

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PARA MONITOREO  
REMOTO DE UN SERVIDOR GNU/LINUX POR MEDIO DE DISPOSITIVOS  
MÓVILES

CATALINA CASAS ARÉVALO  
JUAN CARLOS VALENCIA ORTIZ

UNIVERSIDAD DE SAN BUENAVENTURA  
FACULTAD DE INGENIERÍA  
INGENIERÍA DE SISTEMAS  
BOGOTÁ D.C.  
2010

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PARA MONITOREO  
REMOTO DE UN SERVIDOR GNU/LINUX POR MEDIO DE DISPOSITIVOS  
MÓVILES

CATALINA CASAS ARÉVALO  
JUAN CARLOS VALENCIA ORTIZ

Proyecto de grado como requisito para optar el título de Ingeniero de Sistemas.

Asesor  
Carlos Andrés Lozano Garzón  
Ingeniero de Sistemas

UNIVERSIDAD DE SAN BUENAVENTURA  
FACULTAD DE INGENIERÍA  
INGENIERÍA DE SISTEMAS  
BOGOTÁ D.C.  
2010

Nota de aceptación:

---

---

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Bogotá D.C. (18,05,2010)

Esfuerzos, sacrificios, largas jornadas de trabajo, pocas de descanso que materializaron este sueño conforme al propósito de Dios, a Él por ser mi refugio y mi fortaleza, por guiar e iluminar mi camino, a mi mamá por ser el modelo perfecto a seguir, a mi familia por su presencia, por contagiarme de entusiasmo y por enseñarme que el secreto del éxito está en ser feliz. A ti Leo por ser mi esperanza, por enseñarme que lo que se pide con las fuerzas del alma siempre se hace realidad y este es el momento perfecto para decir “solo me interesa nuestro futuro, porque en el vamos a pasar el resto de nuestras vidas...”

Gracias a todos y cada uno de ustedes.

“La vida no se mide por las veces que respiras, sino por aquellos momentos que te dejan sin aliento” Hitch.

Catalina Casas Arévalo

A Dios que cada día me muestra el camino correcto y el amor que me tienen dándome en todo momento excelentes oportunidades, las mejores cosas de mi vida y la fortaleza para soportar los malos momentos. A mi mamá quien a estado conmigo brindándome su amor y consejos. A mi esposa por su paciencia, amor A mi familia por su respaldo

A mis verdaderos amigos por estar conmigo y darme su apoyo incondicional

A todas las personas que con su conocimiento y experiencia me han compartido su sabiduría, enseñado y apoyado para mejorar profesionalmente y como persona.

Juan Carlos Valencia Ortiz

## **AGRADECIMIENTOS**

Queremos agradecer a nuestro asesor Carlos Andrés Lozano Garzón por su valioso aporte y apoyo en la elaboración de nuestro proyecto de grado.

A nuestros amigos Andrea, Alexander, Luis Carlos y Diego que nos acompañaron en tan largas jornadas de estudio y aportaron sus conocimientos para hacer de este proyecto un sueño hecho realidad.

También queremos extender nuestro agradecimiento a la Universidad de San Buenaventura, a todo su equipo de docentes y funcionarios.

## CONTENIDO

	pág
INTRODUCCIÓN .....	16
1. PLANTEAMIENTO DEL PROBLEMA.....	18
1.1 ANTECEDENTES.....	18
1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA .....	19
1.3 JUSTIFICACIÓN .....	21
1.4 OBJETIVOS DE LA INVESTIGACIÓN .....	22
1.4.1 Objetivo general .....	22
1.4.2 Objetivos específicos.....	22
1.5 ALCANCES Y LIMITACIONES DEL PROYECTO.....	22
1.5.1 Alcances .....	22
1.5.2 Limitaciones.....	22
2. METODOLOGÍA .....	24
3. LÍNEA DE INVESTIGACIÓN.....	25
3.1_LÍNEA DE INVESTIGACIÓN .....	25
3.2 LÍNEA DE INVESTIGACION DE FACULTAD .....	25
3.3 NÚCLEO PROBLÉMICO.....	25
4. MARCO DE REFERENCIA.....	26
4.1 MARCO TEÓRICO - CONCEPTUAL .....	26
4.1.1 Telefonía móvil celular.....	26
4.1.2 WAP - Protocolo de aplicaciones inalámbricas .....	30
4.1.3 WML (Wireless Markup Language) .....	37
4.1.4 UML (Unified Modeling Language).....	39

4.1.5 HSQLDB .....	41
4.1.6 RMI (Remote Method Invocation) .....	41
4.1.7 Servidores LINUX .....	44
4.1.8 Estándar 830-1988 IEEE .....	46
4.2 MARCO LEGAL .....	48
5. DESARROLLO INGENIERIL .....	49
El análisis está compuesto por las etapas de análisis e ingeniería del sistema y el análisis de requisitos del modelo de cascada. ....	49
5.1. ANÁLISIS .....	49
5.1.1 Especificación de requerimientos .....	49
5.1.2 Requerimientos específicos .....	50
5.1.3 Vista de casos de uso .....	51
5.2 DISEÑO .....	54
5.2.1 Diseño del agente .....	54
5.2.2 Diseño interfaz conexión.....	59
5.2.3 Diseño servidor .....	59
5.3 IMPLEMENTACIÓN .....	65
5.3.1 Plataforma utilizada.....	65
5.3.2 IDE .....	67
5.3.3 Construcción del software .....	68
5.3.4 Estructura del código .....	69
5.3.5 Diagrama de componentes .....	71
5.3.6 Interfaz gráfica .....	71
6 PRESENTACIÓN Y ANÁLISIS DE RESULTADOS .....	73
6.1 HERRAMIENTAS DE EJECUCIÓN DE PRUEBAS.....	73
6.1.1 JUnit .....	73

6.1.2 TestNG .....	73
6.2 PLANIFICACIÓN DE LA PRUEBA .....	74
6.3 PREPARACIÓN DEL ESCENARIO.....	74
6.3.1 Configuración del equipo para el Agente.....	74
6.3.2 Configuración de equipo para el Servidor .....	74
6.3.3 Configuración red .....	76
6.4 PRUEBAS DE INSTALACIÓN Y CONFIGURACIÓN .....	76
6.4.1 Instalación del agente .....	76
6.4.2 Instalación del servidor.....	76
6.5 PRUEBAS DE EJECUCIÓN .....	77
6.6 RESULTADOS DE LAS PRUEBAS.....	77
7. CONCLUSIONES .....	79
8. BIBLIOGRAFÍA.....	80



## LISTA DE TABLAS

	pág
Tabla 1: Caso de uso Ver lista Servidores .....	51
Tabla 2: Caso de uso Ver Información Servidores.....	52
Tabla 3: Caso de uso Ver Información Adicional Servidor.....	53
Tabla 4 Equipos .....	64
Tabla 5 Usuarios .....	65
Tabla 6: Comparativo de plataformas .....	66
Tabla 7: Comparativo IDE .....	68
Tabla 8 Número de dispositivos vs Acción .....	78

## LISTAS DE FIGURAS

	pág
Figura 1. Esquema de modelo de cascada.....	24
Figura 2. Evolución de la tecnología móvil en Colombia (2003-2010).....	28
Figura 3. Analogías de servicios ofrecidos en los móviles. ....	28
Figura 4. Arquitectura WAP .....	31
Figura 5. Funcionamiento de WAP .....	34
Figura 6: Estructura WAP vs WEB.....	35
Figura 7: Funcionamiento en Internet de WAP y WEB.....	36
Figura 8: Pila de protocolo WAP .....	37
Figura 9: Arquitectura Java RMI .....	42
Figura 10 Opciones en atsar .....	45
Figura 11: Diagrama de clases del Agente .....	56
Figura 12: Diagrama de colaboración –Obtener información del sistema ....	56
Figura 13: Diagrama de colaboración –Obtener información extra del sistema.....	57
Figura 14: Diagrama de secuencia – Obtener información del sistema .....	57
Figura 15: Diagrama de secuencia – Interfaz conexión.....	59
Figura 16: Diagrama de clases – Componente Servidor.....	60
Figura 17: Diagrama de secuencia – Login y Menú.....	61
Figura 18: Diagrama de secuencia – Ver información del equipo.....	62
Figura 19: Diagrama de secuencia – Ver información extra del equipo .....	63
Figura 20: Diagrama de colaboración – Servidor.....	64
Figura 21: Diagrama de componentes .....	71
Figura 22: Servidores registrados.....	71

Figura 23: Información del equipo .....	72
Figura 24: Listado y detalle de reportes .....	72
Figura 25 Administrador de sucesos Windows .....	75
Figura 26 Monitor del sistema de Linux.....	76

## GLOSARIO

**API:** (Application Program Interface). Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.

**CDC:** Configuración orientada a dispositivos con menos limitaciones.

**CDMA:** Code Division Multiple Access. Estándar de telefonía para sistemas de tercera generación implementado en su gran mayoría en América y una parte de Asia, que permiten una mayor capacidad para el envío y transmisión de datos, ya que los digitaliza y los comprime.

**CLDC:** Configuración orientada a dispositivos con limitaciones computacionales y de memoria, este tipo de configuración es el utilizado para las aplicaciones de los dispositivos celulares.

**CONFIGURACIONES:** Conjunto mínimo de APIs Java, define el entorno de ejecución y librerías de clases disponibles a una categoría de dispositivos.

**CONTROLADOR:** Almacena las comunicaciones y la lógica de la aplicación.

**FAMILIA DE DISPOSITIVOS:** Es un grupo de categorías que comparten las mismas características en cuanto a la capacidad de memoria y almacenamiento.

**GPRS:** General Packet Radio Service. Permite el envío y recepción de datos mediante el protocolo IP. Además permite envío y recepción de mensajes de multimedia (MMS) y ampliar el número de caracteres de un mensaje corto de texto (SMS).

**GSM:** (Global System for Mobile Communications) El sistema más utilizado es la comunicación vía radio, funciona con tres entidades que se interconectan: estación móvil que es el Terminal del usuario, estación base que controla la comunicación del radio Terminal y el sistema de red que realiza la conexión entre los usuarios. Todo esto es lo que permite todas las aplicaciones y servicios que ofrecen hoy en día las empresas de telefonía móvil, tiene una velocidad de transmisión de 14,4 kbps.

**HTTP:** HyperText Transfer Protocol. Es empleado en transferencias de información de páginas de Internet mediante un

navegador o explorador. Éste solo se refiere al mecanismo empleado para ser llevado entre los servidores y el usuario final. Es un mecanismo que se emplea para la comunicación entre servidores. Es empleado en implementaciones de J2ME o protocolo WAP.

HTTPS: Secure HiperText Transfer Protocol. Es un protocolo de comunicación entre servidores con un alto grado de encriptación para enviar o recibir datos mediante un puerto más seguro que el de HTTP. Es implementado en aplicaciones Web o de servicios móviles.

JDK: Java Development Kit. Es un compilador de desarrollo para crear los diferentes programas Java. Es esencial para todo programador de Java no sólo como herramienta de desarrollo sino también como fuente de documentación del propio lenguaje.

J2ME: Java 2 Micro Edition. Es un subconjunto de J2SE (Java 2 Standard Edition) que está destinado a realizar aplicaciones para dispositivos con pocos recursos tanto en la capacidad de memoria, limitaciones gráficas y capacidad de procesamiento.

KVM: Se trata de una implementación de Máquina Virtual reducida y especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria.

MIDLET: Es una aplicación java realizada con el perfil MIDP sobre la configuración CLDC.

MIDP: Perfil para dispositivos con reducida capacidad computacional y de memoria: Conectividad limitada y Entrada de datos alfanumérica.

MODELO: Almacena preferencias, constantes y datos.

PERFIL: Conjunto de Apis que controlan: Ciclo de vida de la aplicación, Interfaces de usuario, Gestión de eventos.

PROTOCOLO: Serie de pasos que involucra a dos o más principales, diseñado para realizar una tarea en particular.

RMI: (Java Remote Method Invocation) es un mecanismo ofrecido en Java para invocar un método remotamente. Al ser RMI parte estándar del entorno de ejecución Java usarlo provee un mecanismo simple en una aplicación distribuida que solamente necesita comunicar servidores codificados para Java. Si se requiere comunicarse con otras tecnologías debe usarse CORBA o SOAP en lugar de RMI.

SMS: Short Message System. Es un servicio para el envío y recepción de mensajes cortos de no más de 160 bytes a través de la red

telefónica inalámbrica.

**TELEFONÍA MÓVIL:** Sistema de telefonía que no requiere un enlace fijo para la recepción y transmisión. Emplea ondas haciendo que una antena transmita éstas hasta una antena repetidora cercana.

**UMTS:** Universal Mobile Communication Service. Estándar de telefonía para sistemas de tercera generación implementado en su gran mayoría en países de Europa y parte de Asia.

**URL:** Uniform Resource Locator. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

**WIN:** (Wireless Identity Module) es una especificación que trata de definir un equivalente en el ámbito de WAP del popular SIM, implementado en las tarjetas de los teléfonos GSM y que contiene la identidad del usuario en la red GSM, es una aplicación para tarjetas inteligentes

**WML:** Wireless Markup Language. Es un lenguaje de marcas que permite hacer aplicaciones para dispositivos móviles.

**WSP:** Wireless Session Protocol. Provee métodos para el intercambio de contenidos entre el cliente y el servidor. Es la cuarta capa del modelo WAP.

**WTLS:** Wireless Transport Layer Security. Suministra conexiones para autenticación, privacidad y seguridad entre aplicaciones cuando es necesario. Es la segunda capa del modelo WAP.

**UA:** (User Agent) Siglas de "User Agent". Se refiere al software capaz de interpretar códigos como el WML o WMLScript, que está instalado en nuestro dispositivo WAP.

**WAE:** Siglas de Wireless Application Environment (Entorno de Aplicaciones Inalámbricas). Parte integrada del Protocolo de Aplicaciones Inalámbricas cuyo fin, basado en tecnología para WWW, es especificar todas aquellas materias que permitan a los operadores y proveedores crear aplicaciones y servicios accesibles desde la mayor gama posible de plataformas.

**WAP Gateway:** Dispositivo de dos direcciones a través del cual se convierte el contenido que hay en el WAP Server al formato WML que pueda entender nuestro teléfono WAP. Igualmente, a través de él y desde el punto de vista del servidor de HTTP, el WAP Gateway envía información acerca del dispositivo WAP por medio de los encabezamientos (headers) HTTP.

WAP Server: Básicamente es un servidor HTTP, es decir un servidor de la red Internet como otro cualquiera. En cambio, Nokia llama WAP Server a un conjunto de dos productos: un gateway WAP y un servidor HTTP (servidor de contenidos). Pero un WAP Server es un simple servidor HTTP.

WML Card: (Carta WML) Bloque de navegación WML, dentro de una baraja WML. Dentro de una baraja WML debe existir una o más cartas WML. Cuando el usuario accede a una baraja WML, se le presentará en la pantalla únicamente la primera de sus cartas.

WML deck: Conjunto de cartas WML. Cuando se requiere un URL se carga automáticamente todo el conjunto de cartas, la carta que se especifique en el URL será a la que se acceda. Si no se especifica ninguna en particular, se accederá a la primera.

WMLScript: Lenguaje de programación con base en JavaScript, aunque menos potente, dirigida a los dispositivos WAP.

WSP: Siglas de Wireless Session Protocol. Hace referencia a la aplicación de más alto nivel que ofrece WAP a través de un interface para servicios de dos sesiones. La primera consistiría en un servicio con conexión que operaría sobre el protocolo del nivel de transacción y el segundo sería sin conexión y operaría sobre el servicio de transporte de información.

WTA: Son las siglas de Wireless Telephony Applications, un entorno para aplicaciones de telefonía que permite a los operadores la integración de funciones de telefonía del propio dispositivo móvil con el micronavegador incorporado.

WTAI: Son las siglas de Wireless Telephony Application Interface. Es una interfaz utilizada en los terminales móviles para operaciones locales de control de llamadas (recepción, iniciación y terminación) y de acceso a listines telefónicos. XML: Extensible Markup Language. Partiendo de este lenguaje se ha formado el WML, por tanto el WML tiene su origen en este lenguaje.

## INTRODUCCIÓN

En el mundo actual el gran avance de la tecnología ha permitido el desarrollo de dispositivos móviles que hacen más fácil, rápido y eficiente el acceso a diversas aplicaciones de uso diario, una muestra de este avance es la implementación de servicios por medio de dispositivos móviles mediante los cuales se puede acceder a información de forma inalámbrica, facilitando las labores diarias.

Uno de estos dispositivos es el teléfono celular, el cual ha venido presentando diversos cambios tales como la utilización de redes GSM/GPRS, implementación de protocolos WAP y arquitecturas que permiten el acceso a información de manera remota. Todos estos cambios han permitido la implementación de servicios móviles, que mejoran la vida cotidiana.

La tecnología móvil ha ido evolucionando con el tiempo; los primeros sistemas llamados “de primera generación” eran análogos, de gran tamaño y muy pesados, su costo era elevado sólo funcionaban en zonas específicas y sus funciones eran limitadas. Después aparecieron los llamados “de segunda generación”, que son digitales, transmiten datos a velocidades pequeñas y utilizan el envío de mensajes SMS.

Los proveedores de telefonía celular han mejorado el mercado de los teléfonos celulares y se han enfocado en aplicaciones de servicios móviles, que ofrecen a los múltiples usuarios; abriendo una gran puerta para facilitar las actividades diarias.

En la actualidad la tecnología se ha centrado en la telefonía móvil, puesto que los teléfonos celulares permiten acceder a la información desde cualquier parte y estar en continuo movimiento, así mismo ofrecen diferentes aplicaciones y servicios a los que los usuarios pueden acceder de una manera fácil y rápida, además hoy en día es más fácil acceder a un celular y la mayoría de las personas cuentan con uno, sin distinción alguna. Por todo esto se escogieron los dispositivos móviles celulares como la mejor alternativa para el desarrollo del proyecto.

La primera etapa del proyecto es el análisis, en la cual se mostraran las diferentes arquitecturas que permiten desarrollar aplicaciones en teléfonos celulares, con sus respectivas ventajas y desventajas, permitiendo así tomar una decisión sobre la mejor opción para el desarrollo de la aplicación.



La segunda etapa involucra el desarrollo del proyecto, el modelado UML, la documentación, el diseño y la implementación de la base de datos, el diseño de la interfaz y el diseño completo del prototipo con todos los requerimientos necesarios. La tercera etapa involucra la verificación del correcto funcionamiento del prototipo con las pruebas y con los requerimientos necesarios.

El proyecto busca facilitar a los administradores de redes conocer el estado de los servidores que están bajo su administración, para así tener una información fiable en cualquier momento. Esta información es importante para cumplir sus labores diarias, pero en ocasiones el administrador no se encuentra en su lugar de trabajo o con acceso a una terminal adecuada para monitorear el estado de los servidores. Con este proyecto se busca facilitar el acceso a esa información por medio de dispositivos móviles como lo son los celulares, PDAs o cualquier dispositivo que pueda tener acceso a un red celular por medio de el protocolo WAP.

## 1. PLANTEAMIENTO DEL PROBLEMA

### 1.1 ANTECEDENTES

El 3 de diciembre de 1979 se puso en funcionamiento en Tokyo (Japón) la primera red de Telefonía Móvil Celular (TMC), con el objetivo principal de brindar a la sociedad más herramientas para comunicarse y hacer más eficaz y productivo su trabajo.

Las principales ventajas que ofrece este servicio son la movilidad, la rapidez y la facilidad en la comunicación. De otro lado, la TMC ofrece una extensa gama de servicios, incluidos los prestados por el sistema fijo tradicional, con la ventaja de no tener que conectar a sus usuarios directamente a un cable, permitiéndoles así desplazarse de un sitio a otro.

En el caso de Colombia, la adjudicación de la Telefonía Móvil Celular (TMC) se adelantó a comienzos de 1994 por medio de una de las licitaciones más importantes en la historia del sector de telecomunicaciones del país.

La prestación del servicio de Telefonía Móvil Celular (TMC) en Colombia, se aprobó por el Congreso de la República mediante la Ley 37 de enero de 1993. De acuerdo con esta Ley, se permite al Ministerio de Comunicaciones adjudicar la concesión de la prestación del servicio de TMC, previo un trámite de licitación pública<sup>1</sup>.

La evolución de la telefonía móvil en Colombia se enmarca en los siguientes antecedentes<sup>2</sup>:

Año 1998. Los operadores lanzan sus redes móviles de datos (CDPD / TDMA) para suplir la demanda de servicios y aplicaciones verticales (telemetría, cajeros automáticos entre otros)

Año 2002 Comcel lanza Internet Movil Corcel asociado con Starmedia, Servicios de alertas y contenido a través de SMS.

Año 2003. Los operadores Comcel, Celumovil / Bellsouth y Celcaribe lanzan servicios SMS de contenido y descargas a través de sus sitios web.

---

1 La Telefonía Móvil Celular en Colombia 1998 Documento Sectorial - DEPARTAMENTO NACIONAL DE PLANEACION UNIDAD DE INFRAESTRUCTURA Y ENERGIA

2 Servicios móviles en Colombia – Ministerio de Tecnologías de la Información y las Comunicaciones Noviembre de 2009

Año 2004. Se consolidan los operadores Comcel y Telefónica.

Año 2005. Entra el tercer operador (ColombiaMóvil) e inicia una fuerte competencia de innovación tecnológica (de TDMA / Cdma hacia GSM) y lanzamiento de servicios. La interconexión SMS se hace obligatoria para servicios SMS P2P.

Año 2007. Aparecen las redes 2.5G sobre GSM (EDGE) e inicia una fuerte el proceso de consolidación del negocio de contenidos móviles y servicios de conectividad móvil.

Año 2008-2009. Aparecen las redes 3G, Servicios de video llamadas, plan de datos ilimitados, Iphone

Con respecto a herramientas similares al proyecto propuesto, solo se han encontrado tres programas. Estos son RemoteWap, WapSH y Host Monitor. RemoteWap<sup>3</sup>: Es un software de administración remota para sistemas operativos que puedan correr una Máquina Virtual de Java (JVM). La administración se hace desde un teléfono celular habilitado con soporte WAP. El proyecto se encuentra actualmente abandonado y su última versión 0.4 fue liberada el 26 de julio de 2004.

WapSH<sup>4</sup>: Es una herramienta que permite al usuario realizar un login o inicio de sesión en un servidor Unix, y así ofrecer un shell remoto, todo esto desde una página WML servida por un servidor WAP su última versión 1.17 liberada el 24 de marzo de 2007.

HostMonitor<sup>5</sup> es una aplicación gratuita que permite monitorear servidores en general, paginas webs y todo tipo de servicios. Posee diversos métodos de testeo, el mas general es hacer un ping, indica si está arriba o no disponible, y envía un email a una cuenta especifica en el caso de falla, podría ser un celular si éste lo permite.

## **1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA**

Las empresas cuentan con unidades de tecnología que soportan en gran medida el negocio para el cual están enfocadas, una corta lista de labores de dicha área podría citarse como: la administración de la infraestructura de

---

<sup>3</sup> <http://remote-ap.sourceforge.net/>

<sup>4</sup> <http://www.exolution.de/>

<sup>5</sup> <http://www.ks-soft.net/hostmon.eng/index.htm>

hardware y/o recursos informáticos, administración de software, administración de aplicaciones, monitoreo y gestión de procesos, configuración, mantenimiento, soporte técnico para usuarios entre otras.

Como lo menciona Red Hat Enterprise Linux 4 “La administración de sistemas es mayormente un asunto de balancear los recursos disponibles con la gente y los programas que utilizan esos recursos”<sup>6</sup>. En este mundo globalizado para el cual no existen fronteras ni físicas, ni geográficas y mucho menos técnicas, los administradores de sistemas se han tenido que afrontar a un problema de “acceso” en el cual se pueden identificar 2 grandes limitaciones; la primera limitaciones de acceso físico como por ejemplo no tener acceso en el preciso momento a los servidores por encontrarse en otra oficina u otro lugar bien sea dentro de la misma área de servicio o fuera de la misma y otra limitación común, es no contar con los recursos tecnológicos para acceder a la red.

La distribución de los sistemas tecnológicos al interior de las organizaciones ha creado la necesidad en los departamentos de TI de poder administrar servidores que están físicamente fuera del centro de datos.

Además, las estaciones de trabajo han dejado de contar como pertenencia de los usuarios y han pasado a tomarse como objetos contables asignados a proyectos particulares. Así se debe considerar todo el proceso de TI, por lo que la administración de servicios es parte fundamental en el manejo moderno de los sistemas tecnológicos de información y en el proceso de alinearlos con el negocio.

La administración remota y la de servicios trabajan sobre la capacidad de operar y mantener equipos a distancia, reduciendo los costos relacionados a recursos humanos y permitiendo ejecutar procesos de manera automática que actualicen la disponibilidad de distintos recursos informáticos<sup>7</sup>.

A partir de estos antecedentes se puede formular la siguiente pregunta de investigación

¿Cómo diseñar y que características deberá tener, un sistema de gestión móvil para monitorear las funciones básicas en servidores Linux?

---

<sup>6</sup> Red Hat Enterprise Linux 4: Introducción a la administración de sistemas Capítulo 1. Filosofía de la Administración de Sistemas <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/s1-philosophy-resources.html>

<sup>7</sup> <http://www.arandasoft.com/blog/?p=8>

### 1.3 JUSTIFICACIÓN

En la actualidad, la infraestructura de redes es un área muy importante para la mayoría de las empresas. No disponer de una red puede causar pérdidas financieras inevitables. El administrador de redes debe poder controlar la actividad en la red y llamar a los técnicos rápidamente en caso de congestión o problemas de acceso. También está a cargo de administrar las cuentas de los usuarios, crear cuentas para nuevos miembros del personal y eliminarlas cuando éstos ya no pertenecen a la compañía.

Debido a los cambios vertiginosos de la tecnología y los medios de transmisión, el administrador de redes debe estar alerta a cualquier novedad que presenten los sistemas administrados para ello deberá contar con una herramienta que esté disponible 7 \* 24 y que le permita acceder inmediatamente para verificar y gestionar para continuar garantizándole a la organización la disponibilidad, confiabilidad y fiabilidad de la información.

Internet transformó al mundo en una sola PC donde puede confluir el globo entero sin necesidad de salir de casa. En cuanto al universo empresarial, los beneficios de acortar distancias equivalen a generar ganancias concretas: ahorrar dinero y mejorar la atención y el servicio ante el cliente. En este sentido, términos como acceso remoto y/o gestión remota aparecieron hace unos años y se han transformado en una herramienta de gran utilidad. No obstante, ante el usuario, los proveedores no siempre logran despejar las incógnitas de los mismos.

El concepto de acceso remoto aparece “cuando un usuario o empleado puede acceder a los sistemas de su empresa desde cualquier punto”. Por otra parte, la acción remota “implica disponer de sistemas de gestión que faciliten la administración de los equipos y resuelvan la problemática existente o se anticipen a futuros problemas”.

Existen tres grandes líneas que definen el tipo de gestión remota. “La primera, consiste en usar ofimática o aplicaciones remotas desde un servidor externo de la empresa. Es decir, la atención del hardware. Una segunda línea aparecería cuando un proveedor brinda asistencia remota: si la empresa tiene un problema en un equipo, un proveedor externo se puede conectar a ésta a través de herramientas del sistema operativo, tomar el control y evacuar las dudas del funcionamiento, asistiendo desde afuera para lograr un mejor uso de la aplicación. La tercera vertiente pasa por contratar los servicios tercerizados de una empresa y la administración del servidor. Luego, un proveedor externo, se encargará de los backups y la limpieza de virus, sin tener que desplazarse.”<sup>8</sup>

---

<sup>8</sup> <http://www.apertura.com/notas/229449-en-largo-camino-la-gestion-remota>

## **1.4 OBJETIVOS DE LA INVESTIGACIÓN**

### **1.4.1 Objetivo general**

Diseñar e implementar el prototipo de herramienta y manejo de comandos para el monitoreo de servidores GNU/LINUX por medio de dispositivos móviles.

### **1.4.2 Objetivos específicos**

- Analizar el monitoreo remoto de un servidor GNU/LINUX.
- Diseñar el software de monitoreo remoto de un servidor GNU/LINUX.
- Diseñar la interfaz de usuario para ser visualizada en los dispositivos móviles a través de WAP.
- Implementar el prototipo funcional para la administración de sistemas GNU/LINUX.
- Realizar pruebas de funcionamiento unitario del prototipo implementado para verificar su correcto funcionamiento

## **1.5 ALCANCES Y LIMITACIONES DEL PROYECTO**

### **1.5.1 Alcances**

El proyecto culmina con el desarrollo e implementación de un prototipo de sistema de monitoreo remoto de un servidor GNU/LINUX por medio de dispositivos móviles que obtendrá valores del estado del sistema y de los servicios que preste, la consulta se realizará a través de WAP y la información será mostrará en paginas WML.

### **1.5.2 Limitaciones**

La interfaz se limita a mostrar los valores actuales del sistema y el estado de sus servicios, pero no se realizará análisis estadístico o comparativo de los valores obtenidos, tampoco permitirá tomar control, realizar cambios en la configuración, corregir valores o estados de los servicios del sistema que se esté monitoreando.

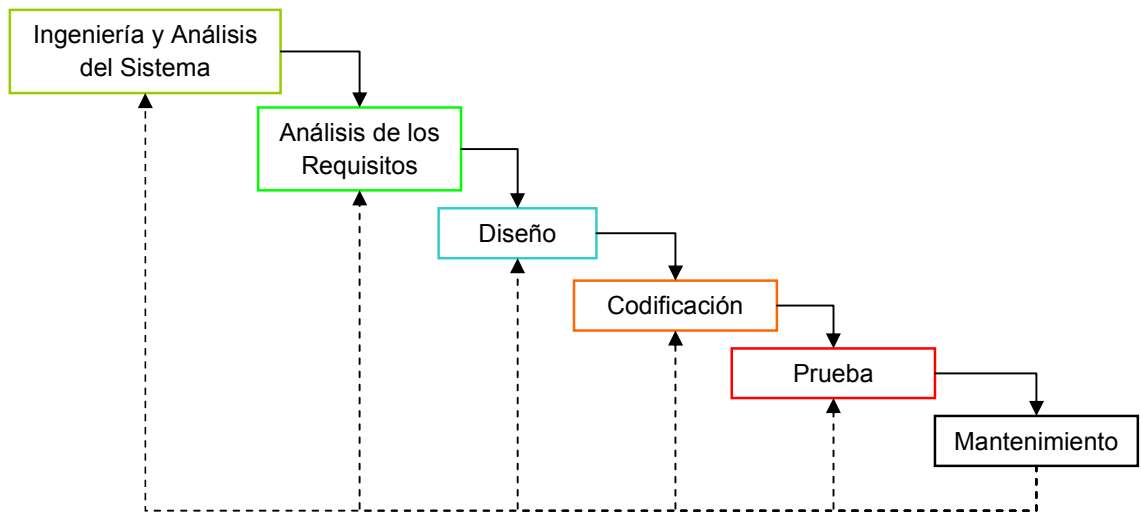
Es importante darse cuenta de que las pruebas unitarias no descubrirán todos los errores del código, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto.

## 2. METODOLOGÍA

El método de desarrollo aplicado es el Modelo Cascada, ver figura 1; basado en el ciclo convencional de desarrollo de software, en el cual se establecen los objetivos globales y después se produce un diseño rápido que se enfoca en la representación de los aspectos visibles al usuario, el diseño rápido conduce a la construcción de un prototipo, el mismo que es evaluado por el usuario y se utiliza para refinar el cumplimiento de los objetivos específicos.

Los flujos fundamentales como modelado de negocio, captura de requisitos, análisis, diseño, implementación, pruebas y despliegue y concluye con una versión más elaborada del producto.

**Figura 1. Esquema de modelo de cascada**



Fuente: Ingeniería del Software: Un enfoque práctico, Roger S. Presuman, 3<sup>ra</sup> Edición, Pág. 26-30.



### **3. LÍNEA DE INVESTIGACIÓN**

#### **3.1 LÍNEA DE INVESTIGACIÓN**

Tecnologías actuales y sociedad.

#### **3.2 LÍNEA DE INVESTIGACION DE FACULTAD**

Tecnologías de Información y Comunicación (TIC).

#### **3.3 NÚCLEO PROBLÉMICO**

Construcción de software.

## 4. MARCO DE REFERENCIA

### 4.1 MARCO TEÓRICO - CONCEPTUAL

#### 4.1.1 Telefonía móvil celular

El sistema de telefonía celular ha pasado por varias generaciones: 1G (Sistema de telefonía celular de primera generación), 2G (Sistema de telefonía celular de segunda generación, conocido generalmente como GSM), 2.5G (Sistema de telefonía móvil de segunda generación avanzado, conocido como GPRS), 3G (Sistema de telefonía celular de tercera generación).

Actualmente en Colombia se encuentran en funcionamiento todas las generaciones de telefonía celular<sup>9</sup>, haciéndose más frecuente el uso de GSM (Global System for Mobile Communication) que permite la transmisión de mensajes cortos de texto (SMS), el acceso a Internet mediante tecnología WAP y GPRS (General Packet Radio Service) y el envío y recepción de datos mediante el protocolo IP. Además permite el envío y recepción de mensajes de multimedia (MMS) y ampliar el número de caracteres de un mensaje corto de texto (SMS).

Un servicio que se hace común para todas las generaciones de telefonía móvil es SMS (Short Message System) que permite el envío y recepción de mensajes cortos de no más de 160 bytes a través de la red telefónica inalámbrica.

Los sistemas móviles han tenido un gran auge y con el avance de la tecnología, tienden a ser más robustos. Éstos se desarrollan en dispositivos móviles como PDA y teléfonos celulares, que requieren de información y diferentes aplicaciones para ofrecer al usuario la posibilidad de realizar muchas de las actividades diarias sin necesidad de moverse de su casa o su sitio de trabajo y que con el tiempo mejorarán y serán más novedosas.

La tecnología se ha centrado en los dispositivos móviles, más exactamente en los teléfonos celulares, sus principios, su crecimiento, auge y características; puesto que la mayoría de las personas pueden acceder a un celular y utilizar sus aplicaciones y servicios de una manera fácil.

---

<sup>9</sup> Esto según estadísticas y registros de 3G Américas sobre operadores GSM/TDMA en América. [http://www.3gamericas.org/Spanish/maps/operators\\_maps/index.cfm](http://www.3gamericas.org/Spanish/maps/operators_maps/index.cfm)

Las comunicaciones móviles aparecieron comercialmente a finales del siglo XX, siendo los países nórdicos los primeros en contar con telefonía móvil<sup>10</sup>, luego se empezó a extender por el mundo, siendo ahora la mejor tecnología implementada y utilizada por las personas en el mundo.

De acuerdo al planteamiento de antecedentes mencionados en el numeral 1.1, los primeros sistemas de telefonía móvil llamados de primera generación TACS, AMPS, NTM, TMA, NAMT2, eran análogos, grandes, y pesados, su costo era elevado y sólo funcionaba en zonas específicas, sus funciones eran limitadas. Estos iniciaron en 1979 en Tokio donde estuvo la primera red de telefonía móvil.

Después aparecieron los llamados de segunda generación GSM, CDMA, TDMA, NADC, PDC, estos son digitales, transmiten datos a velocidades pequeñas y utilizan el envío de mensajes SMS, que iniciaron aproximadamente en 1982, y se implementaron en 1992 cuando en Finlandia estuvo la primera red GSM.

En la actualidad se está implementando una sistema llamado de generación 2.5, HSCSC, GPRS, EDGE, que busca hacer puente con la segunda y tercera generación UMTS, para estandarizar la comunicación a nivel mundial; aunque todavía pueden pasar algunos años; ofreciendo una comunicación más rápida a grandes velocidades, de una manera fácil y se podrá transmitir voz, imagen, video, datos. Sin olvidarnos de nuestro país que en 1994 tuvo sus primeros sistemas de telefonía móvil IS-136, Colombia cuenta con tres tecnologías para comunicaciones móviles, que son: IS-136, GSM/GPRS, CDMA200 1X, la IS136 reemplazada por los de la 2.5 generación y esta última por la tecnología 3G.

Al igual que GPRS, la tecnología 3G (tecnología inalámbrica de tercera generación) es un servicio de comunicaciones inalámbricas que le permite estar conectado permanentemente a Internet a través del teléfono móvil, el ordenador de bolsillo, el Tablet PC o el ordenador portátil. La tecnología 3G promete una mejor calidad y fiabilidad, una mayor velocidad de transmisión de datos y un ancho de banda superior (que incluye la posibilidad de ejecutar aplicaciones multimedia). Con velocidades de datos de hasta 384 Kbps, es casi siete veces más rápida que una conexión telefónica estándar.<sup>11</sup>

Con el pasar de los años los dispositivos móviles han dejado de ser sólo un medio de transmisión de voz para convertirse en un sistema que integra

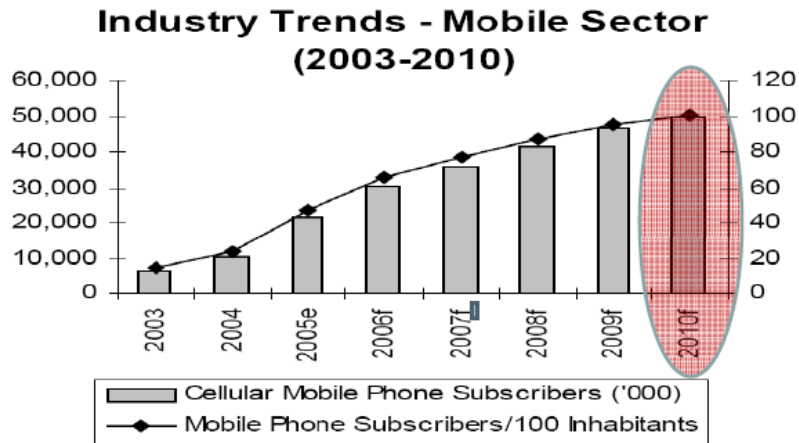
---

10 Mas exactamente en Escandinavia donde se implemento el primer sistema de telefonía celular basado en AMPS (Advanced Mobile Phone System). [http://es.wikipedia.org/wiki/Historia\\_del\\_tel%C3%A9fono\\_m%C3%B3vil](http://es.wikipedia.org/wiki/Historia_del_tel%C3%A9fono_m%C3%B3vil)

11 <http://www.mundotelme.com/tecnologia3g.php>

otros servicios como transmisión y procesamiento de información. La evolución de la telefonía móvil en Colombia entre 2003 y 2010 se puede visualizar en la figura 2, en la cual se evidencia el crecimiento paulatino y prolongado de suscriptores de telefonía móvil en Colombia.

**Figura 2. Evolución de la tecnología móvil en Colombia (2003-2010)**



Fuente: La Telefonía Móvil Celular en Colombia 1998 Documento Sectorial - Departamento Nacional de Planeación Unidad de Infraestructura y Energía.

Los dispositivos móviles ofrecen servicios diferentes y cada vez más útiles algunos se detallan en la Figura 3.

**Figura 3. Analogías de servicios ofrecidos en los móviles.**

Mundo PC	Mundo Móvil	Cualidades Únicas (Móvil)
WEB	WAP (xHTML) / WEB	Ubicación
Email	SMS / MMS / Email Móvil	Inmediatez
Apps de Escritorio sobre plataformas (Microsoft ó Linux/Unix)	Apps Móviles sobre plataformas (iPhone, Android, Symbian, PalmOS, Java, Flash)	Ubicación, Contexto.

Fuente: Documento “La Telefonía Móvil Celular en Colombia 1998 Documento Sectorial” - Departamento Nacional de Planeación Unidad de Infraestructura y Energía.

Respecto a los servicios del mundo móvil, WAP será indispensable en el desarrollo del proyecto, la tecnología WAP es un estándar impulsado por la industria del sector de las telecomunicaciones con el objetivo de proporcionar un sistema avanzado de servicios de internet para dispositivos

móviles.<sup>12</sup>

La gestión de datos para dispositivos móviles, requiere una arquitectura que permita el modelado e implementación de datos, dicha arquitectura se detalla en el numeral 4.1.2.

En Colombia el auge de las tecnologías inalámbricas, móviles y PCS impulsó un mercado de aplicaciones software para estos dispositivos, donde se encuentran dos alternativas:

- Aplicaciones basadas en navegación: permite aplicaciones WAP basadas en el modelo cliente-servidor, se limita a la capacidad de la red móvil.
- Aplicaciones móviles locales: permite aplicaciones que corran en el celular mediante J2ME, esta se limita a la capacidad del celular.

Los dispositivos móviles abren la puerta a un nuevo mundo de posibilidades para las comunicaciones inalámbricas. La tecnología WAP establece un puente mucho más sólido entre el mundo de la telefonía móvil e Internet ofreciendo la capacidad para vehicular un amplio abanico de servicio de valor añadido con total independencia de la tecnología que se utilice.

La arquitectura WAP fue seleccionada gracias a sus múltiples ventajas, entre las cuales se pueden detallar:

- WAP es un estándar global que no está controlado por ninguna compañía, lo que asegura su democracia, su apertura y su universalidad
- Aunque lo más conocido del WAP es la integración entre la red y el móvil, conviene dejar claro, que es capaz de funcionar sobre cualquier dispositivo que disponga de conexión inalámbrica.
- WAP, no solo ha sido ideado para transmitir contenidos desde Internet, sino que cualquier empresa puede disponer de un servidor de este tipo para ofrecer aquellos servicios y contenidos que le parezcan sin que por ello tengan que guardar ninguna relación con Internet. No obstante, cabe mencionar, que el mayor crecimiento de esta tecnología se debe a su interrelación con la red.
- Las especificaciones de WAP definen un conjunto de protocolos que afectan el funcionamiento de las aplicaciones, las sesiones de conexión, las transacciones, la seguridad y los niveles de transporte,

---

<sup>12</sup> <http://www.elcodigo.net/tutoriales/wap/wap1.html>

permitiendo a los operadores, fabricantes y desarrolladores de aplicaciones hacer frente a los requerimientos de flexibilidad y diferenciación.

- Independencia sobre los estándares para la creación de redes de telefonía.
- Completamente abierto y escalable.
- Independiente del sistema de transporte (GSM, IS-136, DECT, TDMA, etc.).
- Independiente del tipo de terminal (teléfono celular, PDA, etc.).
- Adaptable a nuevas tecnologías de transporte (GPRS, UMTS).
- De cara al usuario final, el WAP aporta un acceso fácil y seguro a la información y a los servicios más relevantes de Internet

#### **4.1.2 WAP - Protocolo de aplicaciones inalámbricas**

WAP (Wireless Application Protocol o Protocolo de Aplicaciones Inalámbricas) es una solución unificada para los servicios de valor agregado existentes y futuros para la telefonía móvil. El protocolo incluye especificaciones para las capas de sesión y de transporte del modelo OSI, así como funcionalidades de seguridad. WAP también define un entorno de aplicaciones<sup>13</sup>.

Es escalable, permitiendo así a las aplicaciones disponer de las capacidades de pantalla y recursos de red según su necesidad y en una gran variedad de tipos de terminales. Los servicios podrán ser aplicables a pantallas de una sola línea o a terminales mucho más complejos como las PDA's. Como cualquier estándar, las ventajas son múltiples a la hora de desarrollar aplicaciones, fabricar terminales o estructurar la red.

##### **4.1.2.1 Componentes de la arquitectura WAP**

Una vez introducido el sistema, vamos a ver la arquitectura que le da

---

<sup>13</sup> <http://www.movired.com/wap-wap/wap.htm>

consistencia. La arquitectura WAP ver figura 4, está pensada para proporcionar un "entorno escalable y extensible para el desarrollo de aplicaciones para dispositivos de comunicación móvil". Para ello, se define una estructura en capas, en la cual cada capa es accesible por la capa superior así como por otros servicios y aplicaciones a través de un conjunto de interfaces muy bien definidos y especificados.

**Figura 4. Arquitectura WAP**



Fuente: <http://www.movired.com/wap-wap/wap.htm>

#### 4.1.2.2 Capas de la arquitectura WAP

A continuación se procederá a describir cada una de las capas:

##### Capa de Aplicación (WAE)

Es un entorno de aplicación de propósito general, basado en la combinación del World Wide Web y tecnologías de Comunicaciones Móviles. Este entorno incluye un micro navegador, que posee las siguientes funcionalidades:

- El lenguaje WML, el cual se detallará en el numeral 4.1.3.
- El lenguaje WMLS, similar al JavaScript.

WTA (Wireless Telephony Applications) Es un entorno para aplicaciones u servicios de telefonía.

WTAI (Wireless Telephony Application Interface) Es una interfaz utilizada en los terminales móviles para operaciones locales de control de llamadas (recepción, iniciación y terminación) y acceso a listines telefónicos.

### **Capa de Sesión (WSP)**

Este protocolo proporciona a la Capa de Aplicación (WAE) interfaz con dos servicios de sesión:

- Un servicio orientado a conexión que funciona por encima de la Capa de Transacciones (WTP).
- Un servicio no orientado a conexión que funciona por encima de la Capa de Transporte (WTP), y que proporciona servicio de datagramas seguro o no seguro.

Esta capa proporciona las siguientes funcionalidades:

- Establecimiento y liberación de conexiones entre cliente y servidor.
- Intercambio de información entre cliente y servidor.
- Negociación de las características del protocolo.
- Suspensión y reanudación de la sesión.

### **Capa de Transacciones (WTP)**

Este protocolo funciona por encima de un servicio de datagramas ya sean seguros como no seguros, y proporciona las siguientes funcionalidades:

- Proporciona los servicios necesarios para soportar las transacciones, estos servicios pueden ser de tres clases:
  - Peticiones inseguras de un solo camino
  - Peticiones seguras de un solo camino
  - Transacciones seguras de dos caminos
- También proporciona seguridad en las transacciones.

### **Capa de Seguridad (WTLS)**



La Capa Inalámbrica de Seguridad de Transporte (WTLS) es un protocolo basado en el estándar SSL, utilizado en el entorno Web, para la seguridad en la transferencia de datos. Esta capa proporciona a las capas de nivel superior de WAP una interfaz de servicio de transporte seguro, que lo resguarde de una interfaz de transporte inferior.

Las funcionalidades de esta capa son las siguientes:

- Integridad de los datos: se asegura que la información intercambiada entre el terminal y el servidor de aplicaciones, no haya sido modificada.
- Privacidad de los datos: se asegura que la información intercambiada entre el terminal y el servidor de aplicaciones, no pueda ser captada ni entendida por elementos externos a la comunicación.
- Autenticación: se ofrecen servicios para determinar la autenticidad del terminal y del servidor de aplicaciones.

También puede ser utilizado para el establecimiento de una comunicación segura entre terminales.

### **Capa de Transporte (WDP)**

El Protocolo Inalámbrico de Datagramas (WDP) proporciona las siguientes funcionalidades:

- Proporciona un servicio fiable a los protocolos de las capas superiores de WAP.
- Permite la comunicación de forma transparente sobre los protocolos portadores: CDMA, SMS, GSM...

#### **4.1.2.3 Funcionamiento de WAP**

Para conseguir consistencia en la comunicación entre el terminal móvil y los servidores de red que proporcionan la información, WAP define un conjunto de componentes estándar<sup>14</sup> Ver figura 5:

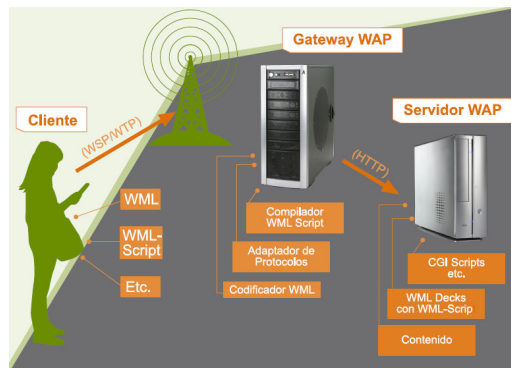
- Un modelo de nombres estándar. Se utilizan las URIs (Uniform Resource Identifier) definidas en WWW para identificar los recursos locales del dispositivo y las URLs (Uniform Resource Location también definidas en el WWW) para identificar el contenido WAP en los servidores de información.

---

<sup>14</sup> <http://neutron.ing.ucv.ve/revista-e/No8/EATimaure%5CWap.htm>

- Un formato de contenido estándar, basado en la tecnología WWW.
- Unos protocolos de comunicación estándares, que permitan la comunicación del micro navegador del terminal móvil con el servidor Web en red.

**Figura 5. Funcionamiento de WAP**



Fuente: [www.infoteca.com.uy/manuales/monografia\\_wap.doc](http://www.infoteca.com.uy/manuales/monografia_wap.doc)

1. El usuario solicita la página WAP que quiera ver.
2. El micronavegador del móvil envía la petición con la dirección (URL) de la página solicitada y la información sobre el abonado al Gateway WAP (software capaz de conectarse a la red de telefonía móvil y a Internet)
3. El Gateway examina la petición y la envía al servidor donde se encuentra la información solicitada.
4. El servidor añade la información http o HTTPS pertinente y envía la información de vuelta al Gateway.
5. En el Gateway se examina la respuesta del servidor, se valida el código WML en busca de errores y se genera la respuesta que se envía al móvil.
6. El micronavegador examina la información recibida y si el código es correcto lo muestra en pantalla

Los protocolos WAP son una especificación abierta, regida por un consorcio (similar al W3C en HTML) de partes interesadas: el Fórum WAP. Los protocolos WAP son análogos a los existentes en la tecnología utilizada en Internet, solo que optimizados para la "potencia" que ofrecen los

dispositivos portátiles, y limitados por el ancho de banda que ofrece la tecnología inalámbrica y cubren tanto la capa de aplicación (WML y WMLScript, conocidos como capa de aplicación), y las capas de transporte (WSP, WTP, WTLS y WDP).

La figura 6 muestra la pila de protocolos de WAP, así como su significado y equivalencia con los ya habituales niveles de TCP/IP.

**Figura 6: Estructura WAP vs WEB**

TCP/IP		WAP
HTML RFC-822	Nivel de Aplicación	WAE
HTTP FTP SMTP ...	Nivel de Sesión	WSP
TCP UDP	Nivel de Transacción	WTP
SSL TLS	Nivel de Seguridad	WTLS
IP	Nivel de Transporte	WDP
ETH ATM ...	Nivel de Red	GSM SMS CDMA

Fuente: [www.rediris.es/.../boletin/54-55/ponencia10.html](http://www.rediris.es/.../boletin/54-55/ponencia10.html)

Como puede observarse, se ha contemplado desde el primer momento la inclusión de soporte para conexiones seguras, de forma similar al papel de SSL en las comunicaciones actuales a servidores web seguros.

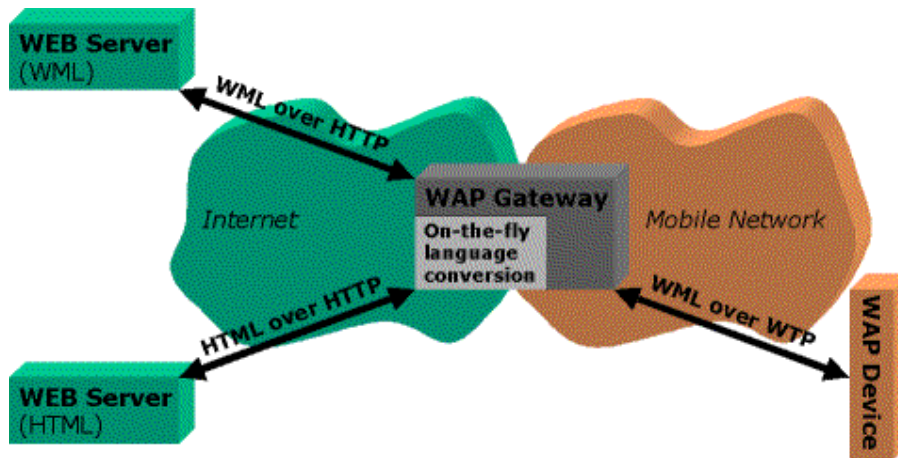
Internet se compone de miles de servidores que almacenan las paginas que vemos (o las aplicaciones que generan dichas paginas) a los que acceden los clientes (los navegadores, generalmente).

El sistema seguido por WAP no es muy diferente. Ahora los clientes son los dispositivos móviles y los servidores deben seguir almacenando las páginas y aplicaciones.

La única consideración que hay que hacer, es que se debe que conectar la red inalámbrica (telefonía móvil) a Internet, de forma que el dispositivo portátil pueda hacer una petición de página WML al servidor.

Para conectar ambas redes las especificaciones WAP asumen que habrá un gateway WAP que convierte las peticiones WAP a peticiones WEB, y las respuestas WEB a respuestas WAP, como vemos en la siguiente figura:

Figura 7: Funcionamiento en Internet de WAP y WEB



Fuente: <http://geneura.ugr.es/~maribel/wap/introduccion/index.shtml>

En teoría, el gateway debe poder convertir páginas HTML, sobre la marcha, en páginas WML.

Sin embargo, puesto que WML está pensado y diseñado para presentar información en pantallas muy pequeñas, y no es tan potente, visualmente, como HTML, esa traducción de información HTML a WML puede ser complicada de hacer (mas aun si hay scripts, animaciones, imágenes grandes, etc.).

Seguridad es un concepto que se usa a menudo con escaso rigor, cuando hablamos de seguridad en tecnologías de la información, estamos hablando de muchas cosas a la vez: que nadie nos robe o modifique los datos, que nadie nos suplante, que nadie acceda a donde no debe.

Dentro del modelo de referencia OSI<sup>15</sup> se define una arquitectura de seguridad dentro de la que existe una serie de servicios de seguridad. Según esta especificación, para proteger las comunicaciones es necesario dotar a las mismas de dichos servicios. Se trata de los siguientes:

- Autenticación de la entidad par: mediante este servicio se verifica la fuente de los datos. La autenticación puede ser de la entidad origen, de la entidad destino o de ambas a la vez.
- Control de acceso: este servicio verifica que los recursos son utilizados

---

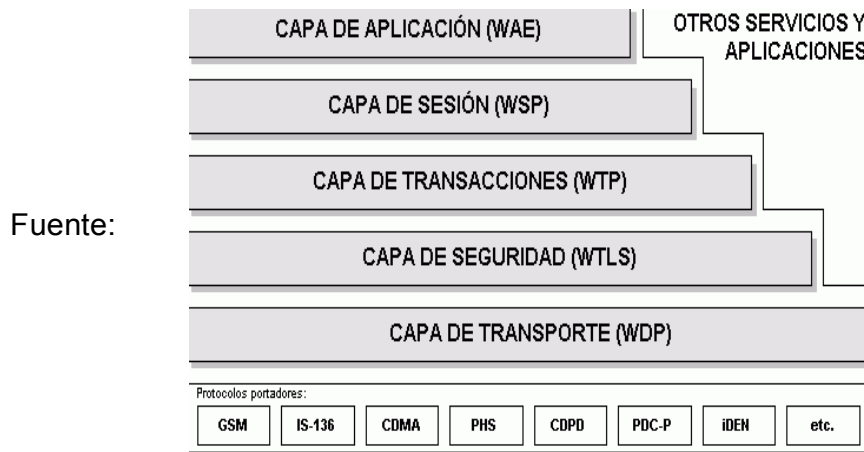
<sup>15</sup> [http://www.comsoc.org/livepubs/50\\_journals/pdf/RightsManagement\\_eid=136833.pdf](http://www.comsoc.org/livepubs/50_journals/pdf/RightsManagement_eid=136833.pdf)

sólo por quien tiene derecho a hacerlo.

- Confidencialidad de los datos: con este servicio se evita que se revelen, deliberada o accidentalmente, los datos de una comunicación.
- Integridad de los datos: este servicio verifica que los datos de una comunicación no se alteren, esto es, que los datos recibidos por el receptor coincidan por los enviados por el emisor.
- Proporciona prueba de que cada una de las entidades ha participado, efectivamente, en la comunicación. Puede ser de dos tipos:
  - Con prueba de origen o emisor: el destinatario tiene garantía de quien es el emisor concreto de los datos.
  - Con prueba de entrega o receptor: el emisor tiene prueba de que los datos de la comunicación han llegado íntegramente al destinatario correcto en un instante dado.

La pila de protocolos WAP es más ligera que la habitual HTTP/TCP/IP y su relación con ésta, viene representada en la Figura 8. La pasarela WAP, es la encargada de comunicarse con el móvil a través del protocolo WAP, y de convertirlo a protocolo de Internet para enviárselo al servidor Web.

**Figura 8: Pila de protocolo WAP**



<http://www.ordenadores-y-portatiles.com/wml.html>

#### 4.1.3 WML (Wireless Markup Language)

El lenguaje WML (Wireless Markup Language) constituye la base para la

creación de contenidos visualizables desde un terminal WAP (un terminal dotado con un micro-navegador WAP).

Las páginas WML sólo permiten introducir texto y gráficos, debido a las propias capacidades de los terminales móviles. Por otra parte, los micro-navegadores de los terminales también soportan un lenguaje de script, llamado WMLScript, que permite manipular datos y dotar de cierta interactividad a la página WML.

Por otra parte, los gráficos que pueden ser introducidos en una página WML deberán tener un formato especial, denominado WBMP. Este formato sólo permite imágenes en blanco y negro.

Las páginas WML podrán ser ubicadas en cualquier servidor web convencional, siempre que éste haya sido correctamente configurado para servir<sup>16</sup>.

#### **4.1.3.1 Características de WML**

Las características del lenguaje WML pueden agruparse en cuatro áreas principales:

- Ofrece un soporte de texto e imagen y tiene una amplia variedad de formatos y comandos.

- Las cartas (fragmentos de página o pantallas) WML se agrupan en barajas. Una baraja WML es similar a una página HTML identificada por una URL (Universal Resource Location, localización universal de recursos) y es la unidad básica de transmisión de contenidos.

- Ofrece soporte para gestión de navegación entre cartas y barajas, e incluye comandos para su manejo. Estos pueden usarse para navegar o ejecutar scripts, WML también provee de conexiones de anclaje similares a las usadas en el HTML versión 4.

- Se pueden establecer parámetros para todas las barajas de WML usando un modelo establecido. Se pueden usar variables en lugar de cadenas y sustituirse en el tiempo de ejecución. Esta forma de establecer parámetros permite que los recursos de la red sean usados de forma eficiente.

Toda la información de WML se transmite en formato codificado por la red inalámbrica.

---

<sup>16</sup> [www.elcodigo.net/tutoriales/wap/wap2.html](http://www.elcodigo.net/tutoriales/wap/wap2.html)

#### **4.1.4 UML (Unified Modeling Language)**

El lenguaje unificado de modelado (UML), es un lenguaje no propietario para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo.

UML proporciona una forma sencilla de definir y transmitir ideas complejas; por medio de diagramas e iconografías.

UML es manejado y controlado por un consorcio (el Object Management Group OMG), el cual está conformado principalmente por grandes compañías de la industria de la computación. A través de este consorcio se controla la forma en cómo UML evoluciona mediante el consenso de la junta de directores, con la finalidad de crear y mantener una especificación única.

La aplicación de UML, es muy extensa, ya que puede ser utilizado por personas con o sin conocimientos técnicos, para realizar modelos físicos y conceptuales de diversa índole.

En el Diseño de software, es una de las áreas en que el UML tiene una mayor aceptación, debido a que a través de él, se puede pasar de un modelo conceptual de un software en particular, al detalle de implementación y finalmente realizar el modelo de despliegue físico de una aplicación.

El lenguaje UML se compone de tres elementos básicos, los bloques de construcción, las reglas y algunos mecanismos comunes. Estos elementos interaccionan entre sí para dar a UML el carácter de complejidad y no ambigüedad.

Los bloques de construcción se dividen en tres partes:

- Elementos: Son las abstracciones de primer nivel.
- Relaciones: Unen a los elementos entre sí.
- Diagrama: Son agrupaciones interesantes de elementos.

Se utilizan diferentes diagramas dependiendo de qué interés representar en cada momento, para dar diferentes perspectivas de un mismo problema, para ajustar el nivel de detalle, por esto UML soporta un gran número de diagramas diferentes aunque, sólo se utilicen un pequeño número de combinaciones.

UML proporciona un conjunto de reglas que dictan las pautas a la hora de realizar asociaciones entre objetos para poder obtener modelos bien formados, estas son reglas semánticas. UML proporciona una serie de mecanismos comunes que sirve para que cada persona o entidad adapte el lenguaje a sus necesidades, pero dentro de un marco ordenado y siguiendo unas ciertas reglas para que el trasfondo de la adaptación no se pierda la semántica propia de UML.

#### **4.1.4.1 Diagramas**

Los diagramas se utilizan para representar diferentes perspectivas de un sistema de forma que un diagrama es una proyección del mismo. UML proporciona un amplio conjunto de diagramas que normalmente se usan en pequeños subconjuntos para poder representar las cinco vistas principales de la arquitectura de un sistema.

**Diagrama de Clases:** muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos.

**Diagramas de Objetos:** muestra un conjunto de objetos y sus relaciones, son como fotos instantáneas de los diagramas de clases.

**Diagramas de casos de usos:** muestran un conjunto de casos de usos y actores y sus relaciones. Son especialmente importantes para el modelado y organización del comportamiento.

**Diagramas de Secuencia y de Colaboración:** son un tipo de diagramas de interacción. Constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar unos objetos a otros. Los diagramas de secuencia enfatizan el ordenamiento temporal de los mensajes mientras que los diagramas de colaboración muestran la organización estructural de los objetos que envían y reciben mensajes.

**Diagramas de Componentes:** muestra la organización y las dependencias entre un conjunto de componentes, se relacionan con los diagramas de clases ya que en un componente suele tener una o más clases, interfaces o colaboraciones.

**Diagramas de Despliegue:** representan la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.



#### **4.1.5 HSQLDB (Hyperthreaded Structured Query Language Database)**

El HSQLDB (del inglés Hyperthreaded Structured Query Language Database) es un servidor de banco de datos (SGBD), de código abierto, escrito totalmente en el lenguaje Java. No es posible compararlo, en términos de robustez y seguridad con otros servidores SGBD, como Oracle o Microsoft SQL Server, sin embargo el HSQLDB es una solución simple, que utiliza pocos recursos y que posee buen desempeño. Debido a esas características, él es bastante utilizado en aplicaciones que son ejecutadas en desktops y que necesitan integrar con una capa de persistencia a través del lenguaje SQL.

El HSQLDB se ve aplicado dentro del proyecto, en la base de datos elaborada para administrar el almacenamiento y control de los usuarios y equipos a monitorear.

##### **4.1.5.1 Características HSQLDB**

Las principales características de son:

- Soporte al lenguaje SQL básica, incluyendo triggers y visiones.
- Portabilidad en virtud de su implementación ser hecha en Java.
- Repositorios accesibles a través de tecnología JDBC.
- Creación de bancos de datos en archivo texto, banco de datos y en memoria.
- Recurso de dump para backups facilitados.
- Ocupa poco espacio en disco.
- Prácticamente dispensa configuraciones para operar.

#### **4.1.6 RMI (Remote Method Invocation)**

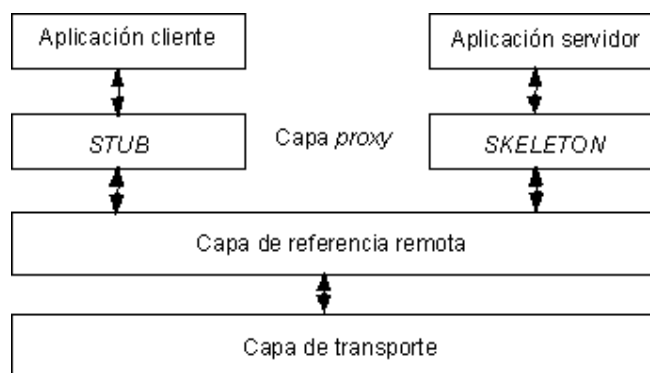
Las aplicaciones RMI normalmente comprenden dos programas separados: un servidor y un cliente. Una aplicación servidor típica crea un montón de objetos remotos, hace accesibles unas referencias a dichos objetos remotos, y espera a que los clientes llamen a estos métodos u objetos remotos. Una aplicación cliente típica obtiene una referencia remota de uno o más objetos remotos en el servidor y llama a sus métodos. RMI proporciona el mecanismo por el que se comunican y se pasan información

del cliente al servidor y viceversa. Cuando es una aplicación, algunas veces se hace referencia a ella como Aplicación de Objetos Distribuidos. Las aplicaciones de objetos distribuidos necesitan:

- Localizar Objetos Remotos: Las aplicaciones pueden utilizar uno de los dos mecanismos para obtener referencias a objetos remotos. Puede registrar sus objetos remotos con la facilidad de nombrado de RMI `rmiregistry`. O puede pasar y devolver referencias de objetos remotos como parte de su operación normal.
- Comunicar con Objetos Remotos: Los detalles de la comunicación entre objetos remotos son manejados por el RMI; para el programador, la comunicación remota se parecerá a una llamada estándar a un método Java.
- Cargar Bytecodes para objetos que son enviados: Como RMI permite al llamador pasar objetos Java a objetos remotos, RMI proporciona el mecanismo necesario para cargar el código del objeto, así como la transmisión de sus datos.

La siguiente ilustración muestra una aplicación RMI distribuida que utiliza el registro para obtener referencias a objetos remotos. El servidor llama al registro para asociar un nombre con un objeto remoto. El cliente busca el objeto remoto por su nombre en el registro del servidor y luego llama a un método. Esta ilustración también muestra que el sistema RMI utiliza un servidor Web existente para cargar los bytecodes de la clase Java, desde el servidor al cliente y desde el cliente al servidor, para los objetos que necesita.

**Figura 9: Arquitectura Java RMI**



Fuente: <http://www.manualespdf.es/manual-java-rmi/arquitecturaRMI.gif>

El sistema RMI utiliza un servidor Web para cargar los bytecodes de la clase Java, desde el servidor al cliente y desde el cliente al servidor.

#### **4.1.6.1 Ventajas de la carga dinámica de código**

Una de las principales y únicas características de RMI es la habilidad de descargar los bytecodes (o simplemente, código) de una clase de un objeto si la clase no está definida en la máquina virtual del receptor. Los tipos y comportamientos de un objeto, anteriormente sólo disponibles en una sola máquina virtual, ahora pueden ser transmitidos a otra máquina virtual, posiblemente remota. RMI pasa los objetos por su tipo verdadero, por eso el comportamiento de dichos objetos no cambia cuando son enviados a otra máquina virtual. Esto permite que los nuevos tipos sean introducidos en máquinas virtuales remotas, y así extender el comportamiento de una aplicación dinámicamente.

#### **4.1.6.2 Interfaces, objetos y métodos remotos**

Una aplicación distribuida construida utilizando RMI de Java, al igual que otras aplicaciones Java, está compuesta por interfaces y clases. Los interfaces definen métodos, mientras que las clases implementan los métodos definidos en los interfaces y, quizás, también definen algunos métodos adicionales. En una aplicación distribuida, se asume que algunas implementaciones residen en diferentes máquinas virtuales. Los objetos que tienen métodos que pueden llamarse por distintas máquinas virtuales son los objetos remotos.

Un objeto se convierte en remoto implementando un interface remoto, que tenga estas características.

- Un interface remoto hereda del interface `java.rmi.Remote`.
- Cada método del interface declara que lanza una `java.rmi.RemoteException` además de cualquier excepción específica de la aplicación.

El RMI trata a un objeto remoto de forma diferente a como lo hace con los objetos no-remotos cuando el objeto es pasado desde una máquina virtual a otra. En vez de hacer una copia de la implementación del objeto en la máquina virtual que lo recibe, RMI pasa un stub para un objeto remoto. El stub actúa como la representación local o proxy del objeto remoto y básicamente, para el llamador, es la referencia remota. El llamador invoca un método en el stub local que es responsable de llevar a cabo la llamada al objeto remoto. Un stub para un objeto remoto implementa el mismo conjunto de interfaces remotos que el objeto remoto. Esto permite que el stub sea asignado a cualquiera de los interfaces que el

objeto remoto implementa. Sin embargo, esto también significa que sólo aquellos métodos definidos en un interface remoto están disponibles para ser llamados en la máquina virtual que lo recibe.

#### **4.1.7 Servidores LINUX**

Linux ha demostrado a lo largo del tiempo ser un Sistema Operativo fiable, seguro y de gran calidad para su uso en servidores. Los Servidores Linux se gestionan mediante SSH, VNC y/o Panel de Control; mientras que los Servidores Windows se gestionan mediante Terminal Server, VNC y/o Panel de Control.

En cuanto a funciones los Servidores Linux no se llevan muy bien con ASP (Active Server Pages), y no pueden ejecutar MS SQL Server o MS Exchange. En cuanto a las demás aplicaciones/servicios, tanto Linux como Windows pueden realizar las mismas funciones, siendo Linux, normalmente, el preferido para ejecutar sistemas basados en PHP/MySQL.

Los servidores dedicados Linux serán siempre más económicos que los Windows. La principal razón de ello es que existe una gran comunidad OpenSource (código abierto) y aplicaciones gratuitas. Linux, normalmente, no requiere de licencias del proveedor. En el caso de Windows, si queremos ejecutar un MS SQL Server o Exchange, deberemos tener en cuenta que supone un costo adicional.

La shell, es el intérprete de comandos de la mayoría de los sistemas Unix o similares a Unix. Un script es un guión, es decir una serie de instrucciones para la consola guardado en un fichero de texto de extensión: .sh. En Gnu/Linux, esta técnica alcanza el nivel de un lenguaje de programación. Para los administradores de Servidores, los scripts son herramientas muy útiles. Permiten hacer tareas de una forma muy productiva, ejecutar aplicaciones cuando se desee, mediante el demonio crontab, etc. Incluso el sistema operativo se inicia gracias a un script (initd).

##### **4.1.7.1 Monitorización sobre servidores LINUX**

Los sistemas compatibles UNIX SV utilizan los comandos sar y sadc para obtener estadísticas del sistema. Su equivalente en GNU/Linux (incluido en Debian) es atsar (y atsadc), que es totalmente equivalente a los que hemos mencionado. El comando atsar lee contadores y estadísticas del fichero /proc y las muestra por la salida estándar. La primera forma de llamar al comando es: atsar opciones t [n]

Donde muestra la actividad en n veces cada t segundos con una cabecera

mostrando los contadores de actividad (el valor por defecto de n es 1).

La segunda forma de llamarlo es: `atsar -opciones -s time -e time -i sec -f file -n day#`

El comando extrae datos del archivo especificado por `-f` (por defecto `/var/log/atsar/atsarxx`, con `xx` el día del mes) y que fueron previamente guardados por `atsadc` (se utiliza para recoger los datos, salvarlos y procesarlos y en Debian está en `/usr/lib/atsar`). El parámetro `-n` puede ser utilizado para indicar el día del mes y `-s`, `-e` la hora de inicio-final, respectivamente. Para activar `atsadc`, por ejemplo, se podría incluir en `/etc/cron.d/atsar` una línea como la siguiente:

La 1.<sup>a</sup> crea el archivo después de un reinicio. La 2.<sup>a</sup> guarda los datos cada 10 minutos con el shell script `atsa1`, que llama al `atsadc`.

En `atsar`, las opciones se utilizan para indicar qué contadores hay que mostrar; algunos de ellos se describen a continuación ver figura 10:

**Figura 10 Opciones en atsar**

Opción	Descripción
u	Utilización CPU
d	Actividad de disco
l	Número de interrupciones/s
v	Utilización de tablas en el kernel
y	Estadísticas de utilización de <code>mys</code>
p	Información de paginación y actividad de swap
r	Memoria libre y ocupación de swap
l	Estadísticas de red
L	Información de errores de red
w	Estadísticas de conexiones IP
t	Estadísticas de TCP
U	Estadísticas de UDP
m	Estadísticas de ICMP
N	Estadísticas de NFS
A	Todas las opciones

Fuente: Administración avanzada de GNU/Linux Joseph Jorba Marzo 2004.

#### 4.1.7.1.1 Utilización de CPU

- `usr` y `sys` muestran el porcentaje de tiempo de CPU en el modo usuario

con nice  $\neq 0$  (normales) y en el modo kernel.

- nice, lo mismo, pero con procesos de usuario con nice  $> 0$  (menor prioridad que la media).
- idle indica el tiempo no utilizado de CPU por los procesos en estado de espera.
- pswch/s es el número de cambios de contextos.
- runq da el número de procesos que espera la CPU.
- nrproc, el número de procesos en el sistema.
- lavg1,5,15 indica la carga media en la cola de ejecución en el último, en los últimos 5 y en los últimos 15 minutos.

#### **4.1.7.1.2 Memoria y swap**

En este caso memtot es la memoria principal (MP) total; memfree, la libre; memshared, la compartida; buffers es la MP utilizada en buffers; cached, memoria principal utilizada en caché de páginas; swptot, el espacio total de swap, y swpfree el espacio libre de swap (donde M = 1.000 Kbytes). Es importante tener en cuenta que si no hay espacio en MP, las páginas de los procesos irán a parar al swap, donde debe haber sitio. Esto se debe contrastar con la utilización de CPU. También se debe controlar que el tamaño de los buffers sea adecuado y esté en relación con los procesos que están realizando operaciones de entrada/salida<sup>17</sup>.

#### **4.1.8 Estándar 830-1988 IEEE (Institute Of Electrical and Electronics Engineers)**

IEEE 830: IEEE Recommended Practice for Software Requirements Specifications traduce Práctica recomendada para la especificación de requisitos de software, su naturaleza es una especificación técnica internacional de un consorcio industrial, el ámbito se detalla como desarrollo de sistemas de información - Procesos del ciclo de vida del software - Especificación de requisitos.

Es un enfoque recomendado de cómo especificar los requisitos del software, con el fin de conseguir un documento completo y sin ambigüedades que ayude a) a los clientes o compradores a describir con precisión lo que quieren

---

<sup>17</sup> Administración avanzada de GNU Linux – UOC Formación de Postgrado

obtener; b) a los suministradores a comprender exactamente lo que el cliente quiere; c) a los individuos al cargo de desarrollar una especificación de requisitos del software (ERS) normalizada para su organización, de definir el formato y contenido de esas especificaciones, o bien de comprobar su calidad.

Los beneficios de una buena ERS son evidentes:

- 1.- Servir como base para el acuerdo entre cliente y proveedor sobre lo que el software ha de hacer.
- 2.- Reducir el esfuerzo de desarrollo
- 3.- Proporcionar la base para la estimación de costes y plazos
- 4.- Proporcionar el punto de partida para las actividades de validación y verificación
- 5.- Facilitar la transferencia del software, a nuevos usuarios o nuevas máquinas
- 6.- Servir de base para ampliaciones o mejoras

La norma está orientada fundamentalmente a la especificación de requisitos para software cuyo desarrollo se va a contratar, pero puede aplicarse también como ayuda en la selección de productos software comercial o desarrollado internamente.

Cuando el software está integrado en sistemas más amplios, como por ejemplo equipos de medicina, es probable que haya que tratar aspectos adicionales a los contemplados por esta norma.

Esta "práctica recomendada" describe el proceso de creación de un producto (la especificación de requisitos del software o ERS) y el contenido del producto en sí. Puede utilizarse directamente o como modelo para una norma más específica, este estándar está compuesto por las siguientes partes:

1. Introducción y alcance
2. Referencias
3. Definiciones
4. Consideraciones en la producción de una ERS
5. Partes de una ERS

6. Plantillas/modelos.<sup>18</sup>

## 4.2 MARCO LEGAL

La implementación de la herramienta de monitoreo se entiende como un servicio de valor agregado, dirigido a un grupo cerrado de usuarios, puesto que para su correcto funcionamiento esta debe ser usada a través de una red de telecomunicaciones.

Los decretos en Colombia que regulan el uso de la aplicación son:

- Ministerio de Comunicaciones Decreto número 600 (14 mar. 2003) por medio del cual se expiden normas sobre los servicios de valor agregado y Telemáticos, y se reglamenta el Decreto Ley 1900 de 1990.
- Decreto número 3055 de 2003 de octubre de 2003 por medio del cual se modifica el decreto 600 de 2003.
- Decreto número 1360 de 1989. Por el cual se reglamenta la inscripción del soporte lógico (software) en el Registro Nacional del Derecho de Autor.
- La ley 37 de enero de 1993. La prestación del servicio de Telefonía Móvil Celular (TMC) en Colombia, se aprobó por el Congreso de la República mediante De acuerdo con esta Ley, se permite al Ministerio de Comunicaciones adjudicar la concesión de la prestación del servicio de TMC, previo un trámite de licitación pública
- El estándar IEEE 830 o IEEE Recommended Practice for Software Requirements Specifications describe una serie de recomendaciones prácticas para la creación de una buena especificación de requisitos de software (ERS/SRS)<sup>19</sup>

---

<sup>18</sup> [http://standards.ieee.org/catalog/olis/arch\\_se.html](http://standards.ieee.org/catalog/olis/arch_se.html)

<sup>19</sup> <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel4%2F5841%2F15571%2F00720574.pdf&authDecision=-203>



## **5. DESARROLLO INGENIERIL**

El proceso que se llevo a cabo para llegar a la solución del problema planteado se enmarca en la interacción de las etapas descritas a continuación, y que dan alcance al método de desarrollo aplicado: Modelo Cascada.

El análisis está compuesto por las etapas de análisis e ingeniería del sistema y el análisis de requisitos del modelo de cascada.

### **5.1. ANÁLISIS**

Esta etapa permite recolectar y detallar las especificaciones de los requerimientos del prototipo a desarrollar.

#### **5.1.1 Especificación de requerimientos**

Se desarrollo siguiendo las directrices del estándar IEEE 830 o IEEE Recommended Practice for Software Requirements Specifications.

##### **5.1.1.1 Propósito**

Tiene como propósito definir los requerimientos que debe tener la aplicación para monitoreo remoto de un servidor GNU/LINUX por medio de dispositivos móviles.

##### **5.1.1.2 Ámbito**

La aplicación para monitoreo remoto de un servidor GNU/LINUX esta orientada a dispositivos móviles con el fin de fomentar el aprendizaje en nuevas tecnologías, esta aplicación esta restringida a tres componentes principales: un agente, un servidor y una interfaz segura de conexión.

##### **5.1.1.3 Referencias**

Se tuvo como referencia los documentos:

- IEEE Std 830 – IEEE Guide to software Requirements Specifications IEEE Standard Borrada.
- Guía del IEEE para la Especificación de Requerimientos Software.

##### **5.1.1.4 Descripción General**

A continuación se presentan los factores que afectan la aplicación y sus

requerimientos.

#### **5.1.1.5 Perspectiva del prototipo**

- La aplicación para monitoreo remoto de un servidor GNU/LINUX orientada a dispositivos móviles esta diseñada para ejecutarse en dispositivos móviles de gama media

#### **5.1.1.6 Funciones del prototipo**

##### **Agente**

El agente estará instalado en los servidores a monitorear y su función será la de recolectar la información en tiempo real del servidor.

##### **Servidor**

El servidor se encargará de procesar la información recolectada por los agentes para luego ser mostrada al usuario vía web a través de un dispositivo móvil.

##### **Interfaz de conexión**

La interfaz de conexión se encargará de hacer un puente de comunicación segura entre el agente y el servidor; garantizando la integridad, fiabilidad y seguridad de los datos.

### **5.1.2 Requerimientos específicos**

#### **5.1.2.1 Requerimientos funcionales**

A continuación se describirán las entradas requeridas y las salidas en el ámbito de funcionamiento del aplicativo:

Entradas:

- Autenticación
- Petición de requerimiento

Salidas

- Visualización de resultado

### 5.1.2.2 Requerimientos no funcionales

Los requerimientos no funcionales del sistema son:

Hardware:

- Dispositivo Móvil compatible con WAP 1.0
- Servidor Linux con salida a Internet
- Servidores Linux para que sean objeto del monitoreo
- Red LAN que interconecte los servidores

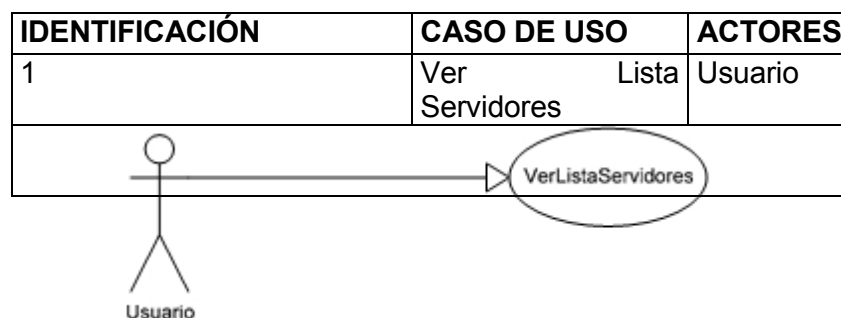
Software:

- Linux con versión de kernel 2.4 o superior.
- JRE 1.4.x o superior.

### 5.1.3 Vista de casos de uso

El análisis de las actividades de cada actor dentro del sistema se detalla en los siguientes casos de uso, pertenecientes a la etapa de análisis de requisitos del modelo de cascada:

**Tabla 1: Caso de uso Ver lista Servidores**



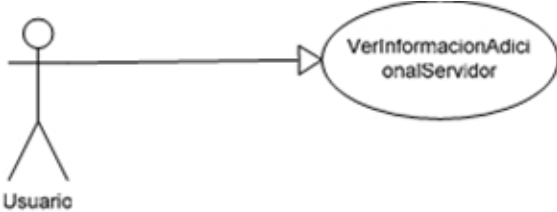
<b>OBJETIVO</b>	
Permitir que el usuario vea el listado de servidores los cuales pueden ser objeto de monitoreo	
<b>DESCRIPCIÓN</b>	
El sistema proveerá al usuario un listado de servidores, previamente inscritos y configurados, con el objeto de que el usuario seleccione un servidor para ver su estado actual.	
<b>Precondiciones</b>	El usuario debe haber ingresado al sistema correctamente identificándose como tal, mediante un nombre de usuario y una contraseña válida
<b>Postcondiciones</b>	El sistema proveerá al administrador de las herramientas necesarias para desempeñar correctamente su trabajo.
<b>Alternativas</b>	Cerrar Sesión
<b>CURSO NORMAL DE LOS EVENTOS</b>	
<b>ACCIÓN DEL ACTOR</b>	<b>RESPUESTA DEL SISTEMA</b>
El usuario solicita la lista de servidores que están siendo objeto de monitoreo.	Mostrar la lista de servidores inscritos para el monitoreo

**Tabla 2: Caso de uso Ver Información Servidores**

IDENTIFICACIÓN	CASO DE USO	ACTORES
2	Ver Información Servidores	Usuario
<p>The diagram shows a stick figure actor labeled 'Usuario' on the left. A horizontal line with an open arrowhead points from the actor to an oval use case labeled 'VerInformacionServidor' on the right.</p>		
<b>OBJETIVO</b>		
Usuario podrá ver la información relevante del servidor, como Estado CPU, Memoria, Discos y procesos.		

<b>DESCRIPCIÓN</b>		
El sistema mostrara al usuario en detalle la información relevante del servidor seleccionado, actualizándola en un intervalo de tiempo previamente establecido.		
<b>Precondiciones</b>	El usuario debe haber seleccionado un servidor valido que este siendo objeto de monitoreo.	
<b>Postcondiciones</b>	El sistema proveerá al usuario la información del estado del servidor.	
<b>Alternativas</b>	Ver Lista Servidores Ver Información Adicional Servidor	
<b>CURSO NORMAL DE LOS EVENTOS</b>		
<b>ACCIÓN DEL ACTOR</b>	<b>RESPUESTA DEL SISTEMA</b>	
El usuario selecciona el servidor del cual	Muestra información servidor	la del

**Tabla 3: Caso de uso Ver Información Adicional Servidor**

IDENTIFICACIÓN	CASO DE USO	ACTORES
3	Ver Información Adicional Servidor	Usuario
 <pre> graph LR   Usuario[Usuario] --&gt; UC((VerInformacionAdicionalServidor)) </pre>		
<b>OBJETIVO</b>		
El Usuario podrá ver la información adicional del servidor, como tiempo de encendido, usuarios conectados y versión del		

sistema.	
<b>DESCRIPCIÓN</b>	
El sistema mostrara al usuario en detalle la información adicional del servidor seleccionado.	
<b>Precondiciones</b>	El usuario debe haber seleccionado un servidor valido que este siendo objeto de monitoreo
<b>Postcondiciones</b>	El sistema proveerá al usuario la información adicional del servidor.
<b>Alternativas</b>	Ver Lista Servidores Ver Información Servidor
<b>CURSO NORMAL DE LOS EVENTOS</b>	
<b>ACCIÓN DEL ACTOR</b>	<b>RESPUESTA DEL SISTEMA</b>
El usuario selecciona el servidor del cual desea ver su información adicional	Muestra la información adicional del servidor seleccionado.

## 5.2 DISEÑO

Teniendo en cuenta el levantamiento de información y los requerimientos establecidos en la etapa de análisis, se obtuvieron los datos suficientes para llegar al siguiente diseño, el sistema contará con tres componentes principales: un agente, un servidor y una interfaz segura de conexión.

### 5.2.1 Diseño del agente

El agente es el encargado de recolectar la información a monitorear; esto lo realizará dependiendo de las peticiones que se le realice. El agente se encargará de recolectar dos grandes tipos de información: básica y extra. La información básica consta de los siguientes elementos:

- CPU

Valor de CPU usada por el usuario

Valor de CPU usada por el sistema

Valor de CPU de procesos en espera

Valor de CPU en procesos inactivos

Carga del procesador promedio en intervalos de 1, 5 y 15 minutos.

- Memoria RAM

Valor memoria usada

Valor memoria libre

Valor memoria total

- Memoria Swap

Valor memoria swap usada

Valor memoria swap libre

Valor memoria swap total

- Discos y particiones

Puntos de montaje

Tipo de dispositivo

Cantidad de espacio libre

Cantidad de espacio total

- Procesos

Número de procesos corriendo

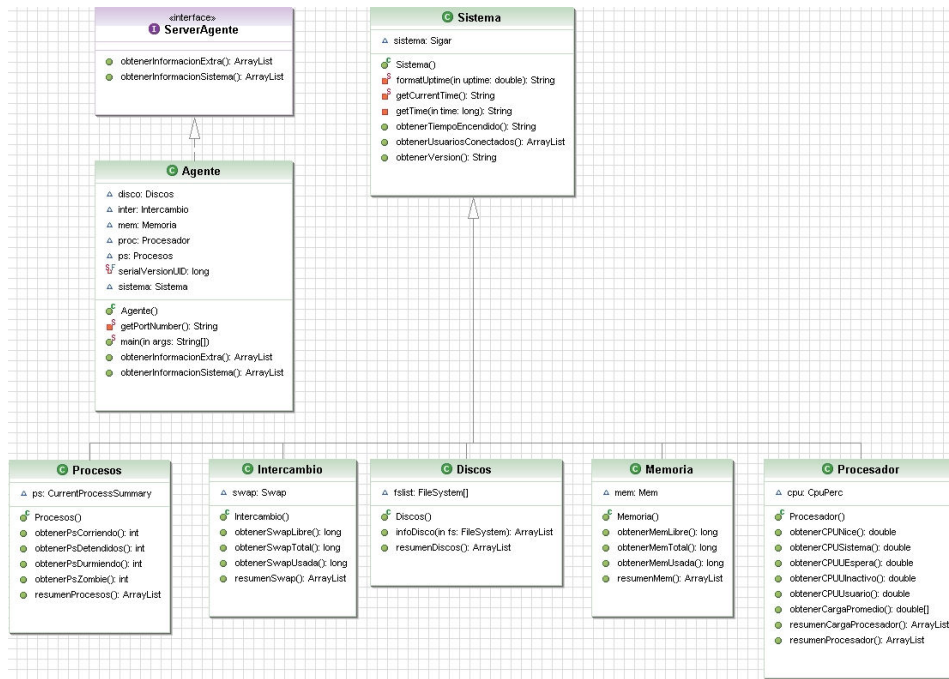
Número de procesos durmiendo

Número de procesos zombie

Número de procesos detenidos.

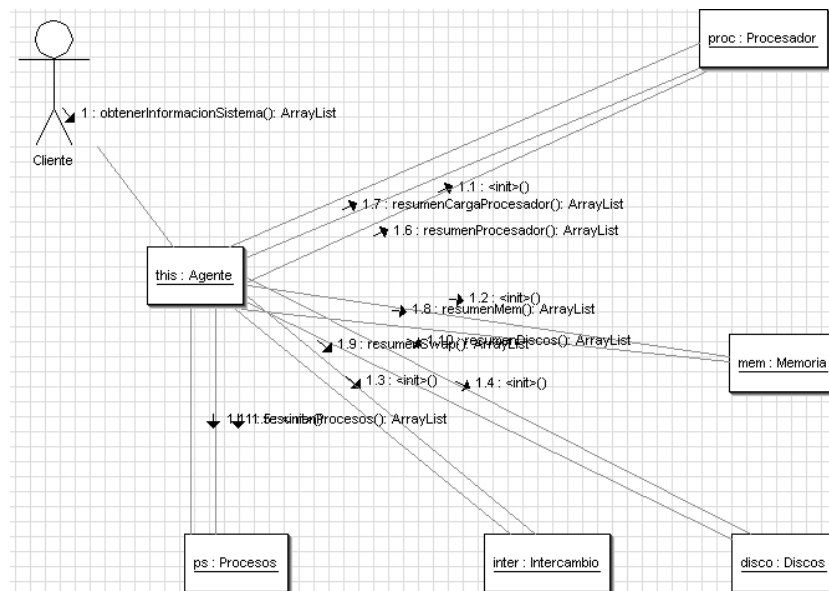
Para dar mayor claridad de la clase Agente se detalla en la figura 11 la cual muestra la consulta de las variables en el servidor que se desea monitorear

**Figura 11: Diagrama de clases del Agente**



La figura 12 permite obtener de forma grafica como el cliente obtiene la información del equipo a monitorear.

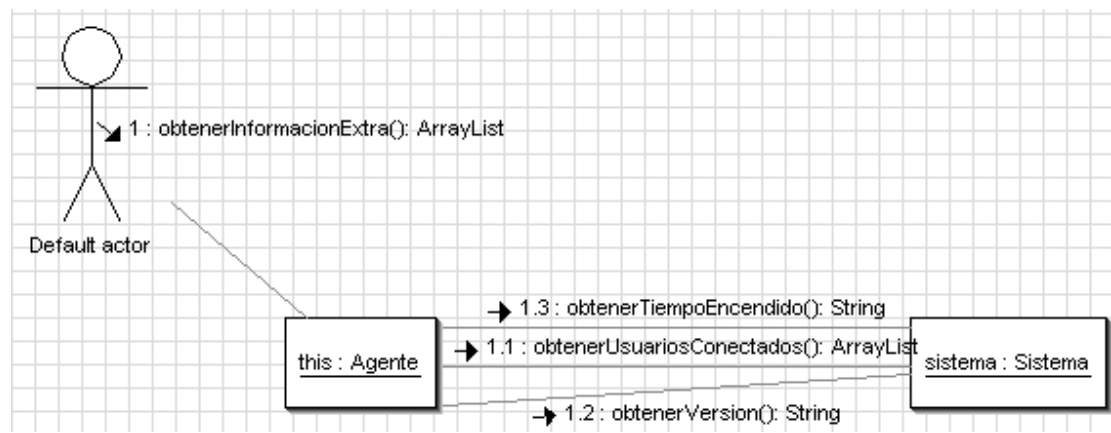
**Figura 12: Diagrama de colaboración –Obtener información del sistema**



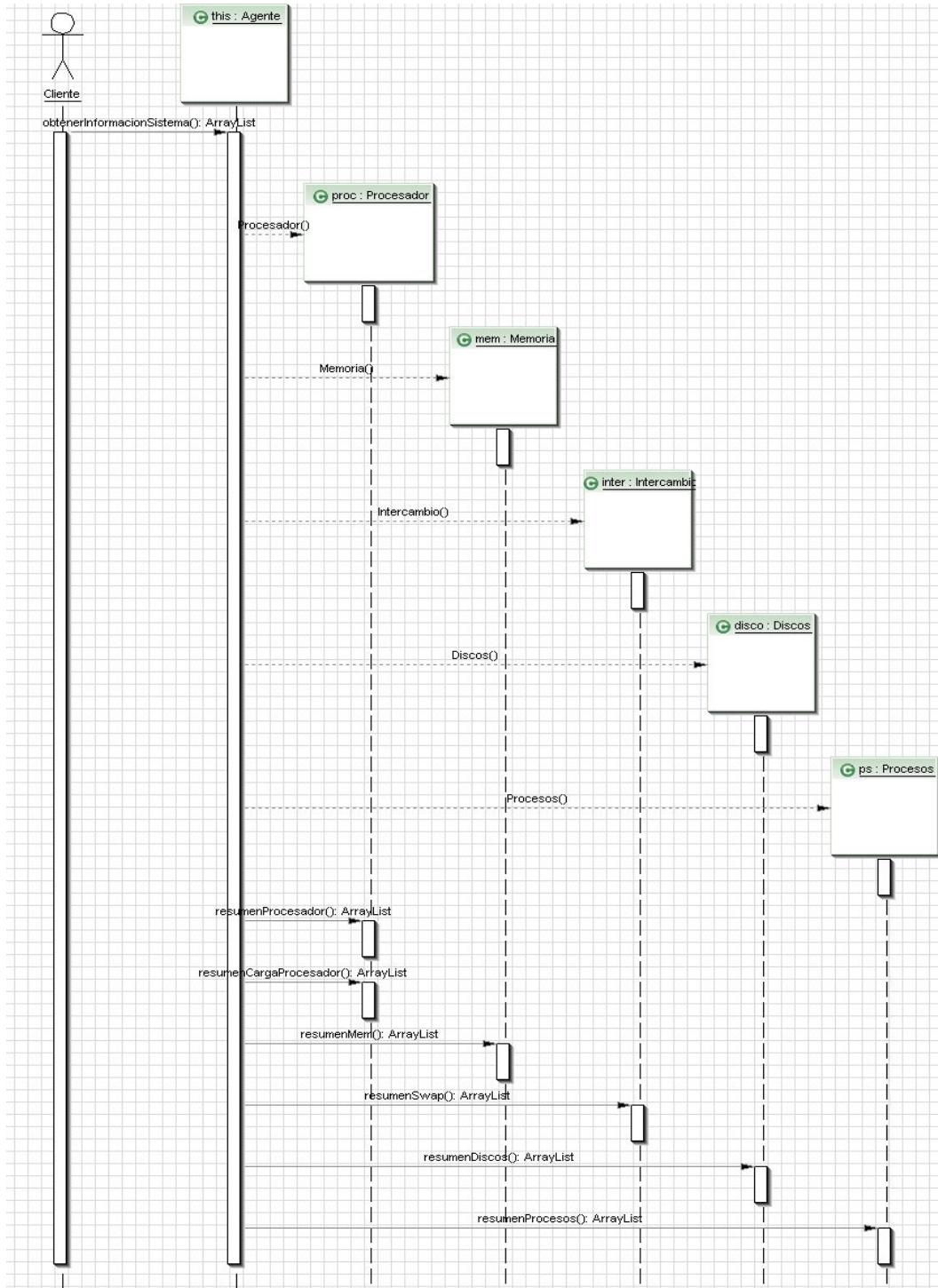


El cliente puede solicitar y visualizar información extra como el tiempo de encendido y los usuarios conectados tal como lo ilustra la figura 13

**Figura 13: Diagrama de colaboración –Obtener información extra del sistema**



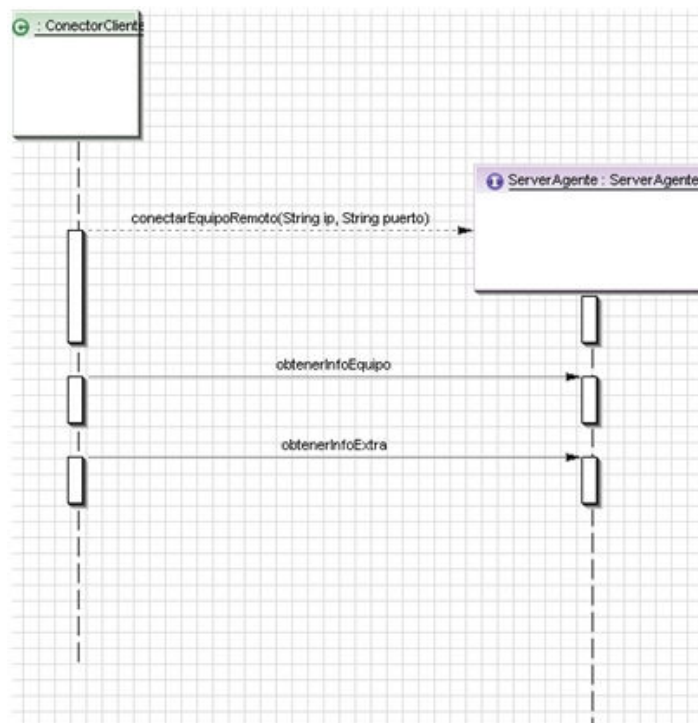
**Figura 14: Diagrama de secuencia – Obtener información del sistema**



### 5.2.2 Diseño interfaz conexión

La interfaz de conexión ver figura 15 es el componente que ayuda a la comunicación entre el servidor y el agente. El servidor debe solicitar los datos al agente y este a su vez debe retornar la información. Este componente será construido con RMI. RMI es la técnica que Java utiliza para obtener una distribución transparente de los objetos; es decir, objetos distribuidos de manera que el cliente no tiene conocimiento de donde reside el objeto.

**Figura 15: Diagrama de secuencia – Interfaz conexión**



### 5.2.3 Diseño servidor

El servidor es el encargado de recibir las peticiones de los usuarios a través de los dispositivos móviles, y a su vez realizar la petición al agente (por medio de la interfaz de conexión descrita anteriormente), en las figuras 16 - 20 se detallan las clases con sus respectivos procedimientos. Cuando este le responde, el se encarga de organizar la información y traducirla a lenguaje WML para ser mostrada a los usuarios. El servidor posee dos

componentes: un servidor WAP y un procesador de información. El servidor WAP es el punto de acceso a los usuarios. Este se encarga de recibir las peticiones y dar respuesta a las mismas. El procesador de información se encarga de traducir la información entregada por el agente, y codificarla en WML para ser mostrada en los dispositivos móviles.

Figura 16: Diagrama de clases – Componente Servidor

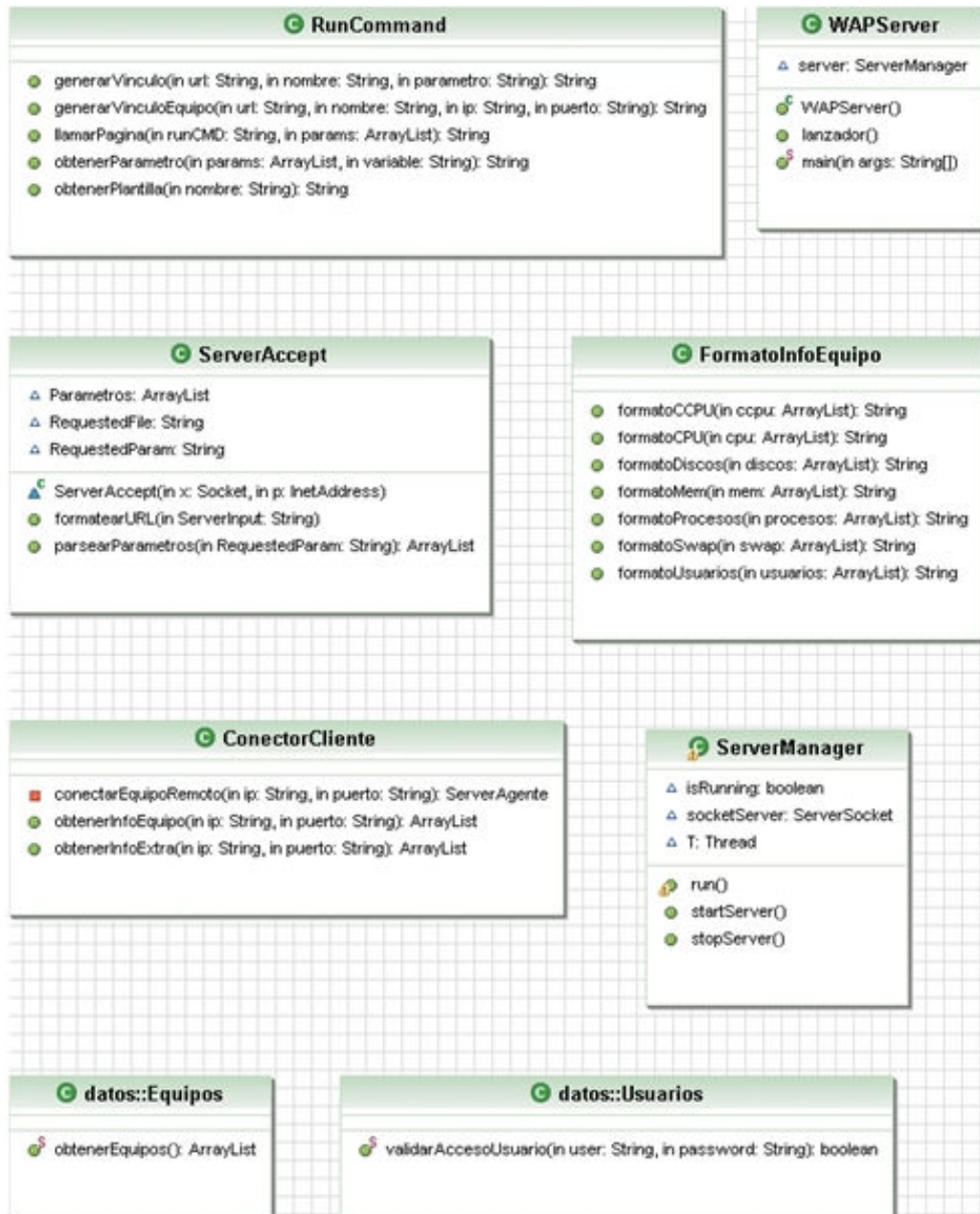
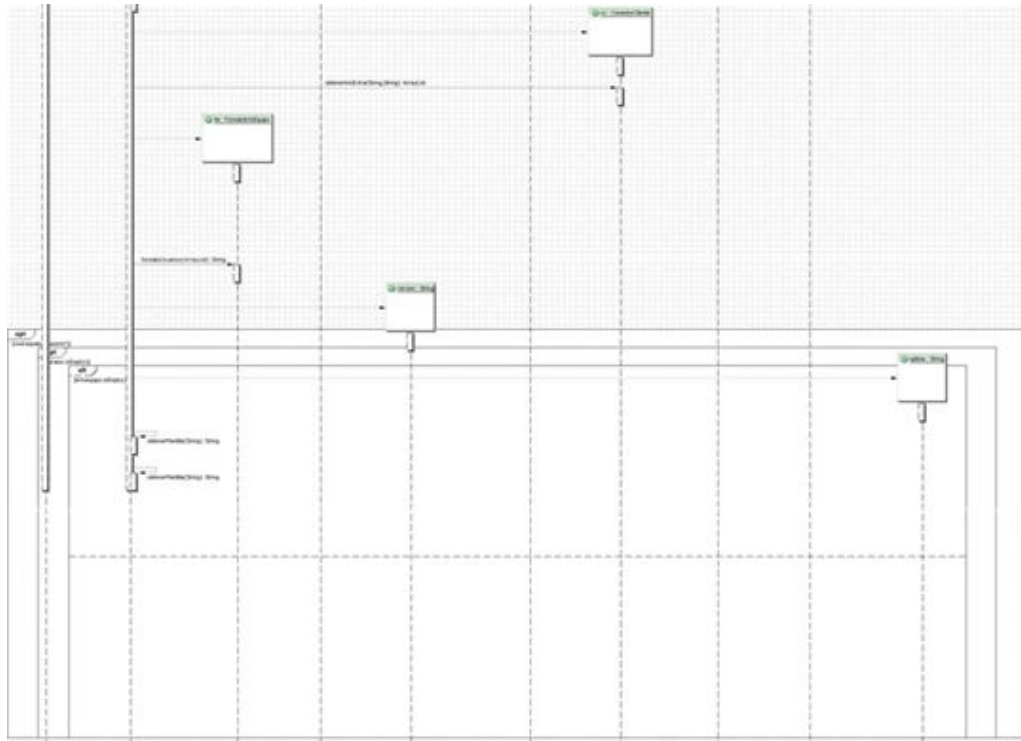






Figura 19: Diagrama de secuencia – Ver información extra del equipo







Nombre	varchar(32)	No	Null	Alias de equipo monitoreado
Ip	varchar(16)	No	Null	IP del equipo
Puerto	varchar(5)	No	Null	Puerto de conexión al equipo

**Tabla 5 Usuarios**

Tabla Usuario				
Campo	Tipo	Nulo	Predeterminado	Comentarios
Id	int(11)	No	Null	
User	varchar(16)	No	Null	Nombre de usuario
Pass	varchar(16)	No	Null	Contraseña de usuario

### 5.3 IMPLEMENTACIÓN

Se procede a realizar la construcción del software de acuerdo al diseño descrito anteriormente, utilizando la tecnología Java en nuestro proyecto, ofrecemos una gran flexibilidad para cubrir los objetivos y requerimientos planteados. Algunas de las características que impulsaron el uso de Java en nuestro proyecto son las siguientes:

- Modelo de distribución de objetos de forma transparente (RMI).
- Soporte para redes TCP/IP
- Soporte de seguridad propio del lenguaje.
- Programación orientada a objetos madura.
- Utilización de librerías externas que expanden las capacidades del lenguaje.

#### 5.3.1 Plataforma utilizada

Después de analizar las ventajas y desventajas que tiene cada plataforma se optó por implementar con WML. La selección de la plataforma se basó en el resultado del siguiente comparativo

**Tabla 6: Comparativo de plataformas**

	<b>J2ME</b>	<b>WAP</b>	<b>.NET</b>	<b>WML</b>
<b>Código Interpretado</b>	*	*		*
<b>Aplicaciones Internet</b>	*	*	*	*
<b>Aplicaciones Standalone</b>	*		*	*
<b>Aplicaciones Distribuidas</b>	*		*	*
<b>Aplicaciones Multimedia</b>	*		*	*
<b>Licencia Gratuita</b>	*	*		*
<b>Portabilidad</b>	*	*	*	*
<b>Ambiente de Desarrollo</b>	Completo	Completo		*
<b>Soporte de Colores</b>	Escala de grises y Colores	Aplicaciones sencillas y monocromáticas		*
<b>Imágenes</b>				Imágenes en tamaño reducido
<b>Plataforma</b>	Multiplataforma	Multiplataforma	Multiplataforma	Multiplataforma

El Wireless Markup Language es un lenguaje cuyo origen es el XML (eXtensible Markup Language). Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDA) dotados de tecnología WAP. Es una versión reducida del lenguaje HTML que facilita la conexión a Internet de dichos dispositivos y que además permite la visualización de páginas web en dispositivos inalámbricos que incluyan la tecnología WAP.

La visualización de la página dependerá del dispositivo que se use y de la forma en que este interprete el código, ya que varían entre sí. WML es un metalenguaje, lo que implica que además de usar etiquetas predefinidas se pueden crear componentes propios y tiene ciertas similitudes con otro lenguaje de etiquetas bastante conocido, el HTML (Hypertext Markup Language), utilizado para la creación de páginas web convencionales. Un consorcio formado por Nokia, Phone.com, Motorola y Ericsson, el WAP Forum, define la sintaxis, variables y elementos utilizados en WML. Algunos fabricantes han desarrollado capacidades adicionales a este estándar. Al igual que el HTML se sirve de un lenguaje de script como javascript para dotar de cierto dinamismo a sus documentos, WML dispone del WMLS que es un lenguaje bastante similar al Javascript, pero con alguna diferencia fundamental.<sup>21</sup>

---

<sup>21</sup> Fuente: [http://es.wikipedia.org/wiki/Wireless\\_Markup\\_Language](http://es.wikipedia.org/wiki/Wireless_Markup_Language)

### 5.3.2 IDE

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk u Objective-C. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. Este es el caso de Eclipse, al que mediante plugins se le puede añadir soporte de lenguajes adicionales<sup>22</sup>.

---

<sup>22</sup> Fuente: [http://es.wikipedia.org/wiki/Entorno\\_de\\_desarrollo\\_integrado](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado)

**Tabla 7: Comparativo IDE**

<b>NET BEANS</b>	<b>JDEVELOPER</b>	<b>ECLIPSE</b>
Desarrollo de aplicaciones multiplataforma sobre: MacOS, Windows, Linux. Add-ons para desarrollo Móvil, desarrollo Web grafico, integración con SOA, optimización de aplicaciones y desarrollo con C y C++. Cliente CVS integrado. Crecimiento de plataforma por medio de plugins. Entre los plugins que existen se tienen los siguientes: Herramientas java que sirven para la mejora de desarrollo de aplicaciones Herramientas de modelado UML Herramientas XML, etc. Struts, JSF, EJB, WebServices, etc.	Es un entorno de desarrollo muy completo que proporciona herramientas de última generación para el desarrollo de aplicaciones JAVA. Esta también es una IDE de distribución gratuita auspiciada por Oracle group. Entre las capacidades que proporciona se tienen las siguientes: ·Desarrollo de aplicaciones siguiendo la metodología ADF. ·Creación de interfaces visuales. ·Uso de HTML ·Aplicaciones XML · Manejo de Bases de datos ·Struts, JSF, EJB, WebServices, etc. · Cliente CVS integrado	Es una herramienta de las que se denominan “entornos integrados de desarrollo” porque incluye en un sólo programa la capacidad de realizar múltiples operaciones integradas sobre proyectos, facilita enormemente la tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. Aunque Eclipse pretende ser un entorno versátil soportando varios lenguajes de programación, es con el lenguaje Java con el que mejor se integra y con el que ha ganado su popularidad. Eclipse es una aplicación gratuita y de código abierto

### 5.3.3 Construcción del software

Luego de haber realizado el diseño de los componentes del sistema, se procede a realizar la construcción del software de acuerdo al diseño descrito anteriormente y siguiendo los lineamientos de los requerimientos funcionales y no funcionales establecidos para el proyecto.

Uno de los requerimientos no funcionales requería que el lenguaje de desarrollo debe ser Java debido a que la tecnología Java ha evolucionado hacia estándares cada vez más abiertos y aceptados por las principales compañías de software a nivel mundial y facilita en gran medida la integración entre sistemas heterogéneos.

Por tanto, utilizando la tecnología Java en nuestro proyecto, ofrecemos una gran flexibilidad para cubrir los objetivos y requerimientos planteados. Algunas de las características que impulsaron el uso de Java en nuestro proyecto son las siguientes:

- Modelo de distribución de objetos de forma transparente (RMI).
- Soporte para redes TCP/IP
- Soporte de seguridad propio del lenguaje.
- Programación orientada a objetos madura.

- Utilización de librerías externas que expanden las capacidades del lenguaje.

Se hace necesario la utilización de una base de datos para almacenar la información de usuarios y equipos que son objeto de monitoreo. Debido a la cantidad de información que se va a manejar, no se hace necesario una base de datos robusta, por el contrario se requiere una base de datos ligera, portátil y que no sea necesario instalarla; debido a que como se expuso anteriormente con la librería SIGAR, la base de datos debe tener soporte para diferentes sistemas operativos.

Luego de consultar las diferentes alternativas y teniendo en cuenta el entorno de desarrollo se decidió a usar la base de datos HSQLDB.

HSQLDB (Hypersonic SQL Data Base)<sup>23</sup> es un sistema gestor de bases de datos libre escrito en Java.

### **5.3.4 Estructura del código**

Dado a los lineamientos especificados en el diseño, toda la construcción del prototipo debe ser orientada a objetos, siguiendo los estándares del lenguaje de desarrollo. Como se ha explicado anteriormente, la aplicación se distribuye en dos componentes: Agente y Servidor, lo cual obliga a independizar el desarrollo de cada componente como se explica a continuación:

#### **5.3.4.1 Estructura del agente**

El agente está constituido por: un paquete y la librería SIGAR. El paquete es un mecanismo para organizar los elementos en grupos. El paquete posee las siguientes clases:

- Agente: Es la clase principal de la aplicación.
- Sistema: Es el puente de comunicación con los métodos de las librerías
- Procesador: Es el encargado de recolectar los datos del procesador.
- Memoria: Se encarga de recolectar los datos de la memoria física del sistema.
- Intercambio: Es el encargado de recolectar los datos de la memoria de intercambio.
- Discos: Se encarga de recolectar la información del estado de los discos físicos, particiones y puntos de montaje.

---

<sup>23</sup> <http://hsqldb.org/web/hsqldbDocsFrame.html>

- Procesos: Es el encargado de recolectar la información de procesos del sistema.
- ServerAgente: Es una interface que permite que una aplicación externa invoque los métodos remotamente.

#### **5.3.4.2 Estructura del servidor**

El servidor está compuesto por tres paquetes y la librería de la base de datos. Los paquetes y sus clases son las siguientes:

##### **5.3.4.2.1 Paquete base de datos**

Agrupar el conector hacia la base de datos y está compuesto de la siguiente clase:

- ConexionHsqldbFD: La clase se encarga de realizar la conexión JDBC hacia la base de datos.

##### **5.3.4.2.2 Paquete datos**

Contiene las clases que interactúan con las tablas de la base de datos y se encargan de manipular la información. Se compone de las siguientes clases:

- Usuarios: Se encarga la información de usuarios.
- Equipos: Gestiona la información de equipos inscritos para el monitoreo

##### **5.3.4.2.3 Paquete servidor**

Este paquete agrupa todas las clases que constituyen el servidor WAP. Está compuesto por las siguientes clases:

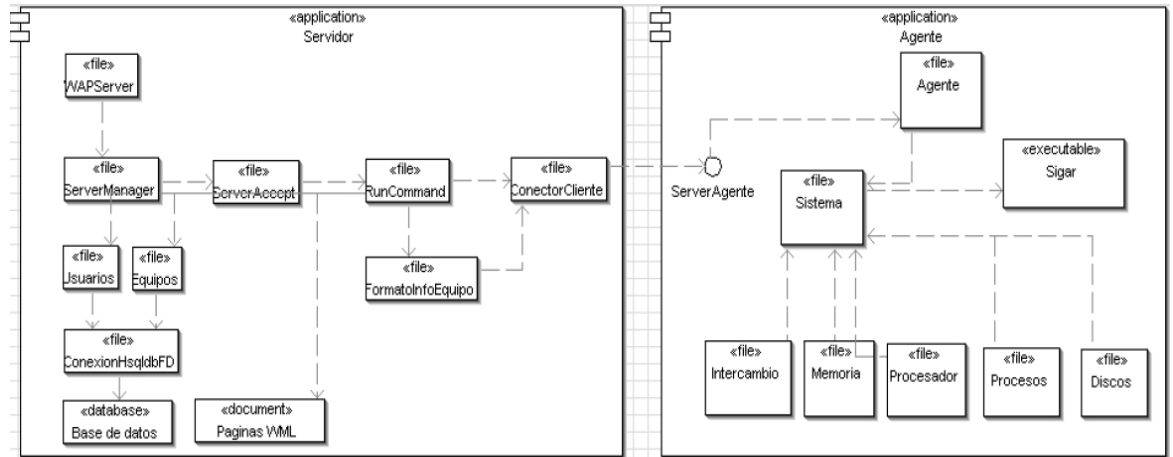
- WAPServer: Clase principal de la aplicación.
- ServerManager: Se encarga de en ejecución la aplicación y de determinar su estado.
- ServerAccept: Es el encargado de procesar las peticiones realizadas desde los dispositivos móviles.
- RunCommand: Se encarga de la preparación y procesamiento de las páginas WML.
- ConectorCliente: Realiza la invocación a los objetos remotos ubicados en el agente.
- FormatoInfoEquipo: Formatea la información obtenida por ConectorCliente, para ser mostrada a los usuarios por medio de las páginas WML.

Aparte de los paquetes descritos anteriormente, el servidor contiene un directorio donde se encuentra ubicada la base de datos.

### 5.3.5 Diagrama de componentes

Para entender en conjunto la estructura y el funcionamiento interno de la aplicación, se muestra a continuación un diagrama de componentes de todo el prototipo.

**Figura 21: Diagrama de componentes**



### 5.3.6 Interfaz gráfica

Las pantallas relacionadas a continuación corresponden a la información que contiene cada uno de los menús:

**Figura 22: Servidores registrados**



**Figura 23: Información del equipo**

**Información del Equipo**

[Ir a Menu Principal](#) | [Ir a Informacion Adicional](#)

**CPU**  
*Usuario: 7.8% Sistema: 9.4%*  
*Espera: 0.0% Nice: 0.0%*  
*Inactivo: 82.7%*

**CARGA CPU**  
*1 min: 0.0 5 min: 0.0 15 min: 0.0*

**MEMORIA**  
*Libre: 573M Usada: 1.2G Total: 1.7G*

**SWAP**  
*Libre: 2.8G Usada: 1.2G Total: 4.0G*

**DISCOS**

Unidad	Uso	Total	Tipo
C:	72%	88G	NTFS/local
D:	-	0	cdrom/cdrom
J:	87%	84G	NTFS/local

**PROCESOS**  
*Corriendo: 76 Durmiendo: 0*  
*Zombie: 0 Detenidos: 0*

[Ir a Menu Principal](#) | [Ir a Informacion Adicional](#)

**Información del Equipo**

[Ir a Menu Principal](#) | [Ir a Informacion Equipo](#)

**USUARIOS**

Usuario	Dispositivo	Tiempo	Host
SERVICIO LOCAL	console	unknown	(NT AUTHORITY)
Servicio de red	console	unknown	(NT AUTHORITY)
Juan Carlos Valencia	console	may 01 06:05	(JCVO)
SYSTEM	console	unknown	(NT AUTHORITY)

**VERSION**  
*Microsoft Windows Vista*  
*Nombre: Win32*  
*Version: 6.1*  
*Arquitectura: x86*  
**Actualizaciones:**  
**Maquina:**

**TIEMPO ENCENDIDO**  
*Hora Actual: 11:54 AM*  
*Tiempo Encendido: 49 min*

[Ir a Menu Principal](#) | [Ir a Informacion Equipo](#)

**Figura 24: Listado y detalle de reportes**

**RemoteWAP - Listado Reportes**

**Reportes**

[Uso CPU](#)

[Uso Memoria](#)

[Salir](#)

**RemoteWAP - Uso CPU**

**Uso CPU**

Los siguientes valores corresponden al porcentaje total de CPU

*Usuario: 0.0% Sistema: 0.0%*  
*Espera: 0.0% Inactivo: 0.0%*

[Ir a Menu Principal](#) | [Salir](#)



## 6. PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

Una vez realizada la implementación del prototipo, se ejecutaron las pruebas necesarias para detectar fallos y componentes defectuosos, así mismo probar el flujo de operación del prototipo.

### 6.1 HERRAMIENTAS DE EJECUCIÓN DE PRUEBAS

Existen diferentes herramientas entre ellas las que se describen a continuación:

#### 6.1.1 JUnit

Es un framework de código abierto desarrollado especialmente para crear, ejecutar y hacer reportes de estado de conjuntos de Prueba Unitaria automatizadas hechos en lenguaje Java. JUnit es uno de los frameworks más populares en Java para realizar pruebas unitarias y llevar un desarrollo utilizando la práctica de Test Driven Development.

Al momento de ejecutar los tests, JUnit ejecuta los siguientes métodos propios de cada clase de test:

Método `@BeforeClass` (método estático, antes de la construcción de la clase)

Por cada método anotado con `@Test`

Constructor de la clase de test

Método anotado con `@Before`

Método anotado con `@Test`

Método anotado con `@After`

Al finalizar todos los `@Test`, se ejecuta `@AfterClass`

#### 6.1.2 TestNG

Es un framework para pruebas y testing que trabaja con Java y está basado en JUnit (para Java) y NUnit (para .NET), pero introduciendo nuevas funcionalidades que los hacen más poderosos y fáciles de usar, tales como: Anotaciones JDK 5 (Annotations) (JDK 1.4 también es soportado con JavaDoc annotations).

Configuración flexible de pruebas.

Soporte para pruebas para data-driven testing (with `@DataProvider`).

Soporte de pasaje de parámetros.

Permite distribución de las pruebas en máquinas esclavas.

Modelo de ejecución poderoso (TestSuite nunca más).

Soportado por herramientas y plugins importantes y variados como: (Eclipse, IDEA, Maven, etc.).

Permite embeber BeanShell para una flexibilidad más amplia.

Funciones JDK por defecto de runtime y logging. (sin dependencias)

Métodos dependientes para pruebas sobre servidores de aplicación.

TestNG está diseñado para cubrir todas las categorías de las pruebas: unitarias, funcionales, end-to-end, integración, etc.

## **6.2 PLANIFICACIÓN DE LA PRUEBA**

Para la realización de las pruebas del prototipo se hacen necesario el siguiente escenario de trabajo:

- 2 computadores con sistema operativo Linux
- 1 Un switch o hub para realizar una conexión Ethernet preferiblemente con soporte 100 base T
- Una conexión a Internet con una IP pública.
- 1 dispositivo móvil con soporte WAP y conexión a Internet.

## **6.3 PREPARACIÓN DEL ESCENARIO**

### **6.3.1 Configuración del equipo para el Agente**

- COMPAQ Proliant M300
- Procesador Pentium III a 1 GHz
- 2 GB en RAM
- RAID 1 Disco duro SCSI de 38 GB
- Linux Ubuntu 6.10 Server (kernel Linux 2.6.18)
- Tarjeta de red 10/100

Para el funcionamiento del prototipo se instalara JRE 1.4.x o superior

### **6.3.2 Configuración de equipo para el Servidor**

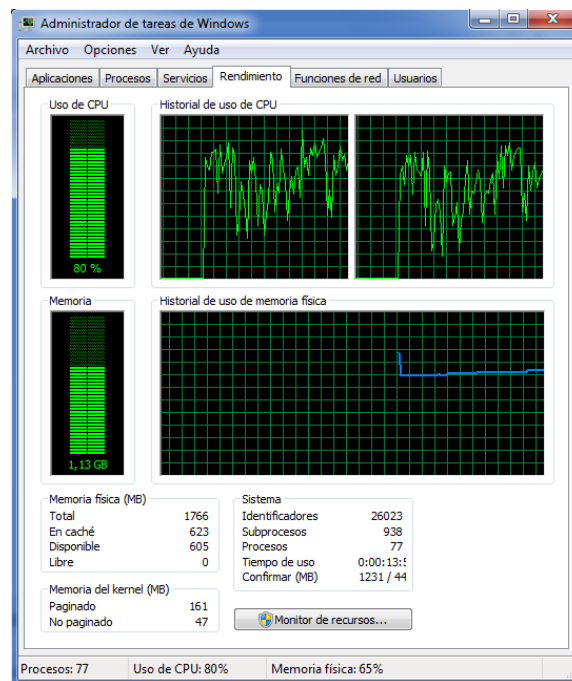
- COMPAQ Evo
- Procesador Pentium IV a 1.7 GHz

- 1 GB en RAM
- Disco duro IDE de 40 GB
- Linux Ubuntu 7.04
- Tarjeta de red 10/100

Para el funcionamiento del prototipo se instalara JRE 1.4.x o superior

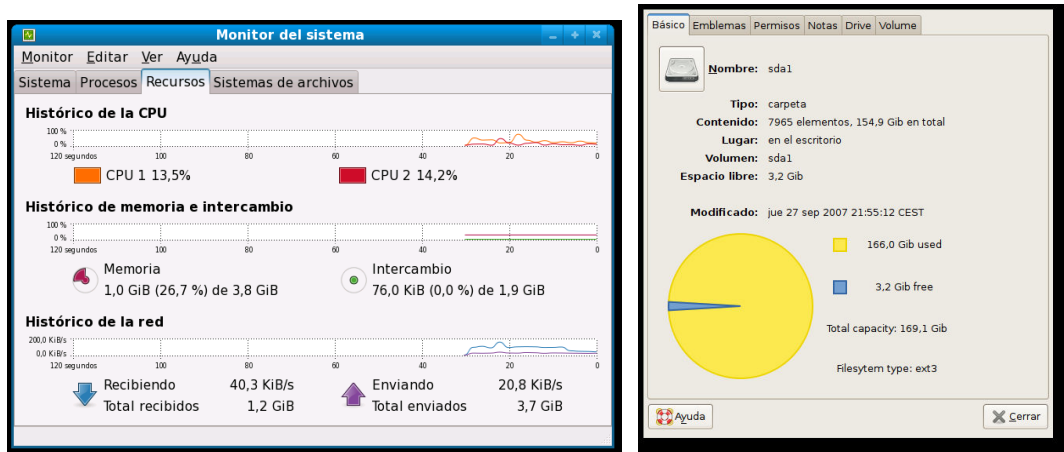
El visor de sucesos es un sistema de logs que guarda todos aquellos errores que padece Windows, gracias a los logs del visor de sucesos se pueden encontrar problemas del equipo, depurar aplicaciones o visualizar los sucesos importantes que han acontecidos.

**Figura 25 Administrador de sucesos Windows**



Para los servidores LINUX se chequea los procesos, el espacio en disco y la carga media general de la máquina en la cual está instalado el monnitor; también permite crear constancia de errores que se produzcan en el sistema mediante el uso de archivos de registro, el reporte arrojado por el monitor del sistema LINUX se visualiza en la figura 26.

**Figura 26 Monitor del sistema de Linux**



### 6.3.3 Configuración red

La configuración de la red es la siguiente:

- Switch 3Com
- IP Agente: 192.168.0.3
- IP Servidor: 192.168.0.2

## 6.4 PRUEBAS DE INSTALACIÓN Y CONFIGURACIÓN

### 6.4.1 Instalación del agente

Se procedió con el empaquetado de las clases del agente en un formato autoejecutable de Java (JAR) y posteriormente se realizó un instalador usando el software Izipack.

Luego se procedió con la instalación del software. Se revisó que los puertos en el equipo estuvieran habilitados y fueran accesibles. La instalación del agente concluyó sin ningún contratiempo.

### 6.4.2 Instalación del servidor

Se procedió con el empaquetado de las clases del servidor en un formato autoejecutable de Java (JAR) y a continuación se realizó un instalador

usando el software Izpack.

Posteriormente se realizó la instalación. Se registro la información del equipo agente en la base de datos. Se verifico que los puertos fueran accesibles. También se verifico que la conexión a Internet estuviera habilitada para este equipo.

## **6.5 PRUEBAS DE EJECUCIÓN**

Se inicio la ejecución de el agente y el servidor en las maquinas respectivas. Una vez hecho esto se utilizo el dispositivo móvil para invocar la página del prototipo. Se hizo la autenticación con usuario inexistente en la base de datos dando como respuesta el rechazo al ingreso de la aplicación. Posteriormente se realizo en ingreso con usuario con credenciales validas, permitiendo el acceso al menú principal de la aplicación.

Se invoco la información básica del equipo objeto del monitoreo obteniendo respuesta en un tiempo menor a 10 segundos. Luego se solicito la información extra del equipo obteniéndola satisfactoriamente.

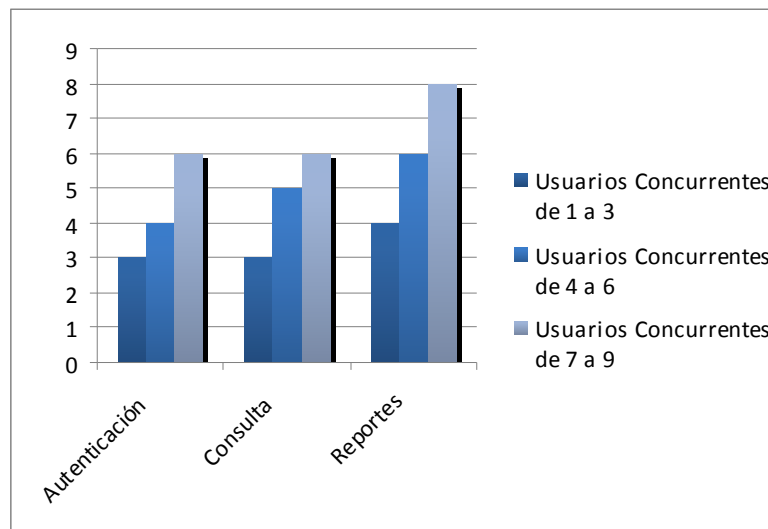
En la ejecución no se encontraron defectos en el prototipo.

## **6.6 RESULTADOS DE LAS PRUEBAS**

Una vez realizadas las pruebas necesarias se concluye que el prototipo funciona correctamente ciñéndose al diseño realizado y cumpliendo los objetivos propuestos. Dado que las pruebas fueron satisfactorias no se hace necesario regresar a una etapa de diseño e implementación del prototipo. En la tabla 8 se encuentran los resultados por cada 3 dispositivos conectados simultáneamente realizando acciones como Autenticación, Consultas y Reportes.

**Tabla 8 Número de dispositivos vs Acción**

No de Dispositivos / Acción	Autenticación	Consulta	Reportes
Usuarios Concurrentes de 1 a 3	3	3	4
Usuarios Concurrentes de 4 a 6	4	5	6
Usuarios Concurrentes de 7 a 9	6	6	8



## 7. CONCLUSIONES

El administrador de redes tiene a cargo una de las labores más importantes de gestión, es el monitoreo de la red y sus componentes, algunos procesos usuales son monitorear el consumo de CPU y Memoria (RAM y SWAP); cada cierto tiempo es necesario que entremos a revisar cuando espacio en disco hay libre en los servidores que administra.

Un sistema administrador de redes debe estar orientado a permitir mayor control sobre la red, en este caso de estudio sobre el servidor proporcionando herramientas que faciliten las labores diarias.

Una de las ventajas de trabajar con Java es que es compatible con la mayoría de móviles disponibles en el mercado y para dispositivos con capacidades limitadas, En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.

El software desarrollado se encarga de realizar acciones en el cliente como solicitar datos, mostrar mensajes, herramienta que cobra valor agregado al momento de tomar decisiones.

El diseño de las interfaces a través de WAP permitió alojar páginas para ser enviadas a través de la web y ser visualizadas en los navegadores de los dispositivos móviles, aunque WAP trabaja bajo restricciones de memoria, esta limitación no afectó el cumplimiento del objetivo planteado.

Dichas interfaces se encargaron de hacer el puente de comunicación segura a través de autenticación entre el agente y el servidor, garantizando en todo momento la integridad, fiabilidad y seguridad de datos.

La implementación del prototipo cumplió con el aseguramiento de consultas de las variables básicas para el monitoreo de servidores LINUX, a través de dispositivos móviles

Las pruebas ejecutadas al software, permiten asegurar la calidad de cada uno de los elementos que lo componen y así poder establecer los tiempos de respuesta.

Los usuarios de este sistema afectan el funcionamiento del sistema debido a que ellos son los que ejecutan los procesos y tareas e interactúan directamente con el sistema, las pruebas permitieron establecer el umbral de tolerancia para determinar una medida precisa de rendimiento.

## 8. BIBLIOGRAFÍA

ACEVEDO Marco A, SANCHEZ Miguel, TOLEDO Carlos, Monitoreo en tiempo real, con el protocolo WAP, Mexico, 2002.

AREHART Charles, PROFESIONAL WAP with WML, WMLScript, ASP, JSP, XML, XSLT, WTA Push and VoiceXML, Estados Unidos, 2000.

AREHART Charles, CHIDAMBARAM Nirmal, GURUPRASAD Shashikiran. PROFESIONAL WAP with WML, WMLScript, ASP, JSP, XML, XSLT, WTA Push and VoiceXML, Estados Unidos, 2000.

BELLALTA Boris, RINCÓN David, OLIVER Miquel, Análisis de Tráfico y Capacidad para Servicios WAP sobre Redes GSM/GPRS, España, 2000.

DEPARTAMENTO NACIONAL DE PLANEACIÓN UNIDAD DE INFRAESTRUCTURA Y ENERGÍA La Telefonía Móvil Celular en Colombia 1998 Documento Sectorial - Departamento Nacional de Planeación Unidad de Infraestructura y Energía.

FREIER, A. O., KARLTON, P. y KOCHER, P. C. (1996). The SSL Protocol Specification 3.0. Disponible en <http://www.wmlclub.com/ind=http://home.netscape.com/>

FORTA Ben, Keith Lauver, Paul Fonte, Robert M. Juncker, Ronan Mandel and Dylan Bromby. WAP Development with WML and WMLScript, United States, 2000

GROSSO, William, Java RMI, United States, 2004 MONJAS Miguel Angel. Seguridad en WAP II. Disponible en <http://www.wmlclub.com/articulos/seguridadwap.htm>

JORBA, Joseph Administración avanzada de GNU/Linux. Marzo 2004.

LETELIER, Patricio, "Metodología y Tecnología de la Programación", Servicio de Publicaciones Universidad Politecnica de Valencia, 269 paginas, Año 1997.

MARTÍNEZ, Normand, "Introducción a la ingeniería del software" Delta Publicaciones so Amo, Loïc , 542 páginas, Año 2005.

MORELO Xavier, Monitorización de sistemas y programas, España, 2004.



NOKIA. Nokia WAP Toolkit 2.1 User's Guide, Product Number SDK-01-000-001, 2001. 99

NOKIA. Nokia WAP Toolkit 2.1 Developer's Guide, Product Number SDK-01-000-002, 2001.

OAKS, Scott. Java Security, United States, 2001

PEREZ Elias, Administración de sistemas operativos, Mexico, 2001.

SETCo. SET Secure Electronic Transaction Specification. Book 3: Formal Protocol Definition. Version 1.0. Disponible en [http://www.setco.org/download/set\\_bk3.pdf](http://www.setco.org/download/set_bk3.pdf)

TANENBAUM, Andrew S, "Redes de Computadoras" Pearson Educación, 891 páginas, Año 2003.

TANENBAUM Andrew. VAN STEEN Maarten, Distributed Systems: Principles and Paradigms Prentice Hall, 2002

VALLESPÍ Albert, OLIVER Miquel, Personalización de servicios WAP mediante gestión de contenidos XML y bases de datos relacionales, España, 2000.

WAP Forum. Wireless Application Protocol Architecture Specification. URL: <http://www.wapforum.org/what/technical.htm> (Octubre 2001).

## WEBGRAFÍA

Consulta en las páginas web listadas a continuación

<http://www.dosideas.com/wiki/JUnit>. Consultada por última vez Febrero 2010.

<http://www.elcodigo.net/cgi-bin/DBread.cgi?tabla=herramientas&=0&clave=46&info=1>. Consultada por última vez Abril 2010.

[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/msp/aldrette\\_m\\_r/capitulo1.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/msp/aldrette_m_r/capitulo1.pdf). Consultada por última vez Febrero 2010.

<http://www.chuidiang.com/java/rmi/rmi.php>. Consultada por última vez Febrero 2010.

<http://proactiva-calidad.com/java/rmi/introduccion.html>. Consultada por última vez Abril 2010.

[http://es.wikipedia.org/wiki/Java\\_Remote\\_Method\\_Invocation](http://es.wikipedia.org/wiki/Java_Remote_Method_Invocation). Consultada por última vez Febrero 2010.

<http://www.wapforum.org/faqs/index.htm>. Consultada por última vez Febrero 2010.

<http://www.wapforum.org/what/benefits.htm>. Consultada por última vez Febrero 2010.

<http://www.nokia.com/corporate/wap/future.html>. Consultada por última vez Abril 2010.

<http://www.phone.com/industry/wap.html>. Consultada por última vez Febrero 2010.

<http://www.ericsson.com/WAP/what/what.shtml>. Consultada por última vez Abril 2010.

<http://www.wap.com/findwap.html>. Consultada por última vez Febrero 2010.

<http://www.elviajero.org/antoniux/tutos/servidores8.pdf>. Consultada por última vez Febrero 2010.

[http://aprender21.com/cursolinux.php?cpg=adw\\_erirblnx&gclid=CMrq5YbTvKECFROG5godUCAC9w#](http://aprender21.com/cursolinux.php?cpg=adw_erirblnx&gclid=CMrq5YbTvKECFROG5godUCAC9w#). Consultada por última vez Febrero 2010.

[http://java.com/es/download/faq/whatis\\_j2me.xml](http://java.com/es/download/faq/whatis_j2me.xml). Consultada por última vez Abril 2010.

[http://standards.ieee.org/catalog/olis/arch\\_se.html](http://standards.ieee.org/catalog/olis/arch_se.html). Consultada por última vez Mayo 2010.

FECHA	01-06-2010
-------	------------

NÚMERO RAE	
PROGRAMA	Ingeniería de Sistemas

AUTOR (ES)	CASAS ARÉVALO, Catalina y VALENCIA ORTIZ, Juan Carlos.
TÍTULO	Diseño e implementación de una aplicación para monitoreo remoto de un servidor GNU/LINUX por medio de dispositivos móviles

PALABRAS CLAVES	
	Monitoreo remoto, servidor GNU/LINUX, dispositivo móvil, Telefonía móvil celular, Telefonía móvil celular, WAP, WML, UML, HSQLDB, RMI, Estándar 830-1988 IEEE,

DESCRIPCIÓN	
	<p>En la actualidad la tecnología se ha centrado en la telefonía móvil, puesto que los teléfonos celulares permiten acceder a la información desde cualquier parte y estar en continuo movimiento, así mismo ofrecen diferentes aplicaciones y servicios a los que los usuarios pueden acceder de una manera fácil y rápida, además hoy en día es más fácil acceder a un celular y la mayoría de las personas cuentan con uno, sin distinción alguna. Por todo esto se escogieron los dispositivos móviles celulares como la mejor alternativa para el desarrollo del proyecto.</p> <p>La primera etapa del proyecto es el análisis, en la cual se mostraran las diferentes arquitecturas que permiten desarrollar aplicaciones en teléfonos celulares, con sus respectivas ventajas y desventajas, permitiendo así tomar una decisión sobre la mejor opción para el desarrollo de la aplicación.</p> <p>La segunda etapa involucra el desarrollo del proyecto, el modelado UML, la documentación, el diseño y la implementación de la base de datos, el diseño de la interfaz y el diseño completo del prototipo con todos los requerimientos necesarios. La tercera etapa involucra la verificación del correcto funcionamiento del prototipo con las pruebas y con los requerimientos necesarios.</p> <p>El proyecto busca facilitar a los administradores de redes conocer el estado de los servidores que están bajo su administración, para así tener una información fiable en cualquier momento. Esta información es importante para cumplir sus labores diarias, pero en ocasiones el administrador no se encuentra en su lugar de trabajo o con acceso a una terminal adecuada para monitorear el estado de los servidores. Con este proyecto se busca facilitar el acceso a esa información por medio de dispositivos móviles como lo son los celulares, PDAs o cualquier dispositivo que pueda tener acceso a un red celular por medio de el protocolo WAP.</p>

## FUENTES BIBLIOGRÁFICAS

- ACEVEDO Marco A, SANCHEZ Miguel, TOLEDO Carlos, Monitoreo en tiempo real, con el protocolo WAP, México, 2002.
- AREHART Charles, PROFESIONAL WAP with WML, WMLScript, ASP, JSP, XML, XSLT, WTA Push and VoiceXML, Estados Unidos, 2000.
- AREHART Charles, CHIDAMBARAM Nirmal, GURUPRASAD Shashikiran. PROFESIONAL WAP with WML, WMLScript, ASP, JSP, XML, XSLT, WTA Push and VoiceXML, Estados Unidos, 2000.
- BELLALTA Boris, RINCÓN David, OLIVER Miquel, Análisis de Tráfico y Capacidad para Servicios WAP sobre Redes GSM/GPRS, España, 2000.
- FREIER, A. O., KARLTON, P. y KOCHER, P. C. (1996). The SSL Protocol Specification 3.0. Disponible en <http://www.wmlclub.com/ind=http://home.netscape.com/>
- FORTA Ben, Keith Lauver, Paul Fonte, Robert M. Juncker, Ronan Mandel and Dylan Bromby. WAP Development with WML and WMLScript, United States, 2000
- GROSSO, William, Java RMI, United States, 2004 MONJAS Miguel Angel. Seguridad en WAP II. Disponible en <http://www.wmlclub.com/articulos/seguridadwap.htm>
- LETELIER, Patricio, “Metodología y Tecnología de la Programación”, Servicio de Publicaciones Universidad Politécnica de Valencia, 269 paginas, Año 1997.
- MARTÍNEZ, Normand, “Introducción a la ingeniería del software” Delta Publicaciones so Amo, Loïc , 542 páginas, Año 2005.
- MORELO Xavier, Monitorización de sistemas y programas, España, 2004.
- NOKIA. Nokia WAP Toolkit 2.1 User’s Guide, Product Number SDK-01-000-001, 2001. 99
- NOKIA. Nokia WAP Toolkit 2.1 Developer’s Guide, Product Number SDK-01-000-002, 2001.
- OAKS, Scott. Java Security, United States, 2001
- PEREZ Elias, Administración de sistemas operativos, Mexico, 2001.
- SETCo. SET Secure Electronic Transaction Specification. Book 3: Formal Protocol Definition. Version 1.0. Disponible en [http://www.setco.org/download/set\\_bk3.pdf](http://www.setco.org/download/set_bk3.pdf)
- TANENBAUM, Andrew S, “Redes de Computadoras” Pearson Educación, 891 páginas, Año 2003.
- TANENBAUM Andrew. VAN STEEN Maarten, Distributed Systems: Principles and Paradigms Prentice Hall, 2002
- VALLESPÍ Albert, OLIVER Miquel, Personalización de servicios WAP mediante gestión de contenidos XML y bases de datos relacionales, España, 2000.
- WAP Forum. Wireless Application Protocol Architecture Specification. URL: <http://www.wapforum.org/what/technical.htm> (Octubre 2001).

## WEBGRAFÍA

- Consulta en las páginas web listadas a continuación
- <http://www.dosideas.com/wiki/JUnit>
- <http://www.elcodigo.net/cgi-bin/DBread.cgi?tabla=herramientas&=0&clave=46&info=1>
- [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/msp/aldrette\\_m\\_r/capitulo1.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/msp/aldrette_m_r/capitulo1.pdf)
- <http://www.chuidiang.com/java/rmi/rmi.php>
- <http://proactiva-calidad.com/java/rmi/introduccion.html>
- [http://es.wikipedia.org/wiki/Java\\_Remote\\_Method\\_Invocation](http://es.wikipedia.org/wiki/Java_Remote_Method_Invocation)
- <http://www.wapforum.org/faqs/index.htm>
- <http://www.wapforum.org/what/benefits.htm>
- <http://www.nokia.com/corporate/wap/future.html>
- <http://www.phone.com/industry/wap.html>
- <http://www.ericsson.com/WAP/what/what.shtml>
- <http://www.wap.com/findwap.html>
- <http://www.elviajero.org/antoniux/tutos/servidores8.pdf>
- [http://aprender21.com/cursolinux.php?cpg=adw\\_erirblnx&gclid=CMrq5YbTvKECFROG5godUCAC9w#](http://aprender21.com/cursolinux.php?cpg=adw_erirblnx&gclid=CMrq5YbTvKECFROG5godUCAC9w#)
- [http://java.com/es/download/faq/whatis\\_j2me.xml](http://java.com/es/download/faq/whatis_j2me.xml)

NÚMERO RAE	
PROGRAMA	Ingeniería de Sistemas

CONTENIDOS	
1. PLANTEAMIENTO DEL PROBLEMA	<p>Se detallan los antecedentes de temas importantes tales como la evolución de la telefonía móvil en Colombia y una breve descripción de herramientas similares. A partir de estos antecedentes se puede formular la siguiente pregunta de investigación ¿Cómo diseñar y que características deberá tener, un sistema de gestión móvil para monitorear las funciones básicas en servidores Linux?</p> <p>En la actualidad, la infraestructura de redes es un área muy importante para la mayoría de las empresas. No disponer de una red puede causar pérdidas financieras inevitables. El administrador de redes debe poder controlar la actividad en la red y llamar a los técnicos rápidamente en caso de congestión o problemas de acceso. También está a cargo de administrar las cuentas de los usuarios, crear cuentas para nuevos miembros del personal y eliminarlas cuando éstos ya no pertenecen a la compañía.</p> <p>Debido a los cambios vertiginosos de la tecnología y los medios de transmisión, el administrador de redes debe estar alerta a cualquier novedad que presenten los sistemas administrados para ello deberá contar con una herramienta que esté disponible 7 * 24 y que le permita acceder inmediatamente para verificar y gestionar para continuar garantizándole a la organización la disponibilidad, confiabilidad y fiabilidad de la información.</p> <p>Objetivo general</p> <p>Diseñar e implementar el prototipo de herramienta y manejo de comandos para el monitoreo de servidores GNU/LINUX por medio de dispositivos móviles.</p> <p>Objetivos específicos</p> <ul style="list-style-type: none"> <li>• Analizar el monitoreo remoto de un servidor GNU/LINUX.</li> <li>• Diseñar el software de monitoreo remoto de un servidor GNU/LINUX.</li> <li>• Diseñar la interfaz de usuario para ser visualizada en los dispositivos móviles a través de WAP.</li> <li>• Implementar el prototipo funcional para la administración de sistemas GNU/LINUX.</li> <li>• Realizar pruebas de funcionamiento unitario del prototipo implementado para verificar su correcto funcionamiento</li> </ul>

El proyecto culmina con el desarrollo e implementación de un prototipo de sistema de monitoreo remoto de un servidor GNU/LINUX por medio de dispositivos móviles que obtendrá valores del estado del sistema y de los servicios que preste, la consulta se realizará a través de WAP y la información será mostrará en paginas WML.

## 2. METODOLOGÍA

El método de desarrollo aplicado es el Modelo Cascada; basado en el ciclo convencional de desarrollo de software

## 3. LÍNEA DE INVESTIGACIÓN

Tecnologías actuales y sociedad - Tecnologías de Información y Comunicación (TIC) - NÚCLEO PROBLÉMICO - Construcción de software.

## 4. MARCO DE REFERENCIA

Telefonía móvil celular

El sistema de telefonía celular ha pasado por varias generaciones: 1G (Sistema de telefonía celular de primera generación), 2G (Sistema de telefonía celular de segunda generación, conocido generalmente como GSM), 2.5G (Sistema de telefonía móvil de segunda generación avanzado, conocido como GPRS), 3G (Sistema de telefonía celular de tercera generación).

Respecto a los servicios del mundo móvil, WAP será indispensable en el desarrollo del proyecto, la tecnología WAP es un estándar impulsado por la industria del sector de las telecomunicaciones con el objetivo de proporcionar un sistema avanzado de servicios de Internet para dispositivos móviles.

Los dispositivos móviles abren la puerta a un nuevo mundo de posibilidades para las comunicaciones inalámbricas. La tecnología WAP establece un puente mucho más sólido entre el mundo de la telefonía móvil e Internet ofreciendo la capacidad para vehicular un amplio abanico de servicio de valor añadido con total independencia de la tecnología que se utilice.

WAP - Protocolo de aplicaciones inalámbricas

WAP (Wireless Application Protocol o Protocolo de Aplicaciones Inalámbricas) es una solución unificada para los servicios de valor agregado existentes y futuros para la telefonía móvil. El protocolo incluye especificaciones para las capas de sesión y de transporte del modelo OSI, así como funcionalidades de seguridad.



#### WML (Wireless Markup Language)

El lenguaje WML (Wireless Markup Language) constituye la base para la creación de contenidos visualizables desde un terminal WAP (un terminal dotado con un micro-navegador WAP).

Las páginas WML sólo permiten introducir texto y gráficos, debido a las propias capacidades de los terminales móviles. Por otra parte, los micro-navegadores de los terminales también soportan un lenguaje de script, llamado WMLScript, que permite manipular datos y dotar de cierta interactividad a la página WML.

Por otra parte, los gráficos que pueden ser introducidos en una página WML deberán tener un formato especial, denominado WBMP. Este formato sólo permite imágenes en blanco y negro. Las páginas WML podrán ser ubicadas en cualquier servidor web convencional, siempre que éste haya sido correctamente configurado para servirlos

#### UML (Unified Modeling Language)

El lenguaje unificado de modelado (UML), es un lenguaje no propietario para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. UML proporciona una forma sencilla de definir y transmitir ideas complejas; por medio de diagramas e iconografías.

UML es manejado y controlado por un consorcio (el Object Management Group OMG), el cual está conformado principalmente por grandes compañías de la industria de la computación. A través de este consorcio se controla la forma en cómo UML evoluciona mediante el consenso de la junta de directores, con la finalidad de crear y mantener una especificación única. La aplicación de UML, es muy extensa, ya que puede ser utilizado por personas con o sin conocimientos técnicos, para realizar modelos físicos y conceptuales de diversa índole.

Los bloques de construcción se dividen en tres partes:

- Elementos: Son las abstracciones de primer nivel.
- Relaciones: Unen a los elementos entre sí.
- Diagrama: Son agrupaciones interesantes de elementos.

## Diagramas

Los diagramas se utilizan para representar diferentes perspectivas de un sistema de forma que un diagrama es una proyección del mismo. UML proporciona un amplio conjunto de diagramas que normalmente se usan en pequeños subconjuntos para poder representar las cinco vistas principales de la arquitectura de un sistema.

## HSQLDB (Hyperthreaded Structured Query Language Database)

El HSQLDB (del inglés Hyperthreaded Structured Query Language Database) es un servidor de banco de datos (SGBD), de código abierto, escrito totalmente en el lenguaje Java. No es posible compararlo, en términos de robustez y seguridad con otros servidores SGBD, como Oracle o Microsoft SQL Server, sin embargo el HSQLDB es una solución simple, que utiliza pocos recursos y que posee buen desempeño. Debido a esas características, él es bastante utilizado en aplicaciones que son ejecutadas en desktops y que necesitan integrar con una capa de persistencia a través del lenguaje SQL. El HSQLDB se ve aplicado dentro del proyecto, en la base de datos elaborada para administrar el almacenamiento y control de los usuarios y equipos a monitorear.

## RMI (Remote Method Invocation)

Las aplicaciones RMI normalmente comprenden dos programas separados: un servidor y un cliente. Una aplicación servidor típica crea un montón de objetos remotos, hace accesibles unas referencias a dichos objetos remotos, y espera a que los clientes llamen a estos métodos u objetos remotos. Una aplicación cliente típica obtiene una referencia remota de uno o más objetos remotos en el servidor y llama a sus métodos. RMI proporciona el mecanismo por el que se comunican y se pasan información del cliente al servidor y viceversa. Cuando es una aplicación, algunas veces se hace referencia a ella como Aplicación de Objetos Distribuidos.

## Servidores LINUX

Los Servidores Linux se gestionan mediante SSH, VNC y/o Panel de Control; mientras que los Servidores Windows se gestionan mediante Terminal Server, VNC y/o Panel de Control. En cuanto a funciones los Servidores Linux no se llevan muy bien con ASP (Active Server Pages), y no pueden ejecutar MS SQL Server o MS Exchange. En cuanto a las demás aplicaciones/servicios, tanto Linux como Windows pueden realizar las mismas funciones, siendo Linux, normalmente, el preferido para ejecutar sistemas basados en PHP/MySQL.

Los servidores dedicados Linux serán siempre más económicos que el Windows. La principal razón de ello es que existe una gran comunidad OpenSource (código abierto) y aplicaciones gratuitas. Linux, normalmente, no requiere de licencias del proveedor. En el caso de Windows, si queremos ejecutar un MS SQL Server o Exchange, deberemos tener en cuenta que supone un costo adicional.

La shell, es el intérprete de comandos de la mayoría de los sistemas Unix o similares a Unix. Un script es un guión, es decir una serie de instrucciones para la consola guardada en un fichero de texto de extensión: .sh. En Gnu/Linux, esta técnica alcanza el nivel de un lenguaje de programación. Para los administradores de Servidores, los scripts son herramientas muy útiles. Permiten hacer tareas de una forma muy productiva, ejecutar aplicaciones cuando se desee, mediante el demonio crontab, etc. Incluso el sistema operativo se inicia gracias a un script (initd).

La implementación de la herramienta de monitoreo se entiende como un servicio de valor agregado, dirigido a un grupo cerrado de usuarios, puesto que para su correcto funcionamiento esta debe ser usada a través de una red de telecomunicaciones.

Los decretos en Colombia que regulan el uso de la aplicación son:

Ministerio de Comunicaciones Decreto número 600 (14 mar. 2003) por medio del cual se expiden normas sobre los servicios de valor agregado y Telemáticos, y se reglamenta el Decreto Ley 1900 de 1990.

Decreto número 3055 de 2003 de octubre de 2003 por medio del cual se modifica el decreto 600 de 2003.

Decreto número 1360 de 1989. Por el cual se reglamenta la inscripción del soporte lógico (software) en el Registro Nacional del Derecho de Autor.

La ley 37 de enero de 1993. La prestación del servicio de Telefonía Móvil Celular (TMC) en Colombia, se aprobó por el Congreso de la República mediante De acuerdo con esta Ley, se permite al Ministerio de Comunicaciones adjudicar la concesión de la prestación del servicio de TMC, previo un trámite de licitación pública.

El estándar IEEE 830 o IEEE Recommended Practice for Software Requirements Specifications describe una serie de recomendaciones prácticas para la creación de una buena especificación de requisitos de software (ERS/SRS).

## 5. DESARROLLO INGENIERIL

El proceso que se llevó a cabo para llegar a la solución del problema planteado se enmarca en la interacción de las etapas descritas a continuación, y que dan alcance al método de desarrollo aplicado: Modelo Cascada.

El análisis está compuesto por las etapas de análisis e ingeniería del sistema y el análisis de requisitos del modelo de cascada.

### ANÁLISIS

Esta etapa permite recolectar y detallar las especificaciones de los requerimientos del prototipo a desarrollar. El análisis se detalla en cada uno de los siguientes apartes:

Especificación de requerimientos – Propósito – Ámbito – Referencias - Descripción General - Perspectiva del prototipo - Funciones del prototipo - Requerimientos específicos funcionales y no funcionales - Vista de casos de uso.

### Agente

El agente estará instalado en los servidores a monitorear y su función será la de recolectar la información en tiempo real del servidor.

### Servidor

El servidor se encargará de procesar la información recolectada por los agentes para luego ser mostrada al usuario vía web a través de un dispositivo móvil.

### Interfaz de conexión

La interfaz de conexión se encargará de hacer un puente de comunicación segura entre el agente y el servidor; garantizando la integridad, fiabilidad y seguridad de los datos.

## DISEÑO

Teniendo en cuenta el levantamiento de información y los requerimientos establecidos en la etapa de análisis, se obtuvieron los datos suficientes para llegar al siguiente diseño, el sistema contará con tres componentes principales: un agente, un servidor y una interfaz segura de conexión

### Diseño del agente

El agente es el encargado de recolectar la información a monitorear; esto lo realizará dependiendo de las peticiones que se le realice. El agente se encargará de recolectar dos grandes tipos de información: básica y extra.

### Diseño interfaz conexión

La interfaz de conexión es el componente que ayuda a la comunicación entre el servidor y el agente. El servidor debe solicitar los datos al agente y este a su vez debe retornar la información. Este componente sera construido con RMI. RMI es la técnica que Java utiliza para obtener una distribución transparente de los objetos; es decir, objetos distribuidos de manera que el cliente no tiene conocimiento de donde reside el objeto.

### Diseño servidor

El servidor es el encargado de recibir las peticiones de los usuarios a través de los dispositivos móviles, y a su vez realizar la petición al agente (por medio de la interfaz de conexión descrita anteriormente). Cuando este le responde, el se encarga de organizar la información y traducirla a lenguaje WML para ser mostrada a los usuarios. El servidor posee dos componentes: un servidor WAP y un procesador de información. El servidor WAP es el punto de acceso a los usuarios. Este se encarga de recibir las peticiones y dar respuesta a las mismas. El procesador de información se encarga de traducir la información entregada por el agente, y codificarla en WML para ser mostrada en los dispositivos móviles.

## IMPLEMENTACIÓN

Se procede a realizar la construcción del software de acuerdo al diseño descrito anteriormente, utilizando la tecnología Java en nuestro proyecto, ofrecemos una gran flexibilidad para cubrir los objetivos y requerimientos planteados. Algunas de las características que impulsaron el uso de Java en nuestro proyecto son las siguientes:

- Modelo de distribución de objetos de forma transparente (RMI).
- Soporte para redes TCP/IP
- Soporte de seguridad propio del lenguaje.
- Programación orientada a objetos madura.
- Utilización de librerías externas que expanden las capacidades del lenguaje.

Después de analizar las ventajas y desventajas que tiene cada plataforma se optó por implementar con WML, y respecto al IDE se seleccionó NetBeans que es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word.

## 6 PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

Una vez realizada la implementación del prototipo, se ejecutaron las pruebas necesarias para detectar fallos y componentes defectuosos, así mismo probar el flujo de operación del prototipo. Las etapas de las pruebas fueron:

Planificación de la prueba

Preparación del escenario

Pruebas de instalación y configuración

Pruebas de ejecución

Resultados de las pruebas

METODOLOGÍA	<p>El método de desarrollo aplicado es el Modelo Cascada; basado en el ciclo convencional de desarrollo de software, en el cual se establecen los objetivos globales y después se produce un diseño rápido que se enfoca en la representación de los aspectos visibles al usuario, el diseño rápido conduce a la construcción de un prototipo, el mismo que es evaluado por el usuario y se utiliza para refinar el cumplimiento de los objetivos específicos. Los flujos fundamentales como modelado de negocio, captura de requisitos, análisis, diseño, implementación, pruebas y despliegue y concluye con una versión más elaborada del producto.</p>
CONCLUSIONES	<ul style="list-style-type: none"> <li>• El administrador de redes tiene a cargo una de las labores más importantes de gestión, es el monitoreo de la red y sus componentes, algunos procesos usuales son monitorear el consumo de CPU y Memoria (RAM y SWAP); cada cierto tiempo es necesario que entremos a revisar cuando espacio en disco hay libre en los servidores que administra.</li> <li>• Un sistema administrador de redes debe estar orientado a permitir mayor control sobre la red, en este caso de estudio sobre el servidor proporcionando herramientas que faciliten las labores diarias.</li> <li>• Una de las ventajas de trabajar con Java es que es compatible con la mayoría de móviles disponibles en el mercado y para dispositivos con capacidades limitadas, En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.</li> <li>• El software desarrollado se encarga de realizar acciones en el cliente como solicitar datos, mostrar mensajes, herramienta que cobra valor agregado al momento de tomar decisiones.</li> <li>• El diseño de las interfaces a través de WAP permitió alojar páginas para ser enviadas a través de la Web y ser visualizadas en los navegadores de los dispositivos móviles, aunque WAP trabaja bajo restricciones de memoria, esta limitación no afectó el cumplimiento del objetivo planteado.</li> </ul>

- Dichas interfaces se encargaron de hacer el puente de comunicación segura a través de autenticación entre el agente y el servidor, garantizando en todo momento la integridad, fiabilidad y seguridad de datos.
- La implementación del prototipo cumplió con el aseguramiento de consultas de las variables básicas para el monitoreo de servidores LINUX, a través de dispositivos móviles
- Las pruebas ejecutadas al software, permiten asegurar la calidad de cada uno de los elementos que lo componen y así poder establecer los tiempos de respuesta.

