



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

TEMA: IMPLEMENTACIÓN DE UN ROBOT EXPLORADOR
TIPO VEHÍCULO CONTROLADO POR MEDIO DEL INTERNET
COMO MATERIAL DIDÁCTICO PARA LA UNIDAD DE GESTIÓN
DE TECNOLOGÍAS

AUTOR: DÍAZ CÓRDOVA PEDRO GABRIEL

Trabajo de graduación previo a la obtención del título de:

TECNÓLOGO EN ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN & AVIÓNICA

Año 2014

CERTIFICACIÓN

Certifico que el presente Trabajo de Graduación fue realizado en su totalidad por el Sr. **DÍAZ CÓRDOVA PEDRO GABRIEL**, como requerimiento parcial para la obtención del título de **TECNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA**

ING. JESSY ESPINOSA

DIRECTORA DEL TRABAJO DE GRADUACIÓN

Latacunga, Noviembre del 2014

AUTORÍA DE RESPONSABILIDAD

Yo, DÍAZ CÓRDOVA PEDRO GABRIEL

DECLARO QUE:

El proyecto denominado **“IMPLEMENTACIÓN DE UN ROBOT EXPLORADOR TIPO VEHÍCULO CONTROLADO POR MEDIO DEL INTERNET COMO MATERIAL DIDÁCTICO PARA LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS”**, ha sido desarrollado en base a una investigación científica exhaustiva, respetando derechos intelectuales de terceros, las fuentes utilizadas se incorporan en la netgrafía.

Consecuentemente, este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, Noviembre del 2014

DÍAZ CÓRDOVA PEDRO GABRIEL

AUTORIZACIÓN

Yo, DÍAZ CÓRDOVA PEDRO GABRIEL

Autorizo a la Universidad de las Fuerzas Armadas la publicación, en la biblioteca virtual de la Institución, del trabajo que trata de la **“IMPLEMENTACIÓN DE UN ROBOT EXPLORADOR TIPO VEHÍCULO CONTROLADO POR MEDIO DEL INTERNET COMO MATERIAL DIDÁCTICO PARA LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga, Noviembre del 2014.

DÍAZ CÓRDOVA PEDRO GABRIEL

DEDICATORIA

El presente trabajo va dedicado primero a Dios, que permite que pueda seguir adelante con el mejor regalo que nos pudo brindar que es la vida.

A mis padres, pero principalmente para mi madre que durante su vida se ha sacrificado enteramente por brindar el mejor bienestar para todos sus hijos a demás de ser el pilar fundamental para llegar hasta estas estancias teniéndome paciencia, fe y aconsejándome día a día para no desmoronarme en los peldaños que se presentaron en este camino de aprendizaje.

PEDRO GABRIEL DÍAZ CÓRDOVA

AGRADECIMIENTO

A Dios, porque la vida me ha enseñado que hay alguien allá arriba mirándonos y cuidándonos, que no es necesario encontrarlo a través de nuestros sentidos sino por medio de la fe.

A mis padres, quienes con todo amor, nos han dado a mis hermanos y a mí lo mejor de sí mismos para formar la gran familia que tenemos. Por enseñarnos que cualquier dificultad, por más grande que sea, se puede afrontar con valor, cariño y comprensión. Porque les debo todo lo que soy y cuento con ellos en cada paso que doy.

A mis hermanos, porque con su cariño impulsan cada uno de mis sueños. Porque día con día me enseñan una nueva forma de sonreír.

A mi asesora de tesis Ing. Jessy Espinosa por todo su apoyo y orientación en todo el desarrollo de este trabajo de grado.

A Johana Bonilla, quien se convirtió en mi compañera de la vida y por enseñarme que una promesa no solo son palabras, sino un compromiso que debe cumplirse en tiempo y forma. A todos mis amigos porque de ellos he aprendido que la vida no es un camino que se recorre en solitario. Porque, como las estrellas, aunque no siempre pueda verlos, sé que están ahí.

PEDRO GABRIEL DÍAZ CÓRDOVA

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS.....	vi
ÍNDICE DE CUADROS.....	ix
ÍNDICE DE FIGURAS.....	x
INTRODUCCIÓN.....	xiv
RESUMEN.....	xv
SUMMARY	xvi
CAPÍTULO I.....	1
TEMA	1
1.1 ANTECEDENTES	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	2
1.3 JUSTIFICACIÓN	2
1.4 OBJETIVOS.....	3
1.4.1 GENERAL	3
1.4.2 ESPECÍFICOS.....	3
1.5 ALCANCE	3
CAPÍTULO II.....	5
MARCO TEÓRICO	5
2.1 MINICOMPUTADORA RASPBERRY PI MODELO B.....	5
2.1.1 CARACTERÍSTICAS	5
2.1.2 SISTEMAS OPERATIVOS SOPORTADOS.....	7

2.1.3 PROGRAMAS PREINSTALADOS	8
2.1.4 PYTHON (Python, 2001)	8
2.1.5 TERMINAL DE LINUX	10
2.1.6 PUERTO GPIO.....	12
2.2 HERRAMIENTA WEBIOPI DE RASPBERRY	13
2.3 MJPG-STREAMER	13
2.4 TECNOLOGÍA WI-FI	14
2.5 SERVIDORES WEB.....	14
2.5.1 APACHE2.....	15
2.6 PÁGINA WEB	16
2.6.1 LENGUAJE HTML	17
2.6.2 LENGUAJE CSS	20
2.6.3 LENGUAJE JAVASCRIPT	22
2.6.4 LENGUAJE PHP	23
2.7 SERVOMOTOR	25
2.7.1 FUNCIONAMIENTO	26
2.8 CIRCUITO INTEGRADO.....	27
2.8.1 CONTROLADOR DE MOTORES L298	27
CAPÍTULO III.....	29
DESARROLLO DEL TEMA	29
3.1 INSTALACIÓN Y CONFIGURACIÓN DE RASPBIAN	29
3.1.1 INSTALACIÓN DE RASPBIAN	29
3.1.2 CONFIGURACIÓN INICIAL DE RASPBIAN. (Bejarano, 2013)....	32
3.2 ACCEDIENDO AL ESCRITORIO DEL RASPBERRY PI	43
3.3 TRABAJANDO CON LOS PUERTOS GPIO.	52
3.4 INSTALACIÓN DE LA HERRAMIENTA WEBIOPI. (Google).....	55

3.5	INSTALACIÓN DEL SERVIDOR WEB.....	57
3.6	ACEDIENDO AL GPIO DESDE PHP.....	60
3.7	DISEÑO DE SOFTWARE DEL ROBOT MÓVIL.....	61
3.7.1	PÁGINA WEB.....	62
3.7.2	CONTROL DE LOS MOTORES DE DESPLAZAMIENTO	68
3.7.3	CONTROL DEL SERVOMOTOR DE LA CÁMARA	71
3.7.4	STREAMING DE VIDEO (Grinberg, 2013).....	72
3.8	HARDWARE DEL ROBOT MÓVIL.....	75
3.8.1	CONEXIONES ENTRE LOS CIRCUITOS	79
3.9	CONFIGURACIÓN DEL ROUTER	81
3.10	SIMULACIONES	82
	CAPÍTULO IV	84
	CONCLUSIONES Y RECOMENDACIONES	84
4.1	CONCLUSIONES.....	84
4.2	RECOMENDACIONES	84
	GLOSARIO.....	86
	NETGRAFÍA.....	89

ÍNDICE DE CUADROS

CUADRO 1: EJEMPLO DE HTML	19
CUADRO 2: EJEMPLO CSS	21
CUADRO 3: MUESTRA DE TEXTO EN CÓDIGO JAVASCRIPT	22
CUADRO 4: EJEMPLO PÁGINA PHP	23
CUADRO 5: DECLARACIÓN DE BOTONES EN HTML.....	63
CUADRO 6: PROGRAMACIÓN DE JAVASCRIPT EN HTML	64

ÍNDICE DE FIGURAS

FIGURA 1: PLACA RASPBERRY PI MODELO B.....	6
FIGURA 2: ICONOS RASPBIAN	8
FIGURA 3: PUERTO GPIO	13
FIGURA 4: ESTRUCTURA HTML	17
FIGURA 5: DECLARACIÓN DE HTML	18
FIGURA 6: INICIO HTML.....	19
FIGURA 7: GENERACIÓN DE IMAGEN CON PHP	25
FIGURA 8: ESTRUCTURA SERVOMOTOR	27
FIGURA 9: CIRCUITO DRIVER DE MOTOR L298.....	28
FIGURA 10: ESCRITORIO RASPBERRY PI.....	29
FIGURA 11: PÁGINA DE DESCARGAS RASPBERRY	30
FIGURA 12: SOFTWARE RASPBIAN.....	30
FIGURA 13: PROGRAMA WIN32 DISK IMAGER.....	30
FIGURA 14: WIN32 ABIERTO.....	31
FIGURA 15: SELECCIÓN DEL SISTEMA OPERATIVO.....	31
FIGURA 16: ESCRITURA DE SISTEMA OPERATIVO.....	32
FIGURA 17: MICRO SD Y RASPBERRY PI.....	32
FIGURA 18: MENÚ PRINCIPAL.....	33
FIGURA 19: CAMBIO CORRECTO DE CONTRASEÑA.....	34
FIGURA 20: ACTIVACIÓN DE ESCRITORIO GRAFICO.....	34
FIGURA 21: CAMBIO DE LOCALIDAD	35
FIGURA 22: SELECCIÓN DE IDIOMA DEL SISTEMA.....	35
FIGURA 23: SELECCIÓN DE UBICACIÓN GEOGRÁFICA.....	36
FIGURA 24: SELECCIÓN DE ZONA HORARIA	36

FIGURA 25: SELECCIÓN DE CARACTERÍSTICAS DEL TECLADO	37
FIGURA 26: SELECCIÓN DE IDIOMA DEL TECLADO.....	37
FIGURA 27: CARACTERÍSTICAS DEL TECLADO	37
FIGURA 28: CARACTERÍSTICAS DEL TECLADO	38
FIGURA 29: FINALIZACIÓN DE CONFIGURACIÓN DEL TECLADO	38
FIGURA 30: ACTIVACIÓN DE CÁMARA.	38
FIGURA 31: ACTIVACIÓN DE RASTRACK.	39
FIGURA 32: SELECCIÓN DE VELOCIDAD DE PROCESO DE DATOS.....	40
FIGURA 33: MENÚ DE OPCIONES AVANZADAS.....	40
FIGURA 34: CAMBIO DE NOMBRE DEL DISPOSITIVO.	41
FIGURA 35: MEMORIA COMPARTIDA.....	41
FIGURA 36: CONFIGURACIÓN DE AUDIO.....	42
FIGURA 37: FUENTE DE DESCARGA.	44
FIGURA 38: EJECUCIÓN DE PUTTY.....	45
FIGURA 39: INTRODUCIR IP.....	45
FIGURA 40: ADVERTENCIA.....	46
FIGURA 41: DATOS DE CONFIRMACIÓN.	46
FIGURA 42: DATOS DE CONFIRMACIÓN.	47
FIGURA 43: CONSOLA DE COMANDOS.....	47
FIGURA 44: INSTALACIÓN DE XRDP.....	48
FIGURA 45: BÚSQUEDA DEL ESCRITORIO REMOTO.....	48
FIGURA 46: INTRODUCIR IP RASPBERRY PI.	49
FIGURA 47: DATOS DE CONFIRMACIÓN.	49
FIGURA 48: ESCRITORIO DE RASPBIAN.	50
FIGURA 49: ADAPTADOR HDMI A VGA	50

FIGURA 50: ADAPTADOR HDMI A VGA	51
FIGURA 51: CONEXIÓN DE CABLE EN PLACA HDMI.	51
FIGURA 52: ALIMENTACIÓN EXTERNA.....	52
FIGURA 53: DIAGRAMA PARA EL EJEMPLO.....	52
FIGURA 54: EJECUCIÓN DE PROGRAMA PY	55
FIGURA 55: RESULTADO DE PRACTICA.....	55
FIGURA 56: MONITOREO DE FUNCIONAMIENTO DEL GPIO.....	56
FIGURA 57: INSTALACIÓN DEL SERVIDOR	57
FIGURA 58: REINICIO SERVIDOR.....	57
FIGURA 59: SERVIDOR WEB.....	58
FIGURA 60: CREACIÓN HOLA MUNDO	59
FIGURA 61: ABRIENDO HOLA MUNDO EN LA WEB	59
FIGURA 62: CREACIÓN DE CARPETA PRUEBAS	60
FIGURA 63: ABRIENDO HOLA MUNDO EN LA WEB	61
FIGURA 64: PÁGINA WEB RASPIBOT.....	62
FIGURA 65: ENCABEZADO HTML	63
FIGURA 66: CONTENIDO CSS.....	64
FIGURA 67: DECLARACIÓN DE JAVASCRIPT EN HTML	65
FIGURA 68: ACCIÓN DE BOTONES EN LA CONSOLA.....	67
FIGURA 69: ARCHIVO PHP.....	68
FIGURA 70: CONEXIONES DEL GPIO CON EL DRIVER L298.....	69
FIGURA 71: ESTADO AVANZA	69
FIGURA 72: PROGRAMACIÓN DE MOTORES.....	70
FIGURA 73: CIRCUITO EN ESTADO DE RETROCEDER.....	70
FIGURA 74: CIRCUITO EN ESTADO DE GIRO DERECHA.	70

FIGURA 75: CIRCUITO EN ESTADO DE GIRO IZQUIERDA.....	71
FIGURA 76: PROGRAMACIÓN EN ESTADO DE STOP.....	71
FIGURA 77: PROGRAMACIÓN DE CONTROL DE SERVOMOTOR.	72
FIGURA 78: CORRIENDO MJPG-STREAMER.....	74
FIGURA 79: PÁGINA PRINCIPAL DE MJPG-STREAMER.....	75
FIGURA 80: TRANSMISIÓN EN VIVO.	75
FIGURA 81: CHASIS 4X4 DESARMADA	77
FIGURA 82: CHASIS EN PIE.	77
FIGURA 83: CORTES DE FIBRA DE VIDRIO.....	78
FIGURA 84: DISEÑO PRE ACABADO.....	78
FIGURA 85: ESTRUCTURA CON CIRCUITOS ELECTRÓNICOS.....	79
FIGURA 86: ROBOT MÓVIL ARMADO	79
FIGURA 87: LOCALIDAD DE CIRCUITOS.....	80
FIGURA 88: DIAGRAMA DE CONEXIONES ENTRE CIRCUITOS	80
FIGURA 89: ESTRUCTURA DE BATERÍAS.....	81
FIGURA 90 : ACCESO MODEM USB (ROUTER).....	81
FIGURA 91 : APERTURA DE PUERTOS DE INTERNET.	82
FIGURA 92: 1º INTERIORES DE LABORATORIO.....	83
FIGURA 93: 2º VISUALIZACIÓN DEL PASILLO	83

INTRODUCCIÓN

Los avances en ciencia y tecnología han permitido el desarrollo de la robótica. En los últimos años, la construcción de robots se ha incrementado gracias a las necesidades de la industria, la milicia y el entretenimiento. Porque realizan tareas de forma más exacta y más barata que los humanos.

Al rededor de todo el mundo, los robots están en acción, desde pintando carros en las plantas de Automóviles, caminando dentro de volcanes activos, o manejando grandes trenes. Mientras se vuelven más resistentes, más ágiles, y más sofisticados, los robots de hoy día hacen cosas que los seres humanos no pueden realizar, pueden ser porque sean actividades peligrosas, de gran precisión o simplemente no las quieren hacer.

Por tal razón, ante el notable y gran avance tecnológico de la robótica a nivel mundial, surge la necesidad de implementar un sistema que pueda ser controlado a distancia vía internet; realizando tareas específicas y básicas, y que sirva como apoyo didáctico a nivel estudiantil, todo esto enfocado al campo de la investigación y académico, mirando al futuro pudiendo ser implementado para tareas de exploración o rescate.

El diseño del explorador tipo vehículo controlado por medio del Internet está enfocado en un robot móvil 4x4; posee una estructura que soportar golpes leves y que puede movilizarse gracias a 4 motores DC, el control se lo puede realizar vía internet con una dirección IP específica asignada al robot, este podrá transmitir un flujo de imágenes constantes que pueden ser visualizadas como una transmisión de video en vivo.

RESUMEN

Implementación de un robot explorador tipo vehículo controlado remotamente que servirá como material didáctico en la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas – ESPE, este tipo de tecnología permite tener un medio de comunicación a través del Internet, apoyada con herramientas de software libre y nueva tecnología como Raspbian; se utilizó una minicomputadora denominada Raspberry Pi que está siendo una innovación en el medio electrónico, se explica funcionamiento de sus puertos de comunicación, específicamente los de propósito general de entrada y salida, además del USB donde se pueden conectar dispositivos que permiten el ingreso de video y la comunicación inalámbrica con el Raspberry Pi, apoyado en aplicaciones de código abierto basado en LINUX creando un escenario de Cliente - Servidor entre un Raspberry Pi y una PC, así mismo, con WI-FI como herramienta que permite entrelazar de manera inalámbrica al Raspberry Pi. Usando una aplicación web alojado en el Raspi, permite al usuario la recepción de video en flujo y control de desplazamiento del robot, el mismo que fue creado en una estructura de aluminio y fibra de vidrio que consta de 4 motores de corriente continua, además en el alojamiento de la cámara web se sitúa un servomotor para su movimiento de inclinación y con esto ayuda en una total visualización de objetos que se sitúen en su entorno de desplazamiento. La implementación del robot explorador móvil trae consigo una nueva generación de la electrónica dando a conocer medios de comunicación humano máquina actuales.

Palabras Claves: Innovación Electrónica, Raspberry Pi, Comunicación Inalámbrica, Cliente Servidor

SUMMARY

The implementation of an explorer robot that works with a radio control will be used as a didactic material in the Technology Management Unit of the Army Force University- ESPE. This kind of technology allows having a communication through internet, helped with tools of free software and new technology as Raspbian. A minicomputer called Raspberry Pi was used which is a great innovation in the electronic world. Operation of its communication ports is described especially input and output ports, also USB where video devices and wireless communication with Raspberry Pi are connected, it was supported by free coded Linux applications based on LINUX creating a Client – server field between Raspberry Pi and a computer. As the same way Wi fi as tool allows a wireless connection with Raspberry Pi. A web application permits to receive a video signals and robot's movement orders, it was made of aluminum structure, and fiberglass which consists of 4 DC motors. Camera web has a servomotor used to the tilt movements which help to visualize that objects are placed in their displacement environment. The implementation of an mobile explorer robot brings a new electronic generation, sharing means of human communication and current machines.

Key words: Electronic Innovation, Raspberry Pi, Wireless Communication, Client server.

CAPÍTULO I

TEMA

Implementación de un robot explorador tipo vehículo controlado por medio del internet como material didáctico para la unidad de gestión de tecnologías.

1.1 ANTECEDENTES

El avance tecnológico es inevitable a nivel mundial, en los países desarrollados los niños crecen con tecnología actual en sus manos, existen cursos de apoyo los cuales los inician en el medio electrónico manteniéndolos ocupados en sus tiempos libres y así aprendiendo y desarrollando habilidades especiales. En países subdesarrollados la tecnología no es algo que se la puede encontrar a la vuelta de la esquina y peor aún desarrollarla, porque se carece de capacitación, recursos, apoyo y dedicación.

En el Ecuador desde hace unos años atrás en los Centros Educativos del país, el deseo de incorporar la enseñanza de robótica en las aulas se vio limitada por la situación económica y la difícil accesibilidad a la tecnológica. En pocos Institutos y universidades principalmente las vinculadas con la electrónica, comenzaron sus primeros proyectos de robótica con interfaces conectadas a las PC mediante el puerto paralelo o serial. Los robots contruidos no tenían ningún tipo de autonomía, dado que estaban obligados a seguir conectados a la PC. Es lo que llamamos control automatizado, a diferencia de los robots autónomos que llevan el procesamiento sobre su estructura.

A fines de la década del 90, comienzan a aparecer en el mercado un conjunto de kits educativos de robótica que solucionan estos dos problemas: no exigen conocimientos electrónicos profundos y permiten la construcción de robots autónomos. El más popular de estos kits es el de la firma Lego1, que rompió las fronteras de las instituciones educativas y que hoy, gracias a Internet, posee una comunidad inmensa de educadores, científicos y

hobistas desarrollando software y hardware para extender sus funcionalidades.

También, existe la competencia de los creadores de Arduino la cual es una plataforma de hardware libre destinada para estudiantes, basada en una placa con un microcontrolador y un entorno de desarrollo. Arduino se puede usar para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos de Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en tu ordenador.

1.2 PLANTEAMIENTO DEL PROBLEMA

Actualmente la Unidad de Gestión de Tecnologías cuenta con laboratorios donde los estudiantes de Electrónica Mención Instrumentación y Aviónica pueden realizar prácticas de todo tipo y de toda clase de tecnología ya sea analógica o digital.

Hoy en día la tecnología está avanzando a pasos agigantados, y todo lo relacionado con microcontroladores está actualizando su mercado y está cambiando el entorno tecnológico y dejando atrás muchos instrumentos electrónicos.

Los estudiantes de la Unidad de Gestión de Tecnologías deben tener el privilegio de conocer, manejar y familiarizarse con ella y así no encontrarse con sorpresas en el nuevo mundo.

1.3 JUSTIFICACIÓN

El proyecto se inicia con la idea de diseñar un robot que aporte con la enseñanza y actualización de información para los futuros electrónicos que se están formando en la Unidad de Gestión Tecnológica. La investigación sobre la robótica es un excelente medio para que el alumno se integre a diferentes y nuevas tecnologías, ampliando así su relación en el campo electrónico.

En el nacimiento del Raspberry Pi inician también las altas posibilidades de que las personas se introduzcan en un ambiente que es nuevo para

nuestra sociedad.

El robot cumplirá las funciones de transmisión de vídeo, exploración y conexión a red WIFI sin embargo se le podrá adherir funciones futuras con las ideas que tengan los estudiantes o el profesor. Todo esto se podrá lograr con una micro computadora (Raspberry Pi) que esta revolucionado el mundo con sus prestaciones que brinda facilidad de control y manejo de dispositivos electrónicos por medio de su puerto GPIO, simplemente depende de las necesidades que surjan en la vida diaria y resolverlas utilizando la microcomputadora.

Por todo lo mencionado anteriormente existe la necesidad de Implementar un robot en el área de Electrónica Mención Instrumentación y Aviónica.

1.4 OBJETIVOS

1.4.1 GENERAL

Implementar un robot explorador tipo vehículo controlado por medio del Internet para que sirva como material didáctico para la Unidad de Gestión de Tecnologías.

1.4.2 ESPECÍFICOS

- Indagar información sobre el funcionamiento del Raspberry Pi, para determinar los elementos que pueden intervenir en la implementación del robot.
- Construir una estructura física para el robot y que sea adecuada tanto en forma estética como funcional, para poder realizar prácticas de exploración sobre este medio.
- Configurar el sistema que constituye el robot explorador tipo vehículo además de programar los puertos GPIO para funciones específicas.

1.5 ALCANCE

La implementación del Robot tipo vehículo en el área Electrónica Mención Instrumentación y Aviónica permitirá que la nueva generación de estudiantes tengan a la mano una iniciativas en el ámbito de la robótica,

creciendo así sus conocimientos y más curiosidades en temas actuales que están cambiando el mundo.

Con la implementación del Robot tipo vehículo lo que se quiere es renovar el medio de aprendizaje en los estudiantes. Con aportación de ideas el robot puede cumplir con más funciones, porque contiene una tecnología que está dando sus primeros pasos en nuestro país y no tiene límite, además se mira a futuro que el robot pueda realizar misiones de exploración o rescate.

CAPÍTULO II

MARCO TEÓRICO

2.1 MINICOMPUTADORA RASPBERRY PI MODELO B

Es un ordenador de placa reducida o (placa única) (*SBC*) de bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos “Turbo” para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB).

El diseño no incluye un disco duro ni unidad de estado sólido, simplemente usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa. El 29 de febrero de 2012 la fundación empezó a aceptar órdenes de compra del modelo A, y el 4 de febrero de 2013 del modelo B.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora); y promueve principalmente el aprendizaje del lenguaje de programación Python, y otros lenguajes como Tiny BASIC, C y Perl. (Fundacion, 2011)

2.1.1 CARACTERÍSTICAS

Con unas dimensiones de placa de 8.5 por 5.3 cm, en el modelo B que es el que se comercializa tiene las siguientes características:

- Su corazón un chip integrado Broadcom BCM2835, que contiene un procesador ARM11 con varias frecuencias de funcionamiento y la posibilidad de subirla (overclock) hasta 1 GHz sin perder la garantía.
- Un procesador gráfico VideoCore IV.

- Distintas cantidades de memoria RAM, las últimas Raspberry Pi cuentan con 512 MB de memoria. Todo ello equivale en la práctica a un ordenador con unas capacidades gráficas similares a la XBOX de Microsoft y con la posibilidad de reproducir vídeo en 1080p.
- Dispositivo de Arranque mediante un lector de memorias SD en la cual se carga el sistema operativo (La tarjeta puede ser de cualquier capacidad o de acuerdo a los trabajos que se vayan a realizar).
- 2 Conectores USB 2.0.
- Conector Ethernet RJ-45 10/100.
- Salida de Video Digital HDMI (Audio y Video).
- Salida de RCA Video Analógico (S-Video).
- Audio Analógico (Conector 3,5mm).
- Conector GPIO.
- Conector de alimentación vía Micro USB 5 voltios.

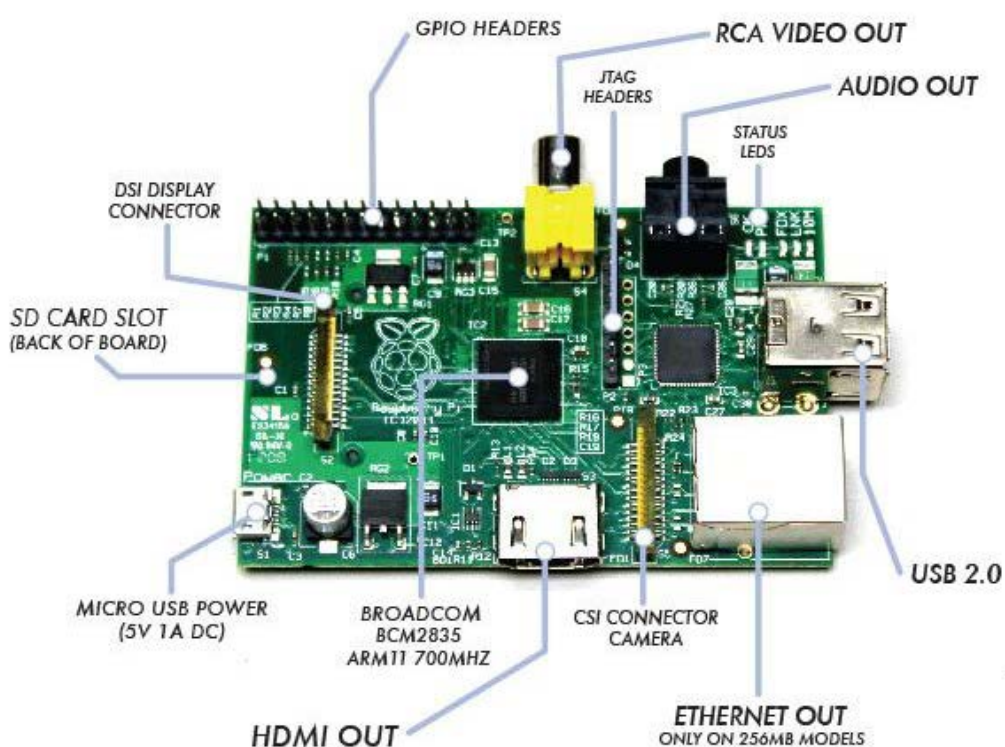


Figura 1: Placa Raspberry Pi modelo B

2.1.2 SISTEMAS OPERATIVOS SOPORTADOS

El software que permite el control de la computadora es el sistema operativo en el que la Raspberry Pi hace la diferencia con las computadoras de escritorio o las portátiles, además del tamaño y el precio que involucra a la minicomputadora.

La Raspberry Pi se diseñó para correr en un sistema operativo llamado GNU/Linux, debido a que este es de código abierto o libre: se puede descargar todo su código fuente y realizar los cambios que se desee. Nada está escondido, y como no se piden permisos para ver los cambios que se realicen, estos pueden ser vistos en su totalidad al público.

El inconveniente que existe es que son compatibles entre sí: un programa escrito sobre un sistema Debian operará perfectamente bien sobre Arch Linux y viceversa. Pero existe una restricción, un programa desarrollado en Windows o OS X no se ejecutará en una versión de Linux.

Para Raspberry Pi existen algunos sistemas operativos disponible pero el más recomendado es Raspbian, es básicamente un sistema operativo gratuito que fue optimizado a partir de Debian para ser utilizado en el Raspberry Pi (Raspberry + Debian = Raspbian).

Raspbian no es el único sistema operativo disponible para el Raspberry Pi. De hecho, originalmente el sistema operativo recomendado para el Raspberry Pi era Fedora de Red Hat, sin embargo en poco tiempo Raspbian fue capaz de desplazar Fedora y convertirse en el sistema operativo recomendado para el Raspberry Pi. (Fundacion, 2011)

Sistemas Operativos Soportados:

- Raspbian “wheezy” (Debian).
- ArchLinux.
- Fedora.
- QtonPi (QT SDK).
- OpenElec.

- Raspbcm.
- Android (en desarrollo por usuarios).

2.1.3 PROGRAMAS PREINSTALADOS

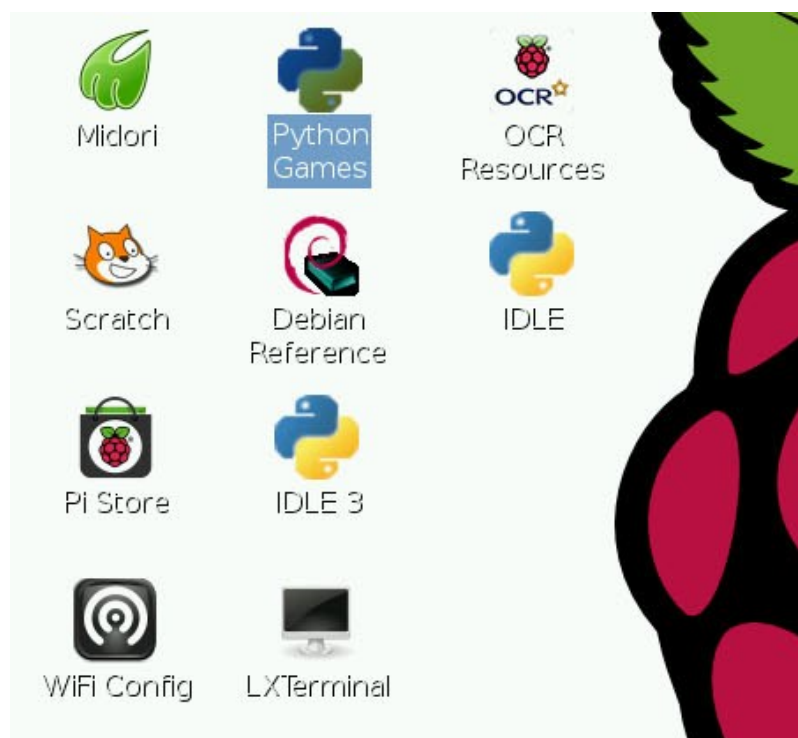


Figura 2: Iconos raspbian

En la figura 2 se puede apreciar con facilidad los detalles en el escritorio. Por “default” en el escritorio aparecen varios accesos directos a diferentes aplicaciones en Raspbian. Tres accesos directos muy útiles, el primero es Midori que es un buscador web, segundo, Wifi config el cual permite la configuración de una conexión de red inalámbrica, por medio de un dispositivo usb y por último, Debian Reference que es un manual de referencia de Debian. Además de los tres accesos directos mencionados anteriormente, existen dos aplicaciones que son muy importantes y serán utilizadas con frecuencia, IDLE (PYTHON) y LXTerminal.

2.1.4 PYTHON (Python, 2001)

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa,

desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

1. La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
2. La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
3. La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
4. Además, Python es gratuito, incluso para propósitos empresariales.
5. Brinda características especiales como son:
 - **Multiplataforma:** Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
 - **Interpretado:** Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.
 - **Interactivo:** Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada

sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

- **Orientado a Objetos:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- **Funciones y librerías:** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.

2.1.5 TERMINAL DE LINUX

Esta herramienta super útil es la más temida por los usuarios nuevos cuando recién aterrizan en Linux, esto debido principalmente a los mitos falsos que se ciernen sobre ella e infundados a su vez por pseudos “especialistas” de informática del sistema Microsoft Windows principalmente.

Lógicamente al hablar de la terminal se refiere a los comandos, Actualmente dominar esos comandos son muy fáciles, y a su vez se aprende a conocer el sistema operativo.

Actualmente apenas es necesario utilizar comandos para la gestión y administración del sistema ya que el mundo Linux ha perfeccionado el uso de herramientas gráficas para todo tipo de tareas. Pero si se conoce la potencia de los comandos y la comodidad de uso del Terminal, se da cuenta de que prácticamente todo en el sistema se puede controlar mediante línea de instrucciones, y si una persona se acostumbra se convertirá en un aliado extraordinario, porque por medio de él se puede tener acceso total al sistema.

Usar la consola o terminal y su línea de comandos puede proporcionar un poco de diversión extra si se decide darle una oportunidad.

Para entender claramente se explicara uno de los comandos típicos que se conoce en linux, sudo apt-get y se desglosara a continuación.

- sudo (super user do o hacer super usuario) es un comando que permite momentáneamente al usuario normal gestionar paquetes con una elevación de privilegios cercana a administrador. (recordar que un usuario no tiene permisos de administración por seguridad, básicamente sería un usuario limitado en los sistemas Microsoft Windows.) Vale la pena recordar que para tener todos los privilegios de administrador, se debe usar el su.
- apt (Advanced Packaging Tool o Herramienta Avanzada de Empaquetado) es el medio para gestionar paquetes, ya sea instalar o eliminar programas.

Si se analiza el comando completo, se darán cuenta de que sudo apt-get, en palabras claras seria “hacerse super usuario para conseguir que el paquete”.

Ejemplo de instalación, para instalar luvciwiew. Antes de comenzar asegurarse de que luvciwiew esté en el repositorio de manera pre-determinada, Linux lo trae de manera predeterminada desde hace tiempo, entonces abrir la terminal y colocar:

- sudo apt-get install luvciwiew

En el comando escrito se podría decir que significa “hacerse super usuario para conseguir que el paquete luvciwiew se instale”, no es nada del otro mundo son comandos que no se necesita de muchos conocimientos, en cuestión de saber que significa y que hace.

Ejemplo de desinstalación, para esta acción también se utilizara el comando de sudo apt-get, pero en esta ocasión se va a remove, no a instalar por lo tanto se usa:

- sudo apt-get remove luvciwiew

Aquí se indica algo como “hacerse super usuario para conseguir que el paquete luvciwiew se remueva”.

2.1.6 PUERTO GPIO

El puerto GPIO de la Raspberry Pi está localizado en la parte superior izquierda de la tarjeta de circuito, etiquetada como P1. Este es un puerto de 26 pines, el cual está dispuesto en dos filas de 13 pines cada una de tipo macho de 2.54 mm de largo de fábrica. La longitud de 2.54 mm de estos pines es importante ya que es muy común ver esa longitud de los pines en electrónica, y este es el espacio estándar para plataformas prototipo.

Cada pin del puerto GPIO tiene su propio propósito. Los números de pines para la GPIO son divididos en dos filas, en la que la fila inferior toma los números impares y la superior los números pares. Esto es muy importante tener en cuenta al momento que se esté trabajando con el puerto GPIO de la Raspberry Pi ya que la mayor parte de otros dispositivos usa un sistema diferente de numeración de pines. Además no están marcados sobre la tarjeta lo que lleva a una fácil confusión de que pin se esté usando. Aunque el puerto GPIO proporciona una fuente de alimentación, aprovechada de la fuente entrante sobre el conector micro-USB para el pin 2, internamente la Raspberry Pi trabaja a 3.3 V lógico. Esto significa que los componentes sobre la minicomputadora trabajan con una fuente de alimentación de 3.3 V.

El puerto GPIO proporciona siete pines para propósitos generales, estos son los siguientes: pin 11, pin 12, pin 13, pin 15, pin 16, pin 18 y pin 22. Adicionalmente, se puede usar el pin 7 para propósito general dándose el caso de que no cumpla el hecho de proporcionar una señal de reloj de uso de propósito general, dando como resultado que se tenga un pin más de los siete de uso de propósito general. Estos pines se pueden activar en dos estados: en alto, donde ellos proporcionan un voltaje positivo de 3.3 V; y un estado bajo, donde el valor de ellos son iguales a tierra o 0 V. análogamente son iguales al 1 y 0 de la lógica binaria y pueden ser usados para encender o apagar otros componentes. (Fundacion, 2011)

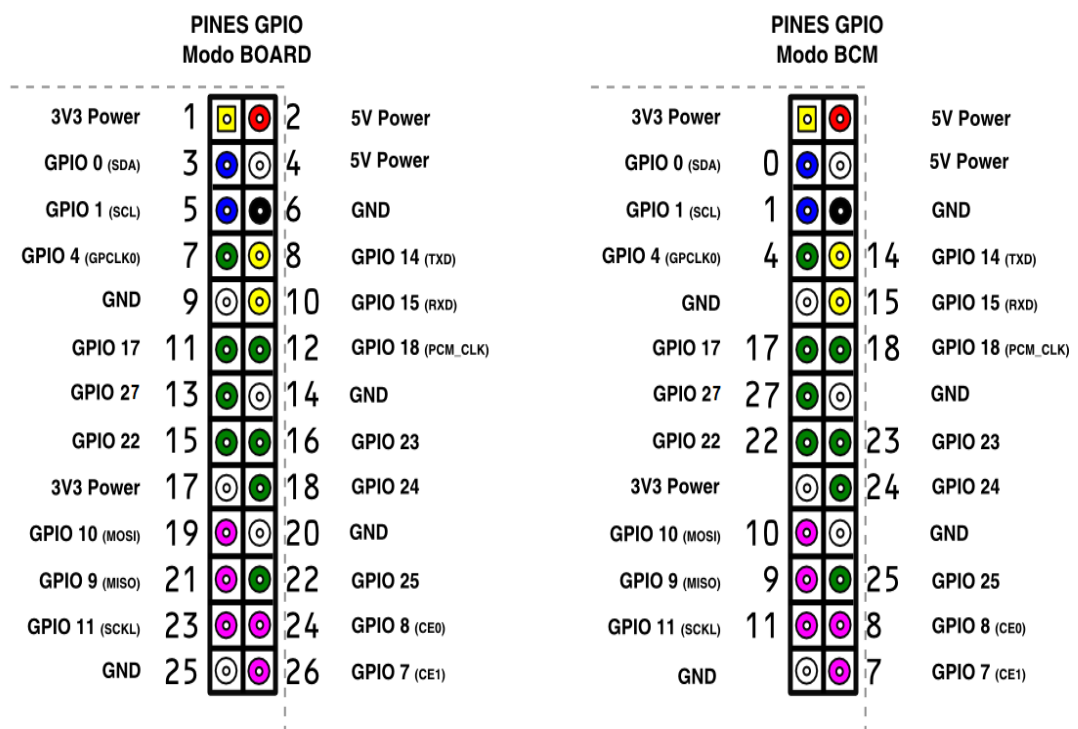


Figura 3: Puerto GPIO

2.2 HERRAMIENTA WEBIOPI DE RASPBERRY

Webiopi es un framework para la Raspberry Pi el cual permite controlar los pines GPIO de manera local y remota, permite controlar los dispositivos que estén conectados a los GPIO como motores vía PWM, sensores de temperatura, etc mediante de una aplicación web la cual es instalada fácilmente. (Google)

2.3 MJPG-STREAMER

Es una aplicación de línea de comandos que copió JPG marco de un único complemento de entrada a múltiples plugins de salida. Puede ser utilizado para transmitir archivos JPEG a través de una red basada en IP de la cámara web a un espectador como Firefox, Cambozola, VideoLanClient o incluso en un dispositivo Windows Mobile que ejecuta el TCPMP-Player.

Fue escrito para dispositivos embebidos con Recursos muy limitados en términos de RAM y CPU. Su origen, el "uvc_streamer" fue escrito, porque las cámaras compatibles con Linux-UVC producen directamente JPEG-data, permitiendo rápido y performant M-JPEG de forma simultánea, incluso desde

un dispositivo embebido corriendo OpenWRT. El módulo de entrada "input_uvc.so" captura tales marcos JPG desde una cámara web conectada.

Esta herramienta puede ser modificada y distribuida de acuerdo a los términos de la GPL v2.

Mjpeg_streamer genera automáticamente un conjunto de páginas HTML que ilustra diferentes métodos para transmitir el flujo de imágenes sobre su navegador. El mejor resultado viene a través de Google Chrome en cualquier sistema operativo, pero también funciona bien en Firefox, Opera e IE.

Actualmente no hay temas conocidos, pero ya que este software es muy joven y no se utilizan ampliamente, puede causar problemas. Se debe saber realmente lo que se está haciendo, si se utiliza este software. (MJPEG-streamer, 2007)

2.4 TECNOLOGÍA WI-FI

WIFI, también llamada WLAN (*wireless lan*, red inalámbrica) o estándar IEEE 802.11. WIFI no es una abreviatura de *Wireless Fidelity*, simplemente es un nombre comercial.

Wi-Fi (Wireless Fidelity) es la tecnología utilizada en una red o conexión inalámbrica, para la comunicación de datos entre equipos situados dentro de una misma área (interior o exterior) de cobertura.

Conceptualmente, no existe ninguna diferencia entre una red con cables (cable coaxial, fibra óptica, etc.) y una inalámbrica. La diferencia está en que las redes inalámbricas transmiten y reciben datos a través de ondas electromagnéticas, lo que supone la eliminación del uso de cables y, por tanto, una total flexibilidad en las comunicaciones.

2.5 SERVIDORES WEB

Los servidores web son aquellos cuya tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo HTTP.

Básicamente un servidor WEB consta de un intérprete HTTP el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla.

Además los servidores pueden disponer de un intérprete de otros lenguajes de programación que ejecutan código embebido dentro del código HTML de las páginas que contiene el sitio antes de enviar el resultado al cliente. Esto se conoce como programación de lado del servidor y utiliza lenguajes como ASP, Apache, PHP, Perl y Ajax. Las ventajas de utilizar estos lenguajes radica en la potencia de los mismos ejecutando tareas complejas. (Rivera, Blog Jefferson Rivera, 2013)

2.5.1 APACHE2

Como se mencionó en la sección anterior Apache es el servidor web más usado y este corre en un sistema operativo de código abierto como lo es el sistema Linux. Se resalta nuevamente que los servidores web se usan para servir páginas web solicitadas por equipos cliente, estos últimos normalmente solicitan y muestran páginas web mediante el uso de navegadores web como Firefox, Opera o Mozilla.

Los usuarios introducen un Localizador de Recursos Uniforme (Uniform Resource Locator, URL) para señalar a un servidor web por medio de su Nombre de Dominio Totalmente Cualificado (Fully Qualified DomainName, FQDN) y de una ruta al recurso solicitado.

Existe una configuración de los servidores web Apache la que consiste en el uso combinado del motor de bases de datos MySQL, el lenguaje de scripting PHP, y otros lenguajes de scripting populares como Python y Perl. Esta configuración se denomina LAMP (Linux, Apache, MySQL y Perl/Python/PHP), con esto se conforma una potente y robusta plataforma para el desarrollo y distribución de aplicaciones basadas en la web. (Apache, 1999)

2.6 PÁGINA WEB

Una página web se define como un documento electrónico el cual contiene información textual, visual y/o sonora que se encuentra alojado en un servidor y puede ser accesible mediante el uso de navegadores. Una página web forma parte de una colección de otras páginas webs dando lugar al denominado sitio web el cual se encuentra identificado bajo el nombre de un dominio.

La creación y desarrollo de una página web se realiza bajo un lenguaje de programación capaz de ser interpretados por los navegadores, lenguajes como el HTML, PHP, Java Script, CSS, ASP, JSP o RUBY son ejemplos entre otros. Al inicio de la era de internet accesible, sobre los años 90, era necesario el conocimiento de algún lenguaje de programación para el desarrollo de una web, siendo una tarea encomendada a personas con altos conocimientos informáticos, hoy en día se cuenta con software especializado capaz de trabajar como un editor de texto ,estilo Word, que transforman toda la información insertada en un lenguaje de programación capaz de ser interpretado por los navegadores, de esta forma se liberalizó y se globalizó la creación de páginas webs con apenas unos escasos conocimientos informáticos, programas como Dreamweaver, Amaya, Sharepoint Designer o Mozilla Composer son entre otros los denominados WYSIWYG (acrónimo del inglés " lo que ves es lo que obtienes") capaces de crear complejas páginas webs con el entorno de un simple editor de texto.

Una vez creada y desarrollada una página web es necesario alojarla en un servidor el cuál lo se puede definir como un ordenador conectado constantemente a la intranet privada o a internet cuyo objetivo es poder disponer de la página web a cualquier hora del día. Para poder acceder al servidor es necesario el uso de programas denominados clientes FTP, los cuales conectan el ordenador personal donde se ha desarrollado la página con el servidor donde se alojará, Filezilla, FileFTP o Cute FTP son entre otros ejemplos de clientes FTP.

Por último, una vez alojadas nuestras páginas webs estas son accesibles mediante el uso de navegadores los cuales permiten visualizar correctamente la información alojada y desarrollada, Chrome, Mozilla, Internet Explorer y Safari son entre otros ejemplos de navegadores ampliamente utilizados. (González, 2011)

2.6.1 LENGUAJE HTML

HTML es el acrónimo de HyperText Markup Language (Lenguaje de Marcado de Hipertexto). Este lenguaje indica a los navegadores cómo deben mostrar el contenido de una página web.

El lenguaje html contiene dos partes:

- El contenido, que es el texto que se verá en la pantalla de un ordenador,
- Las etiquetas y atributos que estructuran el texto de la página web: en encabezados, párrafos, listas, enlaces, etc. y normalmente no se muestra en pantalla.

HTML es el lenguaje que se emplea para el desarrollo de páginas de internet, solo puede crear en cualquier tipo de editor de texto. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que permiten dirigir a otras páginas, saltos de línea, listas, tablas, etc.

Se puede decir que HTML sirve para crear páginas web, darles estructura y contenido.

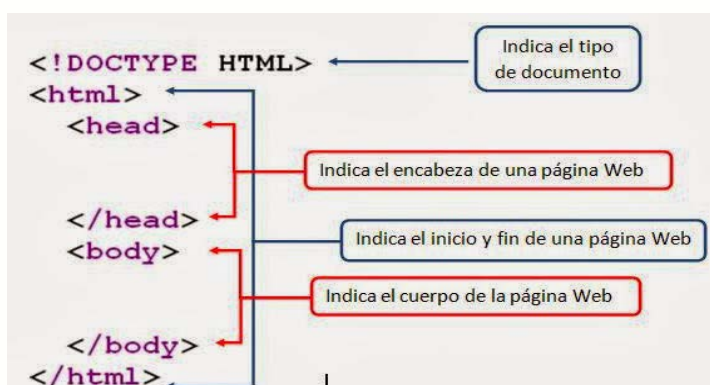
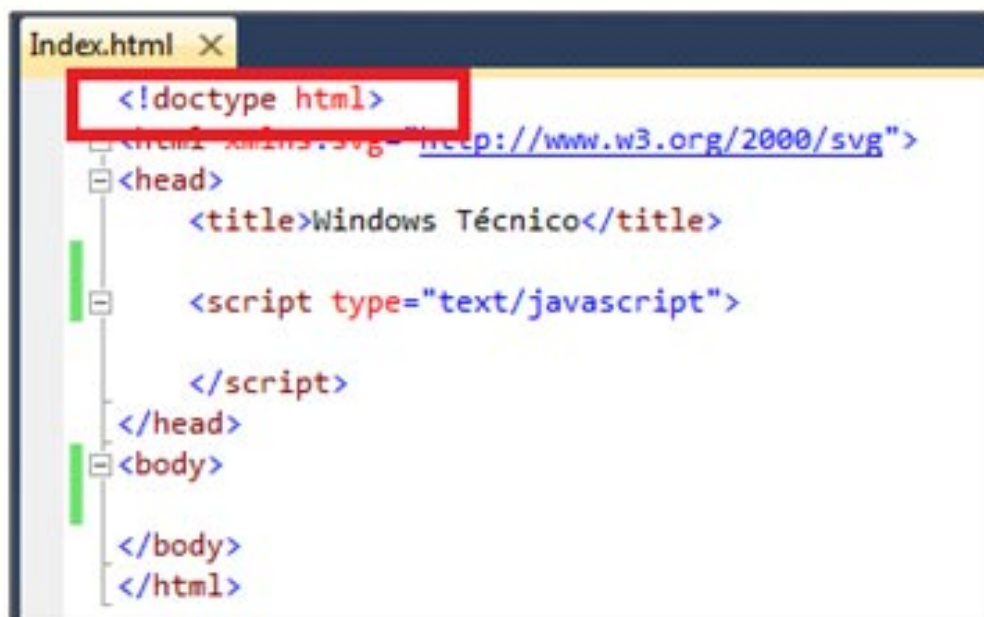


Figura 4: Estructura HTML

1. Declaración del tipo de documento.

Son dos etiquetas que indican que tipo de documento se está generando y que normas se ajusta. Normalmente serán siempre iguales y será el editor el que se encargue de colocarlas al principio de la página, cuando sea necesario.



```
Index.html X
<!doctype html>
<html xmlns:svg="http://www.w3.org/2000/svg">
  <head>
    <title>Windows Técnico</title>
    <script type="text/javascript">
    </script>
  </head>
  <body>
  </body>
</html>
```

Figura 5: Declaración de html

2. Encabezado de la aplicación web

El encabezado se escribe entre las etiquetas `<head></head>`. Dentro de ellas se expresan básicamente:

- La codificación de los caracteres del programa.
- Las etiquetas de html.
- El título del documento que luego aparece en la barra de título de la página.
- La declaración del idioma en que se expresa la página y el enlace a la hoja de estilo.

3. Cuerpo de la página

Se escribe dentro de las etiquetas `<body ></body>` y corresponde a la parte central del documento. Aquí se expresara, elementos de

navegación, como lo es el menú de la página y el contenido de información que se quiera transmitir en ella, organizado en párrafos.

También en el cuerpo de HTML algunas veces contendrá código de javascript que ayudara a una mejor presentación a la página web, estará encerrado en las etiquetas de `<script type="text/javascript">` y `</script>`, cabe recalcar que el lenguaje de javascript no es lo mismo que el código de HTML.

Un ejemplo sencillo de código HTML es:

Cuadro 1

Ejemplo de HTML

```
<html>
<body>
<p>Esto es un párrafo. Bienvenidos a esta página web.</p>
</body>
</html>
```

Este sencillo ejemplo mostraría por pantalla lo siguiente.

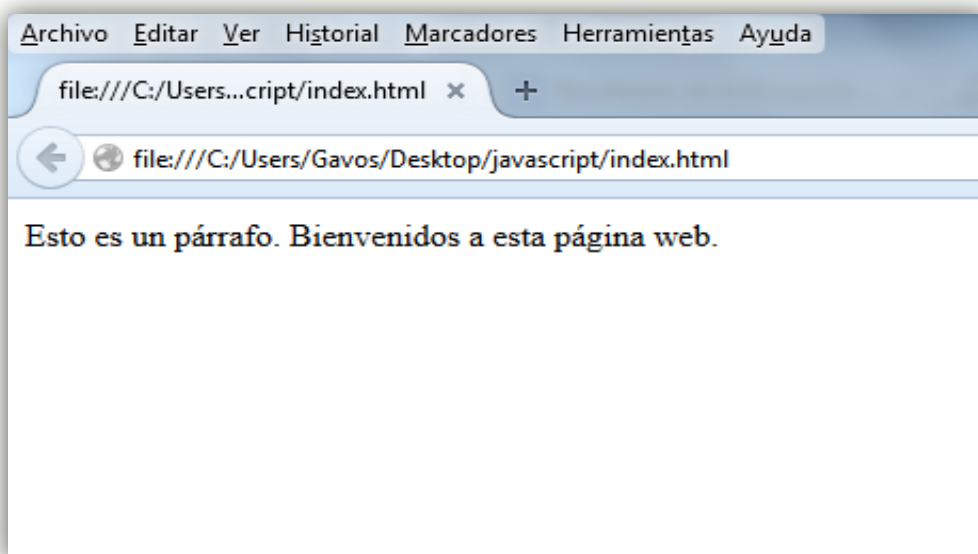


Figura 6: Inicio HTML

Si una etiqueta que se haya abierto no tiene su correspondiente cierre, se trataría de un código HTML mal construido, y los navegadores esto lo pueden interpretar de distintas maneras. Quizás nos muestren la página tal y como se espera sin aparente error. Quizás se muestre una página de error o se quede en blanco el navegador.

Clásicamente se dice que los lenguajes de programación incluyen tres capacidades básicas de generar flujos de procesos: la secuencial (secuencias de instrucciones), la condicional (capacidad para tomar decisiones o ejecutar un proceso u otro en función del valor de uno o varios parámetros) y la de repetición (capacidad para repetir un proceso un cierto número de veces). Los lenguajes clásicos como C, C++, Java, C#, Visual Basic, Fortran, etc. cuentan con estas capacidades. HTML no cuenta con ellas, no porque sea mejor ni peor sino porque es una cosa distinta.

En resumen, se puede decir que HTML no es un lenguaje de programación, es un lenguaje de maquetación web o lenguaje de etiquetas destinado a crear estructuras de documentos HTML. (González, 2011)

2.6.2 LENGUAJE CSS

CSS es un lenguaje utilizado en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente “una página web”. Así, se puede decir que el lenguaje CSS sirve para dotar de presentación y aspecto, de “estilo”, a una página web. Este lenguaje es principalmente utilizado por parte de diseñadores y programadores web para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, imágenes de fondo, bordes, etc.

La filosofía de CSS se basa en intentar separar lo que es la estructura o contenido y configuración básica del documento HTML de su presentación. Por decirlo de alguna manera: la página web sería lo que hay debajo (el contenido) y CSS sería “un cristal de color” que hace que el contenido se vea de una forma u otra. Usando esta filosofía, resulta muy fácil cambiarle el aspecto a una página web: basta con cambiar “el cristal” que tiene delante. Piensa por ejemplo qué ocurre si tienes un libro de papel y lo miras a través

de un cristal de color azul: que ves el libro azul. En cambio, si lo miras a través de un cristal amarillo, verás el libro amarillo. El libro (el contenido) es el mismo, pero lo puedes ver de distintas maneras. CSS permite cambiar el estilo a los contenidos de las páginas web.

Algunas opciones básicas del lenguaje CSS por ejemplo pueden ser el poder cambiar el color de algunas típicas etiquetas HTML como <H1> (h1 es una etiqueta en el lenguaje HTML destinada a mostrar un texto como encabezado, en tamaño grande). Pero también hay funciones algo más complejas, como introducir espaciado entre elementos <DIV> (div es una etiqueta HTML para identificar una determinada región o división de contenido dentro de una página web) o establecer imágenes de fondo, bordes redondeados, etc..

CSS es muy intuitivo y sencillo una vez se llega a aprender, ya que para su definición siempre se hace uso de un identificador de etiqueta HTML (como por ejemplo <H1>), y luego indicar con qué aspecto se quiere que se muestren todas las etiquetas <H1> que aparezcan en un documento. Se puede definir cómo se quiere que se muestren las distintas partes del documento HTML, pudiendo en cada caso definir sus propiedades (color, tipo de fuente, tamaño, espacio de márgenes, bordes, imagen de fondo, etc.) con algún determinado valor deseado.

Ejemplo sencillo de aplicación de CSS, el documento html contiene el siguiente texto de partida (en este caso se llama ejemplo.html):

Cuadro 2

Ejemplo CSS

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo aplicación CSS - aprenderaprogramar.com </title>
  </head>
  <body>
    <p>Texto de ejemplo para visualizar resultados </p>
  </body>
</html>
```

Con CSS se puede realizar efectos de color estilo y forma del contenido de una página web. (González, 2011)

2.6.3 LENGUAJE JAVASCRIPT

JavaScript es un lenguaje utilizado para dotar de efectos y procesos dinámicos e “inteligentes” a documentos HTML. Un documento HTML viene siendo coloquialmente “una página web”. Así, se puede decir que el lenguaje JavaScript sirve para ejecutar acciones rápidas y efectos animados en páginas web.

Las acciones controladas por JavaScript pueden ser el despliegue de un menú, hacer aparecer, desaparecer o cambiar texto e imágenes, realizar cálculos y mostrar resultados, mostrar mensajes de aviso (por ejemplo si faltan datos en un formulario) y “efectos animados” en general.

Este lenguaje es principalmente utilizado por parte de programadores web para dar respuestas rápidas a las acciones del usuario sin necesidad de enviar la información de lo que a hecho el usuario al servidor y esperar respuesta de éste (lo que haría más lento los procesos). El código JavaScript se carga al mismo tiempo que el código HTML en el navegador, y reside en el cliente (computador en el que nos encontramos), por lo que JavaScript sigue funcionando incluso aunque se produzca un corte en la conexión a internet (en este caso no se podrá seguir navegando hacia otras direcciones web, pero sí se puede ejecutar procesos “locales”).

El código JavaScript es interpretado directamente por el navegador web, sin necesidad de otros programas o procesos intermedios. Ejemplo

Cuadro 3

Muestra de texto en código javascript

```
<html>
<head>
<title>
<script lenguaje="JavaScript">
  //Visualizar un mensaje de bienvenida
  alert ("¡Bienvenido a nuestra página!");
</script>
</html>
```

2.6.4 LENGUAJE PHP

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo se va a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Se llama página estática a aquella cuyos contenidos permanecen siempre igual, mientras que páginas dinámicas son aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

Por regla general este tipo de lenguaje suele ser utilizado para crear contenido dinámico y poder interactuar con el usuario.

Ejemplo del contenido de una página web php, que habrá de almacenar en un archivo que termina con la extensión .php y que contiene texto:

Cuadro 4

Ejemplo página PHP

```
<html>
<head>
<title>Ejemplo</title>
</head>
<body>
<?php
echo "Bienvenidos a un breve ejemplo de la aplicación
del lenguaje php en html ";
?>
</body>
</html>
```


Como se pudo comprobar el contenido del archivo incluye código HTML y en un momento dado aparece una etiqueta `<?php` que significa que ahí empieza el código PHP. Por eso se dice que el código PHP está embebido (incluido) en código HTML

Por tanto las páginas PHP contienen HTML con código PHP incluido en ellas para hacer "algo" (en este caso, mostrar *Bienvenidos a aprenderaprogramar.com*). El código PHP está comprendido entre etiquetas de comienzo y finales especiales: `<?php` y `?>`

Estas etiquetas especiales permitirán entrar y salir del "modo PHP". Le sirven al servidor web para saber que tiene que interpretar el código incluido entre esas etiquetas como PHP.

Lo que distingue a PHP de JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente como si fuera una página web estática. El cliente recibirá los resultados que el servidor devuelve después de interpretar el código PHP, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. Es decir, a través de nuestro navegador podríamos ver el código HTML, pero nunca el código PHP que dio lugar al resultado HTML. El servidor web puede ser incluso configurado para que los usuarios no puedan saber si estás o no utilizando PHP.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales y más avanzados.

Con PHP puedes procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies, entre muchas más cosas. PHP lo utilizan desde pequeñas páginas web hasta grandes empresas. Muchas aplicaciones web están construidas usando PHP. Se puede citar Joomla y Drupal (gestores de contenido de páginas web), osCommerce (tiendas on-line para comercio electrónico), phpBB y SMF (sistemas de foros para páginas web), Moodle (plataforma educativa para educación on-line), etc.

Entre las habilidades de PHP se incluye la creación de imágenes a partir de datos. Por ejemplo, Una academia con 20 alumnos matriculados en un curso de informática, 50 alumnos matriculados en un curso de matemáticas y 30 alumnos matriculados en un curso de química. Con PHP se puede generar automáticamente imágenes similares a esta:

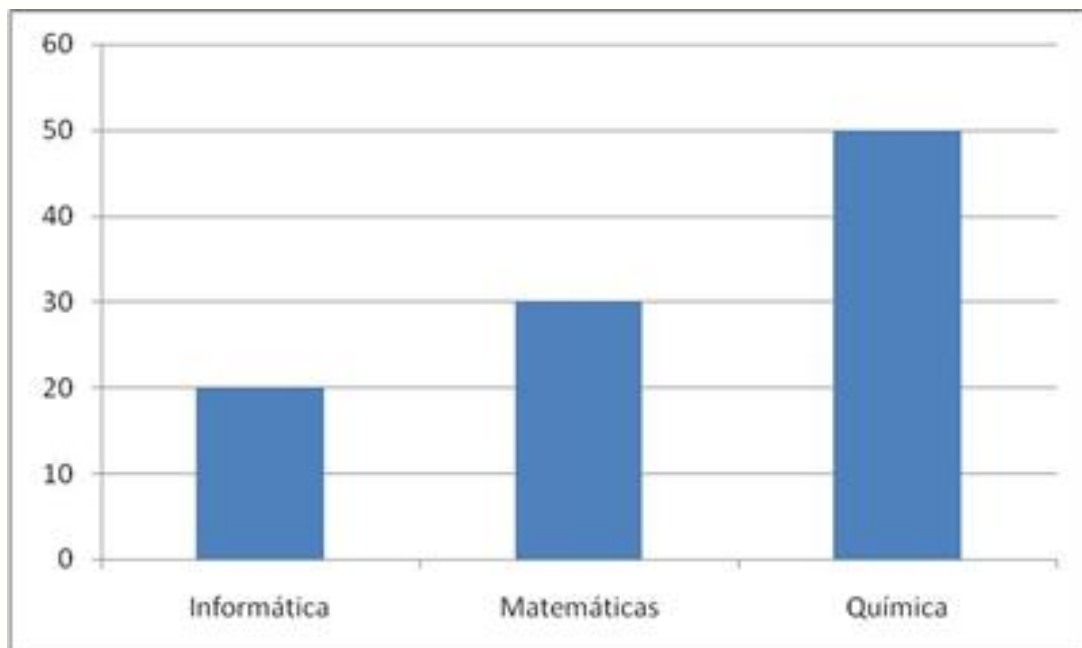


Figura 7: Generación de imagen con PHP

PHP también puede utilizar y presentar resultados en otros estándares de datos o lenguajes propios de los desarrollos web, como XHTML y cualquier otro tipo de ficheros XML. PHP puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla, utilizando estos ficheros para generar contenido dinámico. Es decir, el contenido dinámico puede surgir de otros sitios además de desde bases de datos.

También se puede interactuar con otros servidores usando cualquier protocolo. Por último, PHP puede enlazarse con otros lenguajes muy potentes como Java. (González, 2011)

2.7 SERVOMOTOR

Un Servo es un dispositivo pequeño que tiene un eje de rendimiento controlado. Este puede ser llevado a posiciones angulares específicas al

enviar una señal codificada. Con tal de que una señal codificada exista en la línea de entrada, el servo mantendrá la posición angular del engranaje. Cuando la señal codificada cambia, la posición angular de los piñones cambia. En la práctica, se usan servos para posicionar superficies de control como el movimiento de palancas, pequeños ascensores y timones.

Los Servos son sumamente útiles en robótica. Los motores son pequeños, tiene internamente una circuitería de control interna y es sumamente poderoso para su tamaño.

2.7.1 FUNCIONAMIENTO

El motor del servo tiene algunos circuitos de control y un potenciómetro (una resistencia variable) esta es conectada al eje central del servo motor. Este potenciómetro permite a la circuitería de control, supervisar el ángulo actual del servo motor. Si el eje está en el ángulo correcto, entonces el motor está apagado. Si el circuito chequea que el ángulo no es el correcto, el motor girará en la dirección adecuada hasta llegar al ángulo correcto. El eje del servo es capaz de llegar alrededor de los 180 grados. Normalmente, en algunos llega a los 210 grados, pero varía según el fabricante. Un servo normal se usa para controlar un movimiento angular de entre 0 y 180.

La cantidad de voltaje aplicado al motor es proporcional a la distancia que éste necesita viajar. Así, si el eje necesita regresar una distancia grande, el motor regresará a toda velocidad. Si este necesita regresar sólo una pequeña cantidad, el motor correrá a una velocidad más lenta. A esto se le llama control proporcional. (Murillo, 2011)



Figura 8: Estructura servomotor

2.8 CIRCUITO INTEGRADO

Un circuito integrado (CI), también conocido como *chip* o *microchip*, es una pastilla pequeña de material semiconductor, de algunos milímetros cuadrados de área, sobre la que se fabrican circuitos electrónicos generalmente mediante fotolitografía y que está protegida dentro de un encapsulado de plástico o cerámica. El encapsulado posee conductores metálicos apropiados para hacer conexión entre la pastilla y un circuito impreso.

2.8.1 CONTROLADOR DE MOTORES L298

Circuito controlador de dos motores de corriente continua o motores paso a paso bipolar. El integrado L298 posee en su interior dos puentes clase H lo que hace posible comandar dos bobinas independientemente, el control es de habilitación y de dirección, es decir se puede hacer girar un motor en un sentido o en otro y también se lo puede frenar.

El circuito tiene todo lo necesario como para funcionar con 12Vcc, Los diodos dependen de la frecuencia a la que funcione el circuito y del motor. (Rivera, Blog Jeferson Rivera, 2013)

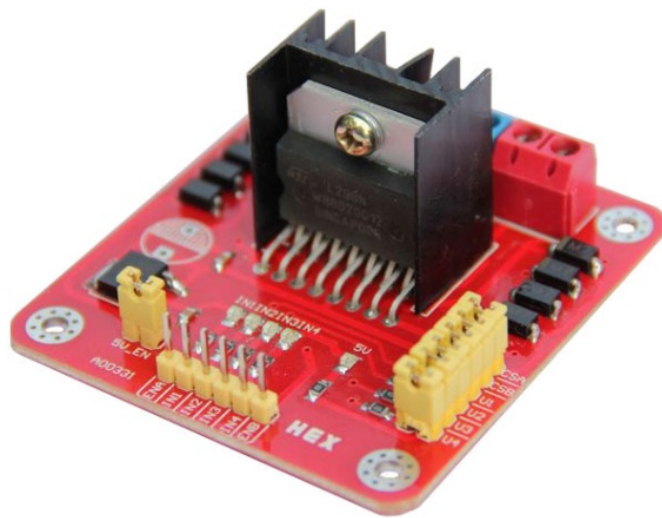


Figura 9: Circuito driver de motor L298

CAPÍTULO III

DESARROLLO DEL TEMA

3.1 INSTALACIÓN Y CONFIGURACIÓN DE RASPBIAN

Para dar uso de la placa Raspberry Pi se debe tener instalado su sistema operativo y estar previamente actualizado para que todos sus componentes funcionen correctamente al igual que una computadora normal, por tal motivo se detalla a continuación todo lo que se debe hacer antes de acceder al escritorio del Raspberry Pi.

3.1.1 INSTALACIÓN DE RASPBIAN

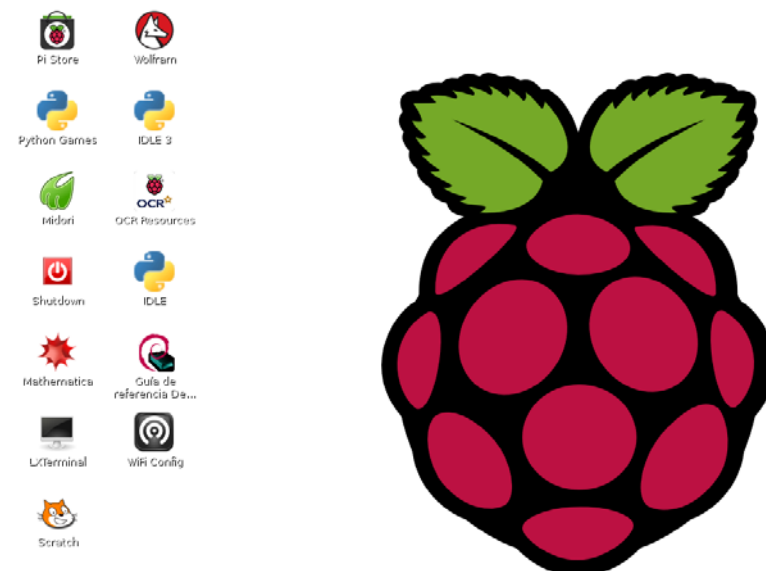


Figura 10: Escritorio Raspberry Pi

La instalación del sistema operativo en la tarjeta SD del Raspi se la realizara en Windows 7. No se va a tratar de una instalación al uso, sino de introducir una imagen en la tarjeta de memoria. Una instalación convencional puede llevar más de 10 horas si se quiere disponer de un entorno de escritorio, por lo que no es recomendado usar ese método.

1. Ir a la página web de descargas de Raspberry-Pi

<http://www.raspberrypi.org/downloads/>

2. Descargar una imagen del sistema operativo, de preferencia Raspbian.

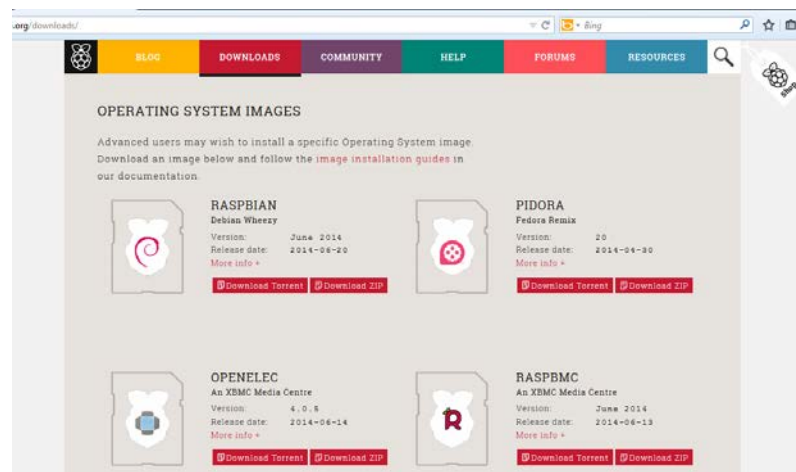


Figura 11: Página de descargas raspberry

3. Teniendo ya el archivo que pesa 700Mb como mínimo proceder a descargar Win32 Disk Imager, Una utilidad que permitirá cargar la imagen de Raspbian en la tarjeta SD.

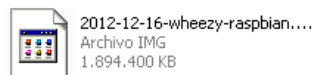


Figura 12: Software Raspbian

4. Abrir el archivo y extraer su contenido a cualquier carpeta, luego instalar y ejecutar.

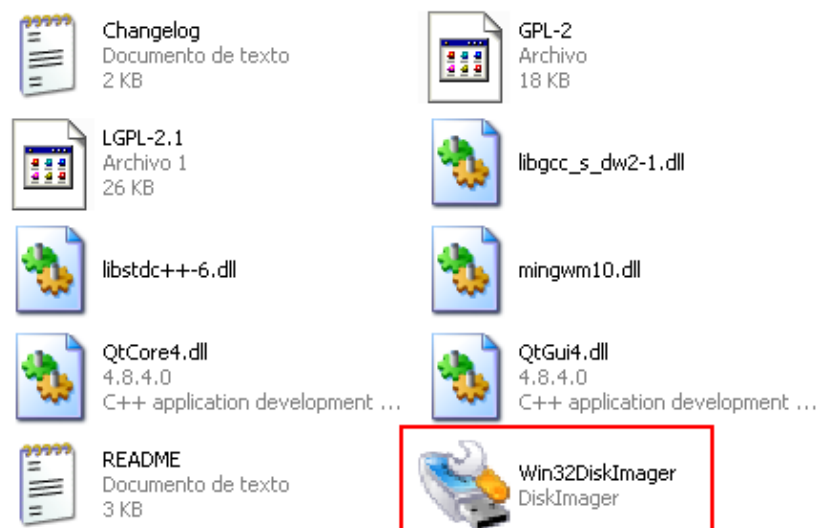


Figura 13: Programa Win32 Disk Imager

- Introducir la tarjeta SD en el lector de del ordenador, mostrara una imagen similar a esta, verificar que este seleccionada la SD card dando click en "Device".

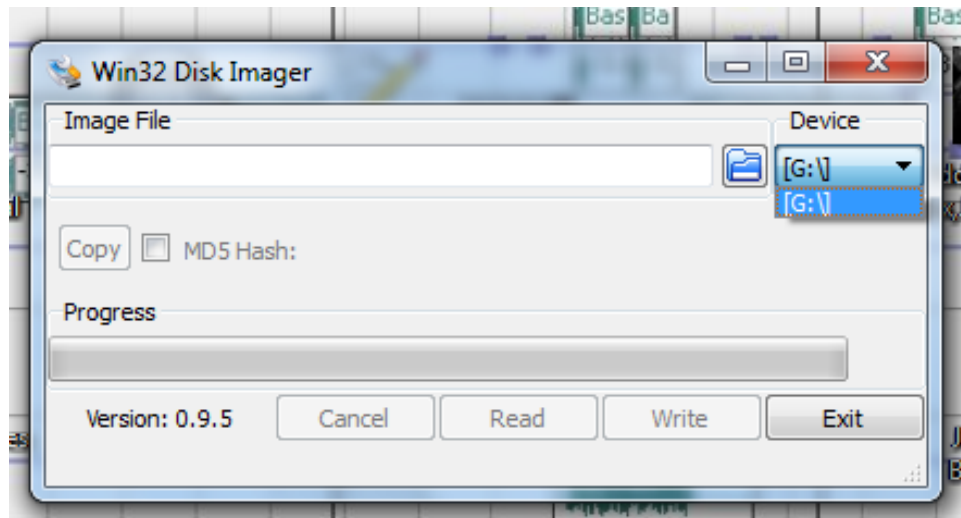


Figura 14: Win32 abierto

- Dar click en la carpeta buscar que está ubicada entre "Image file" y "Divice", seleccionar la imagen del sistema a instalar.

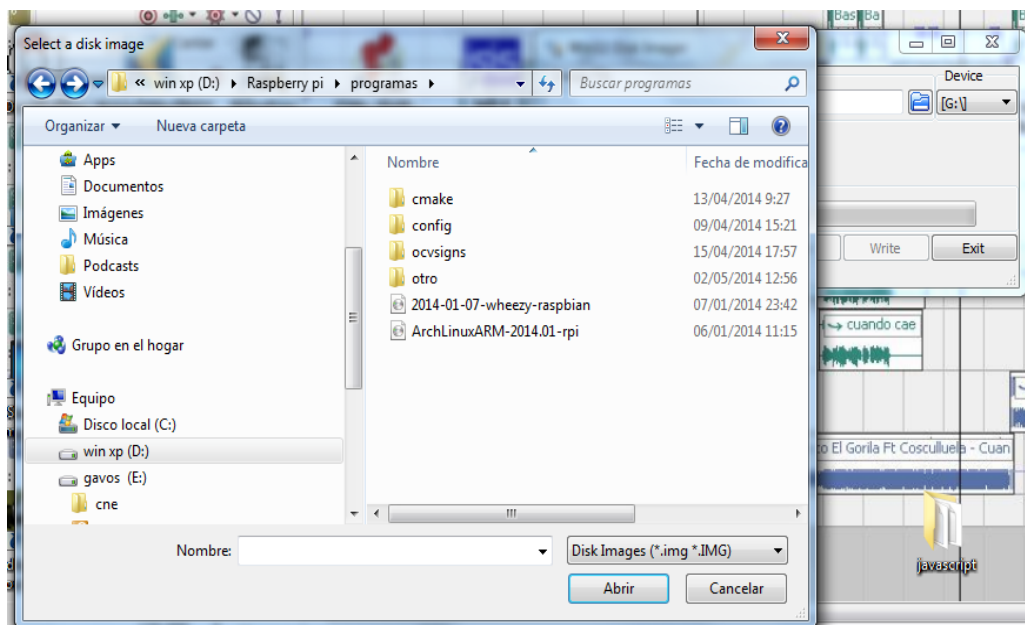


Figura 15: Selección del sistema operativo

- Dar un visto en el recuadro de "Copy", esperar unos minutos a que cargue y dar clic en "Write" y a esperar a que termine de cargar la imagen en la tarjeta.

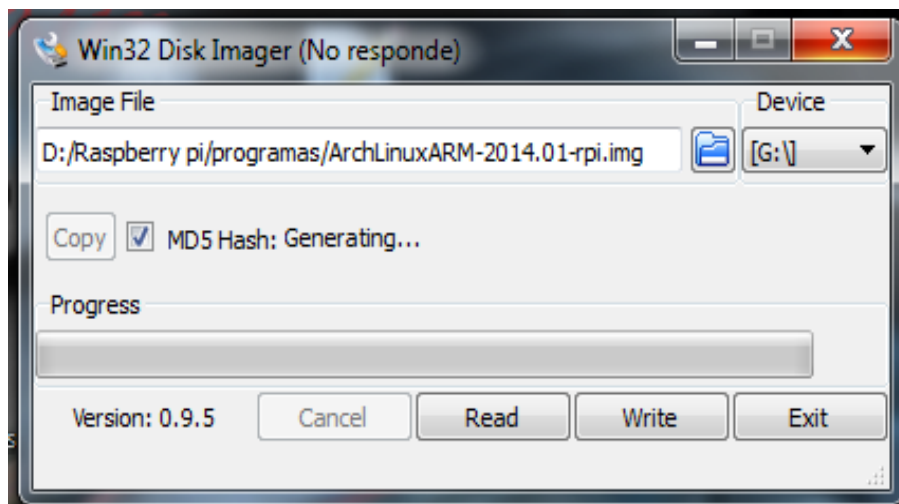


Figura 16: Escritura de Sistema Operativo

8. Una vez copiada la imagen, introducir la tarjeta SD en el Raspberry Pi, con este paso estaría listo para su configuración.

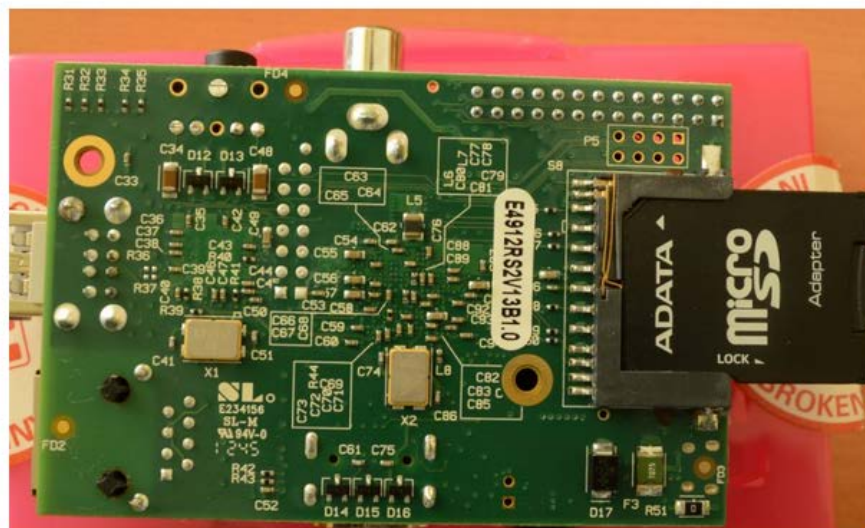


Figura 17: Micro SD y Raspberry Pi

9. Conectar la alimentación con un cargador que tenga salida micro usb de 5v y mínimo 700 mA y encenderá automáticamente.

3.1.2 CONFIGURACIÓN INICIAL DE RASPBIAN. (Bejarano, 2013)

Para configurar Raspbian se debe tener conectado principalmente un teclado y una pantalla. Al encender el Raspberry Pi por primera vez se mostrara directamente el menú que contiene diferentes opciones disponibles que son las siguientes:

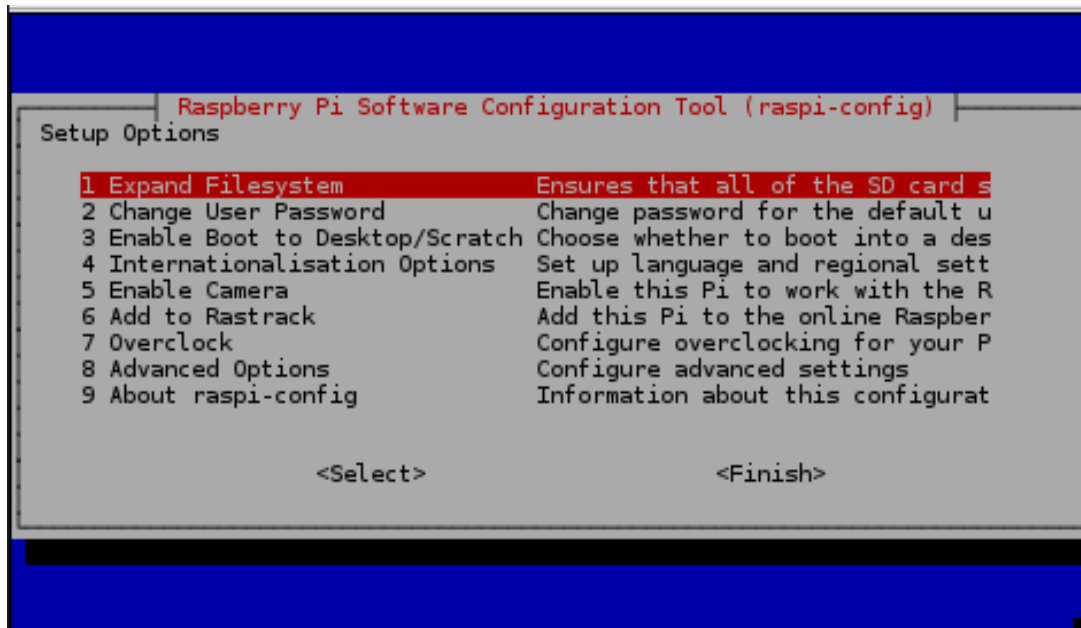


Figura 18: Menú principal

1. EXPANDIR EL SISTEMA DE ARCHIVOS (EXPAND FILESYSTEM)

Esta opción permite expandir el sistema operativo para que utilice todo el espacio disponible en la tarjeta. Cuando se instala Raspbian “Wheezy” la imagen copiada en la tarjeta solo ocupa 2 GB, por lo tanto es necesario ejecutar esta opción para que todo el espacio de la tarjeta SD sea utilizado.

2. CAMBIAR LA CONTRASEÑA DEL USUARIO PI (CHANGE USER PASSWORD)

En el Raspberry Pi y en general en sistemas Linux existen diferentes tipos de usuario, los dos que vienen predeterminados por el sistema son los usuarios “*root*” y “*pi*”.

El más importante que es el administrador del sistema que se llama “*root*”, este tiene acceso privilegiado a todos los archivos, configuraciones y carpetas del sistema. El otro tipo de usuario son los comunes como lo es “*pi*”, este viene predeterminado con la contraseña “*raspberrypi*” por lo tanto cualquier persona podría acceder su sistema. Por eso, es recomendable cambiar la contraseña en esta opción.

Al cambiar la contraseña el sistema le solicitara que ingrese la nueva contraseña y que la repita nuevamente, al finalizar espere un mensaje como el siguiente:

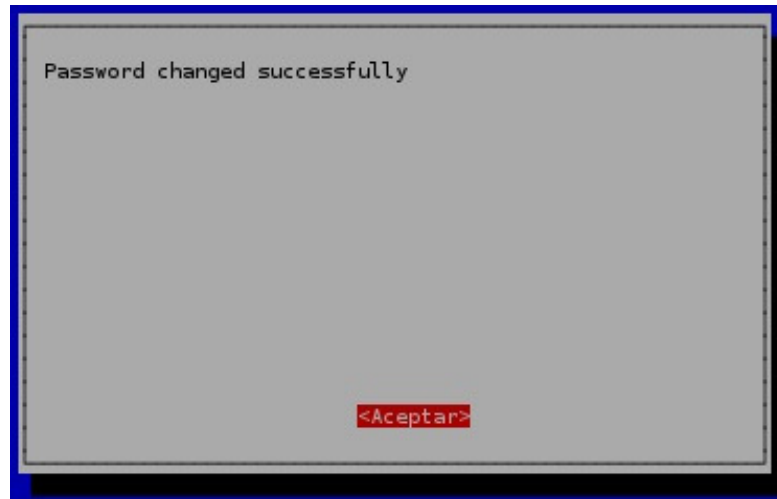


Figura 19: Cambio correcto de contraseña

3. ACTIVAR EL ESCRITORIO AL INICIAR (ENTABLE BOOT TO DESKTOP)

Esta opción permite que el Raspberry Pi después de iniciar el sistema, comience inmediatamente el escritorio modo gráfico o en línea de comando. En caso que inicie en modo de comando y después desee ingresar al modo gráfico solo ingrese el siguiente código en la terminal:

```
sudo stratx
```

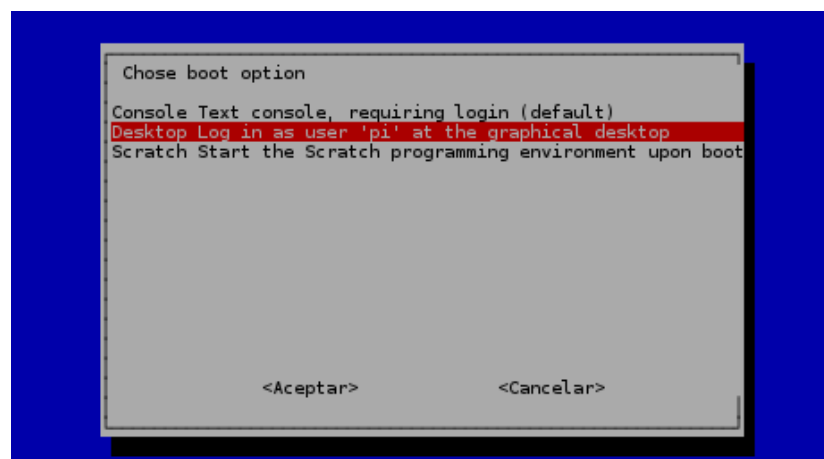


Figura 20: Activación de escritorio grafico

4. OPCIONES DE INTERNACIONALIZACIÓN (INTERNATIONALISATION OPTIONS)

Esta opción permite modificar el lenguaje del sistema operativo, la zona horaria y la distribución de su teclado, para este Raspi se va a establecer el lenguaje Español, con zona horaria de Guayaquil.

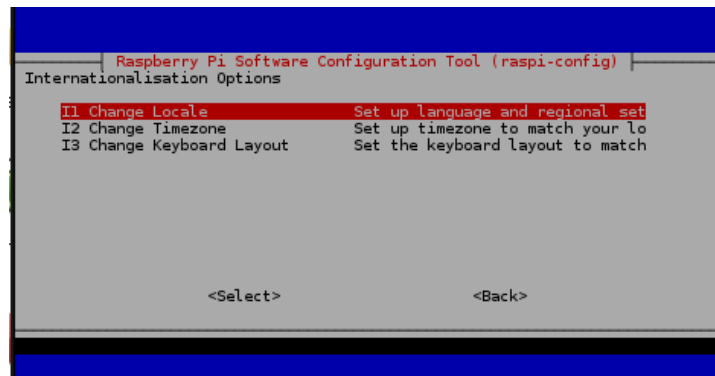


Figura 21: Cambio de localidad

5. CHANGE LOCALE

Sirve para indicar donde se encuentra ubicado, esta opción configura el lenguaje del sistema operativo, los caracteres, la denominación de la moneda, etc. Busque y seleccione la el modo de codificación:

es_ES. UTF-8 UTF-8

Para confirmar oprima espacio y luego enter.

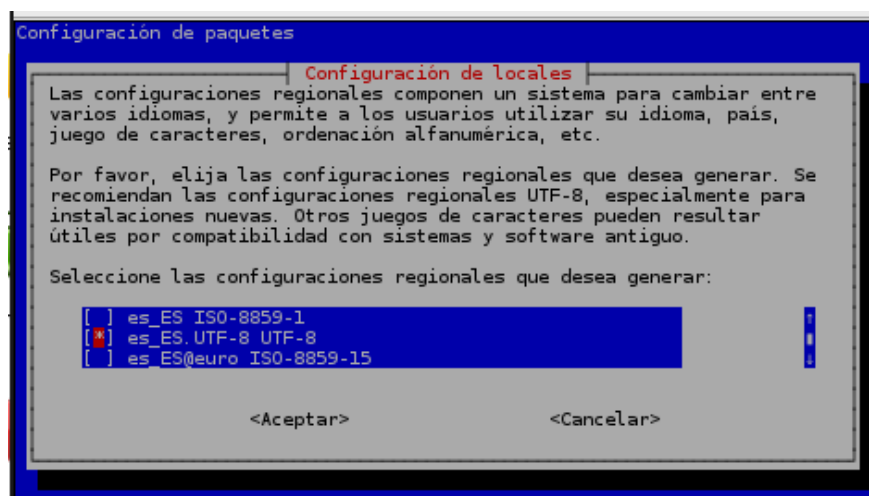


Figura 22: Selección de idioma del sistema

6. CHANGE TIMEZONE

Sirve para cambiar la zona horaria de su sistema la cual se ajusta de acuerdo a la ciudad donde vive o la más cercana. Primero elegir el área geográfica donde se está ubicado y seleccionar América.

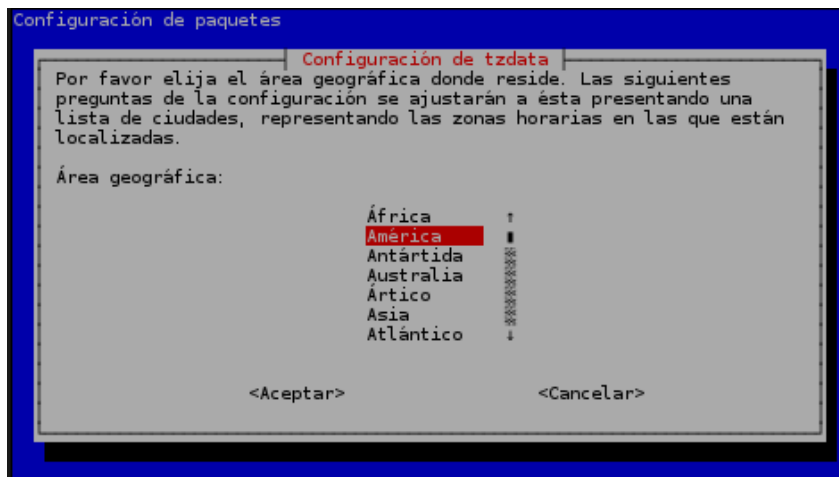


Figura 23: Selección de ubicación geográfica

Luego elegir una ciudad cercana de donde vive para establecer la zona horaria.

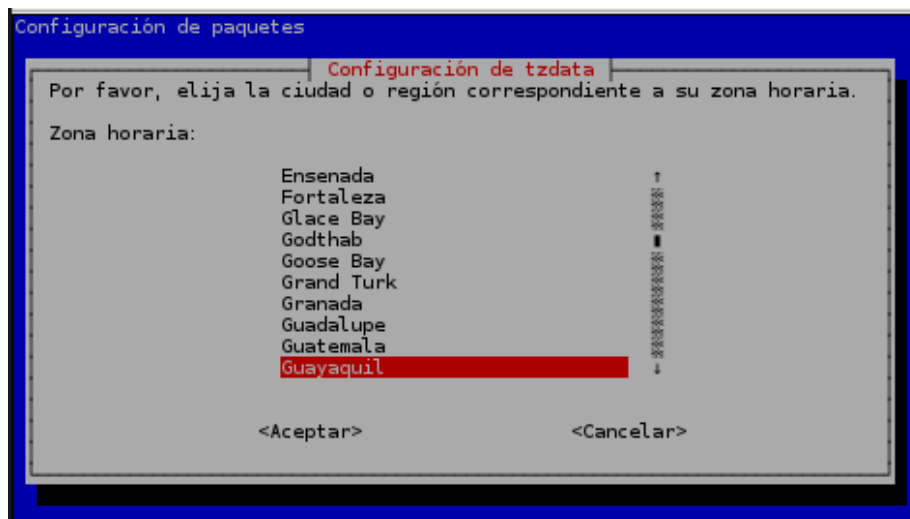


Figura 24: Selección de zona horaria

7. CHANGE KEYBOARD

Le permitirá cambiar la configuración de su teclado, si la marca y tipo de su teclado no aparece, seleccione el predeterminado "PC genérico 105 teclas (intl)".

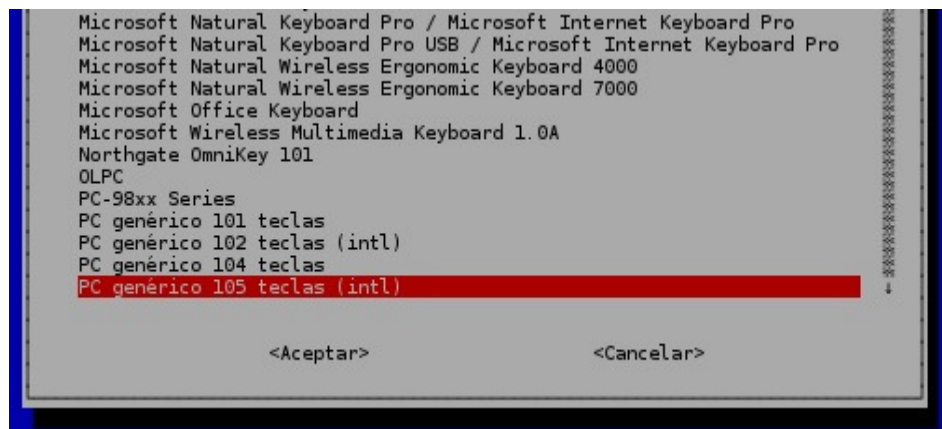


Figura 25: Selección de características del teclado

Luego seleccionar el idioma del teclado, si en la primera pantalla no aparece Español seleccione otro. En la siguiente ventana tendrá una lista de opciones seleccione “Español” o “Español (Latinoamericano)”.

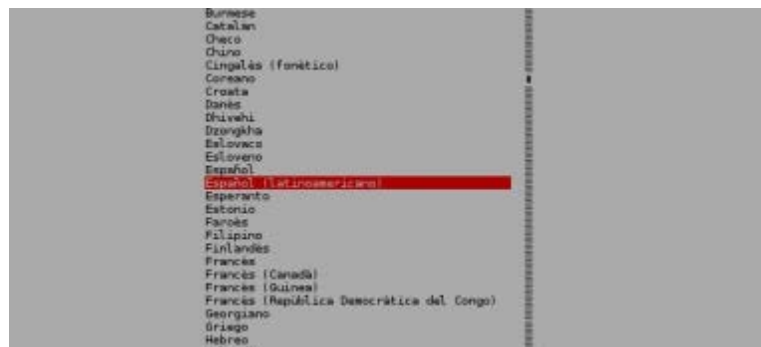


Figura 26: Selección de idioma del teclado

Luego seleccione la distribución de su teclado, se recomienda que utilice la primera opción solo “Español Latinoamericano”.

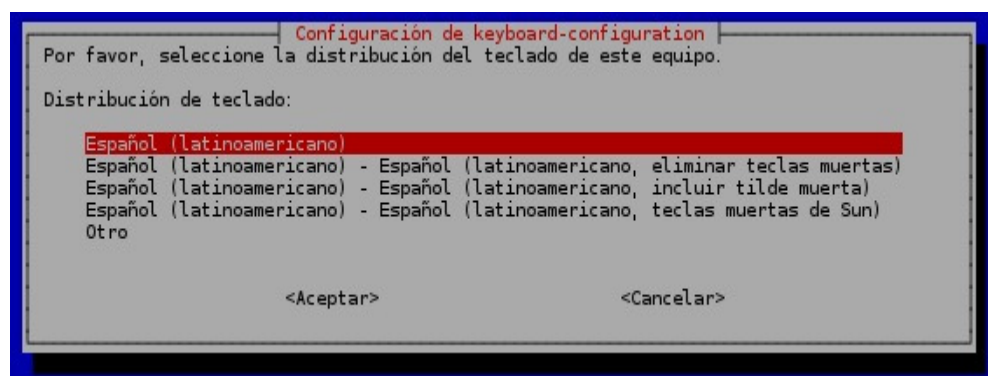


Figura 27: Características del teclado

Las siguientes dos ventanas le permiten configurar la tecla AltGr izquierdo y derecho para funciones especiales, si su teclado posee este botón seleccione la primera opción para que ejecute las opciones predeterminadas.



Figura 28: Características del teclado

Por último el sistema solicitará que si desea activar Control+Alt+Retorno para terminar el Servidor X (x server). Esto significa que si se encuentra en el modo gráfico y oprime esta combinación de teclas, el modo gráfico se cerrará y entrará a modo de comandos. Es recomendable dejar la opción predeterminada "No".

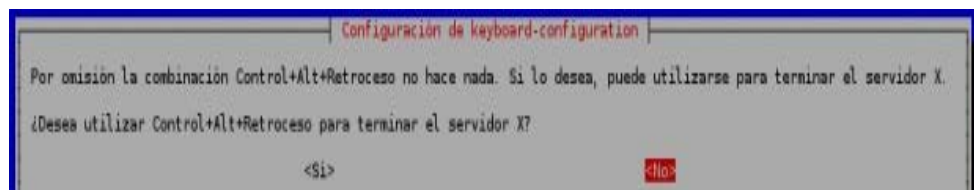


Figura 29: Finalización de configuración del teclado

8. ACTIVAR LA CÁMARA (ENABLE CAMERA)

Esta opción sirve para dar soporte a la cámara de Raspberry Pi, esta opción permite activar el puerto para que haya comunicación entre la CPU y el controlador de la cámara.

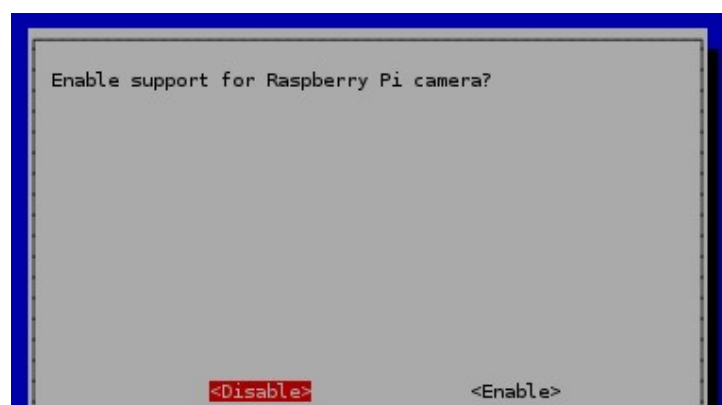


Figura 30: Activación de cámara.

9. ADICIONAR A RASTRACK (ADD TO RASTRACK)

Esta opción permite que su Raspberry Pi sea rastreado por el sitio web Rastrack (<http://rastrack.co.uk>), este sitio no pretende registrar o recolectar información alguna. Es una herramienta para tener la estadística de donde se encuentran los Raspberry Pi en el mundo. Es solo por diversión. Si desea que sea rastreado, solamente ingrese un apodo para identificar el Raspberry Pi y su dirección de correo.

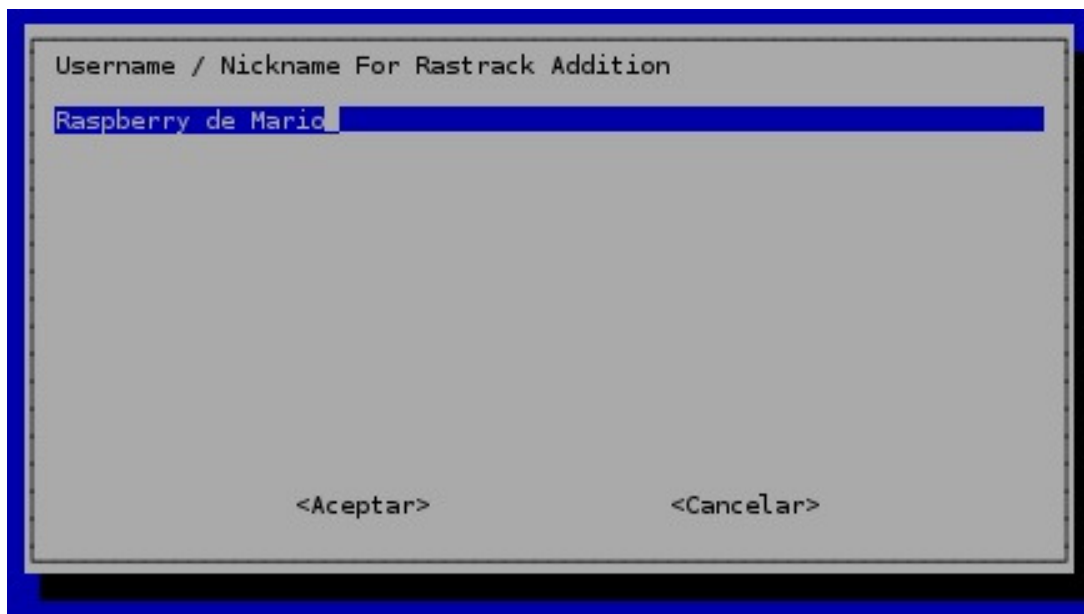


Figura 31: Activación de rastrack.

10. OVERCLOCKING

Esta opción permite aumentar la velocidad del procesador. Tenga en cuenta lo siguiente al modificar la velocidad. Primero, la vida del dispositivo se puede disminuir considerablemente. Segundo, el dispositivo generara más calor, por lo tanto es recomendable tener disipadores en el procesador, en circuito de Ethernet y en regulador de energía. Tercero, va a necesitar una fuente de poder de mayor capacidad para que pueda compensar la nueva velocidad seleccionada. Se recomienda dejar la predeterminada, pero si quiere tener más poder de cómputo siéntase libre de escoger la que más le convenga.

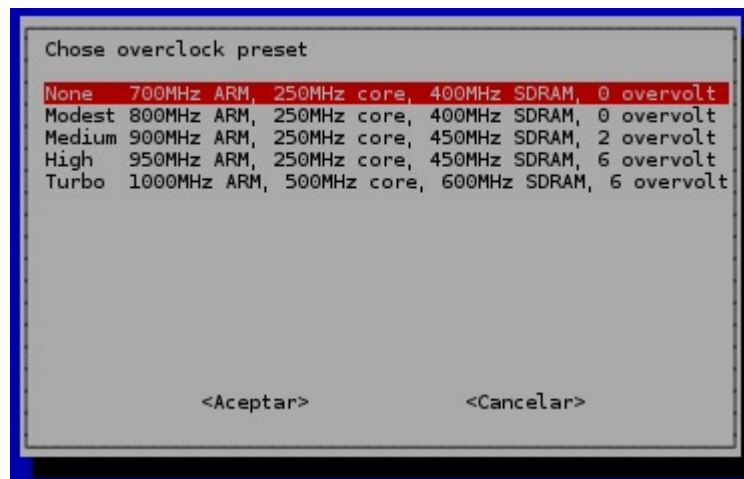


Figura 32: Selección de velocidad de proceso de datos.

11. OPCIONES AVANZADAS (ADVANCED OPTIONS)

Esta opción presenta otro submenú con las siguientes opciones.

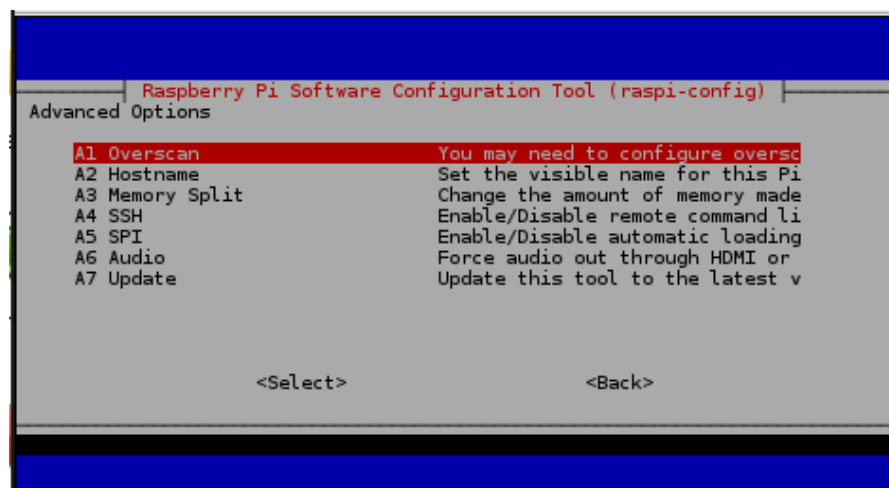


Figura 33: Menú de opciones avanzadas.

A1. OVERSCAN

Sirve para borrar las líneas negras en algunos monitores o televisores

A2. HOSTNAME

Sirve para identificar su Raspberry Pi en su red local, solamente utilice letras y números. Tenga en cuenta que el sistema diferencia mayúsculas y minúsculas, por lo tanto recuerde como escribe este nombre.

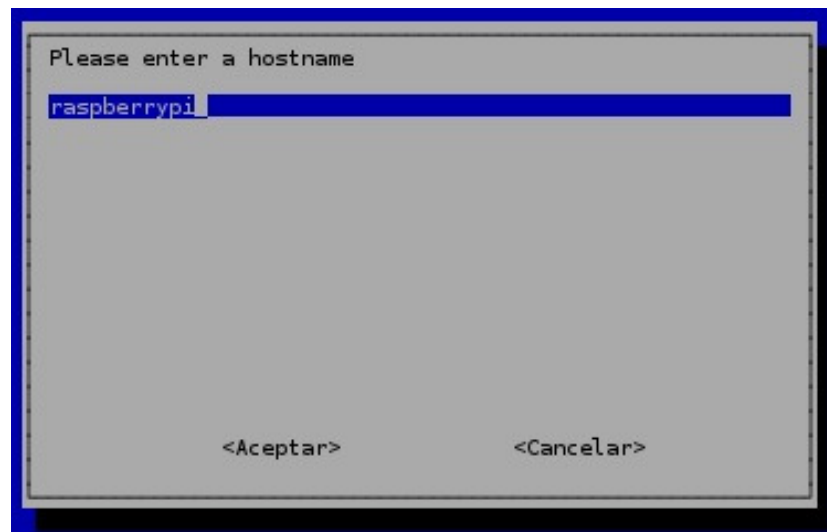


Figura 34: Cambio de nombre del dispositivo.

A3. DISTRIBUCIÓN DE LA MEMORIA (MEMORY SPLIT)

Le permite seleccionar la cantidad de memoria compartida entre la CPU y la unidad de gráficos (GPU), el modelo B versión 2.0 cuenta con 512 MB en total. El predeterminado es 64 MB para la memoria de vídeo, si piensa ejecutar aplicaciones que requieren alto procesamiento gráfico como ver películas o ver imágenes aumente el valor. Tenga en cuenta que al aumentar la memoria de vídeo y disminuir la del procesador, este se vuelve un poco más 'lento' para ejecutar otras tareas, es el precio que toca pagar por ejecutar mejor los gráficos.

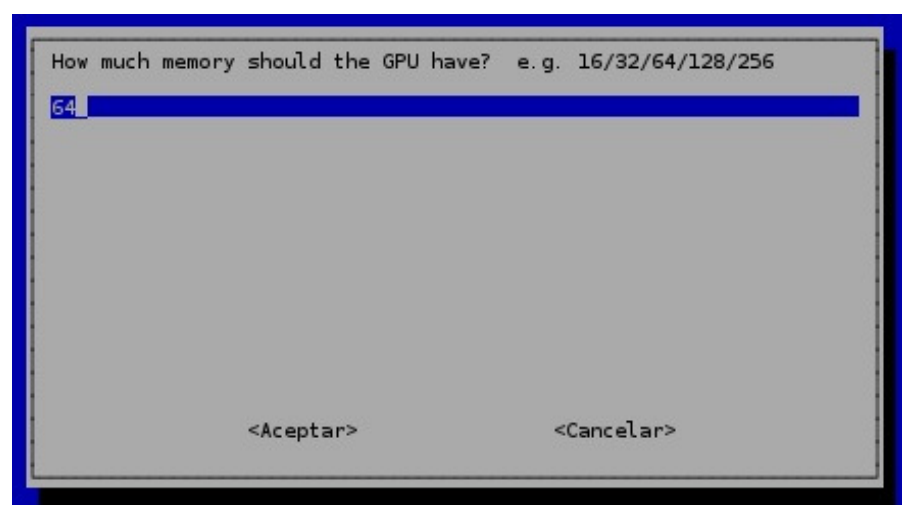


Figura 35: Memoria compartida.

A4. ACTIVAR SSH (ENABLE SSH)

Se utiliza para acceder el Raspberry Pi remotamente desde un cliente SSH. SSH significa “Secure SHell” el cual es una forma segura de conectarse al Raspberry Pi a través de la red, es recomendable activar esta opción, ya que con esto no necesitará utilizar ni un monitor, ni teclado, ni mouse adicionales para poder controlar su dispositivo.

A5. SPI

Está la opción permite activar o desactivar el acceso a la *Serial Peripheral Interface* (SPI).

A6. AUDIO

Permite configurar el tipo de audio que se obtendrá de la tarjeta.



Figura 36: Configuración de audio.

A7. ACTUALIZAR (UPDATE)

Se utiliza para que descargue una actualización del sistema, si ya se encuentra conectado a la red lo puede ejecutar inmediatamente. Si hay nuevas versiones de las librerías o programas se descargarán e instalarán las últimas versiones.

12. ACTUALIZACIÓN DE SOFTWARE COMPLETO

Una vez terminado con la configuración inicial es necesario la actualización de los programas y todos los componentes del Raspberry Pi que pueden ser necesarios para el correcto funcionamiento de los programas.

Esto se lo puede hacer mediante comandos ejecutados en la terminal seguidamente.

- `sudo apt-get update`
- `sudo apt-get upgrade`

Esto actualizará el software.

3.2 ACCEDIENDO AL ESCRITORIO DEL RASPBERRY PI

Para tener acceso al escritorio del Raspberry Pi existe 4 maneras, la primera y la más fácil sería utilizando la salida RCA Video y conectando en un televisor o en un proyector, este método sería el más factible porque no necesita ninguna configuración antes de utilizarla simplemente tener un cable de conexión y nada más pero con una baja resolución de pantalla por su señal analógica.

Las otras 3 maneras necesitan la configuración e instalación de unos complementos en Raspbian por lo cual se debe tener previamente instalado y actualizado el sistema operativo para utilizar los siguientes métodos que a continuación serán detallados.

1. CONEXIÓN SSH (SECURE SHELL). (Ramirez, 2013)

El trabajar con el ordenador, su teclado, ratón y con el monitor conectado a Raspberry Pi a veces se puede convertir en una pesadilla. Una sencilla solución es conectarse de manera remota sin ningún dispositivo más, solo el ordenador.

Hay que mencionar, que se deben tener la PC y Raspberry Pi conectados en la misma red. El sistema operativo sobre el que se explicará es Windows 7.

SSH™ (o Secure *SHell*) es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. Es un protocolo muy utilizado en los sistemas GNU/Linux.

Cuando se adentra en el mundo de Raspberry Pi se da cuenta que la mayoría de cosas se las realiza a través de su Terminal (consola de comandos). SSH va a permitir usar completamente un ordenador mediante una consola de comando. Es decir, se podrá tener acceso a la consola de comando de Raspberry Pi en cualquier ordenador.

Como Windows no tiene implementado este intérprete, se debe instalar un programa SSH. Se usará el programa más conocido para Windows, el programa PuTTY. (Ramirez, 2013)

For Windows on Intel x86				
PuTTY:	putty.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYtel:	puttytel.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSCP:	pscp.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSFTP:	psftp.exe	(or by FTP)	(RSA sig)	(DSA sig)
Plink:	plink.exe	(or by FTP)	(RSA sig)	(DSA sig)
Pageant:	pageant.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYgen:	puttygen.exe	(or by FTP)	(RSA sig)	(DSA sig)

www.tuElectronica.es

Figura 37: Fuente de descarga.

A continuación pasos para el acceso remoto:

- Descargar el programa y no necesita instalar solo se ejecuta como administrador.

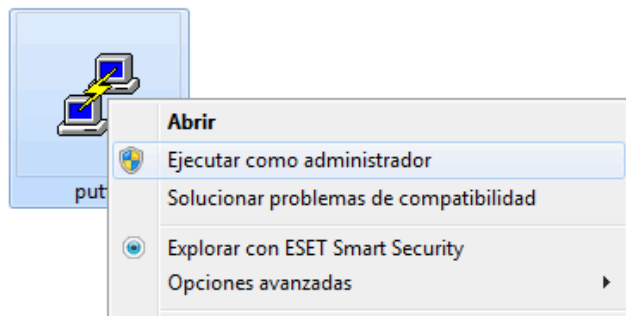


Figura 38: Ejecución de Putty.

- Abierto el programa, en el recuadro "Host Name (or IP address)" colocar la IP del Raspberry pi. Y para abrir la conexión dar clic en el botón *Open*.

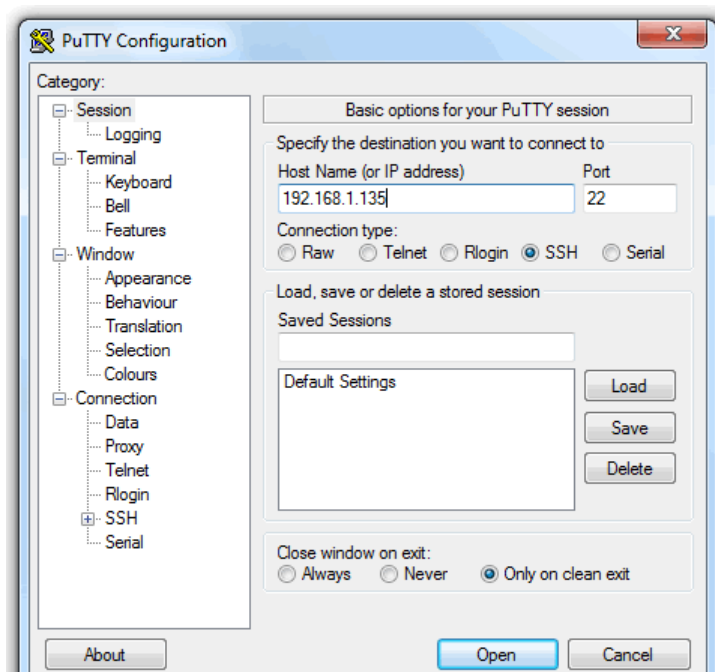


Figura 39: Introducir IP.

- Puede aparecer un mensaje de alerta, y hacer clic en *Sí*.

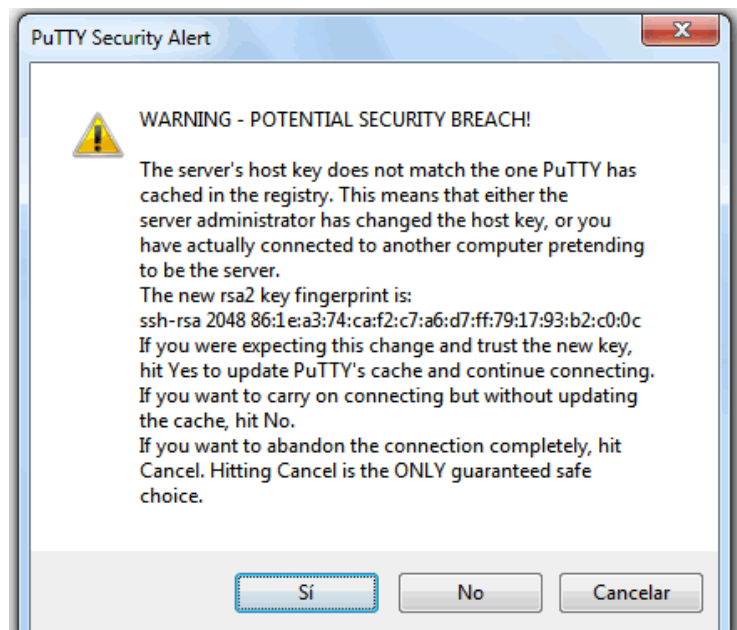


Figura 40: Advertencia.

- Si la conexión es correcta, pedirá una autenticación. poner los siguientes datos *pi* y pulsar *Enter*.

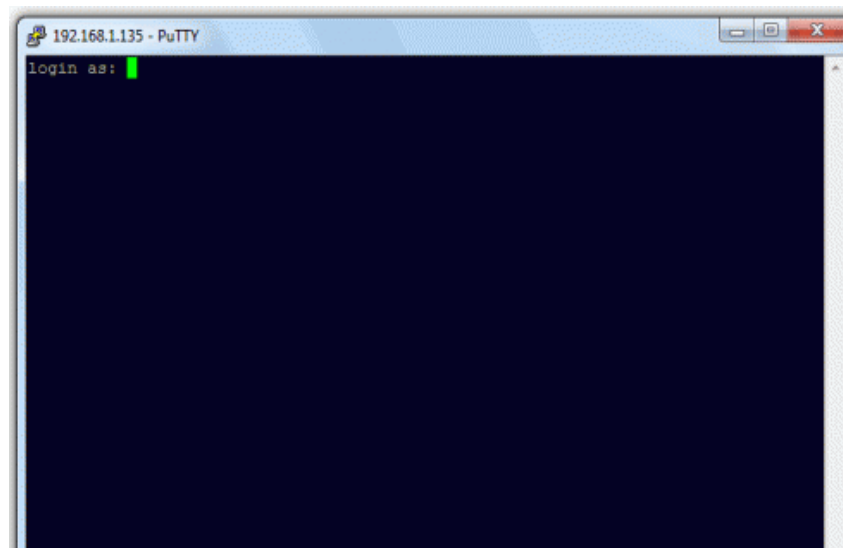


Figura 41: Datos de confirmación.

- Pedirá una contraseña, si no se cambio la contraseña en la configuración inicial se tendrá que poner la contraseña que viene por defecto y teclear *raspberry* y pulsar *Enter*.

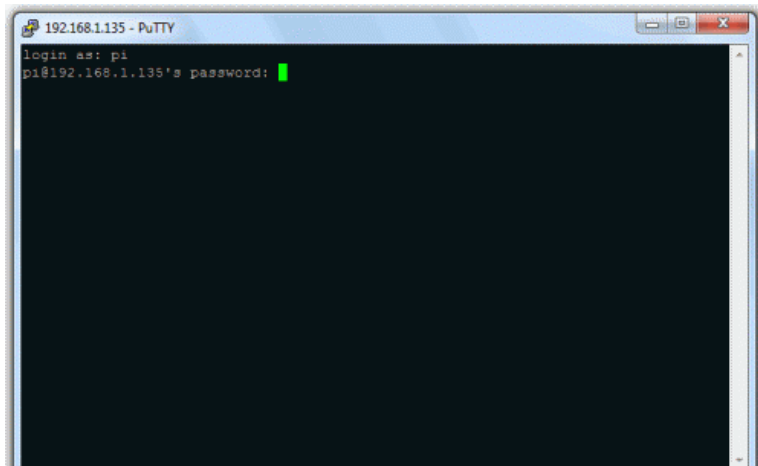


Figura 42: Datos de confirmación.

- Si todo salió bien, no deberá indicar ningún error y deberá salir como en la figura 43. Y ya se tendrá acceso a la consola de Raspberry Pi de manera remota. Desde aquí se puede realizar cualquier cosa que se requiera, pero hay que tener en cuenta que se necesita saber algo acerca de Linux como son sus comandos.

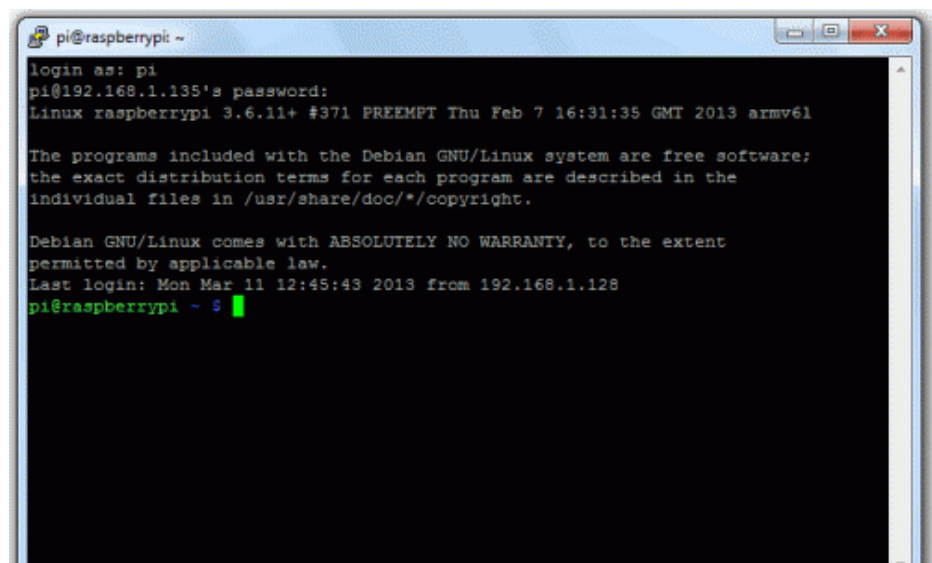


Figura 43: Consola de comandos.

2. ESCRITORIO REMOTO (VNC)

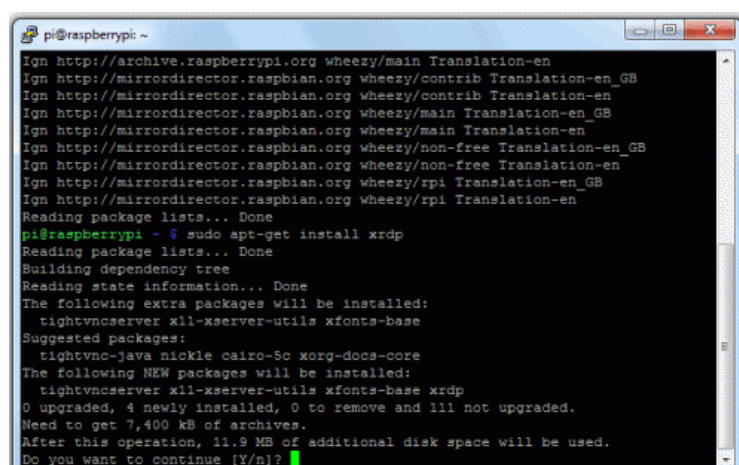
El escritorio remoto, o también llamado VNC (Computación Virtual en Red), permite usar el entorno gráfico de Raspberry Pi desde otro ordenador. Se tendrá acceso total a todo el contenido de forma visual, como si se estuviera viendo directamente el escritorio de Raspbian.

Para lograr esto, se necesita instalar un paquete en la Raspi.

- Abrir el terminal, se lo hará mediante una conexión SSH como se lo hizo en el apartado anterior.
- En la consola escribir lo siguiente, pulsar **S** y **Enter** si pide aceptar la instalación.

```
sudo apt-get update
```

```
sudo apt-get install xrdp
```



```

pi@raspberrypi ~
Ign http://archive.raspberrypi.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Reading package lists... Done
pi@raspberrypi ~$ sudo apt-get install xrdp
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  tightvncserver x11-xserver-utils xfonts-base
Suggested packages:
  tightvnc-java nickle cairo-1.0 xorg-docs-core
The following NEW packages will be installed:
  tightvncserver x11-xserver-utils xfonts-base xrdp
0 upgraded, 4 newly installed, 0 to remove and 111 not upgraded.
Need to get 7,400 kB of archives.
After this operation, 11.9 MB of additional disk space will be used.
Do you want to continue [Y/n]?

```

Figura 44: Instalación de xrdp.

- En Windows, ir a Inicio y en el cuadro de búsqueda poner Escritorio. Deberá aparecer el programa Conexión a Escritorio remoto. Hacer clic en él.

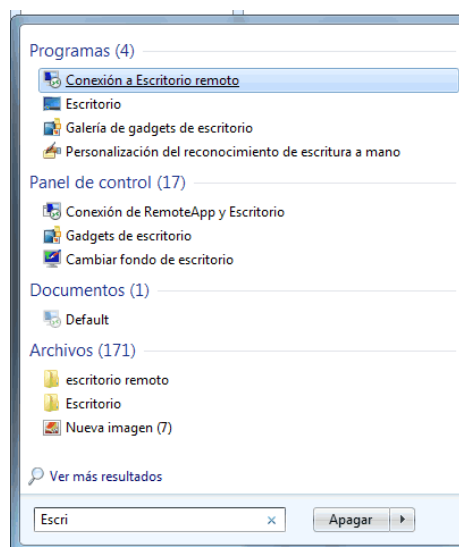


Figura 45: Búsqueda del escritorio remoto.

- Aparecerá una ventana en la cual se deberá ingresar la dirección IP del Raspberry Pi y dar clic en el botón Conectar.

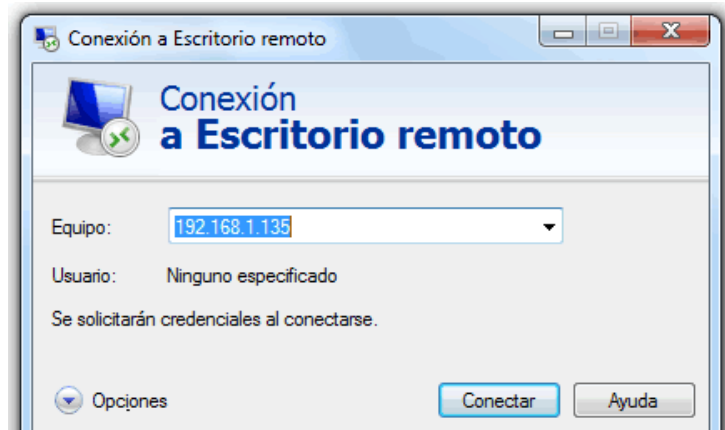


Figura 46: Introducir IP raspberry pi.

- Mostrará un mensaje de alerta, se marcará la casilla No volver a preguntar y hacer clic en el botón Sí.
- Deberá aparecer la siguiente ventana (figura 47) si todo está correcto, rellenar los datos username: pi; y password: raspberry; y hacer clic en OK.

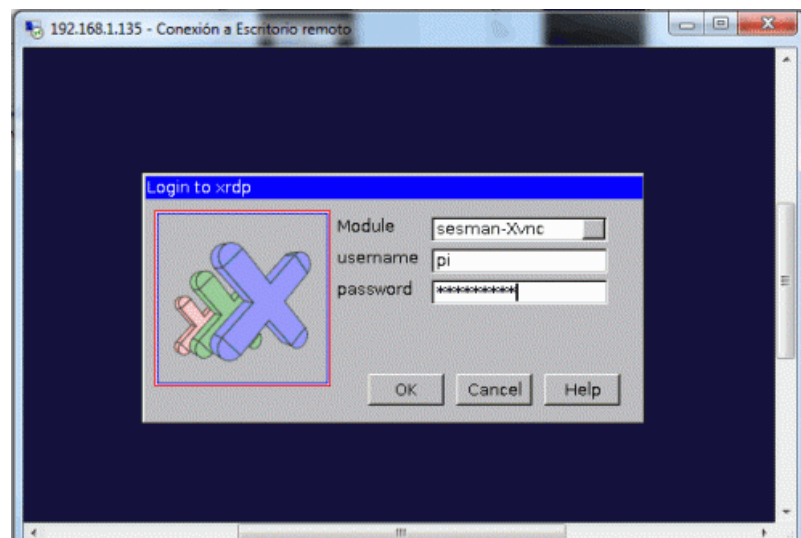


Figura 47: Datos de confirmación.

- Acceso de manera remota y visual al Raspberry Pi.

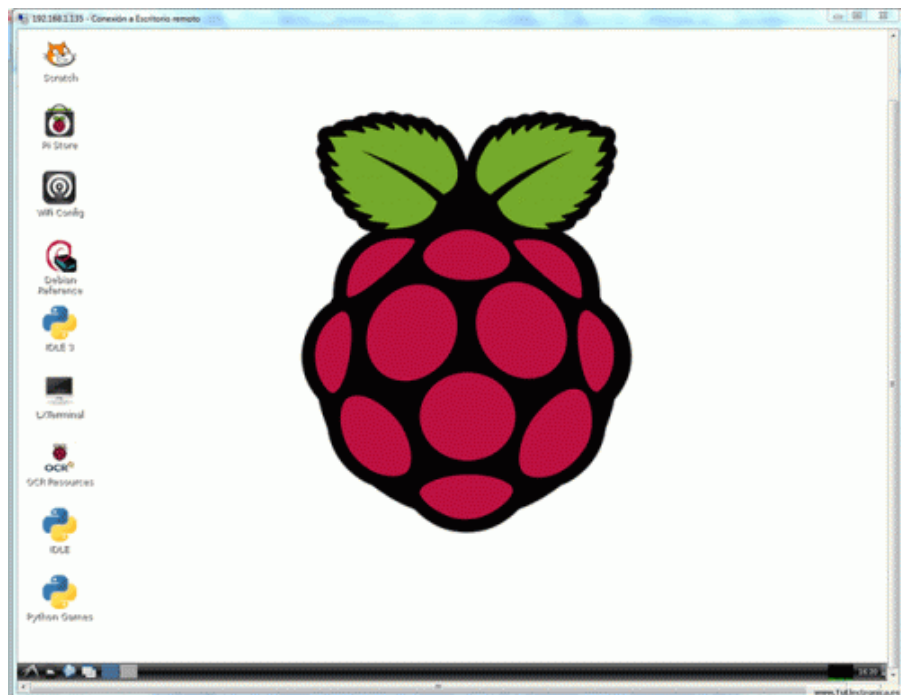


Figura 48: Escritorio de raspbian.

3. VISUALIZANDO EL ESCRITORIO CON UN MONITOR VGA

Para conectar una pantalla VGA se debe adquirir un convertidor de señal digital a señal analógica que tendrá como entrada HDMI y salida VGA.



Figura 49: Adaptador hdmi a vga

Con el adaptador presenta algunas complicaciones ya que en la actualidad existe una sola clase de adaptador en el país y consume una corriente y voltaje superior a la que le proporciona la salida HDMI del Raspberry Pi, pero existe solución (Esto se puede aplicar si el adaptador no funcionara correctamente).

Existe la posibilidad de alimentar el circuito del adaptador con una fuente externa o con la fuente de la GPIO. Lo primero que hay que hacer es abrir el adaptador y localizar el punto de alimentación del circuito. Para saber el pin del conector HDMI que se tiene que usar, se debe guiar en el siguiente esquema.

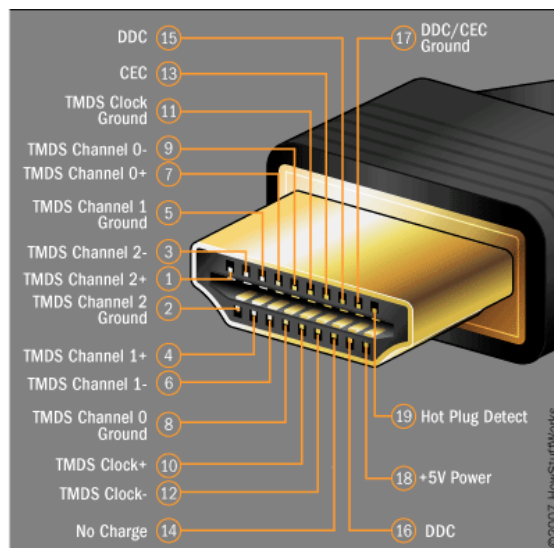


Figura 50: Adaptador hdmi a vga

En el pin 18 se localizó la alimentación a +5 V. Una vez encontrado el punto en la placa donde alimenta, se corta el cable que es de color lila. Luego soldar otro cable en el punto de alimentación de la placa del convertidor, y por medio de este alimentar externamente con el puerto GPIO que tiene en el pin 2 salida de 5v los cuales harán que funcione correctamente el circuito del adaptador.

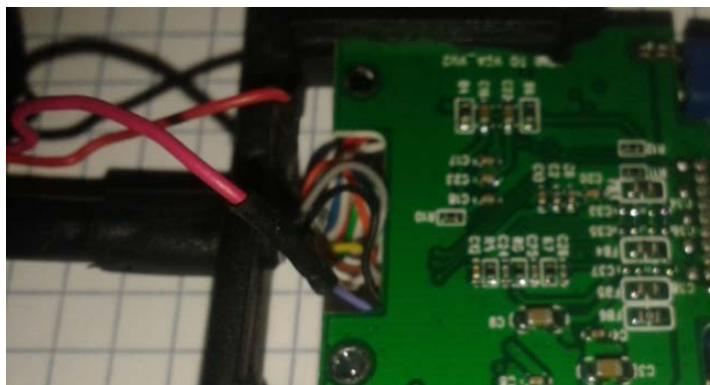


Figura 51: Conexión de cable en placa hdmi.

Con la conexión externa al adaptador HDMI a VGA solo basta con alimentar a la placa y funcionara correctamente.

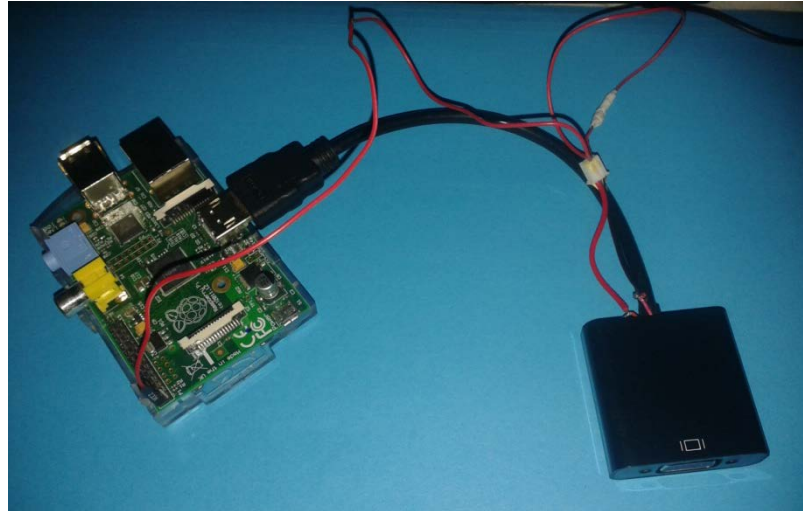


Figura 52: Alimentación externa.

3.3 TRABAJANDO CON LOS PUERTOS GPIO.

Hay dos maneras de hacer código en RPi. La primera es introduciendo línea por línea el código en la plataforma del Terminal ya sea en lenguaje C o Python y la segunda es crear un archivo en Python y luego compilarlo desde el Terminal.

Para poder entender claramente cómo trabaja el puerto GPIO se realizara un ejercicios en diferente programación.

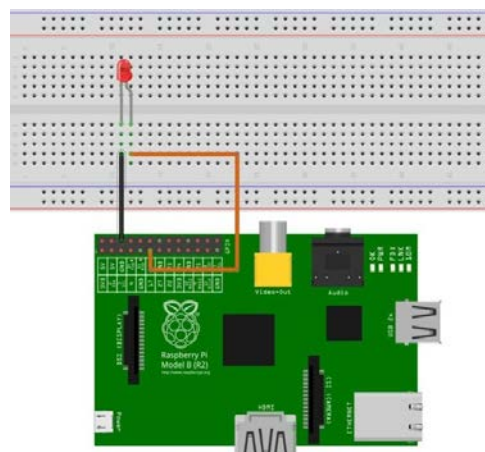


Figura 53: Diagrama para el ejemplo.

1. PROGRAMAR USANDO EL TERMINAL. (Fundacion, 2011)

Se debe hacer con la Raspberry Pi es conectar los cables al GPIO número 17, capaz de aportar 3V para encender el LED. El otro cable irá conectado al pin 6, es decir, el de tierra (Ground). Después recuerda conectar el cátodo (pin corto del LED) al cable de tierra y el ánodo (pin largo del LED) al GPIO 17 para aportar voltaje.

Ya lista toda la parte de la electrónica configurada, ahora pasar a la programación. Linux, en este caso Raspbian, es un sistema operativo UNIX. Y sin entrar en muchos detalles, la filosofía de estos sistemas es que todo es un fichero, incluido el hardware.

Lo que se quiere decir con esto es que los puertos GPIO y por tanto lo que haya conectado a ellos será tratado como un directorio más. Por eso, se puede utilizar comandos de manejo de directorios con nuestro hardware. En este ejemplo solo se escribirá comandos, pero se pueden implementar pequeños programas en algún lenguaje de programación o incluso crear script en Bash para manejarlo.

Si se accede al terminal de Raspbian en la Raspberry Pi, se puede crear una entrada al pin 17 para que el sistema cree un fichero con la estructura del GPIO correspondiente. De la siguiente forma:

```
echo 17 > /sys/class/gpio/export.
```

Ahora indicar al sistema que el pin se usará como entrada o como salida. En este caso tiene que ser de salida para que aporte un voltaje cuando se lo necesite y encienda el LED. Se escribe:

```
echo out > /sys/class/gpio/gpio17/direction.
```

Se encenderá en (1) o apagará en (0). Se Puede emplear las siguientes órdenes:

```
echo 1 > /sys/class/gpio/gpio17/value.
```

```
echo 0 > /sys/class/gpio/gpio17/value.
```

Una vez culminado el experimento, puedes borrar la entrada creada anteriormente para pasar a otra práctica. Para eliminar la entrada hay que teclear:

```
echo 17 > /sys/class/gpio/unexport.
```

2. PROGRAMAR USANDO PYTHON

Esta segunda opción es la que se recomienda, ya que en caso de cometer algún error, sólo es cuestión de abrir el archivo, hacer el cambio y volver a compilar. En el caso de la primera opción, tocaría volver a introducir línea a línea el código.

La aplicación para crear el archivo de python es IDLE, se lo ejecuta desde la terminal tecleando "sudo idle" para que no exista ningún problema a la hora de compilar y se crea el nuevo archivo. Para encender el led se escribe las siguientes líneas de código:

- `import RPi.GPIO as GPIO`

Hace referencia a la librería que se está importando, en este caso es la de GPIO que se necesita para hacer uso de los pines.

- `GPIO.setmode(GPIO.BCM)`
- Establece el sistema de numeración que se quiera usar, en este caso BCM.
- `GPIO.setup(17, GPIO.OUT)`

Es para facilidad de manejo de los pines, con el fin de no confundir con respecto al pin de uso, Además se le está diciendo al pin 17 que su configuración es de salida.

- `GPIO.output(17, True)`

Se le otorga un 1 al pin, es decir, está suministrando los 3V.

Una vez escritas estas líneas de código, se guarda el archivo, y se lo puede ejecutar desde el mismo programa python con la tecla F5 o desde la Terminal con el siguiente comando:

```
"sudo python NombreDelArchivo.py" y dar Enter.
```

```
pi@raspberrypi ~ $ sudo python ledon.py
pi@raspberrypi ~ $
```

Figura 54: Ejecución de programa py

Al ejecutar el comando se obtendrá el resultado de la figura 55.

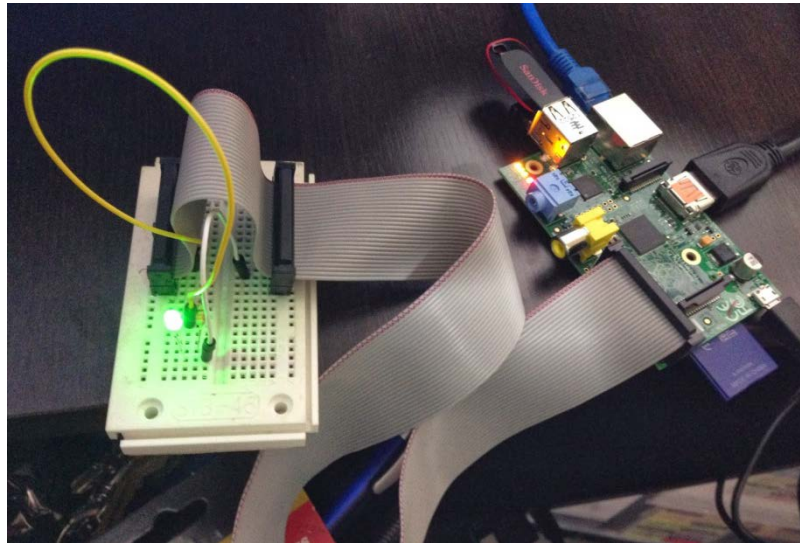


Figura 55: Resultado de practica

Y para apagarlo realizar otro archivo similar solo cambiando la última línea.

- `import RPi.GPIO as GPIO`
- `GPIO.setmode(GPIO.BCM)`
- `GPIO.setup(17, GPIO.OUT)`
- `GPIO.output(17, False)`

Ejecutar y listo.

3.4 INSTALACIÓN DE LA HERRAMIENTA WEBIOPi. (Google)

Pasos a seguir para la configuración. Ejecutar lo siguiente en la terminal:

- `wget http://webiopi.googlecode.com/files/WebIOPi-0.6.0.tar.gz`
- `tar xvzf WebIOPi-0.6.0.tar.gz`
- `cd WebIOPi-0.6.0`

- `sudo ./setup.sh`

Una vez instalado configurar para que se ejecute al iniciar la Raspberry Pi

- `sudo update-rc.d webiopi defaults`

Y ejecutar:

- `sudo /etc/init.d/webiopi start`

Ahora se puede ir a un navegador e introducir la URL que sería la IP mas el puerto que utiliza la herramienta que es el puerto 8000 ejemplo:

- `192.168.1.102:8000`

Introducir el usuario: `webiopi` y la contraseña: `raspberry`. Al haber ingresado en la dirección se desplegara un menú en donde permitirá el monitoreo y control de los pines del GPIO (Figura 56) o algunas otras opciones que brinda la herramienta `webiopi`.

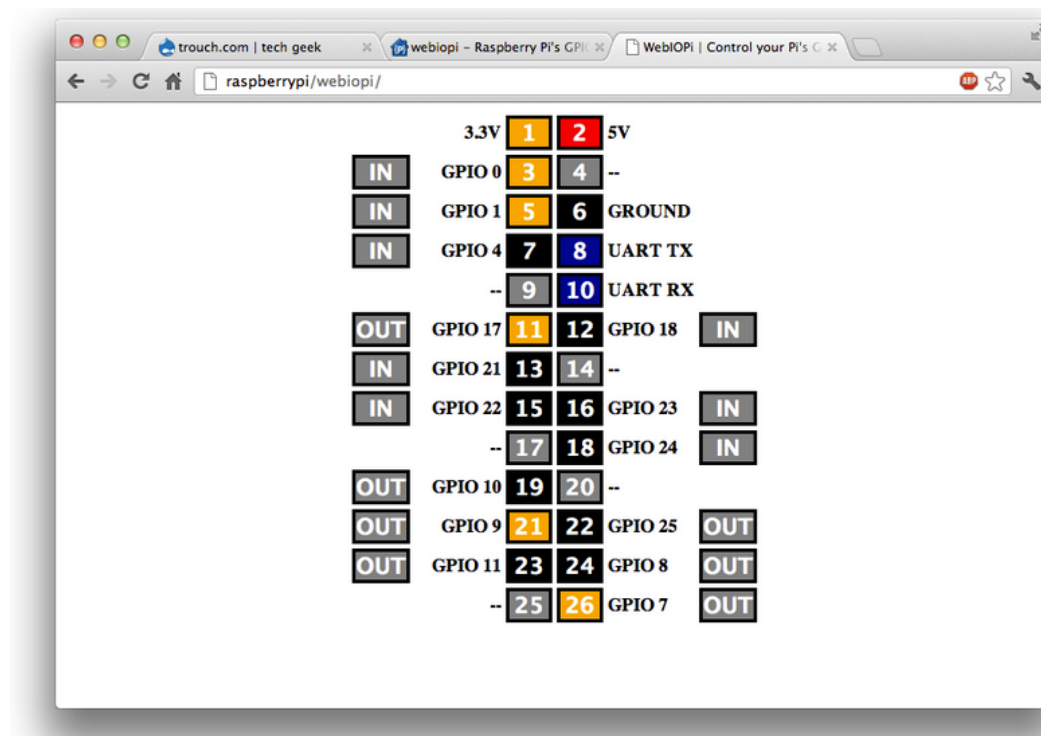


Figura 56: Monitoreo de funcionamiento del GPIO

3.5 INSTALACIÓN DEL SERVIDOR WEB.

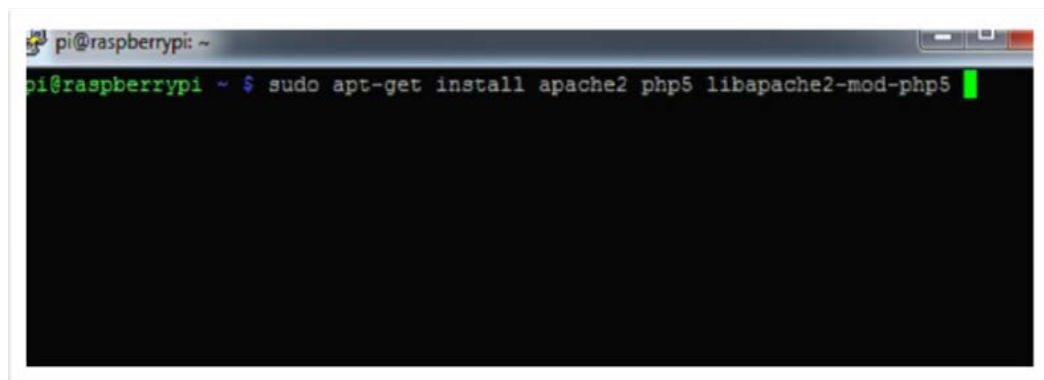
Instalar y configurar servidor APACHE, con PHP y MYSQL, esto usando la conexión remota SSH. (Rivera, Blog Jefferson Rivera, 2013)

1. Verificar que todo el sistema este actualizado

- `sudo apt-get update`
- `sudo apt-get upgrade`

2. Instalar Apache y PHP

- `sudo apt-get install apache2 php5 libapache2-mod-php5`



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ sudo apt-get install apache2 php5 libapache2-mod-php5
```

Figura 57: Instalación del servidor

3. Reiniciar apache

- `sudo /etc/init.d/apache2 restart`



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ sudo /etc/init.d/apache2 restart
```

Figura 58: Reinicio servidor

4. **Otorgamiento de permisos.**- Los directorios típicamente utilizados por un servidor web en linux se sitúan en `/var/www`, y el usuario típico para dicho entorno suele ser `www-data`. Ahora se va a crear el grupo y usuario estándar para nuestro servidor, a la par que se otorga los

permisos pertinentes y añadir el usuario por defecto (pi) al grupo comentado. De esta forma no será preciso que el usuario root (super usuario) sea siempre el que pueda operar en /var/www.

5. Cambiar el usuario y grupo al directorio

- `sudo chown www-data:www-data /var/www/`

6. Dar los permisos a la carpeta www/

- `sudo chmod 775 /var/www/`

7. Añadir el usuario pi al grupo www-data

- `sudo usermod -a -G www-data pi`
- `sudo visudo`

8. Agregar el siguiente código al final.

- `www-data ALL=(root) NOPASSWD:ALL`

9. Reiniciar el Servidor

- `sudo /etc/init.d/apache2 restart`

10. Probar desde cualquier navegador Web. Colocando en la URL la IP de la Raspberry PI.

- `http://192.168.1.5/`



Figura 59: Servidor web

11. Para probar si funciona correctamente crear un archivo de prueba, Hola mundo en PHP.

- `sudo nano /var/www/hola_mundo.php`
- `<?php`
- `echo "HOLA MUNDO DESDE RASPBERRY PI";`
- `?>`



The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. The window contains the GNU nano 2.2.6 editor editing the file '/var/www/hola_mundo.php'. The code entered is: `<?php`, `echo "HOLA MUNDO DESDE RASPBERRY PI";`, and `?>`. The bottom of the window shows the nano editor's command palette with options like Get Help, WriteOut, Read File, etc.

Figura 60: Creación hola mundo

12. Ejecutar con la siguiente línea

- `sudo php /var/www/hola_mundo.php`



Figura 61: Abriendo hola mundo en la web

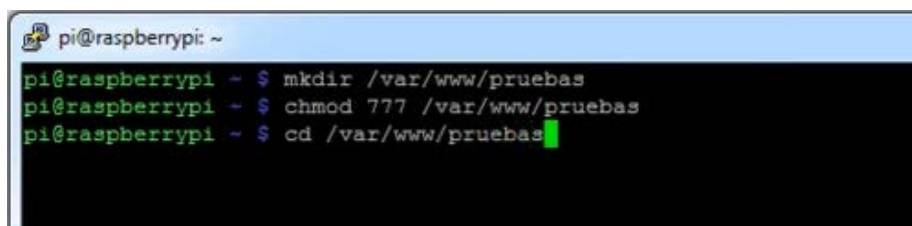
3.6 ACCEDIENDO AL GPIO DESDE PHP.

En PHP existe una función llamada "exec" la cual ejecuta un programa externo. Esta función se la utilizara para ejecutar programas hechos en Python, script y algunas otras funciones que estén en la Raspebrry Pi.

Ejemplo de uso.

Para ejecutar un script de Python se realizara lo siguiente:

1. Dentro de la carpeta `cd /var/www/` crear una nueva carpeta con el nombre de "pruebas" y se le otorgara los respectivos permisos de súper usuario para que no tenga ningún inconveniente a la hora de ejecutar programas dentro de la carpeta.



```

pi@raspberrypi: ~
pi@raspberrypi ~ $ mkdir /var/www/pruebas
pi@raspberrypi ~ $ chmod 777 /var/www/pruebas
pi@raspberrypi ~ $ cd /var/www/pruebas
  
```

Figura 62: Creación de carpeta pruebas

2. Ya dentro de la carpeta crear el archivo
 - `sudo nano prueba_led.php`
3. Ingresar las siguientes líneas de código y guardar.
 - `<?php`
 - `$a=exec('sudo python /var/www/pi/leds/prende.py');`
 - `echo $a;`
 - `?>`
4. En la misma carpeta crear el archivo `prende.py` que será accionado con php, guardar y está listo. El script `prende.py` contiene:
 - `import RPi.GPIO as GPIO`
 - `GPIO.setmode(GPIO.BCM)`
 - `GPIO.setup(17, GPIO.OUT)`

- GPIO.output(17,True)
5. Y para ejecutarlo simplemente llamar al script php en la misma carpeta.
 - sudo php pruebas_led.php
 6. Ir a cualquier navegador e introducir la dirección IP del Raspberry Pi más la carpeta pruebas así:
 - 192.168.1.10/pruebas
 7. En el navegador aparecerá la siguiente imagen donde hay que dar click en prende.py para que se encienda el led.

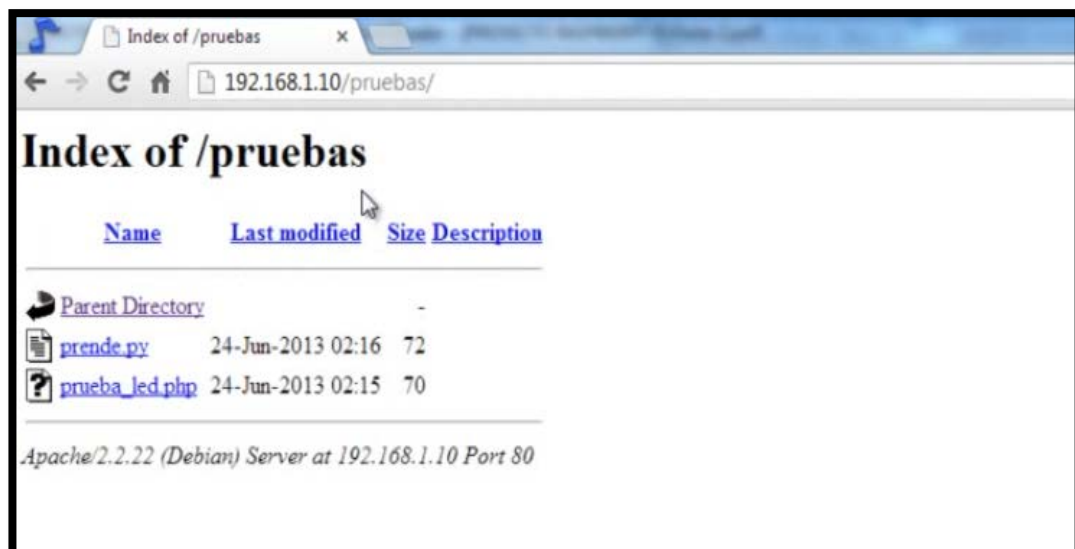


Figura 63: Abriendo hola mundo en la web

3.7 DISEÑO DE SOFTWARE DEL ROBOT MÓVIL

En esta sección se refiere al diseño y características de los diferentes programas que servirán de enlace entre el usuario y el control del robot. Para la creación del software del robot se utilizara todos los recursos mencionados anteriormente.

- Raspbian
- Python
- HTML
- PHP

- CSS
- Javascript
- Apache

3.7.1 PÁGINA WEB



Figura 64: Página web Raspibot

La página contiene archivos que permiten la interfaz entre una PC con el Raspberry Pi, está ubicada en una carpeta llamada Raspibot y alojada en la sección `"/var/www/Raspibot/"` y consta de las siguientes carpetas y archivos.

1. ARCHIVO INDEX.HTML

La parte principal de toda la página web y para su edición se utilizara el programa Notpad++ que es un editor de texto y de código libre con soporte para varios lenguajes de programación. Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple.

El archivo index contiene lo siguiente:

- El encabezado de HTML.
- Etiquetas de java script.
- Etiquetas de CSS.

```

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Control Remoto</title>
  <script src="js/jquery-1.9.1.js"></script>
  <link rel="stylesheet" type="text/css" href="css/estilo.css" />
  <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
</head>

```

Figura 65: Encabezado HTML

Declaración respectiva de los botones que constituyen la página web del robot con su referencia respectiva de imagen que contendrán los mismos.

Cuadro 5

Declaración de botones en HTML

```

<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>

```

Código de javascript introducido en html que ayudara a la mejor presentación de la estura de la página web.

Cuadro 6

Programación de javascript en HTML

```

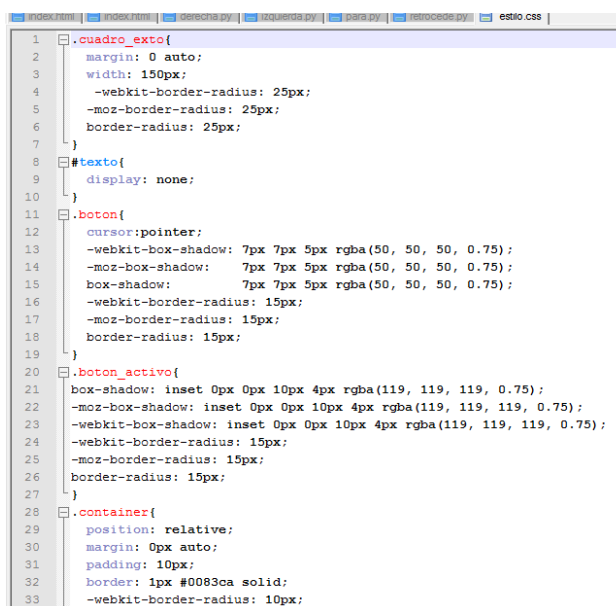
<script type="text/javascript">
var estado="";
$(function(){
$("#avanza").click(function(){
    $('#texto').show().text("Avanza");
    $("#texto").removeClass('btn-info btn-primary btn-success btn-
inverse').addClass( "btn-danger" );
    $(this).removeClass('boton').addClass( "boton_activo" );
    $("#derecha").removeClass('boton_activo').addClass( "boton" );
    $("#izquierda").removeClass('boton_activo').addClass( "boton" );
    $("#retrocede").removeClass('boton_activo').addClass( "boton" );
    $("#para").removeClass('boton_activo').addClass( "boton" );

    window.estado=1;
});
</script>

```

2. CARPETA CSS

Contiene un script que establece los estilos, efectos y otros adornos que contienen cada botón o casillero.



```

1  .cuadro_exto{
2  margin: 0 auto;
3  width: 150px;
4  -webkit-border-radius: 25px;
5  -moz-border-radius: 25px;
6  border-radius: 25px;
7  }
8  #texto{
9  display: none;
10 }
11 .boton{
12 cursor:pointer;
13 -webkit-box-shadow: 7px 7px 5px rgba(50, 50, 50, 0.75);
14 -moz-box-shadow: 7px 7px 5px rgba(50, 50, 50, 0.75);
15 box-shadow: 7px 7px 5px rgba(50, 50, 50, 0.75);
16 -webkit-border-radius: 15px;
17 -moz-border-radius: 15px;
18 border-radius: 15px;
19 }
20 .boton_activo{
21 box-shadow: inset 0px 0px 10px 4px rgba(119, 119, 119, 0.75);
22 -moz-box-shadow: inset 0px 0px 10px 4px rgba(119, 119, 119, 0.75);
23 -webkit-box-shadow: inset 0px 0px 10px 4px rgba(119, 119, 119, 0.75);
24 -webkit-border-radius: 15px;
25 -moz-border-radius: 15px;
26 border-radius: 15px;
27 }
28 .container{
29 position: relative;
30 margin: 0px auto;
31 padding: 10px;
32 border: 1px #0083ca solid;
33 -webkit-border-radius: 10px;

```

Figura 66: Contenido CSS

- `./mjpg_streamer -i './input_uvc.so -d /dev/video0 -r 320x240 -y -f 20' -o './output_http.so -w ./www -p 8080'`

Este archivo debe estar guardado en dos ubicaciones tanto en la carpeta de la página web y en la siguiente dirección `/usr/local/bin`, además deben tener permisos de funcionamiento en las dos localidades.

Otorgamiento de permisos.

- `cd var/www/Raspibot`
- `sudo chmod a+x play`
- `cd usr/local/bin`
- `sudo chmod a+x play`

Para todos los siguientes archivos se debe realizar el mismo procedimiento.

Stop: Este archivo como su propio nombre lo indica da un final al flujo de imágenes. Contiene lo siguiente.

- `#!/bin/bash`
- `cd mjpg-streamer`
- `sudo killall mjpg_streamer`

Power: Este archivo apagará el robot desde la página web, para no cortar la alimentación bruscamente. Contiene lo siguiente.

- `#!/bin/bash`
- `sudo shutdown -h now`

Reboot: Este archivo reiniciará el robot en caso de existir algún percance. Contiene lo siguiente.

- `#!/bin/bash`
- `sudo shutdown -r now`

Wstart: Es el encargado de iniciar la herramienta `webiopi` desde la página web. Contiene lo siguiente.

- `#!/bin/bash`
- `sudo /etc/init.d/webiopi start`

Wstop: Es el encargado de parar la herramienta de webiopi. Contiene lo siguiente.

- `#!/bin/bash`
- `sudo /etc/init.d/webiopi stop`

7. ARCHIVO PROCESA.PHP

Contiene todos los estados a ejecutarse dentro del Raspberry Pi. Al estar presionado el botón se mostrara un valor en la consola.



Figura 68: Acción de botones en la consola

Dentro del archivo PHP recibe el valor, lo valida y si concuerda con algún caso que contiene el archivo, con la función `exec` ejecuta el script de PYTHON ya sea este de avanzar, retroceder, o cualquier otro script.

```

1 <?php
2 $valor_estado=$_POST['valor_estado'];
3 switch ($valor_estado) {
4     case 1:
5         exec('sudo python /var/www/control/py/avanza.py');
6         break;
7     case 2:
8         exec('sudo python /var/www/control/py/izquierda.py');
9         break;
10    case 3:
11        exec('sudo python /var/www/control/py/derecha.py');
12        break;
13    case 4:
14        exec('sudo python /var/www/control/py/retrocede.py');
15        break;
16    case 5:
17        exec('sudo python /var/www/control/py/para.py');
18        break;
19    default:
20        exec('sudo python /var/www/control/py/para.py');
21        break;
22 }
23 ?>

```

Figura 69: Archivo php

3.7.2 CONTROL DE LOS MOTORES DE DESPLAZAMIENTO

El robot podrá desplazarse fácilmente con 4 motores los cuales serán controlados por medio del puerto GPIO del Raspberry Pi con la ayuda del chip L298.

Hay dos razones por las que se debe usar un chip L298 en este proyecto.

1. Es que la salida de la Raspberry Pi es en absoluto lo suficientemente fuerte como para manejar un motor directamente y si se tratara de hacer, eso puede dañar el Raspberry Pi.
2. No es posible realizar el control de la dirección del motor con el Raspberry pi. Esto sólo es posible mediante la inversión de la dirección de la corriente a través del motor, algo que el L298 está diseñado para hacer, con la ayuda de dos clavijas de control.

En el ejemplo explicado anteriormente para encender un led se pudo comprender que es un sencillo programa hecho en Python que simple mente

consta de poner en High o Low a un pin del GPIO, la programación para el control del motor de corriente continua será igual solo se tratara de 1 y 0.

Para el robot se utilizara los GPIO 17, 22, 23, 24 que en numeración de pines serian los siguientes 11, 15, 16, 18.

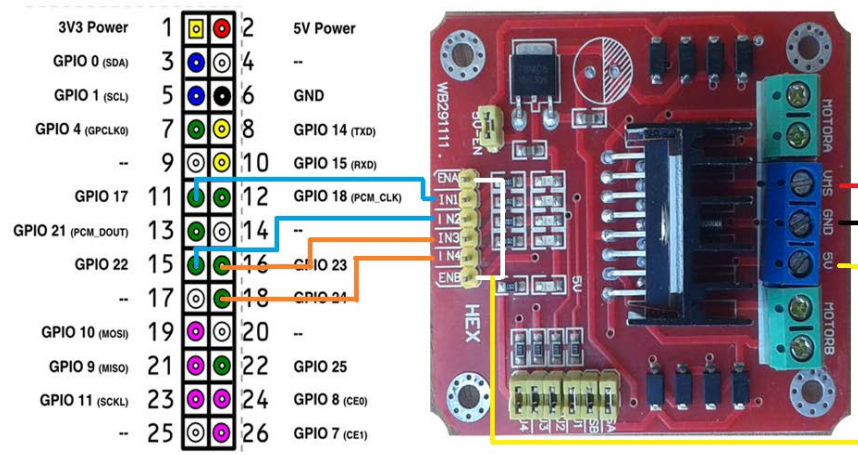


Figura 70: Conexiones del GPIO con el driver I298.

Dichos pines serán programados en el software PYTHON que será el que controle su respectivo accionamiento. Para poder controlar el movimiento del robot, los dos motores deberán realizar un movimiento sincronizado y los puertos deberán estar en los siguientes estados:

1. Avanza.

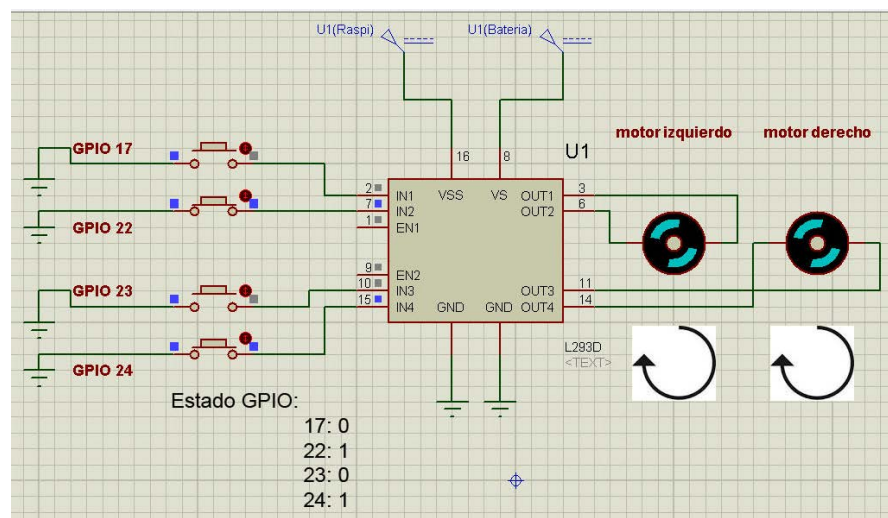


Figura 71: Estado avanza .

```

import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(17,GPIO.OUT)#Motor1
GPIO.setup(22,GPIO.OUT)

GPIO.setup(23,GPIO.OUT)#motor2
GPIO.setup(24,GPIO.OUT)

GPIO.output(17,False)
GPIO.output(22,True)

GPIO.output(23,False)
GPIO.output(24,True)

```

Figura 72: Programación de motores.

2. Retrocede

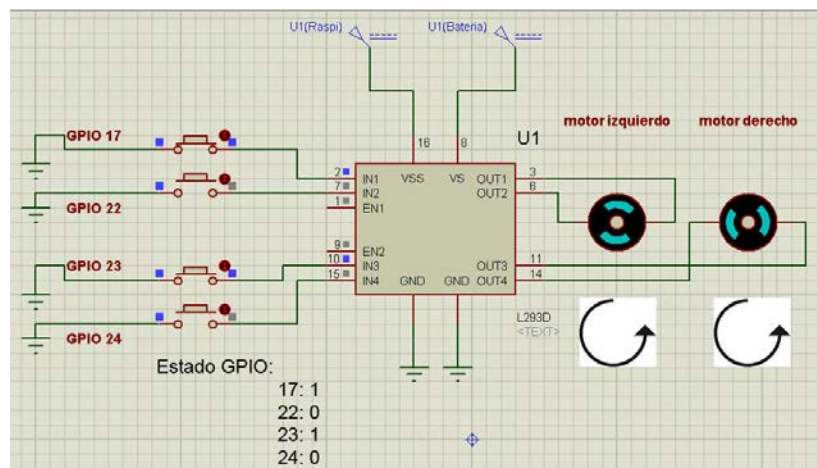


Figura 73: Circuito en estado de retroceder.

3. Giro a la derecha

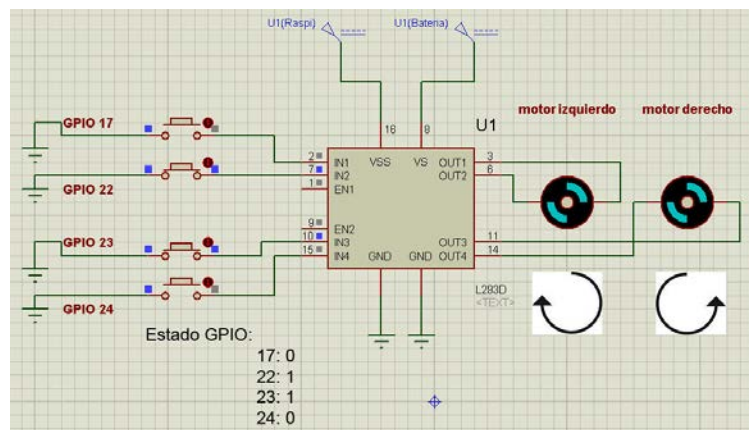


Figura 74: Circuito en estado de giro derecha.

4. Giro hacia la izquierda

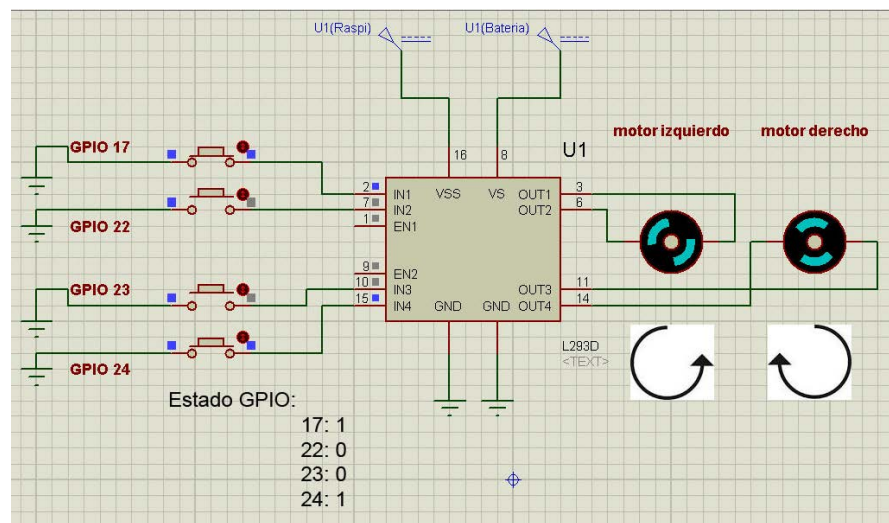


Figura 75: Circuito en estado de giro izquierda.

5. Stop

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(25,GPIO.OUT)

GPIO.setup(17,GPIO.OUT)#Motor1
GPIO.setup(22,GPIO.OUT)

GPIO.setup(23,GPIO.OUT)#motor2
GPIO.setup(24,GPIO.OUT)

GPIO.output(17,False)
GPIO.output(22,False)

GPIO.output(23,False)
GPIO.output(24,False)
```

Figura 76: Programación en estado de stop.

3.7.3 CONTROL DEL SERVOMOTOR DE LA CÁMARA

El servomotor ayudara a controlar el movimiento de la cámara en sentido vertical ya que estando estática no podrá cubrir un alto rango de visibilidad, Pero para realizar esto se tendrá que programar los movimientos en Python.

Como el servomotor realiza un movimiento de 180° se creara programaciones las cuales hagan mover al servomotor en los siguientes

ángulos 45°, 90°, 135° y 180° cubriendo así con la mayor parte de visibilidad a obstáculos en cualquier zona.

```

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(4,GPIO.OUT)

p = GPIO.PWM(4,50)
p.start(4.5)
try:
    while True:
        #p.ChangeDutyCycle(12.5) #180°
        #time.sleep(0.3)
        #p.ChangeDutyCycle(10) #135°
        #time.sleep(0.3)
        #p.ChangeDutyCycle(7.5) #90°
        #time.sleep(0.3)
        p.ChangeDutyCycle(5) #45°
        time.sleep(0.3)
        p.stop(4.0)
except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()

```

Figura 77: Programación de control de servomotor.

3.7.4 STREAMING DE VIDEO (Grinberg, 2013)

Para la transmisión de video se utiliza la herramienta MJPG-streamer, una aplicación que captura JPG de webcams compatibles con Linux-UVC, sistema de archivos u otros plugins de entrada y los distribuye como M-JPEG a través de HTTP para navegadores web, VLC y otros programas. La instalación sólo requiere unos sencillos pasos:

1. Instalar las herramientas de mjpg-streamer
 - sudo apt-get install libv4l-dev
 - sudo apt-get install libjpeg8-dev
 - sudo apt-get install imagemagick
 - sudo apt-get install subversion

- `sudo apt-get install v4l-utils`
2. Descargar la aplicación digitando en la terminal
 - `svn co https://svn.code.sf.net/p/mjpg-streamer/code/mjpg-streamer/mjpg-streamer`
 3. Se descargara en la siguiente dirección `/home/pi` que vendría a ser documentos de Raspberry Pi, y acceder a él con el comando.
 - `cd mjpg-streamer`
 4. Instalar mjpg-streamer
 - `sudo make mjpg_streamer input_file.so input_uvc.so output_http.so`
 5. Para ejecutar el Streamer de video desde cualquier parte de la raspberry es necesario crear un script que contenga los datos respectivos para que pueda ejecutarlo y pararlo, para esto se realiza un archivo en la siguiente ruta.
 - `cd /usr/local/bin`
 6. Elaborar los escript que se los denominara play y stop.
 - `sudo nano play`
 7. Y colocar las siguientes líneas dentro
 - `#!/bin/bash`
 - `cd mjpg-streamer`
 - `export LD_LIBRARY_PATH=.`
 - `./mjpg_streamer -i './input_uvc.so -d /dev/video0 -r 320x240 -y -f 20' -o './output_http.so -w ./www -p 8080'`
 8. Guardar y para que funcione sin ningún problema otorgarle permisos de súper usuario con el siguiente comando.
 - `sudo chmod a+x play`
 9. Siguiete elaborar el script de stop el que detendrá el flujo de video, se lo realizara de la misma manera que el anterior.

- sudo nano stop
10. Y colocar dentro
- #!/bin/bash
 - cd streamer
 - sudo killall mjpg_streamer
11. Guardar y darle permiso de súper usuario.
- sudo chmod a+x stop
12. Ejecutar el script.
- play

```

pi@raspberrypi /usr/src/mjpg-streamer/mjpg-streamer $ export LD_LIBRARY_PATH=.
pi@raspberrypi /usr/src/mjpg-streamer/mjpg-streamer $ ./mjpg_streamer -i './input_uvc.so -d /dev/video0 -r 640x480 -f 15' -o './output_http.so -w ./www -p 8080'
MJPEG Streamer Version: svn rev: Unversioned directory
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 640 x 480
i: Frames Per Second.: 15
i: Format.....: MJPEG
Adding control for Pan (relative)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Tilt (relative)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Pan Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Tilt Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Pan/tilt Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Focus (absolute)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
mapping control for Pan (relative)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Tilt (relative)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Pan Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Tilt Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Pan/tilt Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Focus (absolute)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for LED1 Mode
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for LED1 Frequency
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Disable video processing
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Raw bits per pixel
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
o: www-folder-path...: ./www/
o: HTTP TCP port.....: 8080
o: username:password.: disabled
o: commands.....: enabled

```

Figura 78: Corriendo mjpg-streamer.

13. Si todo sale como esta en la figura 78 la transmisión está siendo procesada, para comprobarlo ir a una computadora que este en la misma red local, abrir un navegador y colocar la siguiente dirección. <http://ip raspberry pi:8080>, deberá salir lo siguiente.

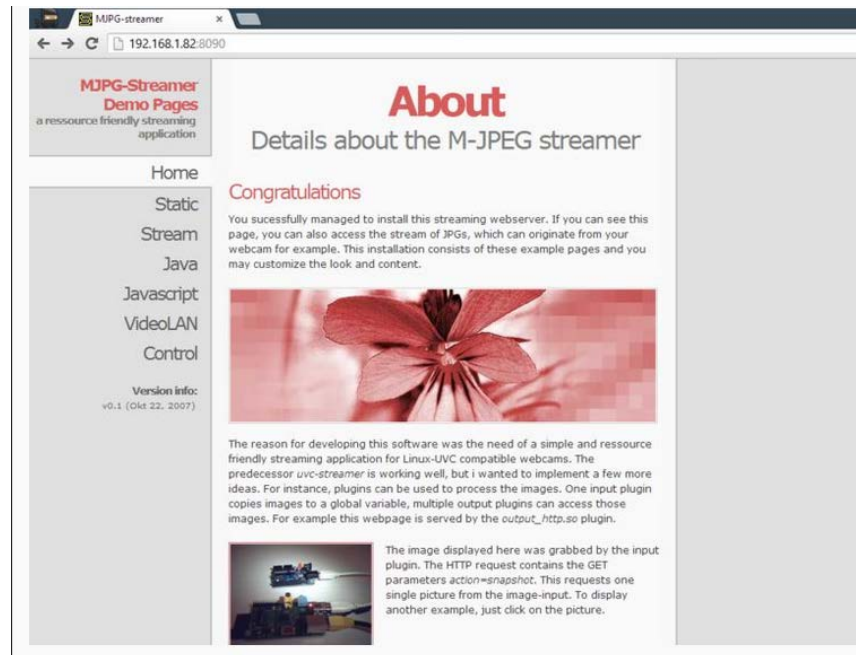


Figura 79: Página principal de mjpg-streamer.

14. Para observar el flujo dar clic en stream.

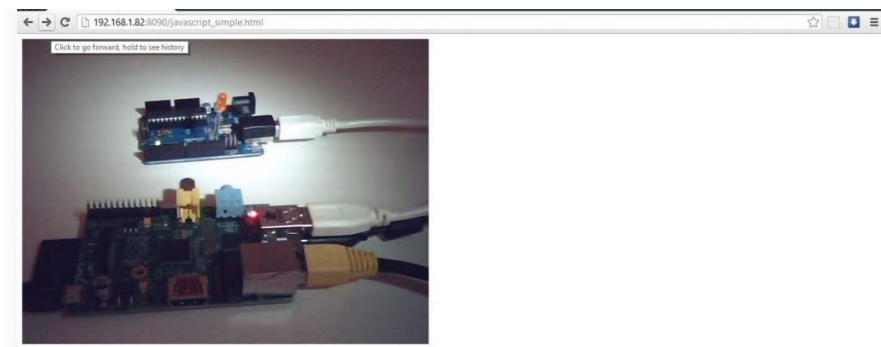


Figura 80: Transmisión en vivo.

15. Si se quiere finalizar abrir otra terminal para ejecutar el script stop.

3.8 HARDWARE DEL ROBOT MÓVIL

Para la creación física del robot móvil se utilizara los siguientes componentes:

- Chasis 4x4 Arduino
- Raspberry Pi Modelo B
- Cámara Web (Speed mind) resolución interpolada por software de hasta 5.0 Megapíxeles.
- SD card de 8gb
- Modem USB de internet
- Adaptador Wifi
- Adaptador hdmi a vga
- 1 Banco de poder de 5v/5600mA
- 2 Bancos de poder de 4v/2600mA
- Driver L298
- Circuito On/Off de motores

El tiempo de funcionamiento del robot será primordial, por lo cual se ha optado por la adquisición de una batería de 5V/5600mA (Solo para uso del raspberry Pi) y dos baterías de 4V (circuitos adicionales) que serán colocadas en serie dando un total de 8v/2600mA. Esto brindará un buen tiempo de funcionamiento dependiendo a la corriente total que se consuma. El Raspberry Pi con todos los componentes instalados como son el adaptador wifi y cámara web, consumirá un máximo de 1100mA; los 4 motores de corriente continua, el driver de los motores y el servomotor de la cámara los cuales conforman los circuitos adicionales y estos consumen un total aproximado de 500 mA.

Realizando una operación sencilla se sabrá con mayor precisión el tiempo de durabilidad de la batería.

Calculo de durabilidad de la batería del Raspberry pi:

$$Tiempo = \frac{\text{Capacidad de la batería (mAh)}}{\text{Velocidad de descarga (mA)}}$$

$$Tiempo = \frac{5600\text{mAh}}{1100\text{mA}}$$

$$Tiempo = 5,09 \text{ h}$$

Calculo de durabilidad de las baterías de los circuitos adicionales:

$$Tiempo = \frac{\text{Capacidad de la batería (mAh)}}{\text{Velocidad de descarga (mA)}}$$

$$Tiempo = \frac{2600\text{mAh}}{500\text{mA}}$$

$$Tiempo = 5,2 \text{ h}$$

Ya listo todos los componentes será armada la estructura del robot, la cual tiene como base un chasis 4x4 Arduino.

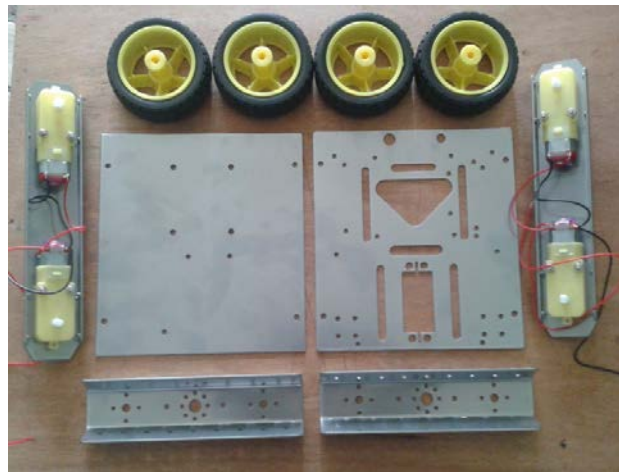


Figura 81: Chasis 4x4 desarmada

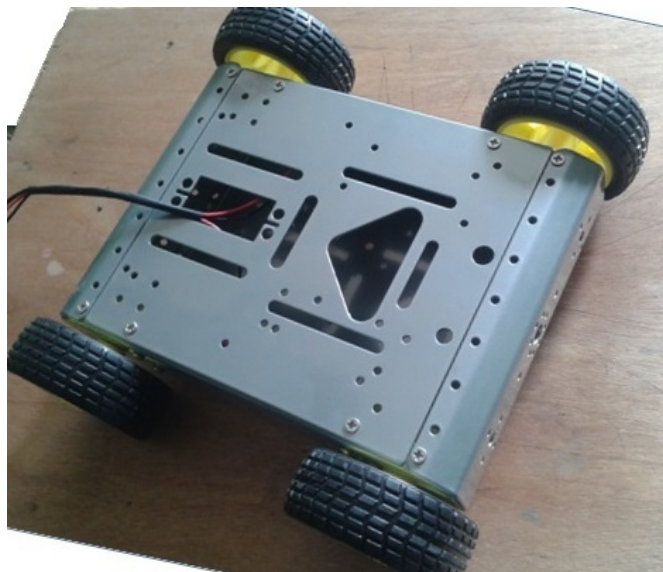


Figura 82: Chasis en pie.

La estructura es diseñada en fibra de vidrio para que el robot no sea tan pesado.

Para trabajar fácilmente con la fibra de vidrio se realizo una plancha grande y luego se hizo las respectivas divisiones para el armado.

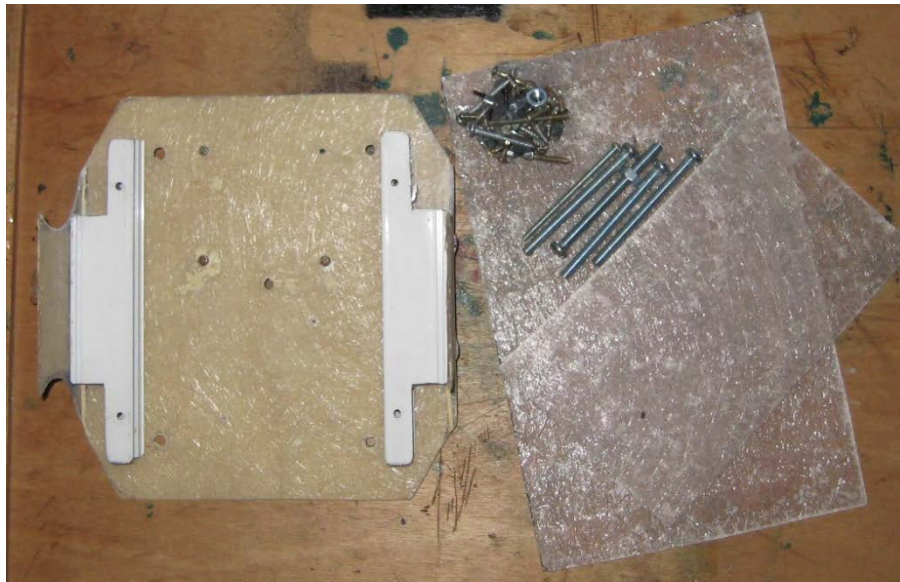


Figura 83: Cortes de fibra de vidrio.

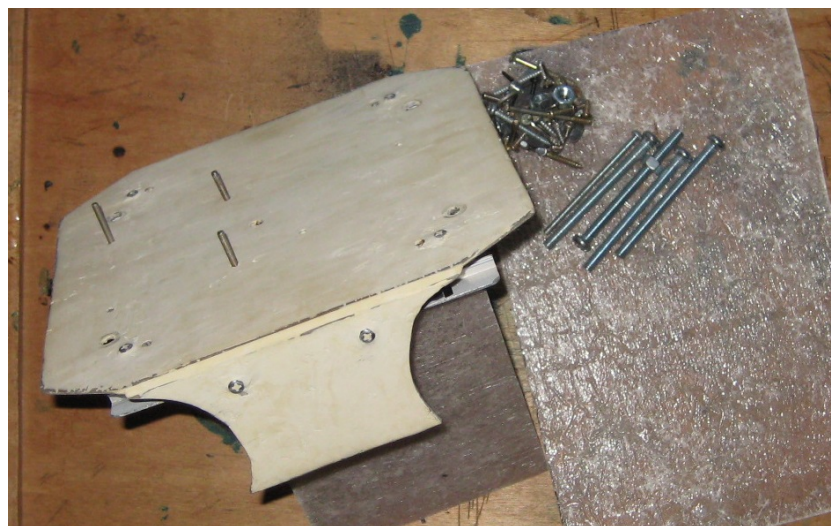


Figura 84: Diseño pre acabado.

Ya tomada la forma es pintada de color negro e instalado los complementos necesarios para su funcionalidad.

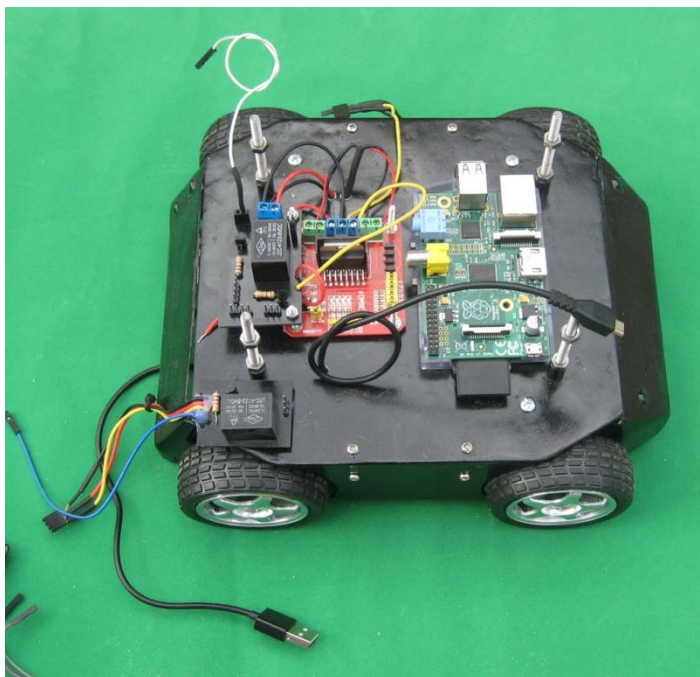


Figura 85: Estructura con circuitos electrónicos

Ya con todos los complementos instalados el resultado será el siguiente. (Figura 86).

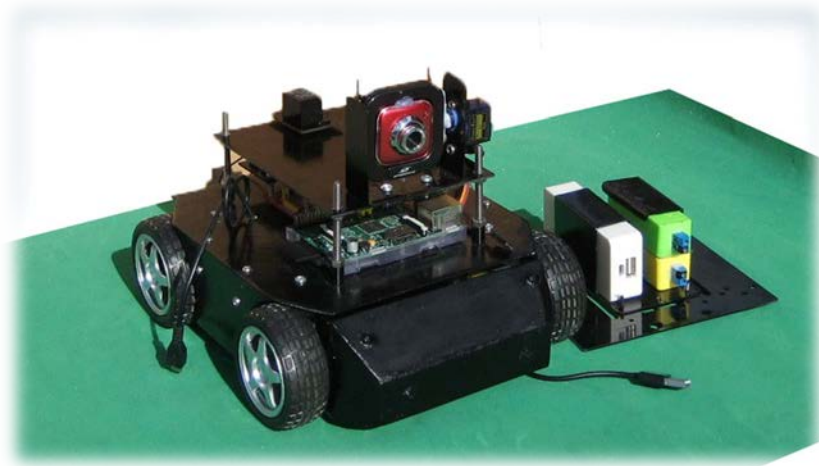


Figura 86: Robot móvil armado

3.8.1 CONEXIONES ENTRE LOS CIRCUITOS

En caso de existir una necesidad de desarmar el robot móvil, los siguientes diagramas servirán para volver a reestructurarlo.

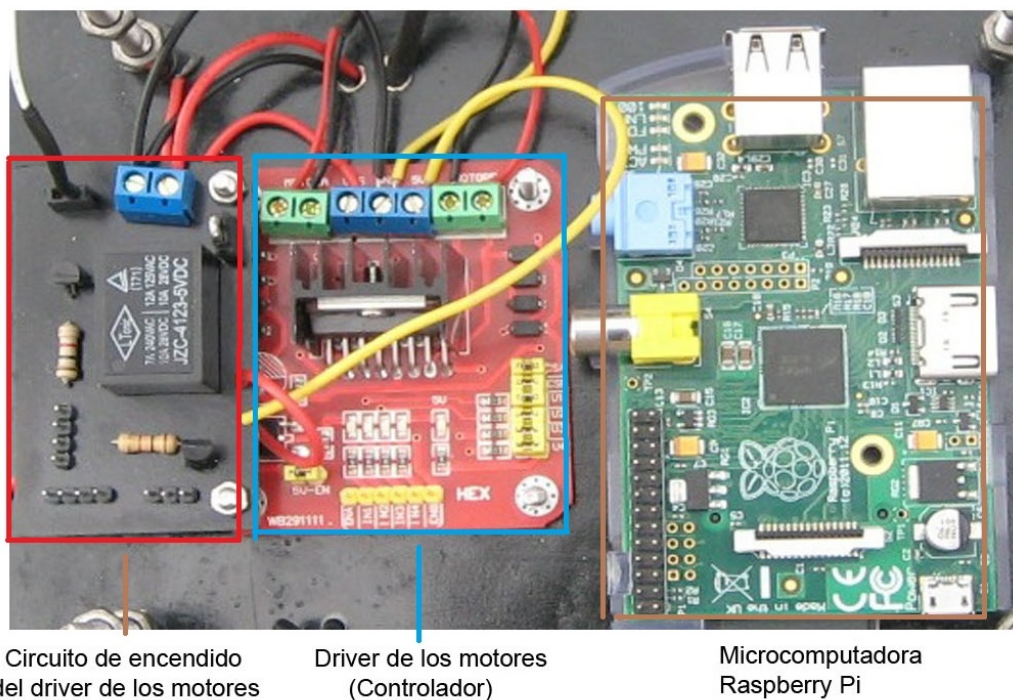


Figura 87: Localidad de circuitos

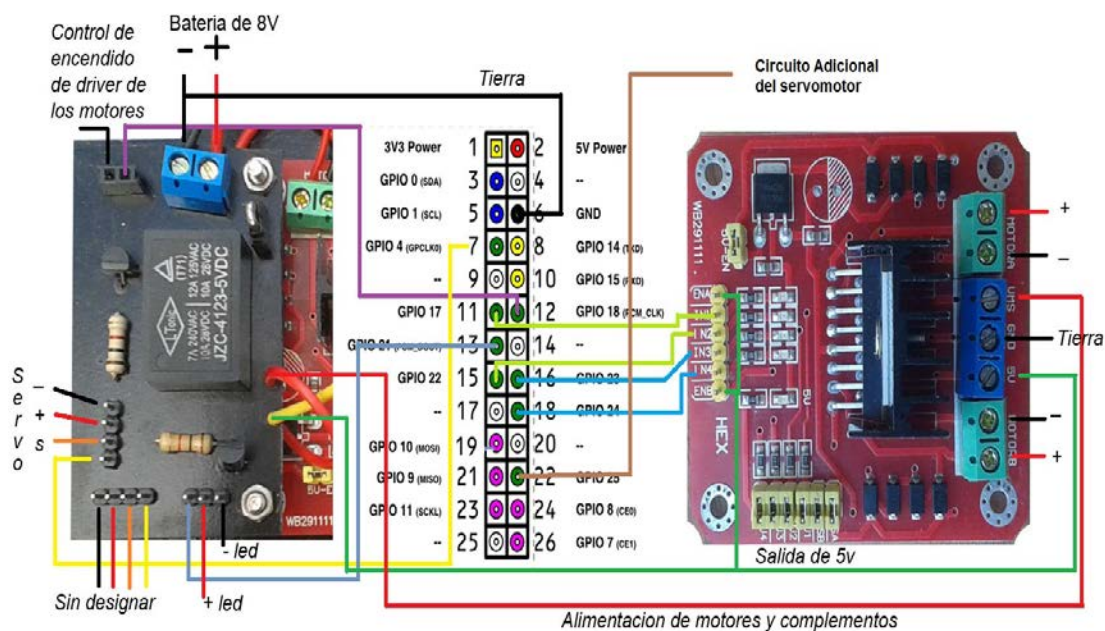


Figura 88: Diagrama de conexiones entre circuitos

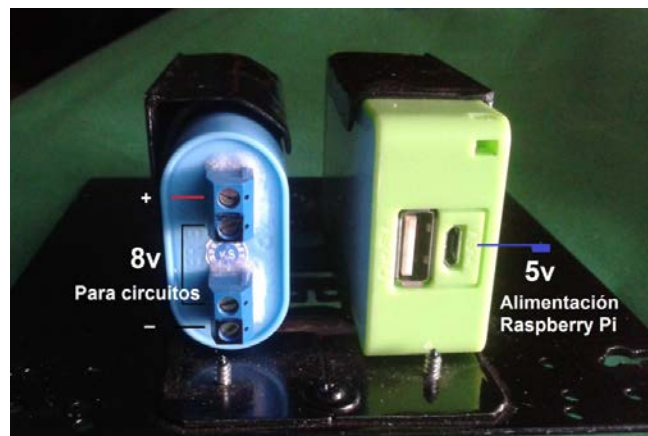


Figura 89: Estructura de baterías

3.9 CONFIGURACIÓN DEL ROUTER

Este punto es importante, para tener acceso a la Raspberry Pi, desde cualquier equipo que esté conectado a internet.

Generalmente un proveedor de servicios de Internet, deja inactivos o bloqueados los puertos del router, y por ejemplo si no está habilitado el puerto 80, no se podrá acceder a las aplicaciones montadas en el servidor apache.

Entonces lo que se debe hacer, es abrir esos puertos accediendo al router, y apuntarlos a la IP local que tenga asignada la Raspberry Pi. Y para eso se hará lo siguiente.

1. Acceder al modem Huawei E355 con nombre de usuario: admin y contraseña: admin y ubicarse en configuraciones.

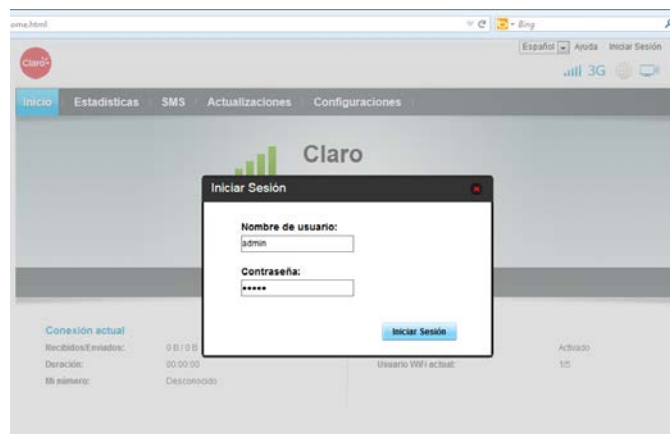


Figura 90 : Acceso modem usb (router).

- Ya en el menú del router dirigirse a seguridad e ingresar en servidor virtual, dar clic en agregar y llenar los datos como se observa en la figura 91. Ir a opciones pulsar en aceptar y dar clic en aplicar.

Inicio | Estadísticas | SMS | Actualizaciones | **Configuraciones**

Configuración rápida
Dial-up
WLAN
Seguridad
Gestión Del Código PIN
Firewall Switch
Filtro IP LAN
-> Servidor Virtual
Aplicaciones Especiales
Configuraciones DMZ
Configuración ALG de SIP
Configuraciones UPnP
Configuraciones de NAT
Sistema

Servidor Virtual

Configure un servidor virtual para permitir que usuarios externos accedan a WWW, FTP u otros servicios proporcionados por la LAN.

- **Dirección IP:** Designar la ubicación de una computadora en la LAN para el suministro de servicios.
- **Puerto LAN/WAN:** El puerto de la computadora que provee servicios es un puerto único y el rango de valores del puerto LAN/WAN es 1-65535.
- **Protocolo:** Protocolos aplicados por los servicios.
- **Nota:** Las configuraciones se implementarán al hacer clic en el botón "Aplicar".

Lista de Servidores Virtuales

Nombre	Puerto WAN	Dirección IP LAN	Puerto LAN	Protocolo	Estado	Opciones
Raspberry	80	192.168.68.102	80	TCP/UDP	Activado	Aceptar Cancelar

[Agregar](#)

[Aplicar](#)

Figura 91 : Apertura de puertos de internet.

Con esos pasos quedaría abierto el puerto.

3.10 SIMULACIONES

En esta parte se realiza una prueba del funcionamiento que realiza el robot, principalmente la transmisión del flujo de imágenes que se receipta en la página web.



Figura 92: 1° Interiores de Laboratorio

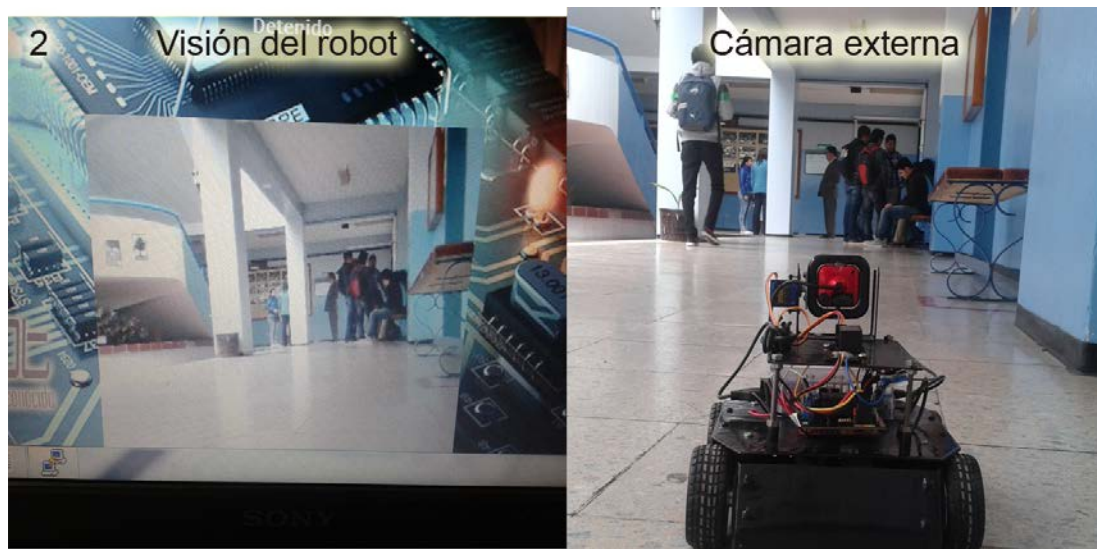


Figura 93: 2° Visualización del pasillo

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Se implemento el robot explorador tipo vehículo que consta de la microcomputadora Raspberry Pi, adaptador wifi, web cam, un driver L298, entre otros, todo esto es manejado desde el internet por medio de una página web instalada en el robot explorador.
- Se obtuvo la información de la microcomputadora Raspberry Pi y se determino los elementos que intervienen en la implementación del robot explorador tipo vehículo.
- Se construyo la estructura del robot en base al chasis 4x4 Arduino que consta de 4 motores de corriente continua que serán accionados con el driver L298.
- Se configuro la microcomputadora sobre el sistema operativo Raspbian y se programo la página web y las acciones del robot en los software de HTML y Python.

4.2 RECOMENDACIONES

- Sería conveniente que el robot explorador sea utilizado como una herramienta de prácticas del laboratorio de instrumentación virtual.
- El presente documento que será designado para la biblioteca de la Unidad de Gestión de Tecnologías sea utilizado como una herramienta para introducir a los estudiantes en el nuevo sistema de la electrónica, que está haciéndose popular a nivel mundial.
- Si se realiza alguna modificación en la estructura tomar en cuenta cuales serian los efectos que pueden causar en el robot, porque puede tener algunas limitaciones al excederse de peso.
- Si en la configuración inicial de Raspbian se realiza el cambio de la contraseña y nombre de usuario es aconsejable guardarlos en un blog de notas o anotarlos en algún cuaderno de apuntes, en algunas aplicaciones serán útiles.

- Si se desea mejorar la velocidad de procesamiento del Raspberry Pi y se procede a realizar un overclock, Instalar un disipador de mayor tamaño o un ventilador, porque el procesador se calentara mucho y eso puede causar daños al Raspberry Pi hasta dejarlo obsoleto.
- Si se adhieren más complementos a la microcomputadora para mejoramiento del robot utilizar una batería que tenga como salida 1.5A como mínimo si no fuese así puede llegar a no funcionar correctamente algún complemento.
- En caso de no poder visualizar el flujo de imágenes en la página web del robot, dirigirse a `var/www/Raspibot` e ingresar al archivo `index.html` y verificar que las direcciones ip que se encuentran dentro sean las correctas.

GLOSARIO

ARM: Es una arquitectura RISC (Ordenador de Conjunto Reducido de Instrucciones) para aplicaciones de baja potencia.

Compilar: Traducir con un compilador un programa en lenguaje de alto nivel a lenguaje de la máquina.

Comandos: Es una instrucción u orden que el usuario proporciona a un sistema informático, desde la línea de comandos o desde una llamada de programación. Puede ser interno o externo (contenido en un archivo ejecutable).

Framework: Es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. El objetivo de los Frameworks es proporcionar una estructura común, de modo que los desarrolladores no tienen que hacer el código de cero cada vez y puede volver a utilizar la gran mayoría.

GPIO: (Entrada y Salida de Propósito General) son pines situados en el Raspberry Pi, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar mediante una programación por el usuario.

Hardware: El término hardware se refiere a todas las partes tangibles de un sistema informático como son todos los componentes que conforman una maquina informática.

HDMI: High-Definition Multimedia Interface o interfaz multimedia de alta definición, es una norma de audio y vídeo digital cifrado sin compresión apoyada por la industria para que sea el sustituto del euroconector.

HTTP: Hypertext Transfer Protocol (protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web (www).

Index: El archivo *index.html* es la página principal del dominio al cual se accede, es lo que se ve en primer lugar a la hora de acceder a una página web.

Linux: Es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquier parte.

Multiplataforma: En informática, multi-plataforma, es un atributo conferido a programas informáticos o métodos y conceptos de cómputo que son implementados e interoperan en múltiples plataformas informáticas.

Overclock: Aumentar forzosamente la velocidad del procesador, memoria ram y hasta los chips de videos de tarjetas de video. Esto se lo hacen para programas que así lo necesiten.

PWM: La modulación por ancho de pulsos de una señal o fuente de energía, es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

Raspbian: Sistema operativo de distribución Linux, funcional solo para dispositivos Raspberry pi.

Raspberry Pi: (Raspberry, Pi, Raspi), Microcomputadora denominada así por su Fundación, Raspberry Pi es una organización benéfica registrada en la Comisión de Caridad para Inglaterra y Gales.

RCA: Conector RCA, frecuentemente llamado conector Cinch, es un tipo negro de conector eléctrico común en el mercado audiovisual.

Root: Ser root equivale a obtener los máximos privilegios posibles en un dispositivo. Al tener el control se puede modificar diferentes cosas que antes no se podía, como usar aplicaciones exclusivas, crear Backups o copias de seguridad ,cambiar la frecuencia de la CPU para que vaya más rápido, etc.

Software: Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.

Stream: Es una corriente o flujo.

Sudo: Denominativo que se le da al súper usuario en algunas distribuciones Linux en línea de comando.

UNIX: Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969, el término UNIX se utiliza en su acepción de familia. Se aplica también a sistemas multiusuario basados en POSIX (tales como GNU/Linux, Mac OS X).

VGA: Adaptador Gráfico de Video que se utiliza para denominar a una pantalla de computadora analógica estándar, a la resolución 640 × 480 píxeles.

NETGRAFÍA

- Apache. (1999). *Apache*. Obtenido de <http://httpd.apache.org/> (En línea)
- Bejarano, M. (10 de Agosto de 2013). *Frambuesa Pi Colombia*. Obtenido de <http://www.frambuesapi.co/2013/08/10/tutorial-3-instalacion-y-configuracion-de-inicial-del-raspberry-pi-raspi-config/> (En línea)
- Fundacion, R. P. (2011). *Raspberry Pi*. Obtenido de <http://www.raspberrypi.org/resources/make/> (En línea)
- González, E. (2011). *Aprender a Programar*. Obtenido de http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=69&Itemid=192 (En línea)
- Google. (s.f.). *WebIOPi*. Obtenido de <https://code.google.com/p/webiopi/> (En línea)
- Grinberg, M. (25 de Mayo de 2013). <http://blog.miguelgrinberg.com>. Obtenido de Stream Video from the Raspberry Pi: <http://blog.miguelgrinberg.com/post/stream-video-from-the-raspberry-pi-camera-to-web-browsers-even-on-ios-and-android> (En línea)
- MJPEG-streamer. (28 de Septiembre de 2007). *MJPEG-streamer*. Obtenido de <http://sourceforge.net/projects/mjpg-streamer/> (En línea)
- Murillo, P. (17 de Diciembre de 2011). *arduteka*. Obtenido de <http://www.arduteka.com/2011/12/componentes-el-servomotor/> (En línea)
- Python. (2001). *Python*. Obtenido de <https://www.python.org/> (En línea)
- Ramirez, F. (13 de Marzo de 2013). *Tu Electronica.es*. Obtenido de <http://www.tuelectronica.es/tutoriales/raspberry-pi/usar-raspberry-pi-sin-monitor.html> (En línea)
- Rivera, J. (23 de Junio de 2013). *Blog Jeferson Rivera*. Obtenido de <http://riverajefer.blogspot.com/2013/06/control-remoto-de-un-led-con-raspberry.html> (En línea)

Rivera, J. (24 de Agosto de 2013). *Blog Jefferson Rivera*. Obtenido de <http://riverajefer.blogspot.com/2013/08/control-remoto-para-un-carrito-traves.html> (En línea)

CURRICULUM VITAE



DATOS PERSONALES:

Nombre:	Díaz Córdova Pedro Gabriel
Nacionalidad:	Ecuatoriana
Fecha de nacimiento:	15-Mayo-1991
Lugar de nacimiento:	Imbabura, Atuntaqui.
Cédula de Identidad:	100339391-3
Teléfono:	0992475177
Correo electrónico:	gavos11@hotmail.com
Dirección Domiciliaria:	Pujilí, Pichincha y Rafael Morales

ESTUDIOS REALIZADOS:

Primaria:

- José Joaquín Olmedo

Secundaria:

- Instituto Tecnológico Otavalo
- Título: Electrónica de consumo

Superior:

- Unidad de gestión de tecnologías-ESPE- Egresado
- Título: Tecnólogo en Electrónica Mención Instrumentación & Aviónica

CURSOS REALIZADOS:

- Mantenimiento de computadoras en el "CENTRO DE CAPACITACIÓN IMBABURA" localidad Imbabura-Otavalo.
- Suficiencia en el idioma inglés

REFERENCIAS PERSONALES:

- Lic. Hugo Quilumbango 0994629199 "MAHUDIEL instalaciones eléctricas domiciliarias".
- Sr. Jesús Morales 0990145877 "Administrador en fábrica textil SANTE en Imbabura".

EXPERIENCIA LABORAL:

- Instalaciones Eléctricas domiciliarias.
- Actualización de Software y Mantenimiento de Hardware de computadoras.
- Prácticas pre-profesionales en la Empresa Eléctrica EmelNorte Atuntaqui y EmelNorte Otavalo (Imbabura).
- Prácticas pre-profesionales “TAME QUITO” área mantenimiento (Ayudante de técnico)

