

**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA
HARDWARE Y SOFTWARE PARA EL CONTROL DE UN ROBOT MÓVIL
UTILIZANDO EL IPAD**

**GRUPO DE INVESTIGACIÓN
En Control Industrial**

DIEGO RICARDO PÁEZ ARDILA

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA ELECTRÓNICA
FLORIDABLANCA
2012**

**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA
HARDWARE Y SOFTWARE PARA EL CONTROL DE UN ROBOT MÓVIL
UTILIZANDO EL IPAD**

DIEGO RICARDO PÁEZ ARDILA

**TRABAJO DE GRADO PARA OPTAR A TÍTULO DE
INGENIERO ELECTRÓNICO**

Director:

MSc. CLAUDIA LEONOR RUEDA GUZMAN

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
FLORIDABLANCA
2012**

TABLA DE CONTENIDO

| | Pág. |
|---|------|
| 1. INTRODUCCIÓN | 1 |
| 2. MARCO REFERENCIAL..... | 3 |
| 2.1 MARCO TEÓRICO | 3 |
| 2.1.1 Ipad..... | 3 |
| 2.1.2 Xcode..... | 5 |
| 2.1.3 Arduino..... | 5 |
| 2.1.4 Wi-Fi..... | 9 |
| 2.1.5 GPS | 10 |
| 2.1.6 Servidor HTTP Apache 2.0..... | 11 |
| 2.1.7 Administrador de Bases de Datos MySQL..... | 12 |
| 2.1.8 Lenguaje PHP | 13 |
| 2.1.9 Métodos Get y Post..... | 17 |
| 2.1.10 JSON | 17 |
| 2.1.11 ASIHTTPREQUEST | 18 |
| 2.1.12 Raspberry Pi | 18 |
| 2.2 ESTADO DEL ARTE..... | 21 |
| • IPAD COMO ASISTENTE DE ACCESO PERCUTÁNEO AL RIÑÓN USANDO MARCADORES DE NAVEGACIÓN BASADO EN: LA EXPERIENCIA CLÍNICA INICIAL | 21 |
| • EL IPAD EN CARDIOLOGÍA | 23 |
| • AYUDANTE ROBÓTICO DE CASA CARE-O-BOT | 23 |
| • ARQUITECTURA PARA EL MONITOREO REMOTO DE FUNCIONES VITALES EN PACIENTES AMBULATORIOS. | 24 |
| • INTERFAZ PARA EQUIPOS DE MONITOREO MÉDICO (RED DE TELEMEDICINA ANTIOQUIA)..... | 25 |
| 3. ESTRUCTURA DEL PROYECTO..... | 26 |
| 3.1 OBJETIVO GENERAL | 26 |
| 3.2 OBJETIVOS ESPECIFICOS | 26 |
| 4. DISEÑO E IMPLEMENTACIÓN DEL HARDWARE Y SOFTWARE | 27 |
| 4.1 Tipos de configuración del sistema | 30 |

| | | |
|-------|--|----|
| 4.2 | Capa Interna Robot..... | 30 |
| 4.2.1 | Motores | 31 |
| 4.2.2 | Brújula y Sensores Ultrasónicos..... | 34 |
| 4.2.3 | GPS | 39 |
| 4.2.4 | Conexión Wi-Fi..... | 40 |
| 4.2.5 | Software del Arduino Uno..... | 45 |
| 4.3 | Capa de Enlace y Monitoreo | 48 |
| 4.3.1 | Configuración Raspberry Pi..... | 48 |
| 4.3.2 | Instalación Servidor Web Apache 2.0..... | 51 |
| 4.3.3 | Instalación PHP5..... | 52 |
| 4.3.4 | Instalación MySQL | 53 |
| 4.3.5 | Configuración acceso remoto..... | 54 |
| 4.3.6 | Base de datos | 59 |
| 4.3.7 | Software Servidor..... | 64 |
| 4.3.8 | Web Monitoreo..... | 68 |
| 4.4 | Capa de Control..... | 71 |
| 4.4.1 | Primera aplicación..... | 71 |
| 4.4.2 | Diagrama de flujo aplicación..... | 78 |
| 4.4.3 | Control Punto a Punto | 79 |
| 4.4.4 | Utilizando Mapas..... | 80 |
| 4.4.5 | Aplicación con pestañas..... | 82 |
| 4.4.6 | Aplicación Final | 84 |
| 5. | CONCLUSIONES | 90 |
| 6. | BIBLIOGRAFÍA..... | 91 |

LISTAS DE FIGURAS

| | Pág. |
|---|------|
| Figura 1. Capas de abstracción iOS | 4 |
| Figura 2. Interfaz Xcode..... | 5 |
| Figura 3. Arduino Uno..... | 6 |
| Figura 4. Diagrama de bloques Atmel328 | 6 |
| Figura 5. Entorno Desarrollo Arduino..... | 9 |
| Figura 6. Wifly Shield..... | 10 |
| Figura 7. GPS Copernicus II DIP Module..... | 11 |
| Figura 8. Cuota de Mercado Apache | 12 |
| Figura 9. Sintaxis Básica PHP | 14 |
| Figura 10. Variables PHP. | 15 |
| Figura 11. Ejemplo Json | 17 |
| Figura 12. Ejemplo ASIFormDataRequest | 18 |
| Figura 13. Raspberry Pi Modelo B | 19 |
| Figura 14. Pines GPIO Raspberry Pi | 20 |
| Figura 15. La experiencia clínica con el iPad | 22 |
| Figura 16. Care-O-bot entregando una bebida a un adulto mayor | 24 |
| Figura 17. Sistema de telemedicina..... | 24 |
| Figura 18. Arquitectura de la red..... | 25 |
| Figura 19. Diagrama diseño del sistema remoto | 28 |
| Figura 20. Diagrama diseño del sistema punto a punto | 29 |
| Figura 21. Tarjeta de sensores y driver..... | 31 |
| Figura 22. Tarjetas Capa Interna Robot..... | 31 |
| Figura 23. Esquemático Motores | 32 |
| Figura 24. Módulos control adelante, atrás. | 33 |
| Figura 25. Módulos control derecha, izquierda | 34 |
| Figura 26. Brújula | 35 |
| Figura 27. Esquemático brújula y motores..... | 35 |
| Figura 28. Distribución Sensórica | 36 |
| Figura 29. Modulo leer brújula | 37 |
| Figura 30. Diagrama de tiempos SRF08..... | 38 |
| Figura 31. Modulo leer Ultrasónicos..... | 38 |
| Figura 32. Esquemático GPS | 39 |
| Figura 33. GPS en la tarjeta | 39 |
| Figura 34. Modulo GPS | 40 |
| Figura 35. Wifly Shield..... | 41 |
| Figura 36. Configuración cliente | 41 |
| Figura 37 Envío de datos vía Wi-Fi..... | 42 |
| Figura 38 Recepción datos vía Wi-Fi | 43 |

| | |
|---|----|
| Figura 39 Configuración ad-hoc..... | 44 |
| Figura 40 Recepción datos ad-hoc | 45 |
| Figura 41 Diagrama de flujo Arduino Uno | 47 |
| Figura 42 Instalar Raspbian en SD | 49 |
| Figura 43 Raspberry Pi con periféricos | 49 |
| Figura 44 Raspberry Pi Inicio..... | 50 |
| Figura 45 Raspberry Pi súper usuario..... | 51 |
| Figura 46. Código para instalar apache2 | 51 |
| Figura 47. Prueba funcionamiento apache2..... | 52 |
| Figura 48. Instalación PHP5 | 52 |
| Figura 49. Código prueba PHP | 52 |
| Figura 50. Prueba PHP | 53 |
| Figura 51. Código instalación MySql..... | 53 |
| Figura 52. Contraseña MySql | 54 |
| Figura 53. Accediendo a MySql | 54 |
| Figura 54. Configuración IP estática | 55 |
| Figura 55. Ventana inicio sesión no-ip | 56 |
| Figura 56. Creación host..... | 56 |
| Figura 57. Software a descargar de no-ip | 57 |
| Figura 58. Instalación de no-ip..... | 57 |
| Figura 59. Instalación función make..... | 57 |
| Figura 60. Archivo leer primero..... | 58 |
| Figura 61. Código script noip.sh | 58 |
| Figura 62. Permisos de ejecución al script..... | 59 |
| Figura 63. Copiando el script en la carpeta init.d | 59 |
| Figura 64. Ingresar a la carpeta init.d..... | 59 |
| Figura 65. Prioridad de ejecución 18..... | 59 |
| Figura 66. Iniciar el cliente noip.sh..... | 59 |
| Figura 67. Tabla control..... | 60 |
| Figura 68. Tabla ultrasónicos..... | 60 |
| Figura 69. Tabla gps..... | 60 |
| Figura 70. Tabla brújula..... | 61 |
| Figura 71. Iniciar sesión MySQL..... | 61 |
| Figura 72. Crear base de datos | 62 |
| Figura 73. Base de datos prueba..... | 62 |
| Figura 74. Agregar tablas y campos | 63 |
| Figura 75. Inicializar campos | 64 |
| Figura 76. Actualizar base de datos desde el Robot. | 65 |
| Figura 77. Diagrama de Flujo Programa Servidor No. 1..... | 66 |
| Figura 78. Recepción y envió de información al iPad..... | 67 |
| Figura 79. Diagrama de Flujo Programa Servidor No. 2..... | 67 |
| Figura 80. Datos de control..... | 68 |

| | |
|---|----|
| Figura 81. Diagrama de Flujo Programa Servidor No. 3..... | 68 |
| Figura 82. Página monitoreo..... | 69 |
| Figura 83. Diagrama de seguridad..... | 70 |
| Figura 84. Página monitoreo..... | 70 |
| Figura 85. Nuevo proyecto <i>Xcode</i> | 71 |
| Figura 86. Selección plantilla | 72 |
| Figura 87. Nombre aplicación | 72 |
| Figura 88. Agregar librerías | 73 |
| Figura 89. Bandera ARC..... | 74 |
| Figura 90. Agregar Frameworks | 74 |
| Figura 91. Interfaz primera aplicación | 75 |
| Figura 92. Dividir Pantalla y agregar IBAction..... | 76 |
| Figura 93. Agregar IBOutlet | 77 |
| Figura 94. Agregar librerías | 77 |
| Figura 95. Acción botón | 77 |
| Figura 96. Acción recepción | 78 |
| Figura 97. Resultado aplicación..... | 78 |
| Figura 98. Diagrama aplicación | 79 |
| Figura 99. Enviar dato directo..... | 80 |
| Figura 100. Ventana Mapas..... | 81 |
| Figura 101. Actualizar Mapa | 81 |
| Figura 102. Tipo de mapa..... | 82 |
| Figura 103. Selección Plantilla Tabbed..... | 83 |
| Figura 104. Selección Storyboards | 83 |
| Figura 105. Storyboard | 84 |
| Figura 106. Icono aplicación | 86 |
| Figura 107. Diseño aplicación..... | 86 |
| Figura 108. Barra de ventanas..... | 87 |
| Figura 109. Ventana control..... | 87 |
| Figura 110. Ventana GPS..... | 88 |
| Figura 111. Ventana sensores | 88 |
| Figura 112. Ventana control local | 89 |
| Figura 113. Configuración botón dos acciones | 89 |

LISTAS DE TABLAS

| | Pág. |
|--|------|
| Tabla 1. Comparación iPad | 3 |
| Tabla 2. Cuatro generaciones de Wi-Fi..... | 10 |
| Tabla 3. Operadores lógicos PHP..... | 16 |
| Tabla 4. Características Raspberry Pi..... | 19 |
| Tabla 5. Dimensiones recomendadas..... | 85 |

AGRADECIMIENTOS

A las personas que con sus ideas y apoyo incondicional aportaron al desarrollo de este proyecto.

A mi Directora Claudia Rueda que con su gran apoyo y respaldo me ha guiado de forma técnica y aporta grandes consejos para lo que viene después de finalizar el proceso universitario.

Finalmente quiero agradecer a mi familia que ha hecho posible que pueda cumplir mi sueño, llegar a ser ingeniero electrónico.

ABSTRACT

TITLE: Design and implementation of a hardware and software platform for control of a mobile robot using the iPad.

AUTOR: Diego Ricardo Páez Ardila

FACULTY: Faculty of Electrical Engineering

DIRECTOR: Msc. Claudia Leonor Rueda Guzmán

ABSTRACT

This Project involves the development of a hardware and software platform, which allows control and supervision of a scale mobile robot, using an iPad Tablet. This work takes part in a macro research, which attempt to bring elderly people into robotics technologies, thanks to the easy handling provided by the current mobile communication devices. Furthermore, the system includes a communication platform built in a LAMP server on a card Raspberry Pi that allows data transfer between a mobile robot and a computer or tablet. Likewise, the robot will be equipped with GPS, ultrasonic sensors, compass, etc. These will be monitored from the tablet, permitting movement control thereof. In addition, it concedes the possibility to supervise the mobile status from a remote computer through the internet, storing the information in a data base for posterior decision making.

Keywords: Mobil Robot, iPad, Apps , Servidor LAMP, Data Bases, Raspberry pi..

V° B° DIRECTOR OF GRADUATE WORK

RESUMEN GENERAL DEL TRABAJO DE GRADO

TITULO: Diseño e implementación de una plataforma hardware y software para el control de un robot móvil utilizando el iPad.

AUTOR: Diego Ricardo Páez Ardila

FACULTAD: Facultad de Ingeniería Electrónica

DIRECTOR: Msc Claudia Leonor Rueda Guzmán

RESUMEN

En este proyecto se desarrollo de una plataforma hardware y software que permite el control y supervisión de un robot móvil a escala utilizando una Tablet iPad, este trabajo está enmarcado en una investigación macro para acercar a las personas de tercera edad con las tecnologías robóticas, gracias a la facilidad de manejo que presentan los dispositivos de computación móvil. El sistema cuenta con una plataforma de comunicación en base a un servidor LAMP , construido en una Raspberry pi que permite la transferencia de información entre el robot móvil el Ipad y un computador. El robot está dotado con sensores de posicionamiento global (GPS), ultrasónicos, brújula etc. que son supervisados desde el iPad, permitiendo además el control de movimiento del mismo. Se cuenta además con la posibilidad de realizar supervisión del estado del móvil vía web, almacenando la información en una base de datos para la toma de decisiones.

Palabras clave: Robot móvil, iPad, Servidor LAMP, control remoto, base de datos, Raspberry pi.

V° B° DIRECTOR TRABAJO DE GRADO

1. INTRODUCCIÓN

En la actualidad la tecnología se ha convertido en una herramienta facilitadora que atiende las necesidades humanas en diferentes ámbitos de la vida, lo cual se evidencia en áreas como la educación, transporte, comunicaciones entre otras. Estas necesidades son cambiantes, por lo tanto la tecnología debe evolucionar al mismo ritmo a través de procesos de investigación desarrollo e innovación.

El iPad es un dispositivo que causó una revolución en el mercado por su facilidad de uso, tiene una pantalla táctil de 9.7 pulgadas lo cual es una gran ventaja para personas que sufren de poca visibilidad y permite una interacción de usuario-máquina muy amigable. Otra razón por la que el iPad causa tanto impacto, es por la variedad de aplicaciones que hay disponibles para su sistema, un desarrollador con una buena idea puede mejorar la vida de muchas personas haciendo una aplicación que cause impacto sobre ellas.

La tecnología forma parte activa en el desarrollo de la medicina en la actualidad y una de las ramas de estudio es la telemedicina donde se puede monitorear el estado de personas de forma remota aportando ventajas a los usuarios de estos sistemas. La asistencia a personas de la tercera edad consiste en brindar un monitoreo remoto que permita a los médicos visualizar en cualquier momento el estado del paciente, recibir mensajes de alerta y llevar un registro del mismo según la afección o dificultad que este padezca.

En este documento se describe el proceso para llevar a cabo el desarrollo de una plataforma hardware y software para el control de un robot móvil a escala, el cual se encuentra dotado con sensores ultrasónicos, brújula, sistema de posicionamiento global (GPS) y transmisión de datos vía Wi-Fi; este es la base para el proyecto de investigación DOCTOR ROBOT en el cual se desarrollará un robot para la asistencia médica a personas de tercera edad de manera remota, con miras a mejorar la calidad de vida de los mismos.

En el primer capítulo se encuentra una descripción detallada de las herramientas de hardware y software que se utilizarán en el proceso de desarrollo del proyecto, además se muestra el estado del arte donde se resalta la ventaja de la implementación de dispositivos móviles en la asistencia personas mayores, finalmente en el primer capítulo se tiene la estructura del proyecto con el planteamiento, objetivo general y objetivos específicos del proyecto.

En el segundo capítulo se realiza el diseño e implementación del hardware y software y se muestra con detalle el desarrollo de las tres capas que componen el sistema, la capa interna robot, capa de enlace y monitoreo y la capa de control. En la primera se explica de forma detallada el hardware y software que posee el robot para su funcionamiento, en la segunda capa se muestran las herramientas utilizadas

para realizar el enlace entre el iPad y el robot y como se visualiza la información de forma remota, finalmente en la tercera capa se explica de forma clara como llevar a cabo el desarrollo de una aplicación para el iPad que permita realizar el control del robot con una interfaz amigable con el usuario.

Finalmente se encuentran las conclusiones resultado del proyecto de investigación..

2. MARCO REFERENCIAL

2.1 MARCO TEÓRICO

2.1.1 iPad

El iPad¹ es un dispositivo tipo Tablet desarrollado por Apple Inc. Este llegó al mercado en el 2010 con su primera versión el iPad 1. Ha ido evolucionando cada año hasta la actualidad, siendo una tecnología que cambia constantemente. En la tabla 1 se puede observar una comparación entre los tres dispositivos que han salido a la venta hasta el momento, cada uno más potente que el anterior para cumplir con las exigencias de los usuarios, lo que se ve reflejado en el aumento de la velocidad de procesamiento y en la calidad de la pantalla, principalmente.²

Tabla 1. Comparación iPad

| | Nuevo iPad | IPad 2 | IPad 1 |
|-----------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Dimensiones | 24.12x18.57x0.94 cm | 24.12x18.57x0.88 cm | 24.12x18.27x1.34cm |
| Pantalla | Retina 9.7"IPS | 9.7"IPS | 9.7"IPS |
| Resolución | 2048X1536 | 1024X768 | 1024X768 |
| Procesador | A5 (Doble Núcleo) | A5 (Doble Núcleo) | A4 |
| GPU | SGXS43MP4 (Cuatro Núcleos) | PowerVR (Doble Núcleo) | PowerVR |
| Almacenamiento | 16,32,64 GB | 16,32,64 GB | 16,32,64 GB |
| Cámara | Trasera: iSight: 5 Mpx, Frontal VGA | Trasera: VGA, Frontal VGA | NO INCLUIDA |
| Wifi | 802.11 a/b/g/n | 802.11 a/b/g/n | 802.11 a/b/g/n |
| Bluetooth | 4.0 | 2.1 + EdR | 802.11 a/b/g/n |
| GPS | Si(WiFi+4G) | Si(WiFi+3G) | Si(WiFi+3G) |
| Acelerómetro | Si | Si | Si |
| Giroscopio | Si(3 ejes) | Si(3 ejes) | NO INCLUIDO |
| Peso | 652 g. (WIFI) - 662 g. (Wi-Fi + 4G) | 601 g. (WIFI) - 613 g. (Wi-Fi + 3G) | 680 g. (WIFI) - 730 g. (Wi-Fi + 3G) |
| Autonomía | Hasta 10 horas de navegación | Hasta 10 horas de navegación | Hasta 10 horas de navegación |

¹ <http://www.apple.com/es/iPad/>, 28/06/2012, all.

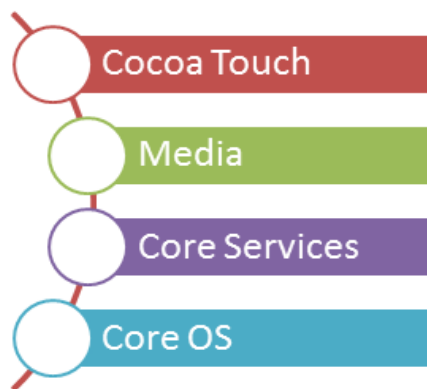
La característica más importante del iPad es la practicidad que esta ofrece a sus usuarios, la tienda de Apple inc. Cuenta con 200.000 aplicaciones que han sido diseñadas específicamente para el iPad, de esta forma se aprovecha la pantalla de 9.7 pulgadas, la tecnología Multi-Táctil, el acelerómetro, el giroscopio entre otras prestaciones que incluye.

El iPad cuenta con una interfaz intuitiva y diseñada de tal forma que el usuario puede utilizarla sin necesidad de entrenamiento, ya que su sistema operativo ha sido creado para ser usado con movimientos muy comunes de los dedos, esto se logra debido a que el hardware y el software son hechos en conjunto y con características que se ajustan entre sí, la marca Apple se encarga de la fabricación tanto de los componentes del iPad como el sistema operativo.

El sistema operativo del iPad es iOS, diseñado por Apple inicialmente para el iPhone y conocido anteriormente como iPhone OS, este sistema operativo ha sido desarrollado pensando en el uso de la pantalla táctil, por esta razón posee un alto desempeño. iOS se deriva de Mac OS X, que a su vez estaba basado en Unix, haciendo de él un sistema operativo seguro y robusto.

Como se puede observar en la figura 1 iOS cuenta con cuatro capas de abstracción: las dos primeras (Core OS, Core Services) proveen los servicios fundamentales del sistema, seguida de estas se encuentra la capa multimedia y finalmente la capa táctil (Cocoa Touch).

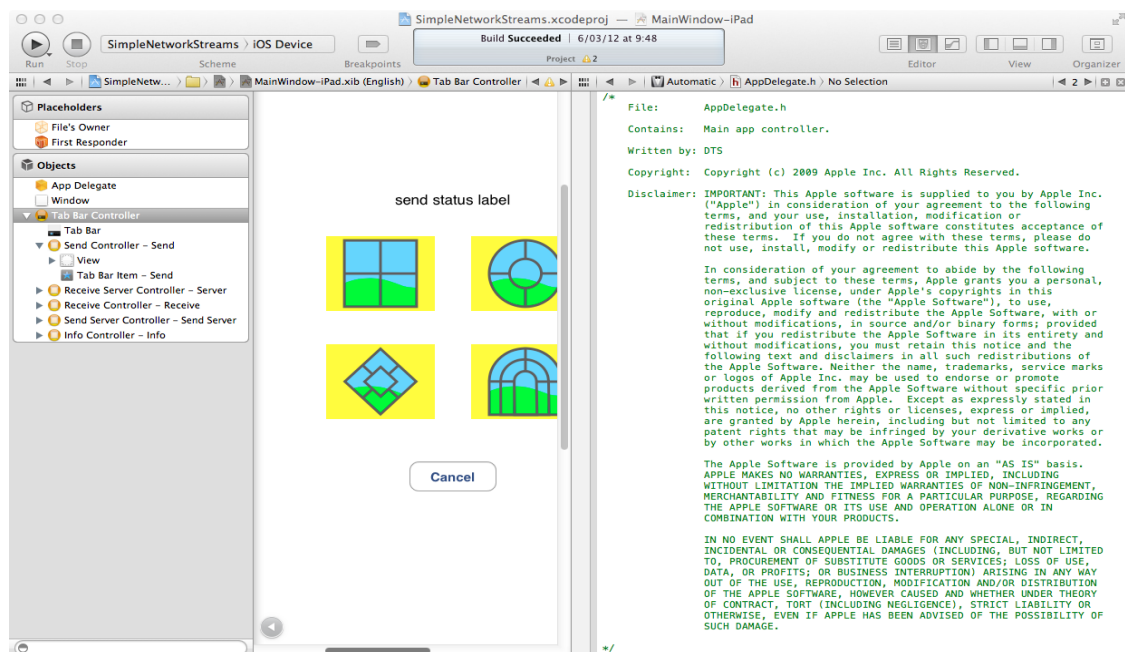
Figura 1. Capas de abstracción iOS



2.1.2 Xcode

Entorno de desarrollo que provee todo lo necesario para la realización de aplicaciones para dispositivos de la marca Apple Inc. Xcode posee una gran variedad de herramientas como son el debugger, simulador y una herramienta que permite verificar el uso adecuado de las propiedades del hardware que deben hacer las aplicaciones. Xcode es de distribución gratuita no siendo software libre ya que para acceder a todas las herramientas es necesario adquirir una licencia de desarrollador. En la figura 2 se observa la interfaz gráfica del Xcode.

Figura 2. Interfaz Xcode



2.1.3 Arduino

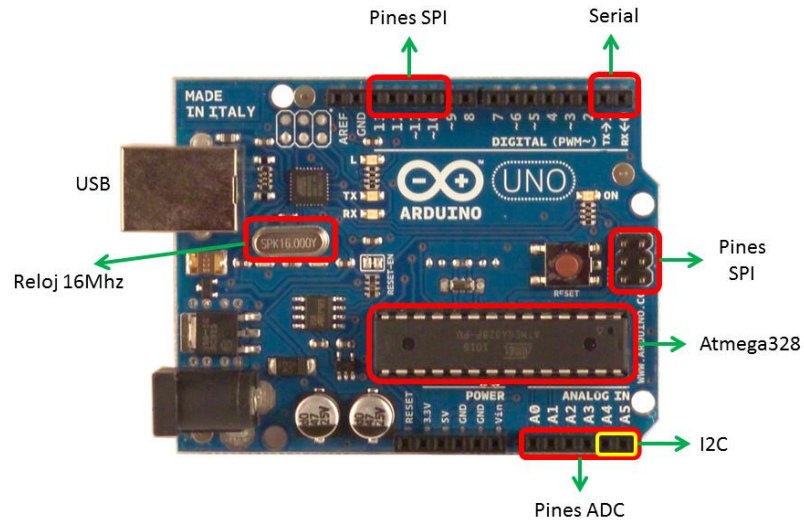
“Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar”³, existen diferentes modelos de Arduino como son el: Arduino Mega, Uno, Leonardo, LilyPad, Fio, Pro, entre múltiples opciones que se pueden encontrar en el mercado, al ser una plataforma de desarrollo libre cada persona puede crear su propia tarjeta.

El Arduino Uno (ver figura 3) está basado en un microcontrolador Atmel Atmega328 cuenta con un total de 20 entradas digitales configurables como I/O, Conversor

³ <http://www.arduino.cc/es>, 18/07/2012, all

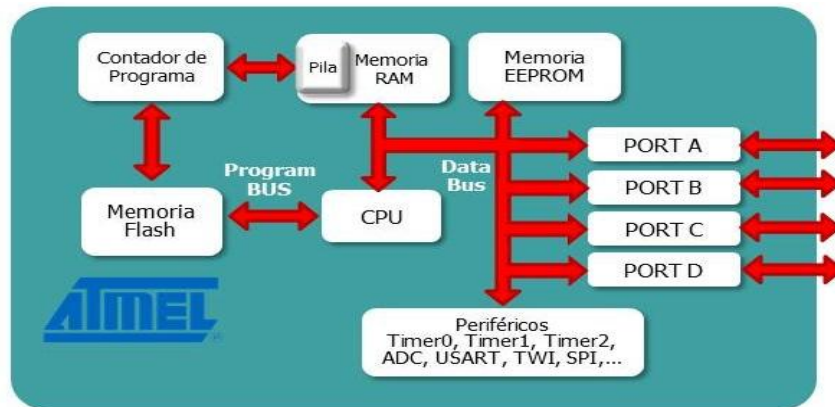
Análogo Digital de 10-bit 6 canales, un puerto de comunicación SPI, un puerto de comunicación I2C, un puerto de comunicación serial, reloj de 16 Mhz, estas prestaciones hacen del Arduino Uno una plataforma capaz de interconectarse con múltiples dispositivos.

Figura 3. Arduino Uno



El microcontrolador Atmel328 (ver figura 4) tiene una arquitectura de tipo Harvard, memoria flash de programa de 32 KB y 2 KB de memoria de datos (SRAM), memoria EEPROM de 1KB. Tiene cuatro puertos configurables como entradas o salidas para realizar la conexión con diferentes dispositivos electrónicos.

Figura 4. Diagrama de bloques Atmel328⁴



El Arduino se programa utilizando un lenguaje propio basado en Processing⁵, el cual es un lenguaje de programación y entorno de desarrollo de código abierto que se encuentra basado en Java, es usado para la producción de proyectos multimedia.

⁴ <http://www.atmel.com/Images/doc8161.pdf>, 18/07/2012, all

⁵ <http://www.processing.org/>, 18/07/2012, all

Se pueden utilizar diferentes lenguajes para programar el Arduino, esto es posible debido a que se comunica mediante la transmisión de datos en formato serie lo cual es soportado por lenguajes como Java, Flash, Python, C, Matlab entre otros.

En la figura 5 se ve el entorno de desarrollo del Arduino, el cual es multiplataforma y no necesita ningún hardware intermedio a la hora de programar el microcontrolador, la estructura básica de un programa está dividida en tres partes, en la primera se realiza la declaración de constantes y librerías, en la segunda se configuran los pines como I/O y se configuran librerías y en la tercera se realiza el programa que se va a ejecutar de forma infinita.

El programa incluye librerías y ejemplos para que el usuario conozca el lenguaje y desarrolle programas. Entre las librerías más importantes se encuentran disponibles: Serial, Ethernet, EEPROM, SPI, I2C, SD, LiquidCrystal, Firmata, Servo y SoftwareSerial.

Al encontrarse el Arduino basado en C tienen la capacidad de soportar las funciones del mismo estándar, se hará un resumen con las estructuras principales del lenguaje Arduino.

Sintaxis Básica:

- Delimitadores: ;, {}
- Comentarios: //, /* */
- Cabeceras: #define, #include
- Operadores aritméticos: +, -, *, /, %
- Asignación: =
- Operadores de comparación: ==, !=, <, >, <=, >=
- Operadores Booleanos: &&, ||, !
- Operadores de bits: &, |, ^, ~, <<, >>
- Operadores compuestos:
 - Incremento/decremento de variables: ++, --
 - Asignación y operación: +=, -=, *=, /=, &=, |=

Estructuras de control:

- Condicionales: if, if...else, switch case
- Bucles: for, while, do... while
- Bifurcaciones y saltos: break, continue, return, goto

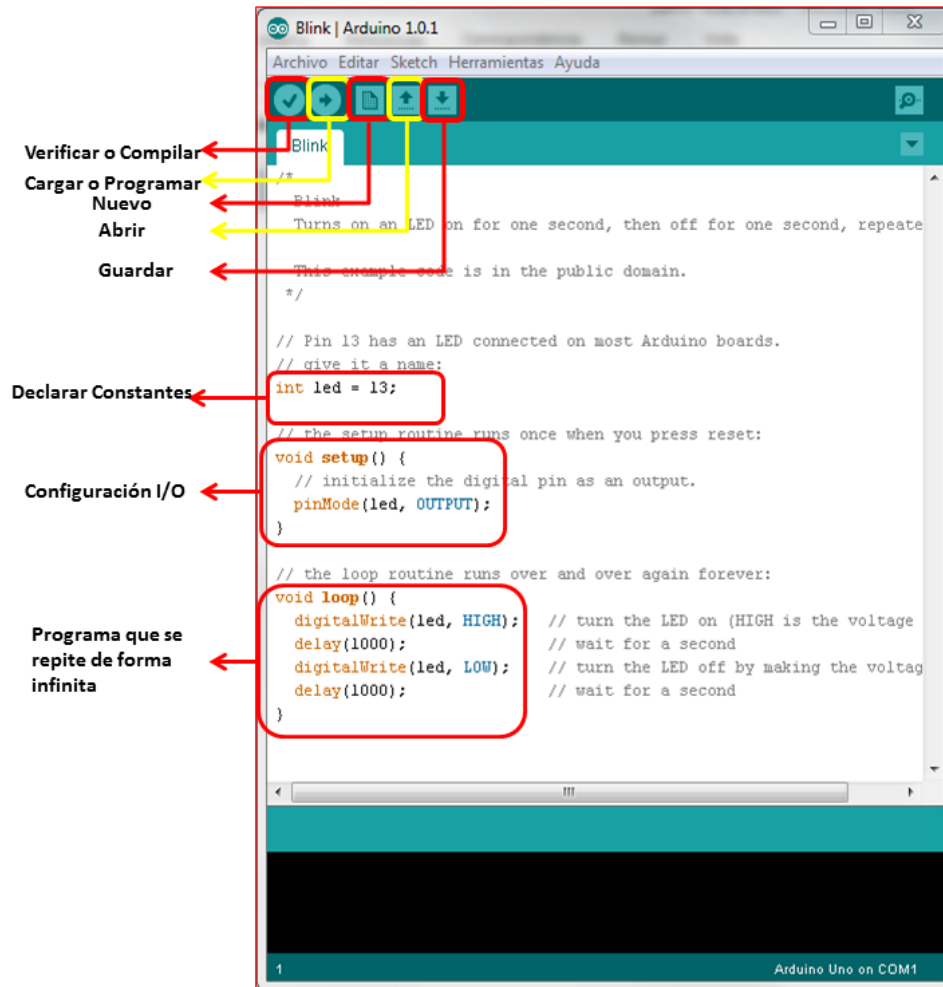
Tipos de datos:

- void, boolean, char, unsigned char, byte, int, unsigned int, word, long, unsigned long, float, double, string, array.

Funciones Básicas:

- I/O Digital
 - pinMode(pin, modo)
 - digitalWrite(pin, valor)
 - int digitalRead(pin)
- I/O Analógica:
 - int analogRead(pin)
 - analogWrite(pin, valor)
- Tiempo:
 - unsigned long millis()
 - unsigned long micros()
 - delay(ms)
 - delayMicroseconds(microsegundos)
- Matemáticas:
 - min(x, y), max(x, y), abs(x), constrain(x, a, b), map(valor, fromLow, fromHigh, toLow, toHigh), pow(base, exponente), sqrt(x)
- Trigonometría:
 - sin(rad), cos(rad), tan(rad)
- Puerto Serie:
 - begin(), available(), read(), flush(), print(), println(), write()

Figura 5. Entorno Desarrollo Arduino



2.1.4 Wi-Fi

El Wi-Fi permite realizar una conexión inalámbrica de dispositivos electrónicos a Internet. La tecnología Wi-Fi es identificada como 802.11, existen diferentes generaciones que se identifican por su desempeño respecto al ancho de banda y la frecuencia. En la tabla 2 se observan las cuatro generaciones.

Tabla 2. Cuatro generaciones de Wi-Fi⁶

| Wi-Fi | Frecuencia | Máxima tasa de transferencia |
|----------------|--|------------------------------|
| 802.11a | 5 GHz | 54 Mbps |
| 802.11b | 2.4 GHz | 11 Mbps |
| 802.11g | 2.4 GHz | 54 Mbps |
| 802.11n | 2.4 GHz, 5 GHz, 2.4 o 5 GHz o 2.4 y 5 GHz. | 450 Mbps |

La WiFly Shield (ver figura 6) equipa al Arduino Uno con la capacidad de conectarse a redes inalámbricas 802.11b/g. La tarjeta está dotada con el chip RN-131C⁷ el cual a través de una comunicación SPI con el microcontrolador permite el envío de datos HTML, de esta manera es posible enviar información a un servidor web para luego administrarla.

Figura 6. Wifly Shield⁸



2.1.5 GPS

El sistema de posicionamiento global (GPS), como su nombre lo dice entrega las coordenadas de la posición en la cual se encuentra el dispositivo, este sistema se utiliza comúnmente para determinar las rutas de los barcos, carros y cualquier persona que se encuentre en movimiento y quiera trasladarse de un lugar a otro.

El GPS envía un mensaje codificado Asociación Nacional de Electrónica Marina (NMEA), el cual se transmite por puerto serial al computador, e implementando el software que incluye el fabricante, se adquieren las coordenadas de la posición en la que se encuentra ubicado.

⁶ <http://www.wi-fi.org/discover-and-learn>, 05/09/2012, all.

⁷ <http://www.sparkfun.com/datasheets/Wireless/WiFi/rn-131-ds.pdf>, 05/09/2012, all.

⁸ <https://www.sparkfun.com/products/9954>, 05/09/2012, all.

En la figura 7 se muestra el Copernicus II DIP Module. Es un módulo GPS de la empresa Trimble, posee unas características destacables, un receptor con 12 canales de operación para una rápida adquisición de satélites, una entrada SubMiniature versión A (SMA) el cual es un tipo de conector roscado para cable coaxial utilizado en microondas, útil hasta una frecuencia de 33 GHz, este es utilizado para la conexión de una antena con mayor ganancia y así lograr una conexión más rápida también incluye dos puertos seriales, por medio de estos se realiza la comunicación con el microcontrolador, tiene un consumo bajo de corriente 30.7mA, es relevante para un sistema que suple su energía con baterías.

Figura 7. GPS Copernicus II DIP Module⁹



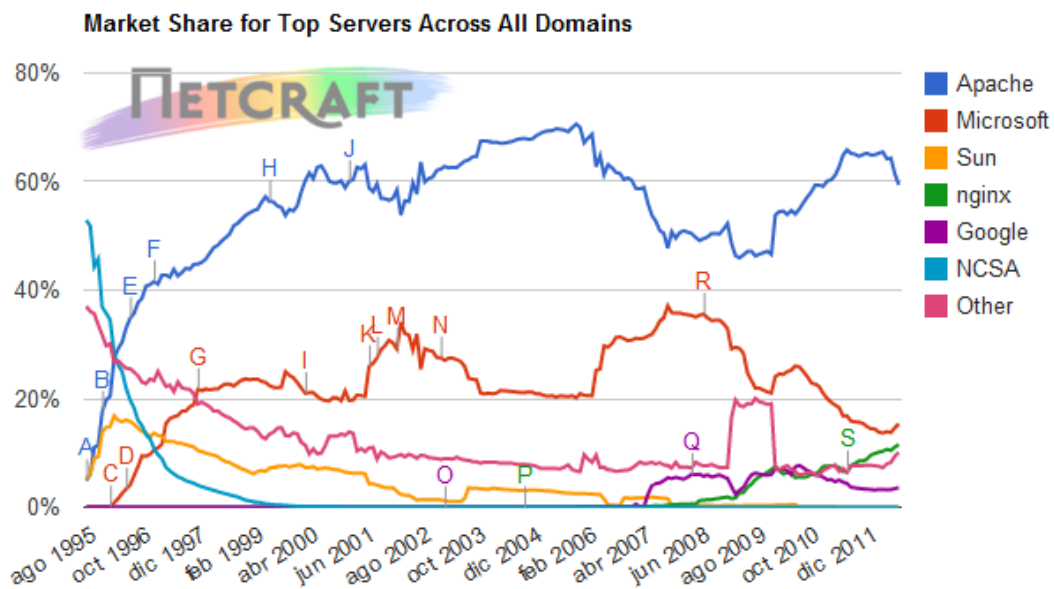
2.1.6 Servidor HTTP Apache 2.0

El servidor Web Apache es un servidor HTTP de código abierto, ha sido desarrollado por el grupo Apache Software Foundation¹⁰ dentro del proyecto HTTP Server. Este servidor ha sido creado sobre Unix y Windows y se ha convertido en uno de los servidores líderes en el mercado como se ve en la figura 8 la cuota de mercado para todos los dominios es liderada por Apache, para lograr esto sus creadores han desarrollado un servidor con los requerimientos de seguridad, eficiencia, extensibilidad y estandarización.

⁹ <http://www.sparkfun.com/products/11067>, 28/06/2012, all.

¹⁰ <http://httpd.apache.org/>, 27/08/2012, all.

Figura 8. Cuota de Mercado Apache¹¹



El servidor Web Apache se basa en una arquitectura modular que permite extender las funcionalidades y características, Apache está compuesto por módulos básicos, módulos multiproceso y módulos externos. En los módulos básicos se realizan las funciones básicas del servidor, en los módulos multiprocesos la gestión de las peticiones y los módulos externos¹² son desarrollados por terceros y añaden funciones adicionales al servidor.

Apache tiene la opción de configurar la distribución de procesos que atenderán a las peticiones realizadas por los clientes. Esto permite optimizar el servidor segunda la función que se le vaya a dar.

2.1.7 Administrador de Bases de Datos MySQL

MySQL¹³ es un sistema de administración de bases de datos relacionales rápido, solido y flexible. Es ideal para crear bases de datos con acceso desde páginas web dinámicas, para cualquier solución que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas. Ofrece varias ventajas respecto a otros gestores de bases de datos ya que es de carácter libre por lo tanto se puede utilizar y además modificarlo para adecuarlo a necesidades específicas. El programa

¹¹<http://news.netcraft.com/archives/2012/08/02/august-2012-web-server-survey.html>, 02/09/2012, all.

¹² <https://modules.apache.org/>, 01/09/2012, all.

¹³ <http://www.mysql.com/>, 27/08/2012, all.

se ha desarrollado en C y C++, lo que facilita su integración en aplicaciones desarrolladas en el mismo lenguaje.

MySQL utiliza el lenguaje SQL (Lenguaje de Consulta Estructurado) que es el lenguaje de consulta estandarizado para acceder a bases de datos relacionales. Soporta la sintaxis estándar del lenguaje SQL para la realización de consultas de manipulación, creación y de selección de datos.

Es un sistema cliente/servidor, permitiendo trabajar como servidor multiusuario y de subprocesamiento múltiple, es decir, cada vez que se estable una conexión con el servidor, el programa servidor crea un subproceso para atender la solicitud del cliente, controlando el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso solo a usuarios autorizados.

Es portable, ya que se puede llevar a cualquier plataforma informática. MySQL está disponible para las distribuciones más usadas de Linux, Mac OS X, UNIX y Microsoft Windows¹⁴.

2.1.8 Lenguaje PHP

PHP¹⁵ es un lenguaje interpretado del lado del servidor que se caracteriza por su potencia, versatilidad, robustez y modularidad. Los programas escritos en PHP son embebidos directamente en el código HTML y ejecutados por el servidor web a través de un intérprete antes de transferir al cliente que lo ha solicitado un resultado en forma de código HTML puro.

Es destacado especialmente por la facilidad para la conectividad con sistemas gestores de bases de datos a través de un gran número de funciones especializadas. PHP fue diseñado inicialmente para realizar contadores y libros de visita de páginas, en la actualidad PHP permite realizar una multitud de tareas útiles para el desarrollo web.

- Funciones de correo electrónico que pueden ser utilizadas para programar completos sistemas de correo electrónico vía web.
- Funciones de administración y gestión de bases de datos específicas para la mayoría de gestores comerciales.
- Funciones de gestión de directorios y ficheros, incluso para la transferencia mediante FTP.

¹⁴ COBO ORTEGA, Angel., et al. El sistema gestor de base de datos MySql. En: Php y MySql: tecnología para el desarrollo de aplicaciones web. 1 ed. España: Días de Santos, 2005. P. 339-340.

¹⁵ <http://www.php.net/>, 27/08/2012, all.

- Funciones de tratamiento de imágenes y librerías de funciones graficas¹⁶.

En la figura 9 se ve una función PHP incrustada en HTML, el código está en medio de etiquetas de comienzo y final especiales `<?php` y `?>` que permiten entrar y salir del modo PHP.

A continuación se muestra un resumen de las estructuras que componen el lenguaje PHP.

Figura 9. Sintaxis Básica PHP

```

<html>
  <head>
    <title>Ejemplo PHP</title>
  </head>
  <body>
    <?php echo '<p>Hola Mundo</p>';
  ?>
  </body>
</html></html>

```

PHP soporta ocho tipos de datos primitivos:

- Escalares:
 - Boolean: False, True-
 - Integer: número del conjunto $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$.
 - Float: "flotantes", "dobles" o "números reales"
 - String: serie de caracteres donde un carácter es lo mismo que un byte.
- Compuestos:
 - Array: Matriz real, Vector, Tabla asociativa.
 - Object: Manejo de Clases y Objetos.
- Especiales:
 - Resource: variable especial, que contiene una referencia a un recurso externo.
 - NULL: representa una variable sin valor.

¹⁶ COBO ORTEGA, Angel., et al. Introducción a PHP MySQL. En: Php y MySql: tecnología para el desarrollo de aplicaciones web. 1 ed. España: Días de Santos, 2005. P. 99-100.

- Callable: Llamada de retorno.

En PHP las variables se representan con un signo dólar seguido del nombre de la variable, es sensible al uso de mayúsculas o minúsculas. En la figura 10 se puede apreciar la declaración de diferentes variables.

Figura 10. Variables PHP.

```
<?php
$var = 'Diego';
$Var = 'Ricardo';
echo "$var, $Var";      // imprime "Diego, Ricardo"

$4site = 'aun no';     // inválido; comienza con un número
$_4site = 'aun no';   // válido; comienza con un carácter
de subrayado
$täyte = 'mansikka';   // válido; 'ä' es ASCII (Extendido)
228
?>
```

Los operadores que se utilizan para gestionar los datos primitivos son: aritméticos, de asignación, bit a bit, de comparación, de control de errores, operadores de ejecución, de incremento/decremento, operadores lógicos, operadores para strings, arrays y de tipo. En la tabla 3 se muestra un resumen de los operadores y el tipo al cual pertenecen.

Todo script PHP está construido en base a las estructuras de control. Una estructura de control puede ser una asignación, una llamada de función, un ciclo, una sentencia condicional o incluso una sentencia que no hace nada. Las estructuras de control utilizadas en PHP son las siguientes:

- If
- Else
- Elseif/else if
- While
- Do-while
- For
- Foreach

- Break
- Continue
- Switch
- Declare
- Return
- Require
- Include
- Require_once
- Include_once
- Goto.

Tabla 3. Operadores lógicos PHP

| Operadores | Información adicional |
|--|-------------------------------|
| clone new | clone and new |
| [| array() |
| ++ -- ~ (int) (float) (string) (array) (object) (bool) @ | tipos e incremento/decremento |
| instanceof | tipos |
| ! | lógico |
| * / % | aritmética |
| + - . | aritmética y string |
| << >> | bit a bit |
| < <= > >= | comparación |
| == != === !== <> | comparación |
| & | bit a bit y referencias |
| ^ | bit a bit |
| | bit a bit |
| && | lógico |
| | lógico |
| ? : | ternario |
| = += -= *= /= .= %= &= = ^= <<= >>= ==> | asignación |
| and | lógico |
| xor | lógico |
| or | lógico |
| , | muchos usos |

PHP es un lenguaje pensando para el principiante, pero a su vez, ofrece características avanzadas para programadores profesionales, además aunque ha sido creado para el desarrollo de páginas web también es posible utilizarlo para

realizar diferentes proyectos como puede ser la interacción por puerto serial con dispositivos externos¹⁷.

2.1.9 Métodos Get y Post

Get y Post son métodos utilizados para enviar datos al servidor, el método más común es Get en el cual se solicita la información de una url específica, un ejemplo de esto es `http://omax.sytes.net/?grados=125`, en este ejemplo se recibe la variable `grados=125`, para almacenarla se utiliza PHP creando una variable de la siguiente forma `$grados=$_GET['grados']`.

Los datos del método Post son enviados por el cliente en el cuerpo de la petición, a diferencia del método Get la información no se encuentra a la vista del usuario. Para almacenar la información recibida utilizando PHP se crea una variable de la siguiente forma `$grados=$_POST['grados']` así se almacena en una variable local.

2.1.10 JSON

Json¹⁸ es un formato ligero de intercambio de datos alternativo a XML¹⁹, se destaca por la simplicidad de uso. El fin de estas dos tecnologías es compartir los datos entre diferentes sistemas de forma segura. Json tiene funciones nativas en PHP que permiten crear objetos con facilidad, las dos funciones son `json_encode` y `json_decode`²⁰. Un ejemplo de un objeto Json se ve en la figura 11, donde `$result` es un vector compuesto por las variables, `s1,s2,s3,s4,s5`, luego el vector se codifica utilizando la función `json_encode`.

Figura 11. Ejemplo Json

```
$result = array("s1" => $s1,  
              "s2" => $s2,  
              "s3" => $s3,  
              "s4" => $s4,  
              "s5" => $s5,  
              );  
sendResponse(200, json_encode($result));
```

¹⁷ <http://www.php.net/manual/es/index.php>, 27/08/2012, all.

¹⁸ <http://www.json.org/json-es.html>, 27/08/2012, all.

¹⁹ <http://www.w3.org/XML/>, 28/08/2012, all.

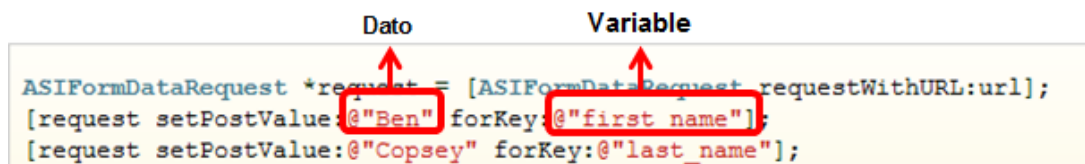
²⁰ <http://www.php.net/manual/es/ref.json.php>, 28/08/2012, all.

2.1.11 ASIHTTPREQUEST

Es una herramienta que funciona en torno a la API CFNetwork²¹, su función es hacer que la comunicación con los servidores sea mucho más sencilla. Es escrita en lenguaje c y funciona para Mac OS X y IOS. Es adecuada para realizar peticiones básicas HTTP e interactuar con servicios usando Get, Post, Put, Delete.

La subclase ASIFORMDataRequest se utiliza para enviar datos utilizando el método POST como se muestra en la figura 12, donde “Ben” es el dato que se quiere enviar y “first_name” es la variable que lo almacena, de esta manera se envían los datos al servidor.

Figura 12. Ejemplo ASIFORMDataRequest



The diagram shows a code snippet for ASIFORMDataRequest. The code is as follows:

```
ASIFORMDataRequest *request = [ASIFORMDataRequest requestWithURL:url];  
[request setPostValue:@"Ben" forKey:@"first_name"];  
[request setPostValue:@"Copsey" forKey:@"last_name"];
```

Annotations in the image:

- An arrow labeled "Dato" points to the string value `@"Ben"`.
- An arrow labeled "Variable" points to the key `@"first_name"`.

2.1.12 Raspberry Pi

Es un sistema de desarrollo con arquitectura ARM (Advanced RISC Machine) y GNU/Linux de muy bajo costo y pequeño tamaño (ver Figura 13), fue desarrollado por la fundación Raspberry Pi²² en la Universidad de Cambridge, Inglaterra con el objetivo de llevar equipos de bajo costo a los jóvenes de todo el mundo que se inician en la programación.

El SoC (System on a Chip) es un dispositivo que integra los diferentes componentes de un sistema, como son el controlador de memoria RAM, tarjeta gráfica, memoria, buses, interfaces externos y comunicación. El Raspberry Pi utiliza un Broadcom BCM2835, que contiene un ARM1176JZFS a 700Mhz GPU Videocore 4, que es capaz de reproducir video en alta definición 1080p, puede llegar a ejecutar 24 GFLOPS de cálculos de propósito general dando así la potencia requerida para llevar a cabo múltiples aplicaciones, en la tabla 3 se ve las características de hardware del Raspberry Pi de forma detallada.

Las distribuciones de Linux soportadas oficialmente por la fundación Raspberry Pi son las siguientes:

²¹<https://developer.apple.com/library/mac/#documentation/Networking/Conceptual/CFNetwork/Introduction/Introduction.html>, 28/07/2012, all.

²² <http://www.raspberrypi.org/>, 27/08/2012, all.

Raspbian: Es la distribución recomendada por el fabricante para principiantes, está basada en Debian²³, cuenta con un entorno de escritorio LXDE²⁴, navegador web y herramientas de desarrollo pre instaladas en el equipo.

Arch Linux ARM: Esta distribución es basada en Arch Linux²⁵, ofrece la posibilidad de configurar el sistema operativo desde cero, dando control total al usuario para incluir las herramientas que desee en el equipo, esta distribución no es recomendada para principiantes.

Qt on Pi: Es una plataforma embebida que incluye un SDK, optimizada para el desarrollo y ejecución de aplicación Qt 5²⁶.

El Raspberry Pi tiene 26 pines de propósito general GPIO (ver figura 14), estos pines permiten que el equipo se conecte diferentes dispositivos electrónicos utilizando puertos comunicación serial, SPI, I2C y pines de control.

Figura 13. Raspberry Pi Modelo B

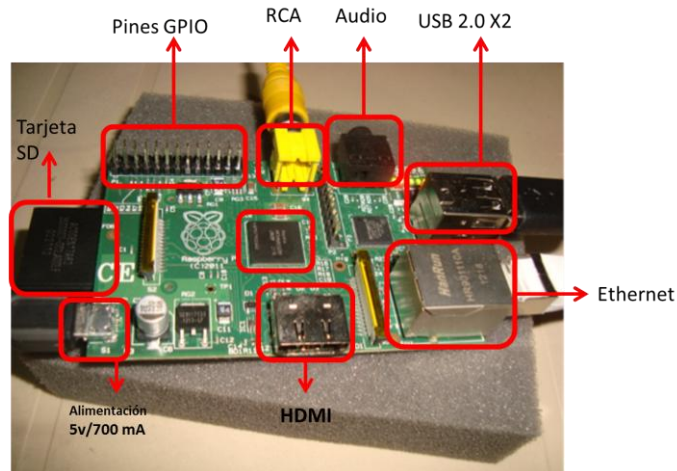


Tabla 4. Características Raspberry Pi

| | Raspberry Pi Modelo B |
|-------|------------------------------|
| SoC | Broadcom BCM2835 |
| CPU | ARM1176JZFS a 700 MHz |
| GPU | Videocore 4 |
| RAM | 256 MB |
| Video | HDMI Y RCA |

²³ <http://www.debian.org/intro/about>, 27/08/2012, all.

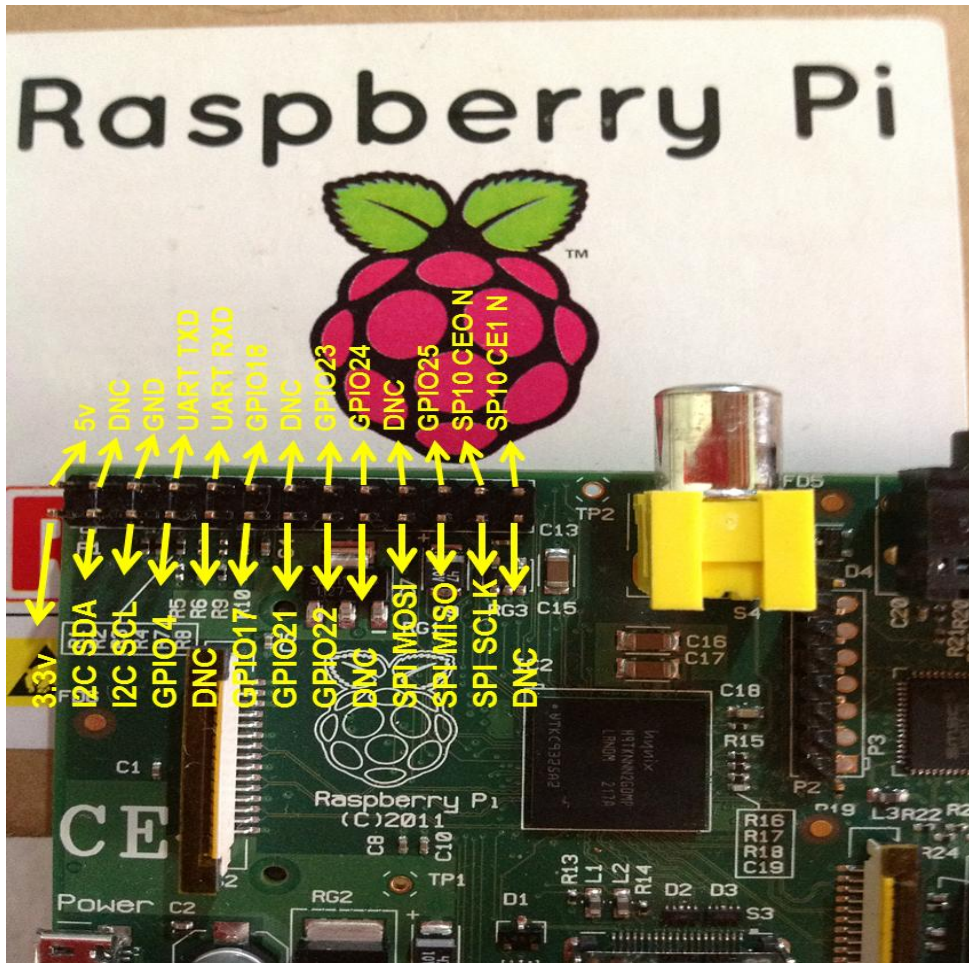
²⁴ lxde.org, 27/08/2012, all.

²⁵ <http://www.archlinux.org/>, 27/08/2012, all.

²⁶ <http://qt-project.org/wiki/Qt-RaspberryPi>, 27/08/2012, all.

| | |
|--------------|---------------------|
| Resolución | 1080p |
| Audio | HDMI Y 3.5 mm |
| USB | 2 x USB 2.0 |
| Redes | Ethernet 10/100 |
| Alimentación | Micro USB 5V, 700mA |
| Dimensiones | 85.6 x 53.98 x 17mm |

Figura 14. Pines GPIO Raspberry Pi



2.2 ESTADO DEL ARTE

En el presente proyecto se realizó una investigación documental donde se exploraron varias bases de datos nacionales e internacionales, así como también se exploraron centros de investigación y Bibliotecas de Universidades, seguidamente se analizaron un conjunto de trabajos que fueron útiles como base teórica para el desarrollo del trabajo que se muestra en este libro.

- **IPAD COMO ASISTENTE DE ACCESO PERCUTÁNEO AL RIÑÓN USANDO MARCADORES DE NAVEGACIÓN BASADO EN: LA EXPERIENCIA CLÍNICA INICIAL**

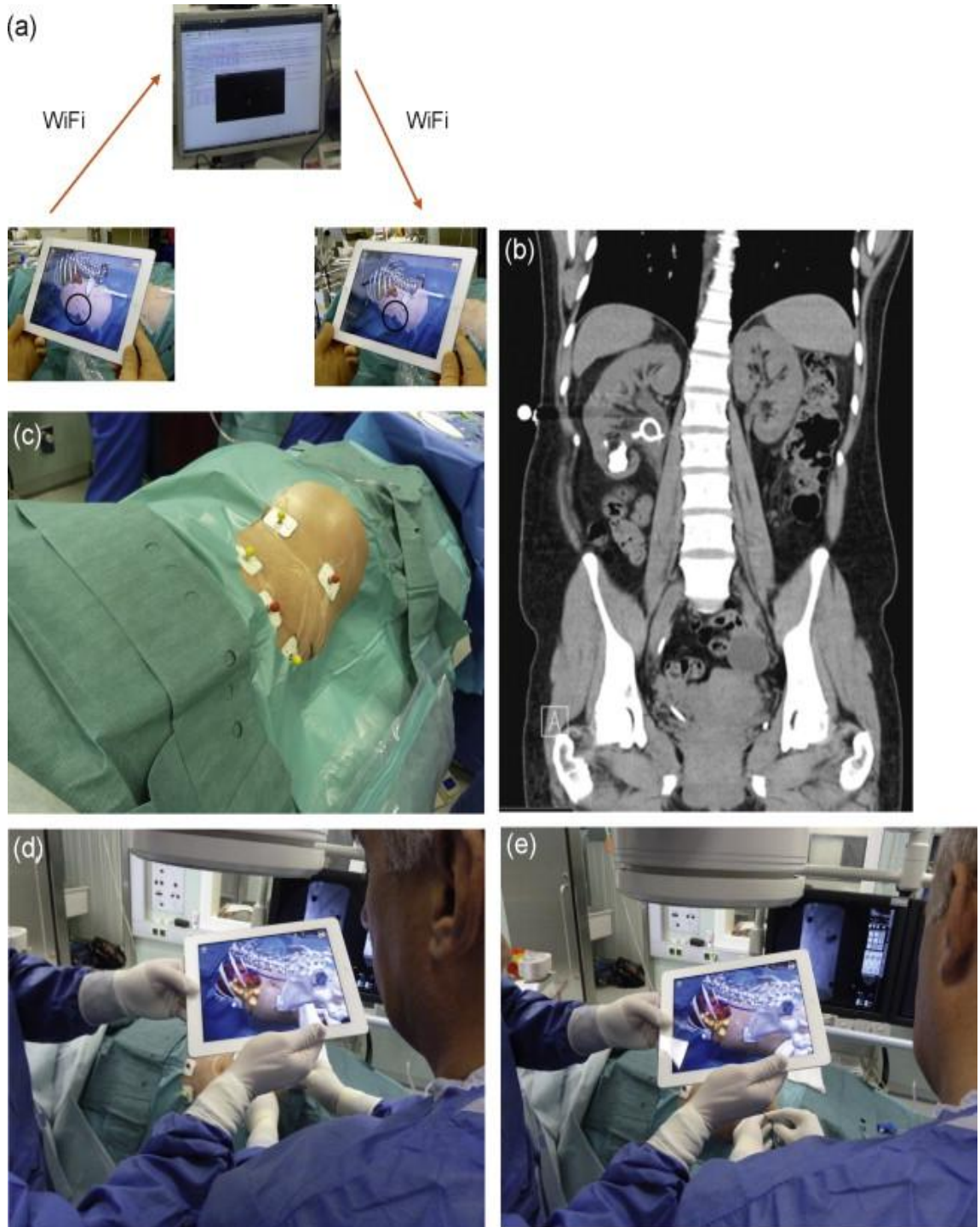
La cirugía asistida por computador para los procedimientos laparoscópicos de la próstata y riñón es un medio por el cual se obtiene experiencia preliminar en este tipo de operaciones. Basados en algoritmos desarrollados para el seguimiento de adentro hacia afuera a través de la laparoscopia. En este artículo se modifica esta técnica de seguimiento basado en el marcador de navegación para el acceso percutáneo al riñón usando un iPad.

El manejo del iPad asistido por la realidad aumentada demostró ser provechoso durante el acceso al riñón en la nefrolitotomía percutánea, proyecto desarrollado por Jens J. Rassweiler del departamento de urología, en la Universidad de Heridelberg en Alemania, el cual fue aceptado el 13 de diciembre de 2011. Este uso de cirugía asistido por computadora es simple y el sistema es rentable. Los estudios futuros deberían comparar este método con las técnicas manuales estándar, así como con robot con ayuda de los enfoques o costosos dispositivos radiológicos.

En la figura 15 se muestra un procedimiento asistido con el iPad por marcador de navegación basada en la del acceso percutáneo al riñón. (A) El iPad se utiliza como una cámara para tomar una imagen de la piel del paciente con los datos de captura de comprimidos y de transmitirlos a través de una red de área local inalámbrica en el servidor ubicado en la sala de control. (B) Vista longitudinal de los cortes. (C) La posición similar propensa de paciente sobre la nefrolitotomía percutánea amortigua con el remplazo exacto de los artículos de la navegación. (d) Determinación del sitio de punción del sistema de recogimiento que usa el iPad .²⁷

²⁷ RASSWEILER, Jens J. iPad -Assisted Percutaneous Access to the Kidney Using Marker-Based Navigation: Initial Clinical Experience. [En línea] <<http://www.sciencedirect.com/consultaremotu.upb.edu.co/science/article/pii/S0302283811013911>> [Consultado en 15 de Abril de 2012].

Figura 15. La experiencia clínica con el iPad ²⁷.



- **EL IPAD EN CARDIOLOGÍA**

La publicación de la Revista de la Universidad Americana de Cardiología (Journal of the American College of Cardiology, JACC) del mes de febrero del año 2011, expone al iPad como un desarrollo muy emocionante para los cardiólogos y ha cambiado la forma en que muchos de ellos han organizado su flujo de trabajo. Si bien no se notará un cambio radical cualitativo en los patrones de trabajo, la evolución acelerada de la transmisión de información será facilitada por la adopción de esta tecnología en la cultura cardiológica. Una vez que una masa crítica de personas que adopta un nuevo estándar, provocará un cambio cultural.²⁸

- **AYUDANTE ROBÓTICO DE CASA CARE-O-BOT**

Care-O-bot® es un desarrollo del Instituto Fraunhofer de Ingeniería de Fabricación y Automatización (IPA) de Stuttgart, Alemania, en el departamento de sistemas de robots. Las diferentes ayudas que proporciona este innovador robot permiten a los adultos mayores y a la gente minusválida vivir independientemente en sus casas particulares, siempre y cuando lo desean. Como una contribución a las soluciones tecnológicas necesarias, una plataforma de demostración de un sistema de atención domiciliario móvil, llamada Care-O-bot. Este artículo da una descripción general sobre funcionalidades Care-O-bot's y pruebas iniciales.²⁹

Care-O-bot es un robot de servicio móvil, como se muestra en la figura 16, que tiene la capacidad de realizar varias tareas de apoyo en entornos hogareños. Proporciona el apoyo y la ayuda de instrucción en la vida diaria de una persona y también promueve la auto iniciativa. Por lo tanto, un ayudante robótico de casa tendría que realizar las tareas siguientes: tareas de casa, ayuda de movilidad, comunicación e integración social, etc.

²⁸ MASHMAN, Walter. The iPad in Cardiology: Tool or Toy [En línea] <<http://www.sciencedirect.com.consultaremot.upb.edu.co/science/article/pii/S1936879811000033>> [Consultado en 15 de Abril de 2012].

²⁹ HANS, M. Robotic home assistant Care-O-bot: past-present-future. [En línea] <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1045652&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1045652> [Consultado en 15 de Abril de 2012].

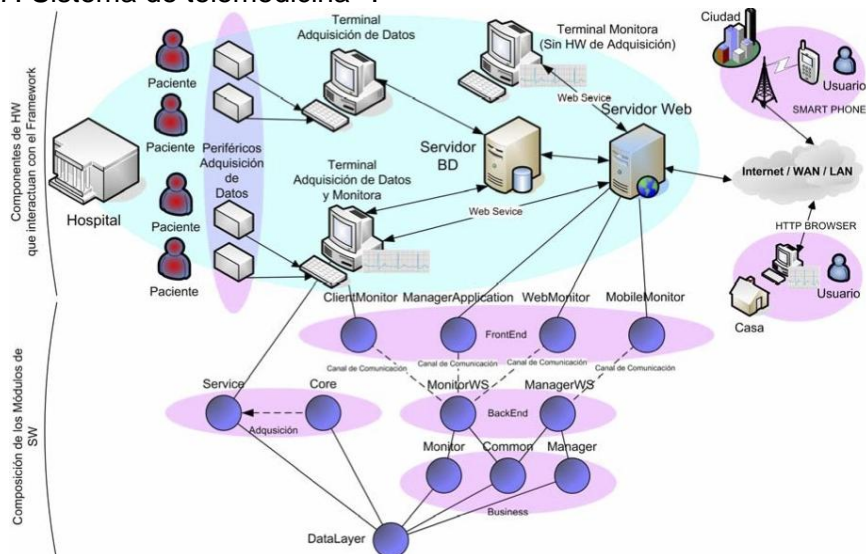
Figura 16. Care-O-bot entregando una bebida a un adulto mayor³⁰



- **ARQUITECTURA PARA EL MONITOREO REMOTO DE FUNCIONES VITALES EN PACIENTES AMBULATORIOS.**

Se propone un sistema de telemedicina aplicado para el monitoreo de ECG con acceso remoto a diferentes clientes (ver figura 17), la investigación se llevó a cabo en la Universidad de Moron Moron, Buenos Aires, Argentina por la Facultad de Informática, David Andrada, Pablo Martín Sparhaki, Hernán Matías Novillo, Jorge Lerache.

Figura 17. Sistema de telemedicina³¹.



³⁰ <http://idwf.eu/de/news432828>, 10/07/2012, all.

³¹ ANDRADA, David, et al. Arquitectura para el monitoreo remoto de funciones vitales en pacientes ambulatorios. En: Congreso Argentino de Ciencias de la Computación (12: 17-21, octubre: Potrero de los Funes, Buenos Aires, Argentina). Memorias. Universidad de Moron Moron. 2006.

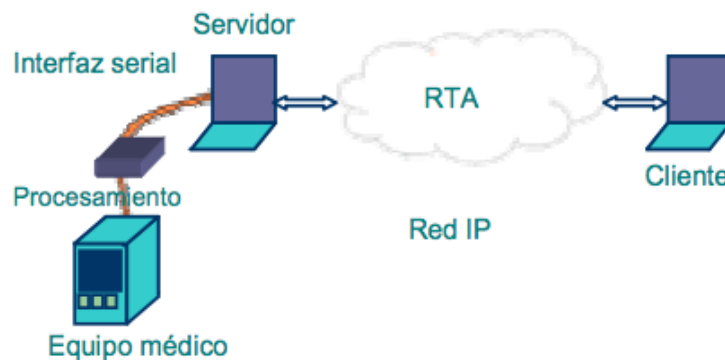
La implementación de servidores para la telemedicina ofrece una amplia red de comunicación, que simplifica a los usuarios el acceso a diferentes aplicaciones que se ajustan a la necesidad de cada uno, de forma remota.

- **INTERFAZ PARA EQUIPOS DE MONITOREO MÉDICO (RED DE TELEMEDICINA ANTIOQUIA)**

En la Universidad Pontificia Bolivariana de Medellín, Colombia, se está llevando a cabo el desarrollo de dos interfaces de monitoreo remoto que buscan conformar la red de telemedicina de Antioquia. El proyecto lo lleva a cabo el EQUIPO INTERDISCIPLINARIO DE TELEMEDICINA (EIT) de la Universidad Pontificia Bolivariana de Medellín.

Con este proyecto se quiere dar un soporte de mayor conocimiento a los médicos rurales, de tal manera que estos puedan acceder a consultas con médicos especialista que se encuentren en centros de alto nivel. Para lograr esto se utiliza una arquitectura Cliente-Servidor (ver figura 18) en la cual el servidor realiza la comunicación con el equipo médico y el cliente es quien interactúa con él desde un sitio remoto.

Figura 18. Arquitectura de la red³²



En este proyecto se han desarrollado dos equipos el DINAMAP PLUS y el NELLCOR N100, estos equipos aportan al crecimiento de la telemedicina en Colombia, además la implementación de estos equipos junto a los equipos médicos brinda nuevas características que algunos equipos médicos no traen de fábrica como puede ser la impresión de informes, almacenamiento de datos, mejor visualización de variables tomadas, entre otros.

³² CARDONA, Oscar, et al. Arquitectura para el monitoreo remoto de funciones vitales en pacientes ambulatorios. En: Congreso de Sociedad Cubana de Bioingeniería (6: 19-22, Julio: Habana, Cuba). Memorias. 2005.

3. ESTRUCTURA DEL PROYECTO

3.1 OBJETIVO GENERAL

Desarrollar una plataforma hardware software para el control de un robot móvil a escala utilizando el iPad, sistemas de comunicación Wi-Fi, sensores de obstáculos y posición.

3.2 OBJETIVOS ESPECIFICOS

- Implementar un software para comunicar el sistema robótico móvil con el iPad vía Wi-Fi.
- Diseñar una interfaz gráfica, para controlar el robot en sus movimientos básicos desde el iPad.
- Implementar una interfaz gráfica para la adquisición de señales de un GPS, una brújula y sensores ultrasónicos emitidos desde el robot.
- Diseñar e implementar el software para el monitoreo del robot vía internet.

4. DISEÑO E IMPLEMENTACIÓN DEL HARDWARE Y SOFTWARE

El sistema diseñado está compuesto por varios módulos de trabajo, en la figura 19 se puede observar un diagrama de bloques que permite la visualización del funcionamiento general, además se muestra el flujo de la información entre cada una de las partes fundamentales del equipo.

Inicialmente se observa la capa interna robot, Donde se cuenta con un robot móvil diseñado anteriormente en la tesis "*Diseño e implementación de una Red neuronal en un DSPIC para el control de trayectoria de un robot detector de obstáculos*" de la cual se tomo la estructura física y sensorica. Adicionalmente se instaló un GPS y una tarjeta de transmisión de información vía Wi-Fi.

Después de la primera capa se encuentra la capa de enlace y monitoreo que está compuesta por un Raspberry Pi configurado como servidor y una página web para el monitoreo de información, esta capa es el punto intermedio al que los clientes del sistema acceden para realizar intercambio de información.

Finalmente la capa control, la compone un iPad para la cual se desarrollo una aplicación que permite al usuario de forma amigable controlar el robot y visualizar la información de los sensores.

Se cuenta además con una segunda opción para el control del robot que se aprecia en la figura 20, la cual presenta una configuración del sistema en modo local de tal forma que el iPad se comunica de forma directa con el robot utilizando una comunicación ad-hoc, aportando como gran ventaja un aumento de velocidad en la comunicación pero a su vez tiene la desventaja de perder el monitoreo remoto y el almacenamiento de información en la base de datos ya que como muestra el diseño la capa de enlace no es requerida.

Figura 19. Diagrama diseño del sistema remoto

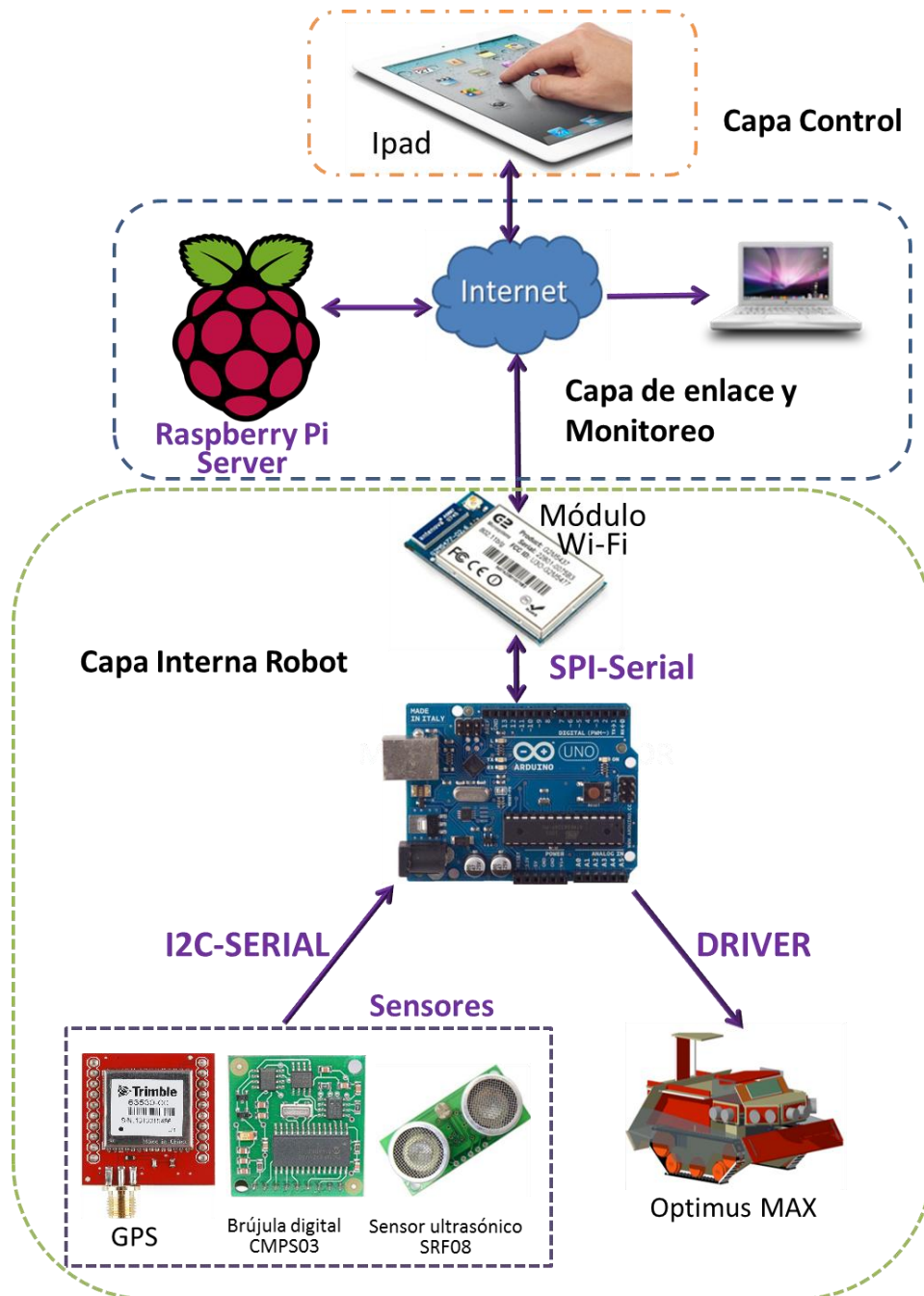
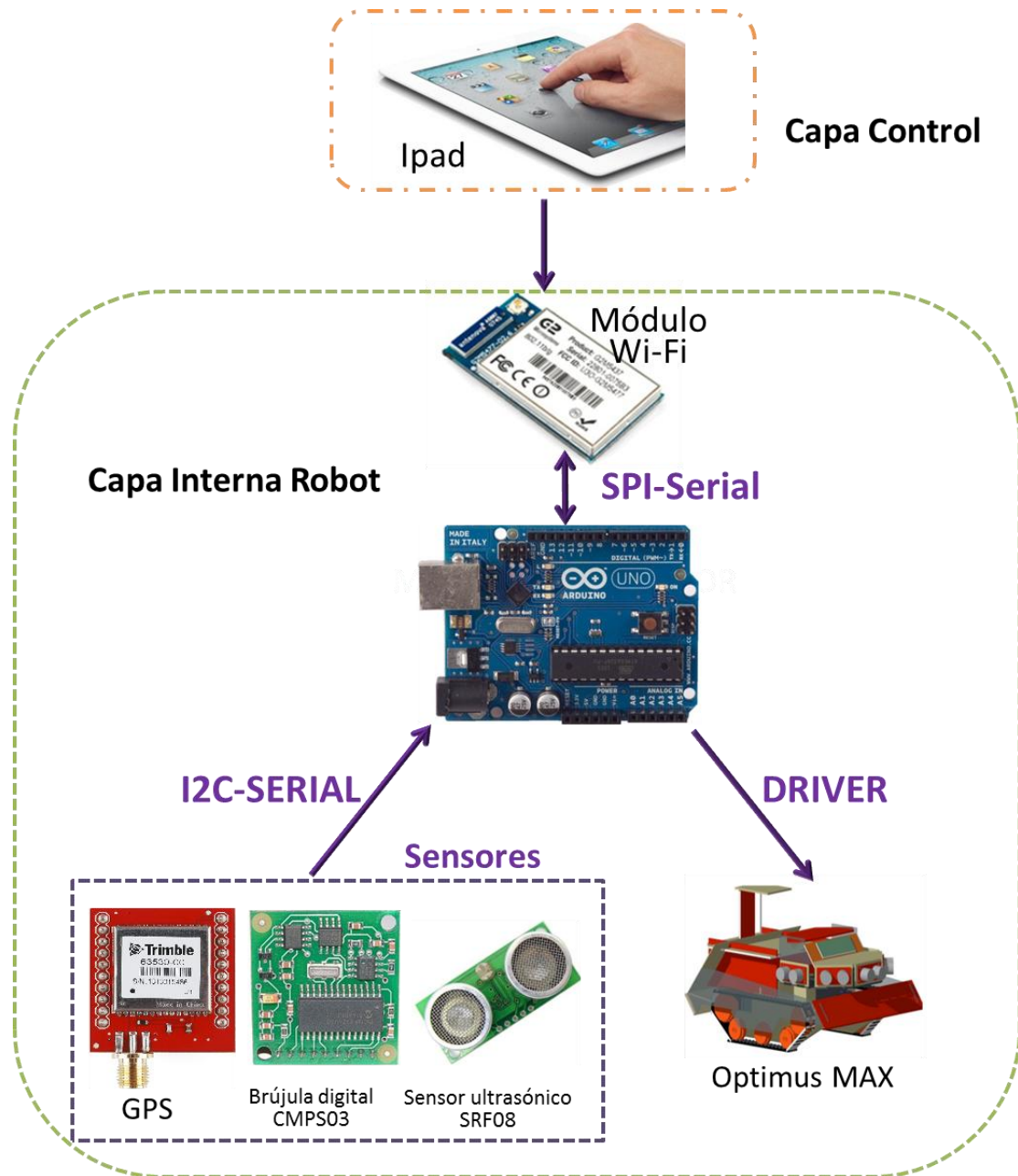


Figura 20. Diagrama diseño del sistema punto a punto



4.1 Tipos de configuración del sistema

Se desarrollaron dos sistemas de control para el robot, el primero de estos se compone de tres capas de funcionamiento que son la interna del robot, capa de enlace y la capa de control. En este sistema se implementa un servidor web para realizar el control del robot, se requiere de conexión a internet para su funcionamiento, este sistema tiene grandes ventajas como son el almacenamiento de información en una base de datos, el monitoreo de forma remota y también el control del robot de forma remota, su principal desventaja consiste en la velocidad de respuesta del robot a los comandos de control, ya que tiene un retardo debido a los diferentes enlaces que realiza la señal para llegar de un punto a otro.

El segundo sistema desarrollado está conformado por dos capas de funcionamiento que son la capa interna del robot y la capa de control, este sistema consiste en la creación de una red ad-hoc por el robot a la cual el iPad se conecta vía Wi-Fi, este sistema no requiere de conexión a internet ya que la misma se realiza punto a punto, la gran ventaja que aporta esta configuración es la velocidad de respuesta del robot al momento de recibir los comandos de control, es prácticamente inmediata. Utilizando esta configuración el sistema pierde sus funciones de monitoreo y almacenamiento de información siendo esto una desventaja considerable.

Es importante señalar que la escogencia del modo de funcionamiento depende de la aplicación que se le quiere dar al sistema.

4.2 Capa Interna Robot

Para integrar los componentes que componen la primera capa del sistema a la estructura del robot se diseñó y desarrolló una tarjeta a la que se conectan los sensores ultrasónicos, brújula, GPS y motores, como se puede ver en la figura 21 la tarjeta se construyó para mantener compatibilidad con el Arduino Uno(ver figura 22). A continuación se describe de forma detalla el diseño.

Figura 21. Tarjeta de sensores y driver

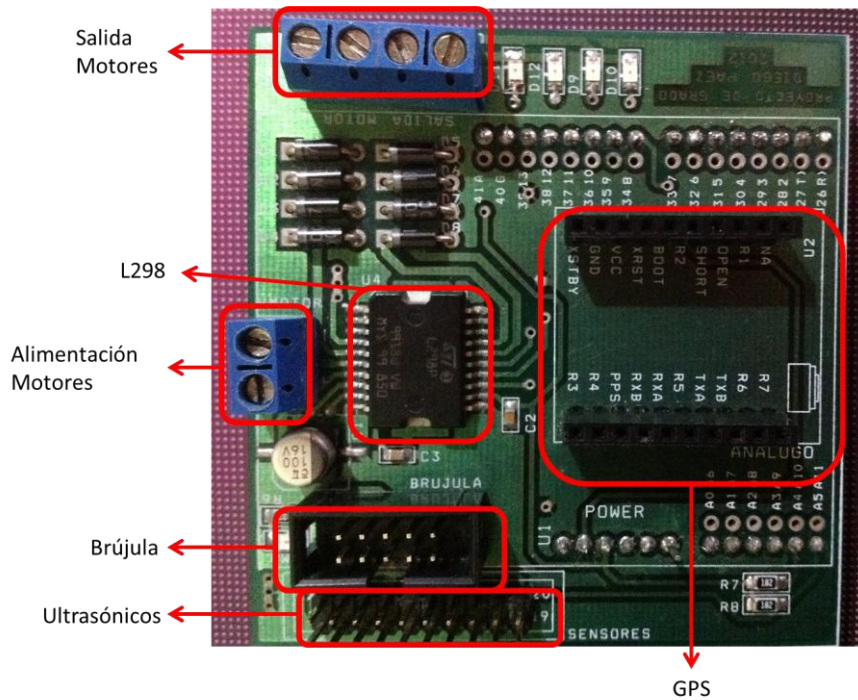
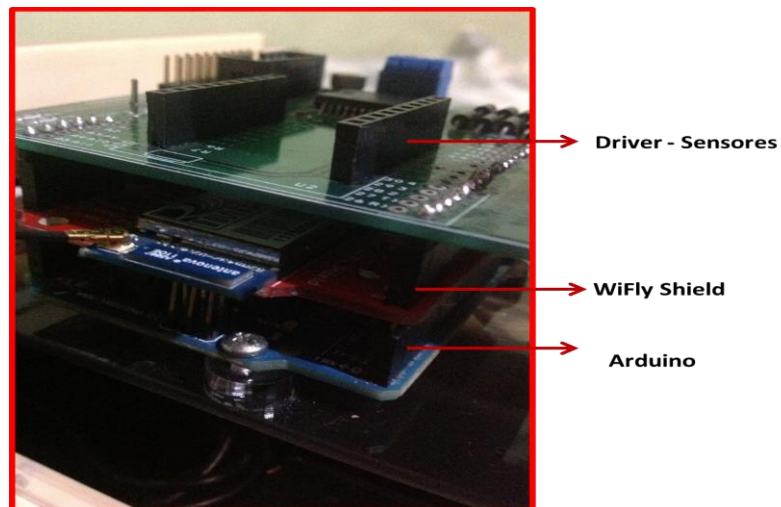


Figura 22. Tarjetas Capa Interna Robot



4.2.1 Motores

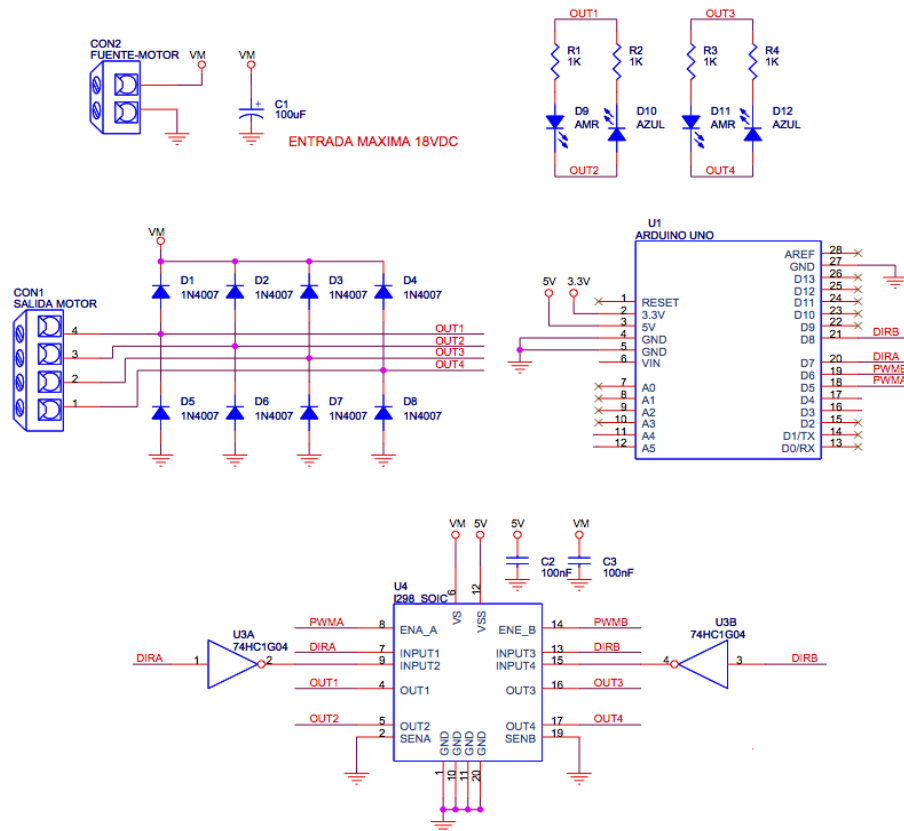
Para controlar los motores se utilizó un puente h dual L298³³, este puente h maneja hasta 4 amperios, controla 4 motores en un sentido y dos si realiza cambio en el sentido de giro, la velocidad de cada motor se puede variar con una señal de PWM

³³ http://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf, 27/08/2012, all.

que proviene desde el microcontrolador. En la figura 23 se observa la conexión que se realizó entre el microcontrolador y el puente h.

Para controlar la dirección de los motores se utilizan dos pines DIRA y DIRB digitales del Arduino Uno, se utilizan dos compuertas negadoras ya que el L298 tiene dos pines para el control del sentido motor de giro por motor en los cuales si los valores digitales son iguales los motores se detienen y si son diferentes se mueven adelante o atrás, las salidas PWMA y PWMB se utilizan para controlar la velocidad, estos dos pines se conectan a los habilitadores del puente h de tal forma que si el ancho de pulso es mínimo el motor se detiene y si es máximo el motor alcanza su velocidad máxima. La alimentación de los motores se realiza con 8 baterías AA de 1.2 V cada una, estas se conectan al pin VS del L298 aislando la alimentación de los motores de la del microcontrolador.

Figura 23. Esquemático Motores



El programa que se utiliza para controlar los motores desde el Arduino Uno se desarrolló por módulos, de tal forma que un modulo pueda ser convocado en cualquier parte del programa sin necesidad de escribir todo el código que corresponde al mismo. Los módulos son los siguientes.

En la figura 24 se encuentran los módulos que controlan el sentido de giro y velocidad de los motores, para que el motor se mueva adelante o atrás la velocidad

debe ser la misma en ambos por esto el ancho del pulso en cada habilitador debe ser igual.

Figura 24. Módulos control adelante, atrás.

```
void MotorAdelante(){  
  
    digitalWrite(dir_a, LOW);  
    digitalWrite(dir_b, HIGH);  
  
    void MotorAtras(){  
  
        digitalWrite(dir_a,HIGH);  
        digitalWrite(dir_b, LOW);  
    }  
    void MotorMaxVel(){  
  
        analogWrite(pwm_a, 250);  
        analogWrite(pwm_b, 250);  
    }  
}
```

Para cambiar la dirección del robot, es decir moverlo a la derecha o izquierda se debe disminuir la velocidad de uno de los motores como se puede ver en la figura 25, esto se logra disminuyendo el ancho del pulso que va a uno de los habilitadores del puente h.

Figura 25. Módulos control derecha, izquierda

```
void MotorDerecha(){  
  
    digitalWrite(dir_a, LOW); //  
    digitalWrite(dir_b, HIGH); //  
    analogWrite(pwm_b, 190);  
    analogWrite(pwm_a, 80);  
  
}  
  
void MotorIzquierda(){  
  
    digitalWrite(dir_a, LOW); //  
    digitalWrite(dir_b, HIGH); //  
    analogWrite(pwm_a, 190);  
    analogWrite(pwm_b, 80);  
  
}
```

4.2.2 Brújula³⁴ y Sensores Ultrasónicos³⁵

La brújula (compas electrónico CMPS03) y los sensores ultrasónicos (SRF08) se comunican con el microcontrolador utilizando el protocolo I2C, solo utilizan dos pines del Arduino Uno, SDA para la transferencia de datos, SCL para el control del reloj. En la figura 27 se muestra el esquemático para la conexión de los sensores con el Arduino Uno, se utilizan dos resistencias pull-up para las líneas de I2C ya que estas son drenador abierto y las necesitan para su funcionamiento. Adicionalmente la brújula tiene una salida PWM que indica la posición de la misma, se instalo un led para verificar su buen funcionamiento. En la figura 26 se ve la tarjeta de sensores con la brújula conectada.

³⁴ <http://www.superrobotica.com/s320160.htm>, 29/08/2012, all.

³⁵ <http://www.superrobotica.com/s320112.htm>, 29/08/2012, all.

Figura 26. Brújula

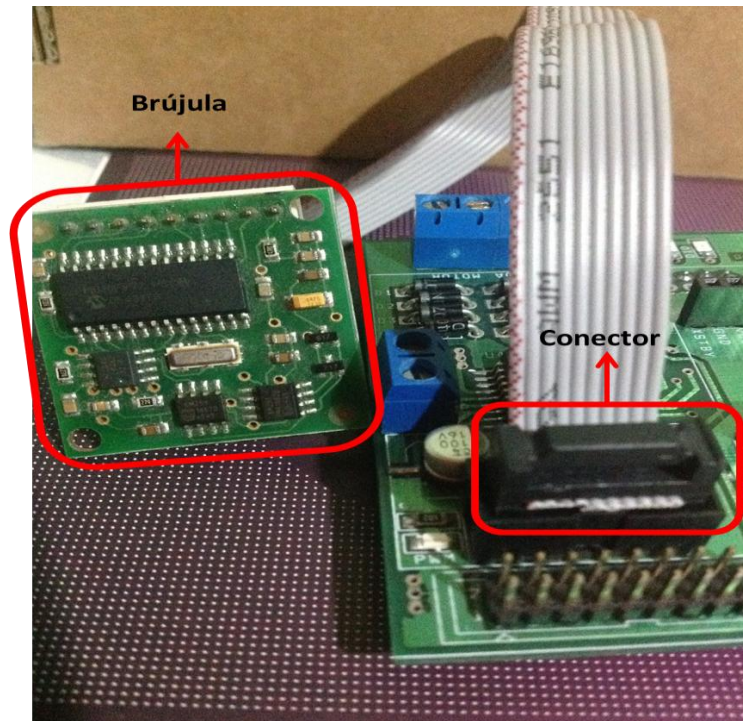
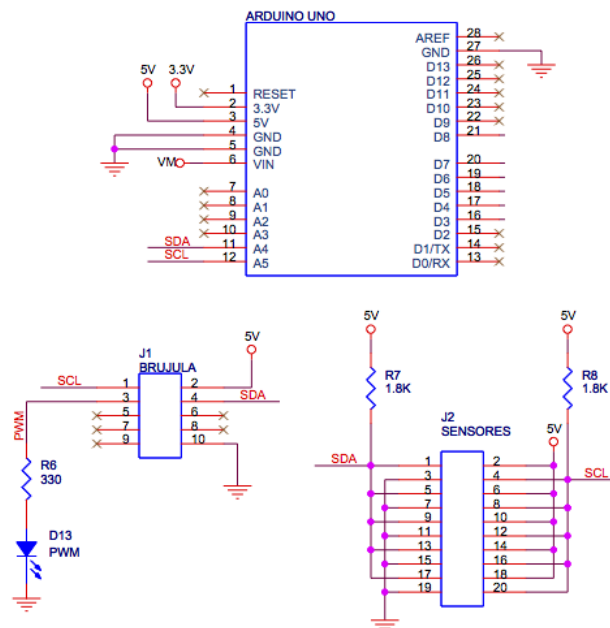


Figura 27. Esquemático brújula y motores

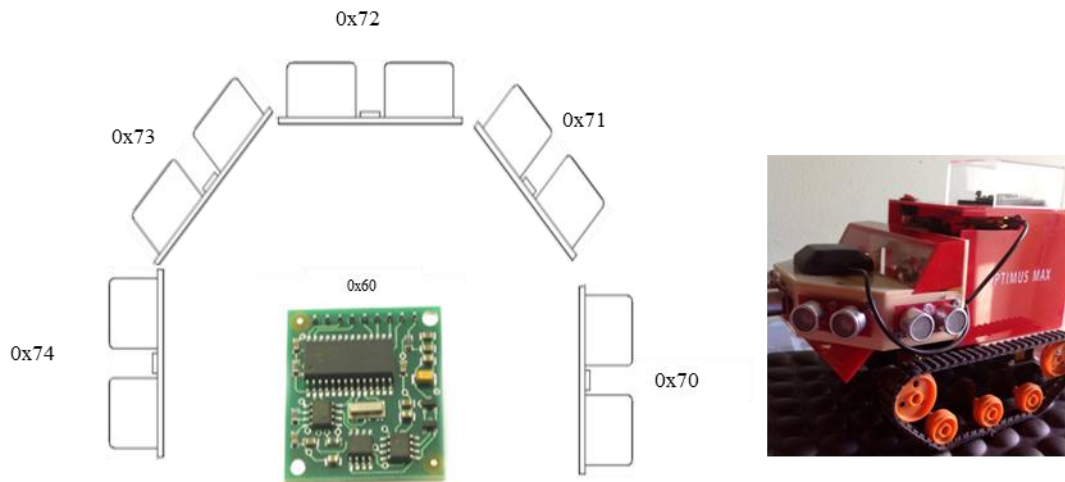


El programa para acceder a la información de la brújula y los ultrasónicos se desarrolló de la misma forma que el de los motores. Este programa requiere la

implementación de la librería Wire.h que incluye el entorno de desarrollo del Arduino Uno, esta librería gestiona el envío y recepción de información utilizando el protocolo I2C.

Al conectar diferentes dispositivos al mismo puerto, cada uno debe tener una dirección diferente, para este caso la brújula utiliza la dirección 0x60 y los ultrasónicos 0x70, 0x71, 0x72, 0x73, 0x74, así cuando se requiera la información de uno de los sensores se hace la petición a una dirección específica lo cual se señala en la figura 28.

Figura 28. Distribución Sensórica



La librería Wire.h se declara en la cabecera del programa utilizando la función `#include <Wire.h>`, en la figura 29 se encuentra el modulo con el cual se realiza la lectura de la brújula. La comunicación inicia con el reconocimiento de la dirección de la brújula, seguido de esto se envía el valor del registro que se quiere leer en este caso el número 2 que indica la dirección de 0 a 360 grados en 2 bytes. El proceso que sigue a continuación es el almacenar los dos bytes recibidos, sumarlos y guardar el resultado para ser utilizado cuando se requiera.

Figura 29. Modulo leer brújula

```
void getBrujula(){
  Wire.beginTransmission(address); //Inicia la comunicación con la
  brújula
  Wire.write(2);                //Se envía el registro que se quiere
  leer
  Wire.endTransmission();

  Wire.requestFrom(address, 2);
  while(Wire.available() < 2); // se mantiene esperando
  mientras envíe datos
  highByte = Wire.read();      //Almacena el byte mas y menos
  significativos
  lowByte = Wire.read();
  grados = ((highByte<<8)+lowByte)/10; // guarda el valor en un
  rango de 0 a 360

}
```

Para acceder a la información de un sensor ultrasónico se utiliza el siguiente módulo (figura 31), de esta manera se obtiene la información de un solo sensor, para evitar un código extenso se realiza la lectura de todos los sensores ultrasónicos al mismo tiempo almacenando el resultado de cada uno en un vector, estos valores se utilizan constantemente para evitar que el robot choque con obstáculos que se encuentre a su alrededor.

La figura 30 presenta el diagrama de tiempos que permite el acceso de cualquiera de los registros internos del módulo SRF08. *“La comunicación comienza con el bit de inicio, a continuación se envía la dirección del módulo deseado en modo escritura, seguidamente se transmite el registro interno al cual se quiere acceder. este proceso se ejecuta 5 veces con la dirección de cada sensor. En este punto se vuelve a generar un bit de inicio y a mandar la dirección del módulo pero en modo de lectura, Acto seguido se empieza a recibir el contenido de los registros internos a partir del indicado. Cuando se recibe el último byte deseado, se transmiten los bits de parada”*³⁶. El proceso descrito anteriormente toma 65 mS por esto es necesario realizar un retardo de 100 mS cada vez que se realiza una lectura.

³⁶ <http://www.unrobotica.com/manuales/srf08.pdf>, 27/08/2012, all.

Figura 30. Diagrama de tiempos SRF08³⁶

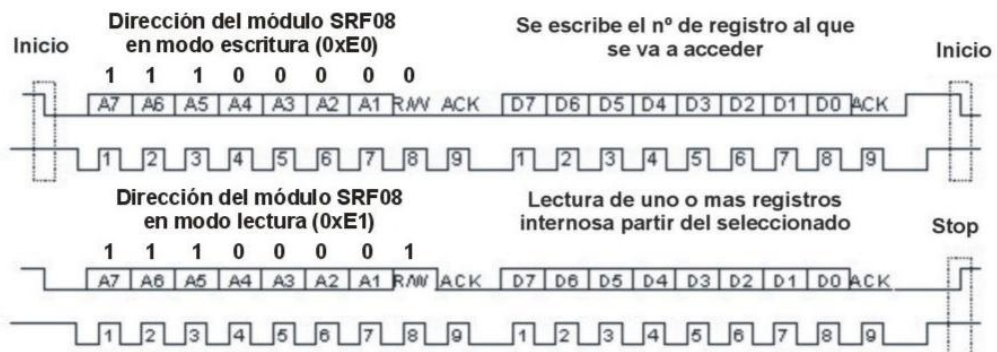


Figura 31. Modulo leer Ultrasónicos

```

void getUltrasonicos()
for (i = 0; i < 5; i = i + 1) {
  direccion=srfAddress[i];
  rangeData[i] = getRange();
  sensorRango=rangeData[i];
}
s1=rangeData[0];
s2=rangeData[1];
s3=rangeData[2];
s4=rangeData[3];
s5=rangeData[4];
}

int getRange(){
  int range = 0;
  Wire.beginTransmission(direccion);           // Iniciar comunicación
  SRF08
  Wire.write(cmdByte);                         // Command Byte
  Wire.write(0x51);                             // Lectura en Centimetrs
  Wire.endTransmission();
  delay(100);
  Wire.beginTransmission(direccion);
  Wire.write(rangeByte);                       // Registro Lectura
  Wire.endTransmission();
  Wire.requestFrom(direccion, 2);              // pide los bytes
  while(Wire.available() < 2);                 // espera la llegada de
  datos
  highByte = Wire.read();
  lowByte = Wire.read();

  range = (highByte << 8) + lowByte;
  return(range); // R
}

```

4.2.3 GPS

El GPS transmite la información al microcontrolador por medio del puerto serial, para comunicarlo con el Arduino Uno se utiliza la librería SoftwareSerial.h y Tinygps.h. La primera permite configurar el Arduino Uno con un segundo puerto serial presentado en la figura 32, de esta manera se crea un puerto exclusivo para la lectura del GPS, la segunda librería utilizada Tinygps.h convierte el código NMEA que permite visualizar la latitud y longitud en la que se encuentra el dispositivo. La figura 33 muestra el GPS sobre la tarjeta diseñada.

Figura 32. Esquemático GPS

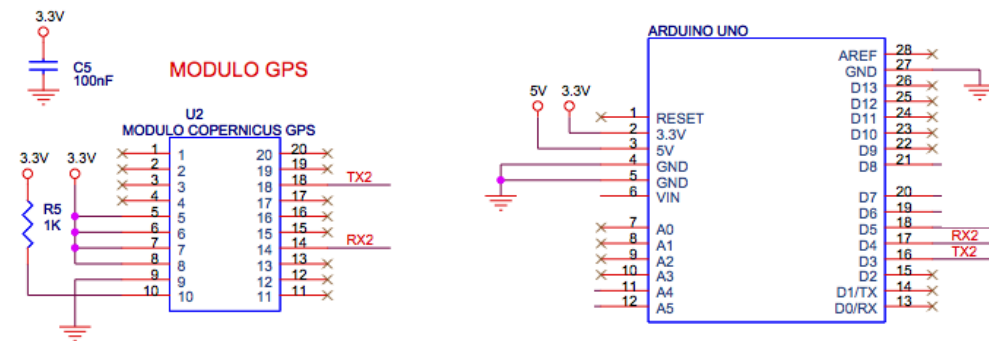
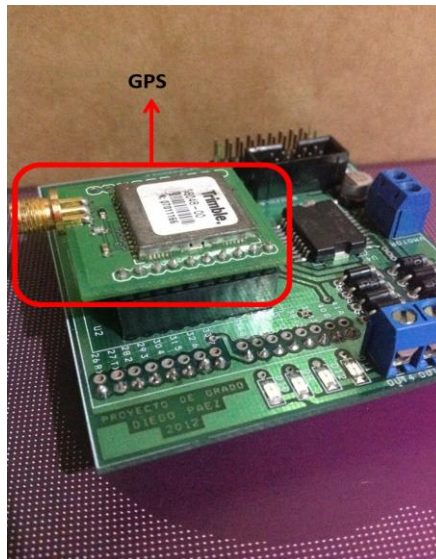


Figura 33. GPS en la tarjeta



En la figura 34 aparece el módulo GPS el cual realiza la lectura del código proveniente del GPS. Mientras el dispositivo se enlaza con al menos 6 satélites no

se obtiene respuesta cuando, el proceso de enlace puede tomar un tiempo incluso es posible que no se realice si el modulo se encuentra en un lugar cubierto.

Figura 34. Modulo GPS

```
void GPS(){
  bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;
  for (unsigned long start = millis(); millis() - start < 2000;)
  {
    while (ss.available())
    {
      char c = ss.read();
      if (gps.encode(c) // Did a new valid sentence come in?
          newData = true;
    }
  }

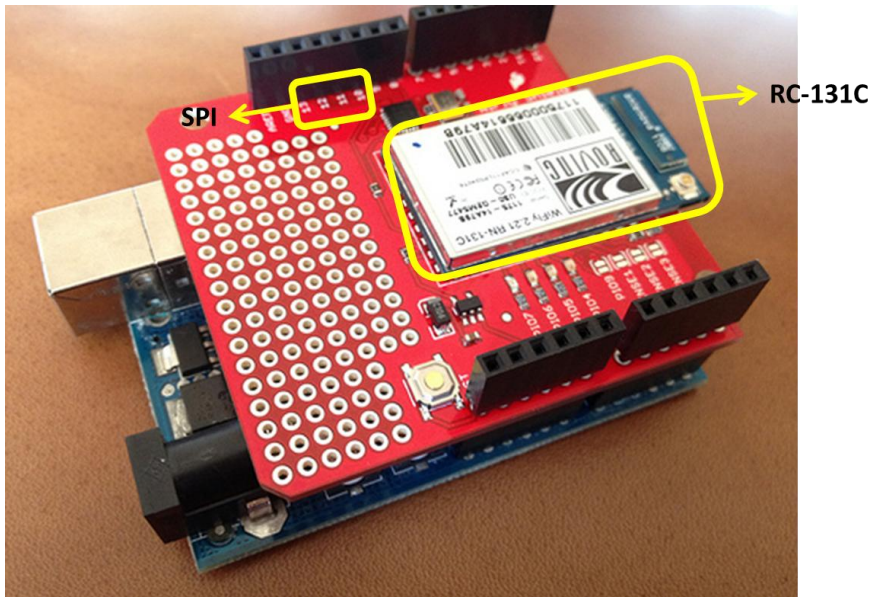
  if (newData)
  {
    float flat, flon;
    unsigned long age;
    gps.f_get_position(&flat, &flon, &age);
    Serial.println(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat,
6);
    Serial.println(flou == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon,
6);

  }
}
```

4.2.4 Conexión Wi-Fi

La conexión a la red Wi-Fi se realiza utilizando la tarjeta Wifly Shield (ver figura 35), esta tarjeta se comunica con el microcontrolador utilizando el protocolo SPI. Para la implementación de esta tarjeta se utiliza la librería SPI.h, que gestiona la comunicación entre el microcontrolador y la Wifly, también se utiliza la librería WiFly.h que cumple la función de configuración del modulo RC-131 que se encuentra en la tarjeta.

Figura 35. Wifly Shield



La Capa Interna Robot es un cliente que hace parte de la arquitectura Cliente/Servidor, por esto la tarjeta Wifly Shield se configura de tal forma que el Arduino Uno sea un cliente en el sistema. El primer paso para la configuración es indicar la dirección IP del servidor al cual se accederá, después de esto se debe ingresar la dirección de la pagina Web y el puerto de acceso, ver la figura 36.

Figura 36. Configuración cliente

```
byte server[] = { 192, 168, 2, 24 }; // Direccion Ip Servidor Local
WiflyClient client("192.168.2.24", 80);
```

Para realizar el envío de datos al servidor se hace por grupos de dos para no saturar el microcontrolador y evitar que se reinicie. El cliente se conecta a la dirección indicada del servidor y así se inicia la transmisión continua de datos, cuando la transmisión de datos finaliza el cliente se desconecta. En la figura 37 se observa un módulo que envía dos datos al servidor, esto se realiza utilizando la función GET, el servidor está configurado para recibir el tipo de información indicado y almacenar la información en variables locales.

Figura 37 Envío de datos vía Wi-Fi

```
void send_data1(int s1,int s2) {  
  
    String request = "GET " + request_url() + "?s1=" + String(s1) + "&s2=" +  
String(s2) + " HTTP/1.1";  
    client.connect();  
    Serial.println(request);  
    client.println(request);  
    client.println("Host: 192.168.2.24");  
    client.println();  
    client.stop();  
}  
  
String request_url(){  
    String url_template = "/index.php/";  
    return url_template;  
}
```

La figura 38 señala el proceso de recepción de información, primero se debe verificar la conexión para no recibir datos erróneos. Si el cliente está conectado al servidor se indica la dirección de acceso y se realiza la lectura de datos de forma continua hasta que encuentra una "@", toma el dato siguiente y según su valor se realiza una acción deseada. El servidor está configurado para imprimir una "@" y después el dato de interés que se utiliza para realizar una acción en el robot.

Figura 38 Recepción datos vía Wi-Fi

```
int getWebValue() {  
  
    if (!client.connected()) {  
        Serial.println("Connecting...");  
        if (!client.connect()) {  
            return 0;  
        }  
    }  
    client.println("GET /control.php HTTP/1.1");  
    client.println("Host: 192.168.2.24");  
    client.println();  
  
    while (client.available()) {  
        ch = client.read();  
        Serial.print(ch);  
        if(ch=='@'){  
            c = client.read();  
            Serial.print(c);  
            switch(c){  
            case '1':  
                seguridad();  
                if(p==0){  
                    MotorMaxVel();  
                    MotorAdelante();  
                }  
                break;  
            case '2':  
                seguridad();  
                if(p==0){  
                    MotorDerecha();  
                }  
                break;  
            }  
        }  
    }  
    break;  
}
```

La Wifly Shield puede crear su propia red Wi-Fi cuando se configura como ad-hoc, es decir se crea una red descentralizada mediante la cual se intercambia información punto a punto entre la tarjeta y el dispositivo que se conecte a ella. Cuando se configura la tarjeta en modo ad-hoc, la dirección IP se vuelve estática, se desactiva la función DHCP que es la encargada de asignar una IP automática al cliente cuando se conecta al servidor. La librería Wifly.h tiene la posibilidad de realizar la configuración ad-hoc, en la figura 39 y figura 40 se puede observar un programa en el cual se realiza la configuración y se lee la información que llega desde un cliente conectado a la tarjeta.

Figura 39 Configuración ad-hoc

```
#include <SPI.h>
#include <WiFi.h>

boolean incoming = 0;

char ssid[] = "Optimus"; //Nombre de la Red

WiFiServer server(80);

void setup() {
  Serial.begin(9600);
  WiFi.begin(true);

  if(!WiFi.createAdHocNetwork(ssid)) {
    while(1) {
      Serial.println("Failed to create network, handle error properly?");
    }
  }

  Serial.println("Network Created");
  Serial.print("IP: ");
  Serial.println(WiFi.localIP());

  Serial.println("Starting Server");
  server.begin();
}
```

Para enviar un dato desde el cliente a la tarjeta se debe ingresar desde un navegador a la dirección IP seguida de un signo dólar y el valor que se quiera enviar de la siguiente manera: [http://169.254.1.1/\\$1](http://169.254.1.1/$1), cuando el cliente se conecta, se realiza la lectura entre el signo dólar y un espacio en blanco señalado en la figura 40, al finalizar la lectura se realiza la acción deseada.

Figura 40 Recepción datos ad-hoc

```
void loop() {
  WiFiClient client = server.available();
  if (client) {
    boolean current_line_is_blank = true;
    Serial.println("Got Client");
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        if (incoming && c == ' ') {
          incoming = 0;
        }
        if(c == '$'){
          incoming = 1;
        }
      }

      if(incoming == 1){
        Serial.println(c);

        if(c == '1'){
          //hace algo
        }
        if(c == '2'){
          //hace algo
        }
      }
    }
    if (c == '\n') {
      current_line_is_blank = true;
    }
    else if (c != '\r') {
      current_line_is_blank = false;
    }
  }
}
delay(10);
}
```

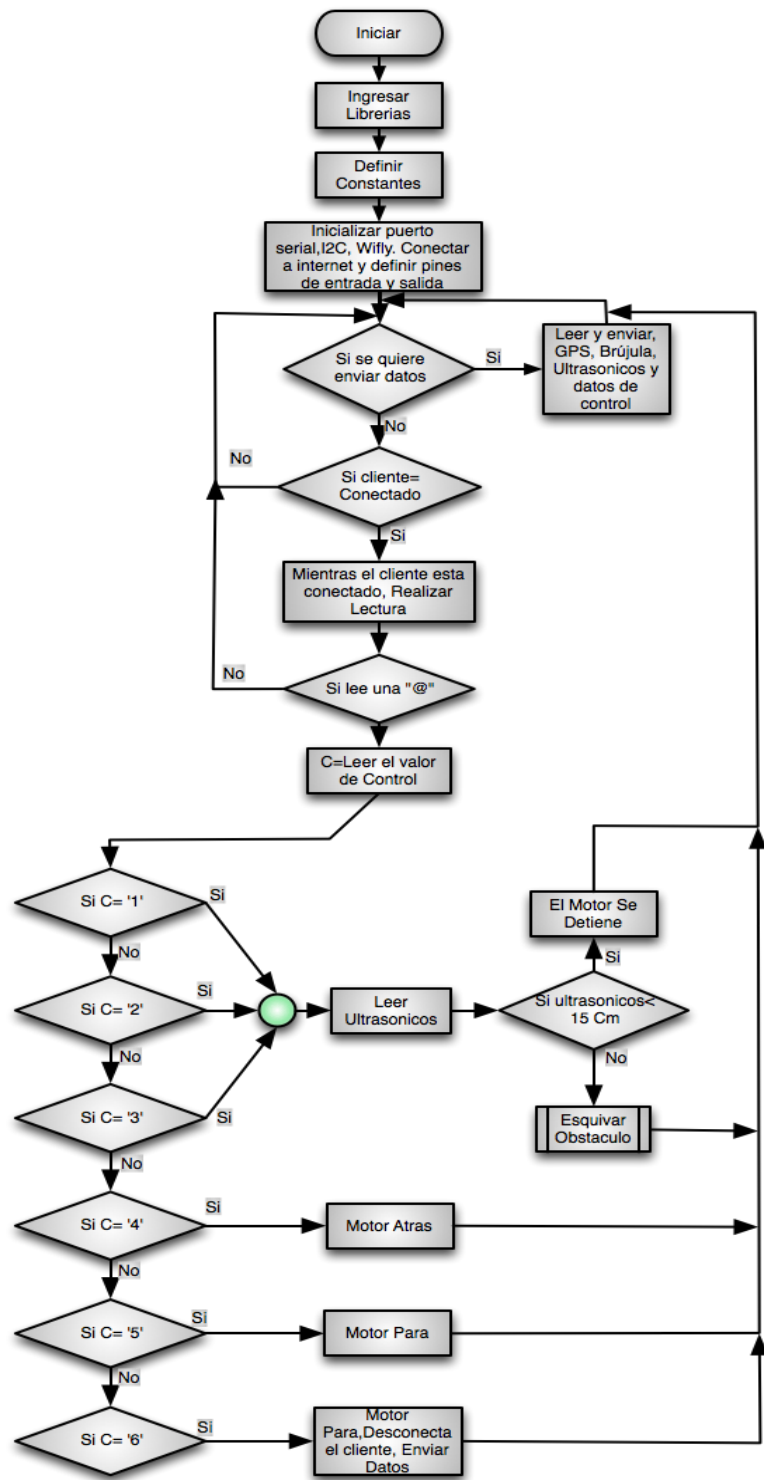
4.2.5 Software del Arduino Uno

El Arduino Uno se programó para realizar el proceso de adquisición de datos de los sensores, transmisión y recepción vía WiFi de la información y el control de motores del robot. En el diagrama de flujo de la figura 41 se muestra como se integran todos los dispositivos de la Capa Interna Robot al microcontrolador.

El programa inicia con la configuración e inicialización de las librerías que requieren los dispositivos mencionados anteriormente. Primero se verifica si el usuario indica que se realice la lectura de los datos, si esto ocurre se procede a la lectura de los sensores y posteriormente el envío de los resultados obtenidos a la base de datos, después de enviar la información el robot queda habilitado para ser controlado por el usuario nuevamente. Los sensores ultrasónicos cumplen la función de evitar

choques con obstáculos detectados a una distancia menor a 15 cm, por esto cuando un obstáculo es detectado el robot se detiene permitiendo al usuario aplicar reversa al robot y así esquivar el obstáculo.

Figura 41 Diagrama de flujo Arduino Uno



4.3 Capa de Enlace y Monitoreo

La capa de enlace y monitoreo es como su nombre lo indica, el enlace entre el iPad y el robot. En esta capa se desarrolla una página web para realizar el monitoreo del sistema de forma remota, también se lleva a cabo la configuración del Raspberry Pi como un servidor LAMP el cual es un conjunto de subsistemas requeridos para la creación de un servidor web, los sistemas utilizados son Linux como sistema operativo, Apache es el servidor web, MySQL el gestor de base datos y Pearl, PHP y Python como lenguajes de programación, la gran ventaja del servidor LAMP es que todos sus subsistemas son libres, además de su estabilidad por estar basado en sistemas Linux.

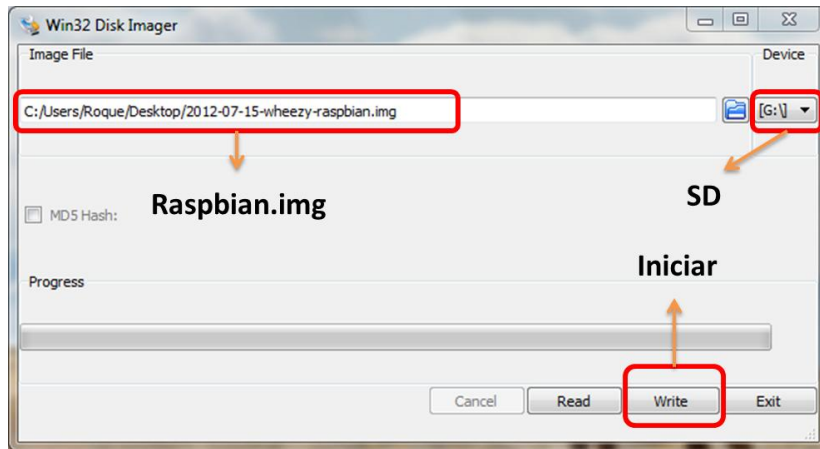
A continuación se explica con detalle el proceso que se llevo a cabo para desarrollar la capa de enlace y monitoreo.

4.3.1 Configuración Raspberry Pi

Para realizar la configuración del Raspberry Pi es necesario conectar una pantalla con entrada HDMI o RCA, un teclado, cable de red, una memoria SD de al menos 4Gb de espacio y alimentación micro usb de 5v/700 mA. El primer paso de la configuración consiste en la instalación del sistema operativo en la memoria SD, este proceso se puede llevar a cabo en un computador con sistema operativo basado en UNIX o en Windows. El primer paso es descargar la imagen de Raspbian³⁷. Raspbian es la distribución que recomienda la fundación Raspberry Pi a los principiantes, esta distribución se encuentra basa en Debian Wheezy y se puede descargar de la web de la fundación, seguidamente se descarga el software Win32Diskmaker el cual es muy intuitivo al momento de hacer la instalación, en caso de hacerlo en un sistema UNIX es necesario realizar el proceso desde el terminal, en este caso la figura 42 muestra el proceso realizado utilizando Win32Diskmaker.

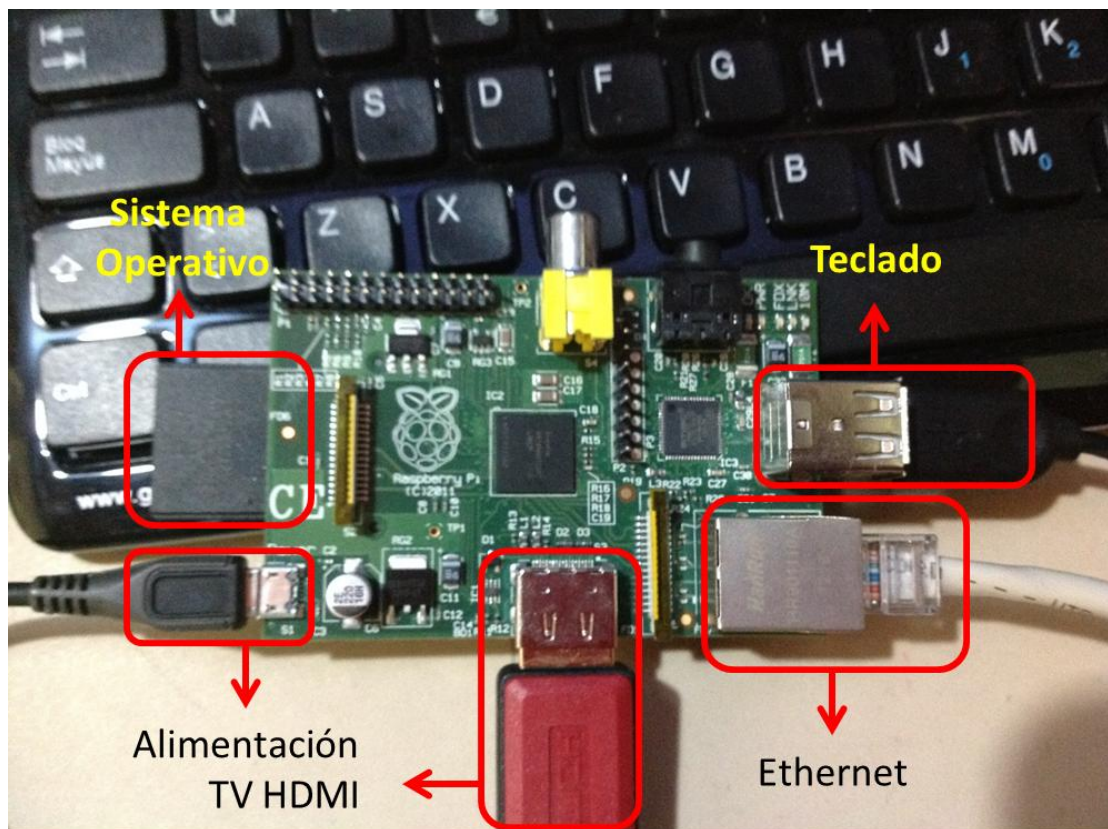
³⁷ <http://www.raspberrypi.org/downloads>, 27/08/2012, all.

Figura 42 Instalar Raspbian en SD



La figura 43 presenta el sistema operativo instalado en la memoria SD, el cual se conecta al Raspberry Pi junto a los periféricos.

Figura 43 Raspberry Pi con periféricos



La figura 44 presenta una toma de pantalla del Raspberry Pi al iniciar. El sistema operativo tiene pre configurado como usuario y contraseña lo siguiente:

- Usuario= pi
- Contraseña = raspberry

Al realizar la configuración del servidor es necesario estar logeado como super usuario ver en la figura 45, toda la instalación se puede realizar desde el terminal o también se tiene la opción de realizar la configuración SSH para acceder desde un computador Windows o Unix.

Figura 44 Raspberry Pi Inicio

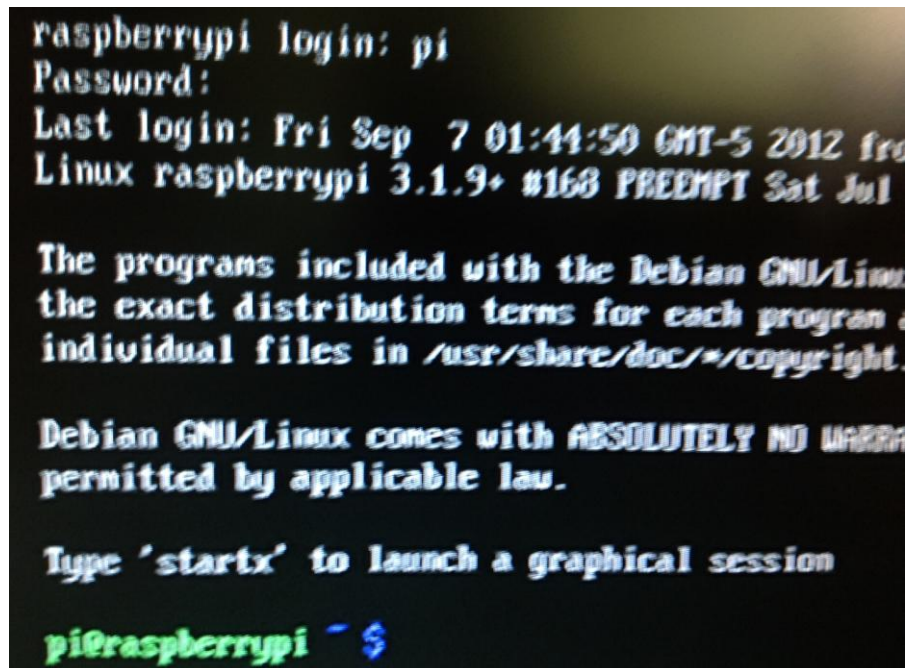
A terminal window showing the login process on a Raspberry Pi. The prompt is 'raspberrypi login: pi'. The user enters the password 'raspberry'. The terminal displays the last login information: 'Last login: Fri Sep 7 01:44:50 GMT-5 2012 from'. Below this, it shows the kernel version: 'Linux raspberrypi 3.1.9+ #168 PREEMPT Sat Jul'. There are three paragraphs of text: 'The programs included with the Debian GNU/Linux', 'the exact distribution terms for each program', and 'individual files in /usr/share/doc/*/copyright.'. The second paragraph states: 'Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY', 'permitted by applicable law.'. The third paragraph says: 'Type \'startx\' to launch a graphical session'. At the bottom, the prompt is 'pi@raspberrypi ~ \$'.

Figura 45 Raspberry Pi súper usuario

```
Debian GNU/Linux wheezy/sid raspberrypi tty1
raspberrypi login: pi
Password:
Last login: Fri Sep  7 01:44:50 GMT-5 2012 from f0-b4-79-1
Linux raspberrypi 3.1.9+ #168 PREEMPT Sat Jul 14 18:56:31

The programs included with the Debian GNU/Linux system are
the exact distribution terms for each program are describe
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the
permitted by applicable law.

Type 'startx' to launch a graphical session

pi@raspberrypi ~ $ su
Password:
root@raspberrypi:~/home/pi#
```

4.3.2 Instalación Servidor Web Apache 2.0

La figura 46 describe el proceso de instalación del Servidor Apache, el primer paso consiste en la descarga e instalación del servidor utilizando la función install de Linux desde el terminal.

Figura 46. Código para instalar apache2

```
# apt-get update
# apt-get install apache2
```

Al culminar el proceso de instalación es importante verificar la correcta instalación del servidor apache. La prueba se puede realizar desde un computador que se encuentre conectado a la misma red ingresando la dirección IP del Raspberry Pi en un navegador o desde el mismo Raspberry ingresando al navegador web e ingresando localhost como se ve en la siguiente figura 47.

Figura 47. Prueba funcionamiento apache2



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

El servidor Web Apache es de arquitectura modular, las funcionalidades mas elementales se encuentran en el modulo base, es necesario un modulo multiproceso que se encarga de gestionar las peticiones.

4.3.3 Instalación PHP5

El lenguaje PHP permite la gestión de las bases de datos, el proceso de instalación se compone de dos pasos, el primero, el cual está representado en la figura 48 y es el ingreso del código en el terminal y después de finalizar la instalación, crear un programa con la función phpinfo señalado en la figura 49 y guardarlo en la página de acceso al servidor, Finalmente se ingresa a un navegador y se accede a la IP del Raspberry Pi seguido del programa creado en este caso prueba.php, al ingresar se debe ver como resultado algo parecido a la figura 50.

Figura 48. Instalación PHP5


```
root@raspberrypi:/home/pi/# apt-get install php5
```

Figura 49. Código prueba PHP

```
<?php  
phpinfo();  
?>
```


Figura 50. Prueba PHP

192.168.2.24/prueba.php

PHP Version 5.3.3-7+squeeze8 

| | |
|--|---|
| System | Linux raspberrypi 3.1.9+ #90 Wed Apr 18 18:23:05 BST 2012 armv6l |
| Build Date | Feb 10 2012 14:43:43 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php5/apache2 |
| Loaded Configuration File | /etc/php5/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php5/apache2/conf.d |
| Additional .ini files parsed | /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/suhosin.ini |
| PHP API | 20090626 |
| PHP Extension | 20090626 |
| Zend Extension | 220090626 |
| Zend Extension Build | API220090626,NTS |
| PHP Extension Build | API20090626,NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | disabled |
| IPv6 Support | enabled |
| Registered PHP Streams | https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip |
| Registered Stream Socket Transports | tcp, udp, unix, udg, ssl, sslv3, sslv2, tls |
| Registered Stream Filters | zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk |

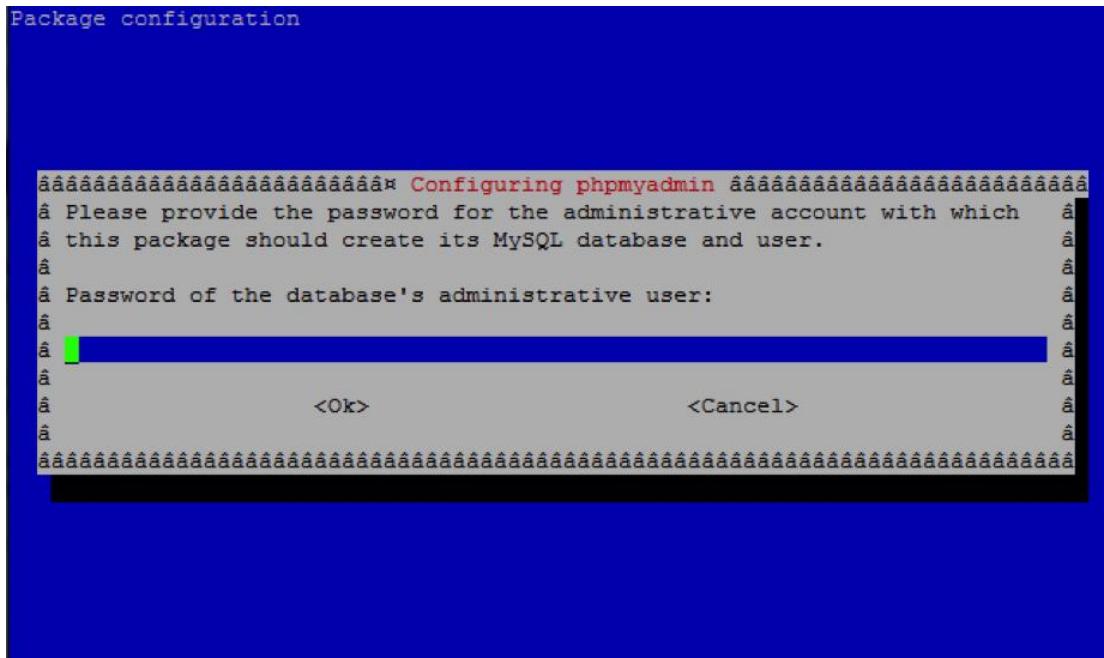
4.3.4 Instalación MySQL

El gestor de bases de datos se instala ingresando el código de la figura 51 en el terminal, en el proceso de instalación se pide crear una contraseña para el usuario que gestiona la base de datos apreciado en la figura 52.

Figura 51. Código instalación MySql

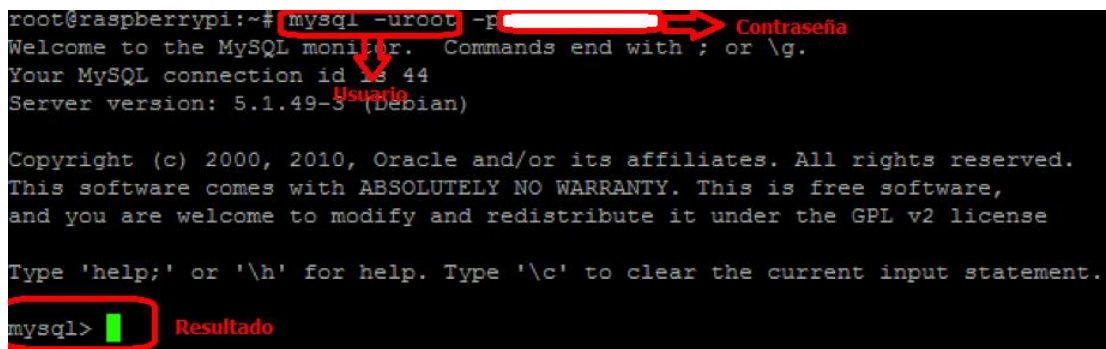
```
root@raspberrypi:/home/pi/# apt-get install mysql-server
mysql-client php5-mysql
```


Figura 52. Contraseña MySql



Al final de la instalación se debe reiniciar el sistema. iniciado el sistema y estando logueado como super usuario se procede a probar que el MySql se encuentra funcionando, ingresando con la clave creada, dicho proceso es presentado en la figura 53.

Figura 53. Accediendo a MySql



4.3.5 Configuración acceso remoto

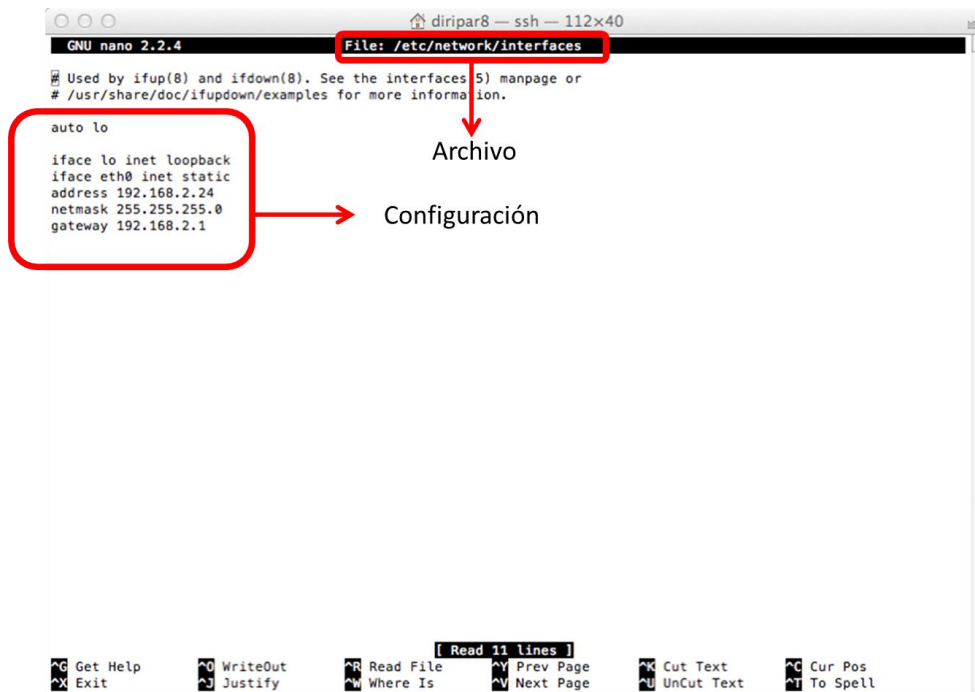
Para acceder al servidor de forma remota se necesita cumplir con tres requisitos, el primero de ellos, realizar la apertura del puerto 80 del router al cual se conecta el servidor, este proceso lo realiza el operador de internet o se puede realizar accediendo a la configuración del router, en este caso el proceso es realizado por el

operador de internet, el puerto 80 es utilizado para acceso HTTP por lo tanto es necesario que este abierto. El segundo paso consiste en asignar una IP fija al Raspberry Pi para que las peticiones que se hagan de forma remota se redirijan al dispositivo deseado y el paso final es la creación de un host de acceso al servidor desde cualquier lugar con conexión a internet.

Puerto 80: Al realizar la apertura del puerto 80 adicionalmente se agrega en el router una dirección IP fija para que todas las peticiones HTTP se reorienten a la dirección fijada, conociendo la dirección IP se debe configurar el Raspberry Pi con dicha dirección fija.

Configuración dirección IP: Para conocer la información de red del sistema se ingresa el comando `ifconfig` en el terminal y así se visualiza la dirección IP actual, Netmask, Mac del dispositivo entre otros. Para establecer la IP de forma fija se accede al siguiente Archivo nano `/etc/network/interfaces`, al estar en este archivo se remplazará la línea “`iface eth0 dhcp`” por la dirección IP obtenida en el primer paso de la configuración, en este caso el resultado presentado en la figura 54.

Figura 54. Configuración IP estática



Creación Host: El host se creará utilizando el servicio de no-ip³⁸, ya que permite de forma gratuita crear hasta 3 Host y tiene la función adicional de gestión del dominio también conocido como DNS, es crear una cuenta en la página de no-ip después de crear la cuenta se ve algo parecido a la figura 55, en donde se ingresará a la opción Add a Host, después de acceder se debe ingresar según lo observado en la figura

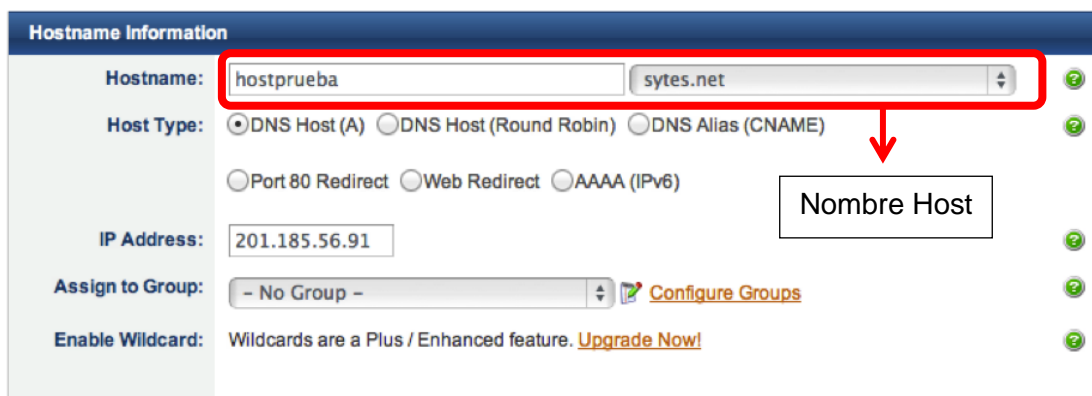
³⁸ <http://www.no-ip.com/>, 27/08/2012, all.

56, la información del Host que se va crear, finalmente en la parte inferior se pulsa Create Host.

Figura 55. Ventana inicio sesión no-ip



Figura 56. Creación host




Con esto se tiene el Host creado, el siguiente paso en la configuración es descargar el software (figura 57) para Linux que se encargara de mantener actualizado el dominio, cuando la descarga finaliza se descomprime el archivo y se procede a la instalación como se indica en la figura 58, en caso de no tener la función make preinstalada se instala con el código de la figura 59. Al finalizar la instalación se debe configurar el programa para que inicie de forma automática cada vez q reinicia, para esto se crea un script con el código que se encuentra en el archivo leer primero dentro de la carpeta descargada como se ve en la figura 60 y posteriormente el código del script en la figura 61.

Figura 57. Software a descargar de no-ip

Dynamic DNS Update Client for Linux

Keep your current IP address in sync with your No-IP host or domain with our Dynamic Update Client (DUC). Our dynamic DNS update client continually checks for IP address changes in the background and automatically updates the DNS at No-IP whenever it changes.

Ubuntu users: You may install this with the apt-get command, see this [guide](#).



Download 2.1.9
Last modified - 2008-11-21

Linux/BSD/Unix

Software

Figura 58. Instalación de no-ip

```
root@raspberrypi:/home/pi/# cd noip-2.1.9-1
root@raspberrypi:/home/pi/# make install noip2.c
```

Figura 59. Instalación función make

```
root@raspberrypi:/home/pi/# apt-get install make
root@raspberrypi:/home/pi/# apt-get install gcc
```

Figura 60. Archivo leer primero

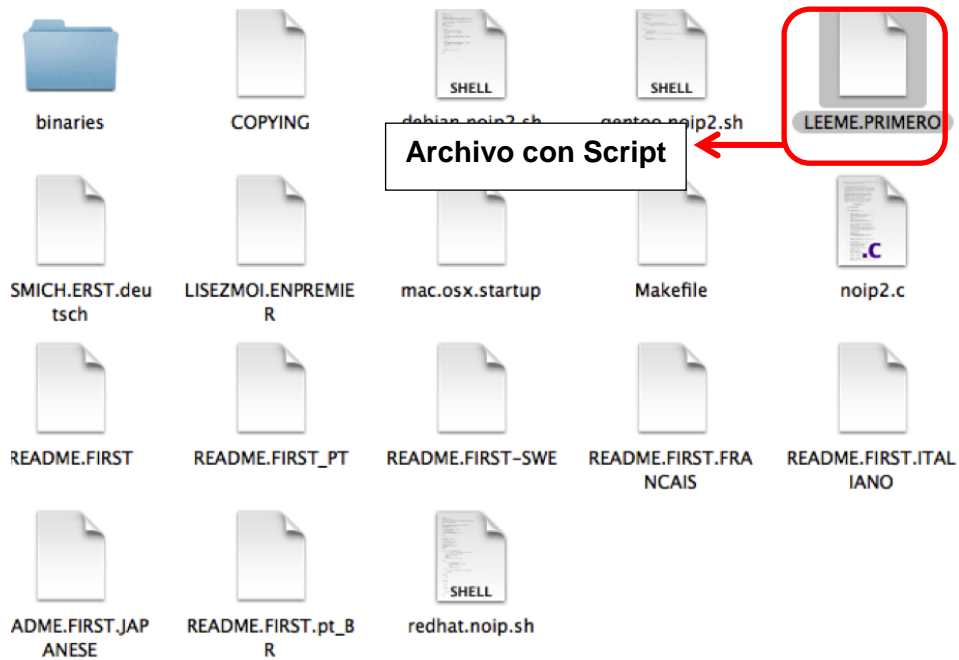


Figura 61. Código script noip.sh

```
#####
#!/bin/sh
#./etc/rc.d/init.d/functions #descomente/modifique para su killproc
case "$1" in
  start)
    echo "Iniciando noip2..."
    /usr/local/bin/noip2
    ;;
  stop)
    echo -n "Apagando noip2..."
    killproc -TERM /usr/local/bin/noip2
    ;;
  *)
    echo "Uso: $0 {start|stop}"
    exit 1
esac
exit 0
#####
```

Crear Script con este código y guardarlo Como noip.sh

El archivo creado se guarda en la carpeta Home del Raspberry Pi llamándolo noip.sh, después de esto se le da permiso de ejecución al archivo con el código de la figura 62, el archivo con permisos de ejecución se copia a la carpeta init.d del sistema esto se realiza ingresando el código de la figura 63, con el archivo en la

carpeta init.d, se debe ingresar a la misma utilizando el código de la figura 64 dentro de la carpeta se realiza la configuración del script con una prioridad baja de ejecución utilizando el código de la figura 65, al terminar los pasos indicados anteriormente se inicia el cliente con el código de la figura 66 y posteriormente se reinicia el sistema dando con esto por finalizado la configuración del servidor.

Figura 62. Permisos de ejecución al script

```
$ chmod +x noip.sh
```

Figura 63. Copiando el script en la carpeta init.d

```
# cp noip.sh /etc/init.d/
```

Figura 64. Ingresar a la carpeta init.d

```
# cd /etc/init.d/
```

Figura 65. Prioridad de ejecución 18

```
# update-rc.d noip.sh defaults 18
```

Figura 66. Iniciar el cliente noip.sh

```
$ sudo /etc/init.d/noip.sh start
```

4.3.6 Base de datos

La base de datos se desarrolló utilizando el software MYSQL³⁹, conformada por cuatro tablas: control, ultrasónicos, GPS y dirección. La tabla control (figura 67) se utilizó para actualizar el estado de control que se aplica al robot (adelante, atrás, derecha, izquierda), también se utilizó para almacenar el dato que finaliza la transmisión desde el microcontrolador. En la tabla de ultrasónicos (figura 68) se almacena la información captada por los sensores ultrasónicos, en la tabla GPS (figura 69) se guarda la latitud y la longitud que marca el dispositivo al realizar su conexión y en la

³⁹ <http://dev.mysql.com/doc/refman/5.0/es/tutorial.html>, 28/08/2012, all.

tabla dirección (figura 70) se almacena el valor de la brújula. Las tablas son actualizadas desde el Arduino Uno y desde el iPad .

Figura 67. Tabla control

| estado | msn | refe |
|--------|-----|------|
| 6 | 8 | 1 |
| | | |
| | | |
| | | |

Figura 68. Tabla ultrasónicos

| s1 | s2 | s3 | s4 | s5 | refe |
|----|----|----|----|----|------|
| 10 | 20 | 15 | 24 | 35 | 1 |
| | | | | | |
| | | | | | |
| | | | | | |

Figura 69. Tabla gps

| latitud | longitud | refe |
|---------|----------|------|
| 7.1 | -73.12 | 1 |
| | | |
| | | |
| | | |

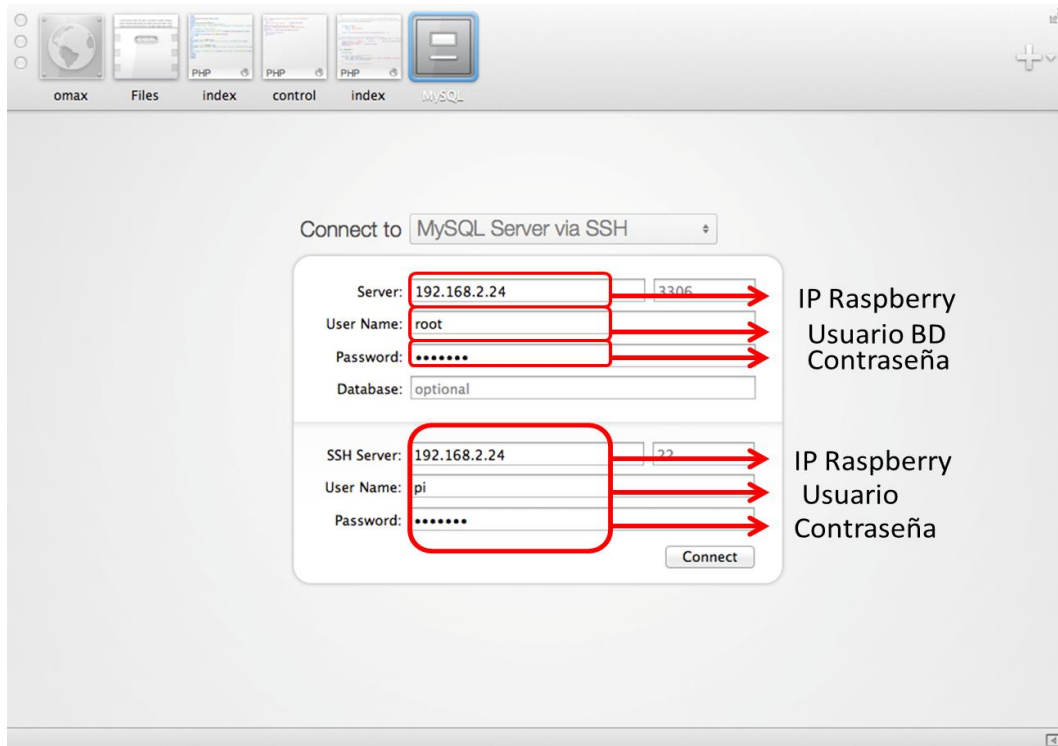
Figura 70. Tabla brújula

| lectura | refe |
|---------|------|
| 100 | 1 |

La creación de la base de datos se llevó a cabo utilizando un gestor de bases de datos, de esta forma no se requiere código para la creación de las tablas. El gestor utilizado en este caso es Coda 2⁴⁰, el proceso para desarrollar una base de datos utilizando Coda 2 se describe a continuación.

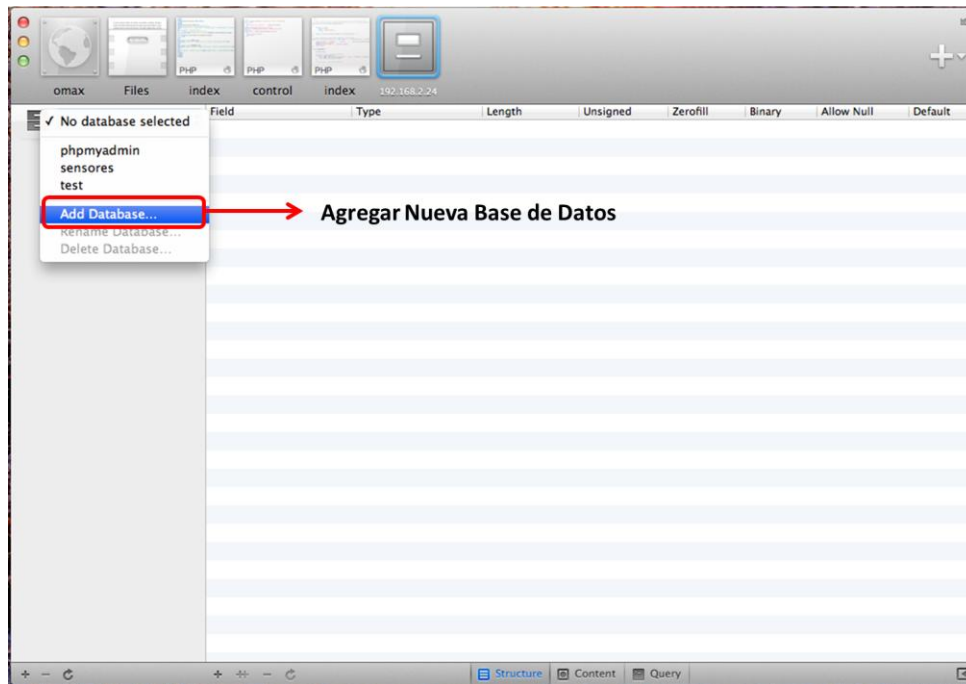
En la figura 71 se inició sesión en la base de datos desde Coda, se ingresó el usuario de la base de datos y el usuario del Raspberry. Después del inicio se procedió a la creación de la base de datos observada en la figura 72.

Figura 71. Iniciar sesión MySQL



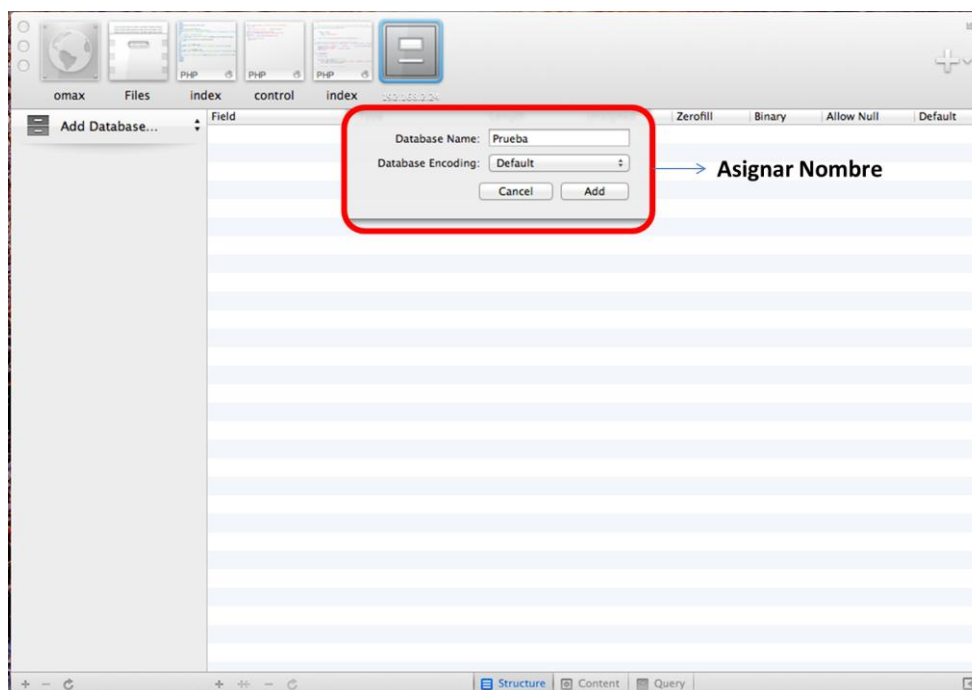
⁴⁰ <http://panic.com/coda/>, 27/08/2012, all.

Figura 72. Crear base de datos



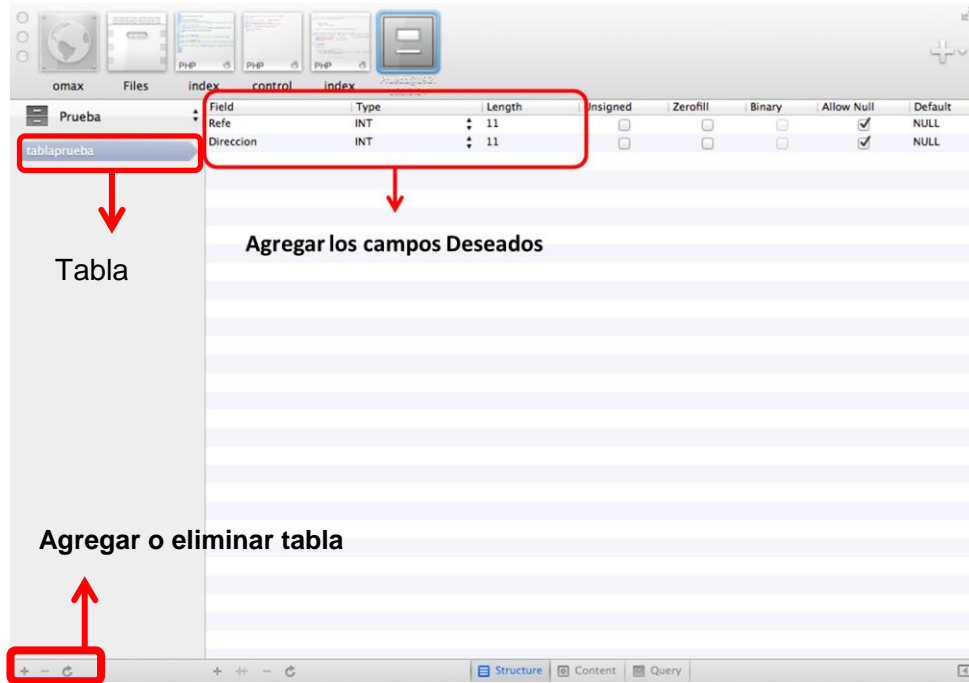
Para este caso se creó una base de datos de prueba representada en la figura 73.

Figura 73. Base de datos prueba



Después de crear la base de datos se agregó una tabla utilizando el símbolo + representado en la figura 74, en la parte izquierda se encuentra la tabla agregada y en la derecha se agregan los campos de la misma forma que se crea una tabla.

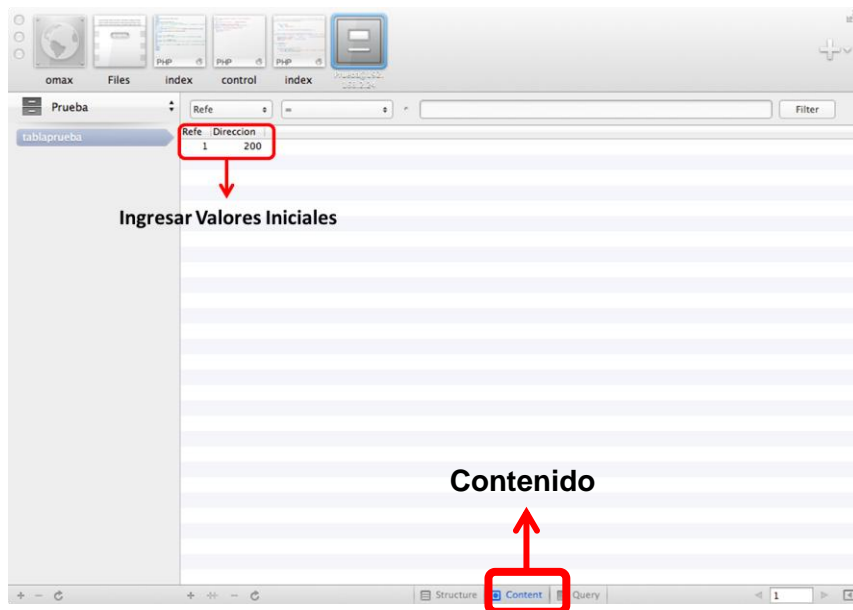
Figura 74. Agregar tablas y campos



Finalmente se ingresaron los valores iniciales a los campos como se indica en la figura 75, este proceso se llevó a cabo de la misma manera que en el paso anterior al insertar una tabla, se debió ingresar al contenido y luego con el símbolo + se creó el campo de ingreso de un valor inicial o de referencia.

Los campos después de ser inicializados se pueden actualizar desde el cliente, en este caso el iPad o el robot.

Figura 75. Inicializar campos



4.3.7 Software Servidor

La instalación y configuración del servidor se explicó de forma detallada previamente, también se detalló como configurar la base de datos y como crearla y se realizó la instalación de PHP5 lenguaje con el que se realizó la gestión de la base de datos. Teniendo todas las herramientas necesarias se procedió con el desarrollo de la aplicación encargada del enlace del iPad con el Robot y de la web de monitoreo.

Se desarrollaron tres aplicaciones en el servidor, la primera se encarga de la recepción de información proveniente del Robot, funciona al mismo tiempo como web de monitoreo, la segunda tiene como función la gestión de la información que se recibe y envía al iPad y la tercera aplicación se utiliza para enviar la información de control al Robot.

Cada aplicación es un archivo de tipo .php que se almacena en la carpeta raíz del servidor, el archivo index.php corresponde a la aplicación principal, cuando se accede al servidor desde un navegador se carga en el mismo la información que se encuentra en este archivo.

A continuación se muestra un sencillo ejemplo de cómo recibir información proveniente del robot y almacenarla en la base de datos. Como se dijo anteriormente los datos se envían desde el robot utilizando la función GET, el servidor la recibe de la misma manera y la almacena en una variable local que se luego se envía a la base de datos, como se señala en la figura 76, se almacena recibe la variable grados que se transmite desde el robot, esta variable corresponde a la lectura de la brújula, después de ser almacenado el valor se conecta a la base

de datos y se actualiza la tabla dirección, el campo estado en donde la referencia sea igual a uno.

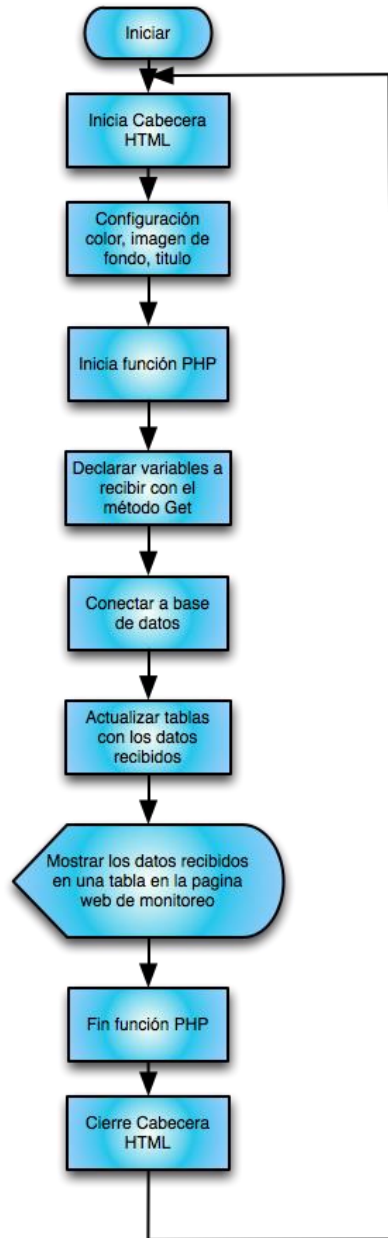
Figura 76. Actualizar base de datos desde el Robot.

```
<?php
$grados=$_GET['grados'];
$con = mysql_connect("localhost","root","Contraseña");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("sensores", $con);
mysql_query("UPDATE direccion SET lectura=$grados WHERE refe='1'");

?>
```

En la figura 77 se puede observar el flujo de información de esta aplicación, en donde se actualizan las tablas de la base de datos y se procede a visualizar la información en la Web de monitoreo.

Figura 77. Diagrama de Flujo Programa Servidor No. 1



En la segunda aplicación se recibe la información enviada desde el iPad utilizando el método Post de la figura 78, se actualiza la tabla control. Desde el iPad se realizan diferentes peticiones por esto se utiliza un condicional en el cual se pregunta en este ejemplo por la variable refe, si la condición se cumple se almacena la variable code en una variable local para posteriormente actualizar la base de datos, para finalizar se envía una respuesta al iPad con la variable code utilizando la función json_encode de PHP que codifica el mensaje en Json que es la forma como el iPad recibe los datos.

Figura 78. Recepción y envío de información al iPad

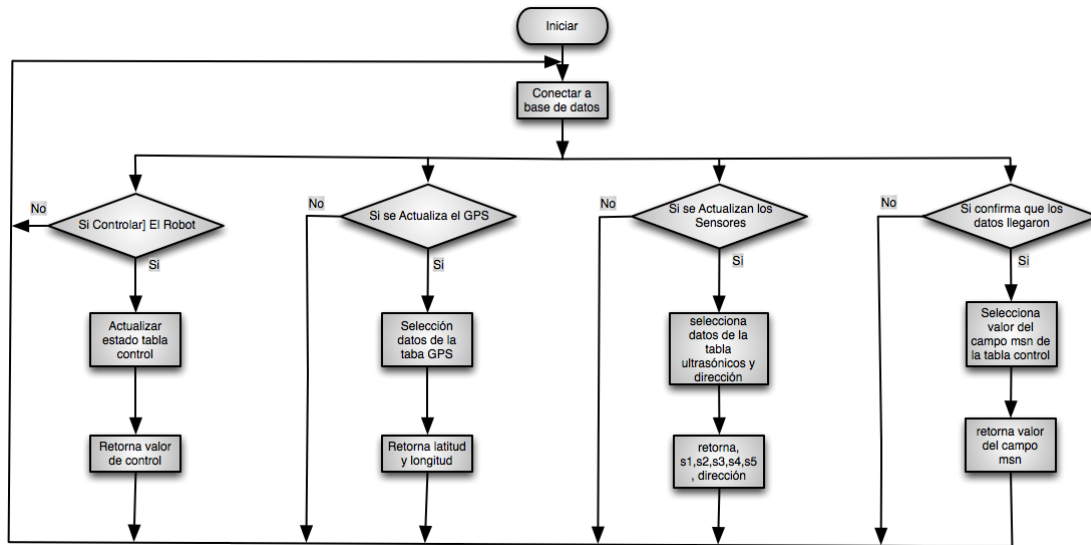
```

if (isset($_POST["refe"])) {

    $state = $_POST["refe"];
    $code = $_POST["code"];
    $this->db->query("UPDATE control SET estado=$code WHERE
refe='1'");
    $this->db->commit();
    $result = array("code" => $code,);
    sendResponse(200, json_encode($result));
    return true;
}
    
```

En la figura 79 se observan las diferentes peticiones a las que responde la aplicación cuando se el iPad lo requiere.

Figura 79. Diagrama de Flujo Programa Servidor No. 2

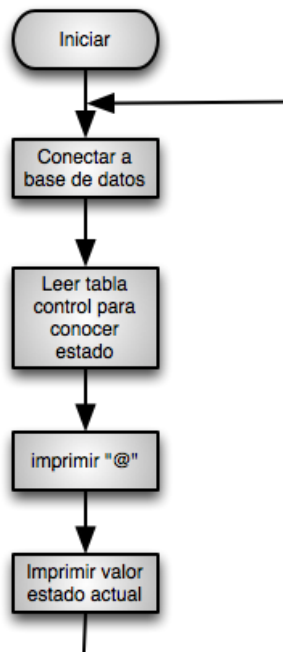


Finalmente la tercera aplicación cumple con el objetivo de imprimir en pantalla el valor de control, en la figura 80 y figura 81 ve el proceso con el cual se imprime una “@” y después se accede a la tabla control para imprimir el valor actual de control que lee el Robot constantemente.

Figura 80. Datos de control

```
<?php
$con = mysql_connect("localhost","monty","Cel1234");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("sensores", $con);
$result = mysql_query("SELECT * FROM control");
echo '@';
while($row = mysql_fetch_array($result))
{
    $sal=$row['estado'];
    echo $sal;
}
?>
```

Figura 81. Diagrama de Flujo Programa Servidor No. 3



4.3.8 Web Monitoreo

La web de monitoreo se encuentra alojada en el servidor instalado en la Raspberry Pi, como se explicó anteriormente en la figura 77. Para acceder a la página web de forma remota se ingresa al host creado utilizando no-ip. En esta página se visualiza

el estado de los sensores ultrasónicos, GPS y brújula representado en la figura 84. La página de monitoreo permite acceder desde un ordenador con internet y así conocer la información de los sensores del Robot.

Para controlar el acceso a la página, se ha adaptado un sistema de seguridad en el cual se debe ingresar el nombre de usuario y contraseña para poder acceder a la información (figura 82). El sistema de seguridad se describe en el diagrama de flujo que se ve en la figura 83, en el cual si el usuario ingresa los datos de forma correcta se accede a la pagina de monitoreo en caso de ingresar la información incorrecta se muestra un mensaje de error y se pide ingresar los datos nuevamente. Estando en la pagina de monitoreo se tiene la opción de salir de forma segura de la pagina dando clic al final de la pagina esto se muestra en la figura 84.

La información que se visualiza en la web se actualiza cuando el cliente lo requiere desde el iPad o cuando se pulsa el botón actualizar que se encuentra incrustado en la página de esta forma se puede solicitar la información en el momento deseado.

Figura 82. Página monitoreo



Autenticación OMAX MONITOR

Introduce tu nombre de usuario y contraseña

Nombre de usuario:

Contraseña:

Ingresar Datos

Figura 83. Diagrama de seguridad

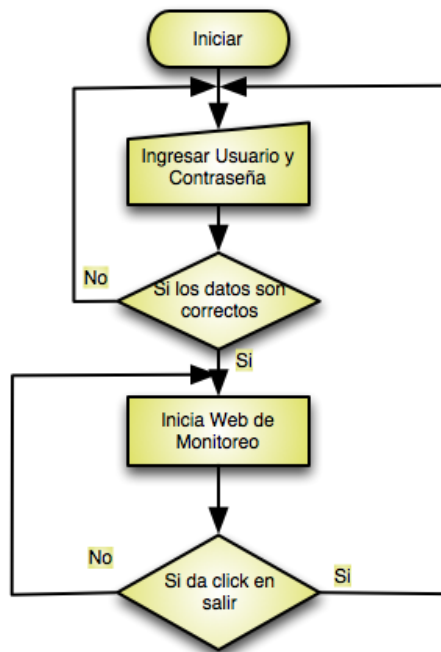


Figura 84. Página monitoreo



4.4 Capa de Control

La capa de control se desarrolló en el iPad con el fin de crear una aplicación amigable con el usuario. Se creó una aplicación con 4 ventanas en las que se encuentra el control remoto en la primera ventana, en la segunda ventana se visualiza un mapa con la posición actual del robot, en la tercera los datos de la brújula y los sensores ultrasónicos y finalmente la ventana con el control local, el proceso de desarrollo se explicara con todo detalle a continuación.

4.4.1 Primera aplicación

La primera aplicación se desarrolló para llevar a cabo el primer contacto con el servidor. El primer paso consiste en crear un nuevo proyecto en Xcode. En la ventana principal de Xcode se debe seleccionar "create a new Xcode Project" como se ve en la figura 85, al realizar la selección aparece en pantalla una nueva ventana mostrando las plantillas que ofrece Apple para crear una aplicación, en esta venta se selecciona Single View Application que es el tipo de aplicación básica como se ve en la figura 86, posteriormente se debe indicar el nombre de la aplicación, nombre de la clase y el dispositivo a utilizar (figura 87) con esto se finaliza la creación del proyecto y se procese a agregar las librerías.

Figura 85. Nuevo proyecto Xcode

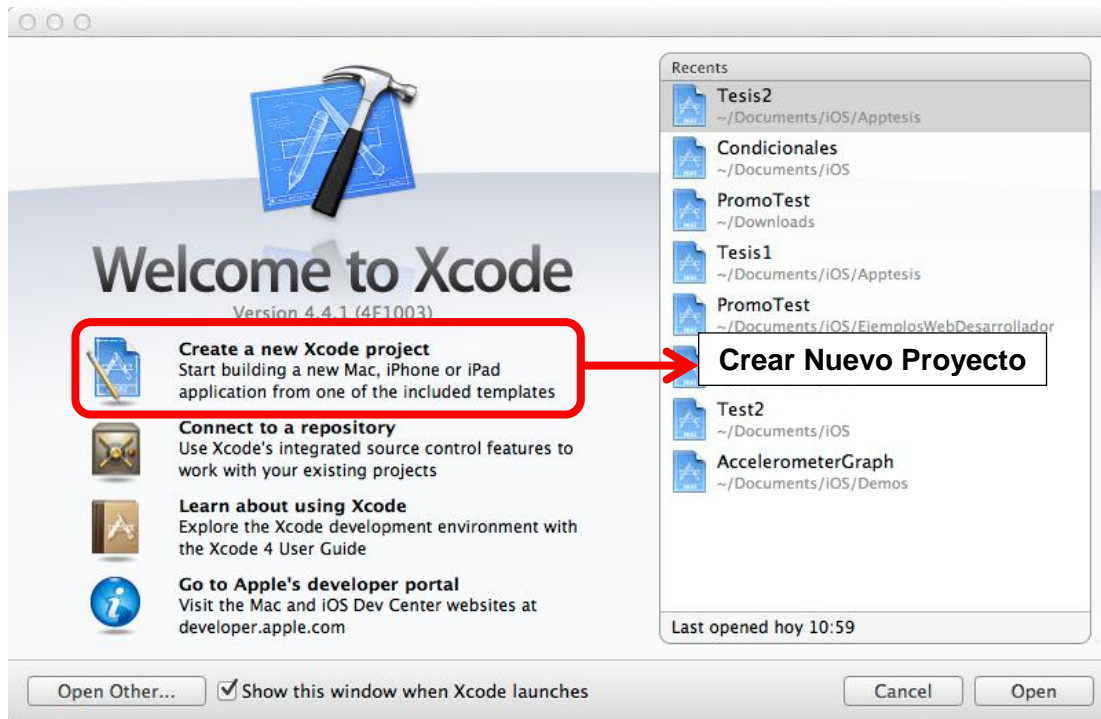


Figura 86. Selección plantilla

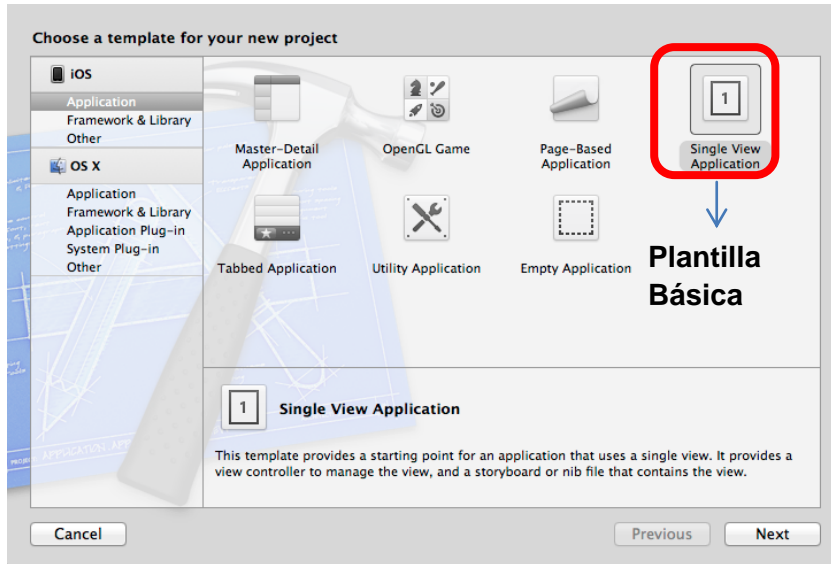
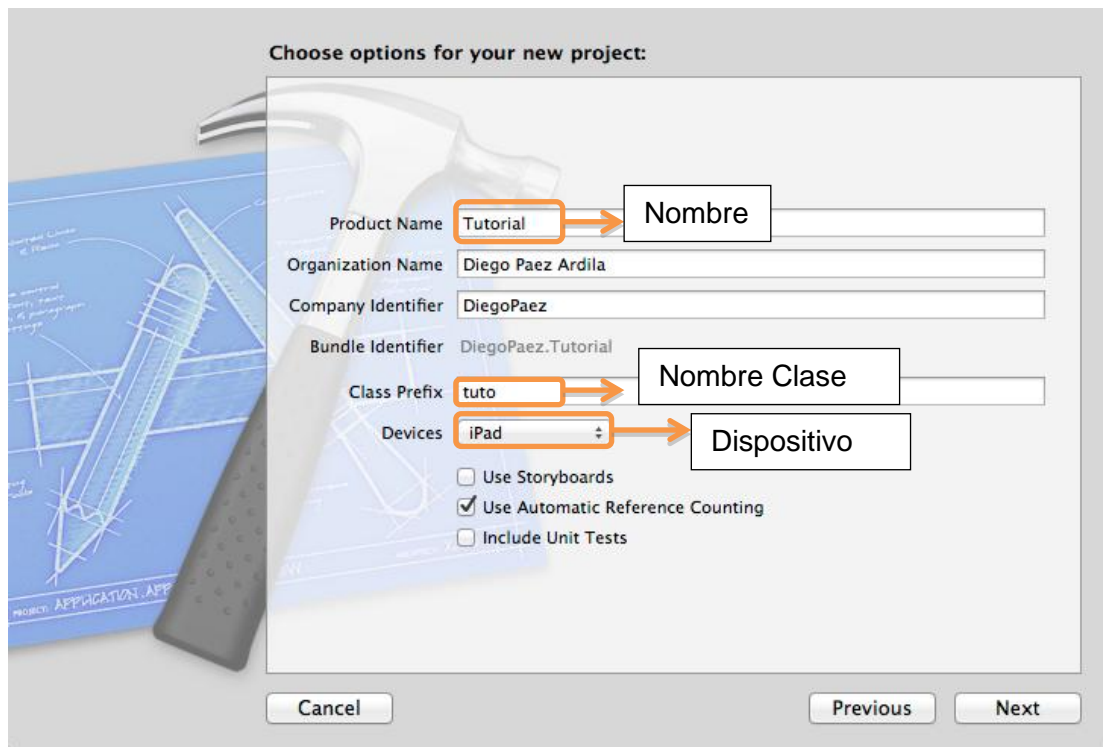


Figura 87. Nombre aplicación



Es necesario agregar las librerías que facilitan la interacción con el servidor, estas librerías son: SBjson, ASIHTTPRequest y opcionalmente MBProgressHUD, se pueden descargar del siguiente enlace (<https://docs.google.com/open?id=0B8e1GtzgrD-IMk14ZGRFekd3bWc>), después de descargar las tres librerías se seleccionan y arrastran a la cabecera de la aplicación creada como se ve en la figura 88, al

agregarlas se ve una ventana que indica si se desea agregar los archivos de las librerías a la carpeta del proyecto.

Con las librerías instalas en el proyecto se debe realizar una configuración extra a la librería ASIHTTPRequest como se ve en la figura 89, se agrega una bandera que indica al sistema que la librería no es compatible con ARC. Se deben agregar los Frameworks, que son librerías de Apple que incluyen funciones extras a la aplicación, para esta se incluyen los siguientes: CFNetwork, SystemConfiguration, MobileCoreServices, libz.1.2.5.dylib como se ve en la figura 90.

Finalmente se debe compilar la aplicación y no se debe presentar ninguna advertencia ni error.

Figura 88. Agregar librerías



Figura 89. Bandera ARC

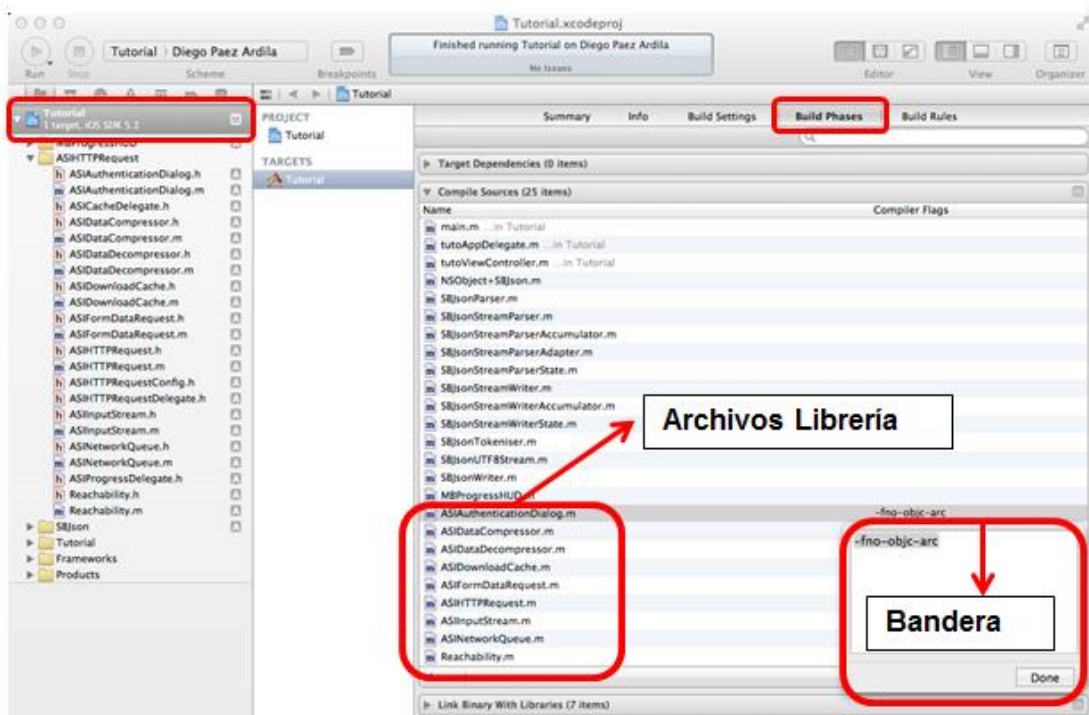
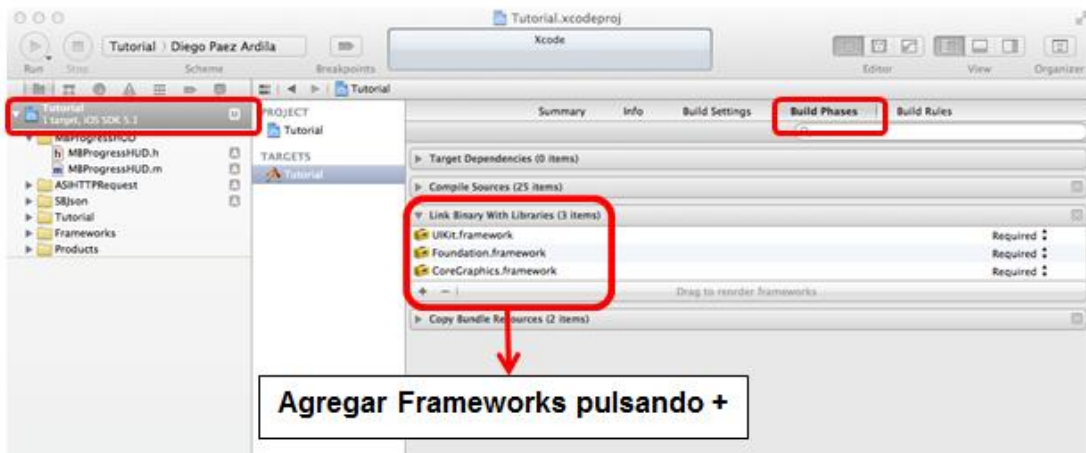
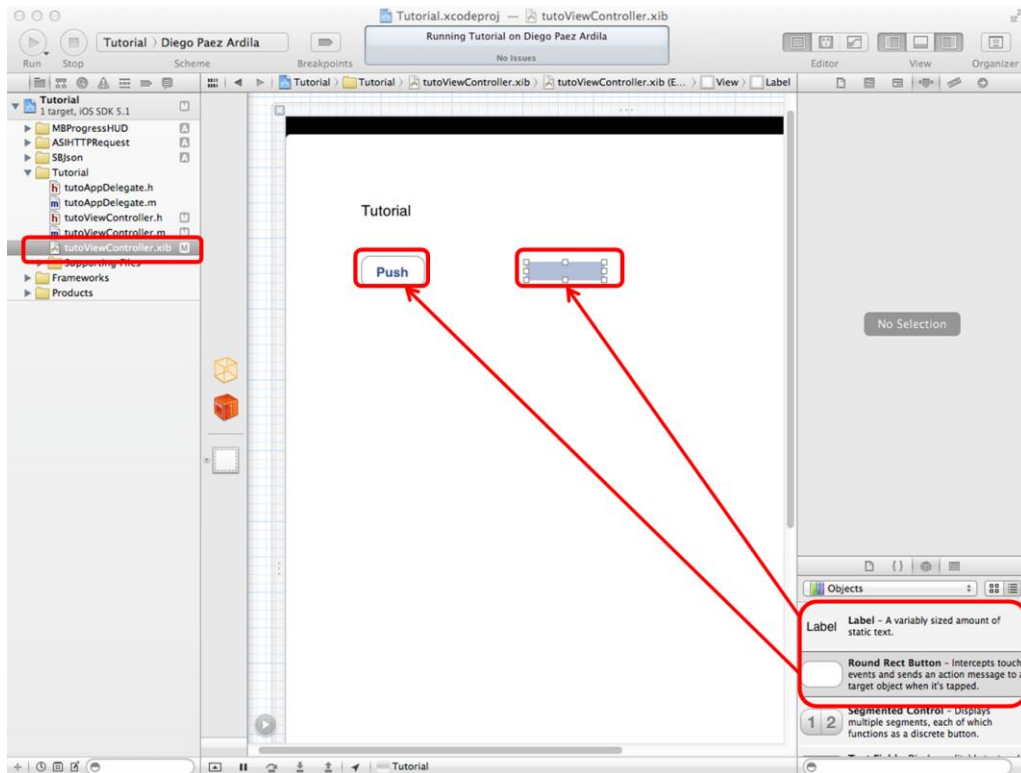


Figura 90. Agregar Frameworks



La primera aplicación tendrá en su interfaz un botón y un label, el botón realizará la acción de enviar datos al servidor y en el label se visualizará el dato que retorna desde el servidor. Para agregar estos objetos a la interfaz se selecciona el archivo finalizado en .xib en la carpeta que recibe el nombre del proyecto en este caso Tutorial, al seleccionar el archivo indicado se ve la interfaz grafica del iPad y para posteriormente agregar el botón y el label como se ve en la figura 91.

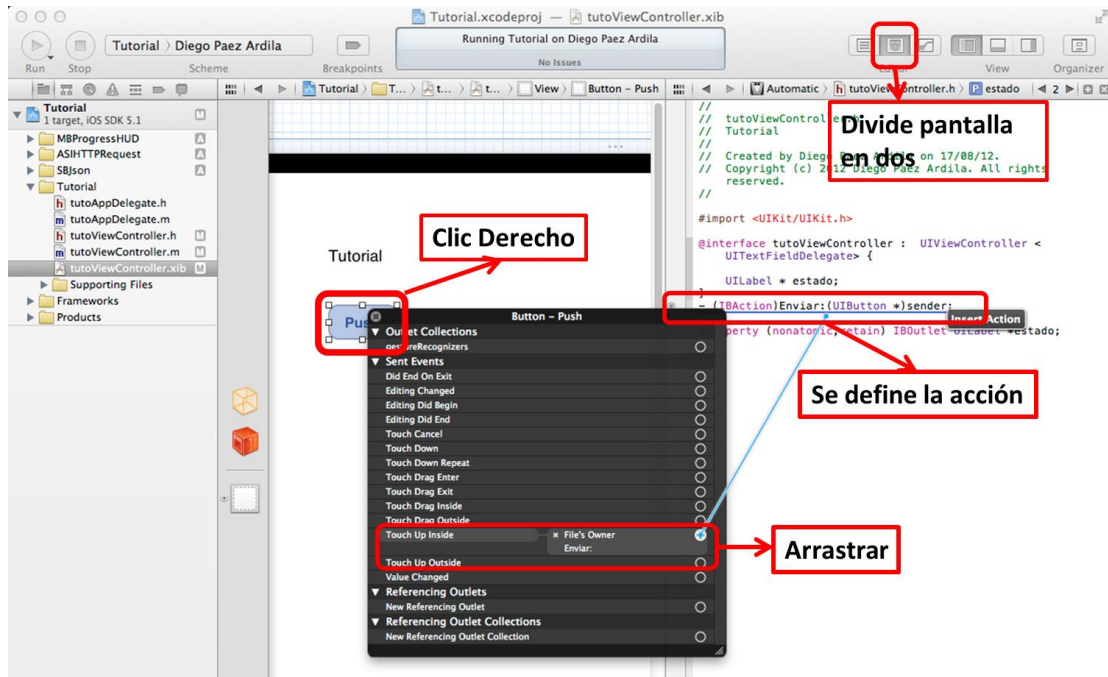
Figura 91. Interfaz primera aplicación



Después de tener los botones en la interfaz se debe crear un vínculo entre la parte táctil y la acción que van a ejercer, para realizarlo de una forma sencilla se divide la pantalla de Xcode en dos seleccionando el icono que de central de la parte superior derecha como se ve en la figura 92, cuando la pantalla se divide en dos en la parte derecha se tiene el archivo TutoViewController.h que es donde se agregan todos los objetos de la interfaz.

Un botón realiza una acción al ser tocado, se denomina `-(IBAction)`, un label es un indicador y se denomina `-(IBOutlet)`, en la figura 92 se agrega el `-(IBAction)` al TutoViewController.h, se selecciona de los Sent Events Touch up inside y se arrastra a la venta de la derecha, de la misma forma se hace con el label la diferencia consiste en que es un `-(IBOutlet)` esto se ve en la figura 93.

Figura 92. Dividir Pantalla y agregar IBAction



El archivo en el cual se implementan los objetos agregados a la interfaz y se asignan las acciones es TutoviewController.m al seleccionar el archivo con el modo pantalla dividida se tienen los dos archivos importantes de esta aplicación TutoviewController.h y TutoviewController.m.

En el archivo TutoviewController.m se agregan las librerías que se añadieron al programa anteriormente, en la figura 94 se ven las librerías en la cabecera del programa después de esto se procede a agregar la acción que ejecutara el botón declarado anteriormente, en la figura 95 se ve el código de la acción, adicionalmente se agrega una función que se ejecuta en el momento que recibe la información proveniente desde el servidor (ver figura 96), esta función decodifica el vector Json enviado desde el servidor y lo almacena en una variable local que se muestra directamente en el label incluido anteriormente.

Figura 93. Agregar IBOutlet

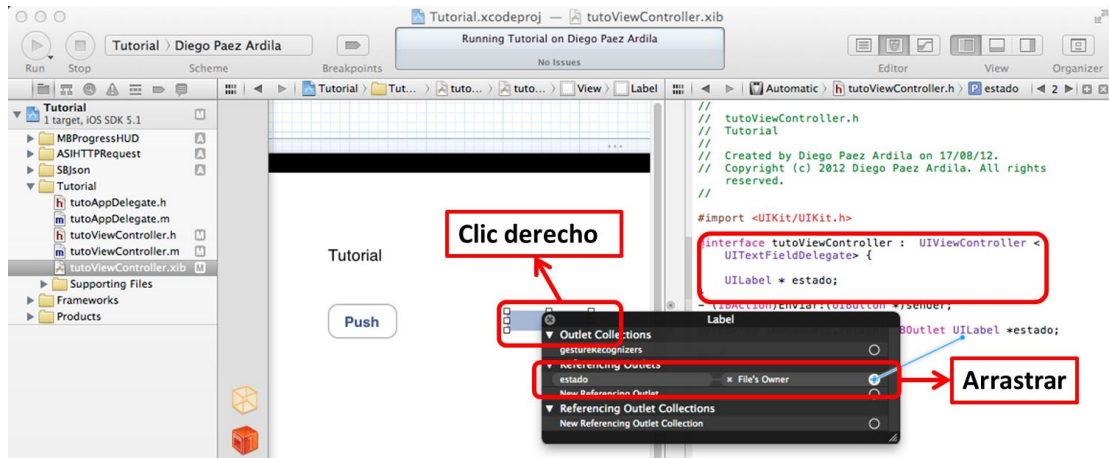


Figura 94. Agregar librerías

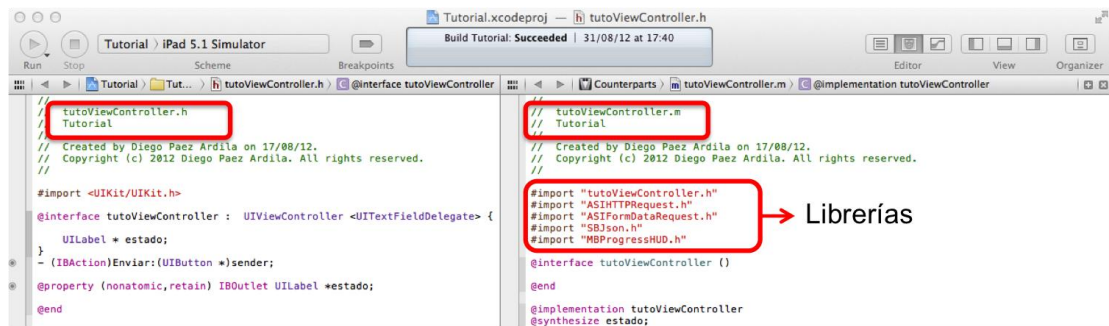


Figura 95. Acción botón

```
//Este boton envia dos datos utilizando el metodo post al servidor
- (IBAction)Enviar:(UIButton *)sender {
    //url del servidor
    // NSURL *url = [NSURL URLWithString:@"http://omax.sytes.net/promos/"]; //remoto
    NSURL *url = [NSURL URLWithString:@"http://192.168.2.24/promos/"]; //local
    ASIFormDataRequest *request = [ASIFormDataRequest requestWithURL:url];
    [request setPostValue:@"3" forKey:@"refe"];
    [request setPostValue:@"OK-Funciona" forKey:@"code"];
    [request setDelegate:self];
    [request startAsynchronous];
    // Clear text field
    estado.text = @"";

    // Start hud
    MBProgressHUD *hud = [MBProgressHUD showHUDAddedTo:self.view animated:YES];
    hud.labelText = @"Enviando...";
}
```


Figura 96. Acción recepción

```
//Funcion para recibir los datos que envia el servidor
- (void)requestFinished:(ASIHTTPRequest *)request
{
    [MBProgressHUD hideHUDForView:self.view animated:YES];
    if (request.responseStatusCode == 200) {
        NSString *responseString = [request responseString];
        NSDictionary *responseDict = [responseString JSONValue];
        NSString *Code = [responseDict objectForKey:@"code"];
        estado.text = Code; //Visualizar en label resultado
    }
}
```

El resultado que se obtiene de la primera aplicación se ve en la figura 97.

Figura 97. Resultado aplicación

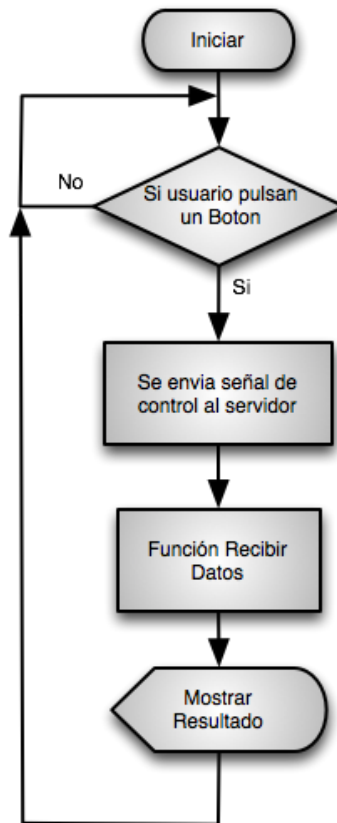


4.4.2 Diagrama de flujo aplicación

En el numeral anterior se explicó de forma detallada como realizar el primer contacto con el servidor a continuación se encuentra el diagrama de flujo que representa lo que ocurre cuando se pulsa un botón en la aplicación (figura 98).

Al iniciar la aplicación el programa se mantiene a la espera de que un botón sea pulsado, cuando el botón se pulsa se ejecuta la acción de envío de la información al servidor, luego la función de recepción entra en acción para recibir el mensaje de vuelta y finalmente mostrarlo en pantalla.

Figura 98. Diagrama aplicación



4.4.3 Control Punto a Punto

En la sección Capa Interna Robot se da a conocer la forma como se debe crear una red ad-hoc, también se explico como se recibe un dato utilizando este tipo de conexión para realizar el control del Robot. Para enviar los datos desde el iPad se utiliza el objeto WebViewController que cumple las funciones de un navegador dentro de la aplicación de tal forma que se puede enviar un url con la notación explicada en Capa Interna Robot, en la siguiente figura 99 se puede ver como enviar un dato, es importante resaltar que las acciones de los botones se encuentran relacionadas al WebViewController por lo tanto no es necesario declararlos en el controlador de interfaz.

Figura 99. Enviar dato directo

```
- (IBAction)up:(id)sender {
    NSURL *url = [NSURL URLWithString:@"http://169.254.1.1/$1"];
    NSURLRequest *req = [NSURLRequest requestWithURL:url];
    [myWebView loadRequest:req];
}

- (IBAction)down:(id)sender {
    NSURL *url = [NSURL URLWithString:@"http://169.254.1.1/$2"];
    NSURLRequest *req = [NSURLRequest requestWithURL:url];
    [myWebView loadRequest:req];
}
```

Enviar Dato

4.4.4 Utilizando Mapas

En la capa interna robot se habla de la implementación de un dispositivo de posicionamiento global, la mejor forma para visualizar este tipo de información es la implementación de los mapas, de tal forma que se pueda observar con claridad la posición en la cual se encuentra el robot. MapKit es una herramienta basada en la API de google maps y google earth y proporciona toda la información necesaria para el uso de los mapas en su Framework. El elemento utilizado en la aplicación es la clase MMapView la cual proporciona un lienzo para visualizar un mapa con la información satelital. La información puede ser presentada en el mapa de forma hibrida o satelital como se ve en la figura 100, la región geográfica se puede aumentar o disminuir manualmente utilizando el gesto de zoom natural del iPad, también se puede desplazar de la misma manera, una función adicional es la de colocar anotaciones en el mapa para resaltar los lugares visitados.

Figura 100. Ventana Mapas



El botón refrescar que se ve en la figura 100, envía una petición al servidor para que este envíe la ultima posición registrada del robot que está almacenada en la base de datos, cuando la información retorna se actualiza el mapa como se ve en la figura 101. El botón tipo de mapa realiza el cambio entre híbrido y satélite como se ve en la figura 102.

Figura 101. Actualizar Mapa

```

- (void)requestFinished:(ASIHTTPRequest *)request
{
    [MBProgressHUD hideHUDForView:self.view animated:YES];
    if (request.responseStatusCode == 200) {
        NSString *responseString = [request responseString];
        NSDictionary *responseDict = [responseString JSONValue];
        latitude = [(NSNumber *)[responseDict objectForKey:@"lati"] floatValue];
        longitud = [(NSNumber *)[responseDict objectForKey:@"lon"] floatValue];
        NSLog(@" latitude: %f, longitud: %f", latitude, longitud);

        // MAPA
        CLLocationCoordinate2D zoomLocation;
        zoomLocation.latitude = latitude;
        zoomLocation.longitude = longitud;

        MKPointAnnotation *annotationPoint = [[MKPointAnnotation alloc] init];
        annotationPoint.coordinate = zoomLocation;
        annotationPoint.title = @"Optimus Max";
        annotationPoint.subtitle = @"Robot Tesis";
        [_map addAnnotation:annotationPoint];

        MKCoordinateRegion viewRegion = MKCoordinateRegionMakeWithDistance(zoomLocation, 0.5*METERS_PER_MILE, 0.5*METERS_PER_MILE);
        MKCoordinateRegion adjustedRegion = [_map regionThatFits:viewRegion];

        [_map setRegion:adjustedRegion animated:YES];
    }
}

```

Recibe y almacena en Variables locales

Se asigna la latitud y longitud Y coloca la anotación

Figura 102. Tipo de mapa

```
    }  
    - (IBAction)Tipo:(UIBarButtonItem *)sender {  
        if ( _map.mapType == MKMapTypeStandard)  
            _map.mapType = MKMapTypeSatellite;  
        else  
            _map.mapType = MKMapTypeStandard;  
    }  
}
```

4.4.5 Aplicación con pestañas

En la aplicación final se reúnen todas las herramientas dadas a conocer previamente en la Capa de Control, estas herramientas se encuentran separadas y hasta el momento solo se ha utilizado la plantilla básica que es funcional para aplicaciones en las que se presenta al usuario una sola ventana.

La plantilla Tabbed Application se encuentra pre configurada de tal forma que se puede seleccionar desde una barra de iconos la ventana que se desea visualizar, anteriormente se explico como crear una aplicación con la plantilla Single View Application de la misma manera se accede a *Xcode* pero se debe elegir la nueva plantilla como se ve en la siguiente figura 103, en la siguiente ventana se asigna el nombre de la aplicación y de la clase principal y se selecciona la casilla Storyboard esto se ve en la figura 104, al realizar esto se guarda el proyecto en el lugar deseado y queda disponible para empezar con la aplicación.

Storyboard es una herramienta para organizar las ventanas que se utilizan en una aplicación de esta manera se puede realizar la aplicación con mayor orden y claridad, un ejemplo de Storyboard es la figura 105.

Figura 103. Selección Plantilla Tabbed

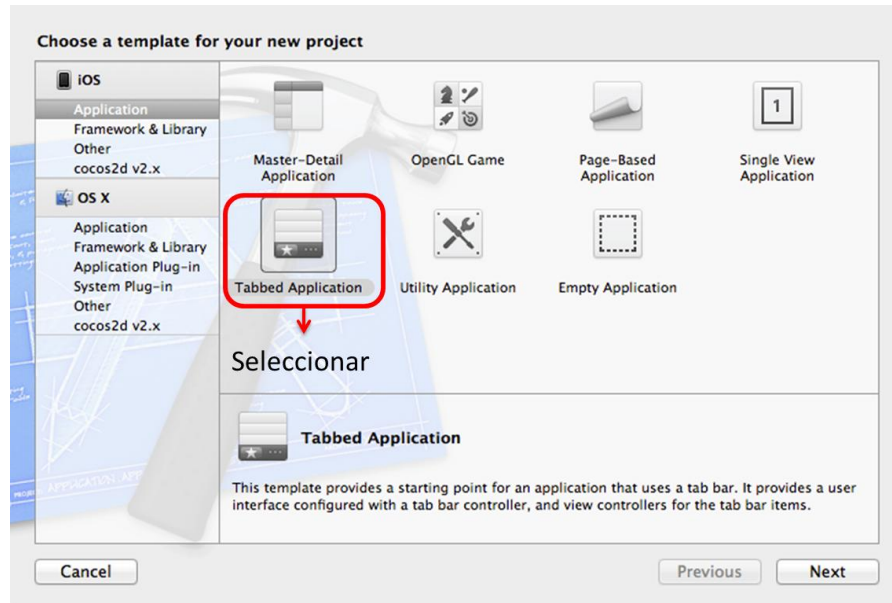


Figura 104. Selección Storyboards

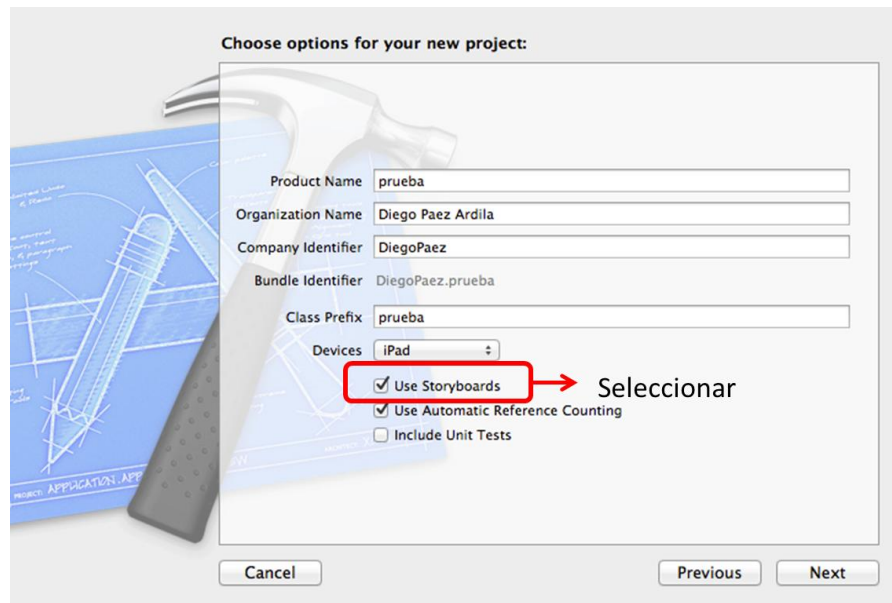
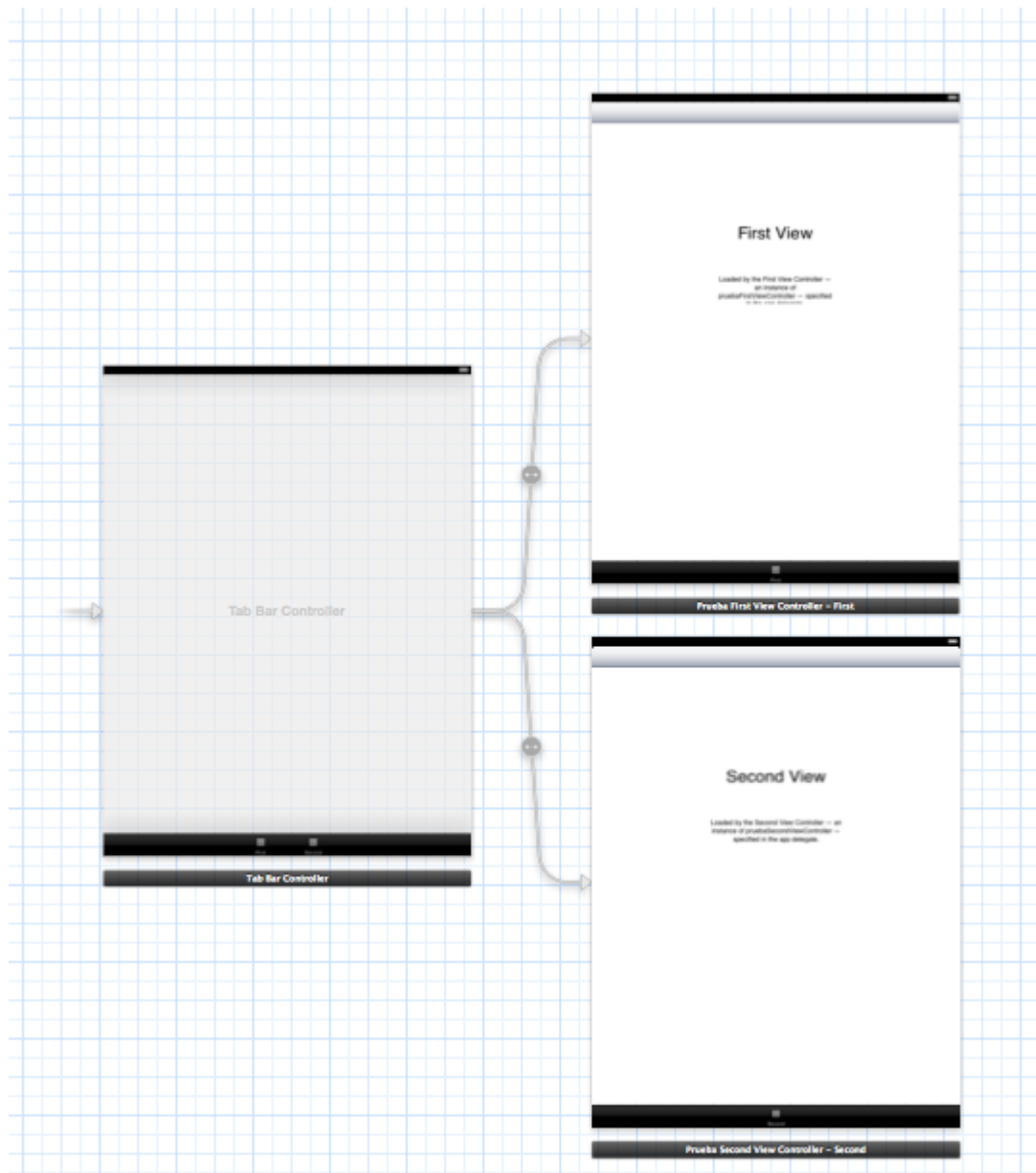


Figura 105. Storyboard



4.4.6 Aplicación Final

La aplicación final es la compilación de todas las herramientas expuestas en la Capa de Control, utilizando la plantilla Tabbed se logra una interacción natural del usuario con la aplicación.

Hasta este punto se tiene un grupo de aplicaciones con fondo blanco y botones simples algo que no es amigable ni llamativo para el usuario, por esto es importante

agregar iconos e imágenes que atraigan la atención del usuario haciendo de la aplicación más atractiva e intuitiva.

Para la selección de iconos y fondos es importante consultar las dimensiones que estos requieren para cumplir con las especificaciones necesarias ya que si se utilizan imágenes con tamaños que no vayan acorde esto no se lograra el fin deseado. En la tabla 5 se ven las dimensiones recomendadas por Apple para utilizar en los dispositivos de la empresa.

Tabla 5. Dimensiones recomendadas⁴¹

| Descripción | Tamaño para iPhone y iPod touch Retina (en pixeles) | Tamaño para iPhone y iPod touch (en pixeles) | Tamaño para iPad Retina (en pixeles) | Tamaño para iPad (en pixeles) |
|-------------------------------------|---|--|---|---|
| Icono de la aplicación | 114 x 114 | 57 x 57 | 144 x 144 | 72 x 72 |
| Icono para la App store | 1024 x 1024 | 1024 x 1024 | 1024 x 1024 | 1024 x 1024 |
| Imagen de inicio y fondo aplicación | 640 x 960 | 320 x 480 | 1536 x 2008 (portrait) 2048 x 1496 (landscape) | 768 x 1004 (portrait) 1024 x 748 (landscape) |
| icono pequeño para el Spotlight | 58 x 58 | 29 x 29 | 100 x 100 58 x 58 | 50 x 50 () 29 x 29 |
| Icono para documento | 44 x 58 | 22 x 29 | 128 x 128 640 x 640 | 64 x 64 320 x 320 |
| Icono para barra de navegacion | Aproximado 40 x 40 | Aproximado 20 x 20 | Aproximado 40 x 40 | Aproximado 20 x 20 |

Contando con la ayuda de un diseñador grafico se diseño un icono para la aplicación (figura 106), el fondo de la misma y los botones que se utilizan para el

⁴¹

<http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/IconImages/IconImages.html>, 29/08/2012, all.

control del robot se diseñó el logo de la aplicación de tal forma que sea fácil de recordar para el usuario, estas características se ven en la figura 107.

Figura 106. Icono aplicación

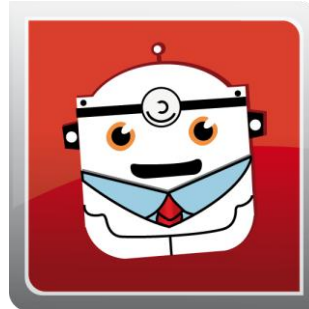
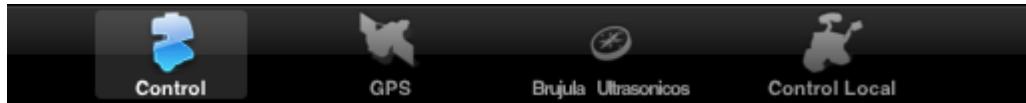


Figura 107. Diseño aplicación



En la aplicación final se incluyó la ventana de control remoto, control local, GPS y sensores como se ve en la figura 108 en la barra de ventanas final.

Figura 108. Barra de ventanas



En las figuras 109 a 112 se ve el diseño final de las ventanas de la aplicación, empezando con la ventana de control tiene 4 botones para dar dirección, uno para detener el robot y otro que realiza la solicitud para actualizar la base de datos con la información obtenida de los sensores.

Figura 109. Ventana control



La ventana de GPS esta compuesta por un mapa en el cual se visualiza un apuntador en el lugar que se encuentra el robot o en su último lugar de conexión, también cuenta con un botón para actualizar la posición y otro para cambiar el tipo de mapa.

La ventana de sensores se utiliza para visualizar las señales obtenidas de los sensores, esta compuesta por indicadores y un botón que actualiza la información después que se realiza la petición en la ventana de control.

Figura 110. Ventana GPS

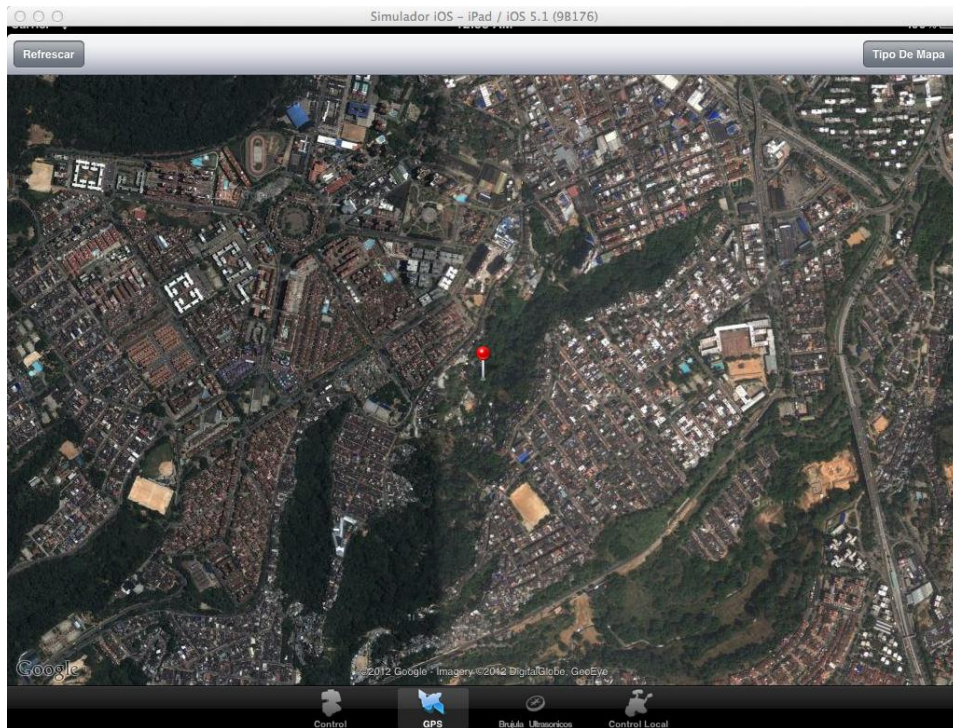


Figura 111. Ventana sensores



En la ventana de control punto a punto (figura 112) se encuentran cuatro botones que tienen como función controlar la dirección del robot, a diferencia de la ventana de control remoto cada botón envía una señal cuando esta pulsado y cuando el botón se deja de pulsar se realiza la acción de detener el robot enviado la señal correspondiente, esto se logra agregando doble funcionalidad a cada botón utilizado como se ve en la figura 113.

Figura 112. Ventana control local



Figura 113. Configuración botón dos acciones



5. CONCLUSIONES

Este proyecto permitió avanzar en la investigación que recibe el nombre de “DISEÑO Y CONTRUCCIÓN DE UN ROBOT MÓVIL CONTROLADO POR UN DISPOSITIVO DE COMPUTACIÓN MÓVIL TÁCTIL PARA LA ASISTENCIA A PERSONAS DE LA TERCERA EDAD” con número de dependencia 084-0611-2100, que se realiza actualmente en la Universidad Pontificia Bolivariana seccional Bucaramanga, financiado por la dirección general de investigaciones, en el cual se desarrollo una plataforma hardware y software para controlar un robot móvil utilizando un iPad.

Teniendo en cuenta los objetivos propuestos, se concluye que se cumplieron a cabalidad con el diseño e implementación de una aplicación propia para un dispositivo móvil tipo Tablet de la empresa Apple Inc. La aplicación tiene la función de controlar un robot de forma local utilizando una red ad-hoc y de forma remota a través de un servidor web, además de monitorear la sensórica del robot y mostrar la información en pantalla al usuario.

Se implemento un servidor web en un dispositivo de procesamiento compacto de arquitectura ARM Linux utilizando la tarjeta Raspberry Pi, que por su bajo consumo energético, pequeñas dimensiones y costo reducido hacen que de dicho dispositivo sea ideal para aplicaciones donde se requiere controlar sistemas electrónicos sin necesidad de un computador. La implementación de un servidor LAMP en dicho dispositivo de procesamiento es innovador en nuestra región y permite abrir una línea de trabajo con esta tecnología. Se diseñó e implementó una pagina web para le monitoreo de la información sensórica del robot a la cual se puede acceder desde un computador con conexión a internet.

Con este proyecto se busca fortalecer el campo de desarrollo de aplicaciones con dispositivos móviles como el iPad, ya que son herramientas que ofrecen grandes prestaciones a los usuarios y a los desarrolladores, como se ha comprobado en el desarrollo de este proyecto. Es muy importante resaltar que al llevar a cabo el desarrollo de un proyecto con un dispositivo como el iPad existe un gran potencial de usuarios finales en el mercado por ser el mismo un dispositivo de consumo masivo.

El campo de desarrollo de aplicaciones crece diariamente de forma exponencial y hasta el momento este no se había explorado en la Universidad Pontificia Bolivariana Seccional Bucaramanga; El desarrollo de aplicaciones puede ser utilizado en diferentes ramas de la ingeniería como son control, robótica, telecomunicaciones, domótica, bioingeniería entre otros campos, siendo lo mas importante tener una excelente idea para llevar a cabo.

6. BIBLIOGRAFÍA

ANDRADA, David, et al. Arquitectura para el monitoreo remoto de funciones vitales en pacientes ambulatorios. En: Congreso Argentino de Ciencias de la Computación (12: 17-21, octubre: Potrero de los Funes, Buenos Aires, Argentina). Memorias. Universidad de Moron Moron. 2006.

FILSON, Eric. Setting Up LAMP: Getting Linux, Apache, MySQL, and PHP Working Together. Primera Edición. Sybex, 2004. 448 p.

BENNETT, Gary; FISHER, Mitch y LEES, Brad. Objective-C for Absolute Beginners: iPhone, iPad and –mac programming Made Easy. Primera Edición. Apress, 2011. 338p.

CARDONA, Oscar, et al. Arquitectura para el monitoreo remoto de funciones vitales en pacientes ambulatorios. En: Congreso de Sociedad Cubana de Bioingeniería (6: 19-22, Julio: Habana, Cuba). Memorias. 2005.

COBO ORTEGA, Angel., et al. El sistema gestor de base de datos MySql En: Php y MySql: tecnología para el desarrollo de aplicaciones web. 1ra ed. España: Días de Santos, 2005. P. 339-340.

HANS, M. Robotic home assistant Care-O-bot: past-present-future. [En línea] <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1045652&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1045652> [Consultado en 15 de Abril de 2012].

KOCHAN, Stephen G. Programming in Objective-C. Cuarta Edición. Addison-Wesley Professional. 2011. 600p.

LEWIS, Rory. iPhone and iPad Apps for Absolute Beginners. Apress, 2011. 336p.

MASHMAN, Walter. The iPad in Cardiology: Tool or Toy [En línea] <<http://www.sciencedirect.com/consultaremoti.upb.edu.co/science/article/pii/S1936879811000033>> [Consultado en 15 de Abril de 2012].

MACIÁ PÉREZ, F., et al. Tecnologías web básicas En: Administración de servicios de internet: De la teoría a la práctica. 1ra ed. España: Publicaciones Universidad de Alicante, 2008. p. 40-43.

MARK, David. Beginning iOS 5 Development: Exploring the iOS SDK. Primera Edición. Apress, 2011. 752p.

NEIRA, Juan Pablo. CASTRO, Vladimir. Diseño e Implementación de una Red Neuronal en un DSPIC para el Control de Trayectoria de un Robot Detector de obstáculos. Bucaramanga, 2007, 98 h. Trabajo de grado (Ingeniero Electrónico). Universidad Pontificia Bolivariana. Facultad de Ingeniería Electrónica. Disponible en el catálogo en línea de la Biblioteca de la Universidad Pontificia Bolivariana: <<http://biblioteca.upbbga.edu.co/material.php?idmaterial=14811>>

RASSWEILER, Jens J. iPad -Assisted Percutaneous Access to the Kidney Using Marker-Based Navigation: Initial Clinical Experience. [En línea] <<http://www.sciencedirect.com.consultaremota.upb.edu.co/science/article/pii/S0302283811013911>> [Consultado en 15 de Abril de 2012].

RAY, Jhon. Sams Teach Yourself iOS 5 Application Development in 24 Hours. Tercera Edición. Sams. 2012. 800p.