

Monitoreo de Variables Ambientales y Control de Alarmas para el Sistema ARIADNA mediante un Dispositivo Móvil



Jairo Eladio Jojoa Cerón
John Jairo Ortiz Canchala

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

Director: Ing. Javier Alexander Hurtado

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, Octubre de 2005

TABLA DE CONTENIDO

INTRODUCCION

1. PLATAFORMAS Y ENTORNOS DE DESARROLLO.....	5
1.1 MODELOS DE COMUNICACIÓN ENTRE EL MÓVIL Y LAS APLICACIONES	5
1.1.1 Tecnología Pull	6
1.1.2 Tecnología Push.....	6
1.2 MODELOS O MÉTODOS DE ACCESO A LOS RECURSOS DE LA RED INALÁMBRICA	7
1.2.1 Servicios WAP.....	7
1.2.2 Servicios Basados en Mensajería.....	9
1.2.2.1 SMS	9
1.2.2.2 Arquitectura y Elementos de Red en SMS.....	10
1.2.2.3 Servicios Basados en SMS.....	12
1.2.3 Servicios Push.....	14
1.3 AMBIENTES Y PLATAFORMAS DE DESARROLLO EN MENSAJERIA	15
1.3.1 J2ME	15
1.3.1.1 Wireless Messaging API (WMA)	16
1.3.1.2 Clases WMA de Nivel Superior	16
1.3.2 J2ME Y PUSH	20
1.3.2.1 Activación de una MIDLet	20
1.3.3 WAP Push	22
1.3.3.1 Push Access Protocol - Generalidades.....	26
1.3.4 OSA (OPEN SERVICE ACCESS)	30
1.3.4.1 API PARLAY / OSA.....	32
1.3.4.2 Arquitectura lógica de red PARLAY / OSA.....	33
1.3.4.3 El papel del FRAMEWORK PARLAY / OSA	34
1.3.4.4 Ciclo de vida de las aplicaciones	35
1.3.4.5 Service Level Agreements (SLA)	36
1.3.4.6 Acceso a los Servicios	38
1.3.4.7 HOSA.....	39
1.4 HERRAMIENTAS O PLATAFORMAS ELEGIDAS Y JUSTIFICACIÓN	40
2. ANÁLISIS Y DISEÑO DEL SISTEMA DE MONITOREO DE VARIABLES AMBIENTALES Y CONTROL DE ALARMAS PARA EL SISTEMA ARIADNA MEDIANTE UN DISPOSITIVO MÓVIL.....	42

2.1	DEFINICIÓN DE REQUERIMIENTOS	42
2.1.1	Descripción del Problema.....	42
2.1.2	Definición de Requerimientos	43
2.2	DESCRIPCION GENERAL DEL SISTEMA	44
2.3	ÁRBOL DE FUNCIONES.....	45
2.4	CASOS DE USO DE ALTO NIVEL SISTEMA MOVAYCAMM	46
2.4.1	Descripción de los Casos de Uso de Alto Nivel.....	46
2.4.1.1	C.U. Gestión de Usuarios	47
2.4.1.2	C.U. Gestión del Sistema Movaycamm	48
2.4.1.3	C.U. Modificar Parámetros.....	48
2.4.1.4	C.U. Acceder al Sistema.....	49
2.4.1.5	C.U. Confirmar Recepción de Alarmas	49
2.4.1.6	C.U. Consultar Información.....	49
2.4.1.7	C.U. Consultar información Menú de voz	50
2.4.1.8	C.U. Registrar Datos.....	50
2.5	DIAGRAMA DE CLASES DEL SISTEMA	51
2.5.1	Diagrama de clases principales del sistema Movaycamm	51
2.5.1.1	Clase Main.....	52
2.5.1.2	Clase Feature.	52
2.5.1.3	Clase SMSProcessor.....	54
2.5.1.4	Clase MPCCProcessor.....	55
2.5.1.5	Clase UIProcessor.....	55
2.5.1.6	Clase AccesoBD.....	56
2.5.1.7	Clase Solicitud	56
2.6	DIAGRAMA DE PAQUETES SISTEMA MOVAYCAMM	57
3.	PRUEBAS Y RESULTADOS.....	59
3.1	CONFIGURACIÓN DEL SERVICIO	60
3.2	NOTIFICACIÓN DE ALARMAS	61
3.3	CONSULTAS.....	64
3.3.1	Consultas mediante solicitudes SMS.....	64
3.3.2	Consultas mediante llamada telefónica al servicio Movaycamm.....	66
3.4	MODIFICACIONES	69
3.4.1	Modificaciones mediante solicitudes SMS.....	69
3.4.2	Modificaciones mediante llamada telefónica al servicio Movaycamm	71
4.	CONCLUSIONES Y RECOMENDACIONES	75
	GLOSARIO	78
	BIBLIOGRAFIA	77
	ANEXO A. Generalidades de las Tecnologías para Telecomunicaciones Móviles y Servicios de Valor Agregado	
	ANEXO B. Análisis y Diseño	
	ANEXO C. Manual de Usuario	

LISTA DE FIGURAS

Figura 1.1	Modelo Tradicional “Pull” Solicitud – Respuesta.....	2
Figura 1.2	Elementos Generales en un servicio WAP.....	4
Figura 1.3	Entidades de red involucradas en el servicio SMS.....	6
Figura 1.4	Componentes Principales en los servicios de Mensajería.....	9
Figura 1.5	Componentes Principales en los servicios Push.....	10
Figura 1.6	Interfaces WMA de nivel superior en el paquete javax.wireless. messaging..	13
Figura 1.7	Alerta en las acciones con WAP Push.....	18
Figura 1.8	Framework WAP Push.....	19
Figura 1.9	Arquitectura y descripción de OSA.....	26
Figura 1.10	Arquitectura lógica de red PARLAY / OSA.....	29
Figura 1.11	Entidades Lógicas Parlay / Osa.....	30
Figura 1.12	Framework.....	33
Figura 2.1	Arquitectura del sistema ARIADNA Móvil.....	40
Figura 2.2	Diagrama de casos de uso de alto nivel Sistema Movaycamm.....	42
Figura 2.3	Diagrama de clases principales del sistema Movaycamm.....	47
Figura 2.4	Interfaz que despliega la clase Main.....	48
Figura 2.5	Diagrama de paquetes Sistema Movaycamm.....	53
Figura 3.1	Resultados de la prueba de configuración del servicio.....	57
Figura 3.2	Mensajes de alarma enviados al número principal y de reenvío.....	59
Figura 3.3	Envío del mensaje de confirmación.....	60
Figura 3.4	Envío del mensaje de consulta y recepción del SMS de respuesta.....	61
Figura 3.5	Consulta mediante llamada telefónica al servicio Movaycamm.....	65
Figura 3.6	Consulta tiempo de lectura.....	66
Figura 3.7	Modificación tiempo de lectura.....	67
Figura 3.8	Modificación de tiempo de lectura mediante llamada telefónica.....	69

***A Dios
A mi Padre por su incondicional apoyo
A mi Madre por su confianza
A mis Hermanos por ser fuente de motivación
y optimismo***

John Jairo Ortiz Canchala

***A mis Padres, A mi Esposa
A mis Hijos
Por Creer en mí***

Jairo Eladio Jojoa Cerón

INTRODUCCIÓN

En los últimos años el desarrollo de las telecomunicaciones inalámbricas ha sido muy importante, de igual manera el desarrollo de servicios de valor agregado que hagan uso de esas tecnologías ha crecido y sigue creciendo a un ritmo vertiginoso. Las tendencias actuales son procurar llevar todo tipo de procesos, soluciones, aplicaciones y servicios al entorno móvil, en estos procesos existirán varios casos que se podrán hacer con relativa facilidad, otros que serán mas complicados y otros que definitivamente será imposible volverlos móviles, dentro de esa gama de necesidades encontramos el tema de las alarmas generadas por cualquier tipo de sistemas, la necesidad es que dichas alarmas sean notificadas en el momento preciso a la persona indicada en cualquier lugar, este “cualquier lugar” ya implica movilidad, por lo tanto es una aplicación que es factible darle una solución de tipo móvil. Estos sistemas cuyas interfaces de administración tienen un registro de eventos o anomalías lo importante es conocerlos en el mismo momento en que se presentan, reportándolos como notificaciones de alarmas, esto es lo fundamental, las medidas a tomar en los casos de alarmas o posteriores consultas o modificaciones en el sistema son cosas posteriores, pero en términos generales la eficiencia del sistema mejora ya que los tiempos de respuesta en la gestión de fallos se reducen de manera significativa.

En la Universidad del Cauca se encuentra en ejecución el proyecto ARIADNA, el cual tiene un Sistema de Información ambiental para detección de alertas tempranas, este sistema tiene las características que se menciono anteriormente, monitoriza permanentemente algunos sensores e informa las anomalías que se presentan, esto se hace por medio de una consola de administración. Actualmente la mayoría de sistemas que reportan alarmas lo hacen a través de dichas consolas de gestión, que debe necesariamente estar bajo la supervisión de una persona encargada. El hecho de que esta persona deba estar ligada a un lugar específico provoca un proceso ineficiente ya que las situaciones de alarma no son constantes pero al mismo tiempo son impredecibles,

para lograr eficiencia de estos sistemas lo ideal es notificar cualquier alarma al encargado del sistema pero sin que el este en un lugar determinado, entregar las alarmas independientemente del lugar donde se encuentre el responsable del sistema, y estas soluciones con características únicas solo las provee las comunicaciones móviles.

El presente proyecto tiene como objetivo principal el diseño e implementación de un prototipo de aplicación telemática que trabaje como un servicio de valor agregado para una red de telefonía móvil, y que interactúa con el Sistema de información ambiental para la detección de alarmas tempranas del sistema Ariadna. Este servicio le permitirá a unos usuarios móviles (por medio de un teléfono celular) consultar información y realizar la gestión de algunos parámetros específicos del sistema Ariadna por medio de su teléfono móvil, y lo mas importante es que le permite al usuario recibir las alarmas que produce el sistema Ariadna directamente en su teléfono móvil independientemente del lugar en el que se encuentre el usuario, obviamente teniendo en cuenta que el teléfono móvil se encuentra activo y dentro del área de cobertura de la red de telefonía móvil. Con la implementación del servicio descrito se pretende lograr un mayor grado de eficiencia en el monitoreo y control de las alarmas que se producen en el sistema Ariadna mediante la notificación instantánea y automática de dichos eventos sin que haya necesidad de estar al frente de una consola de monitoreo y gestión.

Desde el punto de vista técnico y a nivel investigativo, el presente trabajo analiza algunas tecnologías y entornos de desarrollo con los que se pueden desarrollar aplicaciones para las características y necesidades de nuestro entorno, y teniendo en cuenta el factor tecnológico y económico que demanda el desarrollo de un determinado servicio, ya que en muchas ocasiones pueden existir soluciones viables técnica y / o tecnológicamente pero poco aplicables a nivel económico y de presupuesto.

El presente documento se ha dividido en cuatro capítulos: El primero describe las principales tecnologías móviles que están operando a nivel mundial, haciendo principal énfasis a aquellas que se han implementado en nuestro país y con las cuales los operadores móviles locales ofrecen además del servicio de voz, sus servicios de valor agregado. El segundo trata sobre las plataformas y entornos de desarrollo mas apropiadas de acuerdo a nuestro entorno para poder desarrollar nuevos servicios de valor

agregado, se hace un balance sobre ellas y se explica con cual de ellas se implementa el servicio que propone este trabajo. El capítulo tres hace referencia a los procesos de análisis y diseño del software, el cual se apoya en el Lenguaje de Modelado Unificado (UML). Finalmente en el capítulo cuatro se recopila el proceso de pruebas del servicio desarrollado y se realizan algunas observaciones y recomendaciones que puedan ser útiles para trabajos posteriores.

1. PLATAFORMAS Y ENTORNOS DE DESARROLLO

En este apartado se pretende mostrar el resultado de la investigación sobre las herramientas más propicias para el desarrollo de servicios de valor agregado que son aplicables a nuestro entorno, teniendo en cuenta las observaciones presentadas en el anterior capítulo. Entre las diversas aplicaciones para dispositivos móviles que existen el caso específico de este trabajo es el monitoreo y control de alarmas, por lo tanto se debe escoger las tecnologías más acordes para este tipo de soluciones. Partiendo de este punto se ha analizado los métodos de transmisión de datos que tienen las redes móviles de nuestro país y se ha encontrado que el acceso a Internet y la descarga de archivos se hace mediante el servicio Wap y el envío de información de bajas capacidades se realiza mediante el servicio de mensajería corta SMS. Los anteriores servicios están soportados por una amplia gama de teléfonos móviles y son sencillos de utilizar por parte de usuario. Actualmente en nuestro país todavía no se ha presentado una masificación en el uso de estos servicios pero si se tiene en cuenta el entorno global, en donde se han desarrollado varios tipos de servicios teniendo como base esas tecnologías, se puede esperar que su utilización como servicios finales y como soporte para otros servicios se incremente en poco tiempo. Esto depende del tipo de servicios que se desarrollen, la facilidad de uso y la utilidad que ellos representen para los usuarios finales.

1.1 MODELOS DE COMUNICACIÓN ENTRE EL MÓVIL Y LAS APLICACIONES

El modelo que se ha difundido ampliamente para el desarrollo de aplicaciones y servicios móviles es el modelo Cliente / servidor, en el que se da una cooperación entre el cliente que solicita un servicio o un recurso y el servidor que ejecuta el servicio o proporciona el recurso. Dependiendo de quien inicie el proceso de comunicación en este modelo se presentan dos tipos de tecnologías: la tecnología Pull y la tecnología Push.

1.1.1 Tecnología Pull

En el modelo cliente / servidor normal, un cliente solicita un servicio o una información de un servidor, que entonces responde transmitiendo la información al cliente. Esto se conoce como tecnología "pull". El cliente "trae" la información del servidor. El World Wide Web es el mejor ejemplo de la tecnología pull, donde un usuario ingresa una URL (la petición), que se envía a un servidor, y el servidor contesta enviando una página Web o Wap (la respuesta) al usuario (Figura 1.1).

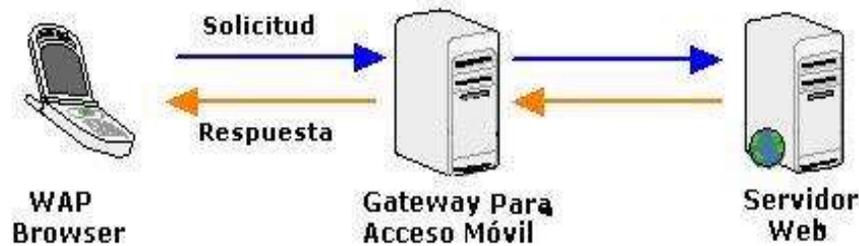


Figura 1.1. Modelo Tradicional "Pull" Solicitud - Respuesta

1.1.2 Tecnología Push

En esta tecnología quien inicia el proceso de comunicación es el servidor a través de un Push. Un Push es un mensaje enviado a un suscriptor móvil sin una petición explícita del suscriptor cuando se entrega el mensaje. En términos generales, estos mensajes podrían ser persona a persona, aplicación a persona o incluso aplicación a aplicación. Algunos ejemplos de las aplicaciones Push que existen actualmente incluyen los mensajes de alertas para las noticias, la cotización de las acciones en la bolsa de valores y los horóscopos diarios. Los usuarios se suscriben a estos servicios y reciben alarmas según sus preferencias. Las aplicaciones predominantes de Push actualmente son las alarmas simples. Estas aplicaciones se conocen como Servicios Push (*Push Services*), este termino describe contenido que es enviado o "empujado" desde un servidor directamente hacia el terminal del suscriptor. El contenido puede ser enviado sobre diferentes tipos de portadores como SMS, MMS e e-mail.

Los servicios Push ofrecen ventajas significativas sobre el método “Pull” usado por muchas aplicaciones móviles. Aquí la aplicación cliente revisa el servidor para buscar contenido nuevo para ser descargado al terminal. En el proceso de revisión el servidor consume ancho de banda independientemente si se descarga o no algún tipo de información. Con los servicios push sin embargo, los nuevos contenidos se pueden enviar automáticamente.

Cuando las aplicaciones móviles necesitan sincronizar información, usualmente se envía una notificación SMS, invitando al usuario a bajar la información manualmente. Sin embargo la presencia de información nueva puede ser detectada, y se puede indicar que las actualizaciones se ejecuten automáticamente en segundo plano, proporcionando al usuario una experiencia “*always-on*”. Los servicios push pueden necesitar de una gateway push, ésta actúa como puente entre las redes móviles y la Intranet o Internet.

1.2 MODELOS O MÉTODOS DE ACCESO A LOS RECURSOS DE LA RED INALÁMBRICA

Cuando se desarrollan aplicaciones o servicios para telefonía móvil, dependiendo de la cantidad, tipo de información que se va a manejar y la forma como va a ser entregada al usuario final, ya sea bajo el modelo pull o push, se deben tener en cuenta todos los recursos (infraestructura) que van a ser necesarios. Estos recursos están en medio del servidor de aplicaciones (donde se hospeda el servicio) y el terminal móvil del usuario, y pertenecen a la infraestructura de Internet y del operador inalámbrico. A continuación se presenta los principales tipos de servicios que pueden ser implementados en Colombia, los recursos necesarios para ellos y se examina si el desarrollador de aplicaciones y servicios móviles debe acceder directamente alguno de ellos.

1.2.1 Servicios WAP

Es uno de los servicios más comunes y fáciles de implementar desde el punto de vista del desarrollador, ya que el esquema de desarrollo es muy similar al desarrollo de

aplicaciones web, obviamente teniendo en cuenta las limitaciones de los dispositivos a los cuales esta dirigido el servicio. Están basados en el esquema cliente (el terminal móvil) – servidor por lo tanto la entrega de la información se realiza con tecnologías Pull. En la Figura 1.2 se muestra los recursos que intervienen en estas aplicaciones.

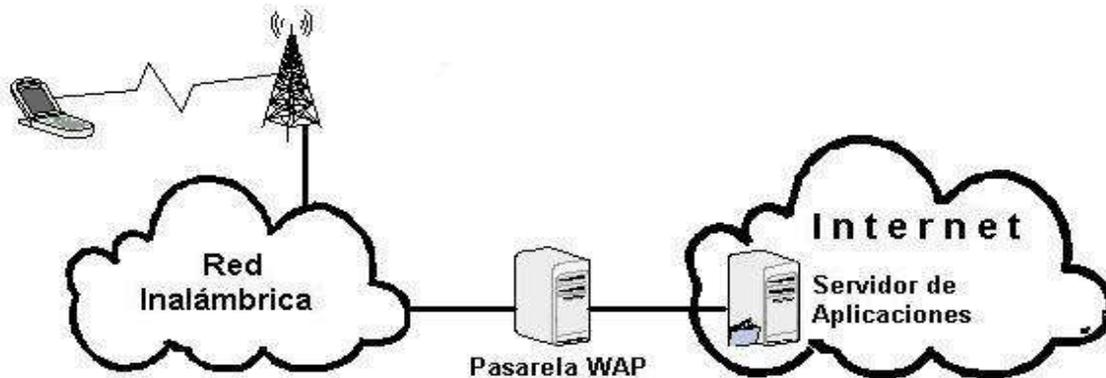


Figura 1.2 Elementos Generales en un servicio WAP

El modelo WAP interpone una pasarela WAP entre los servidores web tradicionales y los clientes WAP. La pasarela traduce HTML a WML, y lo comprime a un formato binario que se usa para ahorrar ancho de banda en el vínculo de radio. El cliente tiene un navegador WAP, el cual interpreta los archivos binarios comprimidos. Generalmente si el operador de telefonía inalámbrico ofrece el servicio WAP, en su infraestructura tendrá la pasarela WAP para su comunicación con las redes IP.

Para estos servicios el desarrollador de aplicaciones no necesita acceder directamente ningún recurso del operador inalámbrico, y su atención esta enfocada solamente en el lado del servidor de aplicaciones y los recursos comunes asociados a ese servidor, como acceso a bases de datos, a un servidor web, comunicación con servicios web etc. Las aplicaciones que hacen uso de estos servicios son los que normalmente se desarrollan por medio de paginas WML a las cuales los usuarios pueden acceder desde el navegador de su dispositivo móvil, y pueden realizar consultas, llenar formularios, descargar información al teléfono como ringtones, imágenes, juegos etc.

1.2.2 Servicios Basados en Mensajería

1.2.2.1 SMS

El Servicio de Mensajes Cortos (SMS, Short Message Service) es un servicio inalámbrico globalmente aceptado que habilita la transmisión de mensajes alfanuméricos entre los suscriptores móviles y sistemas externos como el correo electrónico, paginación, y correo de voz. El servicio de mensajes cortos (SMS) es la transmisión de mensajes de texto cortos desde y hacia un teléfono móvil, maquina de facsímil y / o dirección IP. El termino *mensaje corto* proviene de la longitud relativamente corta de cada mensaje individual, la cual se limita a 160 caracteres alfanuméricos (mapeados sobre 140 bytes) para cada mensaje y no contienen imágenes o gráficos.

El proceso de envío de un SMS se da de la siguiente forma, una vez un mensaje se envía, se recibe por un Centro de Servicio de Mensajes Cortos (SMSC) que a su vez debe enviarlo al dispositivo móvil apropiado. Al hacer esto, el SMSC envía un requerimiento de SMS al Registro de Ubicación Local *Home Location Register* (HLR) para encontrar al cliente libre. Una vez el HLR recibe el llamado responderá al SMSC informando el estado del suscriptor: activo o inactivo. Si la contestación es “inactivo”, entonces el SMSC almacenará el mensaje por un periodo de tiempo que puede ser establecido por el usuario o por el operador, si es necesario o si es solicitado, el SMSC puede enviar una notificación al usuario transmisor sobre el éxito o fracaso del envío de un mensaje. Cuando el suscriptor accede a su dispositivo, el HLR envía una notificación de SMS al SMSC quien intentara la entrega. El SMSC transfiere el mensaje a un punto de entrega. El sistema compagina el dispositivo y si responde, el mensaje se entrega. El SMSC recibe confirmación de mensaje recibido por el usuario terminal, entonces clasifica el mensaje como “enviado” y no intentara enviarlo de nuevo. El sistema de “almacenamiento y reenvió” de los SMS asegura que cuando el móvil receptor no puede ser alcanzado inmediatamente, el mensaje se entregará eventualmente cuando el terminal del suscriptor receptor nuevamente este en servicio y listo para recibir mensajes. El servicio de SMS tiene los siguientes rasgos distintivos:

- Confirmación de entrega del mensaje.

- Los SMS pueden ser enviados y recibidos simultáneamente con la voz, datos y facsímil de las llamadas. Esto es posible porque considerando que la voz, datos y el facsímil de llamada toman un canal de radio especializado para la duración de la llamada, los mensajes cortos viajan sobre el canal de radio usado para señalización.
- Están disponibles los métodos para enviar SMS múltiples. El encadenamiento y condensación de SMS (consiguiendo mas de 160 caracteres de información en un solo mensaje) se ha definido y se ha incorporado en las normas GSM para SMS.

1.2.2.2 Arquitectura y Elementos de Red en SMS

En la figura 1.3 se pueden observar los elementos principales de la arquitectura SMS.

- **Las Entidades de Mensajería corta, SME (Short Message Entity).** Es el dispositivo que origina los mensajes cortos. Las SME puede ser una estación móvil o estar localizada en la red fija o en otro centro de servicio.

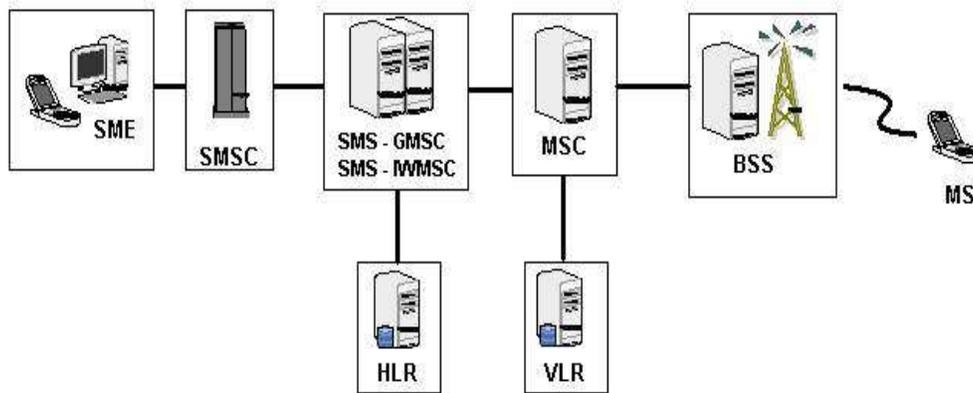


Figura 1.3 Entidades de red involucradas en el servicio SMS

- **El Centro de Servicio de Mensajes Cortos, SMSC (SMS Centre).** Maneja todas las funciones relacionadas con el envío de los mensajes. El SMSC recibe los mensajes de las SME y lo envía al correspondiente receptor. Si un mensaje no

puede ser enviado inmediatamente, es almacenado para un envío posterior.

- **La SMS – Gateway MSC, SMS – GMSC.** Es una función dedicada dentro del MSC que le permite recibir un mensaje de el SMSC, investigar el registro de ubicación local (HLR) con el fin de obtener la ubicación actual del móvil receptor para entregar el mensaje al MSC visitante y eventualmente a una estación móvil. La función SMS - GMSC se usa cuando una estación móvil es la receptora del mensaje.
- **La SMS – Interworking MSC, SMS – IWMSC.** Esta es otra función dedicada dentro del MSC que le permite al SMSC recibir mensajes de un MSC. Este es el caso cuando quien origina el mensaje corto es una estación móvil, aquí el mensaje se envía al SMSC receptor. El SMS – IWMSC se usa para mensajes originados en estaciones móviles.
- **Registro de Ubicación Local, HLR (Home Location Register).** Es un banco de datos que almacena la identidad y datos de usuario de todos los suscriptores pertenecientes al área de un determinado MSC. Para los SMS, los datos importantes en el HLR son la ubicación actual del terminal receptor y si esta encendido y disponible para recibir mensajes.
- **El Centro de Conmutación Móvil, MSC (Mobile Service Switching Centre).** Es el corazón de las redes celulares, maneja las transacciones de circuito conmutado y las no conmutadas por circuito, tal como, las llamadas telefónicas normales o procedimientos de suscriptor. El MSC, es responsable para el establecimiento, enrutado, control y terminación de una llamada; la gestión de handover entre MSCs y servicios suplementarios, accesos a las bases de datos de la red, la recolección de información para cargos e información contable. El MSC también se encarga de coordinar los cambios de un sitio celular a otro, también envía alertas a los teléfonos móviles, registra los momentos en que cada celular es prendido y También actúa como interfaz entre la red móvil y las redes publica de voz y datos.
- **Registro de Ubicación del Visitante, (Visitor Location register, VLR).** Es un

banco de datos que contiene información dinámica, acerca de todas las terminales móviles presentes localizadas dentro del área que atiende un MSC. El MSC y el VLR están siempre integrados dentro del mismo nodo físico.

- **El Sistema de Estación Base, BSS (Base Station System).** El BSS está en contacto directo con las estaciones móviles a través de la interfaz de radio. Por lo tanto, incluye las máquinas encargadas de la transmisión y recepción de radio, y de su gestión. Por otro lado, el BSS está en contacto con los conmutadores del NSS. La misión del BSS se puede resumir en conectar la estación móvil con otros usuarios.
- **La Estación Móvil, MS (Mobile Station).** En el caso particular de SMS, puede ser el emisor o el receptor de un mensaje corto. La estación móvil podría además informar a la red sobre situaciones anormales (erróneas) tales como una saturación de memoria.

1.2.2.3 Servicios Basados en SMS

Estos servicios involucran un recurso muy importante de la red Inalámbrica que es el *Centro de Mensajería (SMSC Service Message Service Centre)*. El desarrollador de aplicaciones debe tener acceso a este recurso para poder implementar su servicio. Asumiendo que se tiene acceso a este recurso, la aplicación tiene dos formas de comunicarse con el SMSC:

- **Comunicación directa:** el desarrollador debe implementar el protocolo de comunicación que maneja el SMSC, esta es una tarea complicada ya que no hay una estandarización en este aspecto lo que hace que existan diferentes protocolos para la comunicación con los centros de mensajería. Un protocolo que ha sido muy bien aceptado es el SMPP (Simple Message Peer to Peer), varios SMSC implementan este protocolo.
- **Utilizando una gateway SMS:** Esta gateway se encuentra entre la aplicación y el

SMSC, esta opción facilita la labor del desarrollador ya que no tendrá que preocuparse por los protocolos de comunicación con el SMSC, ese es precisamente el trabajo de la gateway, obviamente la gateway debe soportar el protocolo con el que trabaja el centro de mensajería. Sin embargo el desarrollador tiene la labor de configurar y gestionar la gateway SMS. Estas gateway son comerciales, y este puede ser el inconveniente al momento de desarrollar un servicio, para solucionarlo se puede recurrir a soluciones de código libre como es el caso de Kannel.

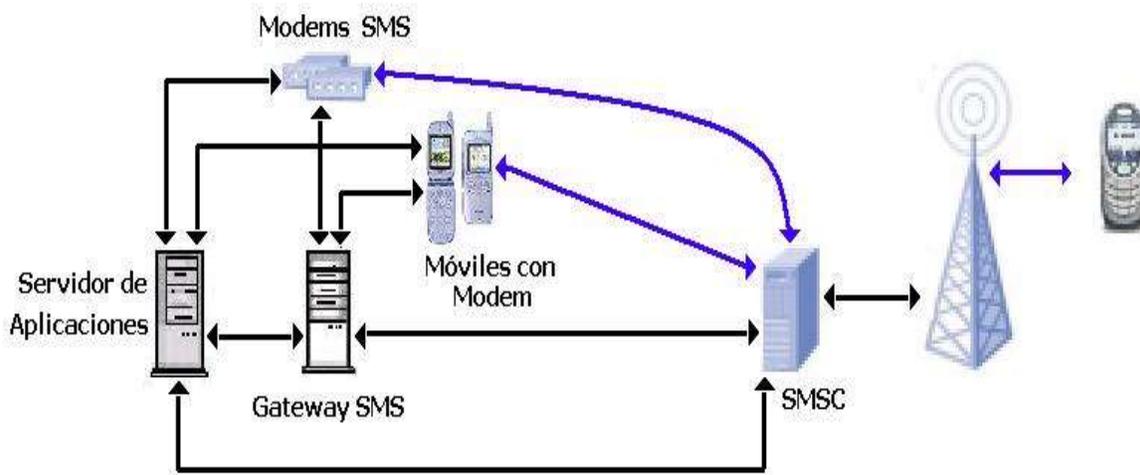


Figura 1.4 Componentes Principales en los servicios de Mensajería

En el caso de no tener acceso al SMSC, dependiendo del tipo de servicio, se puede utilizar un modem GSM para que sea éste quien envíe los SMSs al centro de mensajería. Algunos teléfonos GSM traen incorporado este recurso. La labor del desarrollador consiste en hacer que su servicio o aplicación se comunique con el modem, de tal forma que a través de él se pueda enviar y recibir la información que la aplicación necesita. Si se utiliza la gateway SMS, algunas de ellas traen la opción de poder conectar a ellas los módems SMS, de esta manera el trabajo del desarrollador nuevamente se ve favorecido. En la Figura 1.4 se muestran los elementos involucrados en estos servicios y se representa el tipo de conexiones desde el servidor de aplicaciones hasta el terminal móvil, las flechas negras representan conexiones cableadas y las azules conexiones inalámbricas. Las aplicaciones que hacen uso de este tipo de servicios son aquellas que se basan en una "solicitud - respuesta", las cuales se hacen por medio de un mensaje, de

esta forma se puede realizar consultas, participar en concursos de diferentes tipos, realizar procesos de registro y / o inscripciones etc.

1.2.3 Servicios Push

Como se menciona, estos servicios no necesitan intervención del usuario para iniciar la interacción con el cliente, simplemente cuando hay información nueva o se cumple una programación específica la información es enviada al usuario. La información que se entrega al usuario se puede presentar como SMS o como mensajes WAP PUSH, los cuales se tratarán más adelante. Dependiendo de la tecnología empleada para desarrollar la aplicación, puede haber la necesidad de una gateway Push para el envío de la información, si este es el caso, el desarrollador debe interactuar con dicho recurso, y si el operador inalámbrico no lo tiene, el desarrollador deberá proporcionarlo y gestionarlo.

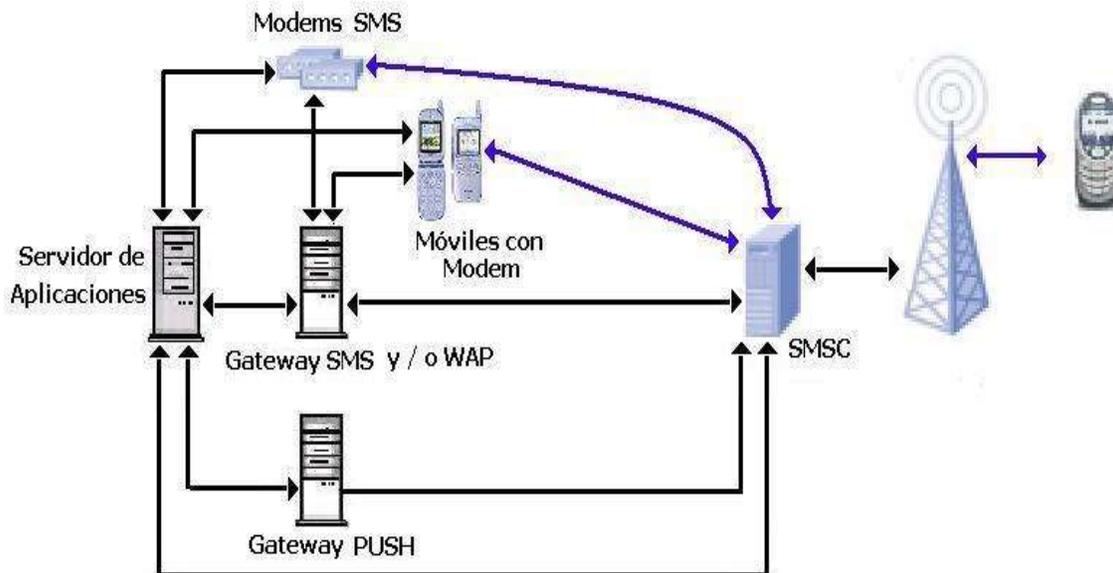


Figura 1.5 Componentes Principales en los servicios Push

El trabajo de la pasarela Push consiste exclusivamente en procesar los mensajes Push generados por la aplicación y enviarlos hacia la red inalámbrica, esta pasarela no puede recibir respuestas de la red inalámbrica que se han generado en los terminales móviles, ver Figura 1.5, por lo tanto si la aplicación demanda algún tipo de contestación o

respuesta por parte del terminal móvil esto se hace mediante Wap o a través de SMS, por este motivo este tipo de servicios puede verse como una ampliación de los expuestos anteriormente.

Estos servicios se utilizan para la transmisión de datos de baja capacidad, por lo que generalmente utilizan como portador el SMS. Este tipo de servicios son los que se emplean o son los mas indicados para emplearse básicamente en la notificación de alarmas, y también son muy indicados para aplicaciones de tipo publicitario, en las que la empresa interesada utiliza este servicio para distribuir su publicidad en forma de SMS o MMS a los usuarios de telefonía móvil.

1.3 AMBIENTES Y PLATAFORMAS DE DESARROLLO EN MENSAJERIA

1.3.1 J2ME

J2ME (Java 2 Mobile Edition) es la respuesta de Sun Microsystems, tanto al éxito logrado por su plataforma Java 2 para el desarrollo de aplicaciones distribuidas como al impresionante nivel de crecimiento de las comunicaciones inalámbricas alrededor del mundo. Cuando los diversos desarrollos en cuanto a tecnologías de comunicaciones inalámbricas comenzaron a abrir la posibilidad de ofrecer servicios de datos a través de dispositivos móviles como teléfonos celulares, PDA's o Pocket Sized PC's, se sintió la necesidad de crear plataformas de desarrollo especializadas que permitieran el desarrollo de tales servicios. Sun Microsystems respondió a esta necesidad creando la plataforma de desarrollo para dispositivos móviles Java 2 Micro Edition J2ME. La plataforma J2ME entrega los beneficios de la tecnología Java a los usuarios de dispositivos móviles, incluyendo una interfaz flexible de usuario, un modelo de seguridad robusto y un gran conjunto de protocolos de red soportados. Como se había mencionado, J2ME extiende la filosofía de portabilidad, de tal manera que las aplicaciones pueden ser desarrolladas para un gran número de dispositivos.

J2ME consiste principalmente de configuraciones y perfiles los cuales juntos definen completos ambientes de ejecución java para dispositivos con recursos limitados y con diversas capacidades. Algunas veces sin embargo, es necesario ampliar las características propias de J2ME para aprovechar tecnologías o recursos que día a día surgen en el entorno inalámbrico. Esto se logra con diferentes APIs que J2ME define como *paquetes opcionales*, los cuales son un conjunto de clases y otros recurso que pueden ser usados en conjunto con mas de un perfil J2ME.

1.3.1.1 Wireless Messaging API (WMA)

Como todos los otros componentes de J2ME, los paquetes opcionales están definidos como *Java Specification Requests* (JSRs) dentro de la comunidad Java. Uno de los primeros paquetes opcionales para j2me es la JSR 120, que hace referencia a la API para mensajería inalámbrica (WMA), que se utiliza por las aplicaciones para enviar y recibir textos cortos o mensaje en formato binario sobre conexiones inalámbricas.

La WMA se basa en el *Generic Connection Framework* (GCF), definido por la configuración CLDC (*Connected Limited Device Configuration*). El GCF proporciona un framework abstracto para ejecutar conexiones de entrada y salida, un framework cuyos perfiles y paquetes opcionales pueden fundamentarse en suministrar las conexiones de entrada / salida que se necesiten. Los mensajes enviados y recibidos con la WMA soporta el servicio de mensajería corta SMS, usado por teléfonos celulares y dispositivos similares en redes GSM y / o CDMA. La WMA particularmente soporta SMS y CBS (Cell Broadcast Short Message Service). Aunque los mensajes WMA son similares a los datagramas, no usa las interfaces de datagrama definidas por el GCF, las cuales actualmente están destinadas para usarse con conexiones UDP. En lugar de eso, la WMA define un nuevo conjunto de interfaces en el paquete *javax.wireless.messaging*.

1.3.1.2 Clases WMA de Nivel Superior

Los desarrolladores de aplicaciones pueden acceder a las características de WMA a través de tres interfaces de nivel superior en el paquete *javax.wireless.messaging*:

- **Interfaz Message:** Define la estructura del mensaje. Las interfaces *TextMessage* y *BinaryMessage* se derivan de *Message* y proporcionan estructuras de mensajes mas específicas.
- **Interfaz MessageConnection:** Representa una conexión de red para los mensajes. Define métodos básicos para el envío y la recepción de mensajes. Por ejemplo el método *MessageConnection.newMessage()* devuelve un objeto *Message* para los mensajes de salida, el método *MessageConnection.receive()* captura los mensajes entrantes.
- **Interfaz MessageListener:** Solo tiene un método *notifyIncomingMessage()*. Un objeto *MessageListener* se registra con una conexión *MessageConnection* de tipo servidor. El método *notifyIncomingMessage()* es llamado cuando hay un mensaje de entrada. La especificación exige que este método retorne rápidamente, por lo que no es recomendable que los mensajes de entrada se procesen al interior de este método. La Figura 1.6 contiene un diagrama UML para esas interfaces.

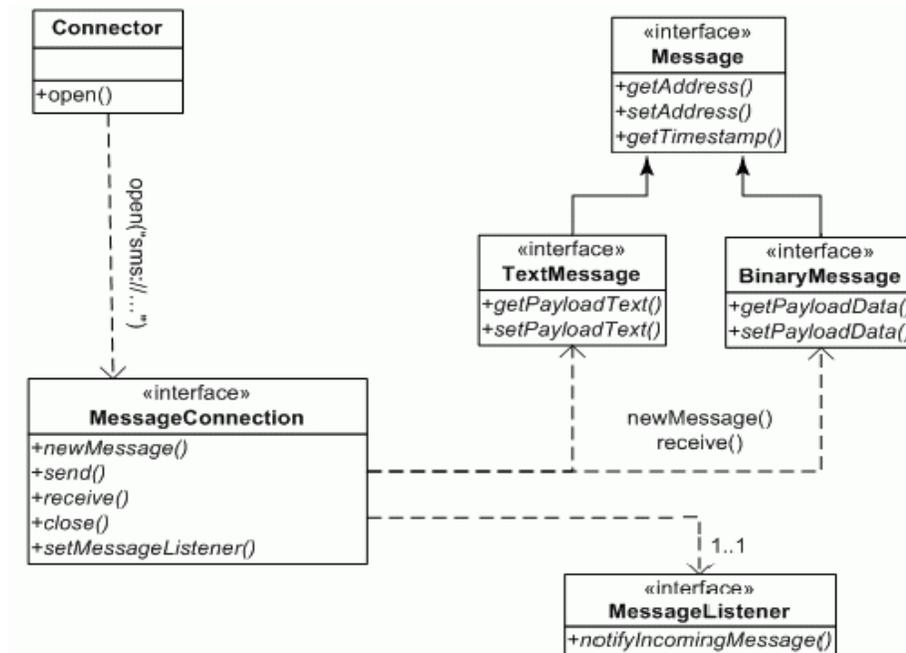


Figura 1.6 Interfaces WMA de nivel superior en el paquete javax.wireless.messaging

Para enviar o recibir mensajes, la aplicación debe primero obtener un objeto de la interfaz *MessageConnection*, usando la fabrica de conexiones del GCF. La URL que se pasa al método *javax.microedition.io.Connector.open()* identifica el protocolo a usar (SMS o CBS), y el numero telefónico del receptor, el puerto o ambos. A continuación se presentan todos los URLs validos:

- **sms://+573113247212** : especifica una conexión de cliente para enviar mensajes SMS al teléfono 573113247212. Si el puerto no esta especificado, se toma por defecto el puerto al cual llegan todos los mensajes de entrada (inbox) del dispositivo receptor, y los mensajes serán manejados por el cliente de mensajería entrante del teléfono.
- **sms://+573113247212:5678** : Abre una conexión de cliente para enviar mensajes SMS al puerto numero 5678 al numero telefónico 573113247212.
- **sms://:5678** : abre una conexión de tipo servidor en el dispositivo local sobre el puerto 5678, permitiéndole a la aplicación esperar y responder a los mensajes que llegan desde otros dispositivos.
- **cbs://:5678** : especifica una conexión que escucha mensajes CBS de entrada en el puerto 5678. diferente a la conexión SMS de tipo servidor, esta conexión no puede enviar mensajes.

A continuación se presentan algunos ejemplos de utilización de la API WMA. Se puede enviar un SMS a un numero telefónico cualquiera y / o a un puerto especifico a través de una *MessageConnection* construida para ese propósito como se muestra en el ejemplo 1.

Ejemplo 1. Envío de un SMS

```
String addr = "sms://+123456789";  
// O: String addr = "sms://+123456789:1234";  
MessageConnection conn = (MessageConnection) Connector.open(addr);  
TextMessage msg = (TextMessage) conn.newMessage(  
MessageConnection.TEXT_MESSAGE);  
msg.setPayloadText( " Hola Mundo " );  
conn.send(msg);
```

También se puede enviar mensajes a través de una *MessageConnection* de tipo servidor. Como se muestra en el ejemplo 2. El método `MessageConnection.receive()` permanece bloqueado hasta que se recibe un mensaje o la conexión se cierra. El ejemplo 3 muestra un bucle servidor que escucha, recibe y contesta los mensajes SMS de entrada.

Ejemplo 2. Envío de un SMS mediante una conexión tipo servidor.

```
MessageConnection sconn = (MessageConnection)
Connector.open("sms://:3333");
TextMessage msg = (TextMessage) sconn.newMessage(
MessageConnection.TEXT_MESSAGE);
msg.setAddress("sms://+123456789:1234");
msg.setPayloadText( "Hola Mundo" );
sconn.send(msg);
```

Para la recepción de mensajes se necesita establecer una conexión de tipo servidor que escuche los mensajes que llegan a un puerto específico.

Ejemplo 3. Recepción y Respuesta de un SMS.

```
MessageConnection sconn = (MessageConnection)
Connector.open("sms://:3333");
while (true) {
    Message msg = sconn.receive();
    if (msg instanceof TextMessage) {
        TextMessage tmsg = (TextMessage) msg;
        String msgText = tmsg.getPayloadText();

        // Construcción del mensaje de retorno
        TextMessage rmsg = (TextMessage) sconn.newMessage(
MessageConnection.TEXT_MESSAGE);
        rmsg.setAddress( tmsg.getAddress() );
        rmsg.setPayloadText( "Gracias!" );
        sconn.send(rmsg);

    } else {
        // procesar el caso que no sea un mensaje de texto
        // podría ser un mensaje binario BinaryMessage
    }
}
```

Se debe tener en cuenta que WMA es responsable por el envío y la recepción de los mensajes, no por su interpretación. Las clases estándar para el manejo de flujos de datos como *DataInputStream*, *DataOutputStream*, etc., pueden facilitar la codificación y

decodificación de todo tipo de datos binarios de una manera portable y eficiente.

1.3.2 J2ME Y PUSH

Con el perfil MIDP 2.0 las aplicaciones han adquirido una serie de nuevas e interesantes características, una de ellas es que ahora las aplicaciones J2ME en el dispositivo móvil pueden ser iniciadas con base en una solicitud realizada desde una conexión remota (la llegada de un SMS por ejemplo) o en un temporizador programado. Esta característica combinada con la API WMA para el manejo de mensajería dan como resultado aplicaciones que funcionan de acuerdo al modelo Push, ya que se entrega contenido a la midlet en el dispositivo móvil y la aplicación se activa sin la interacción del usuario.

1.3.2.1 Activación de una MIDLet

Una vez instalada en el dispositivo móvil, una MIDlet se puede activar por una de las siguientes formas:

- Por solicitud del usuario. La más común antes del perfil MIDP 2.0.
- Por una conexión entrante de la red.
- Por una alarma programada.

A continuación se describe de manera general la activación basada en red y en alarma.

Conexión proveniente de la red. El sistema de administración de aplicaciones (AMS) puede monitorear actividad en un puerto buscando por un tipo de conexión y puerto específico. Con las opciones adicionales de conectividad de red ofrecidas por MIDP 2.0 una solicitud proveniente de la red puede ser alguna de los siguientes tipos:

- Basada en Mensajería, tal como SMS.
- Basada en Paquetes, tal como datagramas.
- Basada en Flujos, tal como los sockets.

Alarma Programada. Una MIDlet también puede activarse a una hora específica. Esto puede ser bastante útil para eventos repetitivos como recordatorios de reuniones y otros compromisos. Un tiempo de alarma se especifica usando un entero largo (*long integer*) con el mismo formato de *Date.getTime()*.

Registro. Para que una MIDlet sea notificada de conexiones provenientes de la red o para “activar” un tiempo programado, la MIDlet debe registrar esas solicitudes con el AMS, éste maneja una lista de aquellas conexiones y alarmas basadas en tiempo que son aceptadas, a esa lista se conoce como registro push. Las solicitudes se pueden registrar de dos formas:

- Registro Dinámico: Ocurre en tiempo de ejecución cuando la MIDlet notifica al AMS intento de permitir conexiones provenientes de la red o para ser activada alguna vez en el futuro a través de una alarma. Los registros se hacen por invocación de métodos específicos pertenecientes a la clase *PushRegistry*.
- Registro estático: las solicitudes provenientes de la red se pueden definir en el archivo JAD de la aplicación. Este tipo de solicitud se considera estático porque el transmisor y el tipo de conexión son conocidos cuando la MIDlet se instala. Un registro estático se define usando el atributo **MIDlet-Push-<n>**

`MIDlet-Push-<n>: <ConnectionURL>, <MIDletClassName>, <AllowedSender>`

Donde:

- *ConnectionURL* es la cadena que especifica la URL para monitorear por conexiones de entrada
- *MIDletClassName* es el nombre de clase de la MIDlet que se invocará.
- *AllowedSender* es el filtro para especificar cuales transmisores están permitidos para solicitar una conexión.

Se debe tener en cuenta que el registro estático no soporta la activación basada en alarma.

1.3.3 WAP Push¹

WAP Push es un estándar abierto desarrollado por el WAP Forum para proporcionar las capacidades push que hacen posible nuevas aplicaciones y mejoras impresionantes a los existentes servicios habilitados para push. Aunque WAP Push también se basa en el modelo cliente / servidor, proporciona medios estándar para enviar datos a un suscriptor móvil sin una petición explícita cuando los datos son entregados.

En la figura 1.7 se presenta un ejemplo, aquí el usuario recibe una grafica de las acciones en la bolsa de valores y puede comprar inmediatamente la acción (símbolo ABC) simplemente con un click. Esto no es posible con otras alternativas al WAP Push , tal como los SMS estándar.

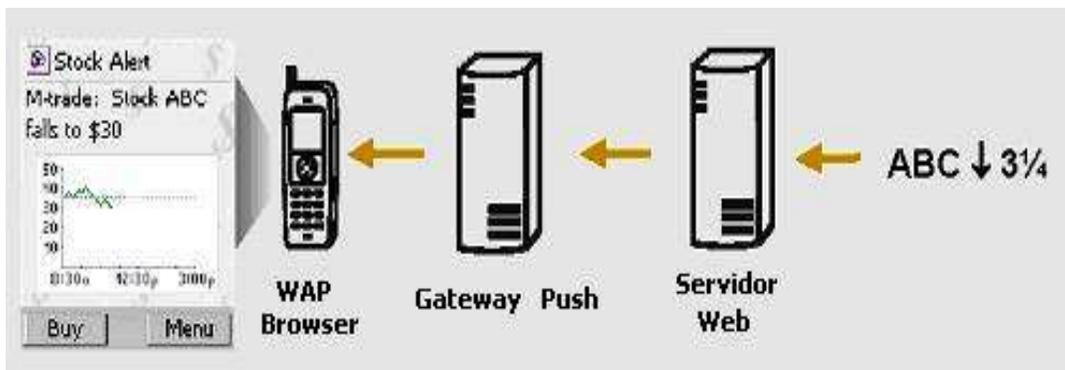


Figura 1.7: Alerta en las acciones con WAP Push

El sistema WAP Push transporta datos sin la intervención de un usuario a un microteléfono habilitado para WAP. Una operación WAP Push ocurre cuando un Iniciador Push en el dominio de Internet transfiere contenido a un cliente en el dominio de WAP a través de la Push Proxy Gateway (PPG). Los componentes principales del sistema se describen a continuación.

- **Iniciador Push (Push Initiator):** Un Iniciador Push es la entidad (una aplicación, tal como un servidor de noticias, que corre en un servidor web ordinario) que origina el contenido Push y lo envía en forma de una petición Push, usando el

¹ Documento de Internet WAP_push_1201.pdf

Protocolo de Acceso Push (PAP *Push Access Protocol* por sus siglas en inglés) sobre HTTP a la *Push Proxy Gateway* para la entrega al microteléfono. Cada envío tiene un identificador único, el Iniciador Push puede solicitar el resultado de un envío (entregado o que no se pudo entregar), comprobar las capacidades de un dispositivo específico del cliente y comprobar el estado de un envío previo o la cancelación de un envío.

- **Push Proxy Gateway (PPG):** Es el punto de acceso para los Pushs enviados a los clientes desde Internet. La PPG puede implementar políticas incluyendo: control de acceso del Iniciador Push, detección de error en el contenido Push, la resolución de la dirección del cliente, búsqueda de las capacidades del microteléfono, traducción del contenido y la conversión de protocolos (HTTP a WSP, HTTP a SMS). La PPG puede también notificar al Iniciador Push sobre el estado de la operación del Push. Desde la PPG, el mensaje Push se entrega al microteléfono usando el protocolo *Push Over-The-Air* (OTA). La Figura 1.8 ilustra los puntos sobresalientes del *framework* requerido para el Push.

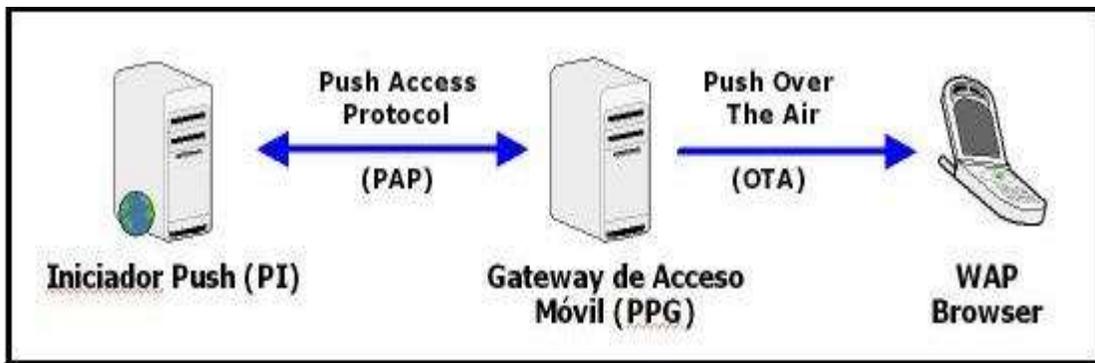


Figura 1.8 Framework WAP Push

El aspecto *over-the-air* del *framework Push* contiene un número de adaptaciones del portador (carrier) según lo especificado por WAP y se puede clasificar en una de las siguientes áreas:

- Basado en SMS: Entrega Push al cliente No confiable / no confirmada; ancho de banda limitado; el portador SMS se puede utilizar como canal de la entrega cuando la confirmación no es crítica o iniciar una sesión de los datos con un

dispositivo que actualmente no esté conectado.

- Conmutación de circuitos: Entrega Push confiable; mayor ancho de banda disponible.
- Conmutación de Paquetes: Uso eficiente de los recursos de la red, mayor ancho de banda disponible, entrega confiable.

Cliente WAP Push: El cliente WAP Push está incluido en un dispositivo (teléfono móvil) capaz de aceptar el contenido WAP Push. Tiene aplicaciones, tales como un navegador WAP, que puede recibir y procesa el contenido Push. También tiene la aplicación de iniciación de sesión (SIA), la cual puede ser utilizada por la gateway para pedir que el cliente inicie una sesión WAP.

La estandarización e implementación de los servicios Push tratan algunos aspectos fundamentales que desarrolladores y portadores (carriers) deben enfrentar cuando se construye y desarrolla aplicaciones de datos:

- Interfaz del desarrollador: La interfaz hacia la PPG es un documento PAP transportado usando HTTP según lo definido en las especificaciones de WAP.
- Direccionamiento del Usuario: Desde la perspectiva del desarrollador, el dispositivo móvil puede ser dirigido por el MSISDN (*Mobile Station ISDN Number*), o por el identificador del suscriptor. Ambos se pueden recoger de peticiones hechas a un sitio Web o desde el usuario directamente señalándolo dentro de la aplicación. El MSISDN está contenido en la cabecera HTTP *x-up-calling-line-id* y el identificador del suscriptor en la cabecera *x-up-subno*. Adicionalmente las cabeceras HTTP que identifican la dirección de la gateway Push son *x-up-wappush-secure* y *x-up-wappush-unsecure*.
- Cliente No Online: El cliente puede no estar en línea cuando el iniciador Push envía un mensaje. La PPG intermedia debe posponer el Push para una entrega posterior (dependiendo de los datos fecha/hora suministrados por el PI) o iniciar la

conectividad con el cliente. El inicio de conectividad se logra usando una de las opciones del transporte descritas en páginas previas, por ejemplo mediante portador SMS. Esto se logra usando un mecanismo estándar, la petición de la iniciación de sesión (SIR por sus siglas en ingles, *Session Initiation Request*), definida dentro de la especificación WAP Push.

- Descubrimiento de las Capacidades del Dispositivo móvil: La capacidad del dispositivo de aceptar Push o es anunciada por el browser durante el establecimiento de la sesión o por el análisis de interacciones anteriores que el dispositivo ha tenido con la gateway. La manera generalmente aceptada en la cual el dispositivo hace esto es usando cabeceras HTTP [RFC2616] estándares. El iniciador Push puede también solicitar las capacidades del cliente usando la pregunta CCQ definida en WAP PAP.
- Acciones en el Content Type: Cuando un dispositivo móvil anuncia soporte para el contenido Push dentro de sus capacidades, cómo se comporta cuando recibe y procesa el contenido Push esta definido dentro de la especificación WAP Push. Como resultado, los desarrolladores pueden estar seguros de un cierto nivel de comportamiento uniforme y determinista del microteléfono. Estas acciones se pueden determinar de dos maneras:
 - Contenido estandarizado: WAP Forum define tres *content types* que podrían comportarse de una forma especifica en los microteléfonos: *Service Indications*, *Service Loads* y *Cache Operations*.
 - El WAP Forum define un mecanismo de envió a las aplicaciones. En el caso donde hay aplicaciones independientes en el dispositivo, el contenido puede ser apuntado a una aplicación particular especificando el ID de la aplicación en la petición Push. Este mecanismo permite al WAP Push ser un mecanismo de entrega extensible que se puede utilizar para entregar el contenido a actuales y futuras aplicaciones en el dispositivo móvil.

Estos tipos de contenidos (*content types*) estandarizados pueden ser utilizados por el desarrollador de contenido para construir una aplicación Push que tenga un nivel de

comportamiento más constante, más universal y determinista.

- La intromisión sobre el usuario puesto que el contenido Push se entrega sin la interacción previa del usuario, el desarrollador de contenido debe considerar cómo el contenido puede interrumpir al usuario. De acuerdo con las reglas de presentación definidas por WAP, el iniciador Push puede especificar diversos ajustes de la prioridad para el contenido Push que se entrega al dispositivo. Una prioridad más baja puede no interrumpir al usuario, mientras un mensaje de alta prioridad podría hacer un ruido y mostrar un aviso en pantalla al usuario. Los proveedores de browser han adoptado estas especificaciones de modo que el desarrollador esta razonablemente asegurado que hay comportamiento uniforme para las aplicaciones.

1.3.3.1 Push Access Protocol - Generalidades

El Push Access Protocol (PAP) proporciona medios de envío de contenidos desde Internet a un dispositivo móvil, tal como un teléfono móvil habilitado para WAP. El PAP alcanza esto manejando comunicaciones entre el servidor de Internet que envía el contenido, conocido en este contexto como *Iniciador Push* (PI) y la *Push Proxy Gateway* (PPG) que es el servidor que envía el contenido al dispositivo móvil destino (ver Figura 1.7). La PPG actúa como intermediario, conectando las redes cableadas con las redes inalámbricas, que de otra manera no tendrían ninguna manera de comunicarse unas con otras porque utilizan diferentes protocolos de comunicación. El PAP es una tecnología Push, por tanto este protocolo le permite a un *servidor* iniciar la transferencia de contenido a uno o más dispositivos móviles del cliente.

El PAP esta construido sobre el *Extensible Markup Language* (XML) y transportado usando HTTP (protocolo de transferencia de hipertexto), aunque otros protocolos de transporte tales como SMTP (*Simple Mail Transfer Protocol*) pueden estar disponibles en el futuro. Una petición Push es un documento multiparte que puede contener tres entidades:

- **La entidad de control:** es un documento XML que contiene instrucciones de entrega destinadas para la Gateway de acceso móvil PPG. La entidad de control es obligatoria; identifica el dispositivo móvil receptor y contiene instrucciones de entrega tales como las restricciones del tiempo de entrega, cuando o en que momento se debe procurar la entrega, cuando expirará y qué portador subyacente de la red debe ser utilizado. Además, se puede solicitar una confirmación y se puede especificar una URL indicando donde se debe entregar esa confirmación después de que el Push es recibido.
- **La entidad de Contenido:** Contiene el contenido destinado para el dispositivo móvil, es decir, es la única parte del envío Push que se entrega al microteléfono. La entidad de contenido se requiere solamente para los envíos Push y debe ser la segunda entidad en el documento multiparte.
- **La entidad opcional de capacidades:** contiene las capacidades del dispositivo móvil para las cuales el mensaje fue formateado. El Iniciador Push puede crear esta entidad para indicar lo que él asume de las capacidades del móvil. Usando el esquema del perfil del agente del usuario WAP, en el lenguaje de la definición del recurso (RDF), el desarrollador de contenido puede indicar qué proveedor de hardware y fabricante de browser se entiende como el receptor. Si la PPG reconoce que un usuario no concuerda con estas capacidades, la entidad de contenido (*content entity*) no se entrega. La PPG también envía una entidad de capacidades en respuesta a un mensaje de solicitud de las capacidades del cliente (CCQ por sus siglas en ingles) que llegue del Iniciador Push.

Estas entidades están juntas como un documento *Multi-Purpose Internet Mail Extensions (MIME)*, que se envía desde el Iniciador Push a la *gateway* de acceso móvil PPG usando HTTP.

Operaciones PAP: PAP hace posible las siguientes operaciones:

- **Envío Push (Push Submission):** entrega un mensaje Push de un Iniciador Push a un dispositivo móvil. Debido a que el envío Push contiene dirección, contenido, y

entidades opcionales de capacidades, éste se entrega como un documento *multipart/related*. Un envío Push puede entregar alguno de los tipos de contenido que se describen posteriormente.

- **Consulta de estado (Status Query):** El iniciador Push puede solicitar el estado actual de un envío Push. Todas las solicitudes de consulta de estado se entregan como documentos XML.
- **Cancelación Push (Push Cancellation):** permite que el Iniciador Push intente cancelar un envío Push. Todas las peticiones de cancelación Push se entregan como documentos XML.
- **Consulta de Capacidades del Cliente (Client Capabilities Query):** El Iniciador Push puede consultar a la PPG para recuperar las capacidades para un dispositivo móvil específico. Todas las solicitudes CCQ se entregan como documentos de XML. La PPG retorna la información CCQ en un documento *multipart/related* que el Iniciador Push deba analizar para recuperar la información relevante.
- **Notificación de Resultado (Result Notification):** La PPG informa al Iniciador Push el resultado final del envío Push, por ejemplo, confirmación de la entrega del contenido al dispositivo del móvil destino. La notificación de resultado es opcional y ocurre solamente si el iniciador Push la solicita. Todas las notificaciones de resultado se entregan al Iniciador Push como documentos XML cuando el resultado final del envío Push es conocido. Las notificaciones de resultado generalmente no están disponibles inmediatamente después de enviar un envío Push.

Tipos de Contenido de los Envíos Push (Push Submission). Como ya se menciono, el componente *entidad de contenido* de la operación *PAP envío Push* es la única parte que se entrega al dispositivo móvil. Están disponibles Los siguientes tipos de contenido estandarizados por WAP:

- **Indicación de Servicio (Service Indication SI):** Conocido comúnmente como una "alerta", este tipo de contenido consiste de notificaciones asincrónicas sobre nuevo e-mail, cambio en los precios de mercancías seleccionadas, titulares de las

noticias, anuncios, recordatorios, etc. En su forma más básica, un SI contiene un breve mensaje y una URL asociada que especifican un servicio, con la opción para cargar dicho URL. El cliente inalámbrico puede comenzar el servicio inmediatamente seleccionando el URI o almacenarlo para una acción posterior. Los identificadores en el contenido de la Indicación de Servicio ofrecen la capacidad de sustituir contenido duplicado, esto quiere decir que en el caso de envíos repetidos, sólo el último será presentado al usuario.

- **Servicio de Carga (Service Loading SL):** este tipo de contenido permite a un agente de usuario en el dispositivo del cliente cargar y ejecutar un servicio, especificado por una URL, sin la intervención del usuario, al usuario no se le da la opción de escoger, el URL se activa automáticamente. Por ejemplo, un SL puede notificar a un dispositivo móvil de un saldo bajo en un servicio prepago y solicitar la acción del usuario cargando una pagina WML que presente una variedad de opciones.
- **Operación de Caché (Cache Operation):** Este tipo de contenido hace posible invalidar objetos de contenido en la caché del dispositivo inalámbrico. Todos los objetos de contenido invalidados, la próxima vez que quieran ser accedidos deben ser recargados del servidor en el que se originaron. Se utiliza la operación de caché si una aplicación no puede predecir cuando expirará el contenido que ella crea. Por ejemplo, se utiliza una Operación de Caché para asegurar que un dispositivo móvil siempre cargue el contenido más reciente de una aplicación de *mailbox*. Una Operación de Caché es un documento XML que consiste de una URL y uno de los siguientes tipos de operación:
 - **Objeto invalido (Invalidate object):** invalida el objeto específico que la URI identifica.
 - **Servicio Invalido (Invalidate service):** invalida todos los objetos que compartan el mismo prefijo URI.
 - **Custom Content:** este tipo de contenido puede virtualmente consistir de cualquier texto, imagen, u otros medios.

En el caso donde el usuario pospone el procesamiento de un Push, un indicador se activa en la pantalla del dispositivo móvil destacando que hay contenido disponible para presentación. La naturaleza exacta de esta indicación depende en particular del proveedor del microteléfono, pero un ejemplo de esto puede tomar la forma de un signo "@" en el dispositivo. Usando el "identificador de la aplicación" el Push puede apuntar a una aplicación particular que funciona en el dispositivo móvil, proporcionando direccionamiento a nivel de aplicación. Por defecto se asume que el Push estará apuntado al browser.

1.3.4 OSA (OPEN SERVICE ACCESS)

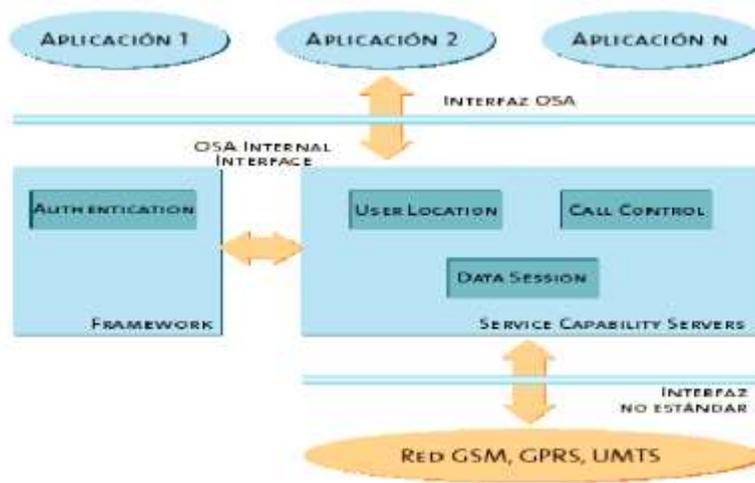


Figura 1.9 Arquitectura y descripción de OSA

OSA, Acceso Abierto a los Servicios, es un estándar del 3GPP (3rd *Generation Partnership Program*) que define una arquitectura que permite al operador y a las aplicaciones de terceras partes hacer uso de las funcionalidades de las redes de telecomunicaciones a través de una interfaz abierta y estandarizada (interfaz OSA). Provee la conexión entre aplicaciones y capacidades de servicio provenientes de la red. Así, las aplicaciones llegan a ser independientes de la tecnología de red que se utilice. Es una forma de publicar las facilidades y funcionalidades de las redes para que puedan ser fácilmente utilizadas, ocultando las características y problemáticas de implementación de los distintos proveedores de redes de telecomunicaciones. La arquitectura de OSA define

tres niveles en la frontera con el operador de red (ver Figura 1.9):

- El nivel de aplicaciones. Es el espacio de las aplicaciones que implementan los servicios de valor agregado (SVA). Estas aplicaciones son las que utilizarán la interfaz OSA para acceder a los recursos de la red móvil. Las aplicaciones pueden ser propias del operador de red o de terceros.
- El nivel de servicios básicos o “Framework”. Es el espacio de las aplicaciones básicas necesarias para establecer el entorno apropiado para utilizar las facilidades de red. Algunos ejemplos son: autenticación, búsqueda de capacidades de servicio, etc.
- El nivel de servidores de capacidades de servicio (SCS). Es el espacio de las aplicaciones donde se proporciona una interfaz OSA sobre las funciones de red del operador móvil.

Las facilidades y / o funcionalidades de red, a las que puede accederse a través de los Service Capability Servers (SCS), son:

- Call Control. Permite el control de 1 a n llamadas, siguiendo un modelo similar al de las Redes Inteligentes. Las aplicaciones pueden definir los eventos de la llamada, eventos que se desean conocer para redefinir el encaminamiento de las llamadas.
- Data Session. Permite el control de 1 a n sesiones de datos y es un modelo similar al utilizado en el Call Control, es decir, pueden definirse los eventos del modelo de conexión del portador, modelos que se desean conocer para modificar algún aspecto del procedimiento de establecimiento de la sesión de datos.
- Network User Location. Permite obtener información de la ubicación geográfica del terminal móvil.
- User Status. Permite obtener información sobre el estado del terminal, por

ejemplo: activo, apagado, etc.

- Terminal Capabilities. Permite conocer las capacidades de un terminal.
- Message Transfer. Permite la transferencia de mensajes a un usuario, pudiéndose definir los eventos que se desea conocer del procedimiento de transferencia.

1.3.4.1 API PARLAY / OSA

Internet ha creado oportunidades para implementar nuevos servicios de contenidos y comunicación, pero el modelo de inteligencia de red de Internet es totalmente opuesto al de RI. En las RI los servicios son prestados y gestionados por el operador de red, Internet abre la posibilidad de que cualquier entidad con infraestructura para ello pueda ofrecer sus servicios.

El entorno de telecomunicaciones de hoy se caracteriza por la coexistencia de estos dos modelos. Uno de los problemas clave para los operadores de red es proporcionar servicios de valor agregado y mantener acuerdos de nivel de servicio (Service Level Agreements) en un entorno mucho más complejo que la red tradicional de telefonía. La API PARLAY / OSA fue la respuesta de la industria a esta problemática. PARLAY / OSA es la definición de una API (Interfaz para la Programación de Aplicaciones) estandarizada y abierta que permite el acceso a los recursos de las redes de los operadores a aplicaciones que no tienen por qué residir en el dominio del operador.

1.3.4.2 Arquitectura lógica de red PARLAY / OSA

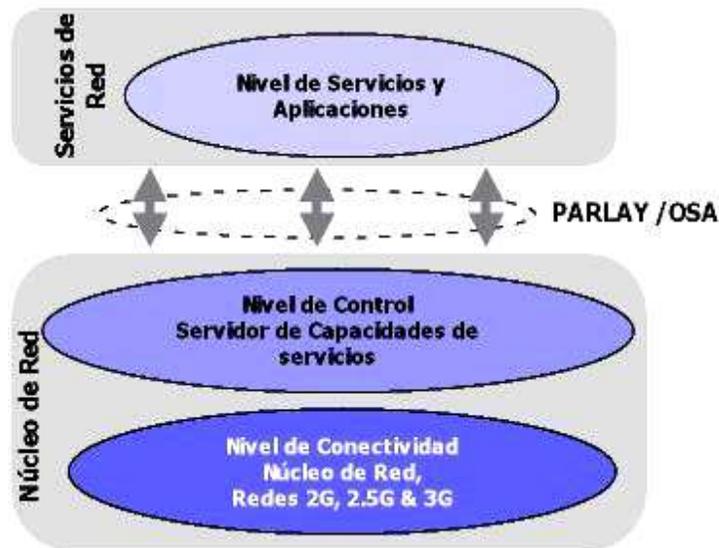


Figura 1.10 Arquitectura lógica de red PARLAY / OSA

En OSA se encuentran tres niveles bien definidos que son: El nivel de servicio en la parte superior, aquí es donde se ubica la gateway Parlay / Osa, los elementos de la red de telecomunicaciones están en los niveles de control y conectividad. El nivel de control maneja la parte de señalización y el nivel de conectividad maneja el tráfico. El gran beneficio de esta estructura es que la gateway Parlay / Osa simplifica mucho la complejidad de las redes de telecomunicaciones ocultando los niveles inferiores y suministrando a los desarrolladores de aplicaciones una interfaz estandarizada, de esta manera se reduce el conocimiento específico de las redes por parte del desarrollador y se incrementa la portabilidad de las aplicaciones desarrolladas.

En la Figura 1.11 se distinguen las siguientes entidades lógicas: Aplicaciones, Servidores de Aplicaciones, Servidores de capacidades del servicio, el Framework PARLAY / OSA y los elementos del núcleo de la red. Las aplicaciones, desplegadas en el servidor de aplicaciones, las cuales pueden estar sobre cualquier plataforma estándar, utilizan las capacidades definidas en Parlay / Osa, provistas por los Servidores de Capacidad de Servicios (SCS) y ofrecidas a las aplicaciones a través de las APIs: los SCS implementan el lado del servidor y la aplicación el lado del cliente de la API. La comunicación entre las aplicaciones y los SCS se realiza a través de infraestructura estándar de mediación. Los

SCSs son las entidades lógicas que implementan la API y que potencialmente interactúan con los elementos del núcleo de red, entre los que se incluyen, el *Home Location Register* (HLR), el *Mobile Switching Centre* (MSC), el *Service Switching Point* (SSP) etc. Así, el SCS hace el papel de un proxy o pasarela para el núcleo de red (ver Figura 1.10). Los servidores de aplicación podrían estar en el mismo dominio del operador como los SCSs o pueden estar en dominios diferentes.

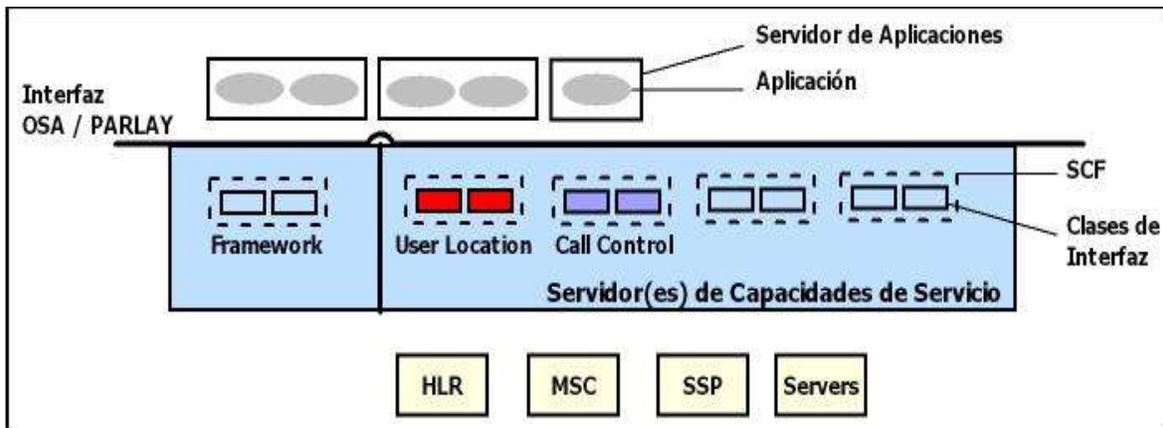


Figura 1.11 Entidades Lógicas Parlay / Osa

Un Servidor de Capacidades de Servicio puede implementar múltiples Características de Control de Servicio (SCF), aunque generalmente se hable de un SCS por SCF. Debido a que un SCS es una entidad lógica que no necesita ser implementado como un componente separado.

Un elemento de gran importancia en PARLAY / OSA es el Framework, quien ofrece a las aplicaciones las diferentes capacidades, esta entidad es una parte esencial para facilitar la interoperabilidad, descubrimiento e integración de nuevas características.

1.3.4.3 El papel del FRAMEWORK PARLAY / OSA

Uno de los requerimientos claves de los operadores de red y proveedores de servicios desde el inicio de Parlay / Osa ha sido asegurar que la apertura de la red a través de una API no exponga la infraestructura de comunicaciones a un uso no autorizado o algún otro

peligro. Esta es la función del Framework PARLAY / OSA, resolver un cierto número de problemas de seguridad y disponibilidad. El Framework se implementa en la gateway Parlay/Osa. Todas las aplicaciones y servicios que quieren usar las APIs de Parlay / Osa primero deben registrarse en el Framework. El Framework es un componente software que autentica la aplicación, y le retorna referencias a objetos para aquellas funciones Parlay/Osa (o Características de Capacidades de Servicio, SCF) que el proveedor de servicios ha permitido utilizar. Este Framework consiste de una familia de interfaces con un núcleo conformado por:

- Gestión Confiable y Segura: autenticación de dominios.
- Registro SCF: registro de nuevos SCF en el Framework.
- Fábrica de SCF: creación de instancias de SCF.
- Descubrimiento de SCF: descubrimiento de SCFs provistos por el operador.

Adicionalmente posee una familia de interfaces para:

- Gestión Integrada: balance de carga, gestión de fallos.
- Notificación de Eventos: notificación eventos específicos (registro nuevos SCS).
- Gestión de Contratos: entre diferentes dominios (entre proveedores de aplicaciones y operadores de red).

La concepción de la API Parlay/Osa busca calidad en el servicio, escalabilidad, desempeño, disponibilidad, redundancia y robustez.

1.3.4.4 Ciclo de vida de las aplicaciones

Una aplicación obtiene los servicios de la gateway Parlay / Osa en su fase de inicio por medio de solicitudes hacia el framework. Durante la fase de finalización de la aplicación, ésta, liberará todos los servicios previamente solicitados, en la fase intermedia la aplicación puede manejar múltiples transacciones. El ciclo de vida para las aplicaciones

tiene básicamente los siguientes pasos:

- Obtener acceso al framework.
- Obtener acceso a los servicios necesarios.
- Manejar múltiples transacciones (llamadas telefónicas, mensajes, etc) usando los servicios.
- Liberar los servicios utilizados.
- Finalizar el acceso al framework.

Cuando la aplicación realiza el descubrimiento de un servicio, es posible que esta reciba múltiples instancias del mismo tipo de servicio. Especificando las propiedades del servicio, puede seleccionarse la instancia correcta de algún tipo de servicio.

1.3.4.5 Service Level Agreements (SLA)

El Framework es un gatekeeper (portero) entre la aplicación y la red de telecomunicaciones. La aplicación puede estar localizada en el dominio del operador, así como también en cualquier lugar en Internet. Si el desarrollador no tiene un acuerdo con el operador, no le estará permitido usar los servicios de la gateway Parlay / Osa. Junto con el operador se necesita definir los *datos de autenticación y autorización*. Cuando ya se tiene ese acuerdo con el operador, dicho acuerdo será denominado Service Level Agreement, SLA, (Acuerdo de Nivel de Servicio). El manejo de las transacciones pueden continuar mientras el SLA no expire o hasta que el SLA sea terminado (por la aplicación o por el operador). La figura 1.12 muestra como el framework la aplicación y los servicios se comunican unos con otros.

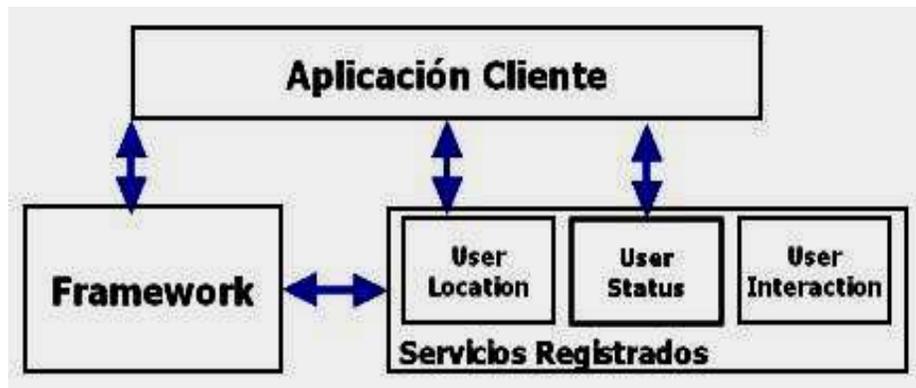


Figura 1.12 Framework

Primero, la aplicación se autentica en el framework (y viceversa). Cuando el framework y la aplicación han validado sus identidades mutuamente, la aplicación puede realizar una solicitud al framework para tener acceso a los servicios mediante la suscripción del SLA. Cuando el SLA se ha establecido el framework retorna la referencia al administrador del ciclo de vida de la aplicación. La aplicación en ese momento se encuentra habilitada para usar esta referencia para obtener acceso al servicio directamente.

Aunque una gateway Parlay / Osa pueda contener varios servicios, esto no significa que a una aplicación se le permita usarlos (como se muestra en la figura 1.12). El operador especifica en los SLA cuales servicios pueden ser utilizados y que tipo de aplicaciones tienen acceso a ellos. Los SLA se definen por la relación aplicación-servicio. Están disponibles dos tipos de propiedades SLA, propiedades *basadas en funcionalidad* y propiedades *basadas en desempeño*. La aplicación puede actuar solamente dentro de los límites del SLA. Las propiedades basadas en funcionalidad son la base para un SLA. El SLA afirma que valores para las propiedades funcionales puede usar una aplicación. Las propiedades en el SLA se chequean en tiempo de ejecución y la gateway Parlay / Osa solamente permitirá acciones dentro de los límites del acuerdo. Los siguientes son ejemplos de propiedades basadas en funcionalidad:

- Los tipos de *triggers* permitidos.
- Cuales métodos de la API están permitidos: por ejemplo llamadas permitidas iniciadas por la aplicación.

- Método de carga permitido.
- Tipos de mensajes permitidos que pueden ser manejados.

Las propiedades basadas en desempeño definen cuanta capacidad de la gateway Parlay / Osa se le permite utilizar al desarrollador en su aplicación. Algunos ejemplos de estas propiedades son:

- Número máximo de llamadas por segundo.
- Número máximo de transacciones en curso.
- Número máximo de mensajes (por ejemplo mensajes SMS) por segundo.

1.3.4.6 Acceso a los Servicios

Antes de que una aplicación pueda tener acceso a un servicio, debe autenticarse a si misma en la gateway Parlay / Osa, la aplicación necesita seguir una secuencia determinada de acciones que es necesario que se ejecuten antes de poder usar un servicio específico. Para utilizar un servicio la aplicación debe realizar las siguientes acciones principales:

- Obtener acceso inicial al framework.
- Autenticar la aplicación en el framework.
- Descubrir los servicios disponibles.
- Seleccionar el servicio que se desea usar.
- Inscribir el SLA (Acuerdo de Nivel de Servicio) para usar el servicio.

De acuerdo a las especificaciones de Parlay / Osa, la aplicación necesita llamar – recibir un mínimo de 13 métodos en total para conseguir acceso al primer servicio que se desea

usar. Para cada servicio extra que la aplicación necesite utilizar, será necesario llamar – recibir cinco métodos por cada servicio adicional.

1.3.4.7 HOSA

Las APIS HOSA (High Level Open Services Access) son extensiones a Parlay / Osa que proveen servicios y funcionalidades adicionales. Estas APIs son propietarias y pueden ser desarrolladas por las empresas de acuerdo a las necesidades que quieran cubrir y los servicios que deseen ofrecer. La compañía Ericsson ha trabajado en el desarrollo de estas APIs para su gateway Parlay / Osa, con ellas se pueden obtener los siguientes servicios:

- Recibir SMS y MMS y enviar SMS, MMS, WAP-Push e E-Mail por medio del *Messaging Service* (Servicio de Mensajería).
- Leer E-Mail usando el *Message Retrieval service* (servicio de Recuperación de Mensajes).
- Solicitar el estado de los teléfonos móviles por medio del *User Status service* (servicio de Estado de Usuario).
- Manejar los contactos del suscriptor usando el *PIM Contact service*.
- Manejar la agenda del suscriptor usando el *PIM Calendar service*.

Se puede notar que estas API permiten trabajar con dos tecnologías muy importantes mencionadas anteriormente, el SMS y el WAP-Push, por lo tanto son una herramienta bastante útil para los desarrolladores que desean implementar aplicaciones y servicios que necesiten hacer uso de esas tecnologías y que no desean tener que trabajar directamente con la red subyacente de los operadores móviles. Para el caso del presente trabajo estas APIs proporcionan las interfaces necesarias para el desarrollo de la aplicación requerida.

1.4 HERRAMIENTAS O PLATAFORMAS ELEGIDAS Y JUSTIFICACIÓN

Después del análisis presentado en este capítulo sobre los tipos de servicios factibles de ser implementados en nuestro entorno, y las herramientas que para ello se tiene, se decide que los servicios basados en mensajería son los que mejor se adaptan a las necesidades del prototipo que se desea implementar, ya que la principal información que se quiere presentar al usuario son señales de alarma que se pueden describir en textos bastante reducidos, por lo tanto el intercambio de información entre la aplicación y el dispositivo móvil se hará por medio de mensajes cortos SMS.

A pesar de que la tecnología J2ME cuenta con una API para mensajería, se descarto su utilización debido a que hasta el momento de desarrollo de este trabajo los dispositivos habilitados para J2ME que ofrecen los operadores nacionales son todavía escasos y bastante costosos, este es un inconveniente técnico y económico bastante importante ya que uno de los objetivos del servicio es que el tipo de terminal móvil no sea uno de los principales factores que influyan al momento de utilizar el servicio. Además para nuestro caso particular J2ME no ofrece ventajas significativas en el desarrollo de una aplicación que se encargue en el dispositivo móvil de gestionar las señales recibidas desde el servidor, ya que es precisamente en el lado del servidor donde se realizará la mayor cantidad de trabajo y procesamiento de información del servicio a desarrollar.

Con respecto a la tecnología WAP PUSH se ha visto que es una de las mejores opciones en cuanto al tratamiento de información que se puede generar de manera aleatoria como es el caso de las señales de alarmas, el inconveniente encontrado en el uso de esta tecnología es que se debe tener acceso directo al centro de mensajería para poder explotar todas las capacidades que WAP PUSH ofrece y esto actualmente es algo complicado ya que este recurso no es de libre acceso a terceras personas, y hasta el momento no existe ningún tipo de convenio o acuerdo entre la Universidad y los operadores inalámbricos, que permita a estudiantes tener la facilidad de acceso al SMSC para realizar pruebas de validación para las aplicaciones. Claro esta, se puede desarrollar un servicio y realizar las pruebas de validación a nivel de simuladores, pero como se manifestó cuando se planteo la realización de este proyecto, uno de los intereses

principales de sus desarrolladores es que este trabajo no quede solamente en un importante ejercicio académico sino que además estuviera lo mas cerca posible de ser implementado en un ambiente real mas allá de los simuladores.

Para poder desarrollar el prototipo de nuestra aplicación sin tener que acceder directamente al centro de mensajería se opta por trabajar con Parlay/Osa, ya que actualmente la compañía Ericsson tiene un Kit de desarrollo de servicios Parlay/Osa¹. Este SDK cuenta con una API desarrollada en java que facilitan el desarrollo de las aplicaciones, como se menciono anteriormente cuenta con unas interfaces especiales para el manejo de mensajería que es la base de nuestro servicio y permiten la comunicación entre un aplicativo residente en un servidor de aplicaciones y el terminal móvil del usuario, además tiene herramientas de simulación de la red GSM, la gateway Parlay/Osa y el simulador del terminal móvil. Después de depurar y tener listo el prototipo tendrá mas factibilidad de probarse en un entorno real ya que en Colombia, el operador OLA cuenta con una Gateway Parlay/Osa, y es mas fácil poder llegar a acuerdos de acceso a esta Gateway que al centro de mensajería SMSC, además al momento de realización de este trabajo se tiene conocimiento de que la Universidad del Cauca esta trabajando en el desarrollo implementación y / o adecuación de un SMSC a nivel de laboratorio con el que se facilite realizar las pruebas que se necesitan en varios proyectos relacionados con aplicaciones móviles y que sea independiente de los operadores de red.

¹ <http://www.ericsson.com/mobilityworld/sub/open/technologies/parlay/index.html>

2. ANÁLISIS Y DISEÑO DEL SISTEMA DE MONITOREO DE VARIABLES AMBIENTALES Y CONTROL DE ALARMAS PARA EL SISTEMA ARIADNA MEDIANTE UN DISPOSITIVO MÓVIL

2.1 DEFINICIÓN DE REQUERIMIENTOS

2.1.1 Descripción del Problema

En el proyecto ARIADNA se tiene un sistema que esta compuesto por sensores los cuales suministran una información que se envía a una estación de monitoreo que cuenta con un sistema SCADA, con el que se registran los datos obtenidos por los sensores para alimentar una base de datos. Cuando se produce una irregularidad significativa en la información registrada, los diferentes procesos de análisis de datos propios del sistema ARIADNA hacen que ésta condición se visualice como una alarma en una consola de administración, donde permanentemente se requiere una persona encargada de la verificación de la información recibida por el sistema y de tomar las medidas pertinentes cuando se presente una situación anormal en la lectura.

El principal inconveniente de este sistema es que los cambios en la información que genera alarmas es poco frecuente, pero ocurren de manera totalmente aleatoria por lo que es imposible poder predecir en que momento pueden presentarse los estados de alarma para que en ese instante una persona se encargue de la supervisión y tome las medidas necesarias, es por eso que el trabajo de supervisión se debe hacer permanentemente, trabajo que es ineficiente debido a que pueden presentarse prolongados periodos de tiempo en los cuales no se produzca ningún cambio que genere alarma, lo ideal sería realizar dicha supervisión y control únicamente cuando los estados de alarma así lo requieran. Por lo tanto, se puede notar que aunque la recolección,

registro y análisis de datos en el sistema ARIADNA se hace en tiempo real y de manera oportuna, no sucede lo mismo en el proceso final de notificación y presentación de información al usuario o administrador, ya que se presenten o no situaciones de alarma o alerta, debe permanecer ligado al lugar donde se encuentre la estación de trabajo con la que pueda acceder a la consola de administración, esto referido principalmente a las señales de alarma ya que la consola de administración es la herramienta principal de gestión del sistema y algunas tareas difícilmente podrían realizarse de otra forma.

2.1.2 Definición de Requerimientos

Se requiere un sistema que sea capaz de gestionar una Base de Datos en un servidor desde un dispositivo móvil (teléfono celular). La Base de Datos será la encargada de almacenar los datos del sistema ARIADNA(estaciones, sensores y alarmas) que a su vez podrá tomar datos para modificar su funcionamiento si es el caso.

El sistema debe estar en capacidad de entender los datos obtenidos de la Base de Datos, clasificarlos en datos normales o en datos anormales los cuales son especificados como alertas; cuando una alerta se produce se enviará un mensaje (SMS) a un celular específico indicando información precisa del hecho que la provoco.

Se deberá tener tres tipos de usuarios, el primero recibe la información de las alertas y tomará las medidas necesarias, también podrá obtener información específica de las estaciones, sensores o alarmas; el segundo tipo de usuario será el administrador móvil que además de los servicios anteriores podrá modificar desde el móvil datos específicos de la Base de Datos y existirá un tercer usuario que será el encargado de realizar la gestión del sistema desde consola, su trabajo básicamente esta enfocado a establecer los parámetros de configuración del sistema y realizar la gestión de los usuarios que utilizarán el sistema.

El sistema debe enviar la información de alerta al usuario indicado hasta que este confirme que ha recibido el mensaje, si pasado un tiempo no se ha recibido la confirmación porque el usuario no puede confirmar, no recibió el mensaje, el teléfono se encuentra fuera de servicio o no esta dentro de la cobertura, el sistema debe estar en

capacidad de enviar el mensaje a otro usuario y así hasta recibir confirmación o crear un registro que especifique que el mensaje de alarma se envió pero no fue confirmado.

2.2 DESCRIPCION GENERAL DEL SISTEMA

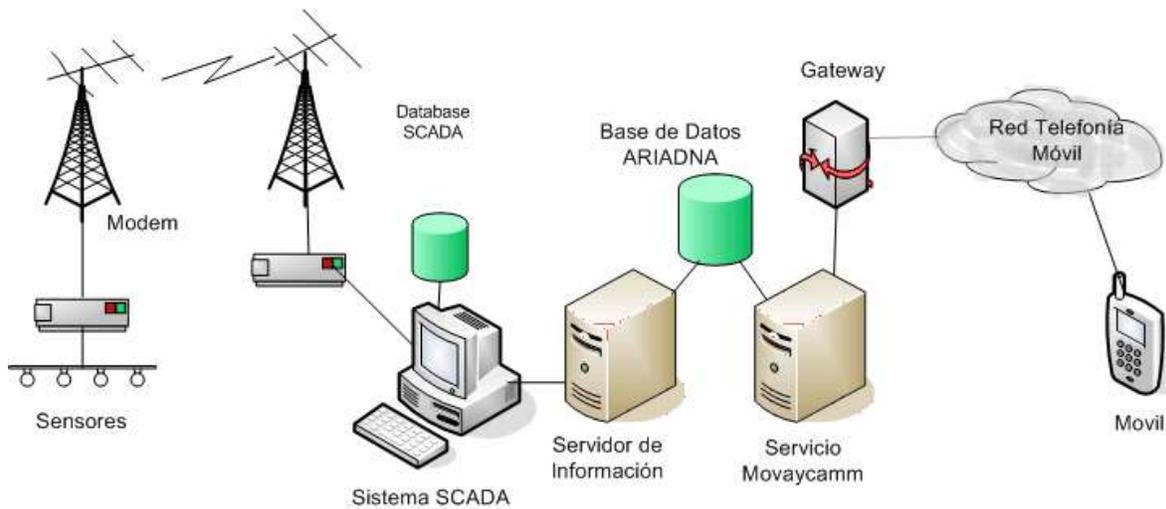


Figura 2.1 Arquitectura del sistema ARIADNA Móvil

El propósito del sistema es monitorear permanentemente la Base de Datos del sistema ARIADNA en la cual se encuentra registrada toda la información del sistema de monitoreo ambiental (estaciones, sensores, alarmas), el sistema compara los datos que pueden provocar alarma con parámetros que nos indican si existe o no una alerta, en el momento en que se presente una alerta el sistema realizará el proceso de envío de mensajes (SMS) al teléfono asignado hasta que el usuario móvil confirma que ha recibido el mensaje, también se enviará un mensaje cuando la alerta se haya controlado.

El sistema trabaja mediante el envío de mensajes cortos utilizando el SMSC (Centro de Servicios de Mensaje Cortos) y se puede adecuar para trabajar con cualquier empresa prestadora del servicio de telefonía móvil celular.

Los usuarios del sistema pueden solicitar información desde su móvil cuando lo deseen pero solo el usuario supervisor podrá modificar algunos parámetros previamente definidos.

2.3 ÁRBOL DE FUNCIONES

1. Administrador

1.1 Acceder al sistema

1.2 Gestión de usuarios

1.2.1 Agregar usuario

1.2.2 Eliminar usuario

1.2.3 Modificar usuario

1.2.4 Consultar usuarios

1.3 Gestión del Sistema Movaycamm

1.3.1 Configurar número telefónico de servicio

1.3.2 Asignar número telefónico de contacto para las alarmas

1.3.3 Asignar número telefónico de reenvío de mensajes

1.3.4 Establecer persistencia de los mensajes de alarma

2. Supervisor

2.1 Acceder al sistema

2.2 Modificar parámetros sistema ARIADNA

2.2.1 Modificar tiempos de lectura en las estaciones.

2.2.2 Modificar y/o establecer valores mínimos y máximos para los sensores

2.3 Gestionar alarma

2.3.1 Enviar mensaje de confirmación de mensaje de alarma recibido.

3. Usuario

3.1 Acceder al sistema

3.2 Consultar información

3.3 Gestionar alarma

3.3.1 Confirmar recepción de mensajes de alarma

4. Sistema Ariadna

4.1 Registrar datos

2.4 CASOS DE USO DE ALTO NIVEL SISTEMA MOVAYCAMM

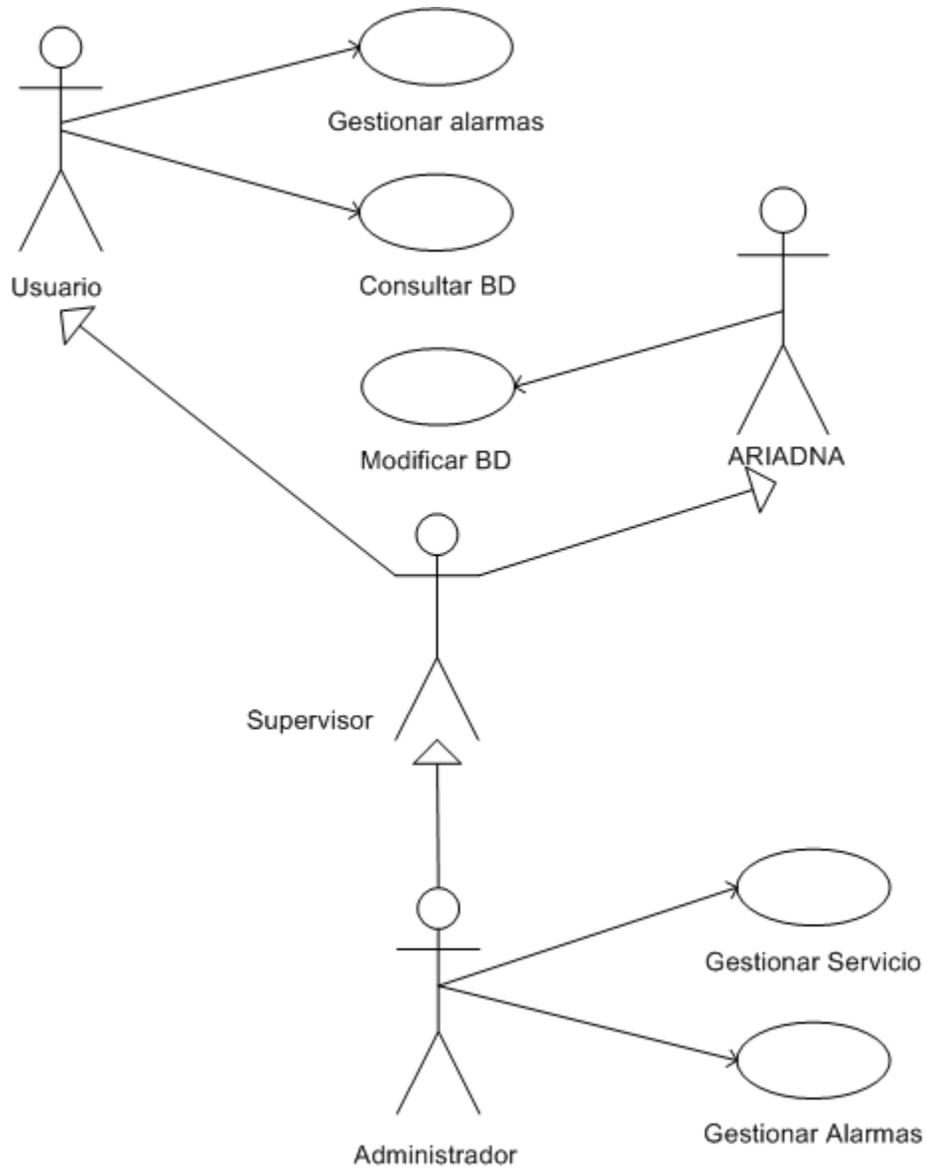


Figura 2.2 Diagrama de casos de uso de alto nivel el sistema Movaycamm

2.4.1 Descripción de los Casos de Uso de Alto Nivel

A continuación se hace una descripción del rol de cada uno de los usuarios que hacen parte del sistema: Administrador, supervisor, usuario.

- **Administrador:** Es el encargado de gestionar todo el sistema, debe realizar los procesos de creación, eliminación, actualización y consulta de usuarios, también debe establecer los parámetros de configuración y funcionamiento del sistema Movaycamm. Es responsable de establecer lo que el sistema debe hacer en caso de presentarse una alarma.
- **Supervisor:** Este usuario esta habilitado para realizar gestión del sistema ARIADNA de manera remota desde su dispositivo móvil, esta gestión hace referencia a la consulta de información y modificación de algunos parámetros en la BD del servidor del sistema ARIADNA.
- **Usuario:** Las funciones para este tipo de usuarios están enfocadas a la consulta de información del sistema por medio de un dispositivo móvil. Solo puede acceder a aquella información que se ha establecido como publica o de conocimiento general. En algunos casos estos usuarios están facultados para recibir los mensajes de alarma que se producen en el sistema ARIADNA y que son procesadas por el sistema Movaycamm, este usuario no puede realizar ningún tipo de modificación.
- **Sistema ARIADNA:** Es el encargado de registrar en la base de datos los eventos de alarma y situaciones anormales que se puedan generar en el sistema ARIADNA y que deben ser notificadas al usuario final para tomar las medidas necesarias.

2.4.1.1 C.U. Gestión de Usuarios

Actor: Administrador

Tipo: Primario

Descripción:

El administrador es el encargado de crear e ingresar los nuevos usuarios, quienes pueden tener tres tipos de perfiles: Administrador, Supervisor o Usuario. Para cada nuevo usuario se deben registrar en la base de datos el nombre, perfil de usuario, código de acceso al sistema, y número telefónico. El administrador es también el responsable de consultar,

modificar y eliminar algún usuario.

2.4.1.2 C.U. Gestión del Sistema Movaycamm

Actor: Administrador

Tipo: Primario

Descripción:

El administrador debe establecer unos parámetros básicos y primordiales para que el sistema Movaycamm pueda funcionar, posteriormente esos parámetros deben ser administrados de acuerdo a las necesidades particulares que se presenten. En el sistema Movaycamm deben especificarse y establecerse como datos básicos el número telefónico asignado al servicio, este número será al que se enviarán las solicitudes desde el teléfono móvil, también se debe especificar el número telefónico de los usuarios del sistema a quienes se les pueda enviar las notificaciones que el sistema genere. Además es posible gestionar parámetros como la persistencia de los mensajes de alarma, definir algún tipo de prioridades para las mismas, establecer un número telefónico de reenvío de mensajes para aquellos casos en los que no se confirme la recepción de los mensajes de alarma.

2.4.1.3 C.U. Modificar Parámetros

Actor: Supervisor

Tipo: Primario

Descripción:

el Supervisor desde su móvil puede realizar procesos de gestión y control del sistema ARIADNA mediante la modificación y / o configuración de algunos parámetro específicos en la base de datos, de esta forma puede cambiar los niveles de comparación en los sensores o el tiempo de lectura para las estaciones.

2.4.1.4 C.U. Acceder al Sistema

Actor: Supervisor, Usuario

Tipo: Primario

Descripción:

Para que un usuario pueda acceder al sistema desde su dispositivo móvil debe registrarse por medio de un código de acceso que fue asignado por el administrador en el proceso de registro de usuarios, y que debe ser validado de acuerdo a la información almacenada en la base de datos del sistema.

2.4.1.5 C.U. Confirmar Recepción de Alarmas

Actor: Supervisor, Usuario

Tipo: Primario

Descripción:

Cuando el Sistema Movaycamm envía un mensaje de alarma a un usuario del sistema, espera una confirmación de ese usuario para saber que el mensaje fue recibido, por lo tanto el usuario ante un mensaje de alarma siempre debe enviar un mensaje de confirmación al sistema, de lo contrario el Sistema Movaycamm seguirá enviando los mensajes de alarma o procederá al reenvío de los mismos a un número telefónico alternativo, si no se recibe confirmación el sistema registrará el hecho en la base de datos.

2.4.1.6 C.U. Consultar Información

Actor: Supervisor, Usuario

Tipo: Primario

Descripción:

Por medio del dispositivo móvil los usuarios del sistema pueden realizar consulta de información específica correspondiente a estaciones, sensores y alarmas en la BD del Sistema ARIADNA. El usuario obtiene como respuesta del sistema un mensaje de texto con la información solicitada

2.4.1.7 C.U. Consultar información Menú de voz

Actor: Usuario, Usuario

Tipo: Primario

Descripción:

El usuario puede realizar consultas a la BD del Sistema ARIADNA marcando el número telefónico del servicio y navegando a través de un menú de voz que guía al usuario por diversas opciones que le facilitan la consecución de alguna información particular. El usuario obtiene como respuesta del sistema un mensaje de texto con la información solicitada.

2.4.1.8 C.U. Registrar Datos

Actor: Sistema ARIADNA

Tipo: Primario

Descripción:

En el sistema ARIADNA se están monitoreando varios aspectos concernientes a las estaciones y sensores que hacen parte del sistema, los datos obtenidos se registran en una BD con la cual trabaja el Sistema Movaycamm.

2.5 DIAGRAMA DE CLASES DEL SISTEMA

2.5.1 Diagrama de clases principales del sistema Movaycamm

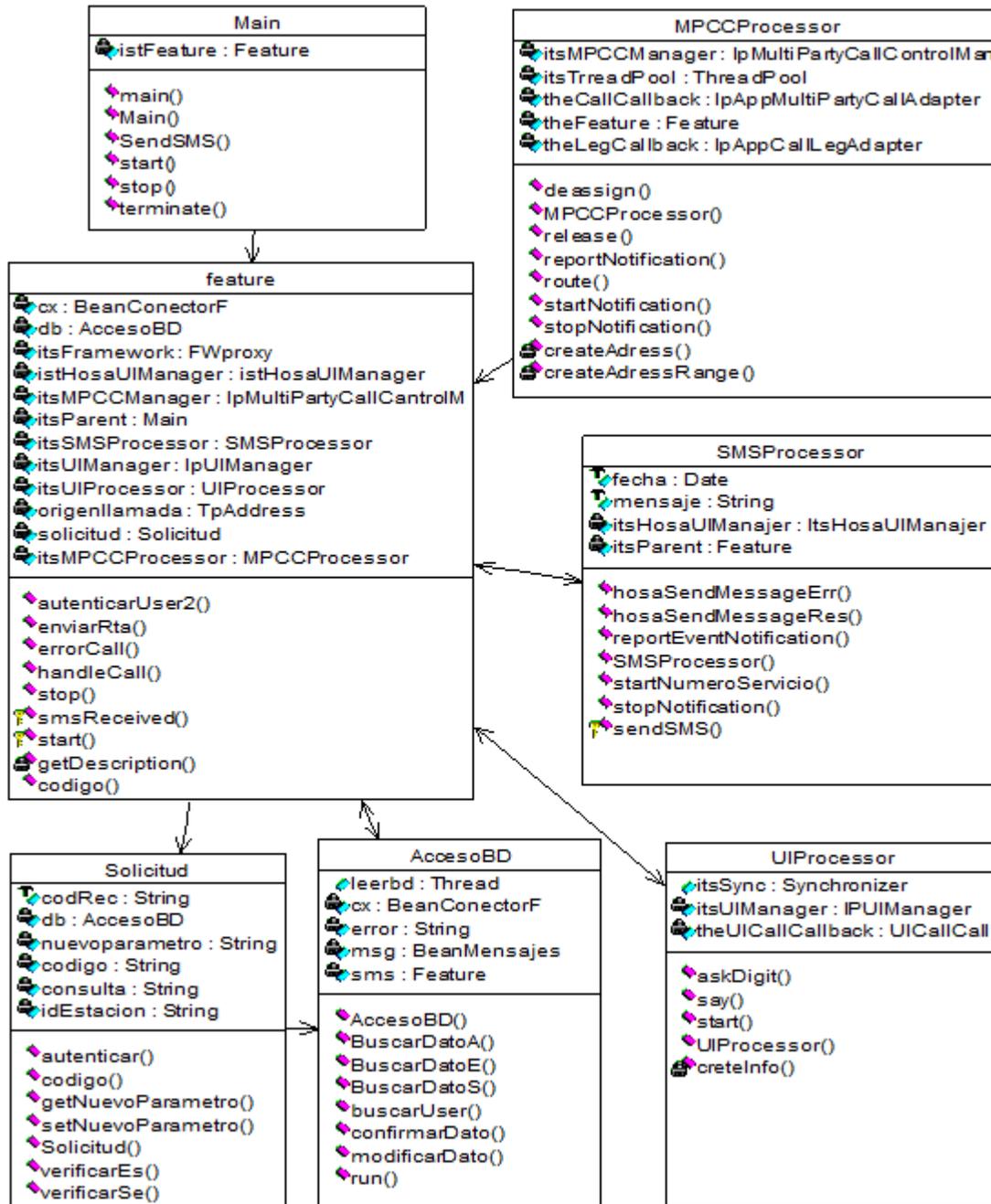


Figura 2.3 Diagrama de clases principales del sistema Movaycamm

El sistema Movaycamm consta de 7 clases principales:

2.5.1.1 Clase Main.

Es la clase de inicio del sistema encargada de crear la GUI principal del sistema la cual permite iniciar, parar o ingresar a la GUI de configuración del servicio. Despliega la siguiente interfaz.

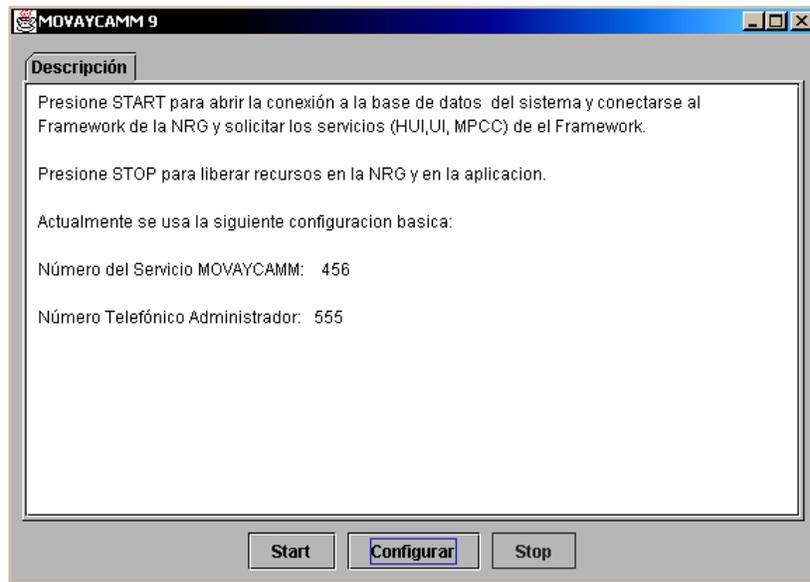


Figura 2.4. Interfaz que despliega la clase Main

Los métodos mas importantes de la clase Main son:

- **Start**, comienza la aplicación en si, accesa los métodos de la clase Feature para activar los recursos que requiere la aplicación como la conexión con la red de telecomunicaciones (el simulador NRG de Ericsson) y con la base de datos.
- **Stop**, libera los recursos utilizados y finalizar la aplicación.

2.5.1.2 Clase Feature.

Es una de las clases mas importantes ya que maneja toda la lógica de la aplicación, a

través de ella se realizan todos los procesos que tienen que ver con el inicio de comunicación con la base de datos, el manejo de llamadas, la recepción de mensajes y las respuestas a los diferentes tipos de solicitudes del usuario móvil.

Los métodos mas importantes de la clase Feature son:

- **AutenticarUser**, se utiliza para autenticar los usuarios cuando las consultas se hacen a través de una llamada, si el código ingresado es correcto podrá continuar, de lo contrario enviara un mensaje de voz que notifica el error.
- **EnviarRta**, envía un SMS al usuario con la respuesta de la información solicitada cuando esa solicitud se hace por el menú de voz.
- **ErrorCall**, cuando la interacción con el servicio Movaycamm es por medio de una llamada, valida las respuestas al menú de voz, si la respuesta es correcta se recibirá la respuesta, de lo contrario se escuchara un mensaje de información no valida.
- **HandleCall**, Este método es el encargado de procesar las solicitudes de voz (las llamadas) al servicio Movaycamm, coloca los diferentes anuncios: los menús, mensajes de error, de operación exitosa de registro etc. La operación final de este método es enviar al usuario un SMS con la información solicitada.
- **Codigo**, Este método verifica la longitud de los SMS que se van a enviar al móvil, si la longitud del mensaje es mayor a 120 caracteres lo divide en las partes necesarias y envía varios mensajes.
- **SmsReseived**, Este método es el encargado de recibir los mensajes que llegan al servicio, realizar su procesamiento y enviar un SMS de respuesta.
- **Start**, da inicio a la conexión con la red de telecomunicaciones a través de la NRG, y también invoca la clase que se encarga de la comunicación con la base de datos.

- **Stop**, libera todos los recursos utilizados por la aplicación y termina la aplicación.
- **GetDescription**, se encarga de poner un breve texto descriptivo que se muestra en la GUI de inicio.

2.5.1.3 Clase SMSProcessor.

Es la encargada de realizar todas las tareas que involucran el procesamiento de los SMS, tales como identificar la llegada de un SMS al servicio, enviar SMSs a los móviles, generación de errores en caso de fallo en el envío. Los métodos más importantes de la clase SMSProcessor son:

- **hosaSendMessageErr**, este método es llamado por el NRG cuando en el mensaje enviado existe un error.
- **HosaSendMessageRes**, este método es llamado por el NRG para enviar un mensaje como respuesta a una petición.
- **ReportEventNotification**, llamado por el NRG cuando se recibe cualquier notificación de recepción de un mensaje para el servicio Movaycamm.
- **StartNumeroServicio**, Configura el número telefónico con el que se identificará el servicio Movaycamm, para la recepción de mensajes desde el dispositivo móvil, en nuestro caso desde el simulador NRG.
- **SendSMS**, método que se comunica directamente con la NRG para el envío de un mensaje SMS.
- **createE164Address**, este método se encarga de transformar un determinado número telefónico al formato o tipo de datos que define las especificaciones de Parlay / OSA para las direcciones de los destinos móviles.

2.5.1.4 Clase MPCCProcessor

El trabajo de esta clase consiste básicamente en identificar las llamadas que se hacen al servicio Movaycamm y crear una sesión para cada una de ellas que permita manejarlas de manera independiente. Los métodos mas importantes de la clase MPCCProcessor son:

- **ReportNotification**, este método crea una serie de sesiones y permite manejar cada llamada por separado, se crea una sesion por cada llamada y se libera cuando la llamada termina. Permite acceder el NRG lo mas rápido posible.
- **StartNotifications**, Este método se comunica directamente con la NRG solicitando una notificación cuando se realiza una llamada al servicio Movaycamm.

2.5.1.5 Clase UIProcessor.

Cuando se crea una sesion para una llamada esta clase se encarga de mantener la interacción con el usuario móvil activando los mensajes de voz que tiene el servicio y reconociendo y recolectando la información que ingresa el usuario desde su teléfono móvil.

- **AskDigit**, activa un mensaje de voz a modo de petición diciéndole al suscriptor que llama que escriba un dígito, recoge el dígito que envía el usuario y devuelve la respuesta. Si no se recibe ninguna respuesta en el plazo de 20 segundos, devuelve un valor nulo que indica que el usuario no ingreso ninguna información.
- **Say**, activa un mensaje de voz a modo de información, ya que contrario del método anterior no se espera respuesta por parte del usuario.
- **Start**, da inicio a la interacción con el suscriptor que llama creando un identificador para cada sesión.

2.5.1.6 Clase AccesoBD

Es la clase principal en lo referente a la conexión, desconexión, monitoreo y todos los procesos de interacción (consulta, modificación) con la Base de Datos de ARIADNA.

- **Run**, ejecuta un Thread que se encarga del monitoreo continuo de la base de datos inspeccionando los campos que pueden generar alertas.
- **BuscarDatoX**, por medio de un identificador ingresado por el usuario realiza la búsqueda de este dato en la base de datos.
- **ModificarDato**, por medio de un identificador y otros datos ingresados por el usuario realiza la modificación de algún parametro en la base de datos.
- **ConfirmarDato**, cuando el usuario confirma la recepción de un mensaje de alarma este método se encarga de que el servicio no siga enviado mensajes.

2.5.1.7 Clase Solicitud

Procesa las diferentes solicitudes provenientes del usuario móvil, ya sea por medio de mensajes SMS o por medio de una llamada al servicio Movaycamm. Para las solicitudes SMS verifica que la trama sea correcta e identifica el tipo de operación que el usuario desea realizar y deja listo al servicio para que procese esta petición y pueda generar la respuesta solicitada.

- **Autenticar**, verifica que el código de usuario ingresado sea correcto.
- **Código**, Este método procesa y verifica la trama ingresada desde el móvil, la cual puede tener hasta 5 partes separadas por un comodín " * " y las almacena en un arreglo de string. Identificando el tipo de operación que el usuario quiere realizar.
- **VerificarIds**, cuando la interacción con el servicio Movaycamm se hace por medio de una llamada este método verifica que los identificadores de estaciones sensores y alarmas existan y sean correctos.

2.6 DIAGRAMA DE PAQUETES SISTEMA MOVAYCamm

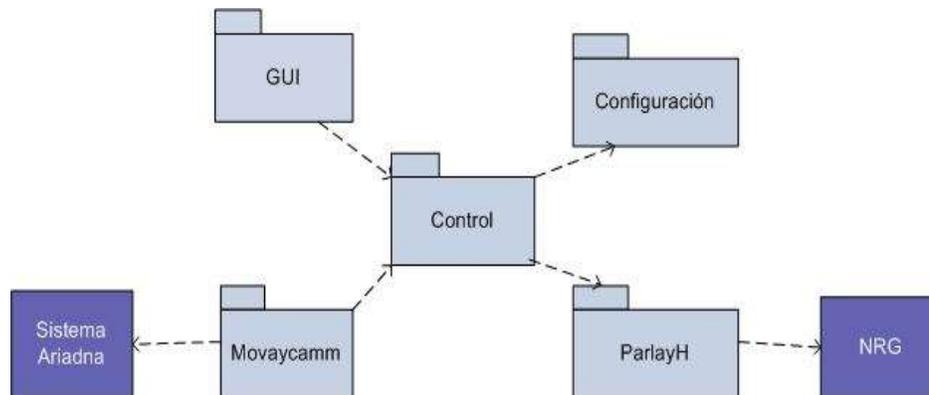


Figura 2.5 Diagrama de paquetes sistema Movaycamm

El sistema cuenta con dos entidades externas, la primera es el Sistema Ariadna el cual se encarga de recoger y procesar toda la información correspondiente a sensores y estaciones, antes de registrarlos en la base de datos con la cual trabaja el sistema Movaycamm, la segunda es el NRG, la gateway Parlay / OSA quien comunica el servicio con toda la infraestructura de la red de telecomunicaciones móviles necesaria para que el servicio llegue hasta el usuario móvil. Estas entidades se comunican con el servicio por medio de paquetes específicos que a su vez interactúan entre si, para desarrollar y ejecutar todas las funcionalidades que es sistema Movaycamm provee. A continuación se hace la descripción del diagrama de paquetes que se representa en la Figura 2.5.

GUI. Paquete encargado de la representación grafica de todas las interfase a través de las cuales el usuario Administrador interactúa con el sistema para realizar la puesta en marcha del servicio y para las tareas de gestión.

Movaycamm. Este paquete es el encargado de todos los procesos relacionados con la interacción con la base de datos del Sistema Ariadna, por medio de él se realiza el monitoreo de las señales de alarma que gestiona el servicio Movaycamm además de todas aquellas funciones relacionadas con consultas y modificaciones.

Configuración. El sistema Movaycamm necesita configurar ciertos aspectos para poder ejecutarse de manera adecuada, estos aspectos y todos aquellos que se hacen por medio de las interfases de configuración se desarrollan por el trabajo directo de este paquete.

ParlayH. Este paquete es el encargado de realizar todos los procesos que tienen que ver con la aplicación de las interfaces y APIs suministradas por Parlay / OSA y HOSA para la comunicación con la gateway, por medio de las clases pertenecientes a este paquete se logra el acceso a todos los servicios de la red de telecomunicaciones que son accedidos a través de la gateway Parlay / OSA.

Control. Este paquete es el que implementa toda la lógica del servicio, determina que tarea o proceso debe desarrollarse, en que momento debe hacerse y que actor externo o método interno debe ejecutarlo.

3. PRUEBAS Y RESULTADOS

La funcionalidad del sistema Movaycamm comprende cuatro aspectos fundamentales que son: la configuración del servicio, la notificación de las alarmas que se presentan en el sistema Ariadna, los diferentes tipos de consultas desde el dispositivo móvil y los procesos de gestión que se realizan desde el dispositivo a través de las modificaciones de algunos parámetros específicos en la base de datos del sistema Ariadna. Cuando la interacción con el servicio Movaycamm se hace a través de mensajes SMS, este mensaje tiene un formato bien definido que es de la siguiente forma: una trama que puede tener un máximo de 5 campos separados por el comodín "*", la trama tiene este formato:

CodAcceso*operacion*Id*parametro*valorNuevo Donde

- **CodAcceso:** corresponde al código del usuario que desea acceder al sistema.
- **Operación:** puede ser una de las siguientes letras:
 - ✓ "E" ó "e" : indica que se hará consultas referentes a estaciones del sistema Ariadna
 - ✓ "S" ó "s": indica que se hará consultas referentes a sensores.
 - ✓ "A" ó "a": indica que se hará consultas referentes a alarmas.
 - ✓ "M" ó "m": indica que se realizará modificaciones.
 - ✓ "C" ó "c": indica que se realizará la confirmación de recepción de un mensaje de alarma.
- **Id:** Corresponde al identificador de la estación, sensor o alarma sobre la que se realizará la operación deseada.
- **Parámetro:** en las operaciones de consulta, es el elemento que se desea

consultar, en las operaciones de modificación indica el parámetro que se modificará, y cuando la operación es de confirmación este parámetro debe ser la palabra “ok”.

- **ValorNuevo:** este valor solo es necesario para las operaciones de modificación e indica el valor nuevo.

Para verificar que estos aspectos funcionarán de la manera esperada se realizarán las pruebas que se nombran a continuación:

3.1 CONFIGURACIÓN DEL SERVICIO

Para que el servicio pueda ejecutarse, este debe tener como mínimo configurado el número telefónico que identifica al servicio y el número telefónico del administrador y el código de acceso del administrador, si estos no han sido configurados, los valores por defecto del sistema son:

- Número telefónico del servicio 333
- Número telefónico del administrador 555
- Código de acceso del administrador 1111

Prueba 1

Objetivo: ingresar a la GUI de configuración del sistema usando los valores por defecto del sistema, crear usuarios para el sistema y configurar los parámetros para las alarmas que se desea gestionar.

Procedimiento

- Ejecutar la aplicación del servicio, sin iniciarlo ingresar a la interfaz de validación de usuario por Configuración e ingresar con el código de acceso por defecto para el administrador.

- Se modifica el código de ingreso para el administrador.
- Se agregan los siguientes usuarios: un supervisor, y un usuario.
- En la interfaz de gestión de alarmas se configura todos los parámetros para tres sensores
- Para verificar las modificaciones anteriores, se realiza una operación de tipo consulta, tanto en la pestaña de gestión de usuarios, como en la interfaz de gestión de alarmas.

Resultados:

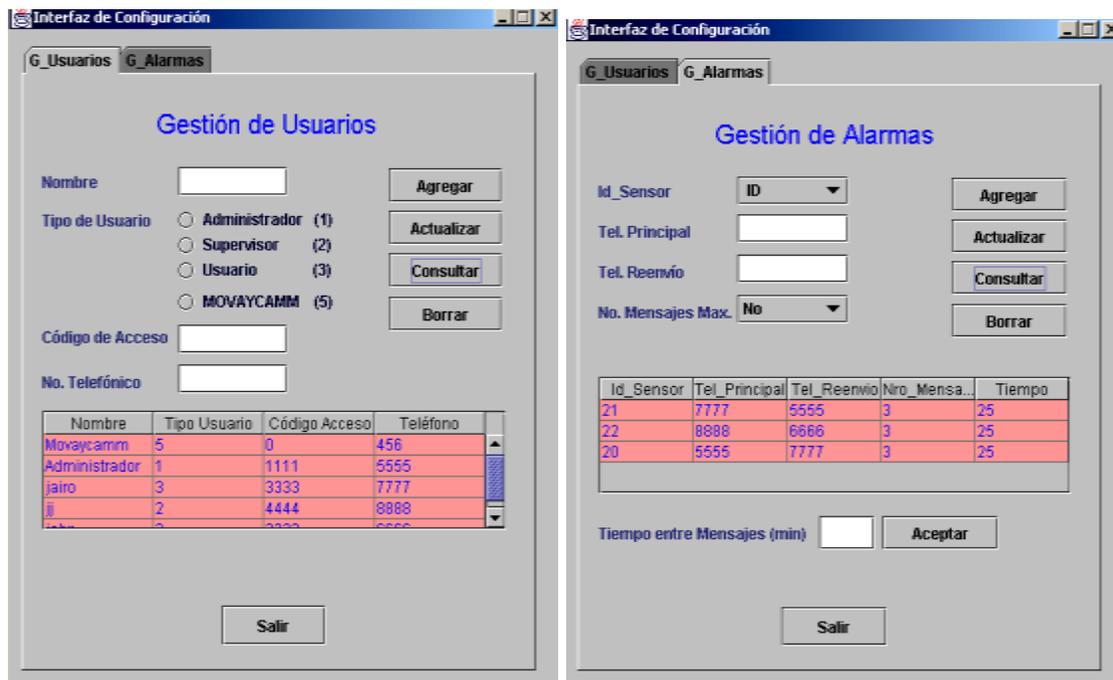


Figura 3.1 Resultados de la prueba de configuración del servicio

3.2 NOTIFICACIÓN DE ALARMAS

Después de la prueba anterior se pone en ejecución el servicio, mediante el botón "Start" de la GUI principal, desde este momento se esta monitoreando la base de datos. Para

realizar esta prueba se debe simular una situación de alerta que se registra en la base de datos.

Objetivos:

- Verificar que se envié la notificación de alarma cuando esta se produce en el sistema Ariadna a los teléfonos principal o de reenvió que se ha configurado previamente.
- Comprobar el correcto funcionamiento del proceso de confirmación de alarma recibida.

Desarrollo de la prueba: Pasos,

- Se simula una alarma en la base de datos de Ariadna para el sensor 20.
- Se elimina la situación de alarma.

Resultados.

- Se recibe la alarma en el teléfono principal que es el 555.
- Después del segundo mensaje se reenvía el mensaje de alarma al número alternativo que para este sensor es el número 777.
- Después del segundo mensaje a este teléfono el envío de mensajes termina y se crea un registro informando que no hubo confirmación a los mensajes.
- Cuando la alarma se elimina se envía un mensaje de "ALARMA 20 controlada" al teléfono principal que se configuro para esta alarma que es el 555.

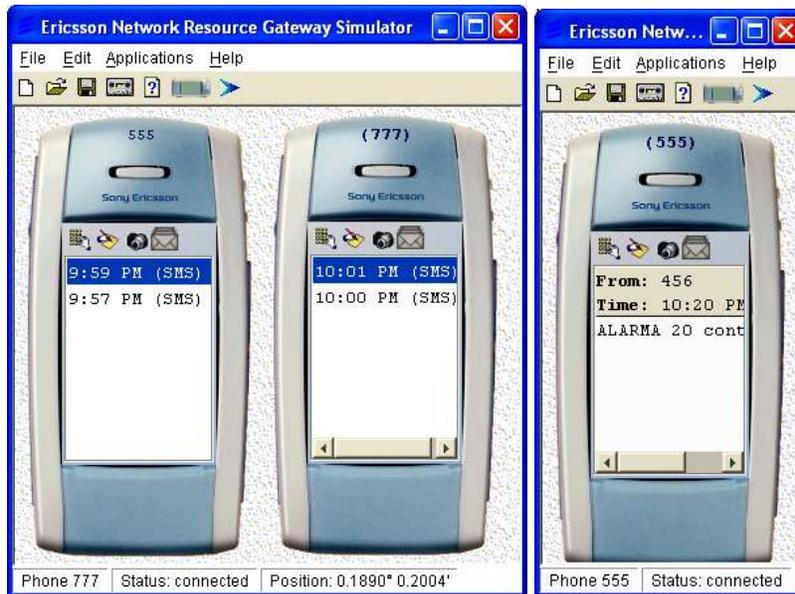


Figura 3.2 Mensajes de alarma enviados al número principal y de reenvío

Se repite la prueba pero ahora con los siguientes pasos:

- Se simula una alarma en la base de datos de Ariadna para el sensor 20.
- Después de recibir el mensaje en el móvil 555 se envía el mensaje de confirmación que tiene el siguiente formato 2222*c*20*ok desde el teléfono móvil al código 456, que es el número del servicio Movaycamm.
- Se elimina la situación de alarma.

Resultados

- Se recibe la alarma en el móvil 555.
- Después de la confirmación el envío de mensajes cesa.
- Cuando la alarma se elimina se envía un mensaje de “alarma controlada” al teléfono principal que se configuro para esta alarma que es el 555.



Figura 3.3 Envío del mensaje de confirmación

3.3 CONSULTAS

Los procesos de consulta y modificaciones se pueden realizar por dos métodos diferentes, uno por medio del envío de un SMS al código 456 del servicio Movaycamm y otro por medio de una llamada telefónica al mismo número.

3.3.1 Consultas mediante solicitudes SMS

Esta prueba consiste en consultar el estado del sensor 20. se asume que quien va a realizar la consulta es el usuario "Jairo" cuyo teléfono es el 777 y quien tiene código 3333.

Objetivos: Comprobar que los procesos de consultas se realizan de manera adecuada.

Procedimiento: Desde el teléfono 777 Se envía el siguiente mensaje SMS al código 456:
3333*s*20*estado

Resultados



Figura 3.4 Envío del mensaje de consulta y recepción del SMS de respuesta

La respuesta obtenida es Sensor: 20 Estado a, lo que me indica que el sensor esta activo.

Del mismo modo se realizaron las siguientes consultas:

Para Estaciones:

- Consulta de parámetros generales de todas las estaciones. SMS enviado: 3333*e
- Consulta de parámetros generales de la estación 21. SMS enviado: 3333*e*21
- Consulta de tipo de la estación 21. SMS enviado: 3333*e*21*tipo
- Consulta de estado de la estación 21. SMS enviado: 3333*e*21*estado.
- Consulta de tiempo de lectura de la estación 21. SMS enviado: 3333*e*21*tiempol

Para sensores:

- Consulta de parámetros generales de todos los sensores. SMS enviado: 3333*s

- Consulta de parámetros generales del sensor 20. SMS enviado: 3333*s*20
- Consulta de estado. SMS enviado: 3333*s*20*estado
- Consulta de Valor máximo. SMS enviado: 3333*s*20*valormax
- Consulta de Valor mínimo. SMS enviado: 3333*s*20*valormin
- Consulta de Variables asociadas. SMS enviado: 3333*s*20*var

Para Alarmas:

- Consulta de parámetros generales de todas las alarmas. SMS enviado: 3333*a
- Consulta de parámetros generales de la estación. SMS enviado: 3333*a*10
- Consulta de Id_sensor de la alarma 10. SMS enviado: 3333*a*10*ids
- Consulta de valor. SMS enviado: 3333*a*10*valor
- Consulta de estado. SMS enviado: 3333*a*10*estado
- Consulta de Fecha de inicio. SMS enviado: 3333*a*10*fechai

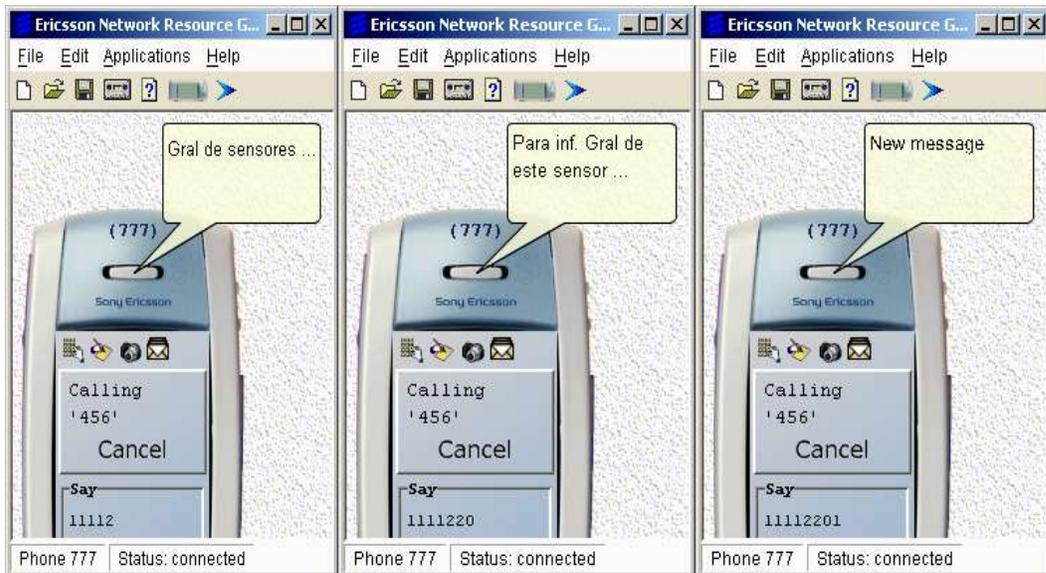
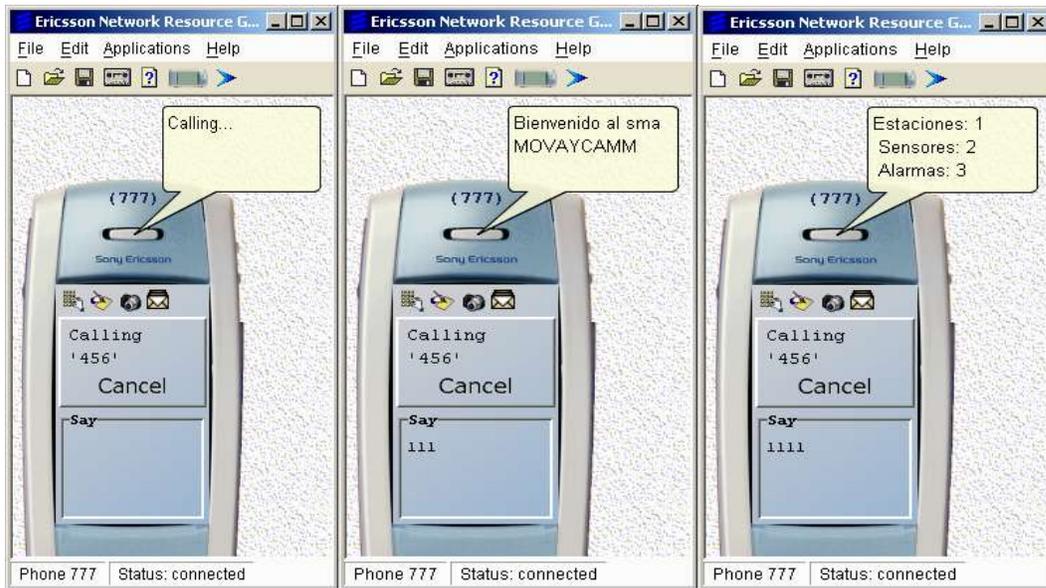
3.3.2 Consultas mediante llamada telefónica al servicio Movaycamm

Se realiza la misma operación que en el numeral anterior, se consulta el estado para el sensor 20, pero ahora se realiza una llamada al servicio discando desde el teléfono (simulador) el número 456, que corresponde al número del servicio.

Objetivos: Verificar que el menú de voz conducen de manera correcta al usuario para realizar las consultas que quiere realizar y obtener la información solicitada.

Procedimiento

- Desde el teléfono 777 Se realiza la llamada al sistema Movaycam marcando el número 456.
- En el teléfono se escucha un anuncio que da la bienvenida al sistema y solicita ingresar el código de usuario.
- Se ingresa el código 3333.
- Se escucha un anuncio que presenta cuatro opciones: consulta de estaciones, consulta de sensores, consulta de alarmas y salir del sistema.
- Se elige la opción 2 para consultar sensores.
- Se escucha un anuncio que da las indicaciones para realizar una consulta general de todos los sensores o solicita ingresar el identificador del sensor para realizar la consulta sobre un sensor particular.
- Se ingresa el identificador del sensor, 20.
- Se escucha un anuncio que da las indicaciones para realizar una consulta general de este sensor o para solicitar un solo parámetro de este sensor.
- Se elige la opción 1 que es para consultar el estado.
- Se escucha un anuncio que indica que la solicitud ha sido procesada.
- Se recibe un nuevo SMS
- El servicio nuevamente regresa al menú principal y solicita escoger una opción.
- Se elige la opción 5 para salir del sistema.



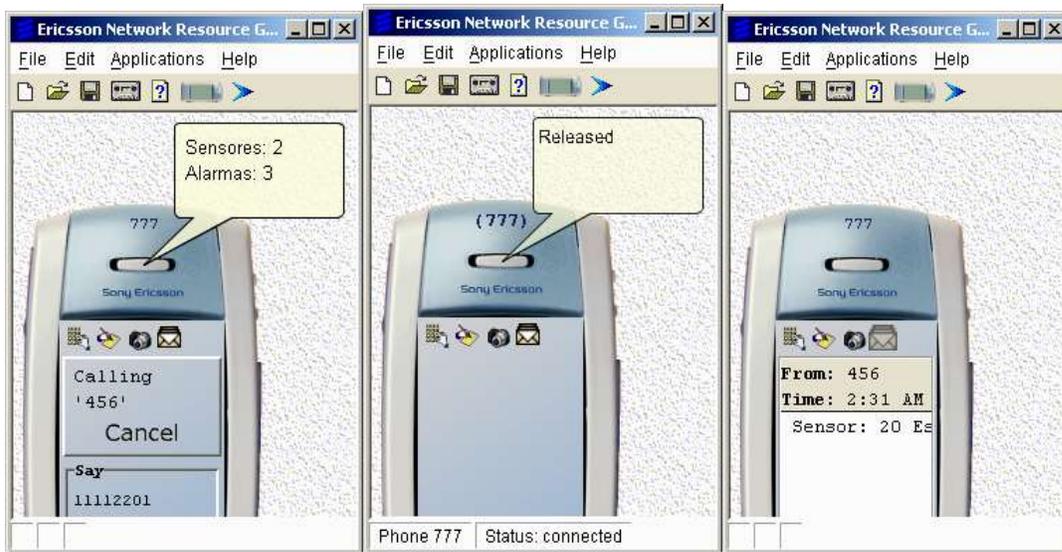


Figura 3.5 Consulta mediante llamada telefónica al servicio Movaycamm

Como resultado se obtiene un SMS con la información solicitada: Sensor: 20 Estado a. Del mismo modo se puede realizar todas las consultas descritas en el punto anterior para estaciones, sensores y alarmas.

3.4 MODIFICACIONES

Como se menciona anteriormente este proceso también se puede hacer a través de mensajes SMS o a través de una llamada telefónica al servicio Movaycamm.

3.4.1 Modificaciones mediante solicitudes SMS

Para esta prueba se va a modificar el tiempo de lectura de la estación 21. las modificaciones desde el dispositivo móvil solo están permitidas para usuarios de tipo administrador, o para supervisores, por esto para esta prueba se ingresara al sistema con el usuario John que es de tipo supervisor según la configuración que se hizo en la primera prueba, el código de ingreso de este usuario es 2222.

Objetivo: Comprobar que mediante el mensaje SMS adecuado enviado desde el teléfono

móvil, se puede cambiar un parámetro específico de la base de datos del sistema Ariadna.

Procedimiento

- Desde el teléfono 123 se realiza una consulta para saber el valor actual del tiempo de lectura de la estación 21, para esto se envía el siguiente SMS al sistema Movaycamm: 2222*e*21*tiempol.
- La respuesta del sistema Movaycamm será: Estación: 30 tiempo lectura 15.



Figura 3.6 Consulta tiempo de lectura

- Se modifica el tiempo de lectura de esta estación de valor 15 a un valor 30 enviando el siguiente SMS: 2222*m*21*tiempol*30.
- Se recibe un SMS que notifica que la operación se ha realizado, el mensaje recibido es: Estación: 21 tiempo lectura 30 parámetro cambiado

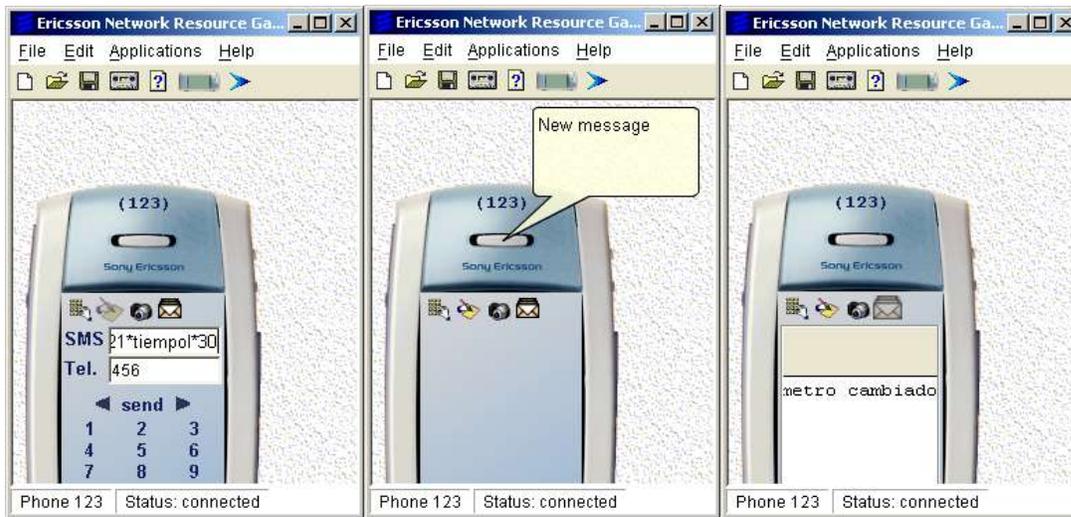


Figura 3.7 Modificación tiempo de lectura

Como resultado se obtiene un SMS que indica que la operación fue realizada. De igual manera se pueden modificar para los sensores los parámetros valor máximo y valor mínimo. Estos son los elementos configurable de manera remota por medio del dispositivo móvil.

3.4.2 Modificaciones mediante llamada telefónica al servicio Movaycamm

Se realiza la misma operación que en el numeral anterior, se modifica el tiempo de lectura para la estación 21, pero ahora se hace una llamada al servicio discando desde el teléfono (simulador) el número 456, que corresponde al número del servicio.

Objetivos:

- Verificar que el menú de voz ofrece la opción al usuario Supervisor para poder realizar modificaciones.
- Comprobar que la operación de modificación asistida por el menú de voz se hace de manera correcta.

Procedimiento

- Desde el teléfono 777 Se realiza la llamada al sistema Movaycamm marcando el número 456.
- En el teléfono se escucha un anuncio que da la bienvenida al sistema y solicita ingresar el código de usuario.
- Se ingresa el código 2222.
- Se escucha un anuncio que presenta cinco opciones: consulta de estaciones, consulta de sensores, consulta de alarmas, modificar datos y salir del sistema.
- Se elige la opción 4 para modificar datos.
- Se escucha un anuncio que permite elegir lo que se desea modificar que puede ser estaciones o sensores.
- Se elige la opción 1 para modificar estaciones.
- Se escucha una opción que solicita ingresar el código de la estación.
- Se ingresa el código 21 para la estación
- Se escucha un anuncio que permite escoger el parámetro a modificar o regresar al menú principal.
- Se elige la opción 1 para modificar el tiempo de lectura.
- Se escucha un anuncio que solicita que se ingrese el nuevo valor.
- Se ingresa el nuevo valor que es 10.
- Se escucha un anuncio que indica que la solicitud ha sido procesada.
- Se recibe un nuevo SMS
- El servicio nuevamente regresa al menú principal y solicita escoger una opción.
- Se elige la opción 5 para salir del sistema.

El proceso anterior se ilustra en la figura 3.8.

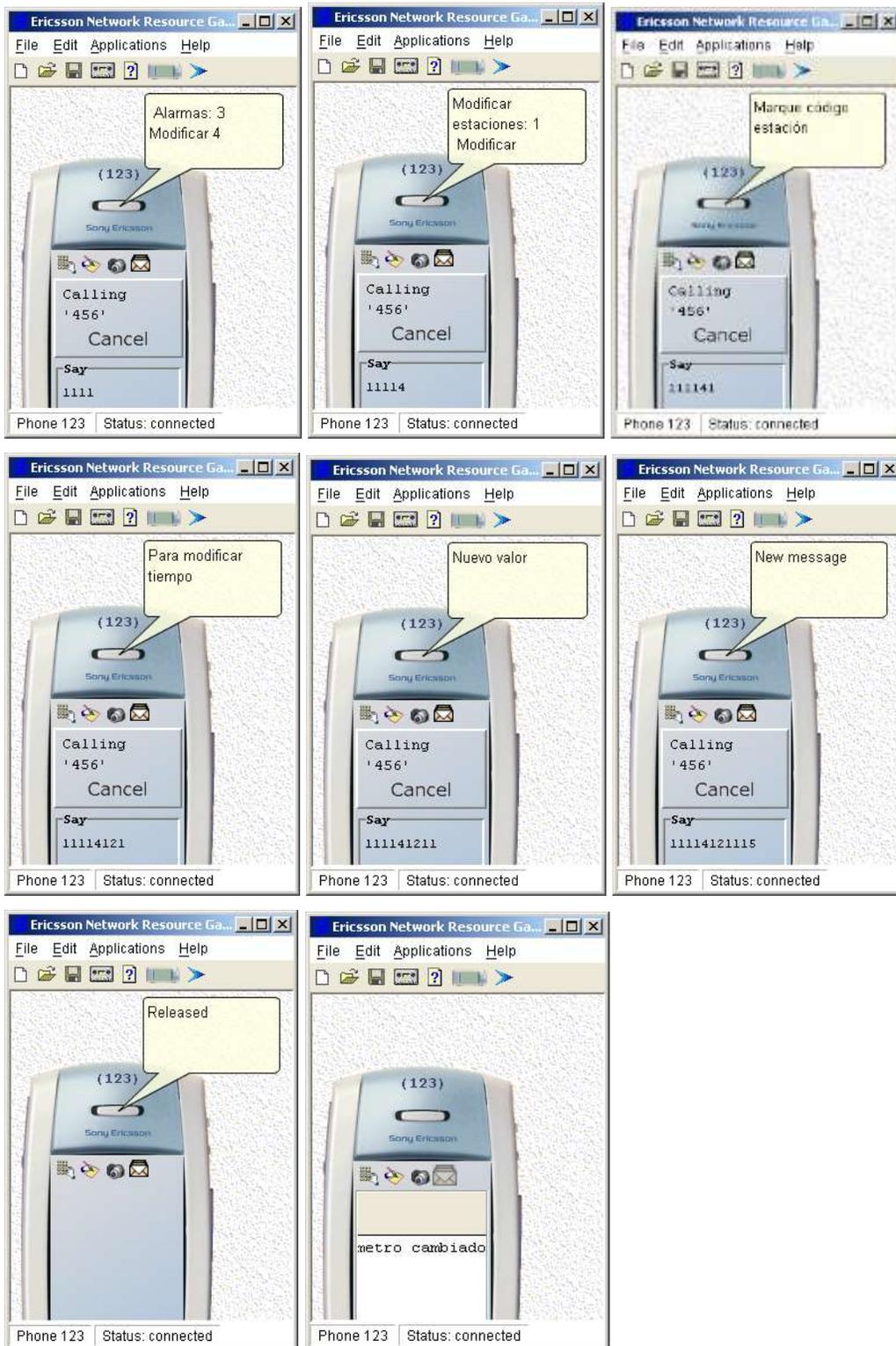


Figura 3.8 Modificación de tiempo de lectura mediante llamada telefónica

Como resultado se obtiene un mensaje SMS que notifica que la operación se realizó con éxito, el mensaje recibido es: Estación: 21 tiempo lectura: 10 Parámetro cambiado. Con el mismo procedimiento se pueden modificar los parámetros mencionados en el punto anterior.

4. CONCLUSIONES Y RECOMENDACIONES

- La telefonía móvil se ha extendido en cifras realmente sorprendentes en periodos de tiempo relativamente cortos en estos últimos años, este es un fenómeno mundial y en nuestro país también se ha dado. El servicio de telefonía móvil hace tiempo que dejó de ser un lujo de unos pocos para convertirse en uno de los medios mas importantes de comunicación, que facilita el desempeño de algunas tareas y que en varios casos es una herramienta de trabajo muy valiosa. Aunque el servicio de voz sigue siendo el servicio líder de los operadores de telefonía móvil, no debe pasar desapercibida el gran número de usuarios móviles que tienen estas redes de telecomunicaciones a quienes sin embargo aun no se les ha ofrecido servicios de valor agregado realmente atractivos tanto por su funcionalidad como por su costo. Es aquí donde se encuentra un entorno bastante apropiado para investigar seriamente sobre las necesidades de estos usuarios y la posibilidad de desarrollar servicios que satisfagan esas necesidades y que de una u otra forma pueden impulsar el desarrollo tecnológico de la región.
- En las redes de telecomunicaciones móviles el servicio de mensajería corta ha tomado una gran importancia en lo que se refiere a la transmisión de información de baja capacidad, esto ha hecho que varios servicios utilicen este medio para transmitir información de una manera sencilla y de fácil acceso a un gran número de usuarios, de esta manera con un servicio útil se puede llegar a una población objetivo bastante grande sin gran cantidad de esfuerzo.
- Existen varios tipos de sistemas, ya sea de tipo informático, electrónico, de telecomunicaciones etc, un aspecto bastante importante es la gestión de alertas o de fallos, su importancia radica en que la información sobre esas eventualidades es de suma importancia y es necesario conocerlas de la manera mas rápida posible para poder tomar algún tipo de acción, la información sobre esas eventualidades es

bastante puntual y específica, es por ello que en este tipo de sistemas las soluciones móviles son una alternativa bastante viable ya que puede aplicarse en el desarrollo de un aplicativo que ayude a resolver la problemática de la notificación de alarmas o fallas en el momento oportuno a la persona o al sistema indicado.

- Hasta hace poco tiempo los operadores de red eran los encargados y responsables directos del desarrollo y soporte de los servicios de valor agregado de sus redes, de esta manera esos servicios siempre estaban en el dominio del operador de red, esto generaba bajo desarrollo de posibles servicios, además si alguien externo deseaba desarrollar servicios de valor agregado que funcionaran en la red de telefonía móvil, tenía que tener amplios conocimientos sobre la red subyacente de telecomunicaciones para poder desarrollar algún servicio, con el desarrollo de tecnologías y estándares que se ocuparon de resolver esta problemática, estos inconvenientes se han visto mermados, logrando que los desarrolladores de servicios y aplicaciones se enfoquen únicamente en los servicios que desean desarrollar; su trabajo consistirá en investigar cual tecnología es mas apropiada para su aplicativo y aprender a utilizarla, esto es lo que sucede con las APIs Parlay / OSA, con la tecnología Wap–Push etc. Las aplicaciones que utilizan Parlay / OSA pueden ser desarrolladas en java ya que la API se ha desarrollado en este lenguaje, así los desarrolladores pueden trabajar en un lenguaje de programación bastante conocido con todas las ventajas y facilidades que este ofrece.
- Desafortunadamente en Colombia aun no se da un buen acercamiento entre los operadores de telefonía móvil y las instituciones y / o empresas interesadas en el desarrollo de servicios de valor agregado y esto retrasa el desarrollo, pruebas y depuración de servicios que pueden tener factibilidad de ser aceptados y aplicados. De manera muy personal creemos que los operadores deben darse cuenta que estos acercamientos son bastante provechosos tanto para desarrolladores como para operadores de red, ya que el desarrollador tendrá la posibilidad de depurar su sistema y ponerlo en ejecución, y el operador tendrá una mayor oferta de servicios en su infraestructura de red, además de que con esos servicios se incrementa el tráfico de información de su red. Esperamos que a corto plazo sea mas fácil poder acceder a los recursos de las redes de telefonía móvil.

- El servicio desarrollado en este trabajo utiliza en su mayor parte las APIs Parlay/OSA, sin embargo se debe tener en cuenta que al trabajar con la NRG, se trabajó con APIs propietarias de la compañía Ericsson, y aunque estas aumentan las funcionalidades que proporciona Parlay / OSA, no dejan de ser un inconveniente si se desea trabajar con una gateway Parlay, por eso en la medida de lo posible es preferible trabajar con tecnologías y / o estándares que proporcionen el mayor grado de flexibilidad y adaptabilidad tanto a plataformas software como hardware, ya que con ello se garantiza que una aplicación o servicio no dependa de una determinada tecnología o de algunos recursos específicos.

GLOSARIO

API: Application Program Interface, es un conjunto de protocolos, rutinas y herramientas para construir aplicaciones software. Un API proporciona elementos básicos que facilitan el desarrollo de programas.

BROWSER WAP: Un browser o navegador WAP es el equivalente al Internet Explorer o Netscape para un PC en un teléfono móvil, permitiendo acceder a un sitio en Internet en su versión WML.

CDC: Connected Device Configuration, configuración para J2ME orientada a dispositivos con pocas limitaciones de procesamiento, memoria e interfaz grafica, tales como decodificadores de televisión digital, televisores con Internet, electrodomésticos, entre otros.

CLDC: Connected Limited Device Configuration, configuración para J2ME orientada a dispositivos dotados de conexión y con limitaciones en cuanto a capacidad grafica, procesamiento, y memoria, tales como teléfonos móviles, PDAs, organizadores personales, etc

CLIENTE WEB: Programa que permite llamar a procedimientos o métodos de un Servicio Web determinado.

CORBA: Common Object Request Broker Architecture, arquitectura que habilita a objetos software para comunicarse con otros objetos sin importar el lenguaje de programación en el que están escritos.

EMS: Enhanced Messaging Service, Servicio de mensajería que aumenta las capacidades de SMS, permitiendo el intercambio de mensajes con imágenes, melodías, animaciones, etc.

FRAMEWORK PARLAY/OSA: Componente software que autentica criptográficamente una aplicación y retorna a esta referencias a objetos para que la NRG pueda prestar sus funciones.

GATEWAY O PASARELA: Hardware o Software que traduce dos protocolos distintos no compatibles.

GSM: Global System for Mobile Communication, Sistema digital de telefonía móvil.

HLR: Home Location Register, base de datos centralizada de la red que almacena y maneja a todos los suscriptores móviles de un operador específico.

HOST: Ordenador conectado a una red y accesible por otros ordenadores.

HTTP: HyperText Transfer Protocol, es un protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP.

J2ME: J2ME es una plataforma desarrollada por Sun, que permite a los desarrolladores usar Java para crear aplicaciones y programas para dispositivos móviles e inalámbricos.

JAX-RPC: API Java para RPC basado en XML. Esta API hace posible escribir aplicaciones Java que usen XML para hacer llamadas a procedimientos remotos (RPC).

MIDlet: Un MIDlet es un programa Java que corre sobre dispositivos móviles que soportan J2ME.

MIDP: Mobile Information Device Profile, Perfil J2ME diseñado para operar con el CLDC y permitir el desarrollo de aplicaciones Java en dispositivos móviles MID (Mobile Information Devices).

MMS: Multimedia Messaging Service, Servicio de mensajería que habilita a los usuarios móviles para el intercambio de mensajes con un alto contenido en multimedia: imágenes animadas, fotos a color, música e incluso videos cortos.

NRG Parlay/OSA: Network Resource Gateway, servidor de capacidades de servicio, a través de las cuales una aplicación utiliza recursos de red de manera segura.

NRG SDK: Conjunto de librerías desarrolladas en Java para facilitar el despliegue de aplicaciones que usan las capacidades de servicio de la NRG Parlay/OSA.

PARLAY/OSA: API para las redes de comunicaciones de nueva generación, la cual provee una capa de abstracción de la red de telecomunicaciones para los desarrolladores de aplicaciones.

SME: Short Messaging Entity, Entidad que puede enviar y / o recibir mensajes cortos.

SMPP: Short Message Peer to Peer, Protocolo usado para la comunicación entre un SMSC y un ESME.

SMS: Short Message Service, es un servicio inalámbrico aceptado globalmente que habilita la transmisión de mensajes alfanuméricos entre usuarios móviles y con sistemas externos tales como correo electrónico, paging y sistemas de correo de voz.

SMSC: Short Message Service Center. El SMSC, es el responsable de la transmisión y almacenamiento del un mensaje corto, entre el SME y una estación móvil.

TCP/IP: TCP/IP es un conjunto de protocolos utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre sí. Los protocolos más importantes son el protocolo TCP y el protocolo IP.

WAP: Wireless Application Protocol, es el estándar que permite a los usuarios acceder a servicios de información instantáneamente desde dispositivos inalámbricos tales como teléfonos móviles.

WAP – PUSH: Es un estándar abierto desarrollado por el WAP Forum para proporcionar las capacidades push que hacen posible nuevas aplicaciones y mejoras impresionantes a los existentes servicios habilitados para push

WML: Lenguaje XML usado para especificar contenido e interfases de usuario en Internet para ser accedido por dispositivos WAP.

BIBLIOGRAFÍA

[http:// www.mincomunicaciones.gov.co](http://www.mincomunicaciones.gov.co)

<http://www.ericsson.com/mobilityworld/sub/open/technologies/parlay/docs>

<http://www.parlay.org>

<http://nmg.upc.es/~mserrano/pdf%20files/%5BCNMT%5DOSA-Parlay%20Spanish.pdf>

3GPP TS 22.060 : General Packet Radio Service (GPRS) ; Service Description ;
Stage 1 (Release 99)

Netsize : El libro blanco del SMS en Europa. Marzo de 2003

Ericsson. ERICSSON NRG SDK PROGRAMMERS GUIDE. 2003 - 2005.

<http://www.ola.com.co>

<http://www.movistar.com.co>

<http://www.comcel.com.co>

<http://www.kannel.org/>

<http://www.wapforum.org>

http://developer.openwave.com/docs/wappush_vs_sms.pdf

Gonzáles Gómez, Juan. El servicio SMS un enfoque practico. Trabajo de doctorado de la asignatura “Nuevas tecnologías para las comunicaciones”. Escuela Técnica Superior de Ingeniería Informática

Las Telecomunicaciones y la movilidad en la Sociedad de la Información. Documento de Internet.

Rendón Gallón Alvaro. EL LENGUAJE UNIFICADO DE MODELADO (UML). Facultad de Ingeniería Electrónica Y Telecomunicaciones. Universidad del Cauca. Popayán. Mayo de 2000.