Revista Tlamati Sabiduria

Volumen II, Número 2. Septiembre-Diciembre 2010

Título del artículo.

Laboratorio de experimentación remota de un robot Khepera.

Autor.

Eric Rodríguez Peralta

Referencia bibliográfica:

MLA

Rodríguez Peralta, Eric. "Laboratorio de experimentación remota de un robot Khepera." *Tlamati*. II.1 (2010): 43-52. Print.

APA

Rodríguez Peralta, E. (2010). Laboratorio de experimentación remota de un robot Khepera. *Tlamati*, *II*(1).

ISSN: 2007-2066.

Laboratorio de experimentación remota de un Robot Khepera

ERIC RODRÍGUEZ PERALTA

Resumen

En este documento se presenta la propuesta de un laboratorio virtual de experimentación remota de un robot Khepera. Los experimentos que se pueden realizar con esta herramienta consisten en probar diferentes estrategias para realizar el recorrido de un laberinto desde un punto inicial a uno final con la ayuda de los sensores infrarrojos del robot y mediante un algoritmo de control hecho en Java. El sistema brinda al usuario información del ambiente remoto a través de imágenes y datos con los que se puede observar los resultados de la tarea a realizar.

Palabras clave

Laboratorio virtual, robótica móvil, educación a distancia.

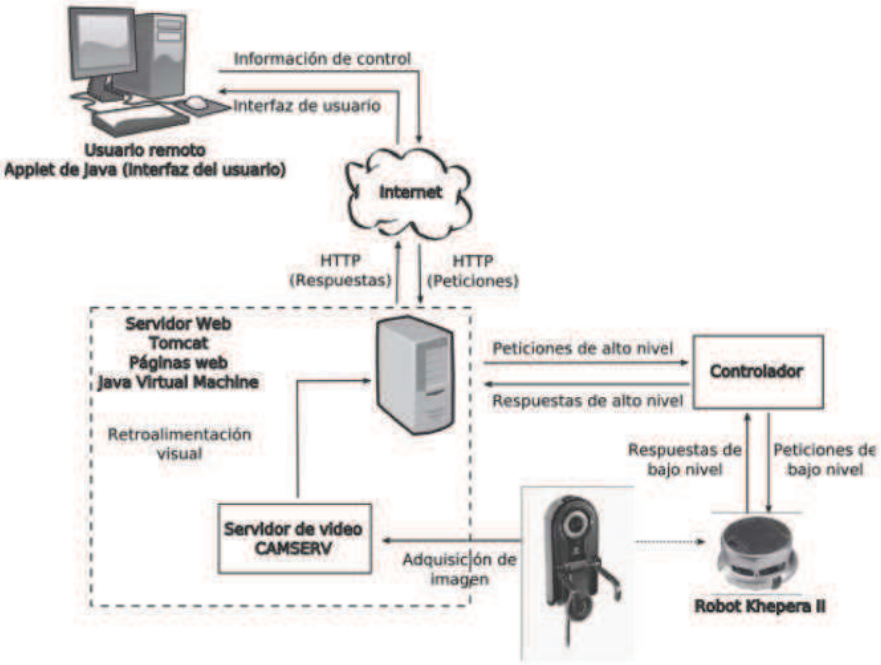

Figura 1: Arquitectura básica de un laboratorio virtual de experimentación remota.

1 INTRODUCCIÓN

En la actualidad, Internet es considerada como una infraestructura global de comunicación que permite una fácil implementación de sistemas distribuidos. En el ámbito de la educación y de la investigación, ha despertado un gran interés en los servicios de comunicación en línea facilitando la transmisión interactiva de audio y video. Esta disponibilidad de comunicación en conjunto con la generalización del uso de computadoras para la adquisición de datos y el control de procesos han conducido a la implementación de proyectos para la interacción con equipos en lugares remotos.

Una de las áreas de interés donde se han aplicado este tipo de proyectos es en robótica debido al uso del World Wide Web para acceder a laboratorios virtuales que permiten el control remoto de robots. La tele-operación de robots basada en Internet tiene una diversidad de aplicaciones en diferentes sectores de la sociedad. Uno de ellos es el de la educación a distancia donde Internet resulta de gran importancia, ya que no todas las instituciones de educación superior cuentan con recursos humanos ni material y equipos suficientes para afrontar de manera eficiente la formación de nuevos ingenieros e investigadores. Los laboratorios remotos pueden ser utilizados para realizar experimentos con robots reales y poder así compartir equipo costoso como es el caso de los robots móviles. De esta manera es posible cubrir los problemas de experimentación que son parte fundamental en el proceso de enseñanza-aprendizaje.

Una de las ventajas de los laboratorios remotos es que están accesibles en la red por largos períodos de tiempo por lo que los estudiantes pueden realizar sus prácticas desde diferentes lugares y en horarios flexibles. Otra de las ventajas de este tipo de sistemas, es que la comparación de resultados entre experimentos puede facilitarse debido a que las condiciones ambientales del robot son idénticas. Es por esto que la realización de un laboratorio virtual remoto resulta una aportación importante como herramienta educativa.

2 LABORATORIOS VIRTUALES

El concepto de laboratorios virtuales ha sido propuesto en la reunión de expertos en laboratorios virtuales [2]. Estos han sido definidos como espacios de trabajo electrónicos para la colaboración y experimentación a distancia en investigación y otras actividades creativas, para generar resultados usando tecnologías de información.

Un laboratorio virtual basado en Internet provee acceso a uno o múltiples usuarios para realizar experimentos ya sea de manera simulada o remota [6]. Estas

herramientas proveen a los usuarios retroalimentación visual, gráfica o basada en texto con lo que puede supervisar e interactuar con un experimento determinado. En general existen dos tipos de laboratorios virtuales: simulados y remotos.

Un laboratorio virtual simulado es una herramienta de software que recrea el comportamiento de un fenómeno o modelo físico de manera interactiva a través de la computadora, este tipo de herramientas dependen únicamente de los recursos propios de un servidor como el acceso a bases de datos. Un laboratorio virtual remoto permite operar remotamente cierto equipo y requieren de un servidor que dé acceso a dichos equipos.

Para construir un laboratorio virtual remoto, un modelo de teleoperación ha sido utilizado como se muestra en la figura 1, [5]. Este modelo se basa en un simple protocolo comúnmente usado en computación distribuida "Protocolo de petición/respuesta". Los clientes interactúan con el sistema utilizando un visualizador de Internet para hacer sus peticiones. Esta peticiones son traducidas a simples peticiones HTTP las cuales son respondidas por el servidor web. El servidor envía estas peticiones convirtiéndolas en peticiones de control de alto nivel que son recibidas por el controlador del robot que a su vez las envía al robot como peticiones de bajo nivel para realizar la tarea requerida. La retroalimentación es necesaria para enviar al usuario información sobre el ambiente remoto del robot y las consecuencias de sus comandos.

3 DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO

En este proyecto de investigación, el laboratorio remoto se construyó bajo una arquitectura genérica cliente/servidor para controlar el robot usando el protocolo estándar TCP/IP basado en sockets, como se muestra en la figura 2. Para el desarrollo de la aplicación se utilizó Java, como lenguaje de programación, obteniendo como resultado un sistema independiente de la plataforma y orientado a objetos. Esto significa que el usuario puede trabajar con cualquier sistema operativo y ser capaz de acceder a la página del laboratorio para interactuar con el robot.

La aplicación se compone principalmente de los siguientes módulos:

- Cliente
  - Autentificación
  - Interfaz de usuario
- Servidor
  - Retroalimentación
  - Control del robot

3.1 EL CLIENTE

El módulo del cliente se implementó por medio de un applet de Java[1], el cual es cargado en una terminal remota a través de un navegador web permitiéndole al usuario interactuar con el robot. Una vez realizada la

Figura 2: Arquitectura Cliente-Servidor con un protocolo de comunicación TCP/IP.

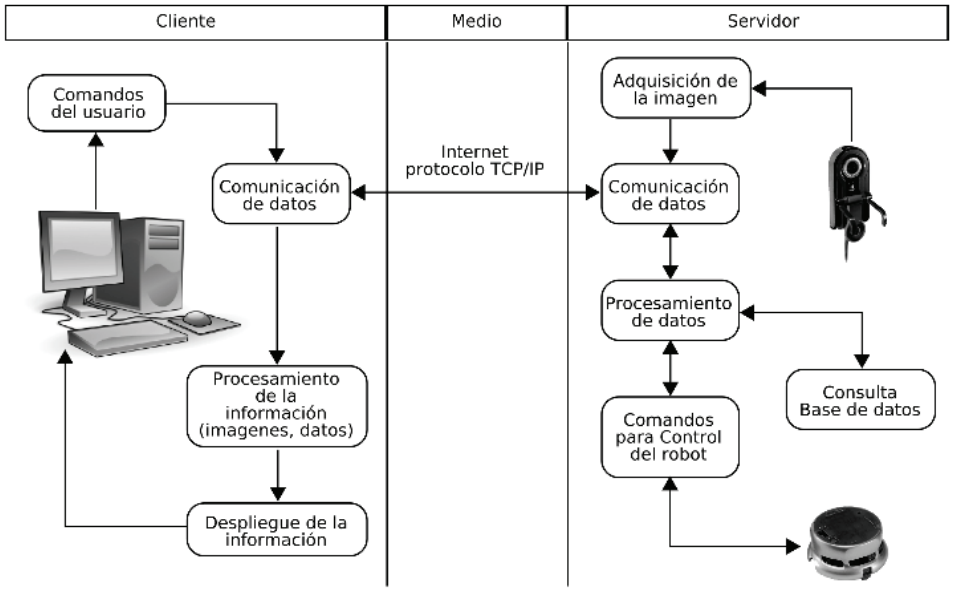

Figura 3: Página de autentificación de usuarios.

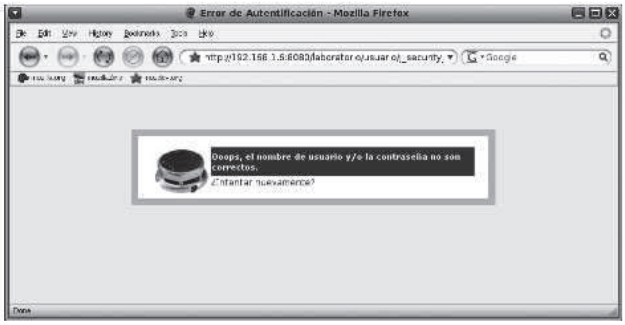

Figura 4: Página de error autentificación inválida.

conexión, el usuario cuenta con un período de tiempo determinado para probar su algoritmo de control y concluir el experimento, esto le da oportunidad a otros usuarios de hacer uso del laboratorio virtual.

3.1.1 MÓDULO DE AUTENTIFICACIÓN

Se decidió utilizar Apache Tomcat para alojar las páginas que dan acceso al laboratorio virtual por lo que esta parte del proyecto se basa más en la configuración que en la programación. Para poder realizar la tarea de autentificación de usuarios válidos y hacer uso del laboratorio virtual, fue necesario editar los descriptores XML tanto del servidor Tomcat, para decidir el método de autentificación que se va a utilizar, como de la aplicación web, para decidir qué recursos serían los protegidos.

Para acceder a la página del laboratorio virtual, se utilizó un mecanismo de autentificación HTTP basada en formularios, implementada por el servidor web, en donde se solicita el nombre de usuario y la contraseña como se muestra en la figura 3. Si los datos son válidos se pasará al recurso protegido, si no se mostrará una página de error como se muestra en la figura 4.

3.1.2 EL PANEL DE USUARIO

Es la parte del cliente del sistema desarrollado y consta de 18 clases. La clase principal es llamada panelUsuario que extiende de Applet. En esta clase se hace la instancia de los objetos de los que consta la aplicación. En el método init() del Applet, se añaden el panel de vídeo, el panel de lectura de sensores, el panel de control del robot , el panel de programación, el panel de resultados y el panel de estado como se muestra en la figura 5.

El usuario a través de un navegador puede acceder a los servicios de la interfaz mediante el protocolo HTTP

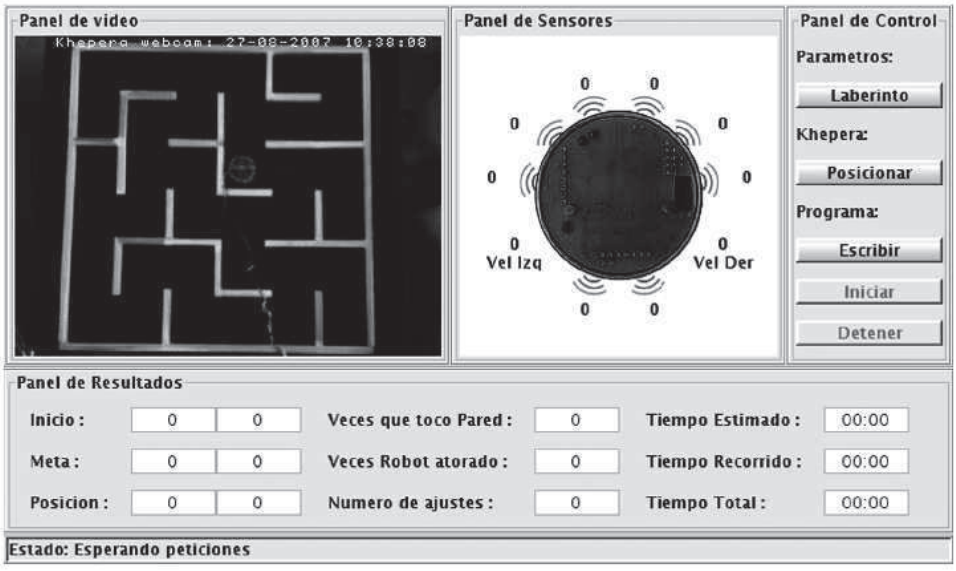

Figura 5: Interfaz de usuario del laboratorio de experimentación remoto.

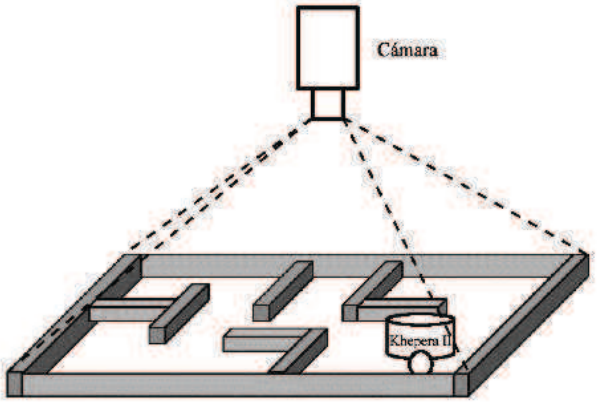

Figura 6: Vista global del entorno de trabajo.

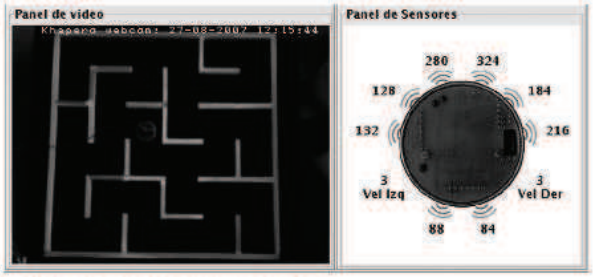

Figura 8: Lectura de los sensores del robot.

(Protocolo de Transferencia de Hipertexto) para establecer una conexión con el servidor. El acceso de los usuarios al laboratorio remoto lo hacen por medio de un Applet que provee información gráfica, textual y visual del ambiente remoto en los diferentes paneles que componen la interfaz.

Una parte importante de este proyecto fue proporcionar al usuario retroalimentación visual del ambiente remoto. Para realizar esta tarea, se instaló una webcam sobre el área de trabajo del robot con la que se obtiene una visión global del entorno en el que se moverá el robot como se muestra en la figura 6.

Para la realización de los experimentos se utilizó una cámara Quickcam pro for Notebooks de Logitech ya que está bien soportada en el sistema operativo Linux que se encuentra instalado en el servidor donde se aloja la aplicación. Los controladores necesarios se pueden obtener de [7]. Las imágenes son adquiridas por la webcam usando video4linux y son recibidas y procesadas por un servidor de video. El servidor de video utilizado es Camserv [8] que está implementado en lenguaje de programación C y es completamente libre y gratuito. Se desarrolló un applet de Java, como se muestra en la figura 7, para mostrar las imágenes en el panel de video de la interfaz del usuario.

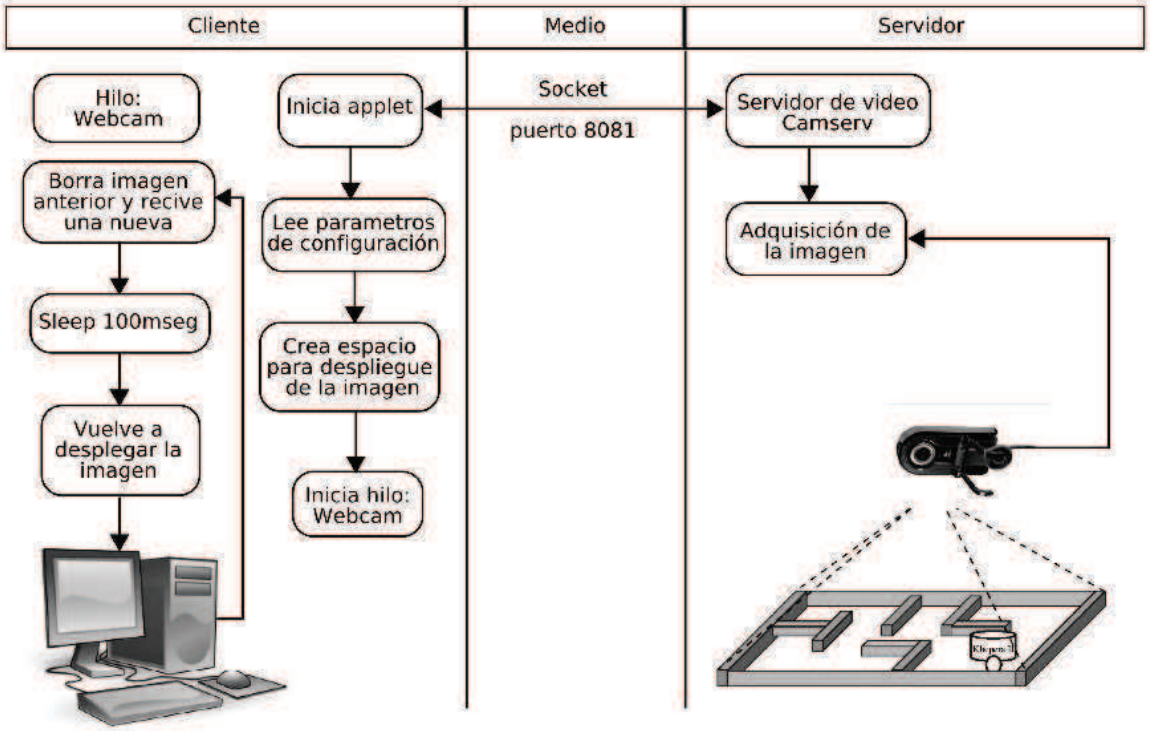

Figura 7: Diagrama de flujo del Applet de video, la imagen es actualizada cada 10ms.

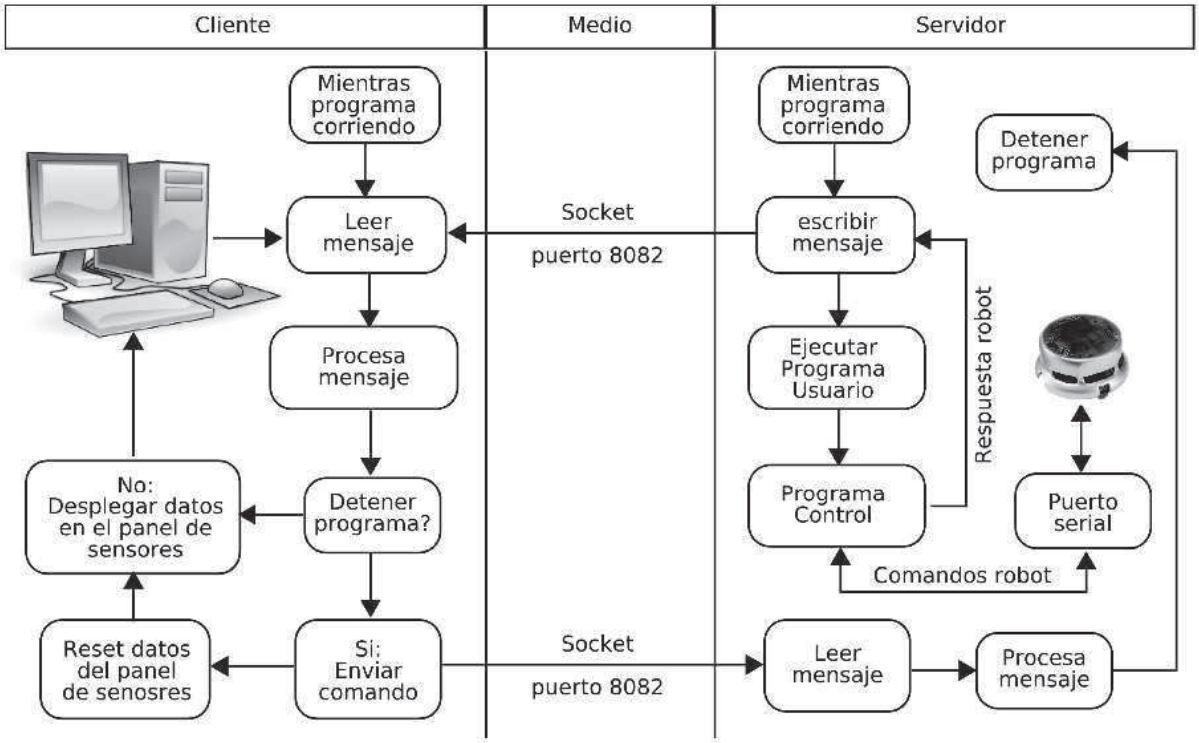

Figura 9: Diagrama de flujo para la lectura de los sensores y actuadores del robot.

El ambiente del robot Khepera, es un laberinto configurable utilizado para probar la inteligencia del robot y funcionalidad del sistema. Las paredes del laberinto son bloques de madera de 2.5 cm de alto por 11.5 cm de largo con lo que se pueden lograr corredores de 10 x 10 cm, espacio suficiente para que el khepera pueda moverse con libertad. Excepto por las paredes externas, todos los bloques internos que forman las paredes son movibles, de tal forma que se puede cambiar fácilmente el diseño del laberinto.

El robot Khepera cuenta con un anillo de 8 sensores infrarrojos. El panel de sensores muestra la lectura de éstos en formato numérico mientras se ejecuta el programa de control, su valor se actualiza continuamente mientras el robot se mueve dentro del laberinto y es desplegado cerca de su respectiva posición como se muestra en la figura 8. Estos componentes son capaces de detectar un obstáculo dentro de un rango aproximado de 50mm.

Los valores varían en un rango de 0 a 1024 unidades [4], donde el primer valor indica que ningún objeto ha sido detectado y el segundo indica el máximo valor de proximidad a un objeto, esto significa que el valor de los sensores es inversamente proporcional a la distancia con el objeto.

El panel de sensores despliega en la interfaz del cliente lo que el robot percibe de su entorno mientras se ejecuta el programa de control. En este panel se muestran también la velocidad de las ruedas del robot que varía en el rango de 1 a 9. La velocidad está dada en pulsos/10ms que corresponde a 0.8 cm/seg en la mínima velocidad y 7.2 cm/seg en la máxima velocidad. La figura 9 muestra el diagrama de flujo de las lecturas de los sensores de proximidad y de la velocidad del robot.

El panel de control incluye los botones necesarios para interactuar con la aplicación. Algunos de ellos abren ventanas que le permiten al usuario manipular de manera directa el robot, escribir y enviar un programa para controlar el robot y observar un laberinto virtual, los demás botones permiten al usuario iniciar o detener el programa en cualquier instante de tiempo.

El botón laberinto del panel de control abre una ventana que muestra gráficamente el diseño de un laberinto virtual, similar al que se encuentra en el espacio de trabajo remoto del robot, figura 10. En esta ventana se especifican las coordenadas tanto del punto de inicio como de la meta, se calcula la trayectoria óptima que debe seguir el robot entre estos dos puntos y el tiempo estimado para ir de un punto a otro. Los datos del laberinto virtual, diseño y valores, son cargados, al momento de abrir esta ventana, desde una base de datos que se encuentra del lado del servidor.

El botón de posicionamiento abre una ventana que muestra los botones que permiten al usuario controlar

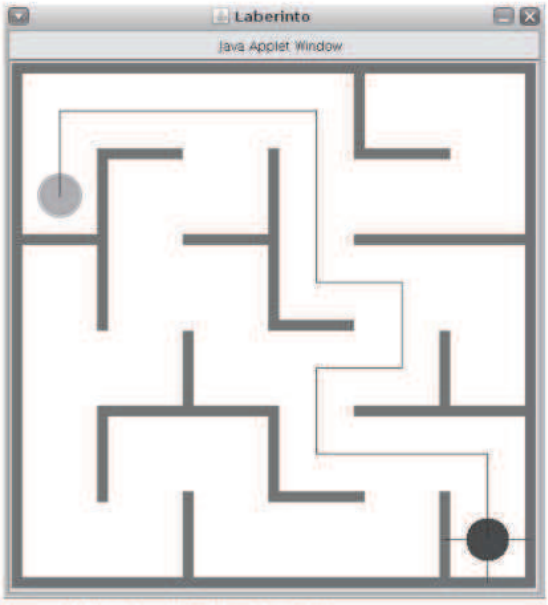

Figura 10: Ventana del laberinto virtual.

el robot de manera remota, con la idea de dirigirlo a la posición de inicio que se muestra en la ventana del laberinto virtual. Esto permite al sistema medir los parámetros de tiempo y control en comparación con los establecidos en esa ventana. Cada uno de los botones de la ventana de posicionamiento, mostrados en la figura 11, activa un programa del lado del servidor con el fin de enviar los comandos necesarios al robot, para su control, a través del puerto serial. El programa finaliza automáticamente al concluir la tarea requerida.

La ventana de programación está compuesta por un editor de texto en donde el usuario puede escribir el conjunto de instrucciones de programación automática es decir, el algoritmo de control que va a ser enviado y ejecutado por el robot. Se utiliza un área de texto para desplegar los posibles errores que se generen en el proceso de compilación del programa y un botón que tiene la función de enviar el código escrito hacia el servidor. La Figura 12 muestra la ventana de programación que incluye el código necesario para que el usuario comience a escribir su programa de control.

Una vez que el usuario ha terminado de escribir su programa en el editor de texto, éste se envía al servidor por medio del botón enviar programa en donde se guarda en un directorio predeterminado con el nombre de Usuario.java y se inicia el proceso de compilación. Si el programa está libre de errores, la ventana de programación se cierra y se habilita el botón iniciar programa en la interfaz del usuario de lo contrario, el servidor captura y envía al cliente a través del socket el posible error de sintaxis que es desplegado en el área de texto que se encuentra en la parte inferior de la ventana de programación indicando el tipo de error y el número de línea en el que se está generando ese error. La figura 13 muestra el diagrama de flujo del proceso de envío y compilación del programa escrito por el usuario.

En este panel, se muestran los parámetros iniciales definidos en el panel de dibujo y los resultados obtenidos al ejecutarse el programa de control escrito por el usuario. Los primeros se refieren a las coordenadas de la posición inicial a partir del cual se van a medir los parámetros de tiempo y control, las coordenadas de la meta y el tiempo estimado para ir de un punto a otro. Los demás resultados son calculados mientras el robot ejecuta los comandos enviados por el programa del usuario mientras se mueve en su ambiente de trabajo.

Los valores *Inicio* y *Meta* están relacionados con las coordenadas del punto de partida del robot dentro del laberinto y las coordenadas del punto que debe encontrar mientras se está ejecutando el algoritmo de control. El robot cumple con el objetivo del experimento cuando los sensores infrarrojos detectan una fuente de iluminación ubicada en la misma posición que el punto meta.

Los dos primeros valores de la segunda columna, veces que el robot tocó pared y número de veces que el robot se atoró se determinan mientras el algoritmo de control mueve el robot dentro de su entorno, el primero de ellos estima el número de veces que el robot choca contra cualquiera de las paredes del laberinto basándose en la información obtenida por los sensores de proximidad. Se considera que el robot tocó una de las paredes cuando cualquiera de sus sensores alcanza el máximo valor de proximidad a un objeto.

Para estimar el segundo valor de la columna, la cantidad de veces que el robot se atoró, se tomó en cuenta el tiempo que el robot se detiene por más de 5 segundos,

Figura 11: Botones de posicionamiento del robot.

mientras el programa de control continua ejecutándose, si esto sucede, una rutina del sistema mueve el robot para continuar con el algoritmo de control. El tercer valor, Número de ajustes, se refiere al número de correcciones que el usuario ha hecho a su programa, éste valor se incrementa cada vez que el programa de control del usuario es detenido y reenviado al servidor.

En la tercera columna se muestra el valor del *Tiempo estimado* que el robot debe tardar en realizar la tarea de ir de la posición de inicio a la meta a una velocidad promedio, el segundo valor, *Tiempo de recorrido*, mide el tiempo que el robot está haciendo para encontrar la meta mientras se ejecuta el programa de control. El tercer valor se refiere al *Tiempo total* acumulado que requirió el usuario en realizar el experimento. La medición de estos parámetros se relacionan con los conceptos de sensores, control y programación de robots mediante el uso de técnicas de inteligencia artificial que un usuario del laboratorio virtual debe conocer para que su aprendizaje sea significativo.

4 EL SERVIDOR

El servidor se implementa como una aplicación que administra la conexión con el cliente mediante el uso de TCP (Transport Control Protocol) basada en Sockets y está formada por 16 clases, entre las cuales se encuentra la clase Servidor que está corriendo en espera de una conexión. Las peticiones hechas por los usuarios son recibidas, procesadas y ejecutadas por esta aplicación, enviando la respuesta correspondiente al cliente para desplegarla en la interfaz de usuario.

Del lado del servidor se compila y se ejecuta el programa Controlador enviado por el usuario. El código le indica al robot, a través del puerto serial de la computadora, como moverse dentro de su entorno basándose en la información que recibe de sus sensores. Para la realización de esta etapa, se desarrolló un módulo de control que es utilizado para ejecutar el programa realizado por el usuario y la comunicación con el robot.

Para la realización del módulo de control, se utilizaron dos hilos, el primero de ellos es un hilo de control encargado de actualizar constantemente la información de los sensores del robot y enviar los comandos de operación a los actuadores por medio del puerto serial. El segundo hilo es responsable de enviar las lecturas de los sensores y actuadores al cliente a través del socket. La clase Proceso es la responsable de iniciar y finalizar los hilos de procesamiento involucrados y de terminar el ciclo de vida de los objetos que componen el sistema.

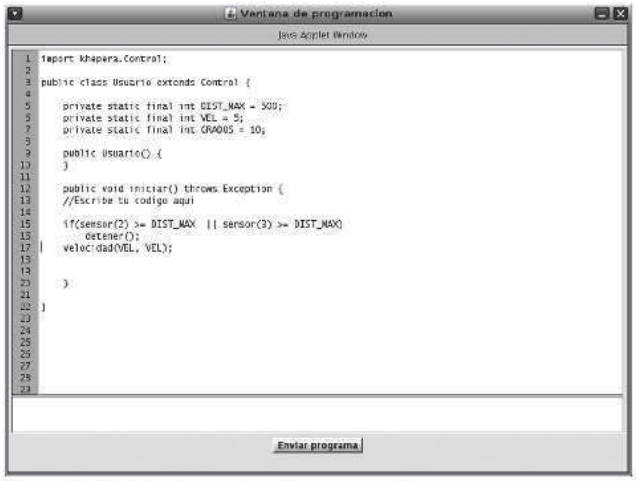

Figura 12: Interfaz de la ventana de programación.

El robot Khepera, utilizado en este proyecto, puede trabajar en dos formas diferentes, de manera autónoma (el programa de control se baja al procesador del robot) o por medio de una conexión al puerto serial de una computadora que es donde se ejecuta el programa de control. Para la implementación de este proyecto se utilizó la segunda opción.

En este modo el robot Khepera utiliza un protocolo de comunicación, basado en comandos y respuestas en código ASCII. Los comandos enviados al robot inician con una letra mayúscula, seguida por el parámetro requerido y finaliza con un retorno de carro representado por el caracter ¶. La respuesta del robot inicia con la letra minúscula correspondiente al comando enviado, seguida de los datos a transmitir y un retorno de carro. Por ejemplo, el comando para leer los sensores de proximidad del robot Khepera es el siguiente: N¶, La respuesta del robot es regresada como una cadena de caracteres que debe ser procesada para determinar los valores numéricos de cada uno de los sensores de proximidad: n,0,59, 1023, 1023, 78, 0, 0 ,0¶.

Dado que este tipo de comunicación entre el robot y la computadora implica abrir una conexión con el puerto serial de la computadora, enviar el comando, recibir una cadena de caracteres como respuesta y procesar esta respuesta implica varias líneas de código, se desarrolló una clase llamada Control.class que facilita la tarea de programación a los usuarios.

En esta clase se encapsulan los métodos para controlar el robot de una manera sencilla. Estos métodos están relacionados con la obtención del valor de proximidad de los sensores infrarrojos, la especificación de la velocidad de las ruedas del robot así como el control de giros.

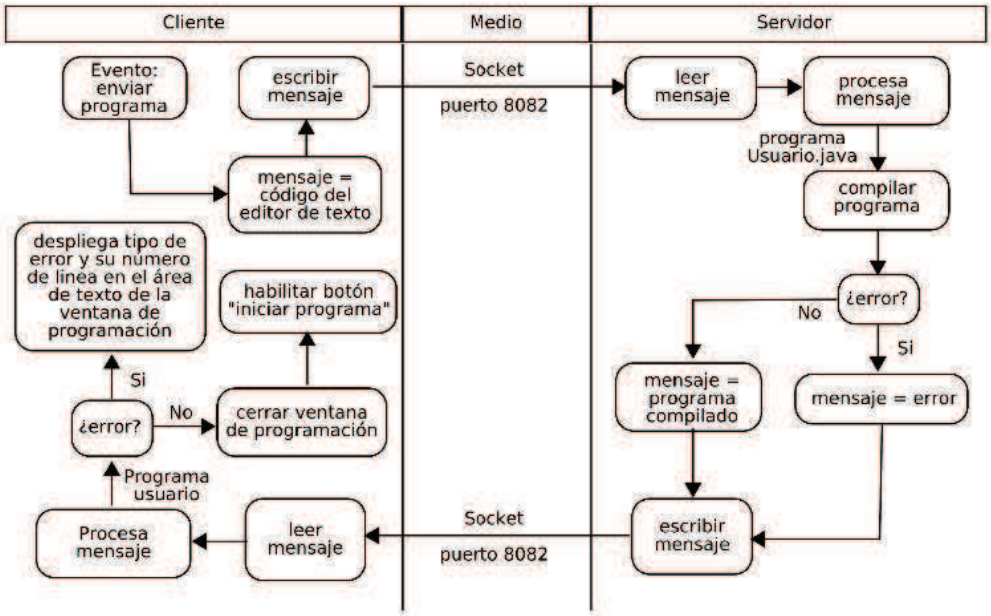

Figura 13: Diagrama de flujo del proceso de compilación del programa de control.

El constructor de la clase Control crea una instancia de la clase ComSerial que provee una interfase con el puerto serial de la computadora a través de la cual el robot se comunica con el algoritmo de control. Para el desarrollo de este proyecto se utilizó el puerto de comunicación serial estándar RS232 a una velocidad de 19200 baudios, 8 bits de datos, 1 bit de inicio, 2 bits de alto, sin paridad, así como se especifica en el manual de usuario del robot Khepera [4]. Para esto, se utilizó el API (Application Programming Interface) de comunicación RXTX [3] que es una librería que provee comunicación con el puerto serial y paralelo para el Kit de Desarrollo de Java (JDK por sus siglas en Inglés).

5 CONCLUSIONES

Este laboratorio de experimentación remota virtual de robótica móvil se diseñó como una herramienta educativa complementaria a la materia de robótica móvil para la realización de experimentos con un robot real, relacionados con aspectos de sensores, control y programación. Este laboratorio facilita a los usuarios interactuar con un robot que se encuentra localizado en un ambiente remoto enfrentándolos a diversas complejidades que involucra un ambiente no simulado.

El sistema brinda al usuario información del ambiente remoto a través de imágenes y datos con los que se pueda observar los resultados de la tarea a realizar. Los experimentos que se pueden realizar con esta herramienta consisten en probar diferentes estrategias de control para salir de un laberinto con la ayuda de los sensores infrarrojos del robot y mediante un algoritmo de control hecho en Java.

La ventaja de utilizar Java como lenguaje de programación es que el laboratorio virtual puede ser ejecutado en cualquier plataforma para la que exista una implementación de la máquina virtual Java. En la actualidad existen multitud de implementaciones de dicha máquina virtual, lo que nos permite que la aplicación pueda ser considerada como multiplataforma, permitiéndose la ejecución, por ejemplo, tanto en plataformas Windows como en entornos Linux. Sin embargo, cabe aclarar que los usuarios del laboratorio deben tener conocimientos sólidos de este lenguaje para la programación de los algoritmos de control lo que implica una limitante.

El hecho de que el programa de control enviado por el usuario se ejecute en el servidor, permite que el tamaño del algoritmo de control no esté restringido por el tamaño de la memoria y del procesador instalado en el robot. Además, el que se utilice un lenguage de alto nivel permite escribir y depurar con mayor facilidad el programa de control.

Las principales aportaciones de este trabajo son el desarrollo de un ambiente de operación remota para un robot Khepera que debe resolver el problema de un laberinto, la medición de los parámetros de tiempo y control de la tarea realizada que proporcionan una idea del desempeño del algoritmo, la obtención de la trayectoria

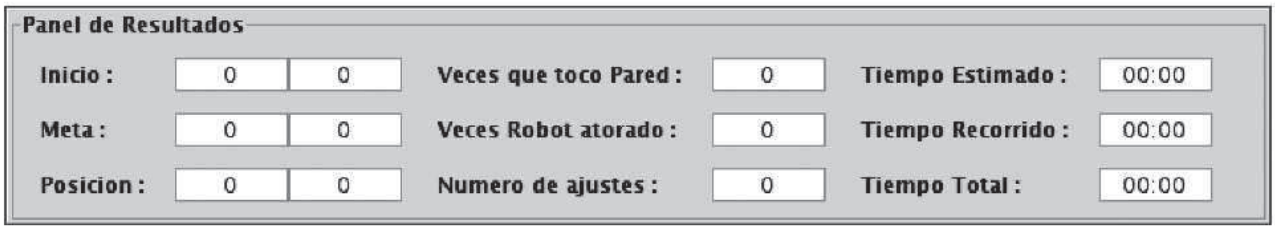

Figura 14: Panel de resultados del experimento.

óptima y el tiempo de recorrido entre el punto de inicio y la meta, la flexibilidad para trabajar con diferentes escenarios y la facilidad para incrementar de manera muy simple el número y tipo de experimentos a realizar.

6 TRABAJO FUTURO

Existen diferentes líneas de trabajo de gran interés para la continuación de la presente investigación, entre las que cabe mencionar las siguientes:

- Realizar pruebas con los usuarios finales del sistema utilizando diferentes algoritmos de control para comprobar la aceptación y funcionalidad de la herramienta.
- Integrar el laboratorio remoto a un curso de robótica para realizar experimentación con un robot móvil que ayude a lograr un entendimiento más profundo de la materia.
- Integración del laboratorio virtual remoto a un tutor inteligente para proporcionar a los estudiantes retroalimentación inmediata sobre el desempeño del experimento mediante la evaluación y diagnóstico de este tutor, contribuyendo así al mejoramiento del proceso de aprendizaje del estudiante.
- Proporcionar retroalimetación visual en forma gráfica de manera tal que se puedan mover al mismo tiempo el robot real en el ambiente remoto y el robot simulado en el laberinto virtual.
- Integrar otros parámetros de evaluación que ayuden a mejorar la evaluación del desempeño de experimentos como el número de veces que el robot ha pasado por un mismo lugar y la estimación de la trayectoria seguida por el robot con la idea de compararla con la trayectoria óptima.
- Incluir en la página del laboratorio remoto un enlace de ayudas que le permitan al estudiante mejorar la eficiencia y desempeño del experimento.
- Integración de un simulador del laboratorio virtual remoto en el que los estudiantes puedan probar sus algoritmos de control antes de que sean enviados al robot real con la idea de hacer un uso eficiente de la herramienta.
- Este laboratorio remoto puede servir como base para el desarrollo de diferentes experimentos de robótica móvil equipando al robot con accesorios y diferente tipo de sensores que le ayuden a obtener un mejor conocimiento de su entorno desarrollando así tareas más complejas.

1 Los Applets son aplicaciones Java que se ejecutan dentro de un navegador Web (generalmente, como parte de una página Web) [1]

REFERENCIAS

[1] Bruce Eckel. *Piensa en Java, Segunda edición.* Prentice Hall, 2002.
[2] IITAP. *Report of the expert meeting on virtual laboratories. Technical report, International Institute of Theorical and Applied Physics,* Ames, Iowa, 10 Dec 1999.
[3] Trent Jarvi. *Api de comunicación para el puerto serial hecho en java.* http://users.frii.com/jarvi/rxtx/license.html, 1997. Fecha de consulta Mayo. 2006.
[4] K-Team. *Khepera User Manual, Version 1.1.* K Team, S.A., Ch de Vassuet, CP 111 1028 Préverenges Switzerland, March 2002.
[5] Alaa M Khamis. *Interacción Remota con Robots Móviles Basada en Internet.* PhD thesis. Universidad Carlos III de Madrid, 2003.
[6] J Moreno. *Un nuevo enfoque metodológico para la enseñanza a distancia de asignaturas experimentales: Análisis, diseño y desarrollo de un laboratorio virtual remoto para el estudio de la automática a través de internet.* PhD thesis, Universidad Nacional de Educación a Distancia (UNED), 2001.
[7] Luc Saillard. *Free phillips usb webcam drivers for linux.* http://www.saillard.org/linux/pwc. Fecha de consulta, Mayo del 2006.
[8] J. Travis. *Camserv software.* http://eserv.sourceforge.net. Fecha de consulta, Mayo 2006.