

APLICACIÓN DE GPRS PARA ARQUITECTURAS DE CONTROL DE ROBOTS

RESUMEN

Se describe un sistema que permite teleoperar un robot a través de teléfonos celulares. Se emplea la red de telefonía celular existente en el país, utilizando el Servicio General de Paquetes por Radiofrecuencia GPRS. De esta manera se permite a los usuarios del sistema un rango más amplio de movilidad. La implementación de dichas aplicaciones se hace mediante el lenguaje de programación JAVA en sus plataformas J2ME para dispositivos móviles y JAVA Servlet para aplicaciones orientadas a servidores. La experiencia se validó con el robot móvil Giraa_02 desarrollado en la Universidad de Antioquia.

PALABRAS CLAVES: GSM, GPRS, Control Remoto, Teleoperación, Robot móvil, J2ME, Servlets.

ABSTRACT

A system that allows the teleoperation of a mobile robot through cellular telephones is described. The net of cellular telephony available in the country is used, by means of the General Packet Radio Service GPRS. This way the users of the system are given a wider range of mobility. The implementation programs of this applications were written in JAVA, namely in J2ME for mobile devices and JAVA Servlet for applications for servers. For the validation of the experience, teleoperation of the mobile robot Giraa_02, developed in the University of Antioquia, was performed.

KEYWORDS: GSM, GPRS, Remote Control, Teleoperation, Mobile robot, J2ME, Servlets.

NELSON DAVID MUÑOZ

Ingeniero Electrónico,
Profesor Depto. Ing. De Sistemas
Universidad de Antioquia
nmunoz@udea.edu.co

NELSON LONDOÑO OSPINA

Ingeniero Electrónico,
Profesor Depto. Ing. Eléctrica
Universidad de Antioquia.
nlondono@udea.edu.co

HARVEY GIRALDO*

Ingeniero Electrónico,
Unidad Edificios,
Empresas Públicas de Medellín.
hgiralar@eppm.com

JUAN FERNANDO EUSSE *

gabeusse_epm@epm.net.co

LUIS GABRIEL MURILLO*

murillo1@gmail.com

EDGAR JULIÁN CRUZ*

juliancruz@epm.net.co

ALEJANDRO ECHEVERRI*

alejoecheverryh@yahoo.com

Estudiantes de Ingeniería
Electrónica, Universidad de
Antioquia

1. INTRODUCCIÓN

El sistema de control de navegación de un robot móvil, permite la realización parcial o totalmente autónoma de operaciones tales como navegación, planificación de tareas, seguimiento de trayectorias, percepción sensorial, evasión de obstáculos, etc. Estas funciones deben de contemplarse de forma integrada en una arquitectura de control, en la que se incluya también la posibilidad de teleoperación y funciones de control de bajo nivel. En el caso de la teleoperación, se sustituiría la generación de trayectorias del control automático por el control manual de un operador, que puede ser en el espacio cartesiano [1].

Los sistemas de control y monitoreo remoto han sido implementados paulatinamente con características de conectividad, directamente con acceso a Internet y últimamente en las redes de telefonía celular, en particular mediante los servicios de transmisión de datos que sobre ella se ejecutan, debido a las ventajas que

ofrece en características como alcance, velocidad y movilidad [2].

El objetivo general de este trabajo es realizar una exploración con estas nuevas tecnologías de la información con el fin de teleoperar un robot móvil como el Giraa_02, mediante la implementación de una aplicación para ejecutar en teléfonos celulares, probando dos métodos de operación de dicho robot. El primero consiste en el envío de ordenes aisladas, es decir, por accionamientos individuales, mientras que en el segundo se envía una secuencia preprogramada al servidor y esta se va ejecutando mediante temporizadores predefinidos.

La implementación se realiza en una red GSM (Global System for Mobile Communications) de telefonía móvil que opera en la banda de 1900 MHz, sobre el Servicio General de Paquetes por Radiofrecuencia GPRS (General Packet Radio Service), además se utiliza un teléfono celular de gamma media alta y el software para desarrollo y simulación de aplicaciones del fabricante

Nokia. La programación tanto de las aplicaciones cliente y servidor se realiza en lenguaje JAVA.

2. J2ME Y JAVA SERVLET

2.1 J2ME (Java 2 Micro Edition)

Se definió como un subconjunto del lenguaje JAVA orientado a la programación de dispositivos móviles, debido a las limitadas capacidades de procesamiento y almacenamiento de éstos. Fue presentada en 1999 por la Sun Microsystems como una nueva versión de la JVM (Java Virtual Machine) que podía ejecutarse en dispositivos Palm.

Esta versión del lenguaje utiliza la KVM (Kilo Virtual Machine), que obtiene su nombre debido a que solo necesita de unos pocos Kilobytes de memoria para ser implementada, además, posee servicios de la máquina virtual reducidos tales como el recolector de basura y las 37 clases básicas incluidas en el paquete. Además en el J2ME existen bibliotecas llamadas perfiles, las cuales son clases específicas (una clase es el elemento fundamental de JAVA), orientadas a implementar funcionalidades de más alto nivel en familias específicas de dispositivos [3]. Sun ha desarrollado: el Perfil para Dispositivos de Interfaz Móvil MIDP, el Perfil RMI, el Personal y el PDA, los cuales son enfocados a diferentes dispositivos y tecnologías [3]. También se da la inclusión de paquetes opcionales como es el de encriptación de datos para conexiones HTTP que agregan funcionalidad al paquete J2ME.

Estos perfiles utilizan configuraciones, las cuales son el conjunto mínimo de APIs Java que permiten desarrollar aplicaciones para un tipo determinado de dispositivos, implementando dos tipos diferentes de máquinas virtuales: La Máquina Virtual Kilo KVM (Kilo Virtual Machine) y la Máquina Virtual Compacta CVM (Compact Virtual Machine). En la figura 1 se observa el flujo de ejecución de la aplicación Java en el equipo.

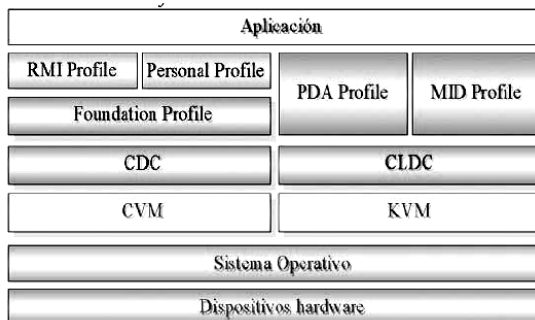


Figura 1. Flujo de ejecución de la aplicación JAVA (J2ME)

Un entorno de ejecución de J2ME se compone de una selección de:

- a. Máquina Virtual
- b. Perfil

- c. Configuración
- d. Paquetes Opcionales

2.1.1 Configuraciones y VM (Virtual Machine): Las principales configuraciones usadas por J2ME son la Configuración de Dispositivos Limitados Conectados CLDC (Connected Limited Device Configuration) y la Configuración de Dispositivos Conectados CDC (Connected Device Configuration). La primera es la más implementada debido a que se enfoca en dispositivos con muchas limitaciones en cuanto a memoria, procesamiento y capacidad gráfica, pero con conexión a redes de datos [3]. Esta configuración usa la KVM que es una máquina virtual liviana, pensada específicamente para este tipo de dispositivos; un ejemplo son los teléfonos móviles, buscaperonas (pagers), PDAs, organizadores personales.

La CDC se implementa para dispositivos con cierta capacidad computacional, de memoria y gráfica. La CDC usa la CVM, que es una implementación compacta de la maquina virtual de Java Standard Edition con limitaciones en el apartado gráfico y de memoria del dispositivo [4].

En la figura 2 se observa una comparación de los tamaños de las diferentes máquinas virtuales de JAVA.

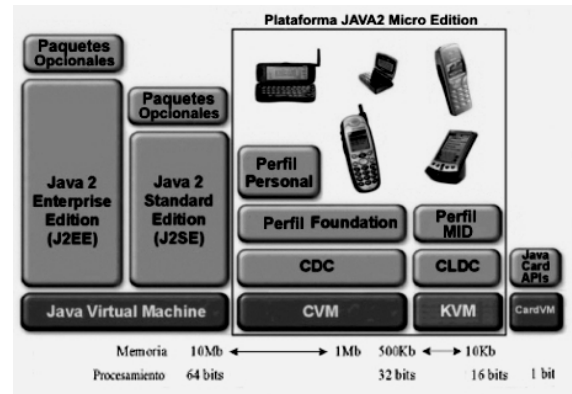


Figura 2. Comparación de las máquinas virtuales de Java.

2.1.2. MIDlets (Mobile Information Device lets): Son las aplicaciones que ejecutará el dispositivo, su nombre proviene por herencia de los otros tipos de aplicaciones desarrolladas en Java (SERVlets (Server), APPlets (Applicated)). Básicamente son programas codificados en Java que pueden ser ejecutados por una VM determinada dependiendo del perfil y la configuración con que se definan.

En el proceso de desarrollo de un MIDlet deben ejecutarse los siguientes pasos:

1. Desarrollo
2. Compilación
3. Preverificación
4. Empaquetamiento
5. Ejecución

6. Depuración

Este proceso de desarrollo es seguido por todas las aplicaciones excepto por las etapas de empaquetamiento y preverificación que es exclusivo de las aplicaciones desarrolladas usando la plataforma J2ME.

A la hora de crear la aplicación JAVA para el dispositivo móvil, se debe utilizar la clase MIDlet perteneciente al paquete javax.microedition.midlet. Todo MIDlet tiene un ciclo de vida básico mientras se ejecuta, determinado por las acciones que realiza el usuario sobre la aplicación. La figura 3 muestra el ciclo de vida en ejecución de un MIDlet. El ciclo empieza cuando se ejecuta la aplicación, y por lo tanto se invoca el método startApp(). El método pauseApp() se invoca cuando el usuario selecciona la opción Pause del celular, o cuando se recibe una llamada o un mensaje. Finalmente el método destroyApp() se invoca cuando se selecciona la opción Close o cuando se genera una excepción.

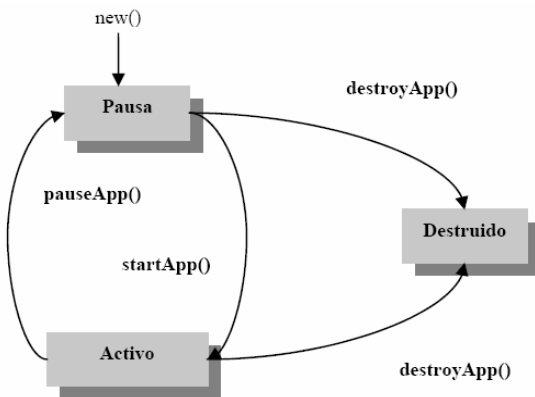


Figura 3. Ciclo de vida de un MIDlet en ejecución.

Además de la clase MIDlet se deben utilizar otras clases para realizar la interfaz de usuario, el apartado gráfico y la comunicación a implementar en la aplicación.

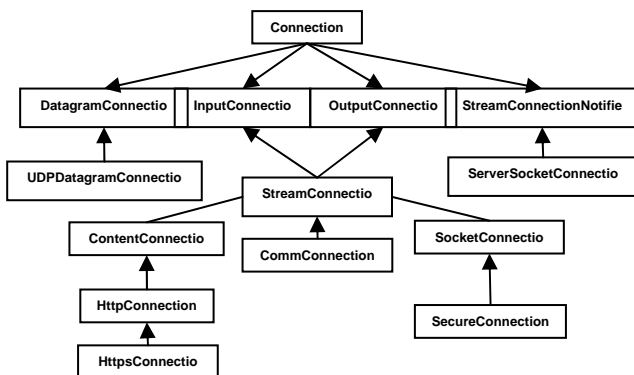


Figura 4. Paquete javax.microedition.io

Para crear la interfaz gráfica para el usuario se utilizan las clases proporcionadas por el paquete javax.microedition.lcdui (interfaz de usuario con pantalla LCD). Para crear y gestionar la comunicación de datos se utilizan las clases proporcionadas por el paquete

javax.microedition.io (entrada y salida para J2ME) y java.io (entrada y salida en aplicaciones JAVA). La figura 4 muestra el conjunto de clases del paquete javax.microedition.io.

2.2. Java Servlet

Es un subconjunto de clases que complementan el paquete J2SE (Java 2 Standard Edition), diseñado específicamente para escribir aplicaciones Server Side (De lado del servidor). Provee a los desarrolladores un mecanismo simple y consistente para extender la funcionalidad de un servidor Web y para facilitar el acceso a sistemas ya existentes. El esquema cliente-servidor basado en Servlet se muestra en la figura 5.

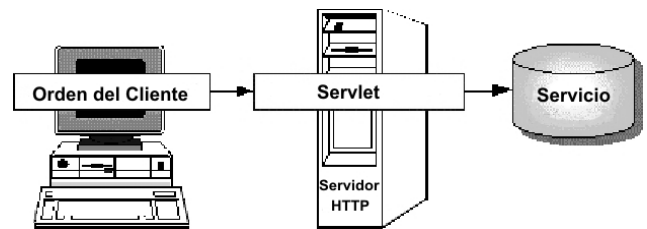


Figura 5. Esquema de comunicación de un servlet.

Los Servlets son un reemplazo efectivo para los scripts CGI. Proporcionan una forma de generar documentos dinámicos fáciles de escribir y de rápida ejecución. Usando las Interfaces de Programación de Aplicaciones API (Application Programming Interface) de los Servlets, los cuales son extensión estándar de Java, los Servlets evitan el problema de desarrollar aplicaciones específicas que dependan de la plataforma [5].

3. REDES CELULARES

3.1 GSM

GSM es un Sistema Global para las Comunicaciones Móviles que constituye un estándar mundial para teléfonos móviles digitales. Es abierto, no propietario y evolutivo (aún en desarrollo). Es el estándar predominante en Europa, así como el mayoritario en el resto del mundo (En 2001 alrededor del 70% de los usuarios de teléfonos móviles en el ámbito mundial, usaban GSM) [6]. En la figura 6 se observa el comportamiento de este mercado durante los últimos años.

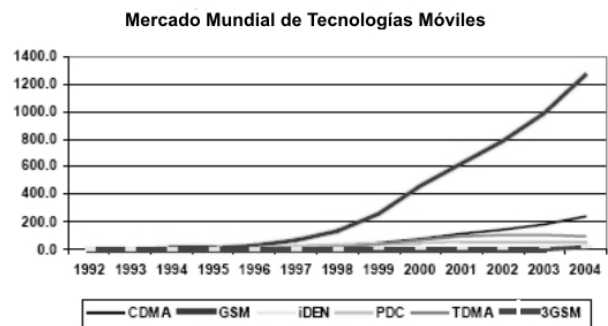


Figura 6. Mercado mundial de tecnologías móviles.

GSM difiere de sus antecesores principalmente en que tanto los canales de voz como las señales son digitales. Se ha diseñado así para un moderado nivel de seguridad. Entre los diferentes servicios que el sistema GSM ofrece, se utilizó el servicio GPRS, en este proyecto.

3.2. GPRS

Es considerada la generación 2.5, entre la segunda generación (GSM) y la tercera (UMTS). Proporciona altas velocidades de transferencia de datos, especialmente útil para conectar a Internet a través de protocolos como los TCP/IP, X.25, y CLNP (Connectionless Network Protocol).

GPRS es sólo una modificación de la forma de transmitir datos en una red GSM, pasando de la conmutación de circuitos en GSM (donde el circuito está permanentemente reservado mientras dure la comunicación, aunque no se envíe información en un momento dado) a la conmutación de paquetes (La conmutación por paquetes es un método de comunicación en la cual el camino por el que viaja la información es determinado por un campo de dirección dentro de cada paquete y en la cual los caminos tomados por la información pueden ser diferentes para cada paquete).

La conmutación por paquetes permite fundamentalmente la distribución de los recursos de la radiofrecuencia. Un usuario GPRS sólo usará la red cuando envíe o reciba un paquete de información, todo el tiempo que esté inactivo podrá ser utilizado por otros usuarios para enviar y recibir información. Esto permite a los operadores proveer a los usuarios más de un canal de comunicación sin miedo a saturar la red, de manera que mientras en GSM sólo se ocupa un canal de subida de datos del terminal a la red y otro canal de bajada de datos desde la red al teléfono móvil, en GPRS es posible tener teléfonos móviles que gestionen cuatro canales simultáneos de bajada y dos de subida, pasando por ejemplo de velocidades de 9,6 kbps en GSM a 40 kbps en bajada y 20 kbps de subida en GPRS.

Otra ventaja de la conmutación de paquetes es que, al ocuparse los recursos sólo cuando se transmite o recibe información, la tarificación por parte del operador de telefonía móvil sólo se produce por la información transitada, no por el tiempo de conexión. Esto hace posible aplicaciones en la que un dispositivo móvil se conecta a la red y permanece conectado durante un período prolongado de tiempo sin que ello afecte en gran medida a la cantidad facturada por el operador.[6]

4. IMPLEMENTACION

4.1. Descripción del robot Giraa_02:

Giraa_02 es un robot móvil de locomoción diferencial, su estructura es circular de 30cm de diámetro y 20cm de altura, principalmente posee 8 sensores de ultrasonido SRF04 y 8 de infrarrojo GP2D120 para medir distancia, Su sistema de procesamiento de datos esta basado en el

microcontrolador Motorola MC9S12DP256 de 16 bits, además tiene puertos libres de expansión para la interconexión de más dispositivos sensores o actuadores [8]. El robot se observa en la figura 7.

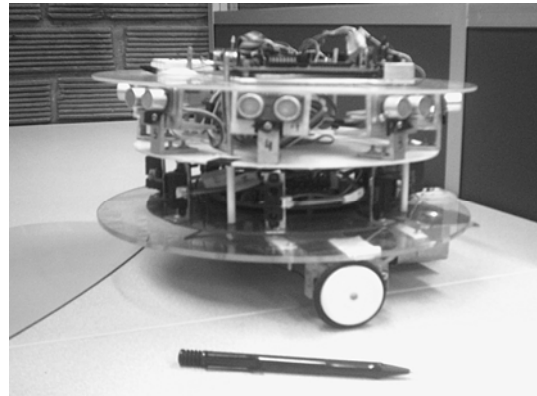


Figura 7. Robot móvil Giraa_02

Este vehículo posee un modulo de comunicación inalámbrica por RF (Radio Frecuencia), el cual permite comunicación con un computador a través de un puerto serial vía RS232, a una velocidad máxima de 5000 baudios.

La principal utilidad del Giraa_02 es servir como una plataforma de investigación, para experimentación y prueba de algoritmos de control autónomo y teleoperado de robots móviles. En este caso, mediante la implementación de ésta aplicación se realiza una exploración en la utilización de tecnologías móviles migrando la capacidad de teleoperación convencional del Giraa_02 a la red de telefonía celular por medio del servicio GPRS.

Básicamente el robot es operado por un conjunto de comandos, de los cuales se utilizan:

- f: Adelante
- a: Atrás
- r: Derecha
- l: Izquierda
- s: Parar
- m: 30 grados derecha
- n: 30 grados izquierda

Este conjunto de comandos se implementó en la aplicación para ser enviado a través de la red celular después de un proceso de autenticación en el servidor que se inicia desde la aplicación.

4.2. Estructura del Sistema

Como se puede ver en la figura 8, el dispositivo móvil se conecta a la red GPRS del proveedor de servicios, luego a través del gateway del proveedor realizar una conexión al servidor de la aplicación mediante Internet. Ya en el servidor, se procede a la autenticación y se realiza la acción requerida por la aplicación pasando la solicitud a

través del puerto serial, para ser transmitida al robot por medio del modulo de comunicación RF.

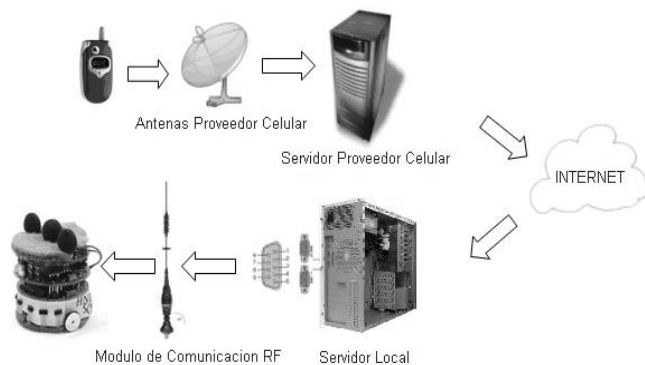


Figura 8. Estructura del sistema

4.3. Descripción de la Aplicación Móvil:

Se procede a mostrar el proceso de ejecución del programa, realizando especial énfasis en los aspectos que cubren la apropiación de la red celular para el envío de datos.

La aplicación debe ser cargada en el teléfono celular por medio del método de provisionamiento sobre el aire (OTA, Over The Air Provisioning. Que significa descargar la aplicación de un servidor), o a través del cable de datos del dispositivo y la aplicación autorizada por el proveedor del mismo para este fin. El proceso de OTA se muestra en la figura 9.



Figura 9. Provisionamiento OTA

Luego de ser descargada e instalada (esto se hace automáticamente), la aplicación puede ser ejecutada desde el menú de aplicaciones del celular o desde la carpeta a la cual se descargó.

La aplicación consta de cinco pantallas que se despliegan según sean las entradas del usuario al interactuar con el sistema, teleoperando el robot móvil desde el teléfono celular. Se muestran las posibles situaciones que se pueden presentar:

A. Pantalla de bienvenida, figura 10, en esta pantalla se muestra un mensaje de bienvenida acompañado de un gráfico. Básicamente es la presentación de la aplicación.

B. Pantalla de validación, figuras 11, 12 y 13, en esta pantalla se realiza la autenticación con el servidor para el acceso a la aplicación, se envían datos ingresados por el

usuario al servidor y este los compara con una base de datos que inicialmente es propuesta en texto plano. Para cada usuario existe un tiempo determinado de conexión, ya que el accionamiento se hace sobre un solo dispositivo físico (robot móvil).

C. Pantalla de modo de teleoperación, figura 14, se le ofrece al usuario la posibilidad de escoger el modo de teleoperación del robot para su desplazamiento. Los modos de teleoperación son: accionamiento y secuencia; el primero hace referencia al envío de comandos u órdenes una a una, mientras que el segundo envía una secuencia de órdenes con un temporizador determinado para el cumplimiento de dichas ordenes. Los comandos son descritos en la sección 4.1

D. Pantalla de Accionamiento, figuras 15 y 16, Envía órdenes una a una mostrando en pantalla la operación deseada.

E. Pantalla de Secuencia, figura 17, envía una secuencia de comandos para el movimiento del dispositivo.

4.4. Descripción del servidor

En el servidor se instaló el paquete Jakarta Tomcat como contenedor de Java Servlet [7]. Consta de tres procesos:

I. Autenticación: En este proceso, el servidor va a la base de datos y confirma que la identidad del usuario exista, que sea su respectiva contraseña y que sea su hora autorizada para la operación (esta característica se implementa por seguridad en el accionamiento del dispositivo), se le asigna al usuario un numero de sesión y se le permite la entrada al sistema

II. Selección de modo: El servlet toma la decisión con respecto a la petición del usuario sobre que operación debe realizarse (envío de acción simple o de secuencia temporizada). Luego de esto envía los caracteres de control por medio de transmisión serial y finaliza la operación.

III. Respuesta al usuario: El servlet responde con un código al usuario que significa que ya terminó la operación del comando y que lo habilita para ingresar más datos.



Figura10. Pantalla de bienvenida

Figura11. Negación de acceso, por horario no permitido



Figura12. Negación de acceso, por error en contraseña



Figura13. Pantalla de validación, datos correctos



Figura14. Selección del modo de teleoperación del robot



Figura15 Modo accionamiento sin una dirección seleccionada

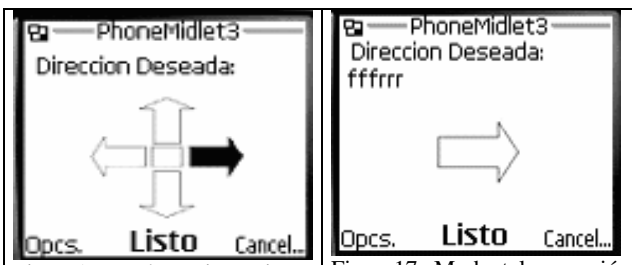


Figura16 Modo accionamiento con una dirección seleccionada



Figura17. Modo teleoperación por secuencia, con una ruta preestablecida

5. TRABAJO FUTURO

Se plantean varias alternativas de aplicación a los resultados de este proyecto, aprovechando las tecnologías móviles de información y haciendo énfasis en aplicaciones totalmente embebidas, que sean capaces de controlar sistemas remotos en locaciones donde sean prácticamente inexistentes los medios de comunicación cableados (fijos), tales como el teléfono y los servicios de Internet por cable. Se puede teleoperar maquinas por medio de ordenes dadas por el usuario y que además se pueda monitorear o reportar el comportamiento de dichos sistemas a través del tiempo. Estas aplicaciones se dan en el campo de la robótica, domótica, la seguridad, el control industrial y el sector agroindustrial, entre otros.

6. CONCLUSIONES

Se observaron diferencias entre los dos métodos de teleoperación (por órdenes o accionamientos y por secuencia), se evidencio que debido a la velocidad de transmisión de las redes de telefonía celular, es más conveniente realizar la transferencia a través de

secuencias pues esto reduce los efectos por retardos de la red.

Precisamente los retardos de la red, hace que las funciones teleoperadas deban ser combinadas con funciones autónomas, en la arquitectura de control del robot, de modo que se incorporan técnicas que reaccionan de forma autónoma a situaciones peligrosas, como evitar colisiones inminentes en tiempo real.

Se comprobó la versatilidad de la plataforma J2ME en la programación de aplicaciones para dispositivos de recursos limitados y el gran campo de acción que deparan las comunicaciones móviles aplicadas al monitoreo y teleoperación de diferentes tipos de maquinas y herramientas. El sistema desarrollado para teleoperar al robot móvil Giraa_02 puede ser adaptado a otros tipos de robots y demás dispositivos como electrodomésticos, maquinas de venta o dispensadores, equipo industrial.

7. RECONOCIMIENTOS

Este trabajo fue financiado por el Comité para el Desarrollo de la Investigación CODI, de la Universidad de Antioquia en el marco de los proyectos MC02-1-09 y FPPI06-1-07, desarrollados por el Grupo de Investigación en Robótica y Mecatrónica – adscrito al GIMEL, [<http://ingenieria.udea.edu.co/giraa/>].

8. BIBLIOGRAFÍA

- [1] OLLERO, Aníbal. Robótica manipuladores y robots móviles. Barcelona: Marcombo, 2001.
- [2] ARANGUREN Gerardo, *et al.* Control Remoto a Través de GSM, Mundo Electrónico, No. 328, Febrero de 2002, pp. 52-57.
- [3] SUN Microsystems J2ME Device List. Junio 20/05 <http://developers.sun.com/techttopics/mobility/device/device>.
- [4] SUN Microsystems. Julio 15/05 <http://java.sun.com/developer/>
- [5] SUN Microsystems Inc. Agosto 02/05 <http://java.sun.com/products/servlet/>
- [6] WIKIPEDIA – Enciclopedia Libre. Agosto 28/05 <http://es.wikipedia.org/wiki/GPRS>
- [7] APACHE tomcat documentation. Sep 15/05 <http://tomcat.apache.org>
- [8] MUÑOZ, Nelson David y LONDOÑO Nelson. Reporte técnico Proyecto CODI MC02-1-09, Universidad de Antioquia, Medellín, 2005.