CONTROL DE UN ROBOT CON ACCESO DESDE UN SERVIDOR WEB UTILIZANDO MATLAB

MARÍA FERNANDA DUEÑAS CORNEJO

OSCAR EDUARDO HURTADO GONZÁLEZ

UNIVERSIDAD PONTIFICIA BOLIVARIANA

FACULTAD DE INGENIERÍA ELECTRÓNICA

DECANATURA DE INGENIERÍA Y ADMINISTRACIÓN

BUCARAMANGA

2008

CONTROL DE UN ROBOT CON ACCESO DESDE UN SERVIDOR WEB UTILIZANDO MATLAB

MARÍA FERNANDA DUEÑAS CORNEJO

OSCAR EDUARDO HURTADO GONZÁLEZ

TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARA OPTAR AL TITULO DE INGENIERO ELECTRONICO

Director

PhD. OMAR PINZON ARDILA

UNIVERSIDAD PONTIFICIA BOLIVARIANA

FACULTAD DE INGENIERÍA ELECTRÓNICA

DECANATURA DE INGENIERÍA Y ADMINISTRACIÓN

BUCARAMANGA

2008

A Dios por la vida y por haberme dado la familia maravillosa que tengo.
A mis padres Fernando y Luz Marina por su inmenso amor y la confianza que depositan en mi.
A mi hermana Daniela por su amor.

María Fernanda

A Dios por brindarme todas mis capacidades.
A mis padres y mi hermana por el apoyo y la confianza brindados.

Oscar Eduardo

AGRADECIMIENTOS

A nuestros padres por su apoyo y financiamiento para el desarrollo del proyecto.

Al Doctor OMAR PINZÓN ARDILA. Por creer en nosotros, por sus enseñanzas y por su apoyo incondicional.

A la facultad de Ingeniería Electrónica de la Universidad Pontificia Bolivariana por su gran aporte a nuestra formación profesional.

Y a todos aquellos que pusieron un granito de arena para que este proyecto fuera posible.

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

RESUMEN

Durante el desarrollo del proyecto se explora la forma de controlar un robot como una aplicación para el desarrollo de un entorno virtual.

Para esto, el robot se conecta a un servidor para que pueda ser controlado a través de una página Web ejecutada desde un computador conectado a Internet.

Esta es una herramienta didáctica que permite al estudiante controlar un sistema real desde cualquier punto de conexión a Internet. En un equipo servidor se desarrolla un hardware de control que se ejecuta por medio de la implementación de Matlab y la comunicación de este con el usuario remoto a través de un servidor Web utilizando el leguaje PHP.

En el computador servidor se instala Apache como servidor de páginas Web y un intérprete de PHP, ya que este es el lenguaje utilizado para el acceso al hardware de forma remota. De esta forma se elimina la interacción directa que siempre ha existido entre el robot y su operario, llevando esta aplicación a un medio muy utilizado en todo el mundo como es la Internet.

El robot utilizado en el proyecto ha sido desarrollado previamente por los mismos autores de éste; evitando así acudir a robots elaborados por terceras personas y, por consiguiente corriéndose el riesgo de encontrar problemas de estrategias de control dado que es casi imposible llegar a modificar el algoritmo de programación con que ya vienen programados.

ABSTRACT

During the development of the project will explore how to control a robot as an application for development of a virtual environment. For this, the robot is connected to a server so it can be controlled through a Web site running from a computer connected to the Internet.

This is a teaching tool that allows the student to control a real system from anywhere with Internet connection. On a computer server develops a hardware control that runs through the implementation of Matlab and communicating this with the remote user via a Web server using the PHP language.

The computer server is installed as Apache web server and an interpreter PHP, as this is the language used to access the hardware remotely. This eliminates the direct interaction that has always existed between the robot and its operator, leading to half this application widely used throughout the world as the Internet.

The robot used in the project has been developed in advance by the same authors thereof; thereby avoiding recourse to robots produced by others and therefore runs the risk of finding problems of control strategies since it is almost impossible to arrive at changing the algorithm programming that are already scheduled.

INTRODUCCIÓN

La utilización de herramientas didácticas para la enseñanza en instituciones de educación superior es cada día más común en nuestro medio. Por medio de estas herramientas el estudiante se motiva en el desarrollo de nuevas habilidades y destrezas en su metodología de estudio.

En la Universidad Pontificia Bolivariana existen algunas de estas herramientas para la enseñanza de Sistemas de Control, pero las opciones didácticas son limitadas. En este proyecto se desarrolla una aplicación para un futuro Laboratorio de Control Virtual con el fin de brindarle al estudiante un elemento interactivo en su educación.

Desde el punto de vista de la enseñanza, un laboratorio virtual al igual que cualquier otra herramienta didáctica es una nueva estrategia de enseñanza dirigida a ampliar de una manera sencilla el conocimiento del estudiante, haciendo de él un estudiante más competitivo en su ámbito profesional.

En este trabajo se muestra una posible aplicación para un Laboratorio Virtual. Además, se espera que este trabajo sea un valioso aporte y punto de partida para el desarrollo de herramientas didácticas que sean comprensibles y amenas para generar diferentes prácticas de laboratorio.

1. ESTRUCTURA DEL PROYECTO

1.1 PLANTEAMIENTO

En el medio universitario se requiere una mayor cantidad de elementos para que cada estudiante pueda tener su propia experiencia en el laboratorio. Es por esto que se plantea que los estudiantes puedan acceder a los laboratorios mediante acceso a Internet para monitorear y controlar un sistema permitiendo que mayor cantidad de estudiantes accedan a dicho laboratorio desde una estación remota.

1.2 JUSTIFICACIÓN

En este proyecto se busca cubrir en el medio universitario la necesidad mencionada en la sección anterior, ofreciendo la posibilidad de acceder a este tipo de herramientas de una manera más practica, cómoda, novedosa y por tanto impactante de hacer practicas de control, teniendo en cuenta que se puede acceder a la experiencia desde un aula virtual y de esta forma se evita el desplazamiento al lugar físico del experimento; además, se hace más fácil la comprensión de las diferentes técnicas de control con la implementación de nuevas herramientas que están al alcance del usuario de este laboratorio.

1.3 OBJETIVO GENERAL

Diseñar, implementar y desarrollar un control de un robot a través de un servidor Web.

1.4 OBJETIVOS ESPECÍFICOS

- Diseñar una interfaz gráfica para acceder de forma remota al control del robot.
- Establecer las técnicas de control adecuadas para las diferentes acciones del robot.
- Realizar un análisis de los diferentes movimientos del robot para establecer el control a utilizar.
- Conocer el lenguaje básico de PHP.

2. MARCO TEÓRICO

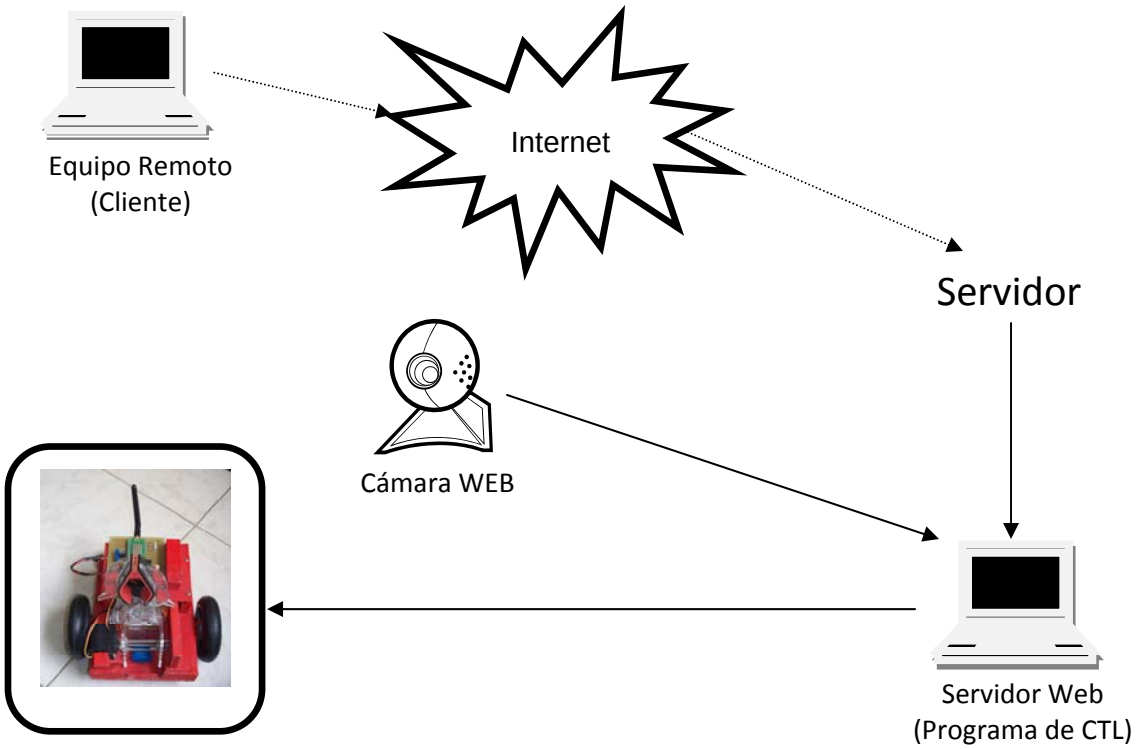

Figura 1. Esquema general del proyecto

En la figura 1 se muestra un diagrama de bloques funcional señalando la interacción entre cada uno de los elementos a utilizar en el proyecto: los 2 equipos (uno utilizado como cliente y el otro como servidor), la cámara con la cual se hace la adquisición de imágenes, el servidor a utilizar y el robot a controlar. El manejo que se le da a cada uno de los elementos anteriores con el fin de llegar al objetivo final (control de posición del robot), consiste en: el equipo cliente se comunica con el servidor a través de Internet, este servidor contiene un administrador Web (Apache), un software de procesamiento de imágenes y posicionador del robot. Para adquirir la imagen de la posición del

robot y el punto al cual se quiere que este llegue se utiliza una *Web Cam*; la comunicación entre el servidor y el robot se realiza por medio de un módulo *Bluetooth* conectado al puerto USB del servidor y enlazado con una tarjeta de protocolo *Bluetooth* del robot.

A continuación, se hace una breve introducción de conceptos básicos de los diferentes temas a tratar en el desarrollo de este proyecto, con el fin de darle al lector una idea general de lo planteado.

2.1 MATLAB WEB SERVER

El servidor web de MATLAB permite ejecutar aplicaciones de MATLAB en un servidor, enviando los datos a MATLAB para que este realice los cálculos necesarios y envíe los resultados a través de Internet para visualizarlos en un navegador web.

El servidor web de MATLAB depende de las redes TCP/IP para la transmisión de datos entre los sistemas cliente y MATLAB. La conexión entre el software y el hardware debe instalarse en el Sistema Operativo antes de utilizar el servidor web de MATLAB.

En la figura 2 se muestra una configuración simple donde un navegador web se ejecuta desde una estación de trabajo (Cliente) mientras que el MATLAB Web Server se ejecuta desde otra máquina (Servidor).[1]

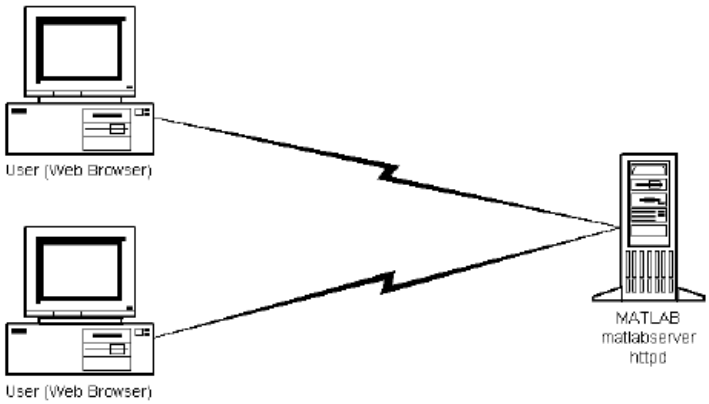

Figura 2. Configuración cliente- servidor, siendo el servidor el MatlabServer[1]

2.2 LENGUAJE PHP

PHP es un lenguaje interpretado que permite la generación dinámica de contenidos en un servidor Web. Su nombre es Hyper Text Preprocessor. Entre sus principales características se pueden destacar su potencia, alto rendimiento y su facilidad de aprendizaje. PHP es una herramienta eficaz de desarrollo para los programadores Web, ya que proporciona elementos que permiten generar de manera rápida y sencilla sitios Web dinámicos. Originalmente es creado por Rasmus Lerdorf como un conjunto de utilidades para añadir dinamismo a las páginas Web. Este conjunto de herramientas gana rápidamente popularidad y es rediseñado por Zeev Suraski y Andi Gutmans y rebautizado como PHP 3.0; más

[1] URL: http://som.yale.edu/unix/research_computing/pdf_doc/webserver/webserver.pdf

tarde se vuelve a rediseñar completamente el intérprete, añadiéndole más capacidad y nuevas funcionalidades, para dar lugar al lenguaje que hoy se conoce como PHP 4. PHP es un lenguaje de programación que contiene muchos componentes de C, Perl y Java. Su sintaxis es muy similar a la de estos lenguajes, haciendo muy sencillo su aprendizaje.

2.3 APACHE

Es un software servidor HTTP de código abierto para plataformas Unix, Windows y otras, que implementan el protocolo específico. Cuando comenzó su desarrollo en 1995 se basó inicialmente en códigos del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, “a patchy server” o un servidor parcheado.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Entre las características más importantes de este servidor, se tienen:

- Tiene una amplia aceptación en la red.
- Es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado.[2]

[2] URL: http://www.apache.org/

2.4 PROTOCOLOS DE COMUNICACIÓN

En este proyecto se trabajan dos protocolos de comunicación. El primero de ellos es el protocolo TCP/IP para la transmisión de datos entre los equipos servidor y cliente y el segundo es el protocolo Bluetooth para la comunicación entre el equipo servidor y el robot.

2.4.1 Protocolo TCP/IP: este protocolo es utilizado en la interconexión y envío de datos entre dos computadores. El modelo TCP/IP estructura el problema de la comunicación en cinco capas relativamente independientes entre si, las cuales son:

- Capa Física.
- Capa de Acceso a la red.
- Capa de Internet.
- Capa extremo a extremo o de transporte.
- Capa de aplicación.

La capa física define la interfaz física entre el dispositivo de transmisión de datos y el medio de transmisión o red. Esta capa se encarga de la especificación de las características del medio de transmisión, la naturaleza de las señales, la velocidad de datos y cuestiones a fines.

La capa de acceso a la red es responsable del intercambio de datos entre el sistema final y la red a la cual está conectado. El emisor debe proporcionar a la red la dirección del destino, de tal manera que esta pueda encaminar los datos hasta el destino apropiado. El emisor puede requerir ciertos servicios que pueden ser proporcionados por el nivel de red. El software en particular que se use en esta capa dependerá del tipo de red que se disponga.

Para sistemas finales conectados a la misma red, la capa de acceso a la red está relacionada con el acceso y encaminamiento de los datos. En situaciones en la que los dos dispositivos estén conectados a redes diferentes, se necesitaran una serie de procedimientos que permitan que los datos atraviesen las distintas redes interconectadas. El protocolo Internet IP se utiliza en esta capa para ofrecer el servicio de encaminamiento a través de varias redes.

Independientemente de la naturaleza de las aplicaciones que estén intercambiando datos, es común requerir que los datos se intercambien de forma fiable. Esto es, seria deseable asegurar que todos los datos llegan a la aplicación destino y en el mismo orden en el que fueron enviados. Los mecanismos que proporcionan esta fiabilidad son esencialmente independientes de la naturaleza intrínseca de las aplicaciones; por tanto, tiene sentido agrupar todos estos mecanismos en una capa común compartida por todas las aplicaciones, esta se denomina capa extremo a extremo o capa de transporte. El protocolo para el control de la transmisión, TCP (*Transmission Control Protocol*), es el más utilizado para proporcionar esta funcionalidad.

La capa de aplicación contiene toda la lógica necesaria para posibilitar las distintas aplicaciones de usuario. Para cada tipo de aplicación, como por ejemplo, la transferencia de archivos, se necesitará un modulo bien diferenciado.

El funcionamiento de una red TCP/IP consiste en transferir datos mediante la unión de bloques de datos en paquetes, cada paquete comienza con una cabecera que contiene información de control; tal como la dirección del destino, seguido de los datos. Cuando se envía un archivo por la red TCP/IP, su contenido se envía utilizando una serie de paquetes diferentes. El *Internet protocol* (IP), un protocolo de la capa de red, permite a las aplicaciones ejecutarse transparentemente sobre redes interconectadas. Cuando se utiliza IP, no es necesario conocer que hardware se utiliza, por tanto ésta corre en una red de área local.

El Transmission Control Protocol (TCP); un protocolo de la capa de transporte, asegura que los datos sean entregados, que lo que se recibe, sea lo que se pretendía enviar y que los paquetes que sean recibidos en el orden en que fueron enviados. TCP terminará una conexión si ocurre un error que haga la transmisión fiable imposible.[3]

2.4.2 Protocolo *Bluetooth:* El protocolo *Bluetooth* es un estándar de comunicación inalámbrica de corto alcance. Los dispositivos dotados de tecnología Bluetooth pueden transmitir información de un dispositivo a otro, en una comunidad cada vez más amplia de innovación electrónica. Al lograr conectividad entre los dispositivos, Bluetooth también elimina los problemas de direccionamiento y enredo de cables, alambres, conectadores y enchufes que por el contrario serían necesarios para enlazar dispositivos.

Bluetooth provee un acuerdo entre dispositivos a nivel físico por medio de transmisión de frecuencia de radio en la banda Médica Científica Industrial (ISM) de los 2,4 gigahercios. También provee un acuerdo con respecto a los protocolos de comunicación: con el propósito de comunicarse con exactitud, los dispositivos deben acordar en la programación y estructura de los datos de modo que se entiendan en ambos extremos de la línea.[4]

Los dispositivos Bluetooth envían datos y voz en un formato digital limpio y claro. Debido a que es digital, la señal de audio no está sujeta a las mismas fuentes de degradación de señal que compromete algunas veces la calidad de transmisiones (FM, AM o inductivas). En la trayectoria de la señal analógica, se amplifica el ruido eléctrico a partir de una variedad de fuentes junto con la señal. En comparación,

[3] URL: http://webuniversitario.ucol.mx/~al980347/modelo_tcp.htm

[4] URL: http://www.elihearing.com.ve/Docs/BluetoothWhatIs1.htm

se extrae una señal Bluetooth digital del ruido; es transmitida y amplificada por sí misma, mientras se rechaza el ruido.

El diseño de baja energía de los sistemas de transmisión Bluetooth tiene dos ventajas. Minimiza el consumo de pila para dispositivos portátiles. También, fija un límite intencional en el rango de transmisión lo cual ayuda a evitar la interferencia entre dispositivos cercanos. Al mismo tiempo, las paredes y otros obstáculos tienen un efecto insignificante en la transmisión Bluetooth.[5]

[5] URL: http://www.elihearing.com.ve/Docs/BluetoothWhatIs1.htm

3. DISEÑO DEL ROBOT

El robot a controlar es un carro autónomo programado por medio de un microcontrolador (DSPIC30F4013) que fue diseñado y construido como un carro seguidor de línea y recoge-bolas.

3.1 SISTEMA DEL ROBOT

En la figura 3 se muestra el diagrama de bloques del sistema del robot, este sistema consta de una fuente de alimentación, un cerebro que es el microcontrolador, unos motores para el movimiento de sus ruedas y el módulo *Bluetooth* para la comunicación con el computador.

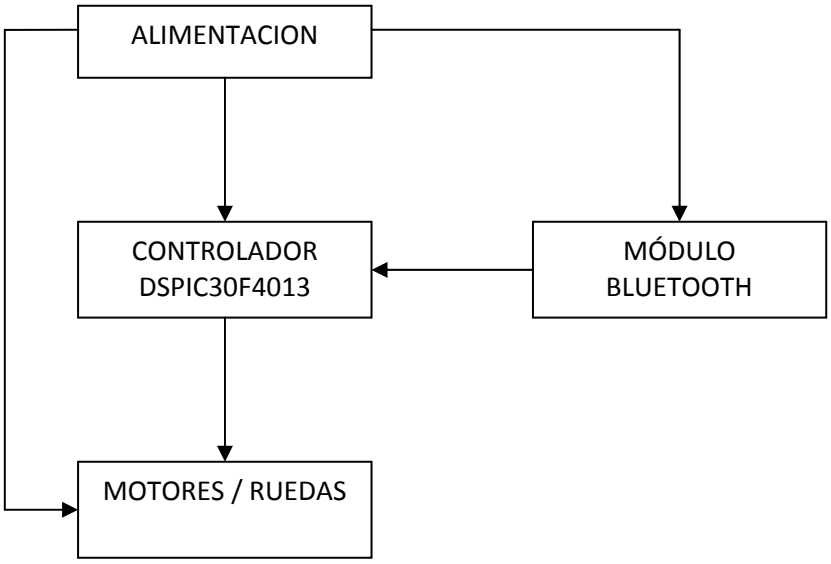

Figura 3. Diagrama de bloques del robot

3.1.1 Alimentación:

El robot se encuentra alimentado por cuatro baterías tipo AA. Su funcionamiento no exige mucha corriente; por esto se necesita un puente H para impulsar más corriente a los motores de las ruedas.

3.1.2 Microcontrolador:

El sistema funciona con un microcontrolador DSPIC30F4013 del fabricante *Microchip;* este microcontrolador es el encargado de activar los motores del robot. Además, posee un pin para comunicación serial que corresponde al módulo UART usado para la comunicación con la tarjeta del módulo *Bluetooth*.

3.2 SERVOMOTORES UTILIZADOS EN EL MOVIMIENTO DE LAS RUEDAS

Las ruedas tienen movimiento por la acción de los servomotores que se encuentran en ellas. Estos servomotores son del tipo *Hobbico Standard CS-60* (ver figura 4), los cuales están trucados, es decir funcionan como un motor DC. El control del sentido de giro de estos motores se hace por medio de un puente H de referencia *L-298* y un programa del microcontrolador.

Figura 4. Servomotor

3.3 ESTRUCTURA DEL ROBOT

En la figura 5 se muestra el robot utilizado en este proyecto. Su estructura está elaborada en madera tipo balso y sus dimensiones son 19 cm de ancho, 20.5 cm de largo y 23.5 cm de alto, aproximadamente.

Figura 5. Estructura del robot a controlar

3.4 PLACA BASE

En la figura 6 se muestra la placa base encargada de impartir las instrucciones a todos los dispositivos conectados a ella; para esto se cuenta con un microcontrolador DSPIC30F4013 con un software que le indica al robot dónde moverse según sean las órdenes. En la placa base se conecta un puente H que se encarga de suministrar la corriente y de invertir el sentido de giro de los motores en cada momento determinado; también, se conecta el módulo *Bluetooth* que se encarga de controlar la comunicación entre el PC y el DSPIC.

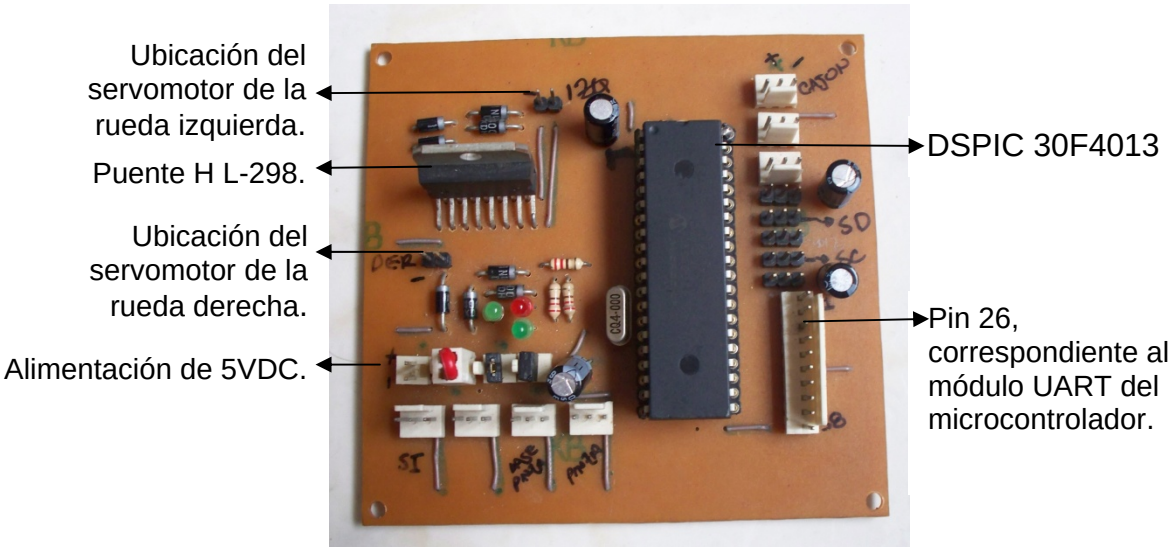

Figura 6. Placa base

4. MÓDULO *BLUETOOTH*

La comunicación inalámbrica del robot con el servidor se realiza utilizando un dispositivo *Bluetooth* de referencia *WRL-00617* de la empresa *SparkFun* mostrado en la figura 7.

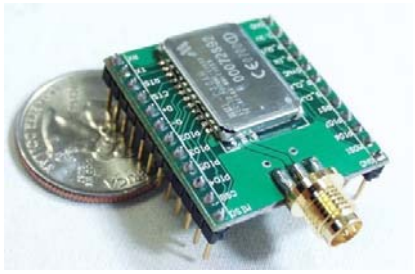

Figura 7. Módulo Bluetooth[6]

4.1 ESPECIFICACIONES DEL MÓDULO *BLUETOOTH* UTILIZADO

Este módulo cuenta con:

- Velocidad mínima de 9600bps
- Ocho bits de datos
- Un bit de parada sin ningún tipo de control por hardware
- Alcance de hasta 100 metros de longitud
- Frecuencia de 2.4GHz
- Alimentación de 3.3V

[6] URL: http://www.sparkfun.com/commerce/product_info.php?products_id=617

Este módulo es de fácil manejo, programable a través de sencillos comandos AT. El único inconveniente que se puede llegar a tener con este módulo es que sólo trabaja con un voltaje de 3.3V, lo que hace necesario una precisa regulación del voltaje para el correcto funcionamiento del mismo.

La figura 8 muestra la configuración del regulador de voltaje LM317. Este regulador es alimentado con 5VDC y por medio de su pin de ajuste se puede obtener a la salida desde 1.2V hasta 37V.

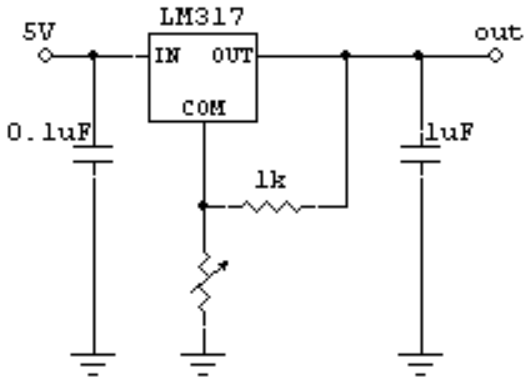

Figura 8. Esquemático de la configuración del LM-317para obtener un voltaje de 3.3V

La regulación anterior debe ser muy precisa ya que el módulo no cuenta con ningún tipo de protección.

4.2 ANTENA

Para obtener una buena transmisión de datos a través de largas distancias es necesario incorporar al módulo *Bluetooth* una antena mostrada en la figura 9. Esta antena entrega 2.2 dBi, tiene 50 ohmios de impedancia, está diseñada para

trabajar a una frecuencia de 2.4 GHz, característica que la hace perfecta para usarla, ya que cumple con las especificaciones necesarias para el correcto funcionamiento del módulo; el tamaño de la antena es de cuatro pulgadas de longitud.[7]

Figura 9. Antena de caucho utilizada para la conexión Bluetooth

4.3 CONEXIÓN DEL MÓDULO *BLUETOOTH*

La conexión de este módulo es muy sencilla debido a que en su parte superior se encuentra impreso el nombre de cada uno de los pines, ofreciendo al usuario una amplia idea de su conexión. Las conexiones necesarias para esta aplicación son las siguientes:

- Los pines 1 y 12 se conectan a tierra.
- El pin 11 es el pin de alimentación y va conectado a la salida del LM317 (pin 2) que es la fuente de 3.3V.

[7] http://www.sparkfun.com/commerce/product_info.php?products_id=145

- Los pines 15 y 16 (RTS y CTS) se puentean entre sí con el fin de hacer el control de flujo.
- El pin 19 es el que indica si existe conexión vía *Bluetooth* con el PC.
- El pin 21 es el que indica si el módulo se encuentra encendido.
- El pin 22 (PIO4) se conecta a tierra. La función de este pin es reiniciar los valores de fábrica del módulo cuando se le entrega un 1 lógico (3.3V).
- El pin 14 (TX) es el pin de transmisión que debe estar conectado al módulo UART del DSPIC, pero debido a que en el pin de transmisión existen 3.3V y el DSPIC trabaja con 5VDC se hace necesario conectar al pin TX del módulo *Bluetooth* un buffer tres estados para obtener 5VDC. Esta configuración es posible con un integrado 74LS125 polarizado a 5VDC y en su entrada la señal proveniente del pin TX del módulo *Bluetooth*; finalmente, su señal de salida se conecta al receptor del módulo UART del DSPIC.

4.4 TARJETA DEL MÓDULO *BLUETOOTH*

En la figura 10 se observa la tarjeta correspondiente al módulo utilizado, señalando cada una de las partes descritas anteriormente.

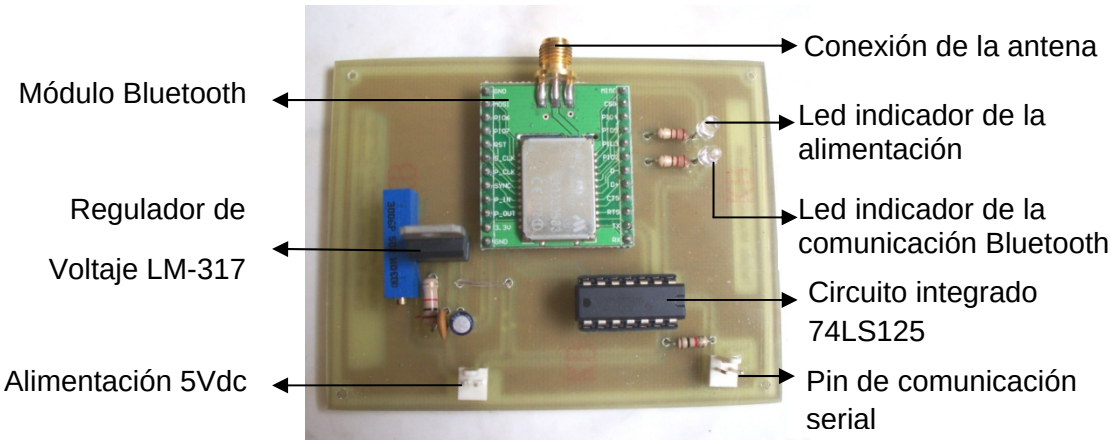

Figura 10. Tarjeta del Módulo Bluetooth

4.5 RECONOCIMIENTO DEL MÓDULO EN EL PC

Para el reconocimiento del módulo *Bluetooth* en el PC se debe tener un programa para acceder a dispositivos con comunicación *Bluetooth*. El programa implementado en este proyecto es el BlueSoleil. En la figura 11 se observa la interfaz gráfica del BlueSoleil. Este programa posee un entorno basado en íconos gráficos ubicados alrededor de un núcleo, cada uno de estos íconos corresponden a todos los dispositivos en cobertura. Además, incluye una serie de íconos para identificar los servicios disponibles o el tipo de conexión: impresión, puerto serie, intercambio de archivos, etc. [8]

[8] URL: http://bluesoleil.softonic.com/

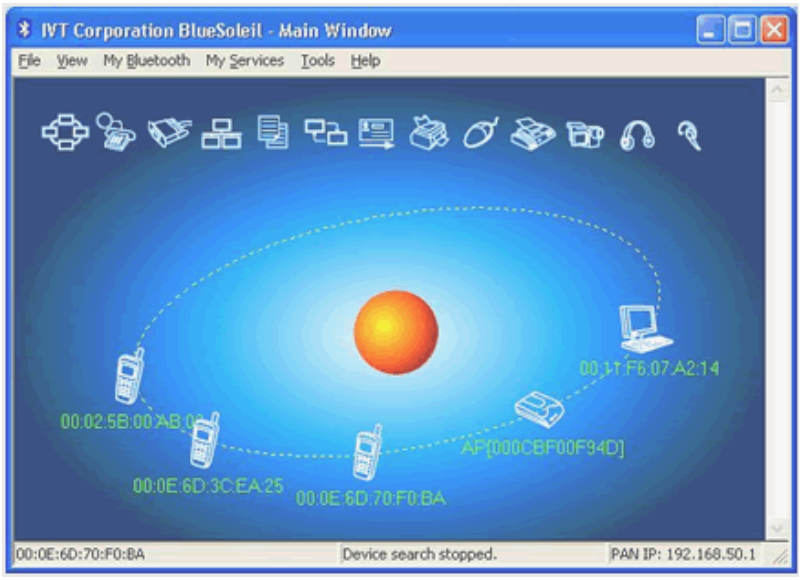

Figura 11. Interfaz gráfica del BlueSoleil[9]

Requisitos mínimos para utilizar el BlueSoleil:

- Sistema operativo: Win2000/XP/Vista
- Procesador: 600 MHz
- Memoria: 128 MB
- Resolución de pantalla: 800x600
- Emisor/Receptor *Bluetooth*.[9]

Una vez descargado el programa se procede a instalarlo. Los pasos que se indican a continuación son los necesarios para instalar el software.

[9] URL: http://bluesoleil.softonic.com/

1. Después de seleccionar la opción de instalar, el programa pide que se elija el idioma en que se ha de observar, esta selección se observa en la figura 12.

Figura 12. Selección del idioma de configuración del BlueSoleil

2. Posterior a la selección del idioma, el asistente de instalación de BlueSoleil presenta una bienvenida (ver figura 13) y presenta su contrato de licencia (ver figura 14).

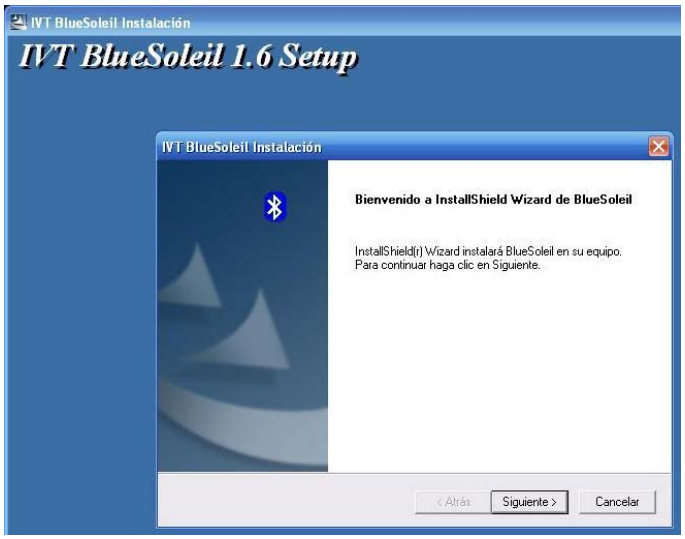

Figura 13. Bienvenida del BlueSoleil

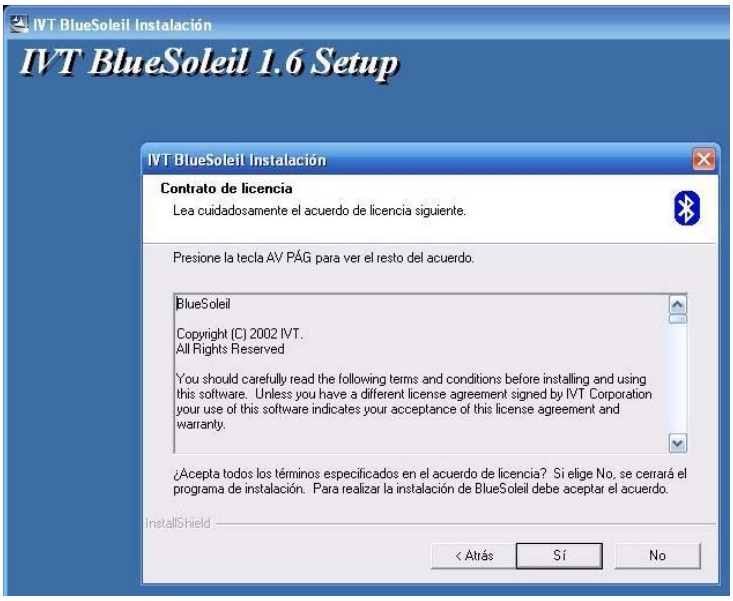

Figura 14. Contrato de licencia

3. Se elige la ubicación donde se quiere instalar el programa. Esta ubicación se observa en la figura 15.

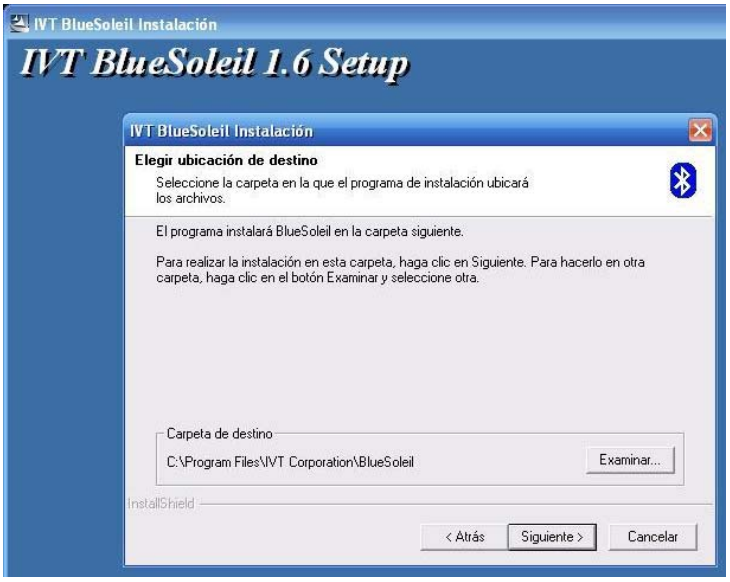

Figura 15. Elección de la ubicación de destino

4. Después de elegir la ubicación del programa, el software se instala automáticamente. El proceso toma unos minutos con el fin de instalar todos y cada uno de los componentes requeridos para su buen funcionamiento; entre ellos, se encuentra la creación de puertos de comunicación del BlueSoleil, es decir, aquellos puertos virtuales que el programa necesita para poder establecer una conexión con diferentes dispositivos *Bluetooth*.

5. Finalizada la instalación se reinicia el PC. Después de haberse reiniciado el equipo aparece la interfaz gráfica del programa mostrada en la figura 16. En esta ventana es donde se puede controlar la conexión entre los dispositivos de comunicación *Bluetooth* y el computador. Para llevar a cabo dicha comunicación es indispensable que el computador cuente con un dispositivo *Bluetooth*, ya sea interno o externo como es el caso de este proyecto, donde el dispositivo *Bluetooth* USB mostrado en la figura 17 se conecta a un puerto virtual COM configurado automáticamente por el sistema.

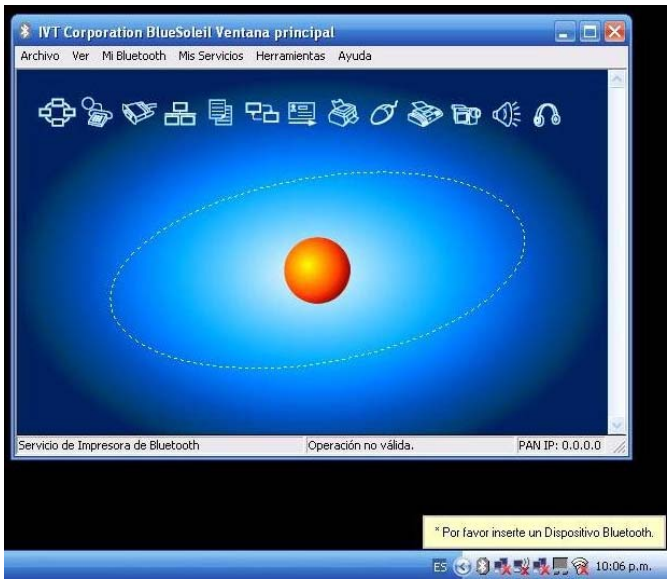

Figura 16. Interfaz gráfica del programa

Figura 17. Módulo Bluetooth USB

5. COMUNICACIÓN ENTRE EL DSPIC Y EL MÓDULO *BLUETOOTH*

La comunicación entre estos dos dispositivos es de tipo serie. Esta comunicación se realiza configurando el módulo UART (*universal asynchronous receiver transmitter*) del DSPIC de tal forma que pueda recibir la información proveniente del módulo *Bluetooth* e interpretarla para realizar la operación respectiva sobre el robot. Entre estos dos dispositivos se encuentra conectado un buffer 74LS125 cuya función es establecer un nivel de 5 voltios a la señal proveniente del módulo.

Para lograr que el DSPIC pueda interpretar la señal proveniente del PC se configura la UART. En la figura 18 se observa el diagrama de flujo del proceso de la configuración del módulo UART. El código en lenguaje ensamblador de la configuración del módulo UART se muestra en el Anexo A.

En el código de programación se muestra la configuración de cada uno de los registros del módulo UART del microcontrolador en donde se fija la velocidad de transmisión (baudios), se configura el control general y el registro status del UART. El registro de control general llamado U1MODE está compuesto de 16 bits donde cada uno permite al usuario configurar la UART como sea necesario; para esta aplicación solo es necesario habilitar el bit 16 del registro con el cual se habilita el módulo UART. El registro U1BRG tiene como función definir la velocidad de transmisión en baudios para que sea compatible con el dispositivo con el que se va realizar la comunicación.

El dato que proviene del módulo *Bluetooth* es almacenado por el microcontrolador en el registro U1RXREG, dicha acción activa una interrupción de recepción permitiendo al dispositivo tomar el dato recibido y manipularlo según como se

haya establecido en su previa programación. Este registro (U1RXREG) está compuesto por un buffer de cuatro datos con estructura FIFO donde el acto de leer el registro moverá la siguiente palabra al tope del registro cumpliendo así con dicha estructura.

Al momento de tomar el dato ingresado el microcontrolador compara dicho valor con uno ya establecido anteriormente, definiendo así hacia dónde se moverá el robot y en qué momento detenerlo si así lo indica el usuario.

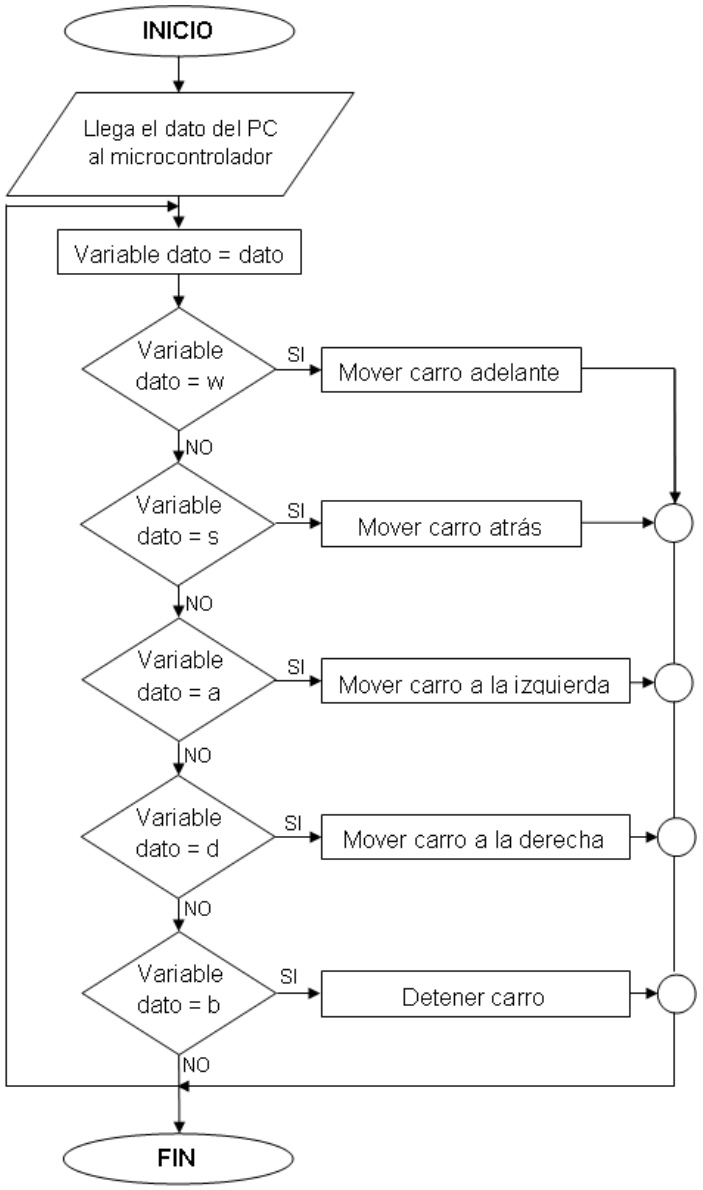

Figura 18. Proceso para la configuración del módulo UART

En la figura 19 se muestra el esquema circuital general del proyecto, en donde se señala la comunicación de la tarjeta del robot (Placa Base) y la tarjeta del módulo *Bluetooth.*

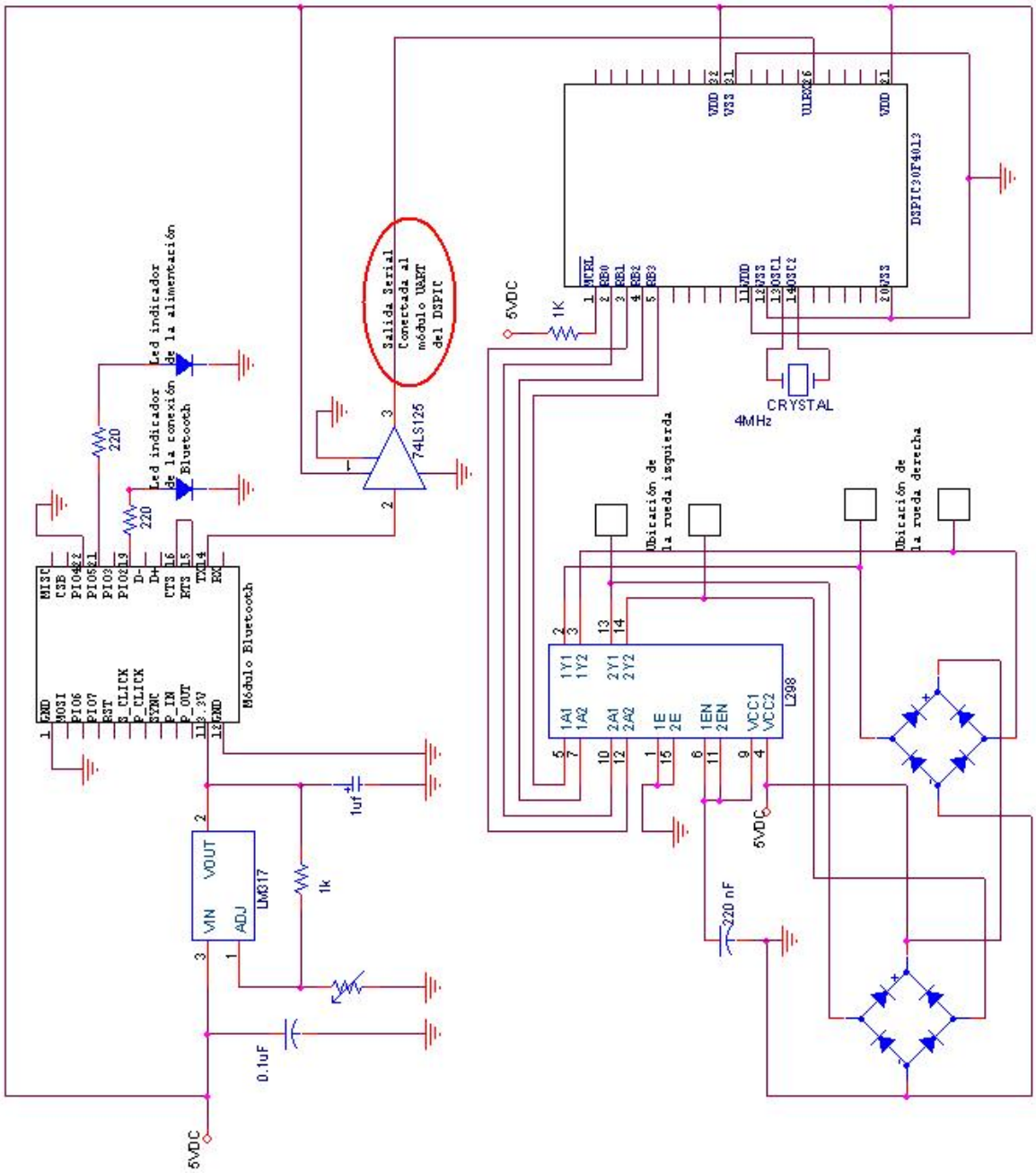

Figura 19. Esquema circuital general

6. TRANSMISIÓN DE VIDEO EN VIVO

Resulta indispensable para la comunicación del cliente con el servidor que el primero visualice el robot si lo que desea el usuario es controlar su ubicación. Para esto se utiliza un software de procesamiento de imágenes (captura y transmisión de video) el cual registre constantemente (haciendo uso de una cámara Web) al robot.

En el momento de escoger este software, se debe tener en cuenta aspectos como:

- La viabilidad que el software presente a la hora de transmitir video a través de Internet.
- Que permita hacer el envío del video a varios equipos clientes simultáneamente.
- Que la transmisión se haga en tiempo real o por lo menos con una diferencia mínima en tiempo.
- Que la transmisión no se haga por un servidor público pues esto podría generar que el servicio fuese suspendido en cualquier momento.

Tomando como base lo anterior es posible encontrar el software indicado para esta tarea, que además de ofrecer todo lo ya mencionado no representa ningún costo para el usuario, dado que es un software ofrecido gratuitamente en Internet y lo mejor, es que es desarrollado por la empresa *Microsoft*, haciéndolo compatible con WINDOWS - el sistema operativo más utilizado – y por supuesto la plataforma de desarrollo del presente proyecto; este software es el WINDOWS MEDIA ENCODER.

Utilizando el programa Windows Media Encoder la tarea de transmitir videos a través de Internet resulta algo muy sencillo. Su función es capturar, codificar y

transmitir video en tiempo real, este proceso es comúnmente conocido como *“Streaming de Video”*.

Además de tener lista la transmisión de video, hay que tener en cuenta la recepción del mismo, para esto se utiliza una herramienta presente en todo PC con sistema operativo *Windows* llamado *Windows Media Player (WMP)*. Para ello se necesita hacer una integración de esta herramienta en el archivo HTML. El código utilizado para realizar dicha integración se muestra en el Anexo B.

Para que esta integración tenga efecto se deben configurar parámetros que identificaran el *WMP* como un reproductor de video como el CLASSID que lo identifica en el campo de la Internet. Adicionalmente, se especifican parámetros de funcionamiento como el URL a ejecutar (lugar desde donde se transmite el video); autostart, que permite al reproductor iniciar automáticamente; uiMode, que activa o desactiva los mandos del reproductor (play, stop, review, next, load), entre otros.

6.1 CONFIGURACIÓN DEL WINDOWS MEDIA ENCODER

Lo primero que se debe hacer para comenzar con la configuración del video en vivo es iniciar el programa *Windows Media Encoder*. En seguida aparecerá una ventana donde se elige la forma de transmisión de video, para este caso *“Broadcast a live event”* que indica una transmisión de video en vivo. Esta ventana se muestra en la figura 20.

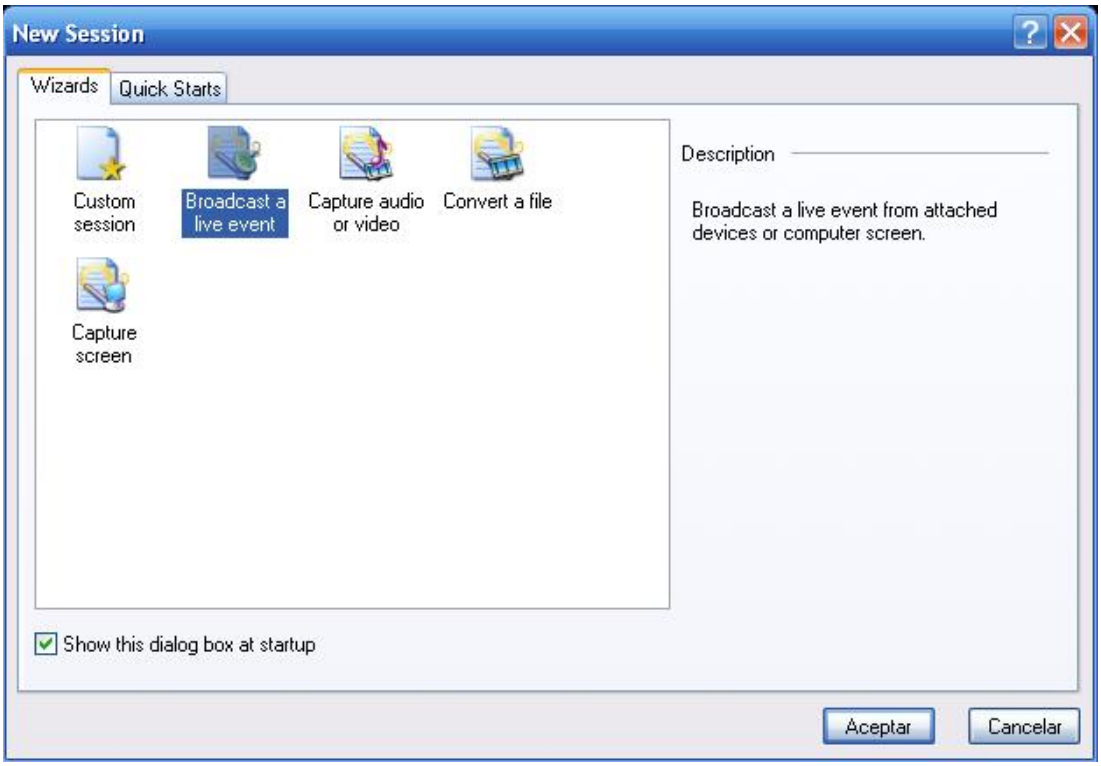

Figura 20. Ventana inicial Windows Media Encoder

Luego de aceptar esta opción, se ingresa a la configuración de una nueva sesión donde se selecciona el dispositivo de audio y video para realizar la transmisión (ver figura 21).

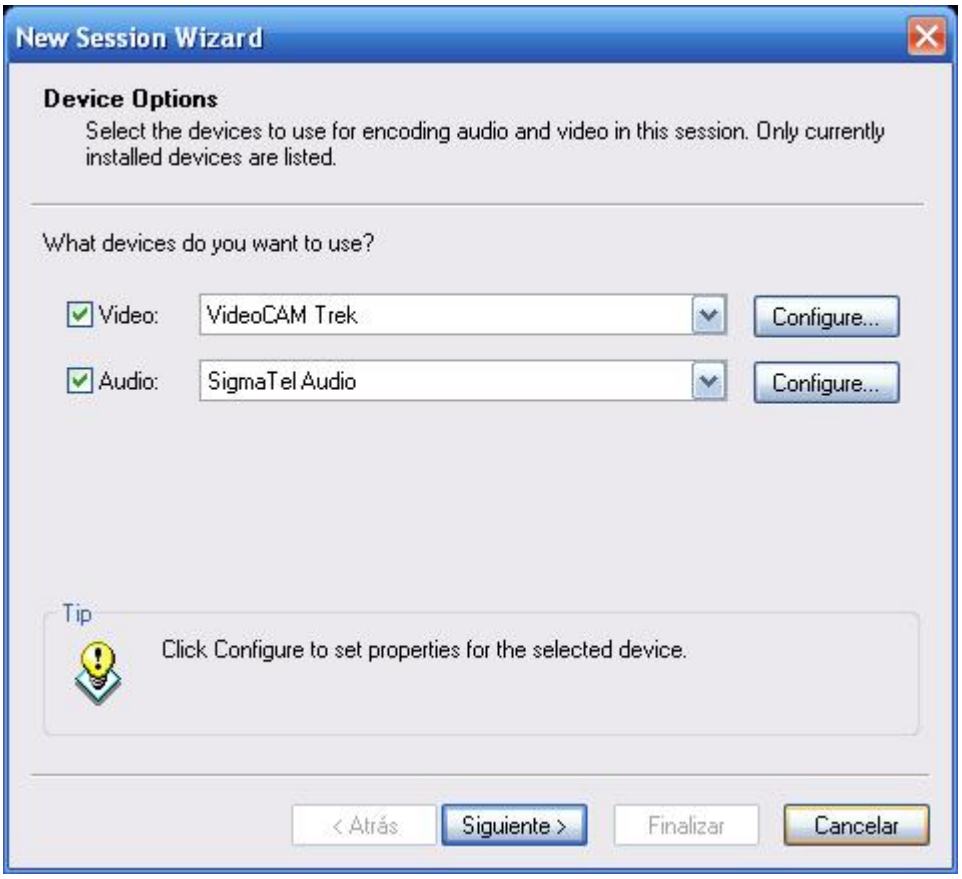

Figura 21. Ventana de configuración de nueva sesión

Después se muestra el puerto a utilizar, la dirección URL y la dirección LAN (si existe) de la conexión que se va a establecer, esta información de conexión se observa en la figura 22, además, los datos mostrados en dicha ventana son importantes al momento de pretender visualizar el video a través de Internet.

Figura 22. Información de conexión

En la siguiente ventana, mostrada en la figura 23, se selecciona el formato del audio y del video. En el momento de realizar esta selección se debe tener en cuenta el ancho de banda con el que se va a manejar la transmisión debido a que si se selecciona un formato muy pesado la transmisión será muy lenta.

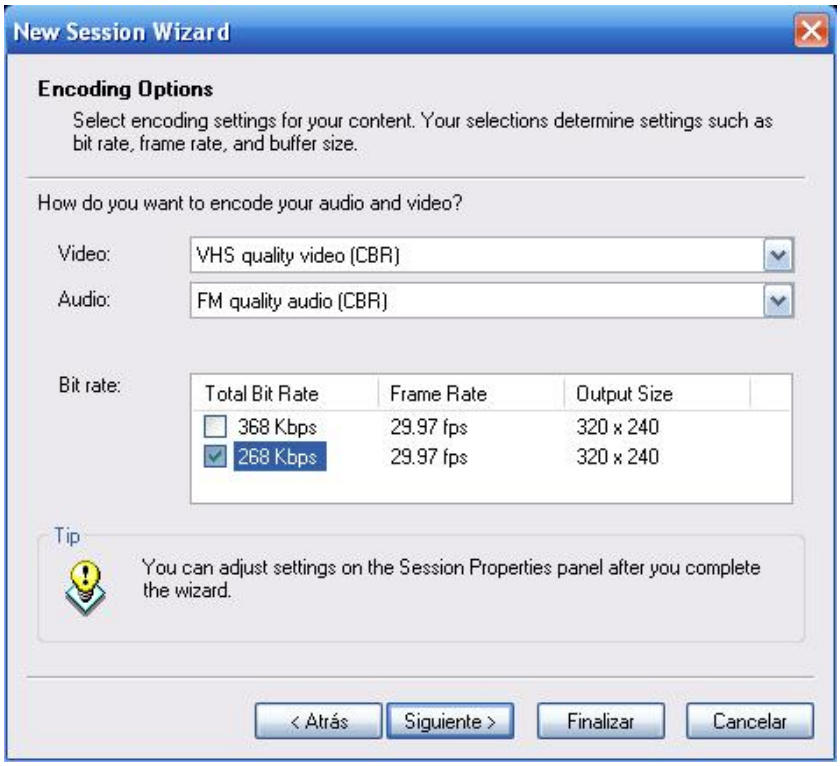

Figura 23. Formato de audio y video

Finalmente, todo está terminado, ya se puede dar inicio a la transmisión de video en vivo. Es en este punto donde se puede ver el video que se trasmite. En la figura 24 se muestra la ventana del Windows Media Encoder donde se aprecia la imagen tomada desde la cámara web (input) y la imagen que sale a través del puerto Broadcast (Output).

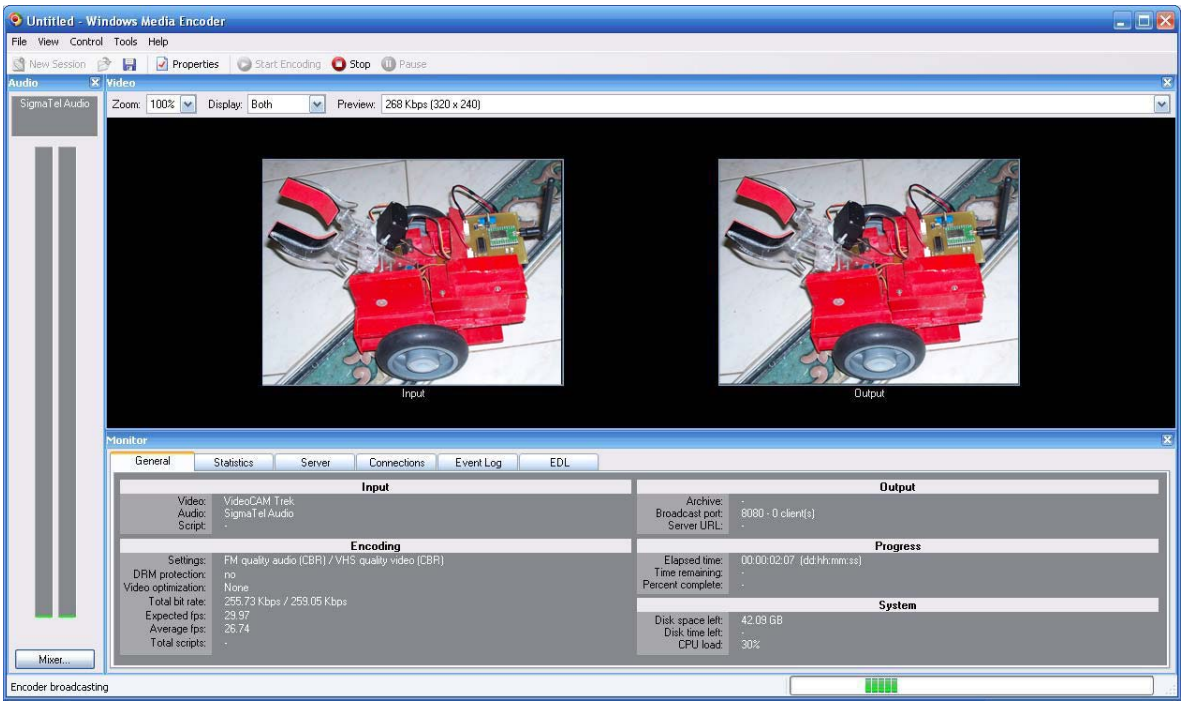

Figura 24. Transmisión del video

7. MANEJO DEL SERVIDOR WEB APACHE

7.1 PASOS PARA LA CONFIGURACIÓN DE APACHE

1. Instalar el servidor Apache (AppServ):

   a. Una vez seleccionado el instalador del AppServ, este presenta una bienvenida (ver figura 25) y el contrato de licencia (ver figura 26); posteriormente se selecciona la ubicación donde será instalado el programa (ver figura 27)

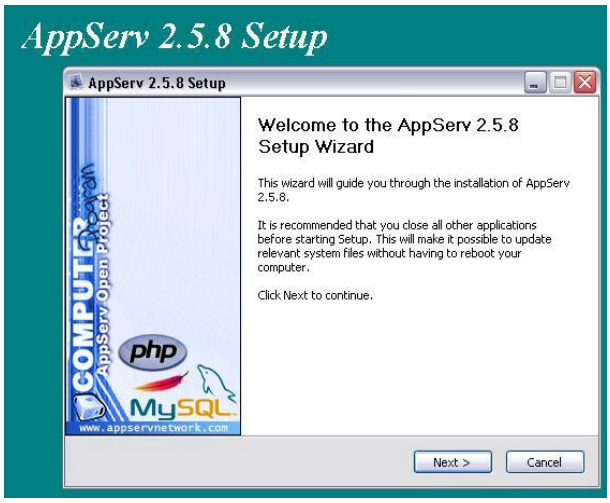

Figura 25. Bienvenida del AppServ

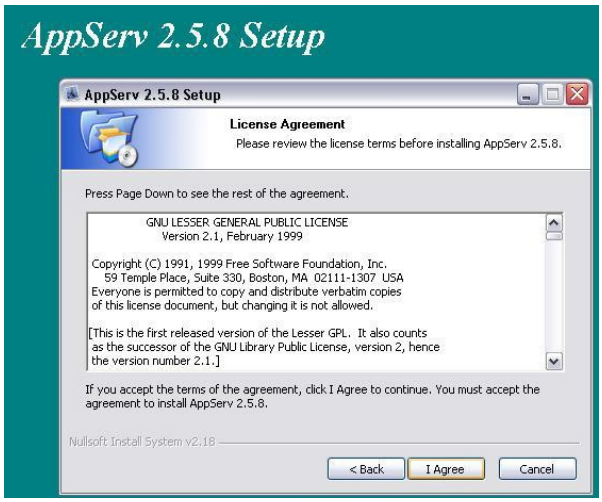

Figura 26. Contrato de licencia del AppServ

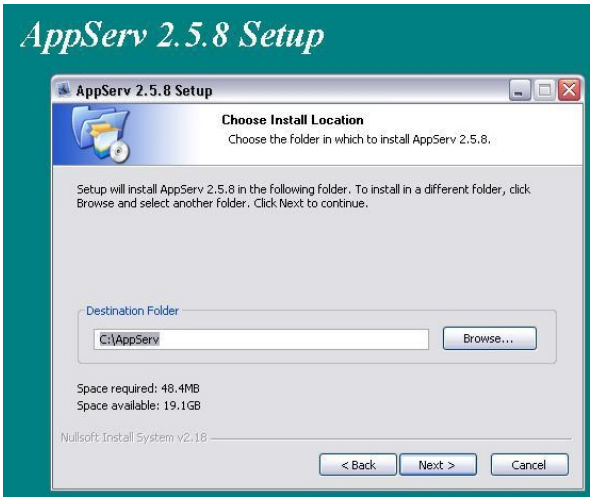

Figura 27. Ubicación de la instalación del AppServ

b. En la figura 28 se muestra una ventana que permite escoger los componentes que se desean instalar, en este caso se seleccionan todas

las opciones: servidor Apache HTTP, base de datos MySQL, procesador de hipertexto PHP y phpMyAdin, así:

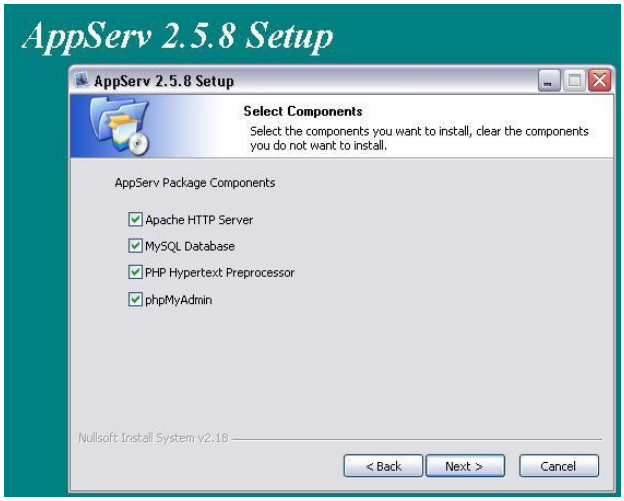

Figura 28. Selección de componentes

c. Posteriormente se solicita la información del servidor web, se escoge el nombre de dominio (localhost) y el correo electrónico del administrador tal y como se muestra en la figura 29. Además, permite escoger el puerto HTTP, que es el puerto manejado por HTML (puerto 80); después de esto se hace la instalación del AppServ. En seguida se reinicia el equipo para que el Sistema Operativo realice los cambios efectuados por el nuevo programa instalado (Ver figura 30).

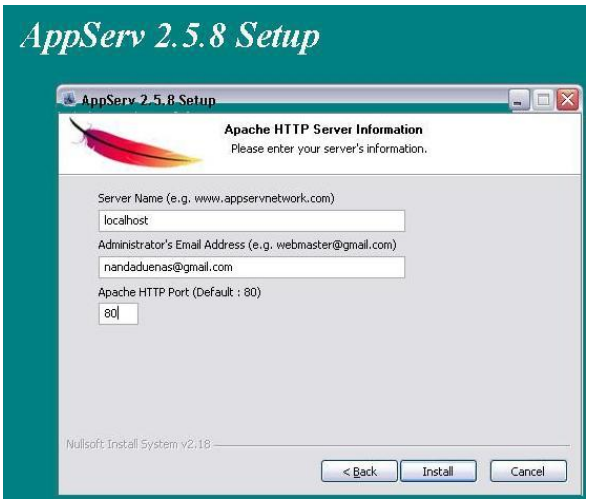

Figura 29. Información del servidor Apache HTTP

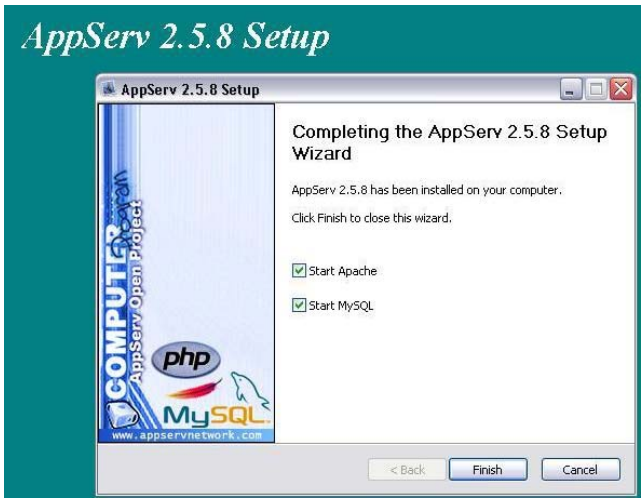

Figura 30. Reinicio del equipo

d. Una vez realizada la instalación, desde la barra de inicio, como se muestra en la figura 31, se ejecuta el AppServ. Cuando se tenga abierto el AppServ en la barra de tareas se muestra el ícono donde se indica que el servidor Apache está funcionando (ver figura 32).

Figura 31. Buscando el programa

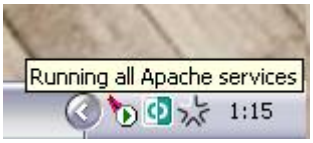

Figura 32. Ícono que muestra que el Apache está funcionando

8. SERVIDOR WEB DE MATLAB

Con el fin de asegurar que se encuentra activo el servidor web de Matlab se hace click derecho sobre el ícono que muestra Apache funcionando y se escoge la opción de abrir servicios (Open services). En la figura 33 se observa esta ventana qué muestra los servicios locales activos, entre ellos Matlab Server.

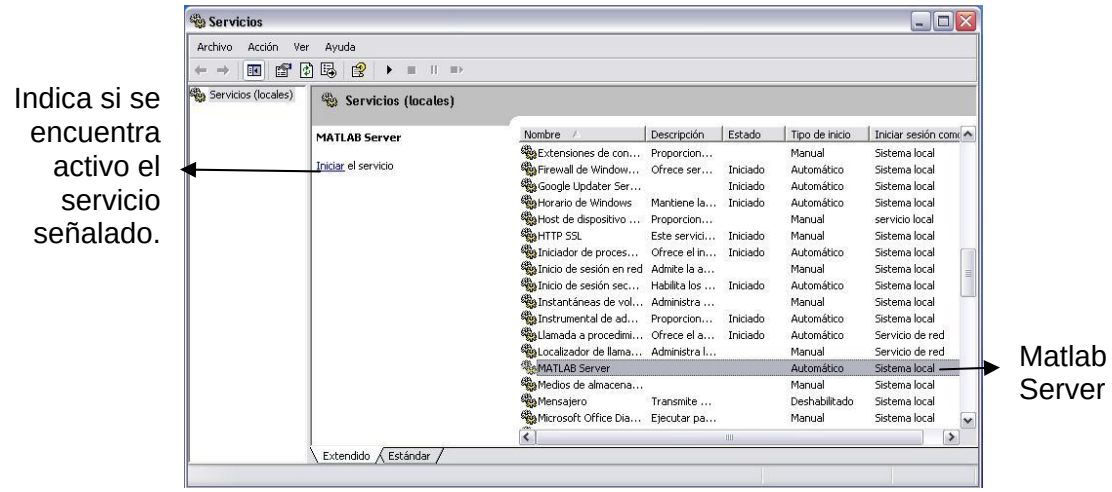

Figura 33. Ventana de servicios del AppServ

Para lograr una interacción entre los dos servidores (Matlab y AppServ) es necesario establecer un enlace entre ellos. Para esto se deben seguir los siguientes pasos.

1. Se Crea un archivo matweb.conf en <*matlab*>*/toolbox/webserver/wsdemos/*, donde <*matlab*> es la dirección raíz donde Matlab se encuentra instalado,

MATLAB crea el archivo matweb.conf automáticamente. El archivo matweb.conf contiene tres datos importantes que son:

- El nombre del archivo .m
- El nombre de dominio del servidor (mlserver).
- La dirección ROOT donde funciona el servidor (mldir).

Inicialmente matweb.conf trae en el mlserver el nombre <matlabserver_host_name> el cual debe ser cambiado por el nombre de dominio, en este caso es: “localhost”. El mldir trae como dirección raíz *<matlab>/toolbox/webserver/wsdemos*, esta dirección raíz debe cambiarse por: *C:/AppServ/www,* que es la dirección del servidor Apache.

2. Se verificar que el archivo matlabserver.conf se cree en <matlab> /webserver. Este archivo contiene la notación “-m 1”, este número representa el número de Matlab’s que pueden ejecutarse simultáneamente.
3. Se copia el archivo matweb.exe localizado en *<matlab>/webserver/bin/win32* en el directorio nombrado como /cgi-bin ubicado en: *C:/AppServ/www/cgi-bin.*
4. Se copia el archivo matweb.conf ubicado en *<matlab>/toolbox/webserver/wsdemos* en la dirección C:*AppServ/www/cgi-bin.*
5. Se copia todos los archivos HTML ubicados en *<matlab>/tollbox/webserver/wsdemos* a la dirección *C:/AppServ/www*, ya que en este directorio es donde deben estar ubicados todos los archivos html.

6. Después de la realizar todos los pasos anteriores se debe reiniciar el equipo para iniciar MATLAB Server como un servicio de Apache. El servicio se inicia automáticamente en el arranque del sistema.[10]

8.1 CONSTRUCCIÓN DE APLICACIONES EN EL SERVIDOR WEB DE MATLAB

Las aplicaciones del servidor Web de MATLAB son una combinación de archivos -m, lenguaje HTML y gráficos.

Los únicos requisitos para manejar este servidor web son dos. El primero es tener conocimientos sobre programación básica en HTML y el segundo es conocer Matlab.

El proceso de crear aplicaciones requiere un número de pasos sencillos:

1. Se crea los documentos HTML para la recolección de datos introducidos por el usuario y mostrar datos de salida. Se puede programar documentos de entrada utilizando un editor de texto o se puede usar un sistema comercialmente capacitado para sistemas HTML como el Front Page de Microsoft, el PageMill de Adobe o el HoTMetaL de SoftQuad.
2. Se introduce el nombre de la aplicación y de los correspondientes datos de configuración en el archivo *matweb.conf.*
3. Se escribe un archivo .m para:
    a. Recibir los datos introducidos por el usuario.
    b. Analizar los datos y generar los gráficos requeridos.

[10] URL: http://som.yale.edu/unix/research_computing/pdf_doc/webserver/webserver.pdf

c. Situar los datos de salida en una estructura de Matlab.

d. Llamar el htmlrep para colocar los datos de salida dentro de un documento HTML de salida. El tamaño máximo de datos HTML que puede recibir de Matlab es de 256KB.[11]

8.2 MANIPULACIÓN DE LAS PLANTILLAS PARA CREAR UNA APLICACIÓN WEB

El proceso de crear una aplicación del servidor Web de Matlab implica la creación de:

- Un documento de entrada HTML para la presentación de datos a Matlab.
- Un documento de salida HTML para mostrar los cálculos realizados por Matlab.
- Un archivo .m para procesar los datos de entrada y computar los resultados.

Este proceso puede ser simplificado a través del uso de una serie de plantillas copiadas anteriormente al servidor Apache. Dichas plantillas tiene como nombre:

- input_template.html: Creación de documentos HTML para la entrada de datos.

[11] URL: http://som.yale.edu/unix/research_computing/pdf_doc/webserver/webserver.pdf

- output_template.html: Documento HTML para mostrar los datos obtenidos.
- mfile_template.m: Documento .m con el cual Matlab procesa los datos.

8.2.1 Creación de documentos de entrada (input_template.html): para crear un documento de entrada se deben seguir los siguientes pasos:

- Primer paso:

En la figura 34 se muestra el código html para definir la plataforma en la cual se va a trabajar. En este caso se ha escogido la versión NT y la parte donde se encuentra *Unix versión* se debe eliminar.

Versión escogida →

```
<!-- STEP 1
Choose either the NT version or the Unix version
of the form tag (depending on which platform the
matweb client program will be run):
-->

<!-- NT version: -->
<form action="/cgi-bin/matweb.exe" method="POST">
<!-- Unix version: -->
<form action="/cgi-bin/matweb" method="POST">
```

Figura 34. Primer paso para la creación del documento de entrada

- Segundo paso:

Crear un campo oculto con el nombre del archivo .m que realiza los cálculos. Reemplazar MY_M_FILE con el nombre de la función principal de Matlab de la aplicación tal y como se muestra en la figura 35.

```
<!-- STEP 2
Create a hidden field naming your M-file.  Replace
MY_M_FILE with the name of main MATLAB function of
your application. (An HTML input field of type "hidden"
is commonly  used to pass variables to a web server.
It is not displayed by the browser.)
-->
<input type="hidden" name="mlmfile" value="my_m_file">
```

Figura 35. Segundo paso para la creación del documento de entrada

- Tercer paso:

En este paso se agrega todo el código html necesario para permitir al usuario introducir los datos. La línea “MY_INPUT_VARIABLE_1” se cambia por el nombre de la variable de entrada que se va a manejar en el archivo .m de Matlab. Ver figura 36.

```
<!-- STEP 3
Add all your other HTML form tags here.  Replace
MY_INPUT_VARIABLE_1 with the name of an input variable in
your application.
-->
<p>My input variable 1: <input type="text" name="my_input_variable_1">


<!--
Create additional input variables here.
-->
```

Figura 36. Tercer paso para la creación del documento de entrada

- Cuarto paso:

Crear un botón por medio del cual el usuario envía las entradas al archivo .m para que sean procesados por Matlab tal y como se indica en la plantilla mostrada en la figura 37.

```
<!-- STEP 4
Create a "submit" input tag for the user to click to send
the input to your program.
-->

<p><input type="submit" name="Submit" value="Submit"></p>

</form>
```

Figura 37. Cuarto paso para la creación del documento de entrada

- Quinto paso:

Este paso no se cumple directamente en el documento html. Consiste en agregar el nombre de la aplicación principal (archivo .m) en el archivo matweb.conf para que sea reconocido por el matweb.exe, como se indica en la figura 38.

```
<!-- STEP 5
Add the name of your main application function to
the file matweb.conf.  See the matweb.conf file in
the wsdemos directory and the documentation.)
-->
```

Figura 38. Quinto paso para la creación del documento de entrada

8.2.2 Creación del archivo .m de Matlab (mfile_template.m): en la plantilla "mfile_template.m" se encuentran los códigos necesarios para aceptar los datos de entrada procedentes del documento HTML de entrada y retornar el resultado hacia el documento HTML de salida. Para crear un archivo .m se deben seguir los siguientes pasos:

- Primer paso:

Se debe reemplazar “mfile_template” por el nombre del archivo .m a crear. La siguiente línea de código inicia el retorno de cadena (ver figura 39).

```
function retstr = mfile_template(instruct, outfile)
% STEP 1
% Initialize the return string.
retstr = char('');
```

Figura 39. Primer paso en la creación del archivo .m

- Segundo paso:

Establecer el directorio de trabajo. La variable instruct.mldir se provee automáticamente a todas las aplicaciones del servidor web que use el programa matweb.exe. Este paso solo es importante si el servidor se va a utilizar para crear aplicaciones gráficas. Este paso se muestra en la figura 40.

```
% STEP 2
% Set working directory.
% The variables INSTRUCT.MLDIR and INSTRUCT.MLID are
provided
% automatically to all MATLAB Web Server applications that
use
% the matweb program.
cd(instruct.mldir);
```

Figura 40. Segundo paso en la creación del archivo .m

- Tercer paso:

Escoger las variables de entrada provenientes del archivo HTML de entrada (ver figura 41)

```
% STEP 3
% Get the HTML form input variables
my_input_variable_1 = instruct.my_input_variable_1;
```

Figura 41. Tercer paso en la creación del archivo .m

- Cuarto paso:

Se introduce la programación necesaria para realizar el procesamiento de la información, la creación de archivos de gráficos y otras acciones necesarias. Este paso se muestra en la figura 42.

```
% STEP 4
% Perform your MATLAB computations, graphics file creations,
% etc., here:
```

Figura 42. Cuarto paso en la creación del archivo .m

- Quinto paso:

Crear las variables destinadas al archivo html de salida. Reemplazar "my_output_variable_1" por la variable a mostrar (ver figura 43).

```
% STEP 5
% Put variables that you want to put into your HTML output
document in an output structure.  You create an HTML output
document from OUTPUT_TEMPLATE.HTML.
outstruct.my_output_variable_1 = More MATLAB computations
creating ...
   scalars, matrices, cell arrays, graphics files, etc.;
```

Figura 43. Quinto paso en la creación del archivo .m

- Sexto paso:

Llamar la función HTMLREP como una estructura de salida creada y un archivo html de salida. Reemplazar “OUTPUT_TEMPLATE.HTML” con el nombre del archivo html de salida como se muestra en la figura 44.

```
% STEP 6
% Call the function HTMLREP with the output structure you
just
```

Figura 44. Sexto paso en la creación del archivo .m

8.2.3 Creación del archivo de salida html (output_template.html): para la creación del documento de salida se deben seguir los siguientes pasos:

- Primer paso:

Mostrar la información enviada por MATLAB en las variables creadas en el archivo .m. Reemplazar “MY_OUTPUT_VARIABLE_1” con el nombre de la variable que se quiere mostrar (ver figura 45).

```
<!-- STEP 1
Display a MATLAB scalar or character string. Replace
<MY_OUTPUT_VARIABLE_1> in the following line with the name
of the MATLAB variable you want to display. Change the other
text to something meaningful within the context of your
application.
-->
My output variable 1 has been computed to be
$<my_output_variable_1>$
```

Figura 45. Primer paso en la creación del archivo de salida

- Segundo paso:

Se introduce el código necesario para mostrarse en la página web como se muestra en la figura 46.

```
<!-- STEP 2
Put all your other HTML tags here.
-->
```

Figura 46. Segundo paso en la creación del archivo de salida

8.3 CREACIÓN DE APLICACIONES UTILIZANDO EL SERVIDOR WEB DE MATLAB

Con el fin de comprender el manejo del servidor web se realizan unos ejemplos sencillos con los que se aprende a usar las plantillas, a continuación se muestran dichos ejemplos:

8.3.1 Primer ejemplo, webnum1.html: este programa consiste en introducir un número con el fin de elevarlo al cuadrado. En la primera página web mostrada en la figura 47, se visualiza un cuadro de texto donde se debe introducir el número, posterior a esto se encuentra un botón con la palabra “CONTINUAR”, una vez se oprime este botón se activa el archivo .m para realizar la operación con el número introducido. Finalmente se muestra la segunda página web mostrada en la figura 48 donde se ve el número introducido y el resultado de elevar dicho número a una potencia dos (operación realizada por Matlab).

Para acceder a este ejemplo, se debe abrir una página de navegador de Internet y como dirección se debe escribir: http://dirección_ip_del_servidor/nombre_del_ejemplo, para este caso es: http://localhost/webnum1.

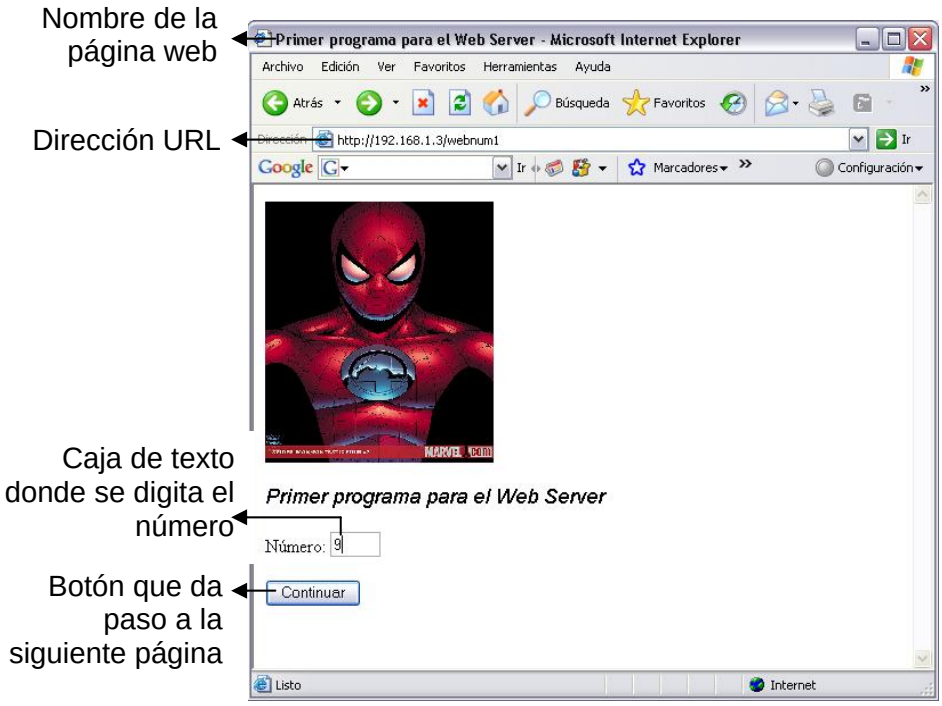

Figura 47. Primer ejemplo del uso del servidor web

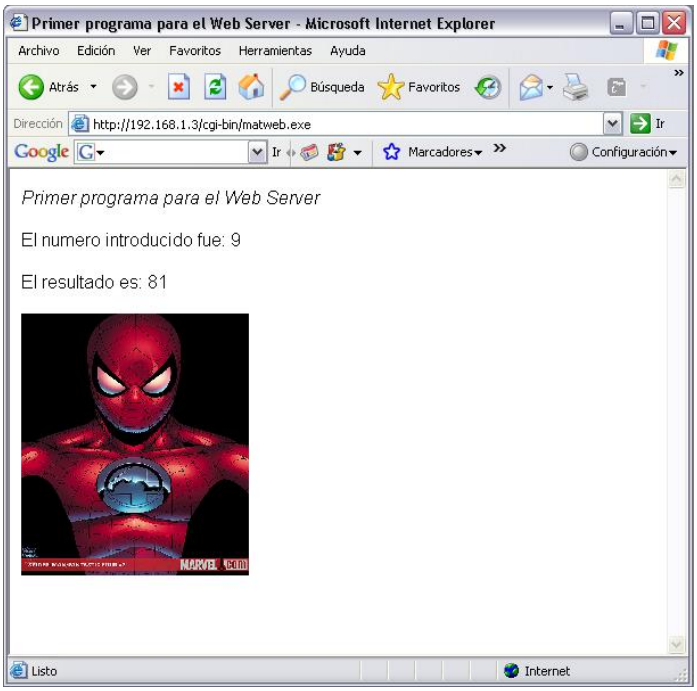

Figura 48. Resultado del primer ejemplo del uso del servidor web

Los códigos de programación de las páginas web de entrada y salida y de las operaciones del primer ejemplo en Matlab se observan en los anexos C, D y E respectivamente.

En la figura 49 se muestra el diagrama de flujo del procedimiento del archivo .m para el primer ejemplo del servidor web

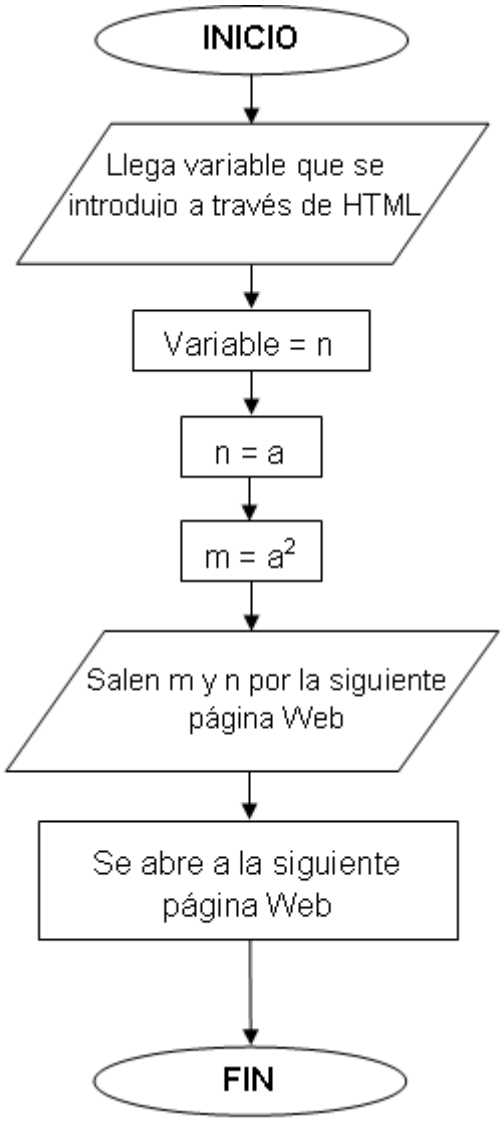

*Figura 49. D*iagrama de flujo del procedimiento del archivo .m para el primer ejemplo del servidor web

8.3.2 Segundo ejemplo, tipog1: este programa consiste en graficar una función trigonométrica en un intervalo elegido por el usuario. En la figura 50 se muestra la primera página web se donde se aprecian tres cuadros de textos para introducir el número menor del intervalo, el número mayor del intervalo y el número de paso que se desee, además de una caja de lista en donde

se puede seleccionar la función trigonométrica que se desea, las opciones son: Sen (seno), Cos (coseno) y Tan (tangente); posterior a esto se encuentra un botón con la palabra “GENERAR”, una vez se oprime este botón se activa el archivo .m para que realice la gráfica correspondiente a los datos introducidos anteriormente. Finalmente, se muestra la segunda página web donde se ve la gráfica generada, esta página se ve en la figura 51.

Para acceder a este ejemplo, se debe abrir una página de navegador de Internet y como dirección se debe escribir: http://dirección_ip_del_servidor/nombre_del_ejemplo, para este caso es: http://localhost/tipog1.

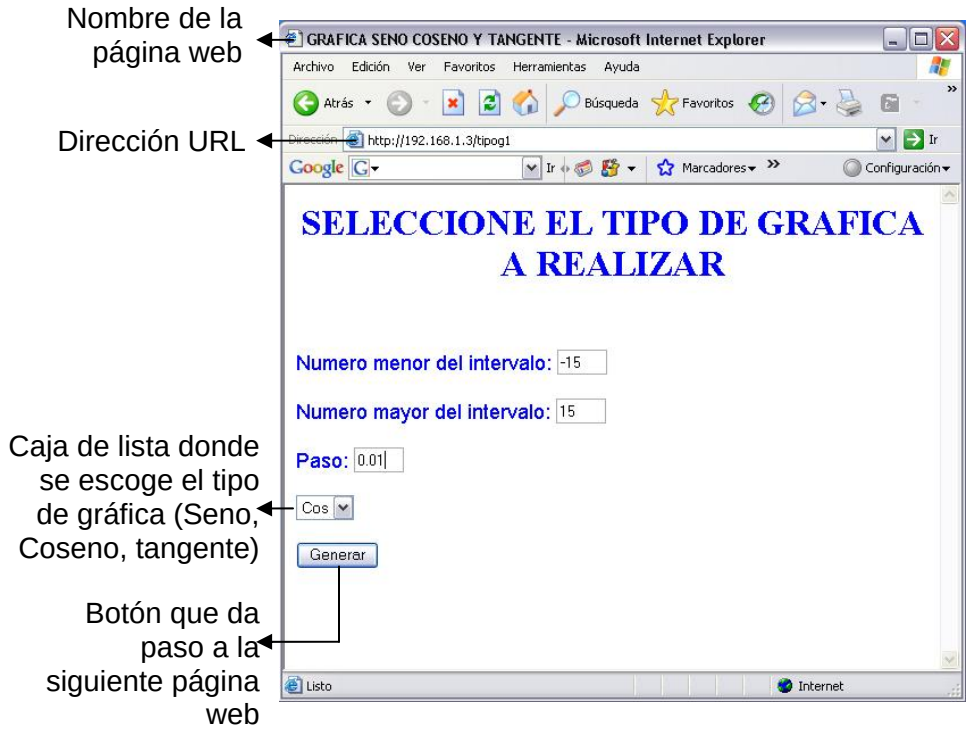

Figura 50. Segundo ejemplo del uso del servidor web

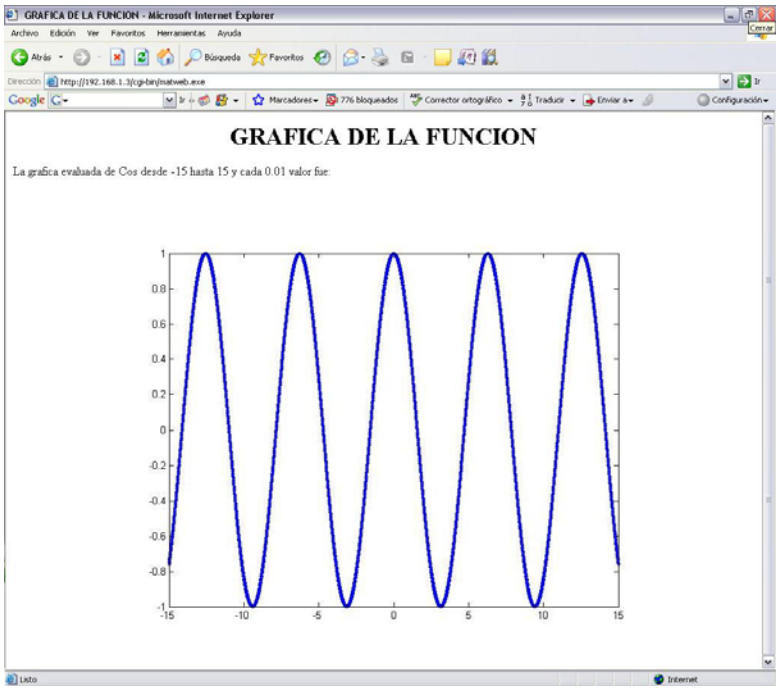

Figura 51. Resultado del segundo ejemplo del uso del servidor web

Los códigos de programación de las páginas web de entrada y salida y de las operaciones del segundo ejemplo en Matlab se observan en los anexos F, G y H respectivamente.

En la figura 52 se muestra el diagrama de flujo del procedimiento del archivo .m para el primer ejemplo del servidor web

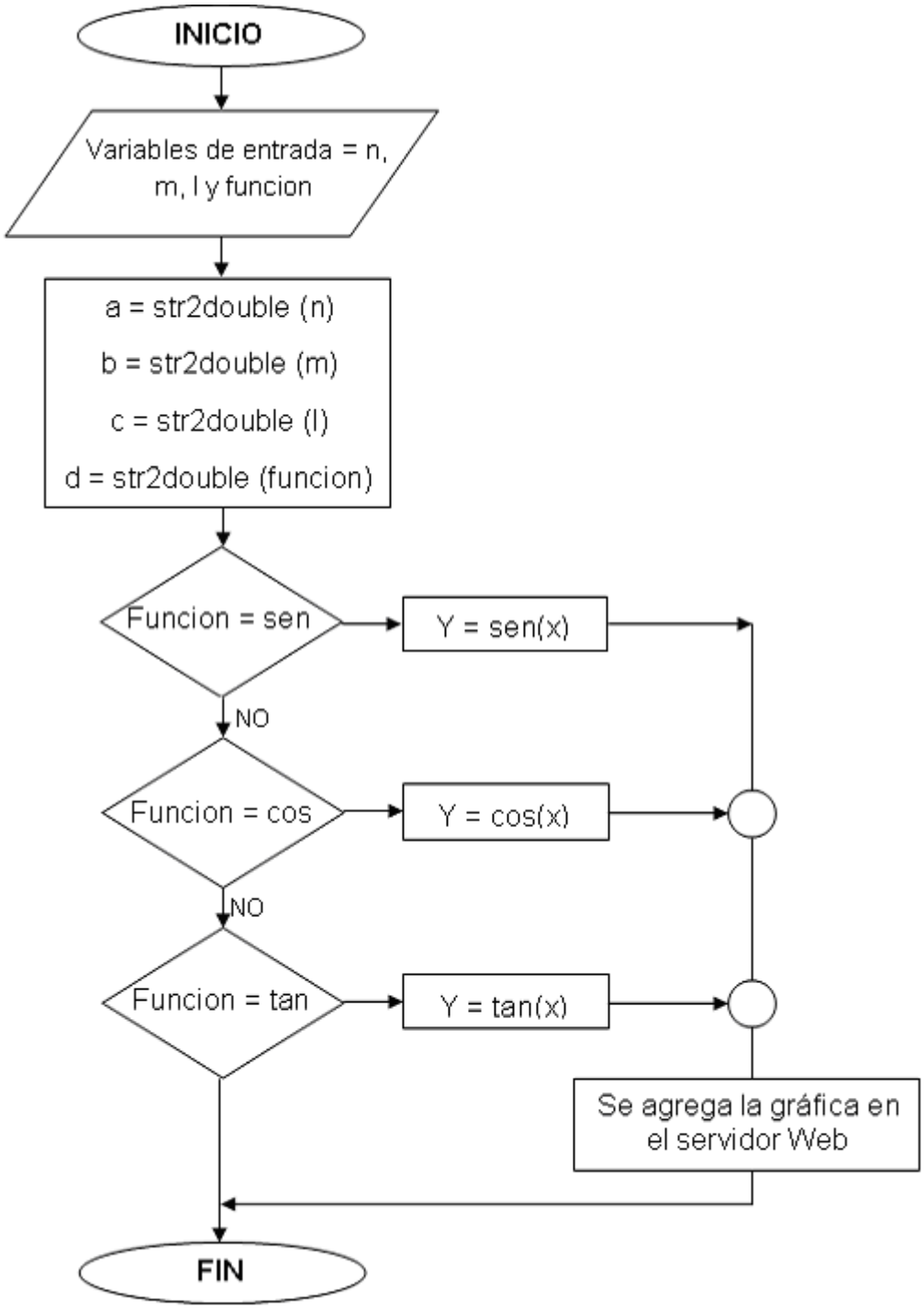

*Figura 52. D*iagrama de flujo del procedimiento del archivo .m para el primer ejemplo del servidor web

9. DISEÑO DE LA APLICACIÓN WEB

9.1 DISEÑO DE LA APLICACIÓN WEB UTILIZANDO EL SERVIDRO WEB DE MATLAB

Después de realizar algunos ejemplos sencillos para aplicar el manejo del servidor web, se ha creado el sitio web por medio del cual se hará posible el control del robot.

Esta aplicación es del todo gráfica ya que se desenvuelve en el ámbito del Internet, por esto es necesario el diseño de unas páginas web que sean del agrado del usuario.

Inicialmente se establecen una serie de páginas para darle la bienvenida al usuario, luego se dirige a una página donde el usuario podrá enviar la instrucción al robot y observarlo a través de una cámara web que está transmitiendo desde el servidor.

En la figura 53 se muestra la primera página del sitio web. Esta página tiene como título el nombre del proyecto «CONTROL DE UN ROBOT CON ACCESO DESDE UN SERVIDOR WEB UTILIZANDO MATLAB», cuatro fotografías del robot, las cuales van siendo mostradas una seguida de otra automáticamente, una caja de texto para introducir el nombre del usuario y un botón llamado “ENTRAR”. En el instante que el usuario presione dicho botón se activa el archivo .m vinculado a esta página para que se ejecute creando una serie de variables a ser mostradas en la siguiente página web.

El código en lenguaje HTML del archivo de entrada del servidor web y el código del archivo .m para las dos primeras páginas del servidor web se encuentran en los anexos I y J respectivamente.

Figura 53. Primera página del sitio web final

En la figura 54 se muestra el diagrama de flujo del archivo .m para las dos primeras páginas del servidor web.

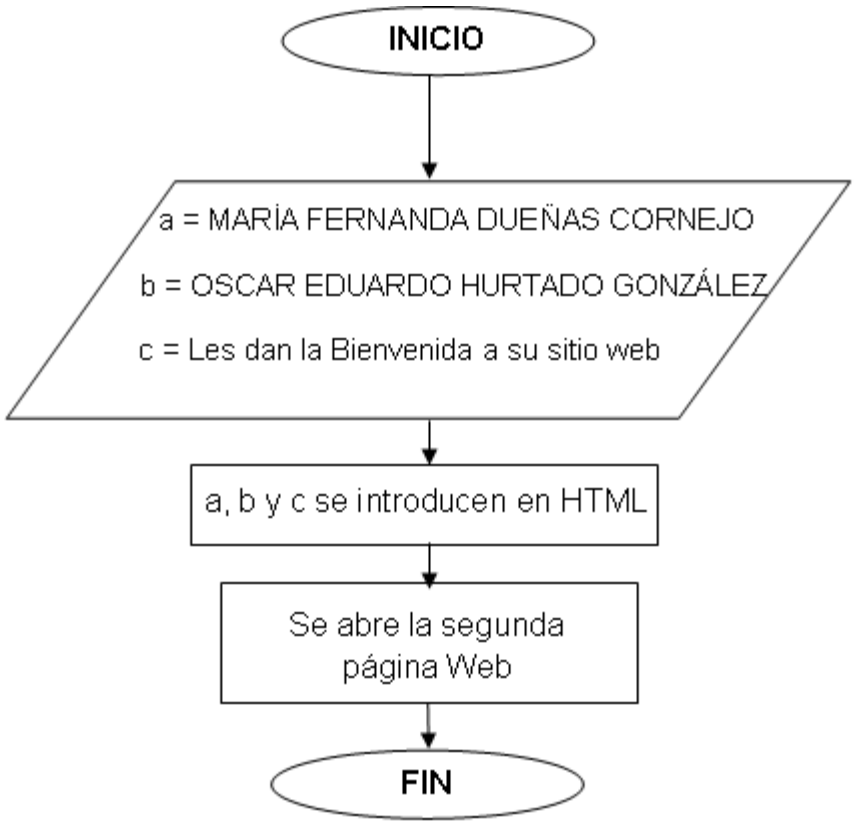

Figura 54. Diagrama de flujo del archivo .m para las dos primeras páginas del servidor web

- En la figura 55 se muestra la segunda página web con los mensajes creados por Matlab donde se da una bienvenida al usuario. Su código HTML se encuentra en el anexo K.

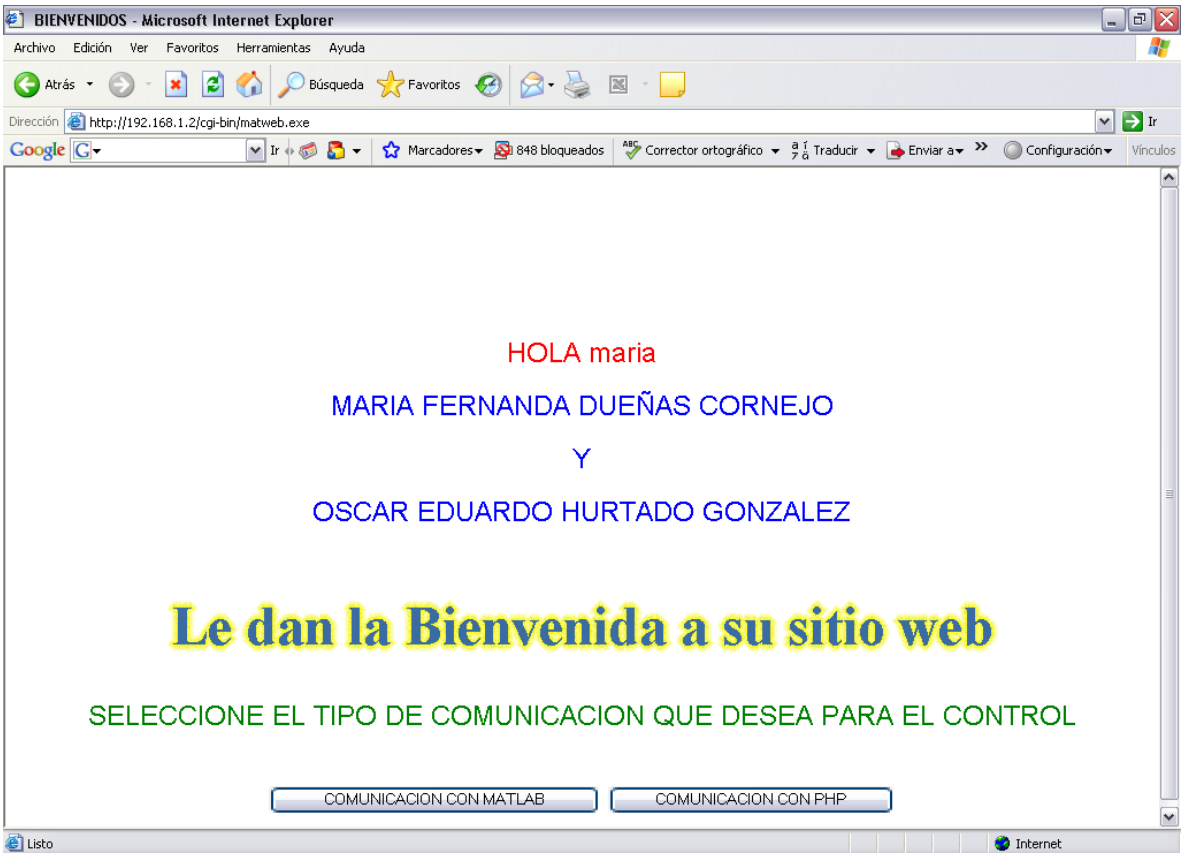

Figura 55. Segunda página del sitio web final

- La tercera página se abre una vez pulsado el botón llamado “COMUNICACIÓN CON MATLAB” y consiste en introducir un dato para ser enviado a través de la red y controlar el robot. La tercera página se muestra en la figura 56.

Figura 56. Tercera página del sitio web final

En la parte izquierda de esta página se observa el video del robot. Para que el dato introducido sea enviado a través de la red el usuario debe pulsar el botón "ENVIAR DATO". Este dato es guardado en una variable establecida en el código HTML de la tercera página. Una vez presionado el botón se activa el archivo .m para establecer conexión con el puerto, toma el dato que el usuario introduce y lo envía a través del puerto serie conectado al módulo Bluettoth.

Después de enviarse el dato se refresca automáticamente la página para que el usuario pueda enviar más datos para controlar el robot. El video siempre se encuentra disponible debido a que la construcción de la página se hizo en *"FRAMES"*, razón por la cual nunca se verá afectado por efectos de envíos de datos o manipulación del puerto a través de Matlab.

El código en lenguaje HTML del archivo de salida utilizado para enviar el dato al servidor web y el código del archivo .m para enviar el dato a través del puerto serie se muestran en los anexos L y M respectivamente.

En la figura 57 se muestra el diagrama de flujo del archivo .m para enviar el dato a través del puerto serie.

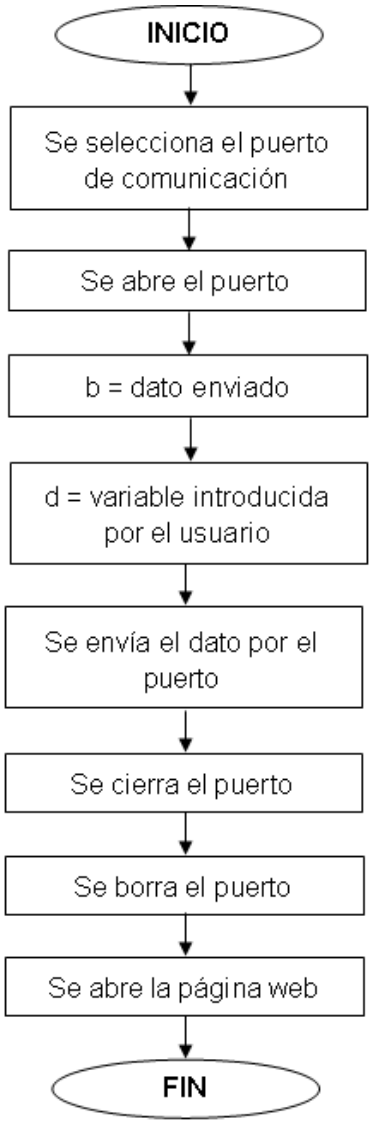

*Figura 57. D*iagrama de flujo del archivo .m para enviar el dato a través del puerto serie

La aplicación anteriormente desarrollada presenta una demora de 17 segundos al momento de establecer la comunicación serie con el equipo servidor. Esto se convierte en una aplicación muy lenta debido a que cada vez que se desee enviar una instrucción al robot se debe establecer dicha comunicación. Es por esto que se implementa otro medio de comunicación y envío de datos por el puerto serie al robot.

9.2 DISEÑO DE LA APLICACIÓN WEB UTILIZANDO LENGUAJE PHP

El propósito de esta nueva implementación es reducir considerablemente el tiempo que toma la comunicación con el puerto serie del servidor. Queriendo optimizar esta aplicación se ha usado el lenguaje PHP en la apertura del puerto y el envío de datos a través de él.

En la figura 58 se muestra la página que se abre al momento de pulsar el botón llamado “COMUNICACIÓN CON PHP” que se encuentra en la segunda página del sitio web. En esta página se encuentra un grupo de botones que al ser pulsados envían instrucciones al robot según lo desee el usuario. Dentro de ese grupo de botones se encuentra un botón llamado “EJECUTAR RUTINA” con el cual se ejecuta en el robot una rutina de movimientos previamente establecida. También se puede observar el robot a través del *“Streamming de Video”* observado en la parte izquierda de la página.

Para el movimiento del robot, se pulsan las flechas que indican “ADELANTE”, “ATRAS”, “DERECHA”, “IZQUIERDA” y el botón de stop para detener el robot; cada uno de estos botones al ser pulsado genera un llamada a una página PHP

que controla la comunicación y el envío del respectivo dato a través del puerto serie.

El código en lenguaje HTML de la página web se muestra en el anexo N.

El código de uno de los botones (adelante) con lenguaje PHP se muestra en el anexo O.

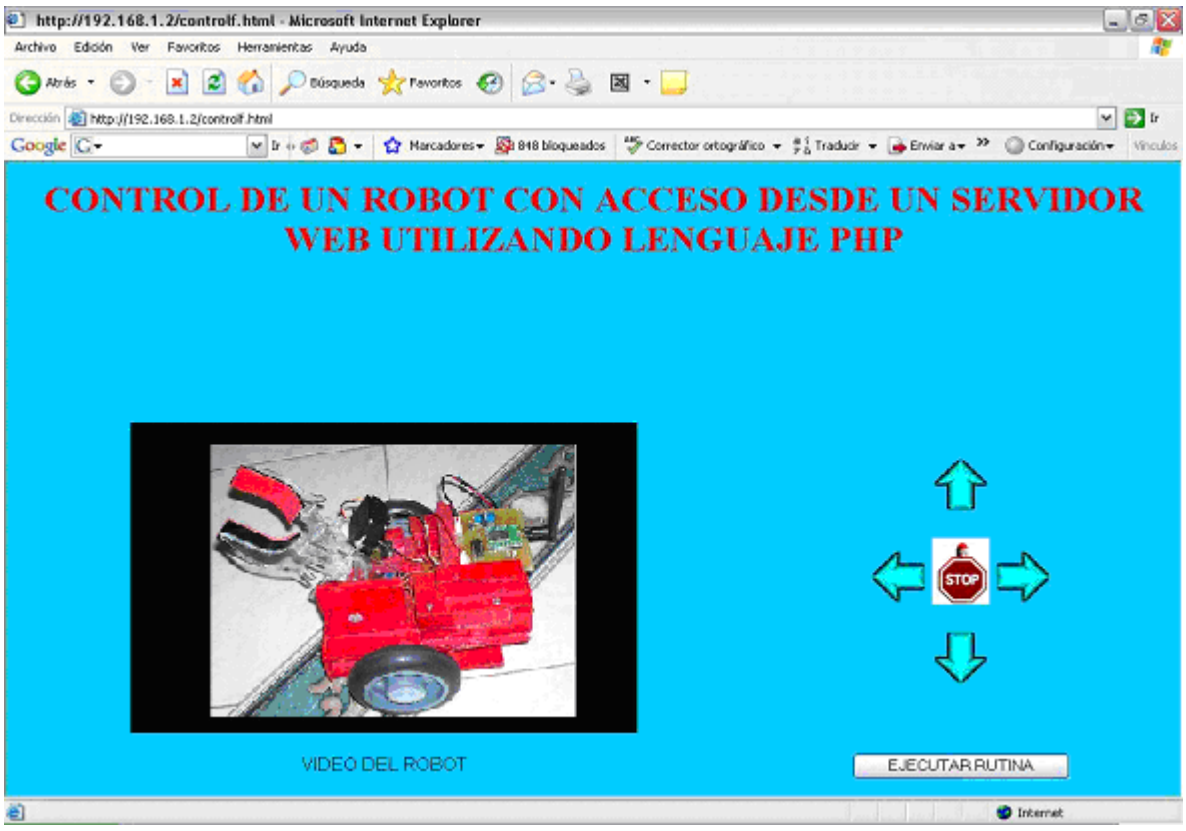

Figura 58. Página web utilizando lenguaje PHP

10. CONCLUSIONES

Siguiendo la metodología señalada en el proyecto, se llega a las siguientes conclusiones:

1. La implementación de un control remoto facilita la interacción de gran cantidad de usuarios con un mismo entorno al mismo tiempo y sin necesidad de acceder directamente a ellos.

2. Las aplicaciones a Laboratorios de Control Virtual son muy útiles en ambientes en que es restringido el ingreso de seres humanos debido a que dichos entornos podrían resultar peligrosos.

3. Una aplicación remota requiere de la buena instalación de los software involucrados en ella, ya que si se omite la selección de algún componente importante o por el contrario se selecciona uno que no va a servir para el fin de la aplicación es bastante probable que esta no funcione correctamente.

4. Además de la buena instalación de los software, una aplicación remota también necesita de un ancho de banda lo suficiente para transmitir video y enviar los datos necesarios sin que el sistema colapse, ya que dependiendo de la aplicación donde se utilice se necesita una respuesta rápida y una visualización adecuada a ciertas situaciones que se puedan llegar a presentar.

5. Matlab es una herramienta muy útil en el procesamiento de datos, pero para implementar esta herramienta como envío de datos a través de un puerto se

convierte en una aplicación muy pesada, por lo que dicho envío demora tiempo en llegar a su destino para realizar la acción correspondiente.

6. Con este proyecto se innovó en la implementación de una nueva herramienta educativa, resultando muy útil debido a que el estudiante puede realizar una práctica de control desde cualquier lugar con el mismo resultado que si estuviera personalmente en el laboratorio.

7. Con la utilización de un módulo *Bluetooth* en la comunicación del equipo servidor con el robot, se hizo más fácil la comunicación entre un elemento fijo y uno móvil ya que se evitan problemas con cables, conectores, circuitos impresos, entre otros.

8. Si se desea tener una buena conexión entre el microcontrolador del robot y el módulo Bluettoth, es necesario tener muy presente la velocidad de transmisión de datos (baudios) del módulo utilizado en la configuración del módulo UART del microcontrolador, debido a que no siempre estos dispositivos manejan la misma velocidad y el mismo alcance, generando así confusiones en el usuario acerca del buen funcionamiento del sistema.

9. Con el fin de simplificar los circuitos electrónicos utilizados se acude a conocimientos básicos de configuración de tensión y de esta manera los circuitos son alimentados con una misma fuente lineal de cinco voltios.

RECOMENDACIONES

1. Estudiar nuevas alternativas de programación para reducir el tiempo que toma la aplicación en trasmitir los datos a través del puerto serie.

2. Desarrollar una aplicación para los módulos del laboratorio de control en la UPB.

3. Montar el servidor con un *HOSTING* y un nombre de dominio público con el fin de acceder a este desde cualquier lugar del mundo y a una mayor velocidad.

4. Estudiar la viabilidad para introducir algoritmos de control vía *web*.

BIBLIOGRAFÍA

ANGULO USATEGUI, José. Microcontroladores Avanzados dspic: Controladores Digitales de Señales. Arquitectura, Programación y Aplicaciones. Ed. Thompson. Australia 2006.

GARCÍA, Luis F.; SARMIENTO, Tatiana M.; DELGADO, Alberto. Control de un robot móvil mediante el uso de un Gameboy Advance con fines pedagógicos. Universidad Nacional de Colombia.

GIL R., Francisco Javier; TEJEDOR C. Jorge A.; PANADERO, Agustín Y.; VILLAVERDE, Santiago Alonso; GUTIERREZ R., Abraham. *Creación de Sitios Web con PHP 4*. Ed. Mc Graw Hill. Madrid, España. 2003.

GOMEZ COLORADO, Jesús; GARCÍA MACIAS, Manuel; GARCÍA GARCÍA, Pilar; ESPINA GARCÍA, Felipe; SÁNCHEZ GUTIERREZ, Javier; HUERTAS TALON, Enrique; SALGADO BENITO, Javier; MATIN, Hilario. Proyecto CRR: Control Remoto de Robots.

PORTER, Lynnette R. Creating the Virtual Classroom. Ed. John Wiley & Sons, Inc. United States of America. 1997.

PUERTO MANCHÓN, Rafael; JIMENEZ GARCÍA, Luis Miguel; FERNÁNDEZ PEROS, César; ÑECO GARCÍA, Ramón P; GARCÍA ARACIL, Nicolás. RECOLAB: Prácticas de control sobre procesos reales vía Internet utilizando Matlab. Universidad Miguel Hernández. España.

STALLINGS, William. Comunicaciones y Redes de Computadores. Editorial Prentice Hall. Séptima Edición. 2005.

VARELA M, A; VALLÉS, M; TORNERO, J. LABCONROB: Laboratorio remoto de control en tiempo real de sistemas robotizados. Universidad Politécnica de Valencia.

ANEXO A

Código de programación en lenguaje ensamblador del módulo UART

```
.EQU    RET1,#0X0804
.EQU    RET2,#0X0806
.EQU    RET3,#0X0808
.EQU    FRONT,#0X080A
.EQU    LEFT,#0X080C
.EQU    RIGHT,#0X080E
.EQU    BACK,#0X0810
.EQU    ABRIR,#0X0812
.EQU    CERRAR,#0X0814
.EQU    SUBIR,#0X0816
.EQU    BAJAR,#0X0818
.EQU    DETENER,#0X081A
.EQU    DATO,#0X081C
.EQU    B_ON,#0X081E
.EQU    VT_PWM1,#0X0820
.EQU    VB_PWM1,#0X0822
.EQU    VT_PWM2,#0X0824
.EQU    VB_PWM2,#0X0826
.EQU    VT_PWM3,#0X0828
.EQU    VB_PWM3,#0X082A
.EQU    VA_PWM1,#0X082C
.EQU    VA_PWM2,#0X082E
.EQU    VA_PWM3,#0X0830
.EQU    C_B_PWM,#0X0832
.EQU    RET11,#0X0834
.EQU    RET12,#0X0836
.EQU    RET13,#0X0838
.EQU    B_T,#0X083A

.include        "P30F4013.inc"
.global         __reset
.global         __U1RXInterrupt
.global         __T1Interrupt
.text
__reset:        GOTO    INICIO

__T1Interrupt:  BTSS    B_ON,#00
                GOTO    M2
                CALL    PWM1
M2:             BTSS    B_ON,#01
                GOTO    M3
                CALL    PWM2
M3:             BTSS    B_ON,#02
                GOTO    M4
                CALL    PWM3
M4:             MOV     #0X0010,W0
                MOV     W0,PR1
                BCLR    IFS0,#T1IF
                CLR     B_ON
                RETFIE

__U1RXInterrupt: MOV    U1RXREG,W0
                MOV     W0,DATO
                CLR     U1RXREG
                CLR     U1RXREG
                CLR     U1RXREG
                CLR     U1RXREG
                BCLR    IFS0,#U1RXIF
                RETFIE
;=====================================
RETARDO:        MOV     #0X002,W0
                MOV     W0,RET1
                MOV     #0X00B,W0
                MOV     W0,RET2
                MOV     W0,RET3
SALTO3:         DEC     RET1,WREG
                MOV     WREG,RET1
                BRA     NZ,SALTO1
                RETURN
SALTO1:         DEC     RET2,WREG
                MOV     WREG,RET2
                BRA     NZ,SALTO2
                GOTO    SALTO3
SALTO2:         DEC     RET3,WREG
```

```
                MOV     WREG,RET3
                BRA     NZ,SALTO2
                GOTO    SALTO1
;===================================
RETARDO3:       MOV     #0X002,W0
                MOV     W0,RET11
                MOV     #0X003,W0
                MOV     W0,RET12
                MOV     W0,RET13
SALTO20:        DEC     RET11,WREG
                MOV     WREG,RET11
                BRA     NZ,SALTO15
                RETURN
SALTO15:        DEC     RET12,WREG
                MOV     WREG,RET12
                BRA     NZ,SALTO22
                GOTO    SALTO20
SALTO22:        DEC     RET13,WREG
                MOV     WREG,RET13
                BRA     NZ,SALTO22
                GOTO    SALTO15
;===================================
INICIO:         MOV     #0XFFFF,W0
                MOV     W0,ADPCFG
                MOV     #0X0836,W15
                MOV     #0X0F09,W4
                MOV     W4,SPLIM
                MOV     #0X8010,W1
                MOV     W1,T1CON
                MOV     #0X0000,W0
                MOV     W0,TRISB
                MOV     W0,TRISC
                MOV     #0XFFFF,W0
                MOV     W0,TRISD
                MOV     W0,TRISF
                BCLR    IFS0,#T1IF
                BSET    IEC0,#T1IE
                BSET    SR,#IPL0
                BSET    SR,#IPL1
                BCLR    SR,#IPL2
                BCLR    CORCON,#IPL3
                MOV     #0X008,W0
                MOV     W0,PR1
                CLR     PORTB
                CLR     PORTC
                CLR     PORTD
                CLR     PORTF
                CLR     TMR1
                MOV     #0X050,W0
                MOV     W0,C_B_PWM
                MOV     #0X0006,W0
                MOV     W0,U1BRG
                MOV     #0X8000,W0
                MOV     W0,U1STA
                MOV     #0X8000,W0
                MOV     W0,U1MODE
                BSET    IPC2,#U1RXIP2
                BCLR    IPC2,#U1RXIP1
                BCLR    IPC2,#U1RXIP0
                CLR     IEC1
                CLR     IEC2
                MOV     #0X00F7,W0
                MOV     W0,FRONT
                MOV     #0X00E1,W0
                MOV     W0,LEFT
                MOV     #0X00E4,W0
                MOV     W0,RIGHT
                MOV     #0X00F3,W0
                MOV     W0,BACK
                MOV     #0X00EA,W0
                MOV     W0,ABRIR
                MOV     #0X00EC,W0
                MOV     W0,CERRAR
                MOV     #0X00E9,W0
                MOV     W0,SUBIR
                MOV     #0X00EB,W0
                MOV     W0,BAJAR
                MOV     #0X00E2,W0
                MOV     W0,DETENER
                CLR     B_ON
                BSET    U1STA,#UTXEN
                BSET    IEC0,#U1RXIE
CICLO:          MOV     DATO,WREG
                SUB     FRONT,WREG
                BRA     Z,ADELANTE
                MOV     DATO,WREG
                SUB     LEFT,WREG
                BRA     Z,IZQUIERDA
                MOV     DATO,WREG
                SUB     RIGHT,WREG
```

```
                BRA     Z,DERECHA
                MOV     DATO,WREG
                SUB     BACK,WREG
                BRA     Z,ATRAS
                MOV     DATO,WREG
                SUB     ABRIR,WREG
                BRA     Z,OPEN
                MOV     DATO,WREG
                SUB     CERRAR,WREG
                BRA     Z,CLOSE
                MOV     DATO,WREG
                SUB     SUBIR,WREG
                BRA     Z,UP
                MOV     DATO,WREG
                SUB     BAJAR,WREG
                BRA     Z,DOWN
                MOV     DATO,WREG
                SUB     DETENER,WREG
                BRA     Z,STOP
                GOTO    CICLO
;=======================================
M1_STOP:        BCLR    PORTB,#03
                NOP
                BCLR    PORTB,#02
                RETURN

M2_STOP:        BCLR    PORTB,#01
                NOP
                BCLR    PORTB,#00
                RETURN

M1_ADELANTE:    BCLR    PORTB,#03
                NOP
                BSET    PORTB,#02
                RETURN

M2_ADELANTE:    BSET    PORTB,#01
                NOP
                BCLR    PORTB,#00
                RETURN

M1_ATRAS:       BSET    PORTB,#03
                NOP
                BCLR    PORTB,#02
                RETURN

M2_ATRAS:       BCLR    PORTB,#01
                NOP
                BSET    PORTB,#00
                RETURN
;=================================
ADELANTE:       CALL    M1_ADELANTE
                CALL    M2_ADELANTE
                GOTO    CICLO
IZQUIERDA:      CALL    M1_ADELANTE
                CALL    M2_STOP
                GOTO    CICLO
DERECHA:        CALL    M1_STOP
                CALL    M2_ADELANTE
                GOTO    CICLO
ATRAS:          CALL    M1_ATRAS
                CALL    M2_ATRAS
                GOTO    CICLO
STOP:           CALL    M1_STOP
                CALL    M2_STOP
                GOTO    CICLO
OPEN:           BSET    B_ON,#01
                MOV     #9,W0
                MOV     W0,VT_PWM2
                MOV     W0,VB_PWM2
                CALL    RETARDO3
                GOTO    CICLO
CLOSE:          BSET    B_ON,#01
                MOV     #20,W0
                MOV     W0,VT_PWM2
                MOV     W0,VB_PWM2
                CALL    RETARDO3
                GOTO    CICLO
UP:             BSET    B_ON,#00
                MOV     #30,W0
                MOV     W0,VT_PWM1
                MOV     W0,VB_PWM1
                CALL    RETARDO3
DOWN:           BSET    B_ON,#00
                MOV     #04,W0
                MOV     W0,VT_PWM1
                MOV     W0,VB_PWM1
                CALL    RETARDO3
                GOTO    CICLO
;=====================================
```

```
PWM1:   BTSS   B_T,#00
        GOTO   TB1
        GOTO   TH1
TB1:    DEC    VA_PWM1,WREG
        MOV    WREG,VA_PWM1
        BRA    NZ,ER
        MOV    C_B_PWM,WREG
        MOV    WREG,VA_PWM1
        BSET   PORTC,#14
        BSET   B_T,#00
ER:     RETURN
TH1:    DEC    VB_PWM1,WREG
        MOV    WREG,VB_PWM1
        BRA    NZ,AR
        MOV    VT_PWM1,WREG
        MOV    WREG,VB_PWM1
        BCLR   PORTC,#14
        BCLR   B_T,#00
AR:     RETURN
;=======================================
PWM2:   BTSS   B_T,#01
        GOTO   TB2
        GOTO   TH2
TB2:    DEC    VA_PWM2,WREG
        MOV    WREG,VA_PWM2
        BRA    NZ,IR
        MOV    C_B_PWM,W0
        MOV    W0,VA_PWM2
        BSET   PORTC,#13
        BSET   B_T,#01
IR:     RETURN
TH2:    DEC    VB_PWM2,WREG
        MOV    WREG,VB_PWM2
        BRA    NZ,ORR
        MOV    VT_PWM2,W0
        MOV    W0,VB_PWM2
        BCLR   PORTC,#13
        BCLR   B_T,#01
ORR:    RETURN
;=======================================
PWM3:   BTSS   B_T,#02
        GOTO   TB3
        GOTO   TH3
TB3:    DEC    VA_PWM3,WREG
        MOV    WREG,VA_PWM3
        BRA    NZ,UR
        MOV    C_B_PWM,W0
        MOV    W0,VA_PWM3
        BSET   PORTB,#9
        BSET   B_T,#02
UR:     RETURN
TH3:    DEC    VB_PWM3,WREG
        MOV    WREG,VB_PWM3
        BRA    NZ,URR
        MOV    VT_PWM3,W0
        MOV    W0,VB_PWM3
        BCLR   PORTB,#9
        BCLR   B_T,#02
URR:    RETURN
        .END
```

ANEXO B

Código HTML para embeber el WMP en Internet

```
<HTML>
<HEAD><TITLE>VIDEO EN VIVO</TITLE></HEAD>
<BODY>
<OBJECT id="VIDEO" width="440" height="280"
                style="position:absolute; left:50;top:50;"
                CLASSID="CLSID:6BF52A52-394A-11D3-B153-00C04F79FAA6"
                type="application/x-oleobject">

                <PARAM NAME="URL" VALUE="http://192.168.50.1:8080/">
                <PARAM NAME="SendPlayStateChangeEvents" VALUE="True">
                <PARAM NAME="AutoStart" VALUE="True">
                <PARAM NAME="uiMode" value="none">
                <PARAM NAME="PlayCount" value="9999">
</OBJECT>
</BODY>
</HTML>
```

ANEXO C

Código HTML de entrada de datos del primer ejemplo del servidor web

```
<html>
<head>
<title>Primer programa para el Web Server</title>
</head>

<body bgcolor="#FFFFFF">
<p><font color="#000000" size="4" face="Arial">
<i><img src="imagenes/spider.jpg" width="200" height="230"></i></font></p>
<p><font color="#000000" size="4" face="Arial">
<i>Primer programa para el Web Server</i></font></p>

<!-- <MY_INPUT>.html = webnum1.html -->

<!-- STEP 1 NT version-->
<form action="/cgi-bin/matweb.exe" method="POST">


<!-- STEP 2 MY_M_FILE = mwebnum.m-->
<input type="hidden" name="mlmfile" value="mwebnum">

<!-- STEP 3 MY_INPUT_VARIABLE_1 = n-->
<p>Número: <input type="text" size="3" name="n"></p>

<!-- STEP 4 -->

<p><input type="submit" name="Submit" value="Continuar"></p>

</form>

<!-- STEP 5
Add the name of your main application function to
the file matweb.conf. -->

<p> </p>

</body>
</html>
```

ANEXO D

Código HTML de salida de datos del primer ejemplo del servidor web

```
<html>
<head>
<title>Primer programa para el Web Server</title>
</head>

<body bgcolor="#FFFFFF">

<p><font color="#000000" size="3" face="Arial">
<i>Primer programa para el Web Server</i></p>

<!-- STEP 1 -->
<p>El numero introducido fue: $n$</p>
<p>El resultado es: $m$</p>

<!-- STEP 2 -->
<p><font color="#000000" size="4" face="Arial">
<i><img src="/imagenes/spider.jpg" width="200" height="230"></i></font></p>

</font>
</body>
</html>
```

ANEXO E

Archivo .m para el primer ejemplo del servidor web

```
 function retstr = mwebnum(instruct, outfile)

% STEP 1
retstr = char('');

% STEP 2
%cd(instruct.mldir);

% STEP 3
n = instruct.n;

% STEP 4
a = str2double(instruct.n);
m = a^2;

% STEP 5
outstruct.n = n;
outstruct.m = m;

% STEP 6
templatefile = which('webnum2.html');
   retstr = htmlrep(outstruct, templatefile);
```

ANEXO F

Código HTML de entrada de datos del segundo ejemplo servidor web

```
<html>
<head>
<h1><center><b><font color=blue>SELECCIONE EL TIPO DE GRAFICA A
REALIZAR</font></b></center></h1>
</head>

<TITLE>GRAFICA SENO COSENO Y TANGENTE</TITLE>

<body>

<form action="/cgi-bin/matweb.exe" method="POST">

<input type="hidden" name="mlmfile" value="mtipog">

<BR><BR>
<FONT size=4 color=blue face=arial>
<p>Numero menor del intervalo: <input type="text" size="3" name="n">
<p>Numero mayor del intervalo: <input type="text" size="3" name="m">
<p>Paso: <input type="text" size="3" name="l">

<p>
<select name=funcion>
<option>Sen
<option selected>Cos
<option>Tan
</select>
</font>

<p><input type="submit" name="Submit" value="Generar"></p>

</form>
</body>
</html>
```

ANEXO G

Código HTML de salida de datos del segundo ejemplo servidor web

```
<html>
<HEAD>
<CENTER><H1><B>GRAFICA DE LA FUNCION</B></H1></CENTER>
</HEAD>

<TITLE>
GRAFICA DE LA FUNCION
</TITLE>

<body>

<!-- STEP 1-->

La grafica evaluada de $funcion$ desde $n$ hasta $m$ y cada $l$ valor fue:

<p> </p>
<p align="center">
<img src="/imagenes/grafun.jpeg">
</p>

</body>
</html>
```

ANEXO H

Archivo .m para el segundo ejemplo del servidor web

```
function retstr = mtipog(instruct, outfile)
retstr = char('');
cd('C:\AppServ\www\cgi-bin');
wscleanup('grafun.jpeg',1);

n = instruct.n;
m = instruct.m;
l = instruct.l;
funcion = instruct.funcion;
a = str2double(instruct.n);
b = str2double(instruct.m);
c = str2double(instruct.l);
d = str2double(instruct.funcion);
Fig = figure('visible','off');
if funcion == 'Sen'
   for x = a:c:b;
   y = sin(x);
   plot(x,y,'.'),hold on
   end
elseif funcion == 'Cos'
   for x = a:c:b;
   y = cos(x);
```

```
    plot(x,y,'.'),hold on
    end
else funcion == 'Tan'
    for x = a:c:b;
    y = tan(x);
    plot(x,y,'.'),hold on
    end
end
PlotFile = sprintf('grafun.jpeg', '/imagenes/');
drawnow;
wsprintjpeg(Fig, PlotFile);
close(Fig);
outstruct.n = n;
outstruct.m = m;
outstruct.l = l;
outstruct.funcion = funcion;

templatefile = which('tipog2.html');
   retstr = htmlrep(outstruct, templatefile);
if ( exist('outfile','var') == 1 );
  s.GraphFileName = [ PlotFile];
  PageString = htmlrep(s, templatefile, outfile);
else
  s.GraphFileName = ['/imagenes/' PlotFile];
  PageString = htmlrep(s, templatefile);
End
```

ANEXO I

Código HTML de entrada de la página web

```
<html>
<head>
<h1><center><b><font color=blue>CONTROL DE UN RBOTO CON ACCESO
DESDE UN SERVIDOR WEB UTILIZANDO MATLAB</font></b></center></h1>

<script language="javascript" type="text/javascript">
var SecuenciaEjecutandose = false
var SecuenciaID = null
var imagen = 0
var duracion = 1500

if (CompruebaVersion()) {
        imagenes = new CreaArray(4)
        //carga las imagenes: especificar aqui las URLs de las imagenes o ruta
fisica y nombre
        imagenes[1].src = "/imagenes/nandabot1.jpg"
        imagenes[2].src = "/imagenes/nandabot2.jpg"
        imagenes[3].src = "/imagenes/nandabot3.jpg"
        imagenes[4].src = "/imagenes/nandabot4.jpg"
}

function CompruebaVersion() {
        if (navigator.appVersion.charAt(0) >= 3  && document.images) return true
        else return false
}
function CreaArray(n) {
        this.length = n
        for (var i = 1; i<=n; i++) {
                this[i] = new Image()
        }
        return this
}
function DetenerSecuencia (){
        if(SecuenciaEjecutandose)
                clearTimeout(SecuenciaID)
        SecuenciaEjecutandose = false
       imagen = 0
```

```
}
function MostrarSecuencia () {
        if (CompruebaVersion()) {
         document.images["secuencia"].src = imagenes[imagen].src
         imagen++
         if ( imagen == 5 )
            imagen = 1
        }
        SecuenciaID = setTimeout("MostrarSecuencia()", duracion)
        SecuenciaEjecutandose = true
}
function IniciarSecuencia () {
        DetenerSecuencia()
      imagen = 1
        MostrarSecuencia()
}

window.onload = IniciarSecuencia;
if (document.captureEvents) {                    //N4 requiere invocar la funcion
captureEvents
        document.captureEvents(Event.LOAD)
}
</script>

</head>

<TITLE>CONTROL DE UN RBOTO CON ACCESO DESDE UN SERVIDOR WEB
UTILIZANDO MATLAB</TITLE>

<body>

<form action="/cgi-bin/matweb.exe" method="POST">

<input type="hidden" name="mlmfile" value="mfinal">

<center><p><img src="/imagenes/nandabot1.jpg" width="440" height="440"
alt="Secuencia" name="secuencia"></p></center>

<center><p><input type="submit" name="Submit"
value="CONECTAR"></p></center>

</form>
</body>
</html>
```

ANEXO J

Archivo .m para las dos primeras páginas del servidor web

```
function retstr = mfinal(instruct, outfile)

retstr = char('');

d = instruct.d;

a = 'MARIA FERNANDA DUEÑAS CORNEJO';
b = 'OSCAR EDUARDO HURTADO GONZALEZ';
c = 'Le dan la Bienvenida a su sitio web';

outstruct.a = a;
outstruct.b = b;
outstruct.c = c;
outstruct.d = d;

templatefile = which('final2.html');
   retstr = htmlrep(outstruct, templatefile);
```

ANEXO K

Código HTML de bienvenida al usuario

```
<html>
<head>
<title>BIENVENIDOS</title>
</head>

<body>

<BR><BR><BR><BR><BR><BR><BR>

<center><font color="red" size="5" face="Arial">
<p>HOLA $d$ </p></font>
<font color="blue" size="5" face="Arial">
<p>$a$</p>
<p>Y</p>
<p>$b$</p></font></center>
<BR><BR>

<span id="theText" style="width:100%">
<h4 align="center"><font color="#3a6ca3"><font size="+5">$c$</font>
 </font><font color="#666666">
 <script>
<!--

var from = 5;                    //the animation start value
var to = 7;                     //the animation end value
var delay = 55;                  //the animation speed
var glowColor = "red";             //the first color
var glowColor2 = "orange";         //the second color
var glowColor3 = "yellow";         //the third color
var glowColor4 = "lime";           //4th color
var glowColor5 = "blue";           //5th color
var glowColor6 = "magenta";        //last color

var i = to;
var j = 0;
textPulseDown();
function textPulseUp()
```

```
{
if (!document.all)
return
if (i < to)
{
theText.style.filter = "Glow(Color=" + glowColor + ", Strength=" + i + ")";
i++;
theTimeout = setTimeout('textPulseUp()',delay);
return 0;
}
if (i = to)
{
theTimeout = setTimeout('textPulseDown()',delay);
return 0;
}
}
function textPulseDown()
{
if (!document.all)
return
if (i > from)
{
theText.style.filter = "Glow(Color=" + glowColor2 + ", Strength=" + i + ")";
i--;
theTimeout = setTimeout('textPulseDown()',delay);
return 0;
}
if (i = from)
{
theTimeout = setTimeout('textPulseUp2()',delay);
return 0;
}
}
function textPulseUp2()
{
if (!document.all)
return
if (i < to)
{
theText.style.filter = "Glow(Color=" + glowColor3 + ", Strength=" + i + ")";
i++;
theTimeout = setTimeout('textPulseUp2()',delay);
return 0;
}
if (i = to)
```

```
{
theTimeout = setTimeout('textPulseDown2()',delay);
return 0;
}
}
function textPulseDown2()
{
if (!document.all)
return
if (i > from)
{
theText.style.filter = "Glow(Color=" + glowColor4 + ", Strength=" + i + ")";
i--;
theTimeout = setTimeout('textPulseDown2()',delay);
return 0;
}
if (i = from)
{
theTimeout = setTimeout('textPulseUp3()',delay);
return 0;
}
}
function textPulseUp3()
{
if (!document.all)
return
if (i < to)
{
theText.style.filter = "Glow(Color=" + glowColor5 + ", Strength=" + i + ")";
i++;
theTimeout = setTimeout('textPulseUp3()',delay);
return 0;
}
if (i = to)
{
theTimeout = setTimeout('textPulseDown3()',delay);
return 0;
}
}
function textPulseDown3()
{
if (!document.all)
return
if (i > from)
{
```

```
theText.style.filter = "Glow(Color=" + glowColor6 + ", Strength=" + i + ")";
i--;
theTimeout = setTimeout('textPulseDown3()',delay);
return 0;
}
if (i = from)
{
theTimeout = setTimeout('textPulseUp()',delay);
return 0;
}
}
//-->
</script></font></h4></span>
<BR>
<center><font color="green" size="5" face="Arial">
<p>SELECCIONE EL TIPO DE COMUNICACION QUE DESEA PARA EL
CONTROL</p>
<BR>
<CENTER>
<FORM NAME="aceButton"><INPUT TYPE="button" VALUE="COMUNICACION
CON MATLAB" onClick="self.location.href=('http://192.168.1.2/final3.html')">
<FORM NAME="aceButton"><INPUT TYPE="button" VALUE="COMUNICACION
CON PHP"
onClick="self.location.href=('http://192.168.1.2/controlf.html')"></FORM>
</CENTER>

</body>
</html>
```

ANEXO L

Código HTML para enviar el dato al servidor web

```
<html>
<head>
<title>CONTROL DEL ROBOT</title>

<script language="JavaScript">

<!-- Comienzo
function CrearArray(n) {
this.length = n;
for (var i = 1; i <= n; i++)
this[i] = i - 1;

this[11] = "A";
this[12] = "B";
this[13] = "C";
this[14] = "D";
this[15] = "E";
this[16] = "F";
return this;
}
hx = new CrearArray(16);

function Hexadecimal(x) {
if (x < 17) x = 16;
var alto = x / 16;
var s = alto+"";
s = s.substring(0, 2);
alto = parseInt(s, 10);
var left = hx[alto + 1];
var low = x - alto * 16;
if (low < 1) low = 1;
s = low + "";
s = s.substring(0, 2);
low = parseInt(s, 10);
var derecha = hx[low + 1];
var cadena = left + "" + derecha;
```

```
return cadena;
}

function ArcoIris(texto) {
texto = texto.substring(0, texto.length);
color_d1 = 255;
mul = color_d1 / texto.length;
for(var i = 0; i < texto.length; i++) {
color_d1 = 255*Math.sin(i / (texto.length / 3));
color_h1 = Hexadecimal(color_d1);
color_d2 = mul * i;
color_h2 = Hexadecimal(color_d2);

k = texto.length;
j = k - i;
if (j < 0) j = 0;
color_d3 = mul * j;
color_h3 = Hexadecimal(color_d3);

document.write("<FONT COLOR=\"#" + color_h3 + color_h1 + color_h2 + "\">" +
texto.substring(i, i + 1) + "</FONT>");
   }
}
// Fin -->
</script>
</head>

<body>

<form action="/cgi-bin/matweb.exe" method="POST">

<input type="hidden" name="mlmfile" value="mdato">

<script>
 nuevaUrl='http://localhost/video.html'
 nuevaWin='_blank'
 nuevoTime=1000
 setTimeout("open(nuevaUrl,nuevaWin)",nuevoTime);
</script>

<CENTER><font size=9><script language="">
<!--
ArcoIris("CONTROL DEL ROBOT");
// -->
</script></font></CENTER>
```

```
<BR><BR><BR><BR><BR>

<B><FONT size=5 color=blue face=arial>
<p>Instrucción: <input type="text" size="3" name="d"></p>
</FONT></B>



<FORM NAME="aceButton"><INPUT TYPE="BUTTON" VALUE="Suba"
onClick="self.location.href=('http://www.tusite.com')"></FORM>

<BR><BR>

<I><FONT size=4 color=red face=arial>
<p>w=Adelante<BR>s=Atras<BR>a=Izquierda<BR>d=Derecha</p>
</FONT></I>

<center><p><input type="submit" name="Submit" value="ENVIAR
DATO"></p></center>

</font>
</body>
</html>
```

ANEXO M

Archivo .m para enviar el dato a través del puerto serie

```
function retstr = mdato(instruct, outfile)

retstr = char('');

fid = serial('COM9');
fopen(fid);
 b = 'Dato enviado';
d = instruct.d;
 fprintf(fid,'%c',d);
fclose(fid);
 outstruct.b = b;
clear fid
 templatefile = which('final4.html');
  retstr = htmlrep(outstruct, templatefile);
```

ANEXO N

Código HTML de la página web de la aplicación con lenguaje PHP

```
<html>
<head>
</head>

<frameset rows="163,*" cols="*" framespacing="0" frameborder="no" border="0" bordercolor="#00FF00">
  <frame src="titulo.html" name="topFrame" scrolling="No" noresize="noresize" id="topFrame" title="TITULO" />
  <frameset rows="*" cols="656,*" framespacing="0" frameborder="no" border="0">
                <frame src="video2.html" name="leftFrame" scrolling="No" noresize="noresize" id="leftFrame" title="leftFrame" />
                <frame src="botones.html" name="mainFrame" id="mainFrame" title="SEPARACION" />
  </frameset>
</frameset>
<noframes><body>
</body>
</noframes></html>
```

ANEXO O

Código PHP del botón delante de la página web de la aplicación con lenguaje PHP

```
<HTML>
<BODY>
<?php
include "php_serial.class.php";

// Let's start the class
$serial = new phpSerial;

// First we must specify the device. This works on both linux and windows (if
// your linux serial device is /dev/ttyS0 for COM1, etc)
$serial->deviceSet("COM5");

$serial->confBaudRate(9600);

// Then we need to open it
$serial->deviceOpen();

// To write into
$serial->sendMessage("w");

// If you want to change the configuration, the device must be closed
$serial->deviceClose();

?>

<META HTTP-EQUIV="REFRESH"
CONTENT="0;URL=http://192.168.1.2/botones.html">
</BODY>
</HTML>
```