ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

"ACTIVACIÓN DE ALARMAS REMOTAS MEDIANTE WIFI ENTRE MINICOMPUTADORAS RASPBERRY PI EN APLICACIONES DE SEGURIDAD"

TESINA DE SEMINARIO

Previa la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Amarilis Silva Rodríguez

Julio Bustamante Torres

GUAYAQUIL – ECUADOR

AÑO 2013

AGRADECIMIENTO

A Dios por guiarnos y acompañado a lo largo de nuestra carrera por ser nuestra fortaleza en los momentos de debilidad porque siempre está en nuestro camino y nos muestra esa luz de esperanza y nos dota siempre de inteligencia.

DEDICATORIA

A Dios por guiarnos por el buen camino y por darnos la fuerza para poder seguir adelante sin perder y permitirnos llegar a este momento tan grande nuestras vidas.

A nuestros padres por ser el pilar de nuestras vidas y formarnos con los mejores valores, sentimientos y hábitos, lo cual nos ha ayudado a salir adelante en los momentos difíciles.

TRIBUNAL DE SUSTENTACIÓN

_____

Ing. Carlos Valdivieso A.

PROFESOR DEL SEMINARIO DE GRADUACIÓN

_____

Ing. Hugo Villavicencio

PROFESOR DELEGADO DE LA UNIDAD ACADEMICA

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral".

(Reglamento de Graduación de la ESPOL)

Amarilis Silva Rodríguez	**Julio Bustamante Torres**

RESUMEN

En el presente trabajo se ha realizado un estudio de la minicomputadora Raspberry Pi para el uso de sus puertos de comunicación y cumplir los objetivos trazados como el caso de transmitir señales entre una y otra de estas minicomputadoras.

Las generalidades de la parte principal de este proyecto, es decir el conocimiento del uso de la Raspberry Pi se ha recopilado en los capítulos 1 y 2. Donde en el capítulo 1 se presentan los antecedentes, objetivos y las limitaciones que se tienen al realizar el proyecto.

Para el capítulo 2 se presentan las partes principales que caracterizan a la Raspberry Pi así como las distintas herramientas que ayudan a la realización del proyecto; estas herramientas son las partes de la minicomputadora, la parte de comunicación inalámbrica como los es el WI-FI y los programas para configurar la parte de comunicación con los puertos de este dispositivo.

La parte final la conforman ejercicios, proyecto que se encuentran en el capítulo 3 y las pruebas y simulaciones de los mismos en el capítulo 4. Por último se dan a conocer las conclusiones y recomendaciones del proyecto en general.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Este proyecto abarca el tema de seguridad con el desarrollo de una alarma remota utilizando tecnología WI-FI como medio de interconexión de otro dispositivo llamado Raspberry Pi siendo este la más importante para la implementación de este sistema de seguridad.

Se ha dividido en dos partes este proyecto los cuales son hardware y software. El hardware está constituido por dos Raspberry Pi, un punto de acceso y un circuito electrónico. El Raspberry Pi posee un puerto de comunicación llamado GPIO que por medio de este se realiza la comunicación con el circuito electrónico, y por medio del punto de acceso y con la configuración de la Raspberry Pi se desarrolla la comunicación inalámbrica entre estas minicomputadoras.

En lo que consiste al software, como la Raspberry Pi sea desarrollado para Linux los programas usados están basados en este sistema operativo de código abierto. Se ha instalado programas como PHP5, APACHE2 y aplicaciones desarrolladas como WiringPi.

El funcionamiento del sistema consiste en activar una alarma mediante un botón de pánico ubicado en una de las Raspberry Pi que envía una señal mediante una conexión inalámbrica a la otra Raspberry Pi donde se enciende un led con luz intermitente. Además una vez que se activa la alarma se puede proceder a la desactivación de la misma desde la misma ubicación de donde se ha encendido o desde donde se ve la luz de alarma.

CAPÍTULO 1

DESCRIPCIÓN GENERAL DEL PROYECTO

1.1 ANTECEDENTES

El uso de las computadoras se ha desarrollado a gran escala en nuestros tiempos dado que gran cantidad de trabajo se resume en solo una computadora, que actualmente se ha hecho un instrumento básico en todas las áreas de desarrollo de la humanidad (laboral, entretenimiento, educación, etc.). La ENIAC (Electronic Numerical Integrator And Calculator) que fue la primera computadora electrónica pesó 30 toneladas aproximadamente, y sus dimensiones fueron de 2.4m x 0.9m x 30m, ocupaba todo un sótano de la Universidad de Pensilvania, consumía 200 KW de energía eléctrica y requería todo un sistema de aire acondicionado. Así mismo fueron creadas las minicomputadoras estas son más pequeñas

que las macro computadoras pero también de un menor costo, diseñadas gracias al uso de los circuitos integrados.**[1]**

En el mercado existe un gran número de herramientas entre ellas el Arduino que utiliza un microcontrolador de 8 bits serie AT mega.

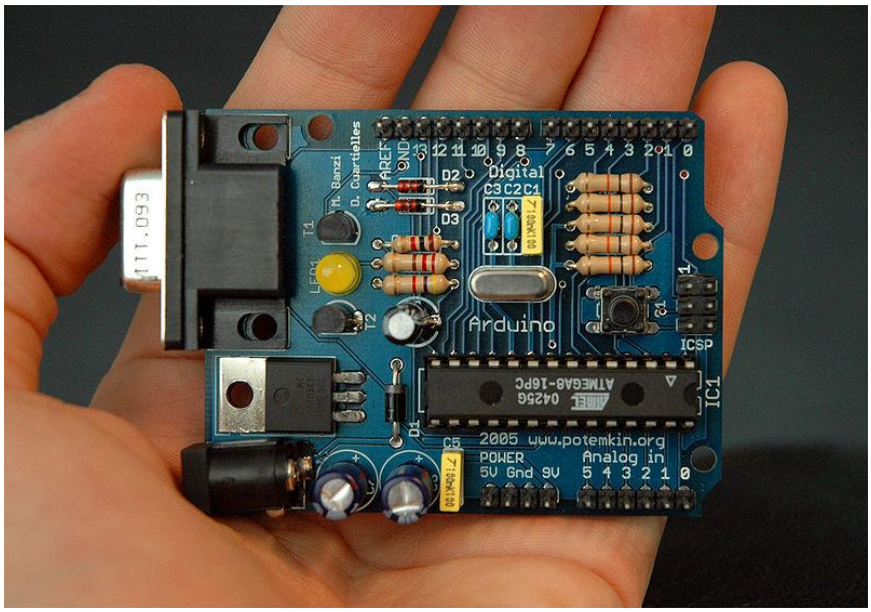


***Figura 1.1Arduino*[2]**

***Figura 1.2 Placa Raspberry Pi*[3]**

En la figura 1.2presentamos a la minicomputadora Raspberry Pi creada en Reino Unido por la fundación Raspberry Pique es básicamente un ordenador creado con el objetivo de estimular la enseñanza de ciencias computacionales en las escuelas. Su diseño incluye un *Systemon a chip Broadcom* que contiene un procesador central (CPU) a 700 MHz, un procesador gráfico (GPU), 256 MB de memoria RAM que le permite ejecutar Linux, utiliza una tarjera SD para el almacenamiento permanente, 2 conectores USB, video RCA, conexión Jack de 3,5 pulgadas para audio y los

más interesante HDMI, la administración de las Raspberry Pi no es tan complicada gracias a la información que se encuentra en internet. **[4]**

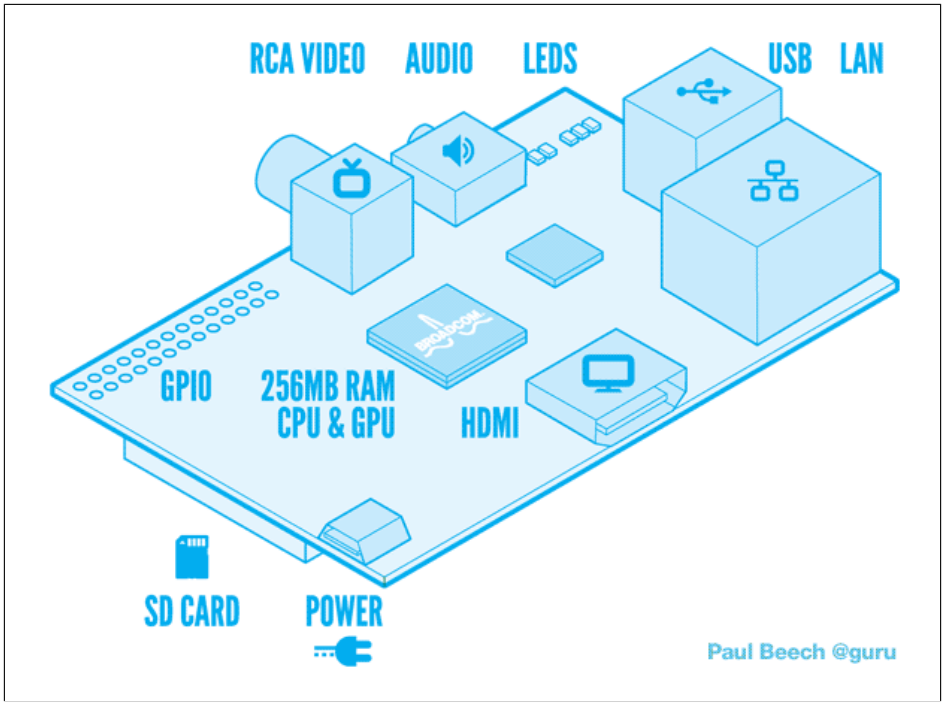


***Figura 1.3 Partes de la Raspberry Pi*[5]**

1.2 OBJETIVOS

OBJETIVO GENERAL

- ✓ El objetivo principal para la implementación de nuestro proyecto de tesina es el usar las minicomputadoras Raspberry Pi conectadas mediante Wi-Fi para activación de alarmas remotas en aplicación de seguridad.

OBJETIVOS ESPECÍFICOS

- Lograr realizar comunicación entre minicomputadoras Raspberry Pi.
- Manejar correctamente el lenguaje de programación C y PHP.
- Implementar una alarma mediante un Botón de pánico para notificar peligro.
- Dominar el manejo de la minicomputadora Raspberry Pi para propósitos generales.
- Implementar un código en lenguaje C.
- Implementar un Código en lenguaje PHP para poder utilizar un web server.

1.3 IDENTIFICACIÓN DEL PROBLEMA

En el medio en el que nos encontramos hay una ausencia de tecnología en comparación con otros países, esto nos ha motivado a que estudiantes y profesionales empecemos a utilizar instrumentos electrónicos que puedan ayudar en el desarrollo del país. La Raspberry Pi como tecnología nueva no se la está explotando en su uso, se requiere el conocimiento necesario para poder desarrollar el uso en nuestra comunidad brindando así mejoras en el estilo de vida para aquellas personas que tienen acceso a estas tecnologías ya sea para las empresas, compañías, universidades, etc.

Un problema a solucionar consiste en la escases de este dispositivo en nuestro país, el ingreso a la industria de la tecnología, las personas no tiene el conocimiento necesario ni la manipulación en su manejo, lo que estamos haciendo es sugerir una alternativa nueva y conveniente ya que es portable por lo tanto tiene facilidad de movilidad y por su peso y tamaño se la puede utilizar en vez de un ordenador y su precio es relativamente bajo.

La implementación de un proyecto de tesina mediante el uso de Comunicación wifi entre 2 Raspberry Pi demanda algunas características fundamentales para permitir el correcto funcionamiento entre estos 2

dispositivos, entre los cuales se encuentran la programación por lo que se ha tenido que realizar estudios y preparación académica para lograr entender de forma correcta el funcionamiento y en fin sacarle el mayor provecho y que así ayuden a tener empresas mejor dotadas tecnológicamente para lograr competir con países vecinos.

1.4 LIMITACIONES

Unas de las limitaciones en este trabajo es recordar y tener el conocimiento necesario de los lenguajes de programación que vamos utilizar en la Raspberry Pi (C, PHP). **[6]**

Es importante tener todas las herramientas disponibles instaladas también en una PC y tener que hacer uso de ellas para poder realizar diferentes pruebas y luego poder plasmarlas en la Raspberry Pi como lo son VNC, PUTTY, WIRESHARK.

Las limitaciones debido a la corriente que suministra el cargador y la corriente que suministra los puertos usb de un ordenador y si se necesita la conexión de otros dispositivos con comunicación usb se necesitaría

conseguir un *hub* con alimentación propia para compartir la red de puertos usb.

En cuanto a las limitaciones de la cantidad de equipos disponibles si por algún motivo se descompone una Raspberry Pi volver a importarlos retrasaría nuestro proyecto así como también materiales necesarios que no se encuentren con facilidad dentro de la ciudad o país.

La minicomputadora Raspberry Pi es relativamente nueva en el mercado esto hace que su investigación del uso y su programación sea más exhaustivo.

CAPÍTULO 2

FUNDAMENTO TEÓRICO

2.1 RESUMEN

Para una mejor comprensión del proyecto, en el presente capítulo se da a conocer la teoría descriptiva de las partes fundamentales que serán puestas en funcionamiento, de esta manera brindamos la información esencial sobre cómo se logra una comunicación WiFi entre minicomputadoras Raspberry Pi y su correspondiente programación con la que además se tiene acceso a las GPIO (Entradas y/o Salidas de Propósito General), estas a su vez nos permiten implementar indicadores, como lo es el caso de un diodo LED, y también actuadores, como un botón. A partir de ahora

describiremos las partes tanto de hardware y software que hacen posible la funcionabilidad del presente proyecto.

2.2MINICOMPUTADORA RASPBERRY PI MODELO B

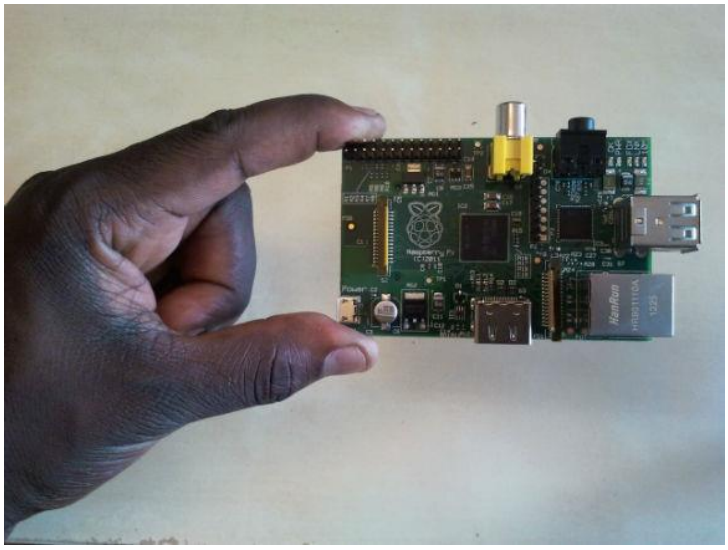


Figura 2.1 Tamaño de Raspberry Pi modelo B[7]

La base fundamental del proyecto es la tarjeta Raspberry Pi que no es más grande que una tarjeta de crédito como se muestra en la Figura 2.1la cual posee las características de procesar datos tal como lo hace una computadora de la que consideramos comunes en nuestro entorno.

El sistema operativo en el que corren los programas se basa en Linux en su versión Debian con la distribución RaspbianWheezy recomendada por la fundación Raspberry Pi.

En el corazón del sistema Raspberry Pi hay un procesador multimedia system-on-chip (SoC) Broadcom BCM2835, el cual se muestra en la Figura 2.2**.**Sin embargo, no sólo que este diseño del SoChace que el BCM2835 sea diferente al procesador se encuentra en su desktop o laptop. Esto también usa una arquitectura de configuración de instrucción diferente conocida como ARM. El BCM2835 basado en ARM es la clave para que la Raspberry Pi esté apto para operar con una fuente de poder de 5V a 1A conectada a la tarjeta por medio del puerto micro-USB que posee el dispositivo. **[8]**

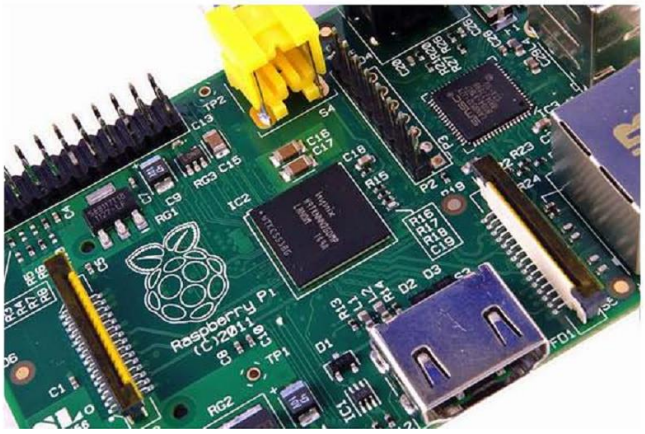


***Figura 2.2 Procesador multimedia Broadcom BCM2835*[8]**

2.2.1CARACTERÍSTICAS

A continuación describimos las especificaciones técnicas más sobresalientes del fabricante:

- Procesador (con gráfica integrada):

Broadcom BCM2835. Contiene ARM1176JZFS, con unidad de coma flotante, funciona a 700Mhz y un Videocore 4 GPU.

- Memoria:256 MB
- Características Técnicas de la GPU:
  - La GPU es capaz de mover contenidos con calidad Bluray, usando H.264 hasta 40MBits/s. Dispone un core 3D con soporte para las librerías OpenGL ES2.0 y OpenVG. Es capaz de decodificar 1080p30 H.264 high-profile.**[9]**
- Dispositivo de Arranque:
  - Memoria SD card.
- Conectores:
  - 2x Conectores USB 2.0, Conector Ethernet RJ-45 10/100
  - Salida de Video Digital HDMI (Audio y Video)
  - Salida de Video Analógico (S-Video)
  - Audio Analógico (Conector 3,5mm
  - Conector GPIO

- Conector de alimentación Micro USB
- Lector de memorias SD (Utilizado para arrancar el dispositivo)

- Alimentación:
  - Vía Micro USB 5 Voltios.
- Sistema sOperativos Soportados:
  - Raspbian “wheezy” (Debian), ArchLinux, Fedora, QtonPi (QT SDK), OpenElec, Raspbcm, Android (en desarrollo por usuarios).**[9]**

2.2.2 PROCESADOR

La parte principal de la Raspberry Pi es el procesador multimedia BroadcomBCM2835 lo que significa que la mayoría de los componentes del sistema además de las unidades de procesamiento central y grafico junto con el hardware de audio y comunicaciones están incorporados sobre un componente oculto bajo un chip de memoria de 256 MB en el centro de la tarjeta.

Esto también usa una diferente configuración de instrucciones, conocido como ARM. Desarrollado por Arcon Computers.a finales de

los años 80, la arquitectura ARM es muy poco común ver computadoras con este componente. Sin embargo, se destaca en dispositivos móviles: en los teléfonos celulares es casi seguro que al menos un núcleo procesador basado en ARM se encuentre escondido en el interior. La combinación de una simple configuración de instrucción reducida y el bajo consumo hacen la mejor elección considerando que los chips de la desktop que poseen alta demanda de potencia y compleja arquitectura de configuración de instrucciones.

Debido a BCM2835 basado en ARM la Raspberry Pi opera con una fuente de poder de 5V 1A suministrada por el puerto micro-USB de la tarjeta. Por esta razón sobre la tarjeta no se ha ubicado disipadores de calor: el bajo consumo de los chips se traduce directamente in muy poco calor, incluso en las tareas de procesamiento complicadas.

En cuanto a la compatibilidad con el software tradicional de una desktop o laptop, hay que considerar que la Raspberry Pi no lo es ya que la mayoría de estas PC's están construidas con la arquitectura de configuración de instrucciones x86 en memoria, tal como se encuentran en procesadores como los de AMD, Intel y VIA. Como resultado esto no correrá sobre Raspberry Pi basado en ARM.**[8]**

2.2.3 DISPOSITIVO DE ARRANQUE

El dispositivo de arranque para que la Raspberry Pi pueda funcionar es una tarjeta de memoria SD y su capacidad puede variar según las aplicaciones que se le quiere dar. En el mercado existen memorias SD de más de 16 GB pero a medida en que aumenta su capacidad así mismo el costo es mayor.

Para poder utilizar la memoria SD en la Raspberry Pi se debe “grabar” un sistema operativo dentro de la misma, esta acción tiene un tiempo de unos pocos minutos ya que es ligeramente más difícil que arrastrar y soltar un archivo sobre la memoria SD. **[8]**

Como un primer paso, para empezar a usar la memoria SD, se debe tomar decisión que distribución Linux se desea usar en la Raspberry Pi. Más adelante veremos sobre las diferentes distribuciones que podemos tener en consideración para la operatividad de esta minicomputadora, aunque cada una de las distribuciones tiene sus ventajas y desventajas no debemos de preocuparnos si por alguna razón cambiamos de opinión y reconsideremos la versión Linux que

después se quiera usar ya que se puede grabar la memoria SD las veces que se desee.

La versión actualizada de Linux compatible con la Raspberry Pi está disponible en la página web que posee el mismo nombre que la minicomputadora la cual es la siguiente y para ser más específicos es donde se puede descargar:

httt://www.raspberrypi.org/downloads.

Además la Fundación Raspberry Pi proporciona enlaces BitTorrent para cada distribución. Se pueden descargar archivos desde otros usuarios con programas BitTorrent para el uso de esto archivos. Con estos enlaces se logra una eficiencia para la distribución de archivos de gran capacidad de memoria y se evita que los servidores de descarga de la Fundación lleguen ser sobrecargados.

Es necesario tener instalado un cliente compatible para el uso del enlace BitTorrent, en el caso de no tener instalado se lo puede

descargar e instalarlo, esto se realizaría antes de descargar la distribución Linux para la Raspberry Pi. El cliente para Windows, OS X y Linux es uTorrent el mismo que se encuentra disponible en: http:/www.utorrent.com/downloads.

Una distribución Debian es la que es recomendable para los que se inician, pero se puede escoger cualquier otra ya que esto depende de cada uno.

Para realizar una descarga más rápida se ha comprimido el archivo imagen que es el que contiene la distribución Linux para la Raspberry Pi. Una vez descargado el archivo comprimido con la distribución escogida, se lo descomprime en algún lugar del sistema donde se ha descargado el este archivo. En gran parte de los sistemas operativos se puede abrir este archivo para luego proceder a extraer el contenido del mismo.

Una vez que se ha descomprimido el archivo el cual lo obtenemos como img, que es el que contiene una copia exacta de una tarjeta SD configurada por los creadores de la distribución de tal manera que la

Raspberry Pi la entienda esa distribución. Este archivo deberá ser "grabada" en la tarjeta SD para poder utilizarla en la Raspberry Pi. **[8]**

La manera de realizar el flash (grabar) de la tarjeta SD se indica en el Apéndice.

2.2.4 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.

Además, en la mayoría de las computadoras de uso personal tienen la capacidad de que en ellas se pueda tener uno o dos sistemas operativos tales como Microsoft Windows o Apple OS X. estas plataformas son de código cerrado, el mismo que son creados de manera que no se pueda compartir libremente ya que se usa técnicas de propietario. Los usuarios de estos sistemas operativos conocidos como código cerrado o privado, pueden ser capaces de obtener un software o programa terminad, pero nunca podrán ver como se hizo.

La cualidad que diferencia a la Raspberry Pi es que 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. La portabilidad es importante para el funcionamiento de la Raspberry Pi, ya que gracias al desarrollo del código abierto ha permitido que Linux se pueda modificar para poder modificarse para la ejecución sobre la Raspberry Pi. Diferentes versiones de Linux, conocidas como distribuciones, se han instalado en la Raspberry Pi las cuales incluyen Debian, Fedora Remix, y Arch Linux. Todas estas distribuciones son de código abierto y 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. Afortunadamente hay muchas alternativas compatibles para la gran mayoría de productos de software común, y aun mejor, la mayoría son libres para usar y de código abierto. **[8]**

2.2.5 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 ese longitud de los pines en electrónica, y este es el espacio estándar para plataformas prototipo.

Cada pin del puerto GPIO tiene su propio propósito, con varios pines trabajando juntos se pueden formar circuitos particulares.

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. Asegurarse que al momento de implementar un circuito con el puerto GPIO de la Raspberry Pi, este debe ser compatible con el voltaje lógico de 3.3 V o pasar el circuito a través de un voltaje regulador antes de conectarlo a este puerto.

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.

Algunas consideraciones al momento de trabajar con el puerto GPIO de la Raspberry Pi se resaltan a continuación:

2.3 TECNOLOGÍA WI-FI

Con esta tecnología que comprende la interconexión de ordenadores de manera inalámbrica, se puede dar la comunicación entre las minicomputadoras Raspberry Pi sin el uso de cables y por consiguiente poder crear una red de estos dispositivos. Adicionalmente, es necesario usar un equipo conocido como punto de acceso para que funcione el Wi-Fi tal como se muestra en la Figura 2.3.

***Figura 2.3 Punto de acceso*[10]**

Una vez adquirido estos equipos tanto las minicomputadoras Raspberry Pi como el punto de acceso, es necesario que los equipos informáticos tengan junto a ellos unas pequeñas unidades que los comunique con el punto de acceso vía radio, estas unidades son conocidas como adaptadores de red. La unidad de radio es la que hace posible la comunicación de manera inalámbrica de los ordenadores.

Existen otros tipos de tecnologías inalámbricas, que cabe resaltar que son complementarias, tales como la que utiliza la telefonía móvil conocida como GSM (Global System for Mobile Communications, Sistema global de comunicaciones móviles) y las de menor cobertura como la tecnología Bluetooh. Como detalles adicionales, estas no se pueden interconectarse directamente y no tienen nada que ver entre sí. La más usada de estas sin duda es la tecnología GSM ya que por su demanda alcanza grandes distancias.

***Figura 2.4 Tecnología GSM y Bluetooth*[10]**

El organismo que normalizan los estándares de comunicación inalámbrica Wi-Fi considerado el más relevante es el IEEE (Institute of Electrical and Electronics Engineers, Instituto de ingenieros eléctricos y electrónicos) quien ha ido desarrollando soluciones para la trasmisión de datos a velocidades superiores a los 11 Mbps que es la velocidad máxima de datos del estándar original de Wi-Fi. A continuación detallamos las normas más interesantes:

- IEEE 802.11b (1999). Siendo esta la norma original permite velocidades de transmisión de hasta 11 Mbps con bandas de frecuencias de 2.4 GHz.

- IEEE 802.11a (1999). Usa bandas de frecuencias de 5 GHz a diferencia de la anterior de 2.4 GHz, y además alcanza velocidades de 54 Mbps pero puede llegar hasta los 72 y 108 Mbps con versiones propietarias como es el caso de Netgear.

- IEEE 802.11g (2003). Esta norma permite la transmisión de datos a una velocidad de 54 Mbps en la banda de frecuencias de 2.4 GHz. Sin embargo, en versiones propietarias la velocidad es de 100 Mbps como la de US Robotic.

- IEEE 802.11n (2009). En comparación a las normas anteriores esta norma habla sobre velocidades de 300 Mbps y de mayores alcances. Otras ventajas más de la norma 802.11n es la compatibilidad con los anteriores estándares (a, b y g) y la incorporación de varias antenas lo que hace posible utilizar canales simultáneamente. Aun esta norma está en desarrollo para alcanzar los 600 Mbps a pesar de estar aprobada en 2009.

Así como las cámaras de inalámbricas suelen disponer de un servidor web interno que es donde se tiene acceso a su configuración y a su contenido

que en su caso son las imágenes, de la misma manera un ordenador puede ser configurado y acceder a la información que contenga este. Del servidor web se desarrollará más adelante en la sección 2.4. **[11]**

2.4 HERRAMIENTAS DE SOFTWARE

En esta sección especificamos la ayuda que obtenemos al utilizar herramientas para el soporte de la programación y comunicación de las minicomputadoras Raspberry Pi. El software es importante ya que corresponde al 50% de la estructura de un sistema informático.

2.4.1 SERVIDORES WEB

El funcionamiento básico del servidor web se da por medio de un intercambio de datos o archivos que se cargan en el servidor y este a su vez cumple el trabajo de servirlo por la red de un navegado ya sea este Internet Explorer, Google Chrome, Firefox, entre otros. Este tipo de archivos son contenidos estáticos y para que pueda haber la comunicación entre servidor y navegador se usa el protocolo HTTP. Este proceso lo podemos aprecien en la Figura 2.5.

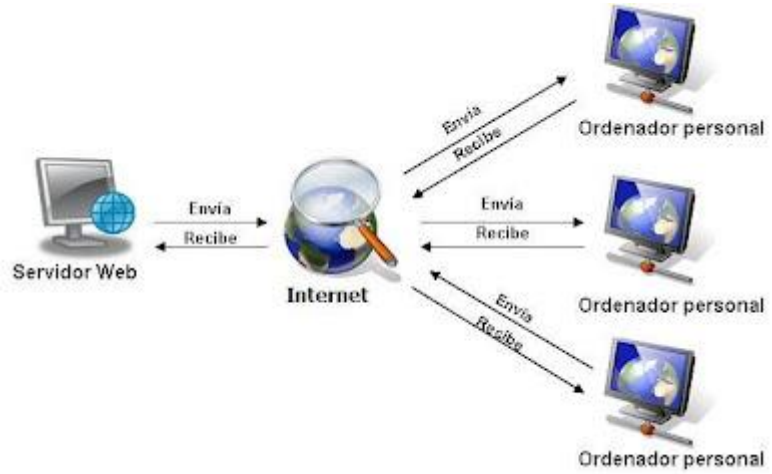


***Figura 2.5 Servidor web*[12]**

En otras palabras podemos decir que un servidor web es el software continuamente se está ejecutando en un computador, al mismo tiempo se esperan peticiones de ejecución por parte de un cliente o un usuario de Internet. Estas peticiones que el servidor recibe son contestadas por él mismo de forma adecuada, obteniendo como resultado una página web o información de todo tipo, esto en concordancia a lo solicitado. De una manera técnica, un servidor es el software que permite realizar tales funciones como las respuestas a las solicitudes pero el término servidor se ha generalizado para referirse a una computadora con un software servidor instalado en la misma. Con

esto, el software servidor es la parte fundamental para poder usar un ordenador con dichas descripciones. **[8]**

Un aumento en la potencia del servidor web a más de entregar páginas web o HTML es por medio del uso de tecnologías las que incluyen scripts CGI, seguridad SSL y páginas activas del servidor.

Una manera de hacer más eficiente una corporación, es establecer servidores web dentro de la misma empresa, ahorrando caros alojamientos en proveedores externos y esto como consecuencia del desarrollo de las redes de datos y a una gran disponibilidad de banda de ancha. Un software que hace posible esto es Apache, este es entre los servidores web el que tiene un mayor uso y considerado uno de los mejores. Esta consideración se debe a que Apache posee una gran estabilidad, confiabilidad y el uso de grupo de voluntarios que planean y desarrollan todo en cuanto a esta plataforma, desde la documentación hasta el mismo código en sí. De esta manera este servidor web se ha ganado una excelente reputación entre los de su misma clase. **[12]**

2.4.2 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.

De los protocolos más comúnmente utilizado para ver páginas Web es el Hyper Text Transfer Protocol (HTTP). Protocolos como el Hyper Text Transfer Protocol sobre Secure Sockets Layer (HTTPS), y File Transfer Protocol (FTP), un protocolo para subir y descargar archivos, también son soportados.

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.

Da el caso que si queremos la Raspberry Pi usarla como servidor web debemos de instalar Apache2 usando el siguiente comando que tiene el lenguaje Linux:

Sudo apt-getinstall apache2 y presionamos enter. **[13]**

2.4.3PHP5

PHP (acrónimo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto que por conocimiento de programadores es el especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. 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.

Con las primeras 2 versiones de PHP, PHP 3 y PHP 4, se había conseguido una plataforma potente y estable para la programación de páginas del lado del servidor. Estas versiones han servido de mucha ayuda para la comunidad de desarrolladores, haciendo posible que PHP sea el lenguaje más utilizado en la web para la realización de páginas avanzadas.

Sin embargo, todavía existían puntos negros en el desarrollo PHP que se han tratado de solucionar con la versión 5, aspectos que se echaron en falta en la versión 4, casi desde el día de su lanzamiento. Nos referimos principalmente a la programación orientada a objetos (POO) que, a pesar de que estaba soportada a partir de PHP3, sólo implementaba una parte muy pequeña de las características de este tipo de programación.

El principal objetivo de PHP5 ha sido mejorar los mecanismos de POO para solucionar las carencias de las anteriores versiones. Un paso necesario para conseguir que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos, incluso los más exigentes. **[14]**

2.4.4 LIBRERÍA WIRINGPi

Esta librería dedicada para trabajar en la minicomputadora Raspberry Pi viene con un programa separado de ayuda de manejo del GPIO. Este programa, llamado gpio, pueden también ser usados en scripts para manipular los pines GPIO.

Para la instalación de WiringPi, la cual podremos usar para programar los pines GPIO, seguiremos los siguientes pasos.

Por línea de comandos, instalamos git-core si no lo tenemos instalado se escribe el siguiente comando:

sudo apt-get install git-core

Ahora obtenemos wiringPi mediante git con el siguiente comando:

git clone git://git.drogon.net/wiringPi

Ahora nos movemos a la carpeta donde se descomprimió que eswiringPi.

cdwiringPi (carpeta descomprimida)

./build (comando para construir un script)**[6]**

CAPÍTULO 3

DESARROLLO DEL PROYECTO

3.1 RESUMEN

En el presente capítulo presentamos las etapas que llevan a cabo el desarrollo y funcionamiento del proyecto. De esta manera, se puede entender cómo trabajan los componentes fundamentales para el objetivo general del proyecto, además de la combinación del desarrollo de estos componentes como son las entradas/salidas de propósito general (GPIO), es decir programar la Raspberry Pi para el uso de estos puertos.

Previo a la realización de los ejercicios se da a conocer que los archivos en los que se ejecutan los programas para las respectivas prácticas son scripts es decir, como se ha instalado PHP5 los archivos en los que se trabajan son

archivos de texto, en cada código realizado se lo guarda en con una extensión php como un ejemplo se guarda el archivo como ejemplo.php, con esto el programa de php reconoce e interpreta el código desarrollado. Estos archivos se los puede cambiar si se desea mediante cualquier editor de texto instalado en la Raspberry Pi.

Para ejecutar un programa desde que arranca el sistema operativo de la minicomputadora se usa el fichero rc.local, es decir que para ejecutar un comando o un script cada vez que se inicia el sistema operativo tipo Linux se lo puede realizar llamando el comando o el script desde el fichero /etc/rc.local. La ejecución se da al final del arranque cuando todos los scripts de nivel superior hayan sido ejecutados.

3.2 EJERCICIO 1: CÓMO TRABAJAR CON LOS PUERTOS GPIO DE RASPBERRY PI

En este primer ejercicio vamos a censar una entrada y aplicaremos el uso de los puertos GPIO de la minicomputadora Raspberry pi. Inicialmente para

poder realizar este ejercicio tenemos que disponer de un circuito electrónico conformado por una botonera que sirve de entrada para la uno de los puertos de la minicomputadora usada, otro componente para este circuito es el uso de un indicador que para este caso un diodo LED es lo más básico y útil para el ejercicio. La parte principal se basa en la programación de la Raspberry Pi mediante un lenguaje de alto nivel conocido como PHP. Para esta acción el uso de la librería wiringPi-PHP es de gran ayuda ya que es un lenguaje muy conocido con sentencias fáciles de entender y usar para la programación de los puertos de propósito general. La botonera usada es la que se presiona para poner en acción la práctica ya que esta es la entrada que se censa, con esto se logra el encendido del LED indicador el cual brilla por el tiempo de un segundo después de haber presionado dicha botonera.

DIAGRAMA DE FLUJO

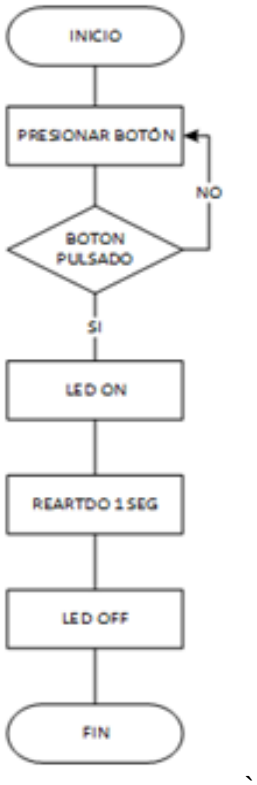


Figura 3.1 Diagrama de Flujo Ejercicio 1

ALGORITMO

1. Inicialmente el programa creado se ejecuta en php.
2. El botón conectado al pin 8 correspondiente a wiringpi (pin 3 de la GPIO Broadcom) es censado. Que además se encuentra con lógica negativa.
3. Al presionar el botón se da la instrucción de encender el diodo LED.
4. El LED brilla por un segundo luego se apaga.

CÓDIGO FUENTE

```
<?php
ini_set("enable_dl","On");
require_once('../libs/WiringPi-PHP/wiringpi.php');
require_once('../libs/gpioBoton.php');
// Como trabajar con los puertos gpio
// del raspberry pi
main();
function main()
{
if (wiringPiSetup() == -1)
exit (1) ;
  $pinBoton = 8; // unabotonera, input
$pinLed   = 1; // salida
configBoton($pinBoton);
  //configuramos pinLed como salida
pinMode($pinLed,OUTPUT);
while(true){
if(pulsoBoton($pinBoton)){
echo "Boton fue pulsado\n";
digitalWrite($pinLed,HIGH);
```

```
sleep(1);
digitalWrite($pinLed,LOW);
sleep(1);
    }
usleep(500000);
echo ".";
  }
}
?>
```

CONCLUSIÓN

Para este ejercicio concluimos que el uso de la librería wiringPi contiene funciones como la wiringPiSetup que es la que provee una interfaz con los puertos GPIO y hacer uso de ellos. Otra parte importante para el desarrollo del ejercicio es el uso de la librería wiringPi-PHP que hace que pueda usar las funciones que se encuentran en la librería wiringPi en un entorno donde se usa lenguaje php. Además con el uso de la función configBoton se configura el pin usado como entrada esta función se encuentra en la librería llamada gpioBoton.

3.3 EJERCICIO 2: CÓMO ENVIAR INFORMACION POR EL PROTOCOLO HTTPS

En el presente ejercicio tenemos como objetivo el envío de información a través del protocolo HTTPS. Este protocolo se usa para la seguridad de la información que se transmite entre ordenadores por tal motivo creemos conveniente el uso de este protocolo ya que se está trabajando con un tipo de alarma remota la cual debe asegurarse que solo los usuarios involucrados tengan acceso a la misma. La programación se desarrolla en el lenguaje php con lo cual damos paso a la estructura de un texto plano el mismo que será transmitido.

DIAGRAMA DE FLUJO

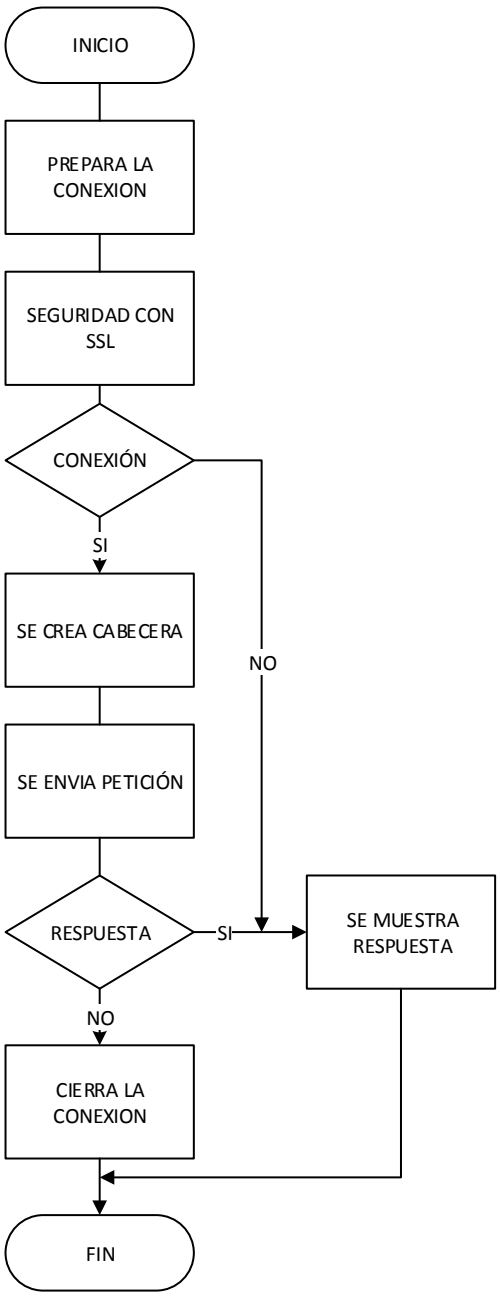


Figura 3.2 Diagrama de Flujo Ejercicio 2

ALGORITMO

1. Ponemos en marcha el programa en php.
2. Se hace conexión ssl para la seguridad de la información que será enviada.
3. Se comprueba si los parámetros de la conexión son los correctos.
4. Con los parámetros correctos se envía la cabecera HTTP hacia el host con la dirección ip al que vamos a transmitir la información.
5. Se muestra en pantalla la respuesta de un envío satisfactorio de la información.

CÓDIGO FUENTE

```
<?php
//Como enviar información por el protocolo http
require_once("../libs/http.php");
$ip_remote   ="192.168.0.50";
$port_remote ="443";
$file_remote ="/practicas/p3.php";
$respuesta = "";
```

```
$conexion = fsockopen("ssl://".$ip_remote,$port_remote);

if ($conexion){
$dataSend   = "saludo=hola&color=amarillo";
$dataLenght = strlen($dataSend);

  $headerRequest = createHeaderHttp($ip_remote,$file_remote,$dataLenght);
fputs($conexion,$headerRequest.$dataSend);
echo "Se enviopeticion al servidor: \n$headerRequest$dataSend\n\n\n";

while(($r = fread($conexion,2048)) != ""){
$respuesta .=$r;
}
fclose($conexion);
}

echo "Se recibio del servidor: \n$respuesta\n";

?>
```

CONCLUSIÓN

Este ejercicio es muy útil ya que nos permite saber cómo el servidor web apache entiende el protocolo https, al momento que el cliente envíe un archivo al servidor este lo entiende y lo interpreta, cabe recalcar que se usa el puerto 443 por esta puerto abro la conexión de lado del cliente por medio de socket (función que me permite comunicarme por medio de la red a otras páginas). Si la conexión es satisfactoria se desarrolla el cuerpo.

3.4 EJERCICIO 3: CÓMO CAPTURAR INFORMACION POR EL PROTOCOLO HTTP

Un ejercicio sencillo para capturar o receptar información del protocolo HTTP, lo presentaremos a continuación. Por medio de esta práctica veremos la manera como trabaja la programación del Raspberry Pi servidor para poder capturar la información que el cliente envía. Además se da a conocer el uso de una librería que ayuda a la implementación del ejercicio y por ende del proyecto de tesis.

DIAGRAMA DE FLUJO

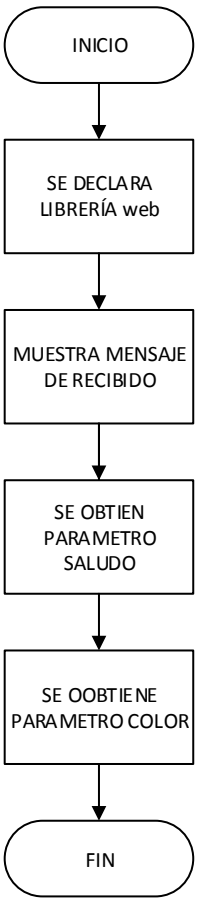


Figura 3.3 Diagrama de Flujo Ejercicio 3

ALGORITMO

1. Se ejecuta el programa realizado en php.
2. Se crea librería para obtener los parámetros que van a ser capturados.
3. Presenta en pantalla mensaje de información recibida.

4. Se obtienen los parámetros que se la ha enviado mediante la función que captura estos parámetros.

CÓDIGO FUENTE

```
<?php
//Como capturo información del protocolo http.
require_once('../../libs/web.php');

echo "ok recibido\n";

echo "saludo= ".getParameter("saludo");
echo "\n";
echo "color= ".getParameter("color");
echo "\n";

?>
```

CONCLUSIÓN

Se concluye que el punto clave de esta práctica es el uso de la librería web.php con la que se obtiene los parámetros que en este caso son saludo y color. Esta

librería contiene la función getParameter con la que verifica si los parámetros están declarados si no lo están devuelve null o nada. De manera general este ejercicio nos muestra la forma de cómo obtener información enviada a través de parámetros declarados para mostrarlos posteriormente en pantalla.

3.5 PROYECTO TERMINADO

Teniendo en cuenta las practicas realizadas anteriormente podemos dar paso la realización del proyecto de tesis el mismo que consiste en definir una comunicación WI-FI entre dos Raspberry Pi y de esta manera dar una orden de encendido o apagado de alarma en uno de ellos para que el otro cumpla con la función de apagar o encender la alarma. Estas órdenes se las realizan por medio de circuitos electrónicos básicos, se los llama básico ya que solo se usan dispositivos como botoneras, LEDs y resistencias, la comunicación de estos circuitos está dada por los puertos GPIO de sus respectivas Raspberry Pi. Dado que el sistema operativo que esta minicomputadora usa es Linux y se requiere un servidor web, se ha instalado el software Apache que es un servidor web dedicado para Linux y que además interpreta el lenguaje php que usaremos para la programación.

DIAGRAMA DE BLOQUES

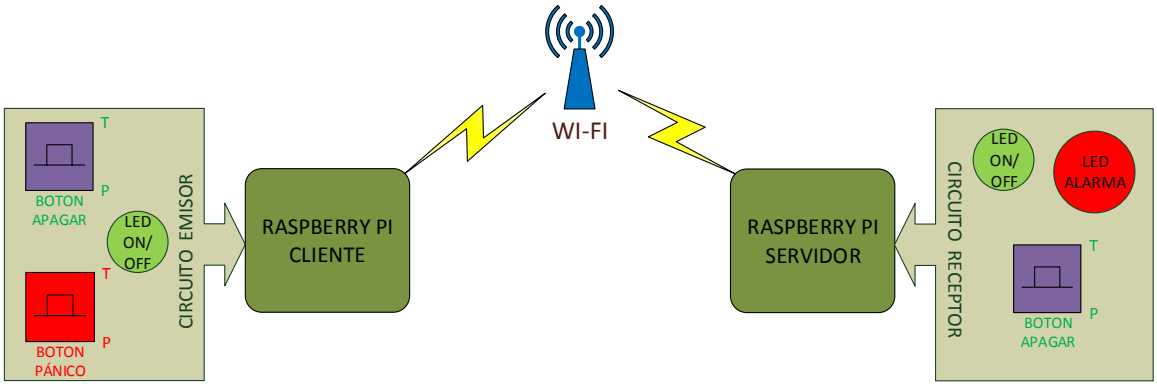


Figura 3.4 Diagrama de Bloques del Proyecto

DIAGRAMA DE FLUJOCLIENTE

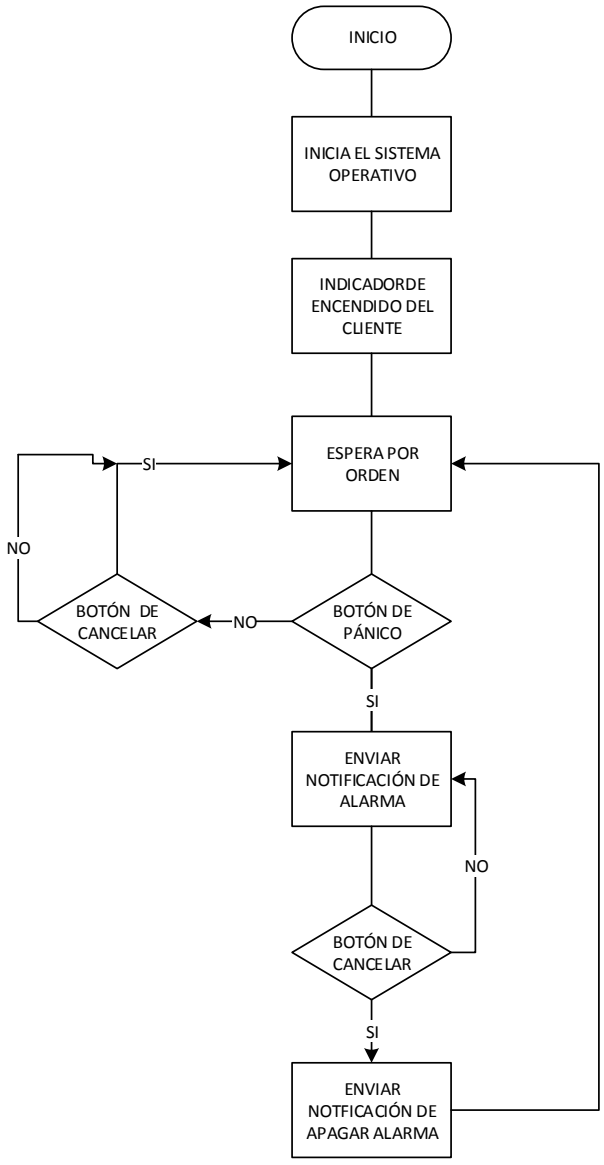


Figura 3.5 Diagrama de Flujo del proyecto escenario Cliente

DIAGRAMA DE FLUJOSERVIDOR

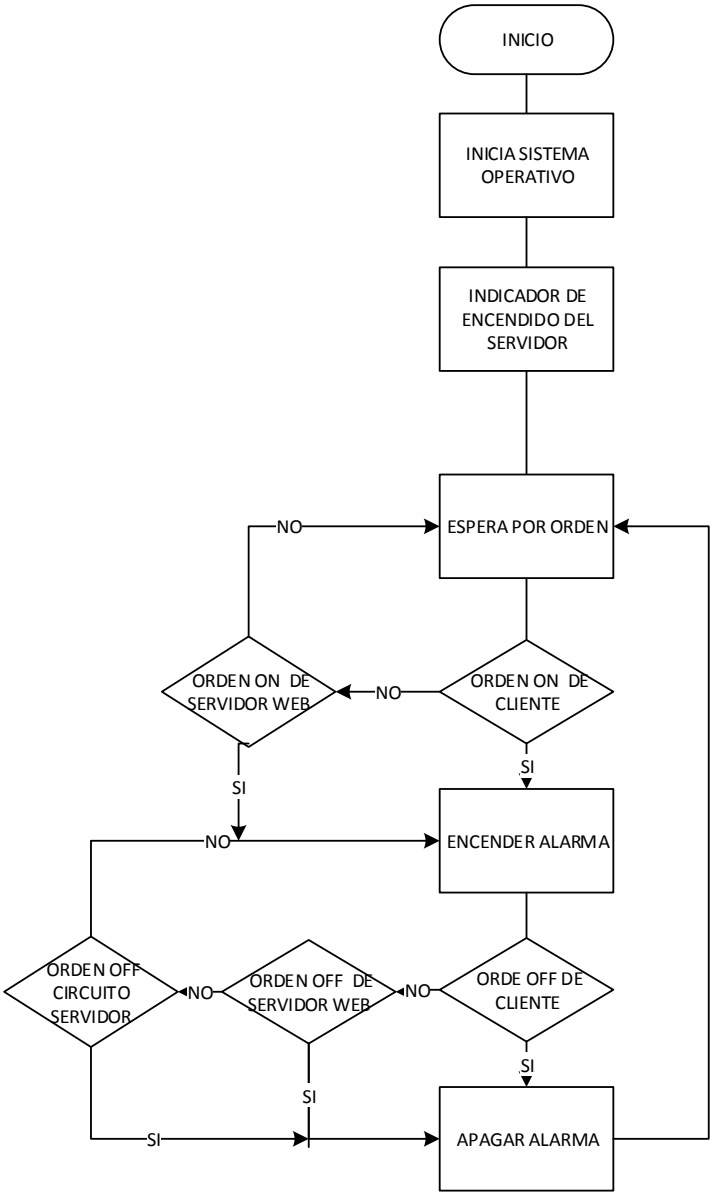


Figura 3.6 Diagrama de Flujo del proyecto escenario Servidor

CÓDIGO FUENTECLIENTE

```
<?php
ini_set("enable_dl","On");
require_once('../libs/WiringPi-PHP/wiringpi.php');
require_once('../libs/gpioBoton.php');
require_once('../libs/logs.php');
require_once('../libs/http.php');



$ipserver   = "192.168.0.50";
$accesscode = "qazwsxedcrfv";
mainProcess($ipserver,$accesscode);

functionmainProcess($ipserver, $accesscode)
{
writeMainLOG("Raspberry Pi start main program client");

if (wiringPiSetup() == -1)
exit (1) ;
```

```
  $pinBotonOffAlm  = 8;
  $pinBotonPanico  = 9;
$pinLedEncendido = 4; // led para indicar inicio del programa

  //Estado inicial del proceso
configBoton($pinBotonOffAlm);
configBoton($pinBotonPanico);
configLed($pinLedEncendido);
encenderLed($pinLedEncendido);

  //El proceso
while(true){
if(pulsoBoton($pinBotonOffAlm)){
parpadearLed($pinLedEncendido);

writeMainLOG("APAGAR ALARMA");
      $respuesta = httpRequest($ipserver,$accesscode,'ALARMA','OFF');
writeMainLOG($respuesta);
    }
```

```
if(pulsoBoton($pinBotonPanico)){
parpadearLed($pinLedEncendido);

writeMainLOG("AVISO DE ALARMA");
     $respuesta = httpRequest($ipserver,$accesscode,'ALARMA','ON');
writeMainLOG($respuesta);
   }

usleep(250000); //espero 1/4 de segundo
}
}
?>
```

CÓDIGO FUENTE SERVIDOR

```
<?php
ini_set("enable_dl","On");
require_once('../libs/WiringPi-PHP/wiringpi.php');
require_once('../libs/resources.php');
require_once('../libs/gpioAlarm.php');
```

```
require_once('../libs/logs.php');
mainProcess();
functionmainProcess()
{
writeMainLOG("Raspberry Pi start main program");

if (wiringPiSetup() == -1)
exit (1) ;

  $pinAlarm        = 1;
  $pinBoton        = 8;
$pinLedEncendido = 4;
  $estadoAnterior  = "free";
  $estadoAlarma    = OFF;

  //Estado inicial del proceso
configAlarm($pinAlarm);
configBoton($pinBoton);
configLed($pinLedEncendido);
setStatusResource("free");
```

```
deleteDataResource();
encenderLed($pinLedEncendido);

//El proceso
while(true){
    //Alarma y boton proceso global
processBoton($pinBoton,$pinAlarm,$estadoAlarma,$pinLedEncendido);
setAlarm($pinAlarm,$estadoAlarma);

    //Verifico estado recurso
    $status = getStatusResource();

    //Si el estado anterior es lock y actual es libre
    //entonces hay una data nueva por leer
if($status == "free" && $estadoAnterior == "lock"){
writeMainLOG("OBTENIENDO DATOS PARA ALARMA");
setStatusResource("lock"); //bloquedo el recurso

     $arrData = getDataResource();
if(isset($arrData) && count($arrData)==4){ //la data esvalida
```

```
if($arrData['valor']== 'ON')
        $estadoAlarma = ON;
else if($arrData['valor']== 'OFF')
$estadoAlarma = OFF;
writeMainLOG("{$arrData['ip']}|{$arrData['accion']}|{$arrData['valor']}|{$arrData['or
igen']}");
}
else
writeMainLOG("FORMATO INVALIDO");

deleteDataResource();
setStatusResource("free"); //libero el recurso
   }

   $estadoAnterior = $status;
usleep(500000); //espero medio segundo
 }
}
?>
```

```
//*****************************************************************************
// Código html para encender o apagar alarma desde un ordenador con
//conexión  a nuestra red.
//*****************************************************************************
<html>
<head>
</head>
<boby>
<form method="post" action="" >
<table border="1">
<tr>
<td colspan="2">
Ingrese codigo de seguridad
</td>
</tr>
<tr>
<tdcolspan="2">
<input type="text" name="c" value="" />
</td>
</tr>
```

```
<tr>
<td rowspan="2">
Alarma
</td>
<td>
<input type="submit" name="v" value="ON" width="50" />
</td>
</tr>
<tr>
<td>
<input type="submit" name="v" value="OFF" />
</td>
</tr>
</table>
<input type="hidden" name="a" value="ALARMA" />
</form>
</body>
</html>
<?php
require_once('../libs/resources.php');
```

```
require_once('../libs/logs.php');
require_once('../libs/web.php');
require_once('../libs/validar.php');
$ip_remote   = $_SERVER['REMOTE_ADDR'];
$access_code = getParameter("c");
$accion      = getParameter("a");
$valor       = getParameter("v");

# Paso 1: Validamos el acceso.
if(!validarAcceso($access_code)){
doResponse("error","Acceso no permitido.");
}
# Paso 2: Validamos la accion.
if(!validarAccion($accion)){
doResponse("error","Accion no es válida, $accion.");
}
# Paso 3: Validamos el valor.
if(!validarValor($valor)){
doResponse("error","Valor no definido, $valor.");
}
```

```
# Paso 4: llamada al proceso principal
mainProcess($ip_remote,$accion,$valor);

functionmainProcess($ip_remote, $accion, $valor)
{
if(waitFreeResource(20)){
if(setStatusResource("lock")){
$data = "$ip_remote|$accion|$valor|HOST REMOTO";
if(setDataResource($data)){
sleep(2);
if(setStatusResource("free")){
doResponse("ok","Accion fue procesada correctamente.");
        }
}
    }
  }
doResponse("error","No se puedo procesar la acción, el servidor no responde.");
}
?>
```

ALGORITMOCLIENTE

1. Encender las Raspberry Pi y el punto de acceso, con esto se ejecuta automáticamente los respectivos programas en las minicomputadoras.
2. Se indica que el programa principal está ejecutándose en la parte cliente y servidor de la Raspberry Pi, esto se observa mediante el encendido de un LED que permanezca así todo el tiempo mientras la minicomputadora no sea apagada.
3. Mientras se esté ejecutando el programa en la Raspberry Pi cliente se espera a que se dé una orden de activar la alarma que se encuentra del lado del servidor.
4. Si no se activa la alarma y se presiona el botón de apagado no se realizará ninguna acción.
5. Al presionar el botón de pánico o también llamado botón de activar alarma se envía la señal y notificación de que se ha ordenado encender la alarma en nuestra minicomputadora servidor además de un mensaje que se ha recibido la señal.
6. Se apaga la alarma si se presiona el botón de apagar montado en el circuito conectado en la Raspberry Pi cliente, de esta manera se envía la señal que desactivará la alarma en el servidor.

7. El sistema está por la espera de una orden en este caso de encender o apagar alarma hasta que la Raspberry Pi sea apagada.

ALGORITMOSERVIDOR

1. Una vez encendido la Raspberry Pi y el punto de acceso, se ejecuta automáticamente el programa implementado para el lado del que se ha configurado como servidor web que es que el recibe las ordenes de encender y apagar alarma por parte del cliente pero también usamos comunicación HTTPS para apagar o encender la alarma desde un ordenador que se encuentre conectado a nuestra red.
2. Se indica que el programa principal está ejecutándose en la parte servidor de la Raspberry Pi, esto se observa mediante el encendido de un LED que permanezca así todo el tiempo mientras la minicomputadora no sea apagada.

3. Se espera la orden de encender alarma que es enviada desde el circuito electrónico de la Raspberry Pi cliente. También se espera la misma orden de un computador o host que se encuentre conectado a nuestra red, pero cabe decir que para poder dar una orden debe cumplir requisitos de seguridad como una palabra clave.

4. Se enciende la alarma que para nuestro caso es un LED que tiene como entrada una señal PWM (pulse wide modulation) que hace que el diodo se encienda y apague de manera intermitente.
5. La alarma se desactiva de las tres siguientes maneras:
    - Se ordena la desactivación de la alarma por medio de una señal eléctrica que es transmitida desde el circuito electrónico conectado a la Raspberry cliente.
    - En el circuito electrónico conectado al servidor se encuentra un botón de apagado de alarma, el mismo que da la orden de realizar esta acción.
    - Un ordenador, que tenga conexión a la red WI-FI que se ha configurado previamente, da la orden via web de apagar la alarma activada, de igual manera que se hace para activar la alarma se necesita de un código de acceso o palabra clave para poder realizar el apagado de la alarma.
6. El LED deja de brillar al aceptar la orden de apagar alarma.

CAPÍTULO 4

SIMULACIONES

4.1 RESUMEN

En este capítulo mostramos los resultados obtenidos de cada uno de los ejercicios propuestos en el capítulo anterior, se presenta el listado de materiales que fue es necesario para cada ejercicio.

4.2 SIMULACIÓN EJERCICIO 1

La figura 4.1 muestra la ejecución del ejercicio 1, la cual muestra en pantalla si es que hay alguna acción de parte del usuario es decir si la botonera fue pulsada, si es así aparece un mensaje: “Botón fue pulsado”.

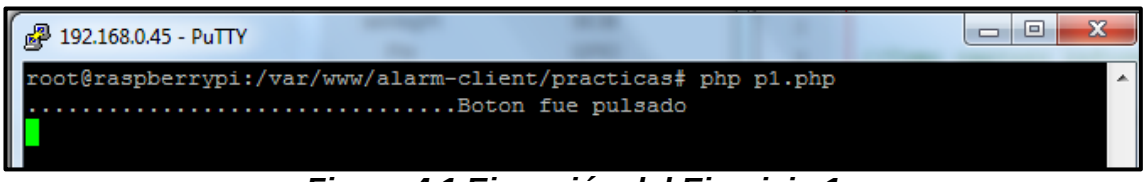


Figura 4.1 Ejecución del Ejercicio 1

La figura 4.2 muestra la implementación del ejercicio 1 el cual es un circuito electrónico que está conformado básicamente por resistencias, botoneras y un led, la comunicación entre el circuito y la Raspberry Pi se realiza mediante los puertos GPIO, de esta manera al momento de estar ejecutando el ejercicio 1 se pulsa la botonera y prende el led, nos podemos dar cuenta que la botonera fue pulsada al ver en pantalla el mensaje de que Botón fue pulsado.

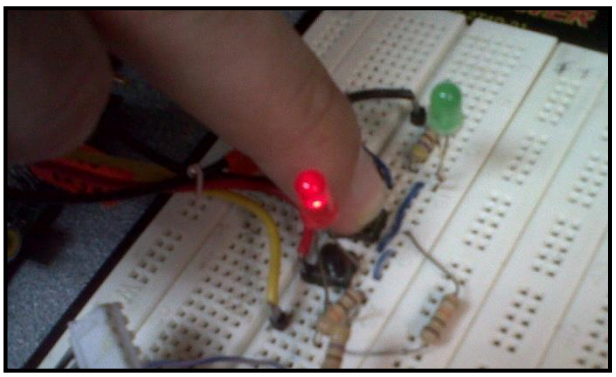


Figura 4.2 Implementación del ejercicio 1

4.3 SIMULACIÓN EJERCICIO 2

En este ejercicio podemos observar el uso del envió de información por HTTPS, en la figura 4.3 observamos como el cliente envía un archivo de texto plano que está conformado por una cabecera y un cuerpo, en la cabecera se adjunta los siguientes datos:

Host: Dirección ip destino, equipo que es de interés para él envió de información y que esté conectado a la misma red del cliente.

Content-Type: Es la clase de tipo de contenido que tendrá el documento. Content-Length: Es el tamaño de carácter que se está enviando al servidor en el caso del ejercicio 2 tiene 26 caracteres y viene dado por la siguiente expresión “saludo=hola&color=amarillo” el “&” es utilizado para tener una mejor apreciación y que no se imprima todo unido.

Algo muy importante es el cuerpo del texto plano, aquí se construye una “Dato de Envió” si es que la conexión con el servidor fue satisfactoria. En el caso del ejercicio 2 el dato de envió es “saludo=hola&color=amarillo” luego se muestra un mensaje de confirmación de petición que es: “Se envió petición al servidor”, seguido de la respuesta del servidor, en este caso la respuesta depende de la conexión.

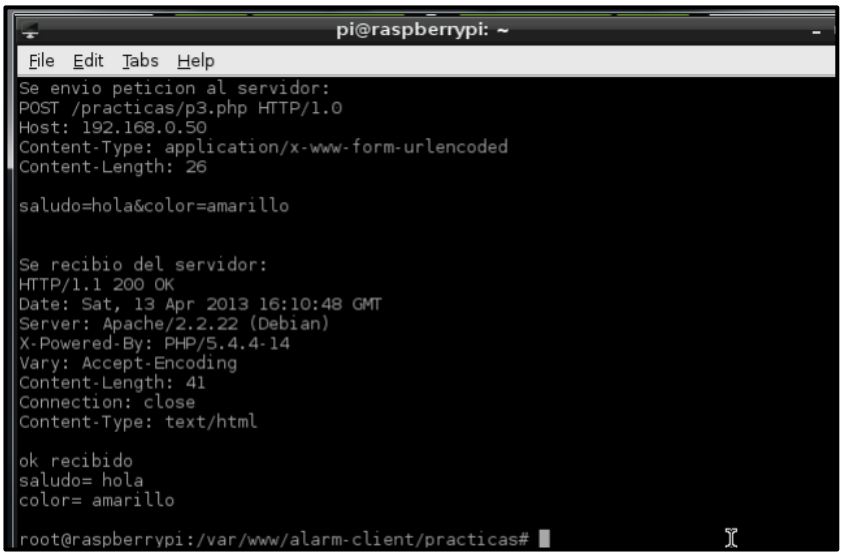


Figura 4.3 Envió de Información por HTTP

4.4 SIMULACIÓN EJERCICIO 3

La captura de información con protocolo HTTPS es decir por el puerto “443”, en la figura 4.4 podemos visualizar la información que el servidor recepta del cliente.

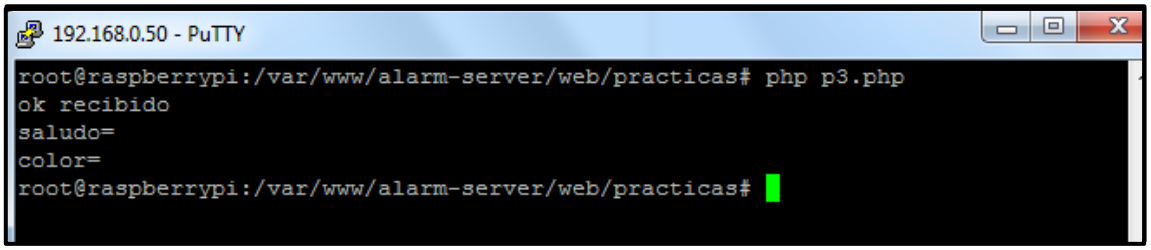


Figura 4.4 Captura de Información en el Servidor

4.7 SIMULACIÓN PROYECTO

La figura 4.5 contiene la implementación del proyecto final asignado, cuyo objetivo principal es la activación de alarmas remotas mediante Wifi haciendo uso de las minicomputadoras Raspberry Pi, se basa fundamentalmente en 2 escenarios: cliente y servidor.

En el escenario del Cliente Figura 4.6, está conformado por un circuito eléctrico y una Raspberry Pi, inicialmente hay un ledque indica inicio del sistema operativo, así como también existen 2 botoneras una para encender alarma y otra para apagarla, en la figura 4.6 podemos observar al usuario presionando la botonera de apagado de alarma.

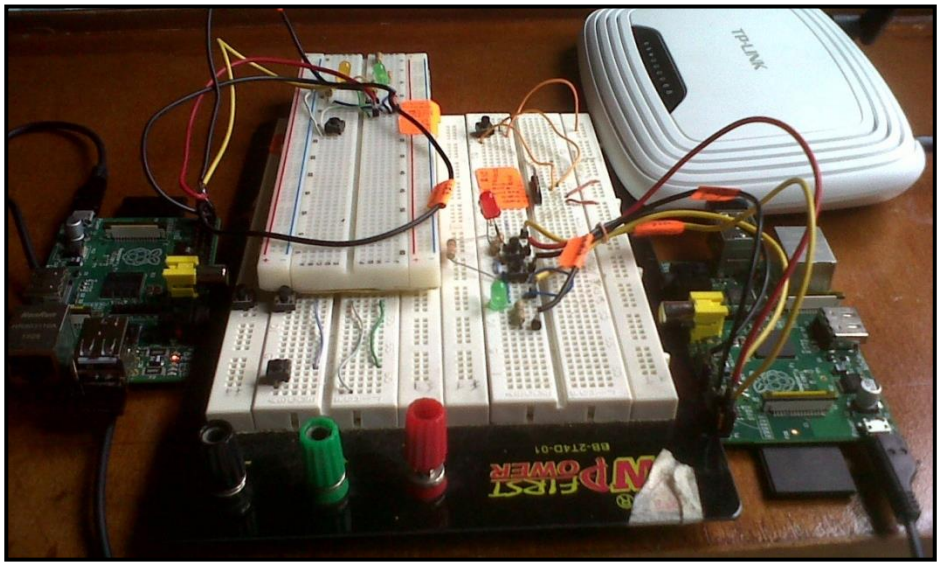
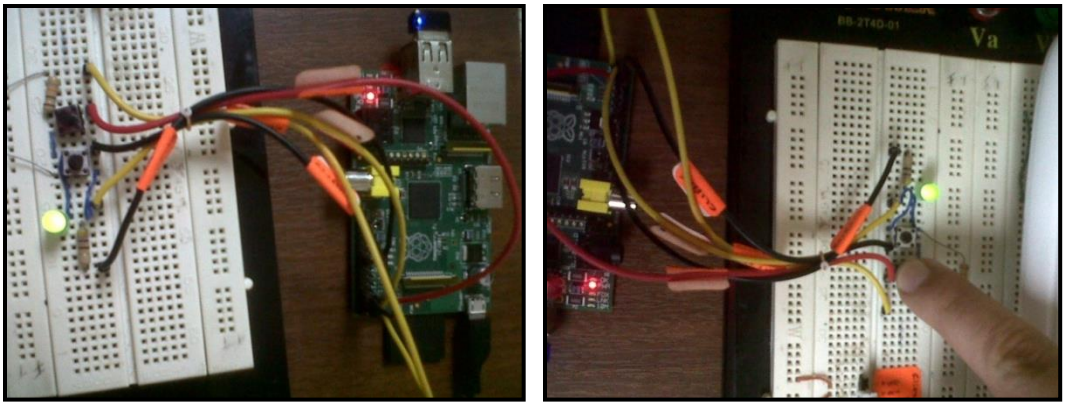


Figura 4.5 Implementación del proyecto terminado



.

Figura 4.6 Cliente mostrando inicialización de programa

En la Figura 4.7 se puede visualizar cómo se envía la información al servidor, al ser presionada la botonera de alarma se muestra el mensaje “Aviso de alarma”.

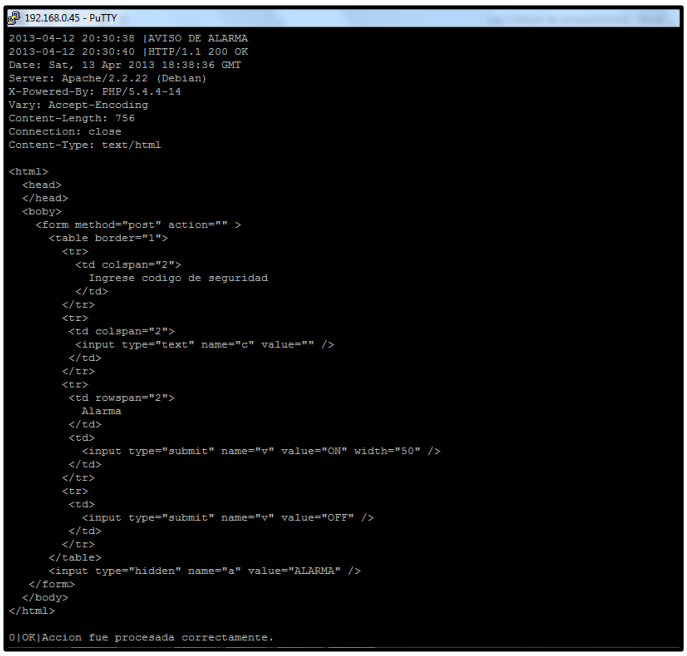


Figura 4.7 Ejecución del programa cliente al presionar la botonera alarma.

En la Figura 4.8 se puede visualizar cómo se envía la información al servidor, al ser presionada la botonera de cancelar alarma se muestra el mensaje “Apagar alarma”.

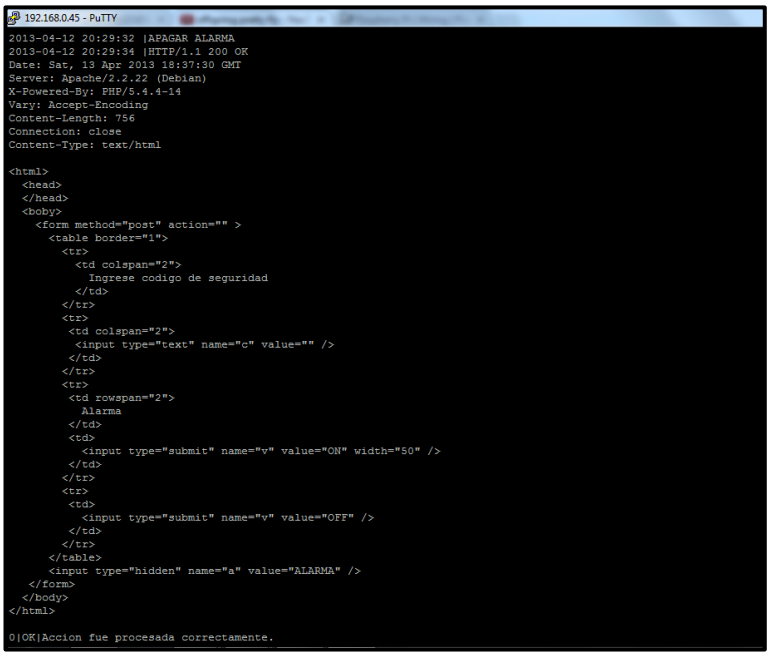


Figura 4.8 Ejecución del programa del cliente al presionar botonera cancelar alarma.

En el servidor tenemos 2 procesos principales el proceso principal main y el proceso principal web; cuando el cliente envía las peticiones con su requerimiento PHP lo recibe la parte web ya que el recibe las decisiones por hacer, cuando inicia el Raspberry Pi inicia el proceso main de forma independiente y se queda iniciado esperando que el proceso web le notifique la tarea por hacer.

En la figura 4.9 podemos observar el escenario del servidor que está conformado por un circuito eléctrico y un Raspberry Pi; el circuito está

compuesto por una botonera que hará el papel de cancelar alarma, así como un led de inicialización del servidor y finalmente el led de encendido de alarma que analizando los datos recibidos del cliente se encenderá de forma intermitente.

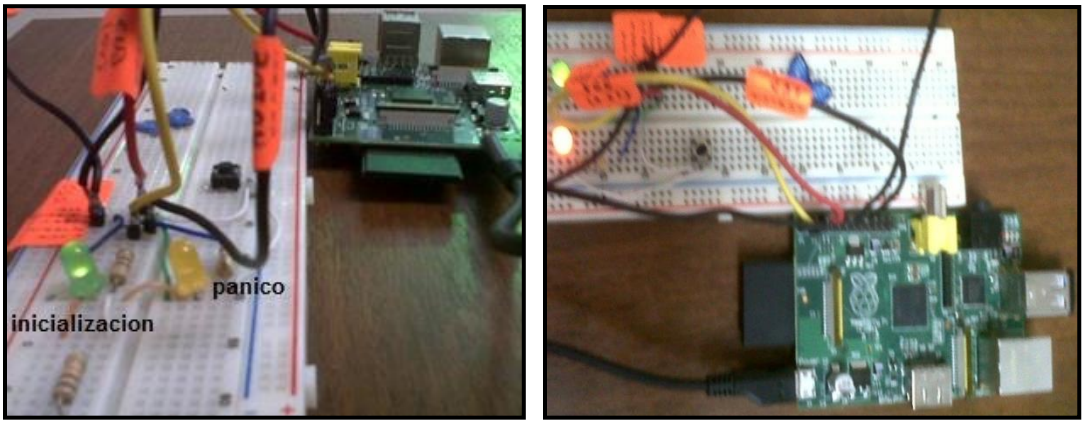


Figura 4.9 Servidor mostrando inicialización del programa y ejecución de Botón cancelar alarma.

En la Figura 4.10 se puede observar otra forma de encendido o apagado de la alarma que va ser por vía web ya que si nos disponemos de uno de los equipos con la facilidad de un ordenador que esté conectado a la misma red del sistema se lo puede hacer.

Como ya mencionamos en el capítulo 3 van existir 3 parámetros para él envió de información por HTTPS uno de ellos es el código en este caso:

qazwsxedcrfv que deberá ser digitalizado antes de encender o apagar la alarma que nos proveerá mayor seguridad.

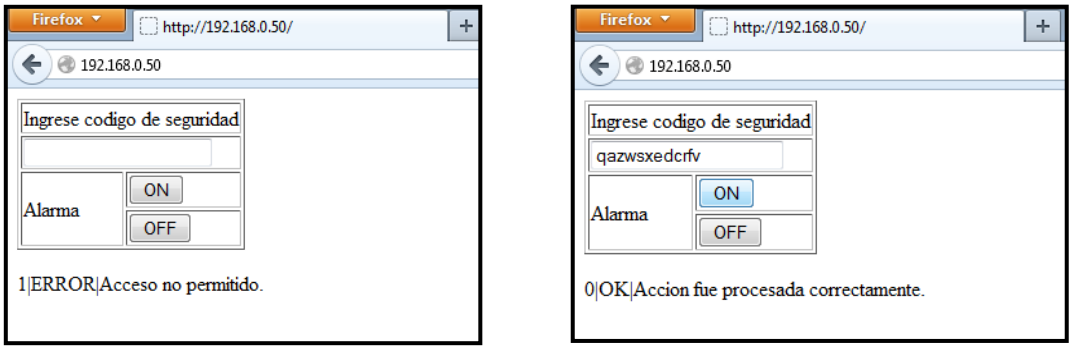


Figura 4.10 Encendido y apagado de alarma vía web.

En la figura 4.11 se puede vizualizar la ip del ordenador que accedio mediante un explorador y ejecuto en este caso el encendido de la alarma presionando el Boton “ON” si se desearia apagarla se debera presionar el boton “OFF”.

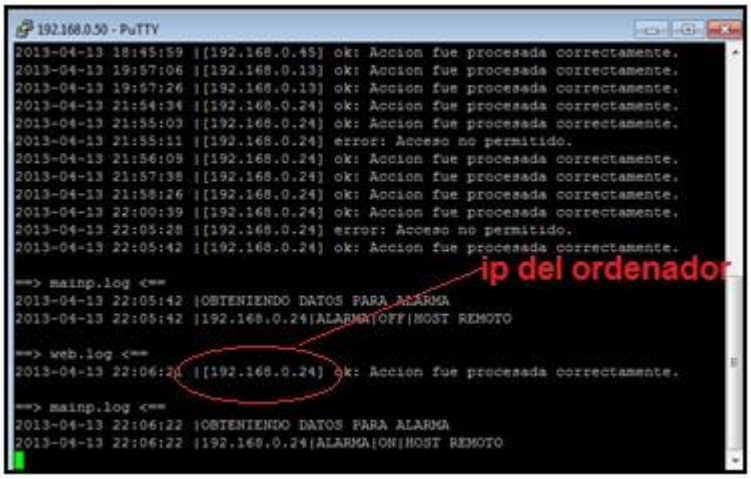


Figura 4.11 Ejecución de alarma en el servidor vía web.

Es decir tenemos 2 formas de encendido de alarma: vía web y desde el cliente así como también para apagar la alarma: vía web, desde el cliente o desde el servidor. Cabe recalcar que cuando se ejecuta alarma “on” u “off” desde un lugar ajeno al escenario de donde se encuentra la alarma (servidor) se podrá observar las direcciones IP de los diferentes dispositivos u ordenadores que puedan realizar alguna acción sobre la alarma. Figura 4.12

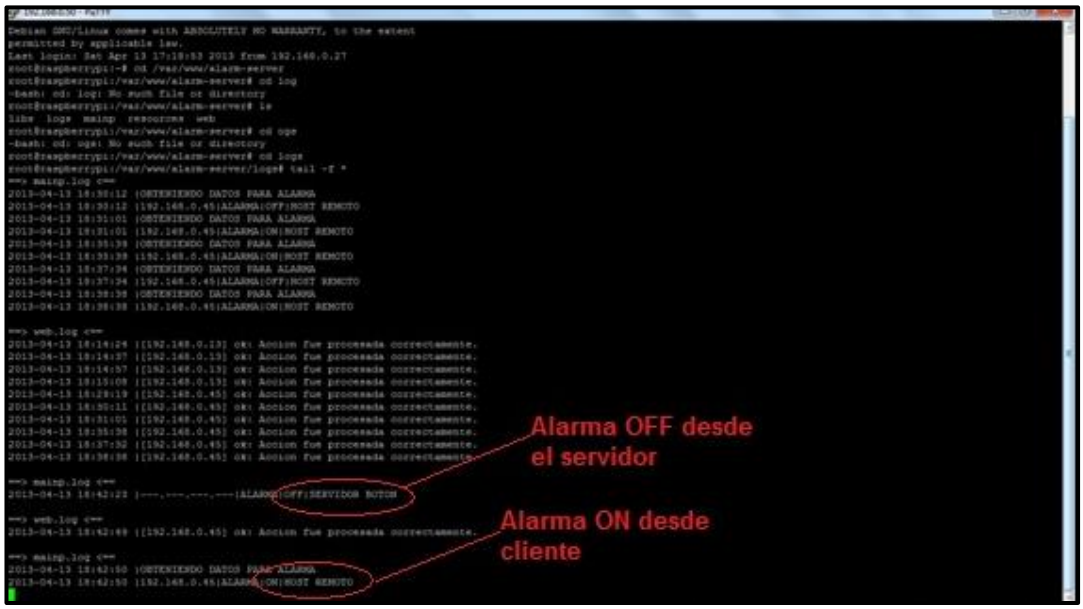


Figura 4.12 Ejecución de alarma en el servidor vía cliente o servidor.

CONCLUSIONES

1. De manera concluyente y en relación con nuestros objetivos decimos que el uso del puerto GPIO permite la comunicación con dispositivos externos cuya configuración sea compatible con la lógica de 3.3 V y 0v que son el estado alto y bajo respectivamente de los pines de uso de propósito general. De esta forma podemos enviar o recibir estos tipos de estado desde o hacia la Raspberry Pi. Con la implementación de un circuito electrónico logramos el adecuado funcionamiento del mismo, configurando correctamente los pines a los que se conecten en el puerto GPIO de la Raspberry Pi.

2. Mediante el desarrollo de los ejercicios y el proyecto pudimos comprender y tener una visión más amplia acerca del funcionamiento de las Raspberry pi así como también sus diversas aplicaciones y la forma en que se realiza la comunicación entre ellos, que en este caso elegimos la comunicación inalámbrica ya que era parte de nuestros objetivos el realizar dicha comunicación entre Raspberry Pi.

3. El estándar IEEE 802.11n es la tecnología conocida como WI-FI, por consiguiente esto sirve para dar comunicación inalámbrica entre los

dispositivos que soporten esta tecnología. Con lo anterior, estos dispositivos consisten en las minicomputadoras Raspberry Pi con los respectivos adaptadores de red y en un punto de acceso que en conjunto arman una red de comunicación WI-FI, en la cual una Raspberry Pi realiza la función de servidor que responde a las peticiones de la otra minicomputadora.

4. La implementación de una alarma remota debe tener la seguridad de que solo las personas involucradas puedan tener acceso para encenderla o si ya está encendida, apagarla, esto se logra con el uso del protocolo HTTPS (Hypertext Transfer Protocol Secure, en español: Protocolo seguro de transferencia de hipertexto) ya que por medio de SSL (Secure Socket Layers) se crea un canal cifrado que permite ser a este tipo de sistemas más confiable.

RECOMENDACIONES

1. Es importante entender primero el funcionamiento individual de cada uno de los microcontroladores que involucra este proyecto antes de combinar sus funciones ya que cada uno de ellos posee su propia estructura y lenguaje.

2. Revisar minuciosamente la sintaxis y tener en cuenta que todas las librerías estén incluidas antes de compilar el código.

3. Documentar correctamente el código para un mejor entendimiento del mismo, detallando cada función implementada con su respectiva descripción.

4. Para no realizar daños irreversibles, es necesario conocer las limitaciones que posee nuestro dispositivo, ya que por el motivo de trabajar con señales eléctricas que se le suministra a la Raspberry Pi puede existir un tipo de peligro para la misma al realizar una conexión no adecuada.

BIBLIOGRAFÍA

- [1] Simon Monk , Programming the Raspberry Pi Getting started with Python , McGraw Hill

- [2] Placa Arduino.
  Disponible en:http://es.wikipedia.org/wiki/Arduino(Consultado el 25 de Marzo del 2013).

- [3] Placa Raspberry Pi.
  Disponible en: http://www.raspberrypi.org/faqs(Consultado el 18 de Marzo del 2013).

- [4] Andrew K. Dennis, Raspberry Pi Home Automation with Arduino, Packt

- [5] Partes de la Raspberry Pi
  Disponible en: http://www.taringa.net/posts/linux/14237599/Tenes-dudas-sobre-el-Raspberry-Pi-entra.html (Consultado el 26 de Marzo del 2013)

- [6] Raspberry Pi. Librería WiringPi. Disponible en: https://projects.drogon.net/raspberry-pi/wiringpi/the-gpio-utility/ (Consultado el 20 de Marzo del 2013)

- [7] Tamaño de RaspberryPi modelo B Disponible en: http://nishati.org/hardware (Consultado el 26 de Marzo del 2013)

- [8] Eben Upton , Raspberry Pi User Guide, Wiley

- [9] Raspberry Pi. Características técnicas Disponible en: http://www.todohtpc.com/analisis/cajas-y-equipos-htpc/raspberry-pi.html (Consultado el 19 Marzo del 2013).[10] José Antonio Caballar Falcón. WIFI, Lo que se necesita conocer , Rc Libros

- [11]Servidores Web, función. Disponible en: http://culturacion.com/2011/12/como-convertir-un-pc-en-servidor-web/ (consultado el 17 Marzo 2013)

- [12]Servidores Web. Disponible en:

http://www.misrespuestas.com/que-es-un-servidor-web.html

(Consultado el 17 Marzo 2013)

- [13] Apache2. Disponible en:

  http://doc.ubuntu-es.org/HTTPD_Servidor_web_Apache2 (consultado el 18 de Marzo del 2013)

- [14] PHP5. Disponible en:

  http://www.desarrolloweb.com/articulos/1696.php (consultado el 19 marzo 2013)

ANEXOS

ANEXO A

Descargando el sistema operativo para la Raspberry Pi

En la configuración del Raspberry Pi se utilizara el sistema operativo Raspbian, el mismo que puede ser descargado desde la web, existen más sistemas operativos compatibles para Raspberry Pi, en este caso instalaremos la distribución RaspbianWheezy.

- Ingresar a la páginawww.rasberrypi.org
- Ir al menú descargas http://www.raspberrypi.org/downloads

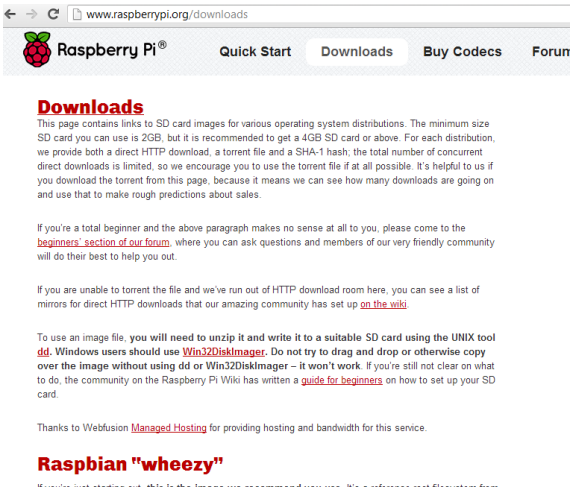


Figura A1 Página de Rapberry Pi org opción descargas

En el menú descargas se procederá a descargar el programa Win32DiskImager que nos servirá para grabar el sistema operativo pre instalado en la memoria SD.

- En descargas se descargara la versión más actualizada de Raspbian.

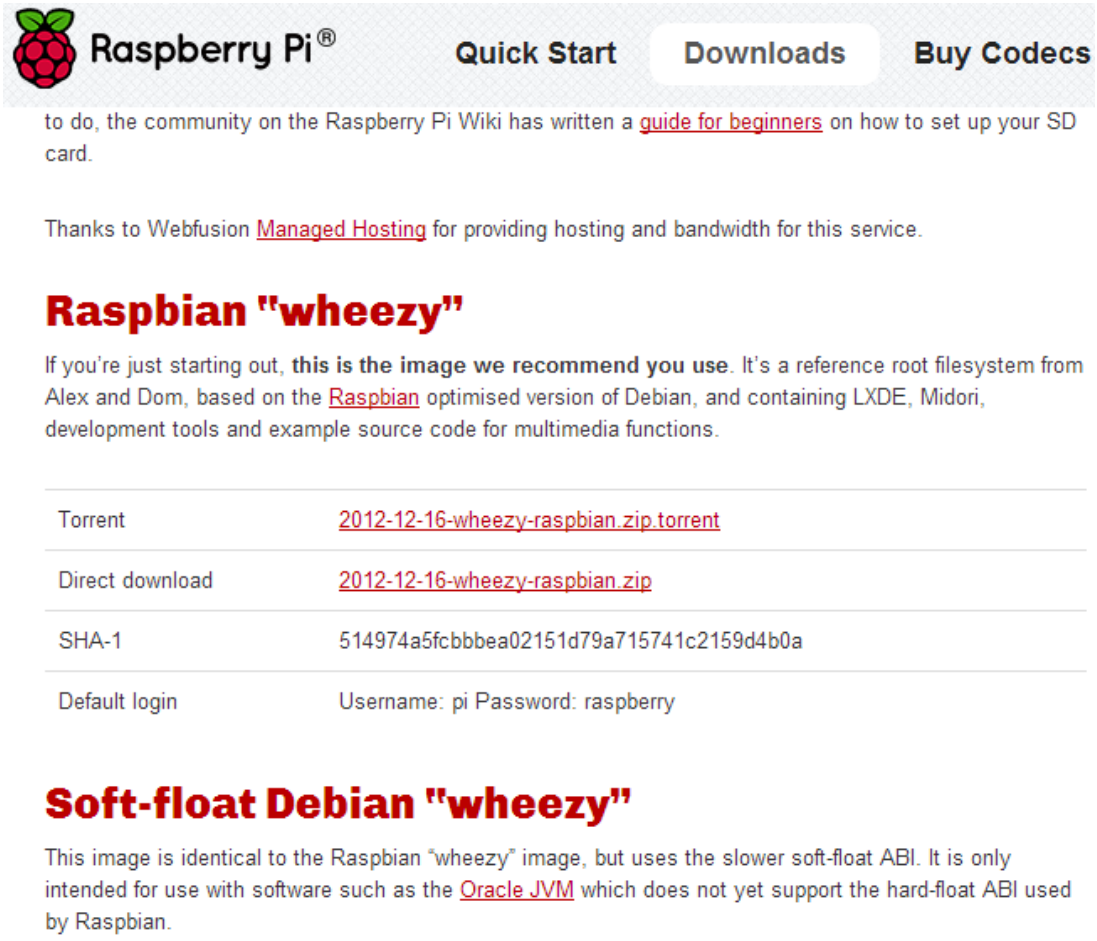
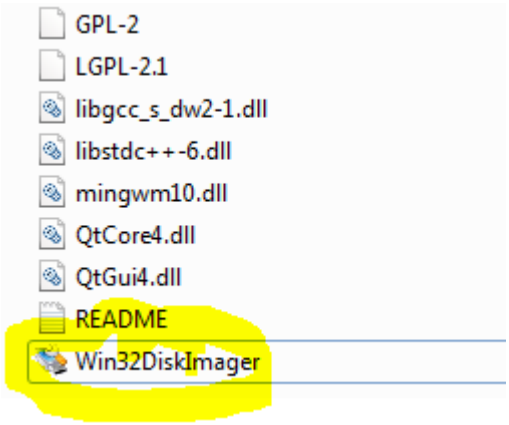
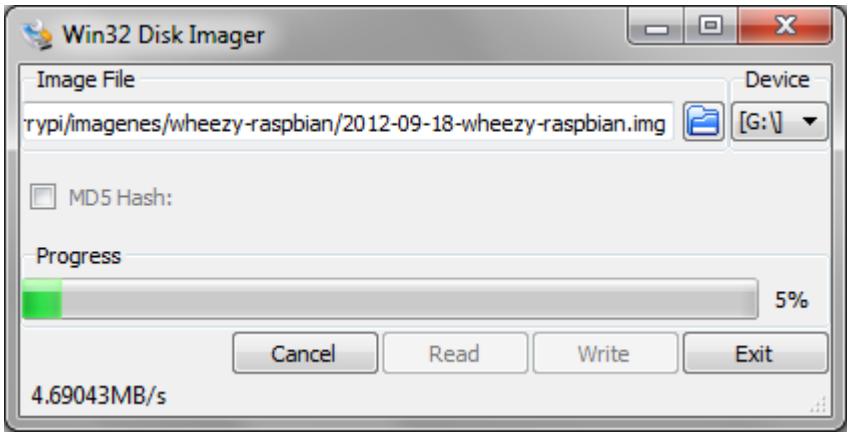
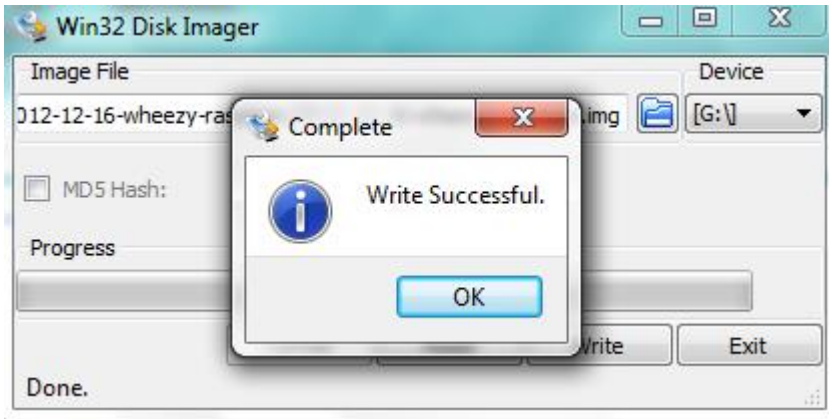


Figura A2 Opción descarga de sistema operativo

- Una vez descargado, descomprimir la imagen y "quemar" en la memoria SD la versión de raspbian usando el programa win32DiskImage.







FiguraA3 Descargando el sistema operatico en la memoria usando DiskImager

Una vez realizado este proceso ya se puede utilizar la memoria en el dispositivo Raspberry pi.

- Una vez colocada la imagen en la tarjeta usando Win32DiskImager, podemos insertar la tarjeta en el Raspberry.
- Se conecta el teclado, mouse, el HDMI y el cable de alimentación a sus respectivos puertos.

ANEXO B

Encendido desde el Raspberry PI

- Configuración en el primer “booteo”, configuración del sistema operativo.

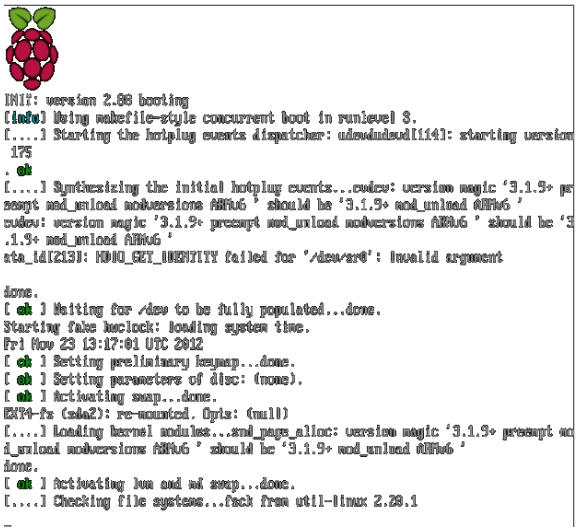


Figura B1 Pagina principal al encender la Raspberry Pi.

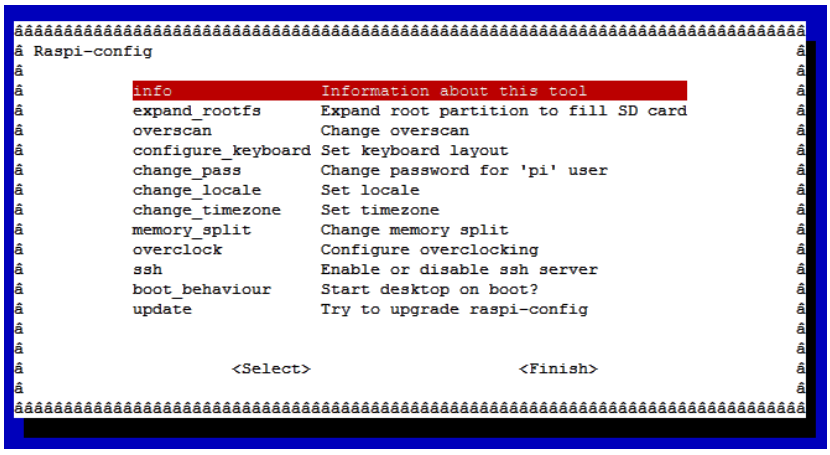


Figura B2 Opciones de configuración de Raspberry Pi.

Entre las opciones tenemos:

info - Información sobre qué es esta herramienta.

expand_rootfs - Para que el sistema de ficheros de nuestra distribución de Linux utilice todo el espacio que permita nuestra tarjeta SD.

configure_keyboard - Para elegir la configuración de teclado (en este caso, uno "Generico de 105 teclas", que aparece como "Generic 105 key") y su distribución (se propone "English (UK)", pero deberíamos escoger "Other" y luego "Spanish"; se nos preguntará ciertas variantes, como el teclado de Apple Macintosh, o el catalán, pero generalmente bastará con dejar simplemente "Spanish").

Finalmente se nos preguntará también cómo queremos que se comporte la tecla AltGr (se puede dejar con su valor por defecto) y si queremos que la combinación Ctl+Alt+Borrar pueda cerrar el entorno gráfico, algo que a nosotros por lo general nos gusta conservar, por si el entorno gráfico llegase a bloquearse, algo que es muy poco frecuente.

change_locale -Para indicar la configuración regional. Lo razonable en España es escoger "es_ES.UTF-8". Aun así, tenemos que asumir que lo habitual es que

los mensajes de error de consola, e incluso los menús en modo gráfico, no estén traducidos al castellano y nos aparezcan en inglés.

change_timezone - Para indicar la zona horaria, que en nuestro caso podría ser Europe / Madrid, aunque no es algo crítico, porque el Pi no tiene reloj interno, y la fecha y hora se perderá de una sesión de trabajo a la siguiente.

memory_split - Para elegir el reparto entre memoria principal y memoria de video. Como mi uso no va a ser intensivo en gráficos, prefiero dejar más memoria principal (224+32).

ssh - Para habilitar o bloquear el servidor de ssh, por si quisiéramos conectar al Pi desde la red.

boot_behaviour - Para indicar el comportamiento de arranque: si queremos que comience con entorno gráfico o con una consola en modo texto. A mí me gusta que arranque en consola, y, si quiero usar el entorno gráfico, sólo tengo que teclear "startx".

update - Para intentar actualizar la utilidad raspi-config. En mi caso, no tengo el Pi conectado a Internet, así que no me he molestado en probarlo.

- Se pueden configurar varias opciones, en este caso vamos a expandir la memoria, el fin es expandir la partición root a la capacidad de la memoria.

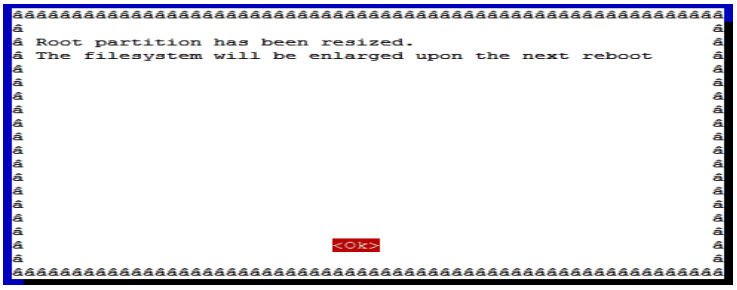
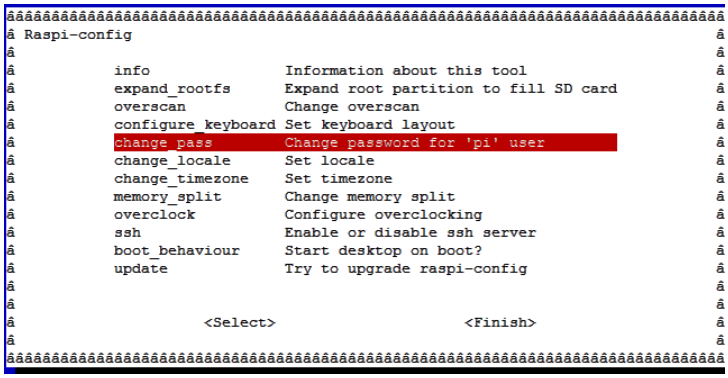


Figura B3 Expandir memoria para mayor capacidad en root

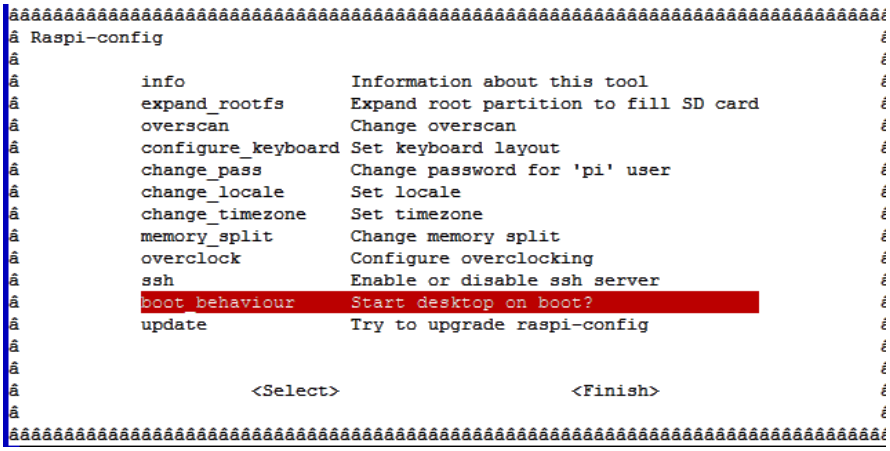
- Además de cambiar el password.



FiguraB4 Opción Change_pass

Figura B5 Confirmación para introducir nueva contraseña

- También se puede configurar para que inicie en modo gráfico.



FiguraB6 Opción en modo Gráfico.

FiguraB7Confirmación para arrancar en modo Gráfico

- Luego finalizamos para iniciar el sistema operativo, se requiere reiniciar.

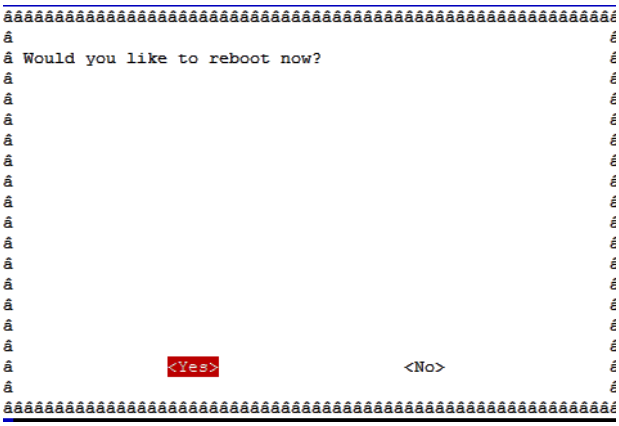


Figura B8 Reiniciar S.O luego de configurar.

Pantalla inicial de instalación del sistema operativo.

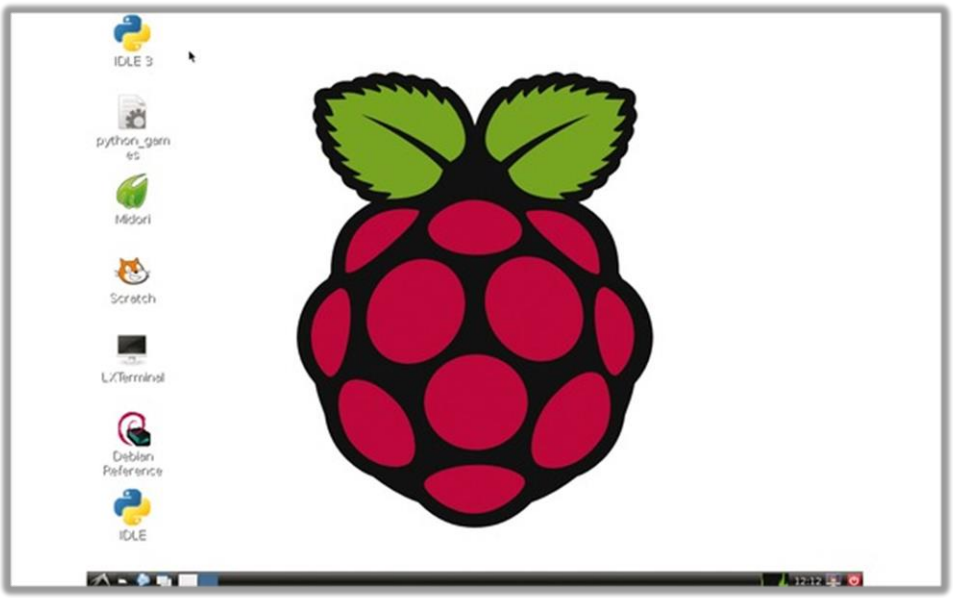


Figura B9 Pantalla principal de S.O en Raspberry pi

ANEXO C

Instalando un programa

Se recomienda actualizar La base local de repositorio de Raspberry PI antes de instalar cualquier programa, en este caso instalaremos un programa que nos ayudara a tener acceso remoto al dispositivo.

Actualizar la base local de repos en el raspberry pi, esto lo hacemos con el comando: pi@raspberrypi ~ $ sudo apt-getupdate

Nota: Se recomienda realizar esto para actualizar y evitar errores

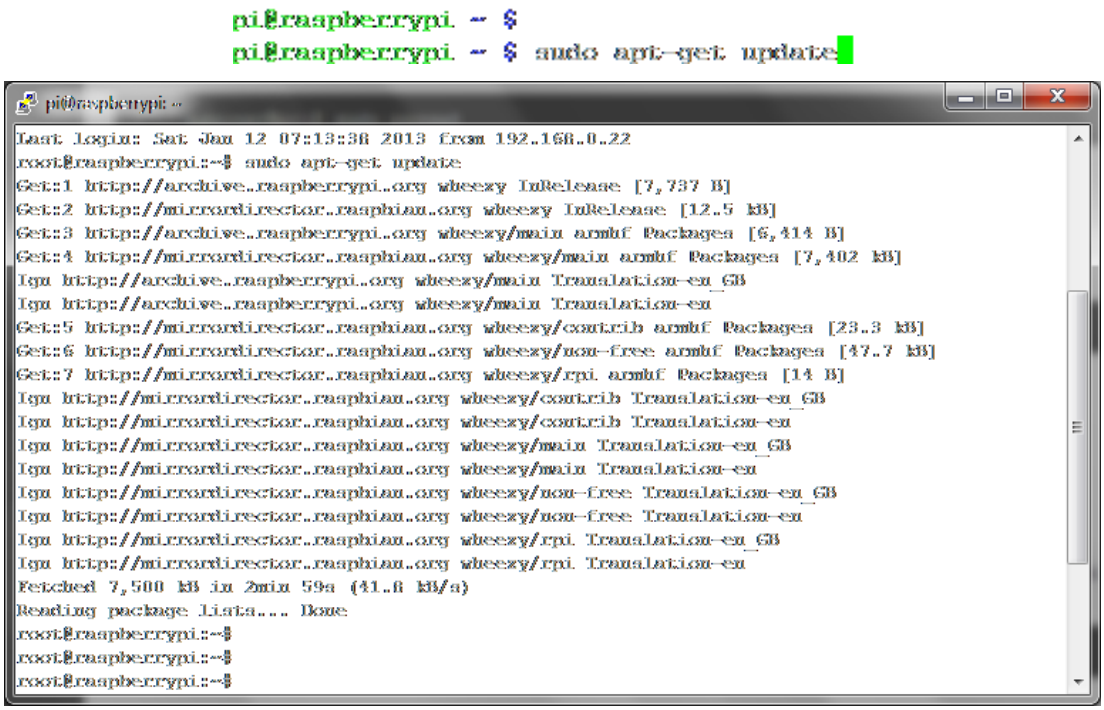


Figura C1 Comando de Actualización

Instalando VNC

Ante todo, VNC *Virtual Network Computing* (Computación Virtual en Red). Es un programa de software libre basado en una estructura cliente-servidor el cual permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente.

Ejecutamos el comando, sudo apt-getinstalltightvncserver , con esto instalamos el paquete de servidor VNC

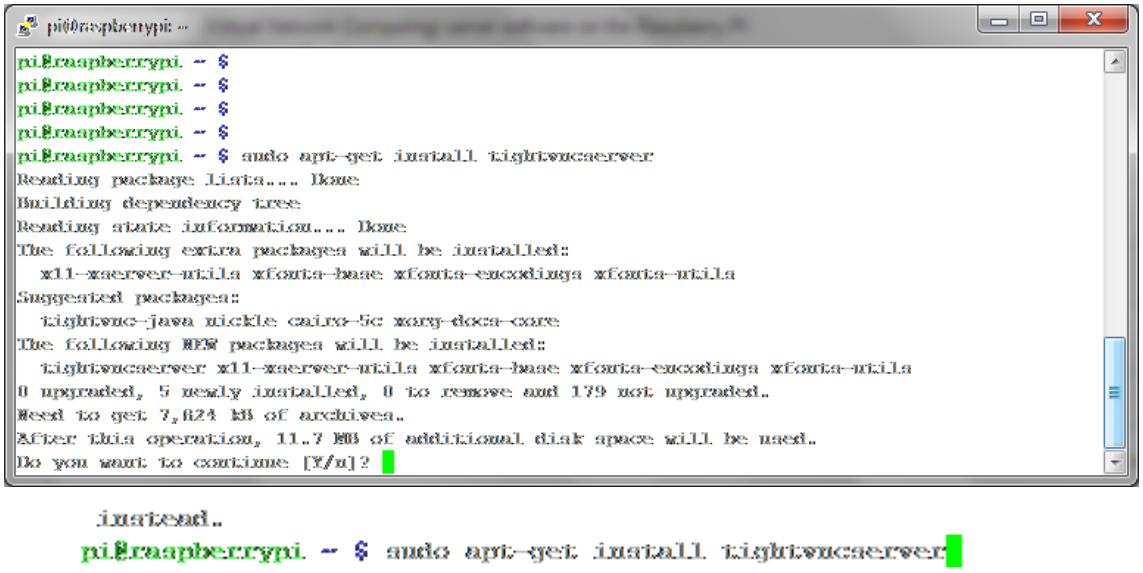


Figura C2 Comando instalación de VNC

Ejecutamos el servidor VNC el cual nos pedirá la contraseña para acceder al usuario pi desde otra pc de la red. Opcionalmente pide la contraseña para solo abrir una conexión de vista.

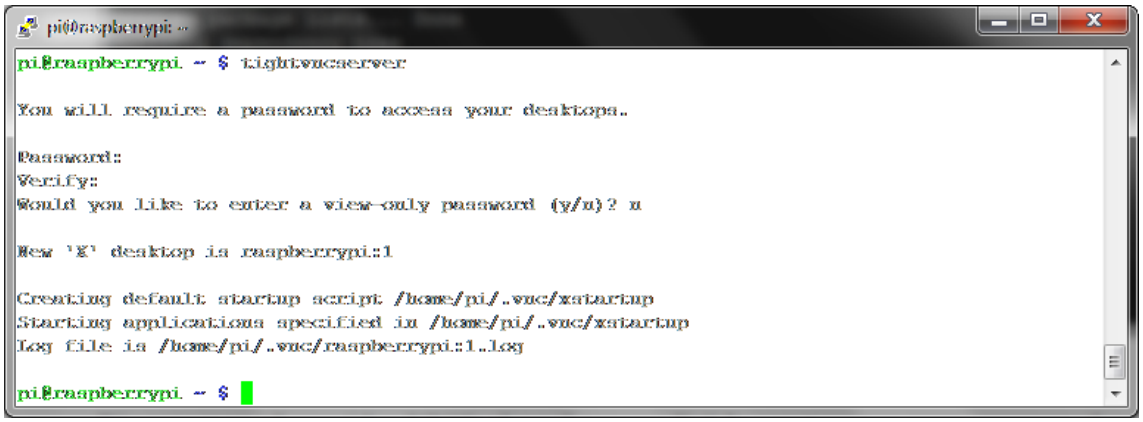


Figura C3 Requerimientos para realizar coneción vnc

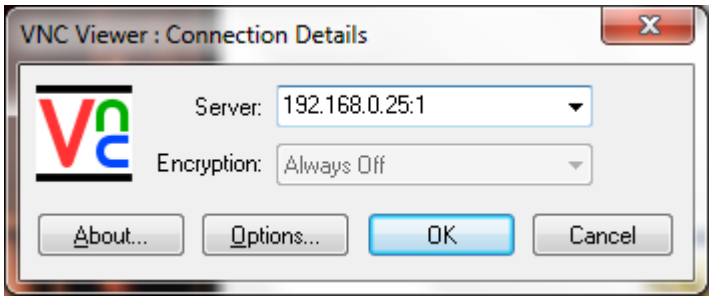
Una vez realizado esto, usted puede empezar a establecer conexión con el servidor VNC. Como podrá ver en la imagen de arriba le indicaque usted puede abrir una sesión VNC en el display raspberrypi:1, lo que también puede ser ip_raspberrypi:1. Ejemplo, 192.168.0.25:1

Nota: Usted debe siempre ejecutar el comando tighvncserver cada vez que reinicia el raspberrypi. Opcionalmente usted puede usar el equivalente de comando tighvncserver por vncserver.

Conectarse remotamente, como cliente desde una PC

Para lograr acceder remotamente al dispositivo se debe instalar el “Cliente” de VNC para lograr visualizar, descargar el cliente de VNC. http://www.realvnc.com/download/viewer/

Conectarnos remotamente nos ayuda para visualizar la interfaz del sistema operativo Raspbian, antes de realizar esto se debe conocer la dirección IP del Raspberry PI.



FiguraC4Interfaz VNC

De esta forma ya tenemos acceso remoto a nuestro Raspberry desde el programa VNC