorio CIM”. “WMI” permite a los usuarios manejar comunicaciones entre las aplicaciones de gestión y proveedores, de esta forma los usuarios guardan sus datos estáticos en el almacén de “CIM”. Las aplicaciones y proveedores se comunican a través de “WMI” que utiliza una interfaz de programación común (COM API). El COM API proporciona la notificación de eventos y requerimientos que procesan servicios y está disponible en lenguajes de programación como C y C++ .

El “Repositorio CIM” incluye datos de gestión estáticos. Los datos estáticos son aquellos que no cambian regularmente. “WMI” también soporta datos dinámicos los cuales deben generarse por demanda ya que varían frecuentemente. Pueden ponerse datos en el “Repositorio CIM” a través de “WMI” o de administradores de red.

Los diseñadores de terceras partes pueden poner información en el repositorio “CIM” con el MOF y su compilador o por medio del COM API.

La figura 3.1 muestra la relación entre los componentes en la tecnología de “WMI”. En lo alto del diagrama se muestran ejemplos de aplicaciones de gestión. Algunas de estas aplicaciones acceden al COM API directamente para actuar recíprocamente con “WMI” y el “Repositorio CIM” para hacer peticiones de gestión. Otras aplicaciones utilizan métodos de acceso como el de la Conectividad a Bases de Datos Abiertas (ODBC-Open Database Connectivity) y el HTML para hacer sus peticiones.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

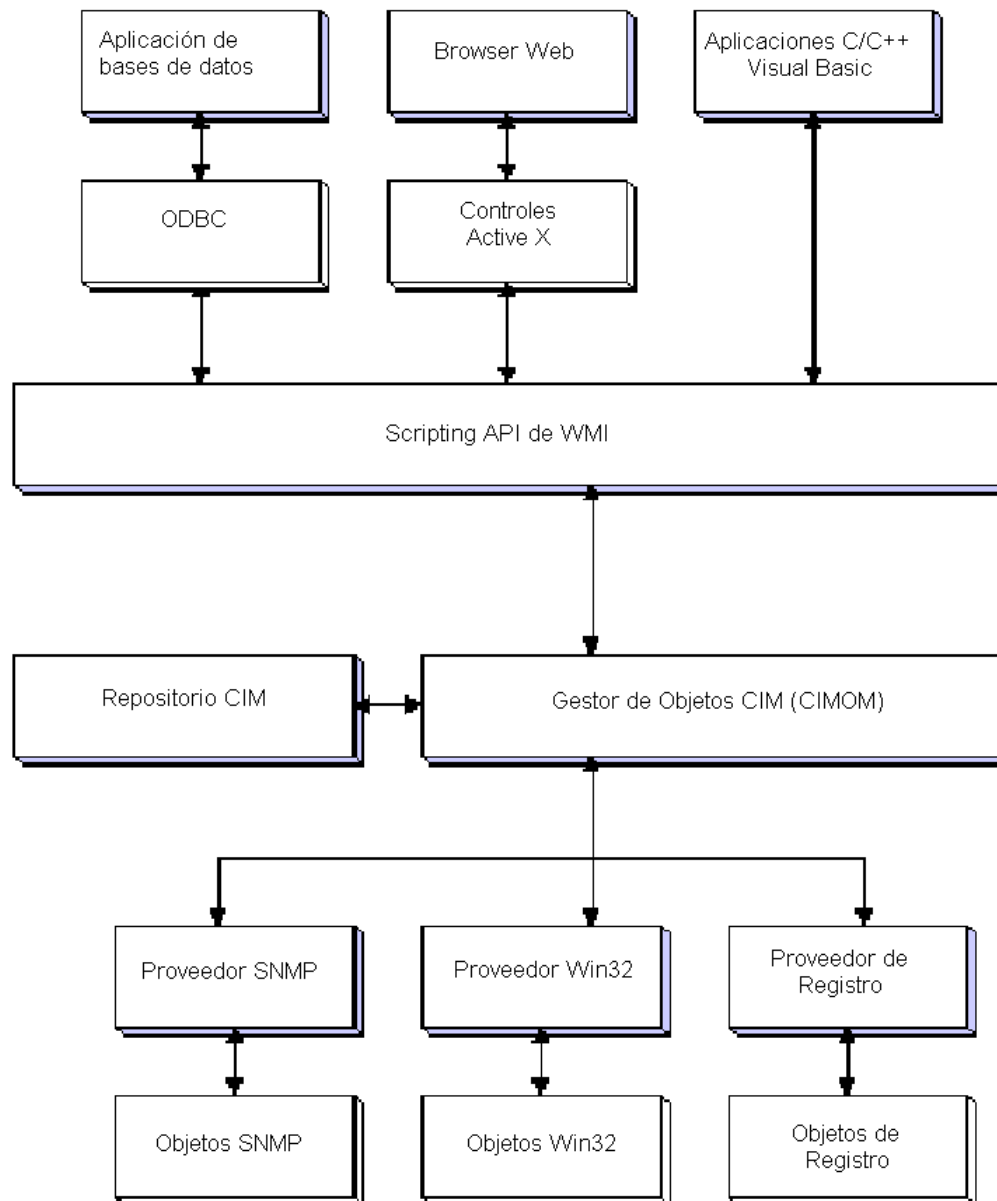


Figura 3.1 Relación entre componentes WMI¹³

¹³ Fuente: Tutorial WMI SDK

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

En la parte baja del diagrama se ilustran ejemplos de objetos gestionados y sus proveedores correspondientes. Hay una variedad de métodos de acceso usados para la comunicación.

El protocolo usado para comunicación entre los componentes locales y remotos es Modelo del Objeto Componente Distribuido (DCOM - Distributed Component Object Model).

3.4 Aplicaciones de gestión

Una aplicación de gestión es una aplicación o un servicio Microsoft® Windows NT®/Microsoft® Windows® 2000 que utiliza la información originada desde uno o más objetos gestionados. Para acceder a ella, una aplicación de gestión hace una petición a “WMI” a través de uno de los siguientes métodos: el COM API o el Scripting API.

“WMI” permite a las aplicaciones de gestión implementar una amplia gama de características como reportes de producción en un inventario de red, desplegar información del sistema, responde a eventos, empezar o detener servicios del sistema y envía un comando de expulsión a una unidad de disco con medios de comunicación reemplazables. Estas características son más fáciles de implementar con “WMI”, aunque es posible llevarlos a cabo sin “WMI”.

“WMI” soporta un número de estrategias para llevar a cabo aplicaciones de gestión. Las aplicaciones pueden acceder a “WMI” directamente a través del COM API o Scripting API o indirectamente con uno de los siguientes métodos de acceso:

- Los browsers Web que pueden utilizar un conjunto de controles Microsoft® ActiveX® para controlar la apariencia, relaciones y conducta de datos que relacionan a los objetos gestionados. Los controles son personalizables y pueden ser incluidos en la definición del esquema.
- Los browsers Web también pueden utilizar HTML, el estándar de páginas HTML y el paradigma hyperlink. HTML es soportado a través de un API para Servidor de

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Internet (ISAPI-Internet Server Application Programming Interface) que interactúa con “WMI”.

- Las aplicaciones de bases de datos pueden utilizar el adaptador “WMI ODBC” para unir las capacidades de la base de datos con las capacidades de gestión de “WMI”. Con el adaptador “ODBC”, una aplicación puede utilizar una amplia gama de paquetes basados en “ODBC” y herramientas como Microsoft® Excel y Microsoft® Access.
- Las aplicaciones de servicio de directorio pueden utilizar la extensión “WMI” llamada Interfaz de Servicio de Directorio Activo (ADSI-Active Directory Service Interface) para integrar el servicio y gestión de datos.

El COM API está directamente disponible para programadores de C/C++.

Se puede utilizar el Scripting API para desarrollar script y aplicaciones basadas en Microsoft® Visual Basic® que se pueden utilizar para visualizar o controlar los objetos gestionados. “WMI” también proporciona soporte para los siguientes lenguajes:

- Visual Basic
- Visual Basic para Aplicaciones
- Visual Basic Scripting Edition
- Microsoft® JScript®

La figura 3.2 muestra aplicaciones de gestión de varios tipos que se comunican con objetos gestionados “CIM” de “WMI”. Dos de las aplicaciones utilizan una de las estrategias de acceso; las otras se comunican directamente al “Gestionador de Objetos CIM” de “WMI”.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

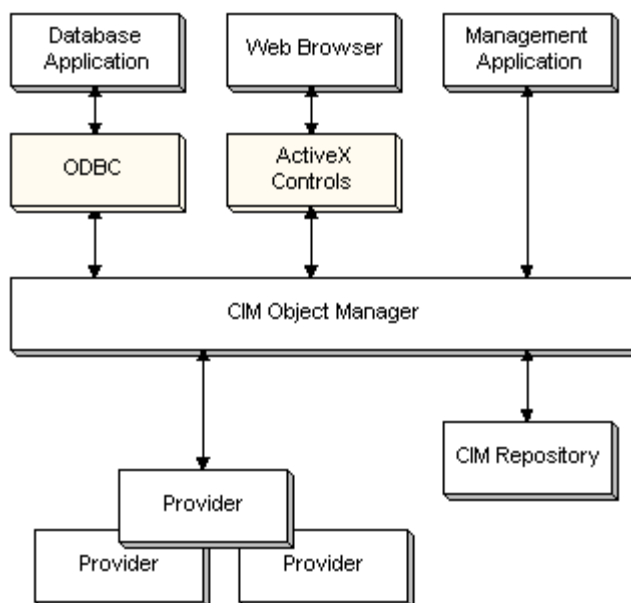


Figura 3.2 Aplicaciones de gestión¹⁴

“WMI”, Directorio Activo, y MMC

Una aplicación de gestión completa a nivel empresarial típicamente comprende tres componentes importantes: “WMI”, el Directorio Activo de Microsoft (MAD-Microsoft® Active Directory™) y la consola de Gestión de Microsoft (MMC- Microsoft Management Console). “WMI” proporciona información específica del computador, como los dispositivos instalados y el software que se está ejecutando además el “MAD” proporciona la vista de la empresa y se enfoca en la información que necesita estar disponible para la red. Se pueden utilizar “WMI” y el Directorio Activo al mismo tiempo para proporcionar una vista completa del ambiente. Puesto que ambas fuentes de información pueden ser necesarias para lograr una tarea dada, “WMI” incluye el Proveedor de Servicios de Directorio para acceder a información guardada en el Directorio Activo.

¹⁴ Fuente: Tutorial WMI SDK

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Se puede utilizar la arquitectura de interfaz de usuario común “MMC” para presentar esta información a los usuarios. La información de “WMI”, “MAD” y otras fuentes pueden ser combinadas con “MMC” para presentar una vista unificada de gestión. Se recomienda que se desarrolle una base embebida para utilizar “WMI” como su fuente de información.

3.5 Proveedores “WMI”

Los proveedores de “WMI” actúan como intermediarios entre “WMI” y uno o mas objetos gestionados. Cuando “WMI” recibe una petición de una aplicación de gestión para datos que no están disponibles en el repositorio “CIM” o para notificaciones de eventos que “WMI” no soporta, “WMI” remite la demanda a un proveedor. Los proveedores proporcionan datos y notificaciones de evento para objetos gestionados que son específicos para un dominio en particular.

Los proveedores son servidores “COM” o “DCOM” estándares que pueden ser implementados utilizando los tipos de servidores soportados:

- Librerías dinámicas en proceso (DLLs)
- Servicios locales o remotos de Microsoft® Windows NT®/Microsoft® Windows® 2000
- Archivos ejecutables estándares, ya sea locales o remotos (.exe files)

Nota: los proveedores que realizan procesos defectuosos pueden impactar negativamente en la fiabilidad del sistema operativo. Los proveedores deben ser implementados como servicios o archivos ejecutables siempre que sea posible.

El “WMI SDK” incluye algunos proveedores incorporados. Los proveedores incorporados, también llamados proveedores normales utilizan fuentes de suministro de datos muy conocidas tales como el registro del sistema y el subsistema Microsoft® Win32®.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

La tecnología “WMI” también soporta la creación de proveedores personalizados por diseñadores de terceras partes. Estos proveedores sirven peticiones relacionadas con objetos gestionados que están en ambientes específicos. Los proveedores utilizan típicamente el lenguaje “MOF” para definir y crear clases, además utilizan el COM API para acceder al repositorio “CIM” y responden a las demandas “WMI” hechas inicialmente por las aplicaciones.

La figura 3.3 ilustra la arquitectura de proveedores. El Proveedor PerfMon, el Proveedor del Registro y el Proveedor WDM son ejemplos de proveedores normales. Proveedor X, Proveedor Y y Proveedor Z son ejemplos de proveedores personalizados. Los proveedores normales proporcionan “WMI” y aplicaciones de gestión con datos obtenidos de objetos gestionados estándares como la aplicación de monitoreo de desempeño, el registro del sistema y el modelos de Drivers de dispositivos Windows (WDM- Windows Drivers Model). Los proveedores personalizados proporcionan un objeto gestor “CIM” de “WMI” y aplicaciones con datos tomados desde objetos personalizados. Los proveedores normales y los personalizados pueden acceder datos del repositorio “CIM” a través del COM API.

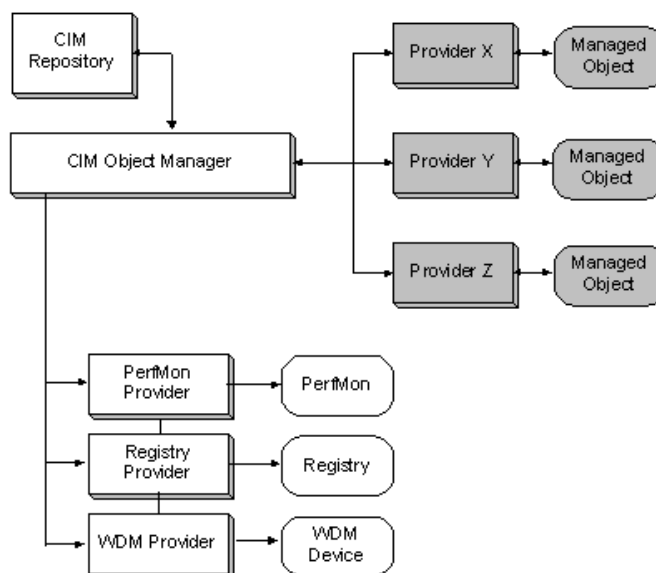


Figura 3.3 Arquitectura de proveedores¹⁵

¹⁵ Fuente: Tutorial WMI SDK

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Hay dos tipos de proveedores básicamente: aquellos que soportan recuperación de datos y modificación, y aquellos que soportan notificación de eventos. Los proveedores tienen las siguientes categorías:

Tipo de proveedor	Descripción
Class	Recupera, modifica, anula, y/o enumera clases específicas de proveedor. También puede soportar procesamientos de consultas.
Instance	Recupera, modifica, anula, y/o enumera las instancias de clases específicas del proveedor. También puede soportar procesamiento de consultas.
Property	Recupera y/o modifica valores de propiedad individuales.
Method	Invoca métodos para una clase específica de proveedor.
Event	Genera notificaciones de eventos.
Event consumer	Soporta notificación de evento para mapear un consumidor físico con un consumidor lógico.

Tabla 13 Categorías de los Proveedores

3.6 Modelo de Información común (CIM)

El Modelo de Información Común presenta una vista consistente y unificada de todos los tipos de objetos lógicos y físicos en un ambiente gestionado. Los objetos gestionados se representan utilizando estructuras orientadas a objetos como clases. Las clases incluyen propiedades las cuales describen los datos y métodos que definen la conducta. El “CIM” está diseñado por la “DMTF” como parte de la iniciativa “WBEM” además “CIM” no depende de una plataforma particular. “WMI” incluye una extensión del “CIM” para plataformas con sistema operativo Microsoft® Windows® .

“WMI” también soporta asociaciones. Las asociaciones son instancias de clases de asociación y son utilizadas para representar relaciones entre otros objetos de “WMI”. Las

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

relaciones de asociación son visibles a las aplicaciones de gestión. “WMI” define las clases de asociación para soportar clases del sistema. Los diseñadores de terceras partes también pueden definir clases de asociación para su ambiente de gestión.

“WMI” soporta el concepto de esquemas para agrupar las clases y casos que se utilizan dentro de un ambiente de gestión particular. El “WMI SDK” incluye dos esquemas: el esquema “CIM” y el esquema Win32. El esquema “CIM” contiene las definiciones de las clases para los primeros dos niveles del “CIM”. Estas clases representan objetos gestionados que son parte de cada ambiente de gestión sin tener en cuenta la plataforma. El esquema Win32 contiene definiciones de las clases para objetos gestionados que son parte de un ambiente típico Win32.

Los diseñadores de terceras partes pueden crear sus propios esquemas personalizados para describir ambientes específicos del vendedor. Como lo que se quiere son esquemas infinitamente extensibles, los diseñadores siempre pueden agregar nuevas clases para representar nuevos objetos gestionados en un ambiente existente. Los esquemas personalizados son extensiones del esquema “CIM” o Win32.

3.7 WinMgmt.exe

WinMgmt.exe es el componente primario en la infraestructura de gestión de “WMI”. En computadores corriendo Microsoft® Windows® 98, WinMgmt.exe corre como un archivo ejecutable normal. En computadores corriendo Microsoft® Windows NT®/Microsoft® Windows® 2000, WinMgmt.exe corre como un servicio. En ambos casos, WinMgmt.exe empieza cuando la primera aplicación de gestión hace una llamada para conectarse. WinMgmt.exe se activa cuando la primera aplicación del cliente se conecta con éxito y corre continuamente cuando las aplicaciones de gestión buscan sus servicios activamente.

En computadores corriendo Windows NT versión 4.0, WinMgmt.exe corre como uno de los dos modos, como un servicio manual o como un servicio automático. Como un servicio automático, WinMgmt.exe entra en un estado de baja memoria cuando todos los clientes

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

empiezan. Como un servicio manual, WinMgmt.exe espera 30 segundos y baja el servicio cuando todos los clientes hayan finalizado.

“WMI” soporta dos interfaces de programación conocidas como el COM API para “WMI” y el Scripting API para “WMI”. El COM API es utilizado para aplicaciones de gestión, para proveedores, y para extensiones del esquema COM. El COM API, habilita sus componentes para interactuar con “WMI” utilizando C y C++ y esta basado en el “COM”. El Scripting API puede ser utilizado para crear aplicaciones cliente con “WMI” utilizando lenguajes de scripting y Microsoft® Visual Basic®.

Cuando una aplicación hace una petición llamando a uno de los dos métodos ya sea el COM API o el Scripting API, “WMI” determina si la petición involucra datos estáticos almacenados en el repositorio “CIM”, o datos dinámicos dados por un proveedor. Los datos estáticos pueden ser manejados por “WMI”, mientras que los datos dinámicos siempre involucran al proveedor. Los proveedores registran su localización y soporte para operaciones particulares como adquisición de datos, modificación, eliminación, enumeración y procesos de consulta. “WMI” utiliza esta información registrada para hacer coincidir los requerimientos de la aplicación con el proveedor adecuado, encontrar y leer dicho proveedor cuando es necesario. Cuando el proveedor ha finalizado el procesamiento de una petición, este retorna los resultados a “WMI”, el cual reenvía los resultados a la aplicación.

Adicionalmente para implementar cada una de las operaciones asociadas con las peticiones de las aplicaciones, “WMI” también proporciona lo siguiente:

- Soporte para notificación de evento
- Soporte para el lenguaje de petición
- Soporte de seguridad
- Soporte de localización

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

3.7.1 Soporte para la Notificación de Evento

“WMI” soporta el registro para la distribución de notificaciones de evento. Los consumidores de evento se deben registrar para recibir tipos particulares de notificaciones y los proveedores de evento se deben registrar para proporcionar tipos particulares de notificaciones.

Para permitir a los consumidores de evento operar independientemente de los proveedores de evento “WMI” actúa como el intermediario para emparejar a los consumidores registrados con los proveedores correspondientes y envía los eventos apropiados.

“WMI” soporta consumidores de evento temporales y consumidores de evento permanentes. Los consumidores de evento temporales sólo reciben notificaciones mientras que estén activos, cuando ellos terminan, su registro es removido. Los consumidores de evento permanentes reciben notificaciones siempre y cuando ellas ocurran. Cuando un consumidor de evento permanente es registrado, “WMI” no termina automáticamente con la última aplicación; se reinicia automáticamente después de que el sistema se reinicia además “WMI” debe estar en todo momento disponible para entregar éstas notificaciones de evento.

Además de los eventos generados por proveedores de evento, el propio “WMI” produce dos tipos de eventos: eventos cronometrados y los eventos intrínsecos. Los eventos cronometrados ocurren periódicamente según un intervalo de tiempo especificado, o una vez en un determinado tiempo y los eventos intrínsecos corresponden a los cambios en datos guardados en el repositorio “CIM”.

Siempre que sea posible, los proveedores deben publicar los eventos intrínsecos pertinentes. Esto previene la pérdida de la ejecución que sería el resultado de registrar por defecto este evento.

Además de publicar eventos intrínsecos que corresponden a los cambios en los objetos gestionados controlados por el proveedor, los proveedores pueden necesitar a veces publicar

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

eventos extrínsecos, es decir, eventos que no corresponden claramente a cualquier cambio en el estado de un objeto gestionado.

“WMI” remite las notificaciones de todos los tipos de eventos a aplicaciones que se han registrado para recibirlos.

3.7.2 Soporte para el lenguaje de consulta

“WMI” actualmente soporta el Lenguaje de Consulta del Instrumental de Gestión de Windows (WQL- Windows Management Instrumentation Query Language) para manejar el proceso de consulta. “WQL2 es un subconjunto del estándar ANSI llamado Lenguaje Estructurado de Consulta (SQL-Structured Query Language) con extensiones “WMI” específicas. Algunas extensiones soportan registro y notificación de eventos, mientras que otros soportan relaciones entre las clases e instancias. “WQL” es un lenguaje de solo lectura el cual no puede ser utilizado para actualizar, insertar o anular datos. “WQL” sólo puede ser utilizado para recuperar información.

3.7.3 Soporte de seguridad

“WMI” soporta una forma limitada de seguridad que positivamente identifica y valida a cada usuario antes de que al usuario se le permita conectarse a “WMI”. “WMI” no soporta protección de clases individuales o instancias de datos dinámicos pero si soporta la protección individual de las clases e instancias de datos dinámicos a través del uso de personificación.

“WMI” también soporta seguridad para “namespaces”¹⁶ individuales.

¹⁶ “namespaces” significa espacio para nombres. Para mayor comodidad se dejará éste término en inglés.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

En las plataformas Windows NT/Windows 2000, no hay ninguna distinción entre acceso local y remoto, excepto que para conectarse remotamente a un namespace de “WMI”, dado que es un derecho separado que puede o no concederse. Sin embargo, con una conexión remota, los usuarios pueden especificar un nombre del usuario y contraseña y pueden reemplazar el nombre del usuario actual y contraseña. Con una conexión local, los usuarios no pueden cambiar el nombre actual y contraseña.

En plataformas Windows 95 y Windows 98 se concede a los usuarios locales todos los derechos, no hay ninguna autenticación. Sin embargo, los usuarios remotos se validan utilizando instancias del sistema de clases “WMI”.

Los administradores pueden utilizar la aplicación de control “WMI” (Wbemcntl.exe) para poner permisos “WMI” a los usuarios.

3.8 Descripción de los objetos “WMI” según el esquema WIN_32

Para el desarrollo de la aplicación de supervisión se utiliza el esquema WIN_32 en el cual están definidos los objetos con sus métodos y propiedades a través de los cuales se lleva a cabo la obtención de la información de gestión necesaria para la supervisión de un PC en particular.

A continuación se describirán los objetos utilizados para esta aplicación en particular.

3.8.1 SWbemLocator

Se puede utilizar los métodos del objeto SWbemLocator para obtener un objeto SWbemServices que representa una conexión a un namespace en una computadora del

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

organizador local o remoto. Se puede utilizar entonces los métodos del objeto SWbemServices para acceder “WMI”.

Métodos

La tabla siguiente lista el método para el objeto SWbemLocator.

Método	Descripción
ConnectServer	Conecta a “WMI” en el computador especificado

Tabla 14 Método del objeto SWbemLocator

Propiedades

La tabla siguiente lista la propiedad para el objeto SWbemLocator:

Propiedad	Descripción
Security_	Usada para leer o cambiar los de seguridad.

Tabla 15 Propiedad del objeto SWbemLocator

En la aplicación no fue necesario utilizar la propiedad Security_ por lo cual solo se hablará del método ConnectServer.

3.8.1.1 SWbemLocator.ConnectServer

El método ConnectServer del objeto SWbemLocator conecta al namespace en el computador especificado en el parámetro strServer. Esta puede ser un computador designado local o remota, típicamente con un sistema operativo Microsoft® Windows NT®/Windows® 2000 o Microsoft® Windows® 95/98. El computador designado debe tener “WMI” instalado. Si el

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

método tiene éxito, un objeto SWbemServices es retornado para que sea ligado al namespace en el computador del organizador.¹⁷

3.8.2 SWbemObjectSet

Un objeto SWbemObjectSet es una colección de objetos SWbemObject.

Se puede conseguir un objeto SWbemObjectSet llamando cualquiera de los siguientes métodos o sus equivalentes asíncronos:

SWbemServices.InstancesOf
 SWbemServices.SubclassesOf
 SWbemObject.Instances_
 SWbemServices.ExecQuery
 SWbemServices.AssociatorsOf
 SWbemServices.ReferencesTo
 SWbemObject.Associators_
 SWbemObject.References_
 SWbemObject.Subclasses_

Nota: El objeto SWbemObjectSet no soporta adición (Add) opcional y remueve (Remove) métodos de colección.

Métodos

La tabla siguiente lista el método para el objeto SWbemObjectSet.

¹⁷ Para una información más completa de este objeto ver el anexo C.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Método	Descripción
Item	Recupera un objeto de SWbemObject de la colección. Este es el método predefinido (por defecto) del objeto.

Tabla 16 Método del objeto SwbemObjectSet

Propiedades

La tabla siguiente lista las propiedades para el objeto SWbemObjectSet.

Propiedad	Descripción
Count	El número de items en la colección.
Security_	Usado para leer o cambiar las configuraciones de seguridad.

Tabla 17 Propiedad del objeto SwbemObjectSet

En nuestra aplicación no fue necesaria la utilización de ningún método ni propiedad. Solo se utilizó este objeto para capturar una colección de objetos del Repositorio.

3.8.3 SWbemServices

Se pueden utilizar los métodos de un objeto SWbemServices para realizar operaciones en un namespace en un organizador local o remoto.

Métodos

La tabla siguiente lista los métodos para el objeto SwbemServices

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Método	Descripción
AssociatorsOf	Retorna una colección de objetos (clases o instancias) que son asociadas con un objeto especificado. Este método desempeña la misma función que el ASSOCIATORS de WQL.
AssociatorsOfAsync	Retorna asincrónicamente una colección de objetos (clases o instancias) que son asociadas con un objeto especificado. Este método desempeña la misma función que el ASSOCIATORS de WQL.
Delete	Elimina una instancia o clase.
DeleteAsync	Elimina asincrónicamente una instancia o clase.
ExecMethod	Ejecuta un método de un objeto.
ExecMethodAsync	Ejecuta asincrónicamente un método de un objeto.
ExecNotificationQuery	Ejecuta un requerimiento para recibir eventos.
ExecNotificationQueryAsync	Ejecuta asincrónicamente un requerimiento para recibir eventos.
ExecQuery	Ejecuta un requerimiento para recuperar una colección de objetos (clases o instancias).
ExecQueryAsync	Ejecuta asincrónicamente un requerimiento para recuperar una colección de objetos (clases o instancias).
Get	Recupera una clase o instancia.
GetAsync	Recupera asincrónicamente una clase o instancia
InstancesOf	Retorna una colección de instancias de una clase especificada.
InstancesOfAsync	Retorna asincrónicamente una colección de instancias de una clase especificada.
ReferencesTo	Retorna una colección de objetos (clases o instancias) que se refieren a un objeto simple. Este método desempeña la misma función que el REFERENCES de WQL.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

ReferencesToAsync	Retorna asincrónicamente una colección de objetos (clases o instancias) que se refieren a un objeto simple. Este método desempeña la misma función que el REFERENCES de WQL.
SubclassesOf	Retorna una colección de subclases de una clase especificada.
SubclassesOfAsync	Retorna asincrónicamente una colección de subclases de una clase especificada.

Tabla 18 Métodos del objeto SWbemServices

Propiedades

La tabla siguiente lista la propiedad para el objeto SWbemServices.

Propiedad	Descripción
Security_	Utilizada para leer o cambiar los configuraciones de seguridad.

Tabla 19 Propiedad del objeto SwbemServices

3.8.3.1 SWbemServices.ExecQuery

El método ExecQuery del objeto SWbemServices ejecuta una pregunta para recuperar objetos. Estos objetos están disponibles a través de la colección SWbemObjectSet retornada.

Este método se utiliza para llamar los objetos que se quieren supervisar y que están definidos por el esquema WIN_32. Por ejemplo, si se quiere consultar las características del procesador, se consulta de la siguiente manera:

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

“ Select * From Win_32Processor ”¹⁸

3.8.4 SwbemObject

Se pueden utilizar los métodos y propiedades del objeto **SWbemObject** para representar una definición de clase “WMI” en particular o una instancia de un objeto.

Este objeto soporta dos tipos de propiedades y métodos. Aquellos definidos en esta sección son propiedades genéricas y métodos que se aplican a todos los objetos “WMI”, adicionalmente estos objetos exponen las propiedades y métodos como una automatización dinámica de las propiedades y métodos de dichos objetos. Dichos tipos de métodos y propiedades dependen del objeto “WMI” que se seleccione.

Desde la perspectiva del cliente “WMI”, este objeto esta siempre en proceso. Las operaciones de escritura solo afectan la copia local del objeto, y las operaciones de lectura siempre retornan valores desde la copia local. Las actualizaciones de “WMI” son desarrolladas solo cuando todos los objetos son escritos utilizando una llamada al método **SWbemObject.Put**. Si se modifican las propiedades o métodos en un objeto **SWbemObject** , los cambios no pueden ser escritos a “WMI” mientras se llame a **SWbemObject.Put**.

El método genérico y los nombres de las propiedades definidas en esta sección siempre terminan con un trailing underscore (“_”) para diferenciarlos de los métodos “WMI” dinámicos.

Métodos

La siguiente tabla muestra la lista de métodos para el objeto **SWbemObject**.

¹⁸ Para más información de el objeto SwbemServices, ver el anexo C.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Método	Descripción
Associators_	Recupera las asociaciones del objeto.
AssociatorsAsync_	Recupera asincrónicamente las asociaciones del objeto.
Clone_	Hace una copia del objeto actual.
CompareTo_	Prueba la igualdad entre dos objetos.
Delete_	Borra el objeto de “WMI”.
DeleteAsync_	Borra asincrónicamente el objeto de “WMI”.
ExecMethod_	Ejecuta un método exportado por un proveedor de métodos.
ExecMethodAsync_	Ejecuta asincrónicamente un método exportado por un proveedor de métodos.
GetObjectText_	Recupera la representación textual del objeto (Sintaxis “MOF”) .
Instances_	Retorna una colección de instancias del objeto (que puede ser una clase “WMI”).
InstancesAsync_	Retorna asincrónicamente una colección de instancias del objeto (puede ser una clase “WMI”).
Put_	Crea o actualiza un objeto en “WMI”.
PutAsync_	Crea o actualiza asincrónicamente el objeto en “WMI”.
References_	Retorna las referencias del objeto.
ReferencesAsync_	Retorna asincrónicamente las referencias del objeto.
SpawnDerivedClass_	Crea una clase derivada del objeto actual (la cual puede ser una clase “WMI”).
SpawnInstance_	Crea una nueva instancia desde el objeto actual.
Subclasses_	Retorna una colección de subclases del objeto (que puede ser una clase “WMI”).
SubclassesAsync_	Retorna asincrónicamente una colección de subclases del objeto (que puede ser una clase “WMI”).

Tabla 20 Métodos del objeto SWbemObject

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Propiedades

La siguiente tabla muestra una lista de las propiedades del objeto **SWbemObject**.

Propiedad	Descripción
Derivation_	Contiene un arreglo de caracteres describiendo la derivación jerárquica para las clases.
Methods_	Un objeto SWbemMethodSet que es la colección de métodos para este objeto.
Path_	Contiene un objeto SWbemObjectPath que representa la ruta del objeto de la clase o la instancia actual.
Properties_	Objeto SWbemPropertySet que es la colección de propiedades para este objeto.
Qualifiers_	Objeto SWbemQualifierSet que es la colección de calificadores para este objeto.
Security_	Contiene un objeto SWbemSecurity usado para leer o cambiar la configuración de seguridad.

Tabla 21 Propiedades del objeto SWbemObject

Este objeto se utiliza solo para capturar un objeto específico de una colección de objetos.¹⁹

WMI es una parte clave de la estrategia de Microsoft para hacer de Windows la plataforma de sistema operativo más administrable para sus clientes . Basado en estándares de la industria y respaldado por el DMTF, WMI proporciona un modelo de objeto extensible que permite que los sistemas de computadores sean administrados de una forma consistente y que se pueda manejar con scripts, ya sea local o remotamente. La mayoría de la información y servicios del sistema operativo central están actualmente instrumentados como parte del sistema básico de WMI que viene con Windows 2000. La instrumentación para otros productos de Microsoft y

¹⁹ Para más información sobre éste objeto vea el anexo C.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

de otras compañías está disponible y en desarrollo, haciendo de WMI el mejor camino para administrar sistemas basados en Windows de una forma escalable, confiable y unificada.

Debido a la facilidad de uso de WMI es factible realizar diversas aplicaciones de gestión utilizando diferentes tecnologías como Visual Basic para la creación de programas con manejo de Bases de datos y para el control de los objetos gestionados, HTML para el despliegue de la información a través de páginas Web y WAP como otro medio de acceder a la información de gestión a través de dispositivos móviles.

Sobre estas aplicaciones se hablará en el siguiente capítulo.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

4. APLICACIONES Y ESQUEMAS DE ACCESO A LA INFORMACION DE GESTION

4.1 Aplicaciones

La gestión tiene muchas caras, pero puede ser dividido en las siguientes disciplinas:

Area	Descripción
Cambio y Gestión de la Configuración	Administración del sistema, gestión de estado, y gestión del ciclo de vida del software.
Gestión de seguridad	Acceso del usuario, autenticación, y seguimiento.
Gestión de desempeño	Trazado, puesta a punto, aplicaciones modeladas y redes; también, provisión a nivel de servicio.
Gestión del problema	Aislamiento del error, diagnósticos, registro del problema, y consolidación de medios de ayuda.
Gestión de eventos	Monitoreo de la información del sistema, consolidación, agregación, y entrega.
Gestión de Batch/Output	Ejecución del trabajo, programa de dependencia, control de submisión, y cobro revertido.
Gestión de almacenamiento	Administración de hardware de almacenamiento, protección de los datos, y ubicación.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

4.1.1 Soluciones de valor agregado.

Estos pueden incluir lo siguiente:

- Cambio en la empresa y en la configuración
- Seguridad heterogénea
- Puesta a punto y monitoreo del desempeño de extremo- a -extremo
- Seguimiento de problemas
- Manejo de evento en la empresa
- Resultados del servicio
- Gestión de la empresa

Todas las áreas de gestión mencionadas anteriormente, pueden ser soportadas por aplicaciones WAP utilizando la plataforma de gestión de Microsoft WMI, alcanzando un gran campo de servicios tanto para los administradores como para los clientes que quieran hacer uso de estos recursos.

A continuación se mencionan algunas de las diversas aplicaciones que se pueden lograr utilizando la plataforma WMI.

4.1.2 Aplicaciones con WMI

Hay tantas cosas que se pueden hacer con WMI, que no hay forma de dar un ejemplo representativo de cada subsistema. Para ilustrar algunas de las capacidades de WMI, se han seleccionado varios ejemplos. Estos ejemplos incluyen:

- Enumerar una lista de todos los servicios instalados y luego identificar aquellos servicios que están detenidos pero están establecidos en el modo de arranque "Automático".

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

- Hacer una lista de todas las particiones lógicas de disco en el computador y determinar si alguna tiene menos del 20 por ciento de espacio libre.
- Cambiar el retardo de tiempo antes de que la selección predeterminada del sistema operativo sea iniciado.
- Ejecutar un respaldo del registro de eventos para aplicación y luego limpiar el archivo de registro.
- Reiniciar el sistema de otro equipo de trabajo. (Se tiene que dar los permisos de seguridad para hacer esto).
- Ejecutar el programa Notepad.exe de Windows a través de un método WMI.
- Establecer un consumidor de eventos para vigilar cualquier nuevo evento que sea añadido al registro de eventos.
- Reconfigurar el script consumidor de eventos para requerir un evento cuando la utilización de la CPU exceda el 50 por ciento.

4.1.2.1 Hacer una lista de todos los servicios en el sistema.

En este ejemplo, se utiliza la clase Win32_Service para entregar una lista de todos los servicios que están instalados en el sistema, sin importar si se están ejecutando o no:

```
Set ServiceSet =
GetObject("winmgmts:{impersonationLevel=impersonate}").
InstancesOf("Win32_Service")
for each Service in ServiceSet
    WScript.Echo Service.Description
Next
```

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Primero, requiere un objeto en la llamada `GetObject()` del servicio WMI (`WinMgmt`). Se hace esto pidiendo todas las instancias de la clase `Win32_Service` a través del método `InstancesOf()`. El resultado de esta llamada es una lista de instancias de la clase `Win32_Service`.

Después de obtener las instancias de vuelta en la variable `ServiceSet`, todo lo que resta por hacer es iterar a través de ellas y desplegar la propiedad de interés, el campo "Description", el cual contiene el nombre propio del servicio. Es importante notar aquí que la propiedad "Description" está siendo manejada como una propiedad de automatización del objeto "Service". Alternativamente, exactamente los mismos resultados pueden ser obtenidos al utilizar una petición SQL (consulta SQL) y enviándola al método:

```
ExecQuery() :
Set ServiceSet =
GetObject("winmgmts:{impersonationLevel=impersonate}").
ExecQuery ("select * from Win32_Service")
```

La petición SQL pide que todas las instancias (y que todas las propiedades de esas instancias) sean seleccionadas y regresadas. El conjunto de objetos regresados son exactamente iguales a los retornados con el método anterior, pero como será claro en los ejemplos que siguen, la forma de petición puede ser utilizada para refinar el número de propiedades, instancias o ambas, regresando solo aquellas de interés.

Una nota final acerca de este ejemplo y aquellos que siguen que involucran el modelo de seguridad que es utilizado. Se puede haber visto el enunciado "{impersonationLevel=impersonate}" como parte de la llamada `GetObject()` mostrada antes. En esencia, esto le dice al sistema que utilice las credenciales actuales del inicio de sesión (login) como aquellas que serán presentadas cuando se pregunte por datos o se ejecuten métodos.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

4.1.2.2 Hacer una lista los servicios automáticos que están detenidos.

Si ahora se desea ver solo aquellos servicios que están establecidos para ejecutarse automáticamente, pero por alguna razón no están ejecutándose (por ejemplo, un usuario manualmente detuvo un servicio), solamente se necesita hacer un cambio a la versión de petición del script, como se muestra aquí:

```
Set ServiceSet =
GetObject("winmgmts:{impersonationLevel=impersonate}").
ExecQuery("select * from Win32_Service where State='Stopped'
and StartMode='Auto'")
for each Service in ServiceSet
    WScript.Echo Service.Description
Next
```

Igual que si se estuviera requiriendo información más específica de una base de datos SQL, todo lo que fue necesario para obtener la lista de los servicios que estaban detenidos era añadir una cláusula WHERE a la petición. La petición especificada estrecha los resultados que se obtienen a solo aquellas instancias que tienen una propiedad "State" con el valor "Stopped" y una propiedad "StartMode" establecida en "Auto."

Las peticiones pueden ser muy sencillas o bastante complejas, dependiendo de qué requerimiento específico se desee.

4.1.2.3 Hacer una lista de todas las particiones del disco con menos de 20 por ciento de espacio libre.

Para poder determinar si una partición de dispositivo está baja de espacio, se necesita obtener dos propiedades de la clase Win32_LogicalDisk: FreeSpace y Size. Con estas dos piezas de

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

información, el porcentaje de espacio libre puede ser determinado y aquellas particiones de disco que están por debajo del límite pueden ser identificadas:

```
Set DiskSet =
GetObject("winmgmts:{impersonationLevel=impersonate}").
ExecQuery("select FreeSpace,Size,Name from Win32_LogicalDisk
where DriveType=3")
for each Disk in DiskSet
    If (Disk.FreeSpace/Disk.Size) < 0.20 Then
        WScript.Echo "Drive " + Disk.Name + " is low on space."
    End If
Next
```

Se puede notar que en esta petición, solo se requieren las propiedades necesarias para construir el script, no todo dentro de la clase Win32_LogicalDisk. Para poder obtener la información de solo los discos duros locales, la cláusula WHERE limita la petición a solo ese tipo de dispositivo (DriveType=3). Sin este filtro, los dispositivos floppy, los CD-ROM y los volúmenes compartidos en la red también serían incluidos en la verificación, lo cual probablemente no sería deseable. La propiedad "Name" es también obtenida y utilizada cuando se necesita desplegar un mensaje.

4.1.2.4 Estableciendo el retardo de arranque del sistema operativo.

En este script se utiliza la clase Win32_ComputerSystem para modificar el ajuste del retardo utilizado al momento de arranque para dar al usuario tiempo para seleccionar una opción diferente de arranque del sistema operativo predeterminado. Una diferencia principal en este script es que una propiedad en la clase Win32_ComputerSystem será modificada y escrita de vuelta a la WMI, la cual a su vez actualiza el ajuste del sistema operativo:

```
Set CompSysSet =
GetObject("winmgmts:{impersonationLevel=impersonate}").
```

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

```
ExecQuery("select * from Win32_ComputerSystem")
for each CompSys in CompSysSet
    CompSys.SystemStartupDelay = 20
    CompSys.Put_()
next
WScript.Echo "Boot up delay time set"
```

Para realizar esta tarea, la clase Win32_ComputerSystem es retornada con una petición como se hizo anteriormente. El script itera sobre cada instancia (en el caso de esta clase, siempre hay solo una instancia debido a que solo hay una instancia sistema del computador para una máquina dada) y establece el valor de la propiedad SystemStartupDelay dentro de la variable temporal al del ajuste deseado.

Finalmente, para escribir la instancia de vuelta con el valor modificado, el método Put_() es invocado. Debe notarse que varias propiedades de las instancias podrían haber sido actualizadas con la misma llamada Put_(). El mismo mecanismo puede ser utilizado para cualquier propiedad que habilite la escritura utilizada por WMI.

Un valor de 20 segundos se escribe en la propiedad, el cual puede ser verificado después de ejecutar el script en la aplicación Panel de Control -> Sistema bajo la pestaña Avanzados, opción Arranque y Recuperación, como se muestra en la Figura 4.1

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

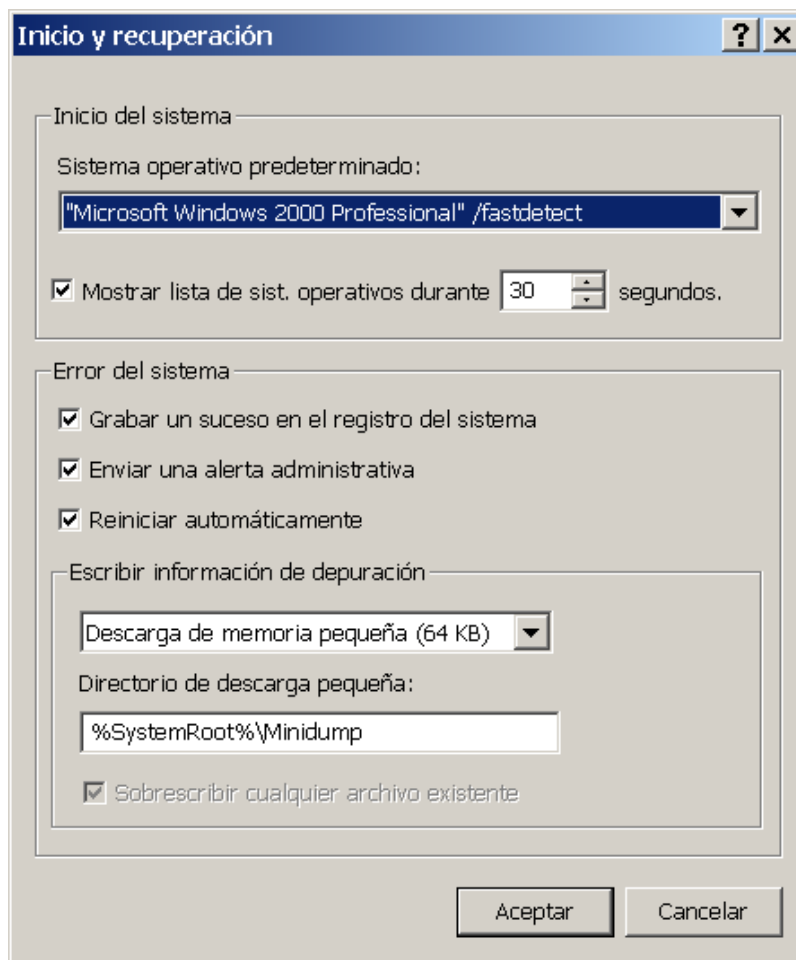


Figura 4.1 Opciones de Arranque y Recuperación.

4.1.2.5 Respaldo del registro de eventos.

La clase Win32_NTEventLogFile modela los registros de eventos de Windows NT 4.0 y Windows 2000. De manera predeterminada, hay tres registros de eventos estándar: Sistema, Seguridad y Aplicación. En este ejemplo se invoca el método para respaldar uno de estos registros. Los métodos dentro de WMI son sencillamente funciones que son aplicadas a objetos administrados particulares; aquí, el método de respaldo se aplica a una instancia de objeto Registro de Evento de Windows NT:

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

```

LogFileSet =
GetObject("winmgmts:{impersonationLevel=impersonate,(Backup)}")
).ExecQuery("select * from Win32_NTEventLogFile where
LogfileName='Application'")
for each Logfile in LogFileSet
   RetVal = LogFile.BackupEventlog("c:\BACKUP.LOG")
    if RetVal = 0 then WScript.Echo "Log Backed Up"
next

```

Lo básico permanece igual, es decir, se hace una petición a la clase Win32_NTEventLogFile, filtrándola para obtener el archivo "Application" de interés. Note la adición del enunciado "(Backup)" dentro de la llamada GetObject(). Para poder respaldar un archivo de eventos, el privilegio "Backup" debe ser asignado a la cuenta. Los miembros de los grupos de seguridad Administrators y Backup Operators de NT tienen este privilegio predeterminado. Cuando se ejecuta una operación privilegiada tal como un "Backup", el privilegio debe ser explícitamente habilitado por el usuario antes de ser utilizado. Si el "Backup" no se hubiese especificado en la llamada GetObject(), el script hubiera recibido el mensaje "Access Denied" cuando el método BackupEventlog() fuera invocado y la operación no sería realizada. El método BackupEventlog() toma la ruta y el nombre del archivo de respaldo como sus únicos parámetros.

Para borrar las anotaciones en el registro de eventos de Aplicaciones después de que se ha respaldado, el script requiere de solo una pequeña modificación:

```

LogFileSet =
GetObject("winmgmts:{impersonationLevel=impersonate,(Backup)}")
).ExecQuery("select * from Win32_NTEventLogFile where
LogfileName='Application'")
for each Logfile in LogFileSet
   RetVal = LogFile.BackupEventlog("c:\BACKUP.LOG")
    if RetVal = 0 then WScript.Echo "Log Backed Up"
    RetVal = LogFile.ClearEventlog()

```

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

```

    ifRetVal = 0 then WScript.Echo "Log Cleared"
next

```

El área en negrilla en el ejemplo anterior añade la funcionalidad de borrar el archivo “logfile” al llamar al método ClearEventlog() y al obtener un resultado exitoso, visualiza un mensaje.

4.1.2.6 Reiniciar una máquina remota.

Las operaciones remotas en la WMI son virtualmente idénticas a aquellas realizadas localmente. Todos los ejemplos previos podrían ser fácilmente modificados para especificar una máquina remota como objetivo, en vez de tomar el sistema local de manera predeterminado. Cualquier instrumento, propiedad o método puede ser accesado entre individuos o entre una empresa completa utilizando un sencillo script. Aquí, se reiniciará un sistema remoto utilizando el método Reboot() en la clase Win32_OperatingSystem:

```

Set OpSysSet =
GetObject("winmgmts:{impersonationLevel=impersonate,(RemoteShutdown)}/ryst10 ").ExecQuery ("select * from
Win32_OperatingSystem where Primary=true")
for each OpSys in OpSysSet
    OpSys.Reboot()
Next

```

Para construir este script, se tiene que preguntar a la máquina remota por la instancia del sistema operativo que está ejecutando, en este caso aquel con una bandera “Primary” establecida en TRUE. Naturalmente el nombre de la máquina, “ryst10”, necesita ser especificado para que el método sea dirigido al sistema correcto. Aparte de especificar la máquina objetivo, nada es diferente acerca de cómo es obtenida la información en este caso, de una máquina local.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

De nuevo, note que esta es una operación privilegiada. (Se debe ser parte del grupo de Administrador en la máquina remota para utilizarla exitosamente). Para que el script funcione, se debe tener el privilegio para realizar un apagado remoto asignado en el sistema remoto y se debe habilitar explícitamente utilizando "(RemoteShutdown)" antes de intentar la operación privilegiada.

4.1.2.7 Ejecutar el Bloc de Notas con WMI.

En este ejemplo, se crea un nuevo proceso a través de un método WMI en la clase Win32_Process. Se ha escogido una aplicación relativamente inocua, el Bloc de Notas (Notepad), para propósito de demostración, pero se puede sustituir por cualquier ejecutable. Esta capacidad puede ser extremadamente poderosa para personal de soporte que interactúa con usuarios relativamente inexpertos. Por ejemplo, en vez de guiar al usuario a través de la interfaz de usuario, los menús y demás para ejecutar una herramienta de diagnóstico, la persona de soporte puede sencillamente hacer aparecer la herramienta en el sistema del usuario utilizando un script que lo invoque remotamente por medio de WMI. Esto por sí solo puede ser un gran ahorrador de tiempo:

```
set process =
GetObject("winmgmts:{impersonationLevel=impersonate}!Win32_Process")
result = process.Create ("notepad.exe",null,null,processid)
WScript.Echo "Method returned result = " & result
WScript.Echo "Id of new process is " & processed
```

Hay un par de nuevos conceptos en este ejemplo. El primer desarrollo nuevo es la utilización de la notación de apodos (monikers) COM en la llamada GetObject() cuando pide la clase Win32_Process. Un apodo es un mecanismo COM estándar para encapsular la localización y adhesión de otro objeto COM. Esto le permite regresar a un objeto particular basado en el nombre que despliega. Después, en vez de enviar una petición a través de ExecQuery() para

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

regresar las instancias de la clase Win32_Process, se ha pedido regresar el objeto de clase en sí mismo. La razón para esto es bastante sencilla, pero puede no ser obvia. Crear un proceso nuevo no es realmente una acción para ser tomada por una instancia de proceso existente. ¿Qué proceso que se ejecuta actualmente debería de ser utilizado para lanzar uno nuevo? Debido a que no hay una respuesta consistente para esta pregunta, se vuelve aparente que crear un proceso es realmente crear una nueva instancia de la clase Win32_Process. Por tanto el concepto de método estático, uno definido contra la definición de clase en vez de sus instancias, se hace necesario. El método Create() para Win32_Process es un ejemplo de tal método estático; los ejemplos de métodos previos, han sido todos métodos dinámicos, esto son, aquellos que operan contra instancias individuales. Incidentalmente el método Delete() es un método dinámico, debido a que típicamente se aplica a instancias particulares en vez de a clases completas.

El método Create() toma varios parámetros de entrada, pero en este ejemplo sencillamente le proporciona el nombre del ejecutable que se desea crear, "notepad.exe". Aparte de regresar un valor para el éxito o el fracaso de la operación, el método, además, regresa el ID del nuevo proceso que fue creado. El script despliega los resultados de ejecución del método y el ID del proceso después de que se ejecuta el método y, por lo tanto, la aplicación Notepad debe aparecer en el escritorio.

4.1.2.8 Recolectando eventos del registro (log) de Windows NT en WMI.

WMI puede abastecer eventos para muchos tipos de cambios u ocurrencias dentro del sistema. En este ejemplo, el script se construye para registrar cualquier evento nuevo que se escriba en los registros de eventos de Windows NT o Windows 2000. Cuando un nuevo evento es escrito, WMI genera un evento WMI que un consumidor de eventos como este script puede recibir. Es importante notar, como sea, que los eventos WMI son solo enviados cuando al menos un consumidor se ha registrado para recibirlos. Si no hay nadie escuchando, WMI no gasta recursos vigilando una ocurrencia dada dentro del sistema:

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

```

set events =
GetObject("winmgmts:{impersonationLevel=impersonate}").ExecNot
ificationQuery_ ("select * from __instancecreationevent where
targetinstance isa 'Win32_NTLogEvent'")
if err <> 0 then
    WScript.Echo Err.Description, Err.Number, Err.Source
end if
' Note this next call will wait indefinitely - a timeout can
be specified
WScript.Echo "Waiting for NT Events..."
do
    set NTEvent = events.nextevent
    if err <> 0 then
        WScript.Echo Err.Number, Err.Description, Err.Source
        Exit Do
    else
        WScript.Echo NTEvent.TargetInstance.Message
    end if
loop
WScript.Echo "finished"

```

Al utilizar el método `ExecNotificationQuery()`, el script envía una petición definiendo los tipos de eventos que desea se le envíen cuando estos ocurran. Cuando un evento es escrito en un “logfile”, en efecto crea una instancia de la clase `Win32_NTLogEvent`. WMI puede detectar cuando una instancia es creada, modificada o eliminada, y esta información es la fuente del evento que buscamos. El enunciado de petición SQL en efecto le dice a WMI que "envíe cualquier evento de creación de instancia cuando la instancia que está siendo creada sea de la clase `Win32_NTLogEvent`." Los eventos de creación de instancia de otras clases no serán reportados dado que no cumplen el criterio de la petición.

El script sencillamente espera en un ciclo, para que los eventos lleguen y luego despliega el mensaje de la propiedad de las instancias recibidas.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

4.1.2.9 Recolectando eventos sobre la utilización alta del CPU.

En el ejemplo anterior los eventos estaban siendo brindados por el proveedor del Registro de Eventos directamente y eran manejados por el proveedor que era llamado de vuelta por el servicio del Registro de Eventos al irse anotando eventos nuevos. No todos los servicios proveen tales mecanismos para reportar eventos interesantes, ni siempre es razonable esperar que lo hagan. En vez de no tener eventos para alguna instrumentación, WMI tiene un mecanismo interconstruido para generar eventos de donde no existe soporte subyacente de eventos. De hecho, una propiedad que esté instrumentada, ya puede ser fuente de eventos. En este ejemplo se demostrará como la utilización de la CPU puede ser monitorizada y cuando el procesador está severamente cargado, se generará un evento y se enviará a los consumidores registrados. Esto está disponible aún y cuando no haya mecanismo de eventos para esta medición en particular construida en el sistema operativo:

```
set events =
GetObject("winmgmts:{impersonationLevel=impersonate}").ExecNot
ificationQuery _ ("select * from __instancemodificationevent
within 5 where targetinstance isa 'Win32_Processor'
and targetinstance.LoadPercentage > 50")
if err <> 0 then
    WScript.Echo Err.Description, Err.Number, Err.Source
end if
' Note this next call will wait indefinitely - a timeout can
be specified
WScript.Echo "Waiting for CPU load events..."
WScript.Echo ""
do
    set NTEvent = events.nextevent
    if err <> 0 then
        WScript.Echo Err.Number, Err.Description, Err.Source
        Exit Do
    else
```

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

```
WScript.Echo NTEvent.TargetInstance.DeviceID WScript.Echo
NTEvent.TargetInstance.LoadPercentage
end if
loop
WScript.Echo "finished"
```

En el script se modifica la versión anterior del script del registro de eventos de Windows NT para vigilar si la CPU está siendo severamente cargada. Se debe notar que ahora se está vigilando eventos por modificación en vez de creación de instancias como en el ejemplo anterior. Además, la clase que está siendo monitorizada es la clase Win32_Processor y existe una restricción adicional en que la propiedad “LoadPercentage” (la cual es un miembro de la clase Win32_Processor) que debe tener un valor mayor a 50. Debido a que WMI esta monitoreando el valor de la propiedad LoadPercentage y generando eventos cuando su valor llena el criterio de la petición, necesita un parámetro que le diga que tan seguido monitorear ese valor. Este es el propósito de la cláusula WITHIN en la petición. La cláusula WITHIN define el máximo intervalo de tiempo que debería pasar antes de que WMI verifique el valor de la propiedad para ver si es que ha cambiado y llena los requerimientos de la petición. Este valor puede también ser visto como la máxima cantidad de tiempo que el consumidor puede esperar antes de obtener el evento deseado.

Igual que antes, cuando el criterio de la petición se satisface, WMI envía un evento al script, el cual a su vez imprime cierta información acerca del evento, en este caso son mostrados el nombre del CPU y el porcentaje real medido de carga. Incidentalmente, este script funciona igualmente bien en sistemas con CPU múltiple o sencilla. Debido a que el script busca cualquier modificación de instancia que afecta a una instancia de la clase Win32_Processor, reportará cualquier CPU que exceda el criterio especificado de carga. Naturalmente, si solo una CPU en particular fuera de interés, la petición podría ser fácilmente modificada para limitar su enfoque.

Esta misma funcionalidad puede ser utilizada para cualquier propiedad que sea visible por WMI, ya sea que pertenezca a una parte de la instrumentación provista en el sistema operativo o añadida por otras compañías. No se necesita hacer absolutamente nada de trabajo

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

adicional para hacer que esto suceda; es una característica estándar que incluida al instalar el producto con WMI. Si el API subyacente es compatible con un sistema de eventos interno que pueda entregar la misma información en una forma tal vez más optimizada, el escritor del proveedor puede fácilmente suplantar el monitoreo de eventos de WMI sin que el consumidor de eventos tenga que estar al tanto de la diferencia. Por tanto, un consumidor obtendrá automáticamente los beneficios del mejor mecanismo de eventos disponible sin hacer un solo cambio a la forma en que funciona. WMI no monitorizará ninguna propiedad en la forma descrita, a menos que exista un consumidor de eventos registrado para este evento en particular. Una vez que el registro es eliminado (por ejemplo, cuando terminar el script) WMI ya no vigilará los eventos solicitados.

4.1.3 Avances de la tecnología inalámbrica

Para soportar soluciones de red en el software del consumidor, dispositivos electrónicos del consumidor, y los aparatos que pueden conectar, Microsoft está trabajando en un conjunto de tecnologías inalámbricas para habilitar un conjunto robusto de los campos utilizables por el usuario para las Redes de Area Local (LANs), Redes de Area Personal (PANs) y Redes de Area Ancha (WANs).

Por ejemplo, Microsoft está desarrollando y está desplegando un juego de servicios y aplicaciones para casas y negocios, como Universal Plug and Play, la Web Inalámbrica, servicios MSN®, mejoras inalámbricas relacionadas con Microsoft Outlook®, y más.

Con este desarrollo se podrá realizar la gestión a través de cualquier dispositivo inalámbrico no solo WAP sino desde cualquier otro dispositivo basándose en la tecnología de gestión WMI.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

4.2 Esquemas de acceso a la información de gestión

4.2.1 WAP

²⁰Una vez terminada la aplicación se realizaron pruebas reales con el terminal WAP R280D de Ericsson conectado a la red de COMCEL obteniéndose resultados satisfactorios. Debido a la poca disponibilidad de este terminal, la aplicación se desarrollará a través de un equipo con el emulador R280D.

4.2.2 Web

Otra de las aplicaciones con las cuales se puede acceder a la información de gestión es la tecnología Web a través de la cual se puede realizar una presentación más amigable al usuario y de fácil acceso para aquellos usuarios que no dispongan de un terminal WAP.

La figura 4.2 es un ejemplo de presentación de la información de gestión de un equipo remoto.

²⁰ Debido a que WAP es una especificación muy extensa se hará referencia a ella y en especial a lenguaje WML en el anexo B.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

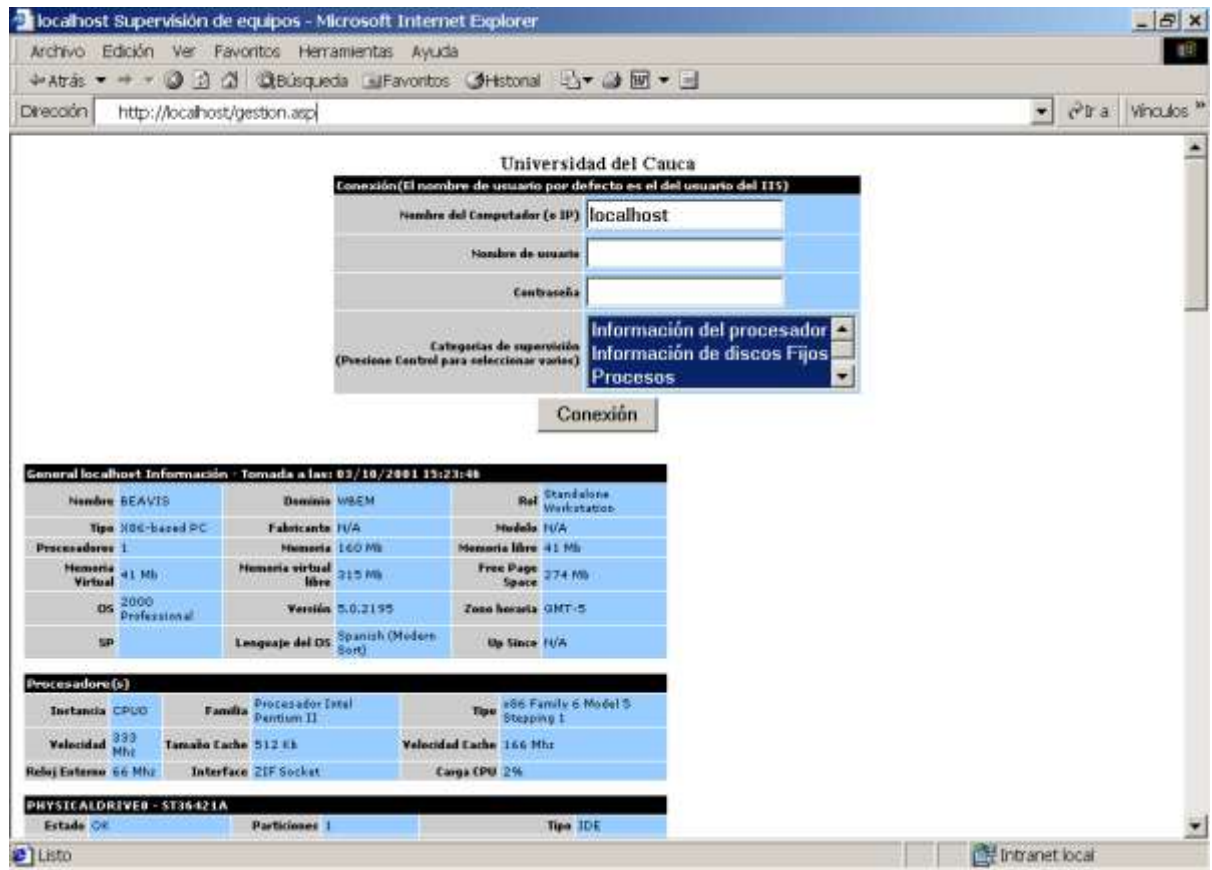


Figura 4.2 Interfaz Web

4.2.3 Visual Basic

Con este lenguaje de programación se puede crear una base de datos que contenga solo la información de gestión que el usuario desee supervisar del equipo remoto, además a través de Visual Basic se pueden realizar operaciones de control como apagar un equipo remoto, reiniciarlo, detener y reiniciar servicios dentro de un servidor Windows.

En la figura 4.3 se presenta la interfaz con la cual se puede observar y controlar las diferentes áreas de gestión para un equipo en particular.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

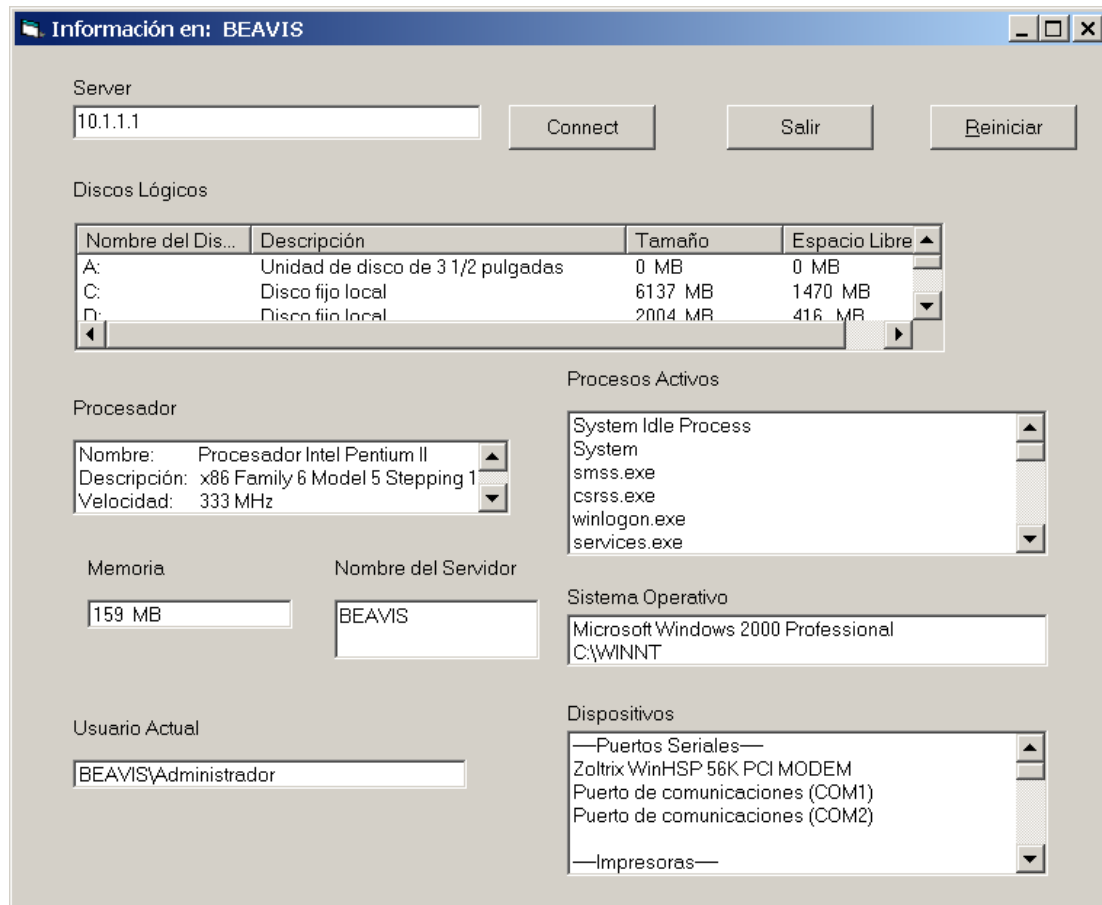


Figura 4.3 Interfaz Visual Basic

Como se pudo observar en este capítulo, el campo de aplicación de la plataforma WMI es muy extenso y no solo se limita a la supervisión de equipos sino que se puede extender a diversas áreas de gestión como los son el manejo de fallas, la configuración de equipos, contabilidad en los servicios, el desempeño de los elementos a gestionar y la seguridad de los mismos.

La aplicación desarrollada se limita a la supervisión de equipos dentro de una LAN, pero se espera que en un futuro no muy lejano se pueda gestionar dichos equipos no solo a nivel de supervisión sino a nivel de control, utilizando un terminal WAP desde cualquier parte del mundo.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

Una manera de extender el campo de gestión de WMI es a través SNMP. Debido a que SNMP es una tecnología ampliamente utilizada y difundida, con la cual se gestionan dispositivos dentro de redes LAN y WAN tales como Enrutadores, Repetidores, Puentes, etc. WMI ha desarrollado un soporte para este protocolo con el cual se puedan configurar y administrar los parámetros de dichos equipos.

Para tal fin WMI ha desarrollado en el SDK las siguientes áreas:

- **Mapeo de SNMP al esquema CIM:** SNMP utiliza un esquema para definir objetos el cual es diferente al esquema utilizado en WMI.
- **Guía SNMP:** Esta guía es diseñada para ayudar a los usuarios a recibir información desde un dispositivo SNMP.
- **Soporte SNMP:** Se explican los componentes que soportan SNMP como: Proveedores de clases, Proveedores de instancias, proveedores de eventos e información sobre el compilador de información SNMP.

Con este soporte WMI no se limita a la gestión de dispositivos que utilicen el sistema operativo Windows sino que podrá gestionar toda clase de dispositivos convirtiéndose en una plataforma universal de gestión.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

BIBLIOGRAFIA

Páginas Web

WBEM

www.dmtf.com

WMI

www.microsoft.com

WAP y WML

www.wmlclub.com

www.nokia.com

www.wapforum.com

XML

www.w3c.com

ASP

www.eggheadcafe.com

www.asp.com

www.aspespanol.com/tutorial/ASP.htm

Buscadores

www.google.com

www.altavista.com

www.yahoo.com

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

GLOSARIO

ADSI (Analog Display Services Interface)

ADSI es el protocolo estándar para habilitar servicios alternados de voz y de datos como un teléfono con despliegue visual sobre una red telefónica análoga. Desarrollada por Bellcore en 1993, ADSI esta ahora incluido en dispositivos telefónicos, decodificadores de Televisión por cable. Asistentes digitales personales y en computadores con aplicaciones telefónicas.

API (Application Program Interface)

Una API es el método específico prescrito por un sistema operativo de un computador o por un programa de aplicación con el cual un programador puede desarrollar un software haciendo peticiones al sistema operativo o a otra aplicación.

CIM (Common Information Model)

El Modelo de Información Común, es un modelo común de datos de un esquema neutral de implementación que sirve para describir la información de gestión global en un ambiente de red. CIM comprende una especificación y un esquema. La especificación define los detalles para la integración con otros modelos de gestión (por ejemplo: Las MIBs de SNMP o las MIFs del DMTF) mientras que el esquema provee la descripción actual del modelo.

CMIP (Common Management Information Protocol)

Es un protocolo de gestión de red construido sobre el modelo de comunicación OSI. El Servicio Común de Información de Gestión (CMIS-Common Management Information Services) define los servicios para acceder a la información de objetos de red o dispositivos, controlándolos y recibiendo reportes de estado desde ellos.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

COM (Component Object Model)

El Modelo de Objeto Componente es un modelo de programación que define la interacción entre objetos con un proceso o entre procesos. En COM, los clientes tienen acceso a un objeto a través de una interfaz implementada sobre el objeto.

ISAPI (Internet Server Application Program Interface)

ISAPI es un conjunto de llamadas a los programas Windows que permiten escribir una aplicación de servidor Web la cual podrá correr más rápido que una aplicación de Pasarela de Interfaz Común (CGI-Common Gateway Interface). Usando ISAPI se puede crear un archivo de una aplicación con librerías dinámicas que puede correr como parte del protocolo http.

Método M-POST

El método M-POST es la extensión del método POST, la M viene de Mandatory, este método es usado para las especificaciones en Frameworks.

Método POST

El método POST se usa para hacer peticiones en las que el servidor destino acepta el contenido de la petición como un nuevo subordinado del recurso pedido. El método POST se creó para cubrir funciones como la de enviar un mensaje a grupos de usuarios, dar un bloque de datos como resultado de un formulario a un proceso de datos, añadir nuevos datos a una base de datos, etc.

La función llevada a cabo por el método POST está determinada por el servidor y suele depender de la URI de la petición. El resultado de la acción realizada por el método POST puede ser un recurso que no sea identificable mediante una URI.

RMON (Monitoreo Remoto)

Es una especificación de Monitoreo estándar que permite que varios monitores de red y sistemas de consola intercambien datos sobre el monitoreo de la red. RMON proporciona a

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

los administradores de red información muy completa acerca del diagnóstico de fallas de red, planeación y puesta a tono del sistema. El RMON es capaz de monitorear un segmento Ethernet y transmitir información estadística de regreso a la consola para RMON

SNMP (Simple Network Management Protocol)

El Protocolo Simple de Gestión de Red es una tecnología de amplia difusión y uso. Es un protocolo usado en conjunto con Redes de Area Local (LAN) y Redes de Area Amplia (WAN) y es principalmente usado para configurar y administrar diversos parámetros en Enrutadores, Repetidores, Puentes y otros.

SSL (Secure Sockets Layer)

El SSL es un protocolo comúnmente utilizado para gestionar la seguridad de la transmisión de mensajes sobre Internet. SSL usa una capa programable localizada entre el Protocolo de Transferencia de Hipertexto (http) y la capa del Protocolo de Control de Transporte (TCP). SSL está incluido como parte de los navegadores de Microsoft, Netscape y en la mayoría de productos de servidores Web.

Aplicación de supervisión dentro del modelo de gestión WBEM a través de terminales móviles WAP

ACRONIMOS

CIM Modelo de Información Común

CMIP Protocolo Común de Información de Gestión

CSS Hoja de Estilo en Cascada

DMTF Grupo de Trabajo de Gestión Distribuída

DTD Definición de Tipo de Documento

EG Equipo Gestionado

MIB Base de Información de Gestión

MIF Formato de Información de Gestión

MOF Formato de Objeto Gestionado

RMON Monitoreo Remoto

SNMP Protocolo Simple de Gestión de Red

SSL Capas de Sockets de Seguridad

TW Terminal con emulador WAP

WBEM Gestión Empresarial Basada en Web

W3C Consorcio World Wide Web

XML Lenguaje Extendido de Marcas

XSL Lenguaje de Hoja de estilo Extensible