



UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

www.bdigital.ula.ve

DISEÑO E IMPLMENTACIÓN DE UNA PANTALLA
INTELIGENTE USANDO UN PANEL LCD, UNA TARJETA
RASPBERRY PI 3 Y UN SERVIDOR WEB

Br. José Antonio Zambrano Molina

Mérida, Julio, 2017

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

www.bdigital.ula.ve

DISEÑO E IMPLEMENTACIÓN DE UNA PANTALLA
INTELIGENTE USANDO UN PANEL LCD, UNA TARJETA
RASPBERRY PI 3 Y UN SERVIDOR WEB

Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero
Electricista

Br. José Antonio Zambrano Molina

Tutor: Prof. Luis Ramón Araujo

Mérida, Julio, 2017

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

**DISEÑO E IMPLEMENTACIÓN DE UNA
PANTALLA INTELIGENTE USANDO UN PANEL
LCD, UNA TARJETA RASPBERRY PI 3 Y UN
SERVIDOR WEB**

Br. José Antonio Zambrano Molina

Trabajo de Grado presentado en cumplimiento parcial de los requisitos exigidos para optar al título de Ingeniero Electricista, aprobado en nombre de la Universidad de Los Andes por el siguiente Jurado.

Prof. Emiro Jose Ibarra
C.I 17.027.152

Prof. Arlex Joel Caliz
C.I 15.685.941

Prof. Luis Ramon Araujo
C.I 10.711.246

La idea detrás de los computadores digitales puede explicarse diciendo que estas máquinas están destinadas a llevar a cabo cualquier operación que pueda ser realizado por un equipo humano.

ALAN MATHISON TURING

A mis padres.

AGRADECIMIENTOS

A mis padres **Clory** e **Yván**, que son las bases de mi vida, mi ejemplo a seguir y los principales merecedores de este trabajo.

Gracias a la Ilustre Universidad de Los Andes, por el esfuerzo que hacen generación tras generación, brindando formación humana y profesional para contribuir con el desarrollo del país.

A la Escuela de Ingeniería Eléctrica, que por sus excelentes profesionales sigue con un nivel de exigencia alto y un estándar de enseñanza único.

A mi tutor y amigo Luis Ramón Araujo, que con su mejor disposición me ayudó en la formación como ingeniero a lo largo de la carrera y mas en esta última etapa, siempre con un ingenio auténtico y sencillez única para resolver un problema.

Al Ingeniero Erson Márquez, por su oportuna colaboración en la etapa final de este proyecto, facilitando herramientas y material para la construcción de la carcasa definitiva del prototipo.

José Antonio Zambrano Molina

José Antonio Zambrano Molina. Diseño e Implementación de una Pantalla Inteligente Usando un Panel LCD, una Tarjeta Raspberry Pi 3 y un Servidor Web. Universidad de Los Andes. Tutor: Prof. Luis Ramón Araujo. Julio, 2017.

Resumen

La era digital es una realidad y es por ello que día a día el contenido informativo está inmerso en la cotidianidad, donde siempre se busca estar actualizados y al tanto de lo que sucede a nuestro alrededor, bien sea que estemos en nuestro tiempo libre o en horas de trabajo. Lo que si es cierto, es que, sin darnos cuenta en cada momento el contenido visual que percibimos, contiene información importante, alguna publicidad, un horario de vuelo e inclusive hasta el reporte climático, haciendo que un dispositivo electrónico sea el encargado de hacer llegar este contenido, particularmente a través una pantalla informativa. Es usual que una panel publicitario sea muy complejo y de alto costo en cuanto a su parte física, ya que consta de un televisor, una computadora y un software que permite reproducir alguna información. Este trabajo está enfocado en la implementación de un dispositivo similar pero mas sencillo y de menor costo, permitiendo que siga siendo útil para el usuario y cómodo para el operador. Siendo muy innovador la simplicidad del hardware lo cual hace que su costo sea bajo, sencillo y eficaz. Mas aún el uso de herramientas libres que dispongan de un software, permite que sea accesible a contribuir con su desarrollo y a su vez a que mas personas puedan aportar mejoras. Se busca mostrar información al usuario a través de una pantalla de cristal liquido o panel LCD a color, que esté controlada por una tarjeta Raspberry Pi, la cual mediante sus facilidades de conexión inalámbrica, permitirá controlarla a distancia, haciendo uso de una base de datos libre MySQL, gestionando el contenido a reproducir mediante un servidor web y pudiendo controlarlo de forma remota desde cualquier lugar.

Descriptor: Raspberry Pi, Base de Datos, Servidor Web, Software Libre, Herramientas de Programación, Prototipo Funcional.

ÍNDICE GENERAL

APROBACIÓN	ii
DEDICATORIA	iii
AGRADECIMIENTOS	iv
RESUMEN	v
INTRODUCCIÓN	1
Capítulo	pp.
1. EL PROBLEMA	4
1.1 Planteamiento del Problema	4
1.2 Objetivo General	5
1.3 Objetivos Específicos	5
1.4 Justificación	6
1.5 Marco Metodológico	6
1.6 Alcance	7
1.7 Limitaciones	7
2. MARCO TEÓRICO	9
2.1 Sistema Físico	9
2.1.1 Raspberry Pi	10
2.1.2 Pantalla LCD	13
2.1.3 Interfaz Electrónica	15
2.1.4 Reloj de Tiempo Real	16
2.2 Sistema Lógico	17
2.2.1 Sistema Operativo	18
2.2.2 Herramientas de Programación	19
2.2.2.1 SQL	20
2.2.2.2 PHP	21
2.2.2.3 HTML	23
2.2.2.4 Lenguaje C	26

2.2.3	Base de Datos	28
2.2.3.1	MySQL	29
2.2.4	Servidor Web	29
2.2.4.1	Apache	30
2.2.5	Reproductor Multimedia	31
2.2.5.1	OSMC y VLC	31
3.	DISEÑO	32
3.1	Prototipo Físico	32
3.2	Programación	37
3.2.1	Base de Datos MySQL	39
3.2.2	Servidor Web Apache	41
3.2.3	Programa de Reproducción en Lenguaje C	49
4.	ANÁLISIS DE RESULTADOS	56
4.1	Análisis de Resultados	56
4.2	Prototipo Final	58
	CONCLUSIONES	63
	RECOMENDACIONES	65
	BIBLIOGRAFÍA	65

ÍNDICE DE TABLAS

Tabla	pp.
2.1 Ejemplo de Sentencias SQL	21
2.2 Ejemplo de Código PHP	22
2.3 Archivo Prueba.html	26
2.4 Archivo de Estilos Styles.css	27
2.5 Ejemplo de Código en C	28
3.1 Acceso a MySQL por Línea de Comandos	40
3.2 Conexión a la Base de Datos en PHP	41
3.3 Creación de Base de Datos y Tabla de Publicidad en PHP	42
3.4 Encabezado HTML	43
3.5 Formulario de Ingreso	44
3.6 Comprobación del Formulario de Ingreso en PHP	45
3.7 Formulario de Ingreso de Publicidad en HTML	47
3.8 Ejecución del Formulario de Ingreso de Publicidad en PHP	48
3.9 Librerías de C	50
3.10 Bucle Principal	50
3.11 Conexión a la Base de Datos	51
3.12 Sentencias para Consulta del RTC	52
3.13 Código Para Reproducción en Caso de Falla Eléctrica	52
3.14 Consulta a la Base de Datos	53
3.15 Llenado del Vector de 36 posiciones, Gestión de Filas Vacías y Reproduccion de 60 Veces	54
3.16 Función de Reproducción de Contenido	55
3.17 Compilar program_repro.c	55
3.18 Ejecución program_repro.c	55

ÍNDICE DE FIGURAS

Figura	pp.
2.1 Raspberry Pi	11
2.2 Pantalla LCD	14
2.3 Partes de un Panel LCD	15
3.1 Principales Componentes	33
3.2 Primera Carcasa	34
3.3 Pines de la Raspberry Pi 3	35
3.4 Controlador PCB800099-RTD2662	36
3.5 Modulo Arduino DS3231	37
3.6 Esquema del Sistema	39
3.7 Vista del Acceso a la Base de Datos desde Terminal Linux	40
3.8 Vista del Ingreso al Sistema	43
3.9 Vista del Menú Principal	45
3.10 Vista del Menú de Clientes	46
3.11 Vista del Formulario de Publicidad	46
3.12 Lista de Publicidades	49
4.1 Pantalla Convencional	59
4.2 Carcasa del Prototipo de Pantalla Inteligente Final 1	61
4.3 Carcasa del Prototipo de Pantalla Inteligente Final 2	62

INTRODUCCIÓN

Las pantallas comerciales con fines publicitarios o informativos son bastante costosas, usualmente conformadas por un televisor, una computadora, cableado y una carcasa que albergue todo lo anteriormente descrito. Por lo que implica que su tamaño suele ser grande y espacioso, haciendo que visualmente no sea tan estético. A su vez puede usar un software que posee licencia de pago, en donde el operador suele tener acceso de manera presencial al computador de dicha pantalla, y así cargar la información requerida para su posterior reproducción, lo cual no refleja ser tarea sencilla, sobre todo para el operador y mucho menos de bajo costo, ya que se incurriría con gastos de manutención para el personal encargado. Esto representa una gran desventaja en el análisis económico de un proyecto de envergadura significativa, pudiendo ser mejorado significativamente haciendo mejor uso de otros componentes electrónicos mucho mas eficientes.

Con el avance comercial que existe actualmente, es bastante común tener acceso a componentes electrónicos, los cuales permitan buscar las piezas de ensamble para algún dispositivo y de esta forma no tener que optar por adquirir el producto final. Esto permite mayor flexibilidad en la ejecución de proyectos y a su vez un eficiente uso de recursos. Con lo que se obtendrá un mejor producto final realmente adaptado a las necesidades del usuario. Permitiendo dar vida a prototipos realmente útiles y de gran eficacia.

Además este proyecto busca destacar la importancia del uso de software libre, así como el uso de componentes electrónicos accesibles, de gran poder y que a su vez representen un ahorro en el costo del producto final, para incentivar a pequeños emprendedores a contribuir con el desarrollo del país. Teniendo una visión en el desarrollo de proyectos mas adaptada al futuro y a la renovación de la tecnología.

Para darle solución al gran tamaño de los paneles informativos comúnmente usados, la parte física de este trabajo de grado se simplifica en un panel LCD de 15 pulgadas y una tarjeta madre Raspberry Pi 3, que serán alojados en una carcasa. Por su parte el software se desarrolla en dos etapas: la base de datos con un servidor web y el programa de reproducción de contenido. Estas partes interactúan entre si de manera simultanea y permiten darle vida a este proyecto, en el cual será muy eficaz la manera como podemos hacerle llegar la información de forma remota, usando las ventajas de la conexión inalámbrica de la tarjeta madre y simplificando así el producto final, en donde la única conexión alámbrica será la alimentación para encender los dispositivos.

Sera muy versátil la forma como se logra abordar destrezas en programación, ya que no es usual que en el transcurso de la carrera de Ingeniería Eléctrica se logren obtener todas las herramientas necesarias para pensar como un desarrollador de software, pero si contamos con lo esencial para poder darle solución a esta etapa del proyecto. Donde haremos uso de sentencias SQL para la base de datos, la programación del servidor web será hecha en HTML y PHP y el software para la reproducción de toda la información será programado en lenguaje C. Tener conocimiento en distintas áreas de la carrera será fundamental para complementar la formación de un ingeniero y mas aun cuando la tecnología nos aborda muy rápido, donde la tendencia a herramientas de fácil acceso, representa simplificar muchos pasos en la ejecución que cualquier proyecto enfocado en microelectrónica y software, como en este caso. Es por ello que el enfoque al uso de herramientas de software libre no solo permitirán a la contribución de diferentes usuarios a complementar el proyecto, sino también, a simplificar su puesta en marcha.

Se obtendrá un producto final muy ajustado a las necesidades reales, donde se podrá almacenar datos relacionados al uso de la pantalla para fines publicitarios o informativos, donde se delimitará un ciclo de servicio de la pantalla, la cantidad de contenido que se mostrará y su funcionamiento de forma automática en las jornadas para las cuales de delimite su uso. A su vez se considera que es un proyecto muy realista, accesible e

innovador. Por sus facilidades de construcción, programación y accesibilidad.

El uso de la tarjeta Raspberry Pi, representa la optimización de gran parte del proyecto, por su versatilidad de configuración, su tamaño ideal y la posibilidad de hacer uso de cualquier herramienta de programación que se haya adquirido hasta el momento. Siendo el cerebro del prototipo, en el cual se busca dar a conocer una versión muy útil y didáctica de sus aplicaciones.

A fin de dar cumplimiento a los objetivos planteados, el presente trabajo se organizó en cuatro (4) capítulos. En el capítulo 1 se describe el planteamiento del problema, la justificación, los objetivos propuestos, el alcance y las limitaciones del estudio, así como también la metodología empleada para el desarrollo de este trabajo.

En el capítulo 2 está contenido el marco teórico en donde se encuentran las definiciones de componentes electrónicos (hardware) y herramientas computacionales (software) implementados en el proyecto.

En el capítulo 3 se describe el diseño de la pantalla, en donde se expone el ensamblaje de la parte física, el uso de la tarjeta Raspberry Pi, el driver de comunicación entre la tarjeta y la pantalla, así como el funcionamiento y uso de la base de datos, el servidor web y la programación del software de reproducción.

En el capítulo 4 se desarrollan los resultados y análisis, junto a la implementación del prototipo funcional.

Finalmente, se establecen las conclusiones y a partir de los resultados obtenidos en la implementación se hacen las recomendaciones. Por su parte, con el fin de hacer uso de materiales reciclados, en el prototipo inicial, se utilizará una pantalla de computadora portátil y una carcasa plástica para alojar cada componente. Posteriormente se diseñará una carcasa más estética y con un mejor acabado para la presentación final del prototipo funcional.

CAPÍTULO 1

EL PROBLEMA

Siendo una iniciativa de emprendimiento, se quiere buscar solución a la obtención de productos culminados, los cuales no se ajustan a las facilidades de comunicación entre el usuario y operador, que quizás resulten muy poco accesibles y no cubran por completo las necesidades. Es por ello que teniendo acceso a determinados componentes, pueden surgir proyectos mas completos, ajustados y rentables. En este capítulo se planteara el problema que lleva al desarrollo de este proyecto, los objetivos generales y específicos a ser cumplidos en la construcción de este junto a la metodología empleada, posibles alcances y limitaciones presentadas y consideradas.

1.1 Planteamiento del Problema

Debido al requerimiento de nuevas obras, tanto de carácter comercial, como de servicios públicos o instituciones privadas de cualquier tipo, suele de ser de mucha utilidad mantener al usuario informado con contenido visual referente a la obra en cuestión, el cual suministre lo que sea necesario para las empresas prestadoras de servicios; es por ello que los contratistas recurren al mercado internacional encontrándose con pantallas informativas de altos costos por no ser fabricados en el país, y a su vez los componentes necesarios para este producto son adquiridos mediante importación. Al adquirir una de estas pantallas, no solo limita el hecho de la inversión económica, mas aun, el principal problema resulta ser que no se ajustan a las necesidades reales del beneficiario final y el operador, esto implica que también sea engorrosa la modificación inmediata de contenido que se desee mostrar.

El desarrollo y avance de la tecnología en el área electrónica y microprocesadores, no solo presenta una solución a un problema como lo encontramos acá, sino que permite darle solución a diferentes problemas con tecnología más accesible para muchos venezolanos, como lo son los microprocesadores, que en ocasiones según el diseño e implementación de circuitos controlados por estos se pueden desarrollar módulos adaptados a las condiciones locales.

1.2 Objetivo General

Proporcionar contenido visual de cualquier índole, a través de una pantalla de cristal líquido con un diseño simplificado y de fácil instalación. El cual funcione de forma predefinida, automática y que mediante su comunicación remota, sea de cómodo acceso para el usuario y el operador. En donde se logre controlar la cantidad de información, el tiempo adecuado para gestionarla y la fácil modificación. Todo lo anterior descrito almacenado mediante un servidor web en una base de datos, donde a su vez se logre cargar información del contenido en forma remota y donde se pueda almacenar información inherente a clientes relacionados con el contenido que se desee mostrar.

1.3 Objetivos Específicos

- Desarrollar un prototipo del sistema físico que alojará una pantalla LCD, una tarjeta Raspberry Pi 3 y los demás componentes necesarios para la implementación.
- Estudiar el funcionamiento de herramientas de programación en la Raspberry Pi 3, para la creación de un algoritmo para el inicio y ejecución de reproducción de contenidos de una manera sistematizada.
- Escoger una base de datos libre, que facilite el almacenamiento de información y proporcionen fácil acceso a modificaciones.

- Implementar el uso de un servidor web para lograr un acceso simplificado y cómodo, donde se pueda almacenar contenido de interés a la base de datos.
- Integrar el programa de reproducción de contenido, la base de datos y el servidor web.

1.4 Justificación

Con la finalidad de contribuir en el desarrollo de proyectos de gran alcance y con proyección comercial, se busca dar soluciones a problemas en el área de electrónica e informática, en cuanto a la transmisión de información digital que se enfoca en publicidad empresarial o información general de diferentes índoles. Para ello se busca poder disminuir costos de productos ya terminados y sustituirlo por un prototipo de diseño y desarrollo nacional, incentivando la producción de pequeñas empresas de emprendedores y trabajando de manera optimizada con los requerimientos reales de la demanda, que puede estar enfocada a clínicas, aeropuertos, centros comerciales, entre otros.

Esta propuesta viene de la mano con el uso e implementación de una tarjeta Raspberry Pi 3 y permitirá el desarrollo de un programa en el cual podamos establecer el control y envío de información y datos a la pantalla LCD. A su vez mediante las facilidades de conexión inalámbrica que la Raspberry Pi 3 nos proporciona, se busca también mediante un servidor web, poder controlar de forma remota la información que se desee transmitir.

1.5 Marco Metodológico

La metodología para realización de este trabajo, consta de la investigación bibliográfica acerca del proceso de programación de la tarjeta Raspberry Pi 3, para la ejecución de funciones requeridas. Además de la investigación acerca de algoritmos necesarios para llevar a cabo las tareas propuestas.

Seguido de esto, la implementación de un prototipo de hardware en donde se pueda tener alojado la pantalla LCD con su tarjeta Raspberry y la interfaz entre ambas. El cual sea simplificado y eficiente.

Posteriormente, la implementación en algún lenguaje de programación para la tarjeta Raspberry Pi 3, utilizando las herramientas software libre, los algoritmos que permitan verificar el funcionamiento de cada uno de los componentes utilizados, utilizando simulaciones en computadora así como en el prototipo. Por otra parte gestionar de manera eficaz la información que sea necesaria para el control del prototipo, que mediante una base de datos y un servidor web se logre integrar en conjunto al software del sistema.

Finalmente unir todos los componentes físicos y de sistema para realizar las pruebas necesarias para validar el prototipo.

1.6 Alcance

Este proyecto contempla el desarrollo de un prototipo alternativo a opciones comerciales y mejor adecuación a los requerimientos reales del beneficiario. Donde se logre exponer la importancia del desarrollo de herramientas de software libre, así como su gran utilidad y envergadura a la hora de proyectos comerciales, los cuales se puedan resolver en el futuro próximo como ingenieros.

1.7 Limitaciones

- Adquirir una pantalla LCD de 32 o 40 Pulgadas, para hacer el proyecto lo mas exacto a un panel real, incrementa sustancialmente el precio del prototipo, es por ello que se va a utilizar una pantalla LCD de 15 pulgadas, de un computador portátil dañado. Lo cual nos proporcionará un diseño mas pequeño pero con las mismas funcionalidades.
- El acceso limitado a componentes electrónicos mediante importación, hace que sur-

ja una barrera económica para incorporarle mas periféricos a la tarjeta *Raspberry Pi*, como por ejemplo una cámara, la cual serviría de complemento al proyecto en cuestión.

- Cada componente o producto terminado, suele ser un artículo de importación, es por ello que se hace difícil acceder a los mismos por las regulaciones de moneda extranjera en el país. Haciendo que su precio sea muy elevado y de complejo acceso.

www.bdigital.ula.ve

CAPÍTULO 2

MARCO TEÓRICO

Más que un hecho, es una realidad el gran avance tecnológico en el que día a día se encuentra involucrada nuestra sociedad y en la mayoría de los casos no se le puede seguir ritmo. Es por ello que usualmente se desconoce el potencial de componentes electrónicos y herramientas de desarrollo para proyectos de gran alcance, en donde a su vez, se consigue una cantidad muy grande información referente a los mismos, y es posible darle vida a casi cualquier idea. Se busca exponer todo lo utilizado para la elaboración de este proyecto, y poder resaltar la gran utilidad de dispositivos de gama alta como la Raspberry Pi 3, así como de herramientas de desarrollo libres que con una comunidad enorme, logran aportar conocimiento para el surgimiento de cualquier proyecto. El presente capítulo inicia con una breve descripción de los principales componentes y herramientas usados en el proyecto, buscando que de manera progresiva, sea mas cómoda la integración de cada una de las partes y poder apreciar el potencial de esta idea.

2.1 Sistema Físico

Si bien no tiene una traducción exacta al castellano, un **sistema físico** se relaciona directamente con la palabra *hardware*, en informática se refiere a las partes tangibles de un sistema informático como sus componentes eléctricos, electrónicos, electromecánicos y mecánicos [1]. Así como también periféricos de todo tipo y cualquier otro elemento físico involucrado componen el *hardware*. El término es propio del idioma inglés, su traducción al español no tiene un significado acorde, por tal motivo se lo ha adoptado tal cual es

y suena. La *Real Academia Española* lo define como: Un conjunto de los componentes que integran la parte material de una computadora [2]. El término, aunque sea lo más común, no solamente se aplica a las computadoras, también es utilizado en otras áreas de la vida diaria y la tecnología y también se considera al *hardware* como uno de tres pilares fundamentales en diseño electrónico.

La historia del *hardware* de computador se puede clasificar en cuatro generaciones, cada una caracterizada por un cambio tecnológico de importancia. Una primera delimitación podría hacerse entre *hardware* principal (como el estrictamente necesario para el funcionamiento normal del equipo) y el complementario (como el que realiza funciones específicas) [3].

Un sistema informático se compone de un **CPU** o Unidad Central de Procesamiento (*Central Processing Unit*), encargada de procesar los datos, así como varios periféricos de entrada, que permiten el ingreso de la información; y uno o varios periféricos de salida, que posibilitan dar salida en forma visual o auditiva, a los datos procesados.

El proyecto para asemejarse más a una pantalla real, va a requerir la utilización de una computadora de placa reducida como la *Raspberry Pi*, una *pantalla LCD* y una *interfaz electrónica* encargada de la comunicación entre el computador de placa única y el panel LCD.

2.1.1 Raspberry Pi

Se puede hablar de un computador de placa reducida, computador de placa única o computador de placa simple de bajo costo, cuando nos queremos referir a la **Raspberry Pi**. Este proyecto fue desarrollado en el Reino Unido por la *Fundación Raspberry Pi*, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas. Es un producto con propiedad registrada pero de uso libre. De esa forma mantienen el control de la plataforma pero permitiendo su uso libre tanto a nivel educativo como particular [4] [5].

En cuanto al *software* sí es de código abierto (*Open Source*), siendo su sistema opera-

tivo oficial una versión adaptada de *Debian (GNU/Linux)*, denominada *Raspbian*, aunque permite usar otros sistemas operativos, incluido una versión de *Windows 10*. En todas sus versiones incluye un procesador Broadcom, una memoria RAM, una GPU o unidad de procesamiento gráfico *Graphics Processor Unit*, puertos USB, HDMI, Ethernet , 40 pines GPIO o entradas y salidas multipropósito (*General Purpose Output and Input*) y un conector para cámara. Ninguna de sus ediciones incluye memoria, siendo esta en su primera versión una tarjeta SD y en ediciones posteriores una tarjeta de almacenamiento MicroSD [6].

La fundación da soporte para las descargas de las distribuciones ARM, Raspbian, 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*. Otros lenguajes también soportados son *Tiny BASIC*, *C*, *Perl* y *Ruby* [7].

Podemos visualizar de manera resumida una tarjeta Raspberry Pi en la Figura 2.1

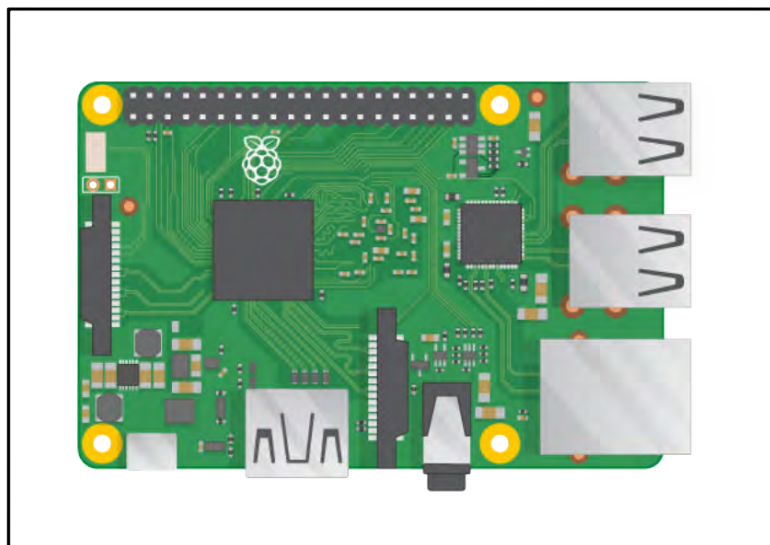


Figura 2.1. Raspberry Pi [23]

A continuación se describe brevemente cada uno de los modelos que han sido comercializados por la fundación Raspberry:

- **Raspberry Pi 1 Modelo A**

Este fué el primer modelo de Raspberry, sus ventas comenzaron en el año 2012. Carecía de puerto Ethernet, por lo que para su conexión a Internet requería de un adaptador Wi-Fi por USB. Poseía 26 conectores GPIO, salida de vídeo via HDMI y Video RCA, un conector Jack de 3.5 milímetros, un único conector USB, Micro-USB (De alimentación) y un conector de cámara. Su procesador fué un Broadcom BCM2835, Single-Core a 700MHz. También tuvo 256 MB de RAM y una gráfica Broadcom VideoCore IV. Requería de una fuente de alimentación de 5 voltios y 2 amperios, elemento común al resto de versiones. Actualmente se ha discontinuado.

- **Raspberry Pi 1 Modelo B y B+**

También del año 2012, es una variante del Modelo A, trajo consigo diversas mejoras, la inclusión del doble de memoria RAM, pasando de 256MB a 512MB. Trajo consigo un puerto USB más, y, por fin, un conector Ethernet (RJ-45) Se mantuvo tanto su tamaño como su coste. No hubo variaciones en el procesador ni en la gráfica. Tiempo después se lanzó el Modelo B+, que incluyó 4 puertos USB y pasó de usar una SD a una MicroSD. Ya no es muy utilizada debido a la evolución en nuevos modelos. Sin embargo siguen en uso.

- **Raspberry Pi 2 Modelo B**

Lanzada en 2014 es el primer modelo que no incluye el mismo procesador usado en los tres anteriores: se sustituye por uno de la misma marca, pero de modelo BCM2836. Pasa de ser de un núcleo a cuatro, y de 700MHz a 900MHz. No obstante emplea la misma gráfica, la VideoCore IV. Dobra la cantidad de memoria RAM, pasando de 512MB a 1GB esta memoria está compartida con la gráfica. También incluye 40

pines GPIO, y mantiene los cuatro puertos USB. Suprime la conexión RCA. Aun tiene vigencia su utilización. Sin embargo representa una alternativa de muy buen precio para el consumidor.

▪ **Raspberry Pi 3 Modelo B**

Sacada a la luz en el año 2016, renueva procesador, una vez más de la compañía Broadcom, una vez más un Quad-Core, pero pasa de 900MHz a 1.20GHz. Mantiene la RAM en 1GB. Su mayor novedad fué la inclusión de Wi-Fi y Bluetooth (4.1 Low Energy) sin necesidad de adaptadores.

2.1.2 Pantalla LCD

Un panel **LCD** o pantalla de cristal líquido (*Liquid Crystal Display*) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. Es frecuente que se utilice en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica. Cada píxel de un *LCD* típicamente consiste en una capa de moléculas alineadas entre dos electrodos transparentes, y dos filtros de polarización. Sin cristal líquido entre el filtro polarizante, la luz que pasa por el primer filtro sería bloqueada por el segundo polarizador. La superficie de los electrodos que están en contacto con los materiales de cristal líquido es tratada a fin de ajustar las moléculas de cristal líquido en una dirección en particular. Este tratamiento suele ser normalmente aplicable en una fina capa de polímero que es unidireccionalmente frotada utilizando un paño. Podemos identificar un panel *LCD* y las partes antes descritas haciendo uso de las Figuras 2.2 y 2.3

Cuando se aplica un voltaje a través de los electrodos, una fuerza de giro orienta las moléculas de cristal líquido paralelas al campo eléctrico, que distorsiona la estructura helicoidal. Esto reduce la rotación de la polarización de la luz incidente, y el dispositivo aparece gris. Si la tensión aplicada es lo suficientemente grande, las moléculas de cristal

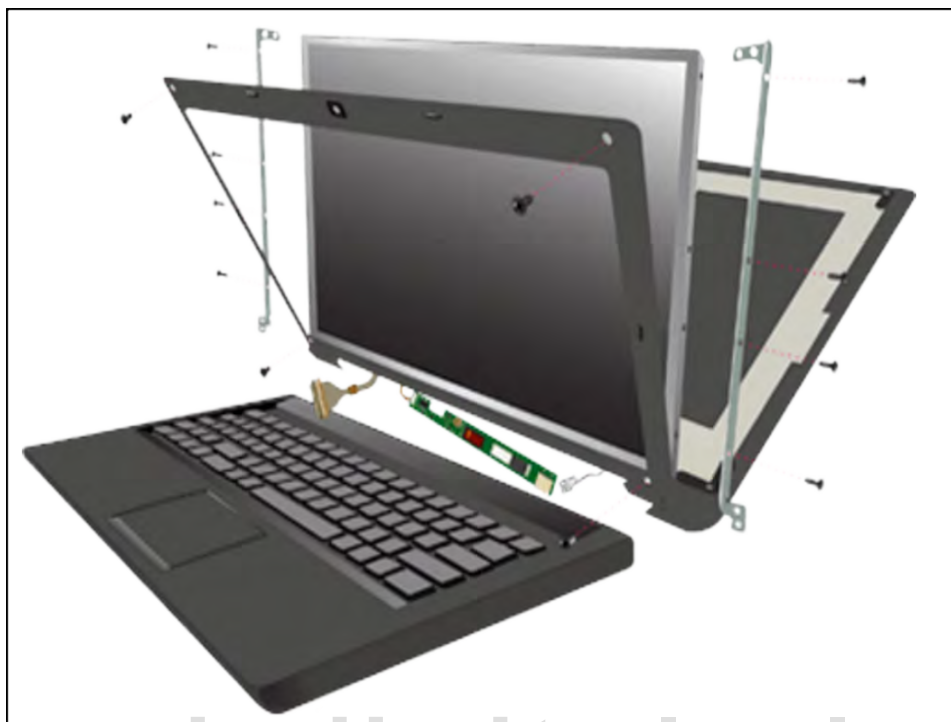


Figura 2.2. Pantalla LCD [24]

líquido en el centro de la capa son casi completamente desenrolladas y la polarización de la luz incidente no es rotada ya que pasa a través de la capa de cristal líquido. Esta luz será principalmente polarizada perpendicular al segundo filtro, y por eso será bloqueada y el pixel aparecerá negro. Por el control de la tensión aplicada a través de la capa de cristal líquido en cada píxel, la luz se puede permitir pasar a través de distintas cantidades, constituyéndose los diferentes tonos de gris.

Cuando un dispositivo requiere un gran número de píxeles, no es viable conducir cada dispositivo directamente, así cada pixel requiere un número de electrodos independiente. En cambio, si la pantalla es multiplexada, los electrodos de la parte lateral de la pantalla se agrupan junto con los cables, y cada grupo tiene su propia fuente de voltaje. Por otro lado, los electrodos también se agrupan normalmente en filas, en donde cada grupo obtiene una tensión de sumidero. Los grupos se han diseñado de manera que cada pixel tiene una combinación única y dedicada de fuentes y sumideros. Los circuitos electrónicos

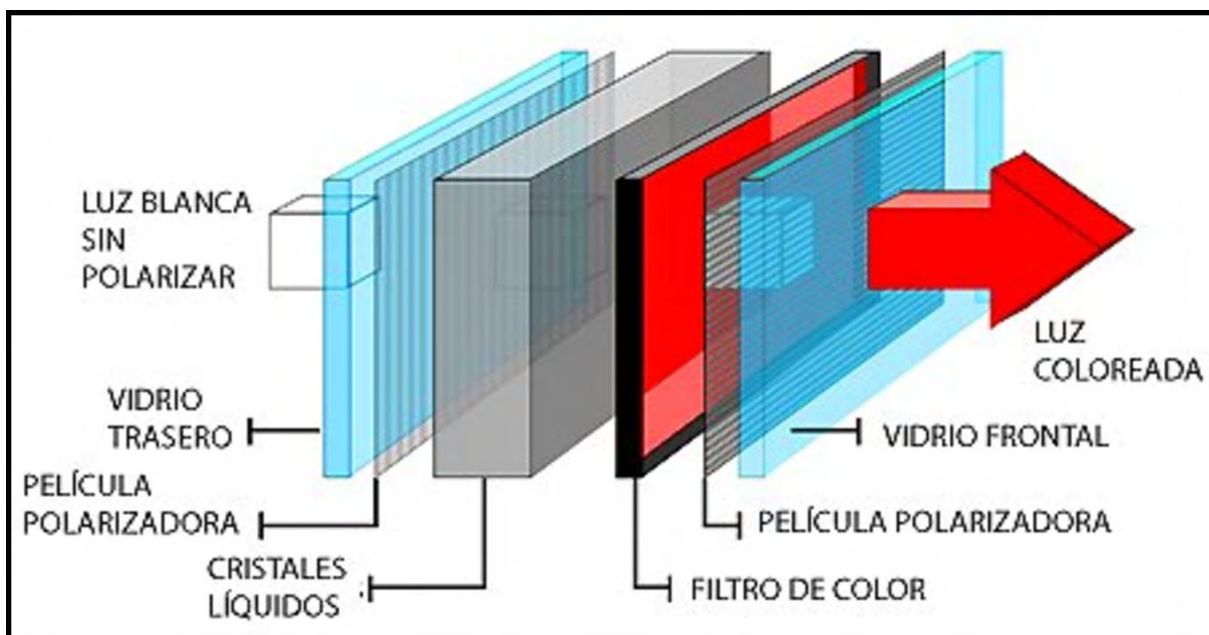


Figura 2.3. Partes de un Panel LCD [25]

o el *software* que los controla, activa los sumideros en secuencia y controla las fuentes de los píxeles de cada sumidero.

2.1.3 Interfaz Electrónica

En electrónica y telecomunicaciones, una **interfaz** es el circuito físico a través del que se envían o reciben señales desde un sistema o subsistemas hacia otros. No existe una interfaz universal, sino que existen diferentes estándares como interfaz *USB*, interfaz *SCSI*, entre otros, que establecen especificaciones técnicas concretas, con lo que la interconexión sólo es posible utilizando la misma interfaz en origen y destino. Así también, una interfaz puede ser definida como un intérprete de condiciones externas al sistema, a través de transductores y otros dispositivos, que permite una comunicación con actores externos, como personas u otros sistemas, a través de un protocolo común a ambos. Una interfaz es una conexión física y funcional entre dos aparatos o sistemas independientes [8].

La computación de entrada salida, o *E/S*, se refiere a la comunicación entre un sistema

de procesamiento de información y los agentes humanos u otro sistema de procesamiento de información. Las entradas son las señales o datos recibidos por el sistema, y salidas son las señales enviadas por este. Un dispositivo de *E/S* es un componente electrónico que permite la transmisión y recepción de información hacia el ordenador o viceversa. Los dispositivos para comunicación entre computadores son típicamente dispositivos de entrada y de salida [9].

En la arquitectura de computadores la combinación entre el *CPU* o unidad central de procesamiento (*Central Processing Unit*) y la memoria principal está considerada el cerebro de la computadora y desde este punto de vista cualquier transferencia de información desde el computador es considerada entrada, y hacia el computador es considerada salida [10]. Y el objetivo principal es interconectar la mayor cantidad de dispositivos a un computador pero se debe tener en cuenta las distintas características que presentan cada uno de ellos y que con frecuencia suelen diferir de las características propias del procesador, podemos destacar que normalmente tienen menor velocidad que el procesador y los códigos que cada uno de ellos emplean para la representación de datos.

Actualmente se usan una multitud de interfaces o controladores para las conexiones entre el procesador y los distintos periféricos donde cada uno de estos últimos suele tener su propio controlador. En ocasiones se puede interconectar los periféricos con la memoria principal directamente, sin pasar por el procesador para lo cual se utilizan dispositivos más avanzados.

2.1.4 Reloj de Tiempo Real

Un *Reloj en tiempo real* o RTC (**R**ea**T**ime **C**lock), es un reloj de computadora, incluido en un circuito integrado, que mantiene la hora actual. Aunque el término normalmente se refiere a dispositivos en computadores personales, servidores y sistemas embebidos, los RTC están presentes en la mayoría de los aparatos electrónicos que necesitan guardar el tiempo exacto. El término se usa para evitar la confusión con los relojes ordinarios que sólo

son señales que dirigen circuitos digitales, y no cuentan el tiempo en unidades humanas. Los RTC no deben ser confundidos con la computación en tiempo real (en inglés, real-time computing), que comparte su acrónimo de tres letras, pero que no se refiere directamente a la hora del día.

Aunque controlar el tiempo puede hacerse sin un RTC, usar uno tiene beneficios como el bajo consumo de energía lo cual es importante cuando está funcionando con una pila; libera de trabajo al sistema principal para que pueda dedicarse a tareas más críticas y algunas veces es más preciso que otros métodos.

Por su parte los RTC tienen una fuente de alimentación alternativa, de esta forma pueden seguir midiendo el tiempo mientras la fuente de alimentación principal está apagada o no está disponible. Esta fuente de alimentación alternativa es normalmente una batería de litio en los sistemas antiguos, pero algunos sistemas nuevos usan un capacitor,? porque son recargables y pueden ser soldados. La fuente de alimentación alternativa también puede suministrar energía a una memoria RAM dentro del mismo RTC, que generalmente almacena la configuración de la BIOS de la placa base.

La mayoría de los RTC usan un oscilador de cristal, pero algunos usan la frecuencia de la fuente de alimentación. En muchos casos la frecuencia del oscilador es 32.768 kHz. Ésta es la misma frecuencia usada en los relojes de cuarzo, y por las mismas razones, que la frecuencia es exactamente 215 ciclos por segundo, que es un ratio muy práctico para usar con circuitos de contadores binarios simples [27].

2.2 Sistema Lógico

Se conoce como *software* al equipo 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* [11].

Los componentes lógicos incluyen, entre muchos otros, las aplicaciones informáticas, tales como el procesador de texto, que permite al usuario realizar todas las tareas concernientes a la edición de textos; el llamado software de sistema, tal como el sistema operativo, que básicamente permite al resto de los programas funcionar adecuadamente, facilitando también la interacción entre los componentes físicos y el resto de las aplicaciones, y proporcionando una interfaz con el usuario.

2.2.1 Sistema Operativo

El *Raspberry Pi* usa mayoritariamente sistemas operativos *GNU/Linux*, que es el término empleado para referirse a la combinación del sistema operativo *GNU*, desarrollado por la *FSF*, y el núcleo o *kernel* Linux. Fue desarrollado por Linus Torvalds y la *Linux Foundation*. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la *GPL* (Licencia Pública General de *GNU*) y otra serie de licencias libres [22].

Debian o Proyecto Debian, es una comunidad conformada por desarrolladores y usuarios, que mantiene un sistema operativo *GNU* basado en software libre. El sistema se encuentra precompilado, empaquetado y en formato para múltiples arquitecturas de computador y para varios núcleos. El proyecto *Debian* fue anunciado inicialmente 1993 por Ian Murdock. El nombre *Debian* proviene de la combinación del nombre de su entonces novia Deborah y el suyo, por lo tanto, Deb(orah) e Ian. Debian 0.01 fue lanzado el 15 de septiembre de 1993 y la primera versión estable fue hecha en 1996.

Raspbian, una distribución derivada de *Debian* que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse [12].

El 3 de junio de 2013, fue lanzado en la web de la fundación para su descarga la aplicación *NOOBS* (**N**ew **O**ut **o**f **B**ox **S**oftware), utilidad que facilita la instalación de diferentes sistemas operativos para Raspberry Pi. Se distribuye en forma de archivo zip

que se copia descomprimido a una tarjeta SD de 4 o más GB, y una vez arrancada la placa con la tarjeta por primera vez, aparece un menú en que se da la opción de instalar una de las diferentes distribuciones en el espacio libre de la tarjeta de memoria, o acceder a Internet con el navegador integrado. *NOOBS* contiene las distribuciones *GNU/Linux* de carácter general Raspbian, Arch Linux ARM y Pidora [13].

2.2.2 Herramientas de Programación

Un **lenguaje de programación** es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana [14].

Está formado por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico o codificación del programa.
- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.
- Desarrollo de la documentación.

Existe un error común que trata por sinónimos los términos *lenguaje de programación* y *lenguaje informático*. Los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como por ejemplo HTML (lenguaje para el marcado de páginas web que no es propiamente un lenguaje de programación, sino un conjunto de instrucciones que permiten estructurar el contenido de los documentos).

A su vez permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural. Una característica relevante de los lenguajes de programación es precisamente que más de un programador pueda usar un conjunto común de instrucciones que sean comprendidas entre ellos para realizar la construcción de un programa de forma colaborativa.

2.2.2.1 SQL

Se define **SQL** o lenguaje de consulta estructurada (*Structured Query Language*), como un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas [17].

Originalmente basado en el álgebra relacional y en el cálculo relacional, *SQL* consiste en un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de control de datos. El alcance de *SQL* incluye la inserción de datos, consultas, actualizaciones y eliminación. También la creación y modificación de esquemas y el control de acceso a los datos. Por su parte *SQL* a veces se describe como un lenguaje declarativo, también incluye elementos procesales. Haciendo uso de la Tabla 2.1 podemos ver un ejemplo de sentencias **SQL**, en donde se crea una base de datos, se crea una tabla, se insertan datos,

Cuadro 2.1. Ejemplo de Sentencias SQL

```

1 CREATE DATABASE mydb;
2 USE mydb;
3 CREATE TABLE mitabla (id INT PRIMARY KEY, nombre VARCHAR(20));
4 INSERT INTO mitabla VALUES (1, 'Will');
5 INSERT INTO mitabla VALUES (2, 'Marry');
6 INSERT INTO mitabla VALUES (3, 'Dean');
7 SELECT id, nombre FROM mitabla WHERE id = 1;
8 UPDATE mitabla SET nombre = 'Willy' WHERE id = 1;
9 SELECT id, nombre FROM mitabla;
10 DELETE FROM mitabla WHERE id = 1;
11 DROP DATABASE mydb;
12 SELECT count(1) from mitabla;

```

se seleccionan datos, entre otras funciones.

SQL fue uno de los primeros lenguajes comerciales para el modelo relacional de Edgar Frank Codd como se describió en su artículo de 1970 *"El modelo relacional de datos para grandes bancos de datos compartidos"*. A pesar de no adherirse totalmente al modelo relacional descrito por Codd, pasó a ser el lenguaje de base de datos más usado.

También pasó a ser el estándar del *Instituto Nacional Estadounidense de Estándares (ANSI)* en 1986 y de la *Organización Internacional de Normalización (ISO)* en 1987. Desde entonces, el estándar ha sido revisado para incluir más características. A pesar de la existencia de ambos estándares, la mayoría de los códigos *SQL* no son completamente portables entre sistemas de bases de datos diferentes sin ajustes.

2.2.2.2 PHP

La sigla **PHP** identifica a un lenguaje de programación que nació como *Personal Home Page Tools*. Fue desarrollado por el programador de origen danés Rasmus Lerdorf en 1994 con el propósito de facilitar el diseño de páginas web de carácter dinámico. Y hoy en día es desarrollado por *The PHP Group* aunque carece de normativa formal, la *Free Software Foundation* considera la licencia *PHP* como parte del software libre [18].

Cuadro 2.2. Ejemplo de Código PHP

```

1  <?php
2  include('conexion\_db.php');
3  $sql = "SELECT * FROM client";
4  $result = $conn->query($sql);
5  if ($result->num_rows > 0)
6  {
7      while ($row = mysqli_fetch_array($result))
8      {
9          echo " <tr> <td>" . $row["cliente_id"] . "</td>"
10             . " <td>" . $row["cliente_name"] . "</td>"
11             . " <td>" . $row["cliente_email"] . "</td>"
12             . " <td>" . $row["cliente_phone"] . "</td> "
13             . " <td>" . $row["cliente_reg"] . "</td></tr>";
14      }
15  }
16  else
17  {
18      echo "0 results";
19  }
20  $conn->close();
21  ?>

```

El lenguaje *PHP* se procesa directamente en el servidor, incrustándose dentro del código *HTML* de las páginas web, y su escritura está basada en un conjunto de tipo estructurado con sintaxis similar que el lenguaje *C*, lo que es una ventaja para que programadores puedan desarrollar aplicaciones complejas en poco tiempo y a demás es libre para descargar y utilizar. Otra característica es que permite establecer una conexión con la base de datos, a través de lo cual es posible, por ejemplo, presentar en pantalla datos personales del cliente cuando éste lo requiera. No solo con eso *PHP* permite modificar dinámicamente el contenido de una página, lo cual resulta esencial hoy en día, pero sumado a esto puede también modificar información en bases de datos, controlar el acceso de usuarios y cifrar información importante que lo requiera. Para comprender algunas características se puede analizar en la Tabla 2.2, donde observamos la utilización de lenguaje *PHP* para realizar

una conexión a la base de datos mediante el archivo `conexion_db.php` y posteriormente hacer una consulta para listar las filas `cliente_name`, `cliente_email`, `cliente_phone`, `cliente_reg` de un campo de tabla `client`.

2.2.2.3 HTML

HTML o lenguaje de marcas de hipertexto (*HyperText Markup Language*), que podría ser definido como un lenguaje que genera un formato de documentos usando hipertexto. Es un lenguaje informático que se utiliza para el desarrollo de páginas de web. El código *HTML* se crea a partir de etiquetas, también llamadas *tags*, que permiten conectar diversos conceptos y formatos. Para la escritura de este lenguaje, se crean etiquetas que aparecen especificadas dentro de corchetes o paréntesis angulares "`< >`". Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades: el contenido en sí mismo y sus atributos.

Se puede comenzar a diseñar una página web utilizando comandos ya preestablecidos de *HTML* donde se le dan características más específicas a la misma. Podemos hablar del:

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

Donde encontraremos instrucciones que dan referencia y características a la página, algunas de estas instrucciones más utilizadas son:

```
<META http-equiv="refresh" content="5">
```

Que recarga la página automáticamente según un tiempo establecido el cual puede ir desde segundos hasta días, en este caso `...content="5">` indica que la misma se va a recargar cada cinco segundos automáticamente. El título de la página es un detalle que nos muestra en la pestaña de los navegadores el nombre de la misma.

```
<title> Titulo Principal </title>
```

Para darle más personalidad a las páginas se utilizan los llamados *shortcut icon* que en pocas palabras son los iconos que aparecen al lado izquierdo del título de la página en las pestañas de los navegadores:

```
<link rel="shortcut icon" type="image/x-icon"/>
```

No podemos dejar de mencionar como se realiza la conexión entre el archivo de estilos y el programa principal, la cual se logra haciendo uso de la siguiente declaración de relación:

```
<link rel='stylesheet' href='Style.css' type='text/css'/>
```

Seguido de estas instrucciones se crea el cuerpo de la página que inicia con los comandos:

```
<body> </body>
```

Esta sección encierra todo el contenido de la página web que será observado por los usuarios, es decir, texto, animaciones, hipervinculos y demás características.

Para comprender un poco mas el funcionamiento de este lenguaje y algunas de sus características básicas tenemos como muestra un código completo de una pagina web con hipervinculos, cabecera **header**, pie de página **footer**, bloques de divisiones y tipo de fuente para las letras, se puede observar el archivo Prueba.html en la Tabla 2.3 que contiene las características antes mencionadas.

Para una mejor visualización de la página web se crea un archivo *CSS* (*Cascading Style Sheet*) donde se guardan todos los estilos gráficos que se utilizarán en la pagina web, tipos de letras, colores de fondo, bloques de divisiones con sus respectivos tamaños y posiciones. De esta forma desde el programa principal, se utilizan todos los estilos como si se trataran de las llamadas librerías en otros lenguajes de programación teniendo un código mas limpio.

En el archivo de estilos **CSS**, visto en la Tabla 2.4 se utilizan instrucciones de diseño como:

- **background-color:** Se utiliza para agregar color al fondo de cualquier bloque de división, el mismo se especifica en código hexadecimal o por el nombre del color en inglés.
- **color:** Define el color de la letra que va a utilizar, también se especifica en código hexadecimal o por el nombre del mismo en inglés.
- **text-align:** Especifica la ubicación del texto que puede ser centrado, justificado, alineado a la derecha o izquierda, en la parte superior del bloque o en la parte inferior.
- **padding:** Tiene como función ubicar el bloque o texto con respecto a los márgenes superior, inferior, izquierdo y derecho. Se puede dar un valor como instrucción en centímetros, píxeles o porcentajes y de otra forma acepta los valores de separación con respecto a cada margen.
- **align:** Se utiliza en bloques de división para establecer su posición en la página.
- **width:** Establece el ancho del bloque de división en píxeles.
- **height:** Establece el alto del bloque de división en píxeles.
- **margin:** Especificando en centímetros la separación entre bloques de división.
- **border:** Dibuja el borde en un bloque de división o encierra textos en un cuadro, dependiendo de la instrucción se especifica el tipo y color de borde.

En este lenguaje no existen reglas para especificar los nombres de las etiquetas que se utilizarán al programar, por eso se dice que es un sistema de formato abierto.

Por otra parte, cabe destacar que el HTML permite ciertos códigos que se conocen como **scripts**, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los scripts que pueden agregarse, los más conocidos y utilizados son javascript [15].

Cuadro 2.3. Archivo Prueba.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html
5     charset=utf-8" />
6 <meta http-equiv="refresh" content="5">
7 <link rel="shortcut icon" type="image/x-icon"
8     href="img/favicon.ico">
9 <title>Titulo Pestaña</title>
10 <link rel='stylesheet' href='Style.css' type='text/css' />
11 </head>
12 <body>
13 <div id="header"><h1>Titulo Principal</h1></div>
14 <div class="left_panel">
15 <div id="Div2"><div ALIGN=center><h1>Titulo</h1></div>
16     </div></div>
17 <div class="right_panel">
18 <div id="botones">
19 <button type="button"><a href="Prueba.html?">
20     <h1>CLICK</h1></a></button></div>
21 <div id="IMAGENES"><img src='imagenes/IMAGEN.png'
22     alt='IMG' /></div></div>
23 <div id="footer"><div <p class="italic">Raspberry Pi 3</p>
24     </div></div>
25 </body>
26 </html>

```

2.2.2.4 Lenguaje C

El lenguaje *C* es considerado un lenguaje de medio nivel estructurado y con características típicas de un lenguaje de alto nivel, permite de alguna forma crear bloque de tareas o procedimientos dentro de otras funciones llevándonos a decir que se trata de un lenguaje estructurado.

Está compuesto por un programa principal definido por una función en la que se configuran los tipos de variables locales a utilizar y un bucle infinito que es donde se ejecutara el programa. Añadido a esto, dependiendo de las necesidades del programador

Cuadro 2.4. Archivo de Estilos Styles.css

```

1  /* CSS Document */
2  #header {
3      background-color:blue;
4      color:white;
5      text-align:center;
6      width: 600px;
7      position:center;}
8  #section {
9      float: center;
10     align:center;
11     width: 600px;}
12 #footer {
13     background-color:black;
14     color:white;
15     clear:both;
16     text-align:center;
17     width: 600px;}
18 #Botones {
19     background-color:gray;
20     margin-bottom: 1cm;
21     width: 70px;
22     height: 50px;
23     p.italic {
24         font-style: italic;}

```

se crean funciones auxiliares y librerías que se llaman a partir del bucle infinito.

Las llamadas funciones auxiliares y librerías pueden encontrarse en el mismo programa o en extensiones para facilitar la depuración o edición de las mismas, de esta forma llevar un orden en la programación. También se puede hablar de estructuras de repetición como la que se utiliza en la Tabla 2.5 y donde se aprecia una estructura básica de un programa escrito en lenguaje C para fines de uso en microprocesadores [16].

El lenguaje *C* posee palabras reservadas en su estructura que son solo para un uso determinado en la programación. También hay clasificaciones para los tipos de datos a utilizar ya que podemos elegir el rango de memoria mas conveniente en cada caso. En

fin podemos decir que el lenguaje *C* es muy completo en cuanto a que nos brinda la oportunidad de diseñar programas bien estructurados.

Cuadro 2.5. Ejemplo de Código en C

```

1  #include <xc.h>
2  #define _XTAL_FREQ 4000000
3  void main(void)
4  {
5      char contador;
6      OSCCON = 0x68; //oscilador interno a 4MHz
7      while(IOFS==0) continue; //frecuencia sea estable
8      TRISA = 0xFF; // Pines del puerto A como entrada
9      TRISB = 0x00; // Pines del puerto B como salida
10     PORTB = 0b00111111;
11     while(1)
12     {
13         // Bucle Infinito
14         while(RA0 == 1) continue;
15         __delay_ms(1);
16         for(contador = 4; contador = 0; contador--)
17         {
18             switch(contador)
19             {
20                 case 4: PORTB = 0b01100110; break;
21                 case 3: PORTB = 0b01001111; break;
22                 case 2: PORTB = 0b01011011; break;
23                 case 1: PORTB = 0b00000110; break;
24                 case 0: PORTB = 0b00111111; break;
25             }
26             __delay_ms(250);
27         }
28     }

```

2.2.3 Base de Datos

Una **base de datos** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impre-

sos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados *DBMS* o sistemas gestores de bases de datos (*Data Base Management System*), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos *DBMS*, así como su utilización y administración, se estudian dentro del ámbito de la informática. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas, también son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

2.2.3.1 MySQL

El *software* de bases de datos **MySQL** consiste de un sistema cliente/servidor que se compone de un servidor *SQL* multihilo, varios programas, clientes, bibliotecas, herramientas administrativas y una gran variedad de interfaces de programación. Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido y más fácil de manejar. Una base de datos es una colección estructurada de datos. La información que se puede almacenar en una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas o la información generada en una red corporativa. Y para agregar, acceder, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como **MySQL** [19]

2.2.4 Servidor Web

Un **servidor web** utiliza protocolo de transferencia de hipertexto *HTTP* (*Hypertext Transfer Protocol*), es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas

con el cliente y generando una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente es renderizado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo.

2.2.4.1 Apache

El servidor *HTTP Apache* es un servidor web de código abierto, para plataformas *Unix* (*BSD*, *GNU/Linux*), *Microsoft Windows*, *Macintosh* y otras, que implementa el protocolo *HTTP* y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular *NCSA HTTP*, pero más tarde fue reescrito por completo. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu *Apache* fue la última en rendirse al que pronto se convertiría en gobierno de EEUU.

Además *Apache* consistía solamente en un conjunto de parches a aplicar al servidor de *NCSA*. En inglés, a *patchy server* (un servidor "parcheado") suena igual que *Apache Server*. El servidor *Apache* es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la *Apache Software Foundation* dentro del proyecto *HTTP Server*.

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

Apache tiene amplia aceptación en la red: desde 1996, *Apache*, es el servidor *HTTP* más usado. Jugó un papel principal en el desarrollo fundamental de la *World Wide Web* y alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70 % de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. En 2009 se convirtió en el primer servidor web que alojó más de 100 millones de sitios web [20].

2.2.5 Reproductor Multimedia

Un **reproductor multimedia**, reproductor de medios es un programa informático o un dispositivo capaz de mostrar gran cantidad de contenidos audiovisuales. Por norma general, esto incluye la reproducción de sonido, vídeo e imágenes. De esta manera, el reproductor de medios permite el disfrute personal de música, videoclips, películas, fotografías, entre otros.

Existen multitud de programas informáticos que pueden denominarse reproductores de medios, si bien difieren en funcionalidades y popularidad.

2.2.5.1 OSMC y VLC

- **OSMC** de sus siglas en inglés *Open Source Media Center*, es un reproductor multimedia gratuito y de código abierto basado en Linux. Fundado en 2014, *OSMC* le permite reproducir multimedia desde una red local, almacenamiento adjunto e Internet. *OSMC* es el centro de medios líder en términos de conjunto de características y comunidad y se basa en el proyecto *Kodi* y es el reproductor predeterminado de Raspbian.
- **VLC media player** de sus siglas en inglés **V**ideo **L**AN **C**lient, es un reproductor y *framework* multimedia, libre y de código abierto desarrollado por el proyecto *VideoLAN*. Es un programa multiplataforma con versiones disponibles para muchos sistemas operativos, es capaz de reproducir casi cualquier formato de video sin necesidad de instalar códecs externos y puede reproducir videos en formatos DVD, Bluray, a resoluciones normales, en alta definición o incluso en ultra alta definición o 4K. *VLC* es un reproductor de audio y video capaz de reproducir muchos códecs y formatos de audio y video, además de capacidad de streaming. Es software libre, distribuido bajo la licencia *GPL* [21].

CAPÍTULO 3

DISEÑO

Es imprescindible poder identificar el funcionamiento de cada etapa de este proyecto, con el fin de resaltar su potencial y utilidad, es por ello que en este capítulo se abordará el desarrollo del prototipo funcional, empezando por su parte física y culminando por el sistema lógico. Dando así un recorrido extenso por cada parte que ha sido esencial en este trabajo y que va a representar un punto de inicio para pequeños emprendedores en donde resalte la utilidad de componentes electrónicos que tienden al futuro, como los computadores de placa reducida y destacando el uso de los mismos de forma remota, a través de herramientas libres como servidores web y bases de datos.

3.1 Prototipo Físico

En esta etapa se desea integrar de manera sistemática cada elemento principal del proyecto, como lo son la **Pantalla LCD**, la **Raspberry Pi 3** y el **Controlador de Pantalla**. Este último representa la interfaz para la comunicación entre la computadora de placa reducida y el panel LCD. Para lograr un mejor entendimiento del prototipo funcional, haremos uso de la Figura 3.1, la cual resume los principales elementos que forman el hardware, siendo estos: la tarjeta Raspberry Pi, la interfaz de comunicación de la pantalla, el inversor de la pantalla, la pantalla y el modulo RTC.

La *Raspberry Pi 3* como modelo mas reciente y gracias a sus características nuevas, como la conexión inalámbrica, fue en principio bastante atractivo para la ejecución del proyecto. Esto implicaría que la comunicación de forma remota se lograra con éxito. El

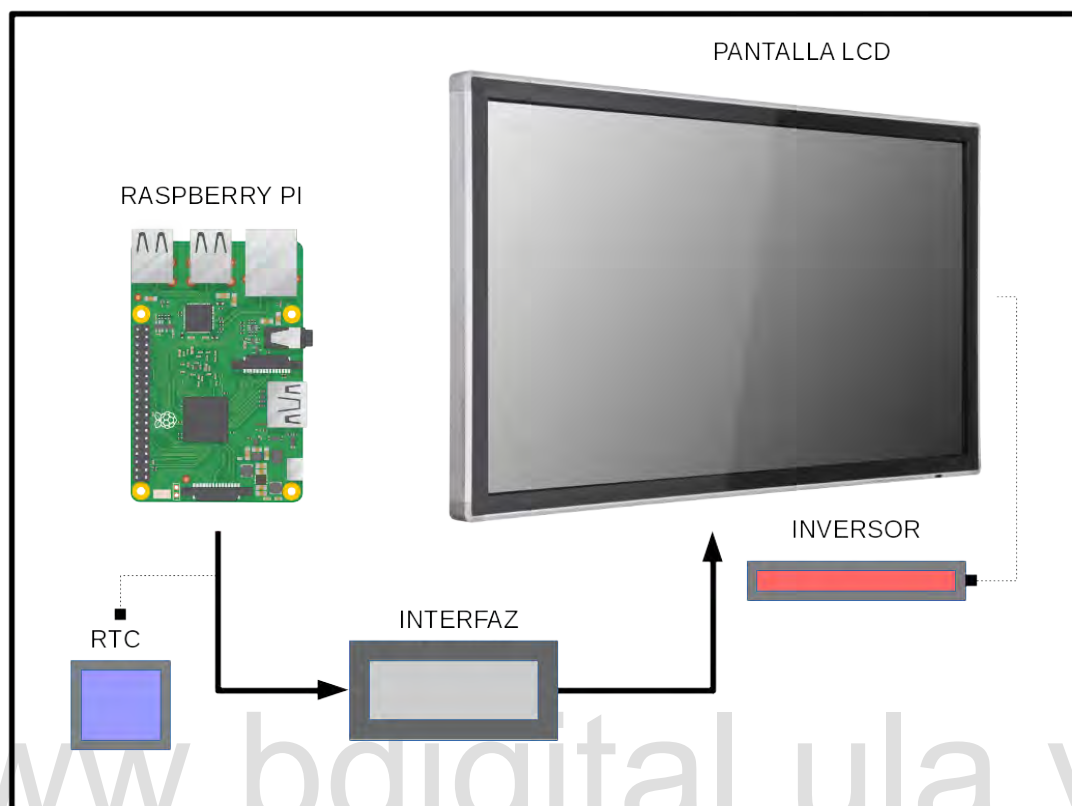


Figura 3.1. Principales Componentes

prototipo debía convertirse en un solo sistema, y es por ello que el principal paso fue alojar todos sus componentes en una carcasa. En donde se sitúa la pantalla y detrás de la misma la computadora de placa reducida y el controlador de pantalla. En una primera etapa se ubicó cada uno de los componentes en una carcasa plástica de manera que pudiera hacerse posteriormente las conexiones de alimentación, como se observa en la Figura 3.2. Cabe destacar que para esta etapa no se disponía del módulo RTC, posteriormente fue añadido al prototipo.

Para lograr que la Raspberry Pi funcione se alimenta con 5 Voltios, esta alimentación la logramos mediante un puerto microUSB o por los pines de la *GPIO* 02 y 06 como se observa en la Figura 3.3, esto va a representar una simplificación importante ya que del controlador de pantalla vamos a poder sacar esta tensión requerida de 5 Voltios. Dicho

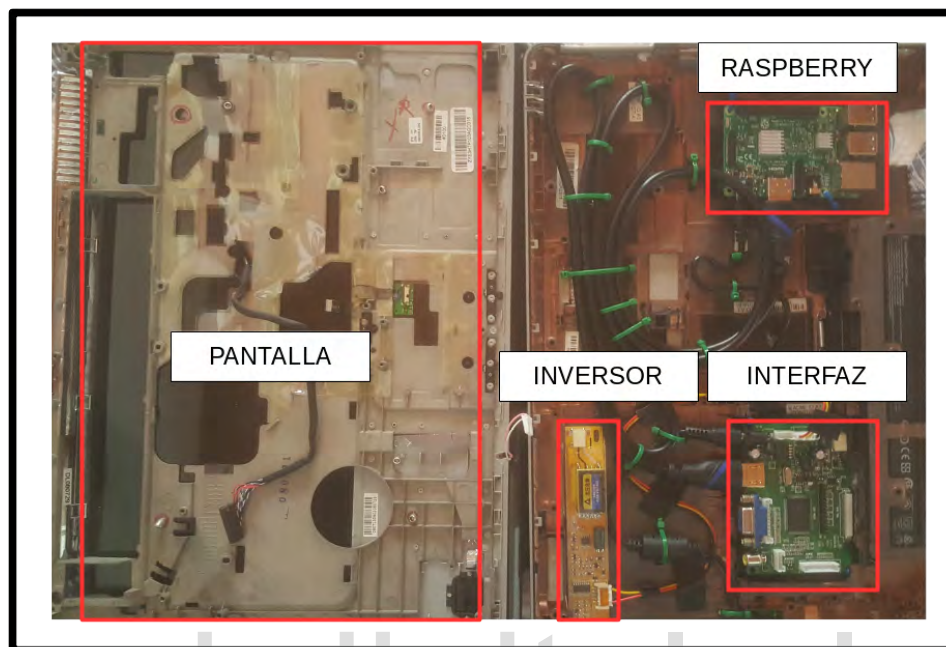


Figura 3.2. Primera Carcasa

controlador requiere a su vez 5 y 12 Voltios para su funcionamiento, es decir, vamos a alimentar un solo dispositivo, en este caso el controlador de pantalla; y de manera interna realizaremos un puente para alimentar la *Raspberry Pi 3* y el modulo RTC. El principal objetivo es la simplificación de cableado que va a representar ese puente de alimentación, ya que podemos hacer uso de un mismo transformador para alimentar nuestras dos partes vitales del proyecto como la computadora de placa reducida y el controlador de pantalla.

Es momento de hablar del controlador de pantalla, el cual será la interfaz de comunicación entre la *Raspberry Pi 3* y la pantalla, en este caso el controlador usado fue PCB800099-RTD2662, que es un controlador para pantallas *LCD* el cual tiene una conexión *LVDS*, la cual es muy usual en pantallas de computadoras portátiles y a su vez permite que hacer uso de los puertos *VGA* o *HDMI* de la misma para establecer comunicación con

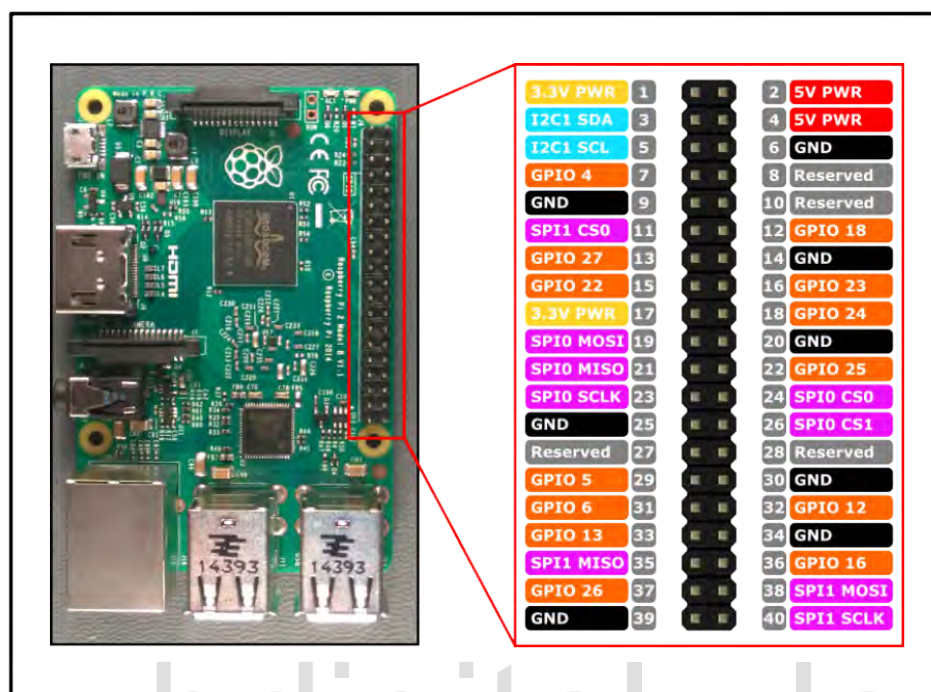


Figura 3.3. Pines de la Raspberry Pi 3 [26]

la *Raspberry Pi 3*.

Haciendo uso de la figura 3.4, podemos identificar el controlador de pantalla que se usó. Este controlador está basado en un procesador *Realtek RTD2662*, *Realtek* fabrica y vende una amplia variedad de productos en todo el mundo, y sus líneas de productos pueden clasificarse en dos subdivisiones: circuitos integrados para *Comunicaciones en Red*, y circuitos integrados *Multimedia* y de periféricos de computadora.

La comunicación entre la pantalla y el controlador no es mas que a travez de un conector *LVDS* o sistema de señal diferencial de bajo voltaje (*Low Voltage Differential Signaling*), es un sistema de transmisión de señales a alta velocidad sobre medios de transmisión de bajo costo, como puede ser el par trenzado. Fue introducido en 1994 y se hizo popular en redes de computadores de alta velocidad para la transmisión de datos. También

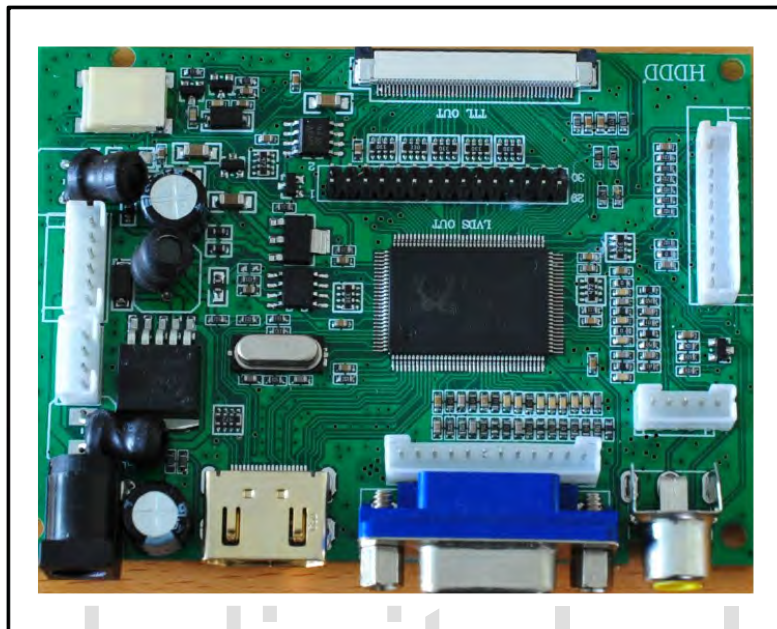


Figura 3.4. Controlador PCB800099-RTD2662

su respectiva alimentación al inversor de la pantalla, en donde, este inversor es un componente en las computadoras portátiles que proporciona la energía para proporcionar una luz de fondo para el monitor, lo que permite al usuario ver lo que hay en la pantalla. La mayoría de pantallas *LCD* contiene luces frías fluorescentes, que proporcionan los tubos de la luz en la parte posterior de la pantalla. Los inversores del *LCD* toman la entrada de la corriente continua y la invierten a una mayor tensión de alimentación de corriente alterna a los tubos de la luz de la pantalla. La mayoría de los inversores de *LCD* proporcionan 5 o 12 Voltios de corriente continua y entre 1.200 y 2.500 *VRM* (módulos de regulación de voltaje) y están hechos para funcionar entre menos 40 grados centígrados y 85 grados centígrados.

Por ultimo la incorporación del modulo RTC, el cual no represento mayor dificultad ya

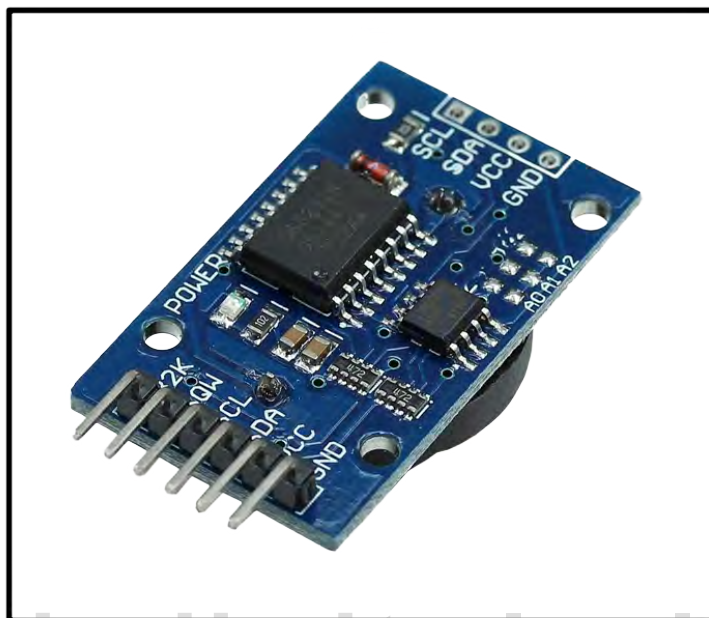


Figura 3.5. Módulo Arduino DS3231

que se utilizó un módulo Arduino DS3231, el cual se adapta completamente a la Raspberry Pi, utilizando cuatro pines de conexión: VCC (pin 04), GND (pin 06), SCL (pin 03), SDA (pin 05). En la Figura 3.5 podemos ver el módulo usado.

3.2 Programación

En esta sección formalizaremos la manera en como se elaboró la parte lógica del proyecto, empezando con la construcción de la base de datos, pasando por el diseño del servidor web y por último la programación del sistema de reproducción. Este último siendo muy importante, ya que es el que va a determinar el funcionamiento en conjunto de todo el proyecto.

A su vez, se enfocó el desarrollo del proyecto en un funcionamiento exclusivo para fines publicitarios de ambientes comerciales, donde se desee gestionar la propaganda alusiva a algún producto o local. Estableciendo tiempo de funcionamiento predeterminado según los requerimientos del beneficiario. El funcionamiento de esta etapa se enfoca principalmente en un sistema de gestión integral para el control de usuarios y de información. El cual permitirá poder registrar nuevos usuarios, cargar el contenido y hasta modificar los planes de gestión de publicidad, los cuales serán preestablecidos según las necesidades del cliente.

En términos resumidos el sistema funcionará de la siguiente forma:

- El sistema dispondrá de un acceso exclusivo para el operador o encargado de subir la información, el cual tendrá un nombre de usuario y contraseña único.
- Se dispondrá de un sistema de menú principal, en el cual se podrá seleccionar otros subsistemas como el de gestión de clientes, gestión de planes y gestión de publicidades.
- Los subsistemas mencionados en el ítem anterior cuentan con herramientas para crear, modificar, actualizar o borrar cualquier entrada de los formularios correspondientes. Así como particularmente el sistema de gestión de clientes, cuenta con la posibilidad de cargar los videos a la base de datos desde el servidor web.

Para lograr un completo cumplimiento de los objetivos cabe destacar, que este proyecto se desarrolló única y exclusivamente con herramientas y sistemas operativos de licencia libre. Donde de manera simultanea todo fue probado en un computador portátil con un sistema operativo *Debian*, en donde se utilizó diferentes interfaces de desarrollo como *NetBeans* para la programación del servidor web, a su vez el acceso a las bases de datos para las primeras pruebas se hizo por la línea de comandos en el terminal *Linux* y por último el uso de *Vim* como editor de texto para el desarrollo del código responsable de la reproducción de contenido, como el compilador interno de *Linux gcc*.

En la Figura 3.6 se puede ver un esquema resumido del sistema:

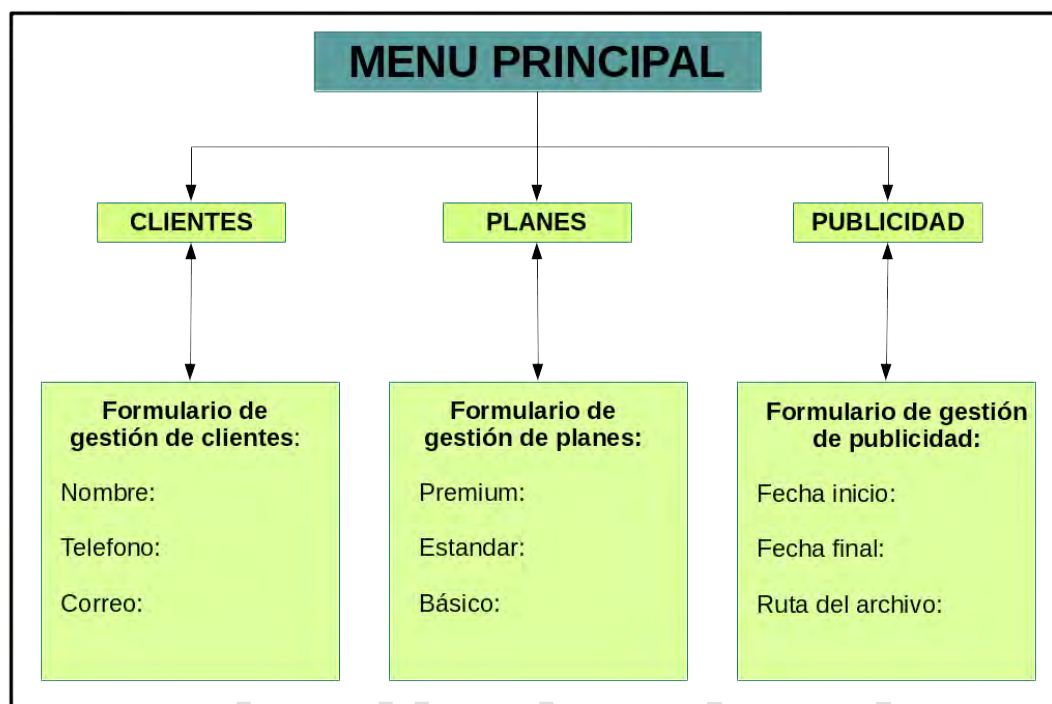


Figura 3.6. Esquema del Sistema

3.2.1 Base de Datos MySQL

Es sin duda la base de la pirámide, ya que se requiere almacenar en alguna parte información esencial del prototipo funcional, es decir, se debe cargar en algún lugar el contenido que se desea reproducir, y que mejor que en una base de datos, en donde también se guarde información de clientes y se pueda gestionar hasta un sistema de pago si se quiere.

Esta etapa inicia con la utilización de *MySQL* como gestor de bases de datos, si bien en una primera etapa se logra crear y acceder a las bases de datos a través de línea de comandos, como se puede observar mediante el código del Cuadro 3.1, que es muy usual en sistemas operativos como *Debian*.

Cuadro 3.1. Acceso a MySQL por Línea de Comandos

```
~$ mysql -h localhost -u root -p password
```

Por su parte la Fig. 3.7, describe el ambiente al que se tiene acceso mediante el terminal *linux*, pero no resultaba del todo funcional. Es por ello que se busca lograr crear una conexión a la base de datos y hacer uso de la misma de una forma mas sencilla. Donde sea mas intuitivo el ingreso de información y esto represente ergonomía para el operador.

```
mysql> use myDB;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_myDB |
+-----+
| client          |
| plan           |
| publicidad      |
+-----+
3 rows in set (0.00 sec)

mysql> select * from publicidad;
+-----+-----+-----+-----+-----+-----+
| publi_id | plan_id | publi_ruta | publi_inicio | publi_fin | client_id |
+-----+-----+-----+-----+-----+-----+
| 1        | 1       | v1.mp4     | 2017-07-13 18:20:42 | 2017-10-01 00:00:00 | 1         |
| 2        | 1       | v2.mp4     | 2017-07-13 18:21:01 | 2017-10-01 00:00:00 | 1         |
| 3        | 1       | v3.mp4     | 2017-07-13 18:21:11 | 2017-10-01 00:00:00 | 1         |
| 4        | 1       | v4.mp4     | 2017-07-13 18:21:21 | 2017-10-01 00:00:00 | 1         |
| 5        | 1       | v5.mp4     | 2017-07-13 18:21:30 | 2017-10-01 00:00:00 | 1         |
| 6        | 1       | v6.mp4     | 2017-07-13 18:21:39 | 2017-10-01 00:00:00 | 1         |
| 7        | 1       | v7.mp4     | 2017-07-13 18:21:47 | 2017-10-01 00:00:00 | 1         |
| 8        | 1       | v8.mp4     | 2017-07-13 18:21:57 | 2017-10-01 00:00:00 | 1         |
| 9        | 1       | v9.mp4     | 2017-07-13 18:22:05 | 2017-10-01 00:00:00 | 1         |
| 10       | 1       | v10.mp4    | 2017-07-13 18:22:13 | 2017-10-01 00:00:00 | 1         |
| 11       | 1       | v11.mp4    | 2017-07-13 18:22:22 | 2017-10-01 00:00:00 | 1         |
| 12       | 2       | v12.mp4    | 2017-07-13 18:22:30 | 2017-10-01 00:00:00 | 1         |
| 13       | 1       | v13.mp4    | 2017-07-13 18:22:42 | 2017-10-01 00:00:00 | 1         |
| 14       | 1       | v14.mp4    | 2017-07-13 18:22:48 | 2017-10-01 00:00:00 | 1         |
| 15       | 1       | v15.mp4    | 2017-07-13 18:23:02 | 2017-10-01 00:00:00 | 1         |
| 16       | 2       | v16.mp4    | 2017-07-13 18:23:13 | 2017-10-01 00:00:00 | 1         |
+-----+-----+-----+-----+-----+-----+
```

Figura 3.7. Vista del Acceso a la Base de Datos desde Terminal Linux

En la Raspberry se utilizó una manera alternativa de acceder y gestionar la base de datos para la creación de tablas y manipulación de contenido, esta fue el uso de *phpm-*

yadmin, la cual no es mas que una aplicación con interfaz gráfica mucho mas sencilla para usar la base de datos que desde el terminal de Linux.

Esto nos lleva al siguiente punto, donde crearemos un espacio gráfico que sea mas sencillo y donde se logre gestionar mas información. Sin embargo el desarrollo de la base de datos tendrá su lugar a lo largo del proyecto mediante programación conjunta en *PHP*, donde lograremos crear una conexión a la base de datos, hacer uso de la misma, crear tablas, insertar información, modificar información y extraer información.

A continuación se describe mediante el código del Cuadro 3.2 la forma en como se creó la conexión a la base de datos con un código *PHP* y sentencias *SQL*.

Cuadro 3.2. Conexión a la Base de Datos en PHP

```
1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "raspberrypi";
5 $dbname = "myDB";
6 $conn = new mysqli($servername, $username, $password, $dbname);
7 if ($conn->connect_error)
8 {
9     die("Connection failed: " . $conn->connect_error);
10 }
11 ?>
```

Por su parte la creación de una base de datos y de tablas se ejecuta mediante el siguiente código del Cuadro 3.3. De esta forma se realizó la creación de tablas para su posterior uso, que en la siguiente sección se logrará acceder haciendo uso del servidor web.

3.2.2 Servidor Web Apache

Luego de la instalación del servidor *Apache* en la *Raspberry Pi 3*, procedemos a desarrollar un entorno mas intuitivo y sencillo para darle continuidad al proyecto. En donde sea de mucha utilidad para el operador poder ingresar información relacionada al contenido

Cuadro 3.3. Creación de Base de Datos y Tabla de Publicidad en PHP

```

1  <?php
2  include 'conexion_db.php';
3  $sql = "CREATE DATABASE myDB";
4  if ($conn->query($sql) === TRUE) {
5      echo "Database created successfully";
6  }
7  else {
8      echo "Error creating database: " . $conn->error;
9  }
10 $sql = "CREATE TABLE publicidad (
11     publi_id      INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
12     plan_id       INT(6),
13     publi_ruta    VARCHAR(60),
14     publi_inicio  TIMESTAMP,
15     publi_fin     TIMESTAMP,
16     client_id     INT(6))";
17 if ($conn->query($sql) === TRUE) {
18     echo "La tabla fue creada correctamente";
19 }
20 else {
21     echo "Error creando tabla: " . $conn->error;
22 }
23 $conn->close();
24 ?>

```

multimedia que se desee exponer, como información del cliente, datos de la suscripción, fecha en la que se inicia la reproducción del contenido, entre otros.


El funcionamiento es básicamente acceder al servidor mediante un usuario y contraseña, agregar un cliente con algunos datos de interés y por ultimo cargar un video publicitario con fechas predeterminadas de reproducción y con una ubicación o ruta específica para poder ser cargado a la base de datos mediante el servidor web.

Como se explicó en la sección 2.2.2.3 del Capítulo anterior, todo el desarrollo del código *HTML* va a estar encabezado por el código del Cuadro 3.4. En donde se evidencia el uso de *Bootstrap* para el uso de las hojas de estilo y así simplificar el desarrollo de esta sección.

Cuadro 3.4. Encabezado HTML

```
1 <!DOCTYPE html >
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="device-width, initial-scale=1">
6 <title>Theme Template for Bootstrap</title>
7 <link href="css/bootstrap.min.css" rel="stylesheet">
8 <link href="css/bootstrap-theme.min.css" rel="stylesheet">
9 <style> .container{margin-top:100px} </style>
10 </head>
```

El desarrollo de esta sección se hizo mediante *HTML*, en el cual se personalizó el proyecto, creando un nombre de usuario y contraseña como en la Fig. 3.8 para tener acceso a un menú principal como en la Fig. 3.9, en donde se gestionará la información inherente a clientes, los planes de reproducción y la publicidad o contenido que se desea reproducir.



The image shows a login interface for a system titled "Sistema de Información Pi3". The interface consists of a header with the system name and a central login form. The form includes labels for "User" and "Password", corresponding input fields for "Email" and "Password", and a "Sign in" button.

Figura 3.8. Vista del Ingreso al Sistema

El formulario de ingreso se hizo mediante el código del Cuadro 3.5

Cuadro 3.5. Formulario de Ingreso

```

1 <body role="document">
2 <div class="jumbotron text-center">
3 <h1>Sistema de Información Pi3</h1>
4 </div>
5 <form class="form-horizontal" action="login.php" method="post">
6 <label for="inputEmail3" class="col-sm-4 control-label ">User</
  label>
7 <div class="col-sm-4">
8 <input type="email" class="form-control" name="email" id="
  inputEmail3" required>
9 </div>
10 <label for="inputPassword3" class="col-sm-4 control-label ">
  Password</label>
11 <div class="col-sm-4">
12 <input type="password" class="form-control" name="password" id="
  inputPassword3" required>
13 </div>
14 <div class="col-sm-offset-4 col-sm-6">
15 <button type="submit" class="btn-default">Sign in</button>
16 </div>
17 </form>
18 </body>

```

Por su parte la verificación de cada dato de ingreso se gestionó mediante el código del Cuadro 3.6 que a través del método `post` hace uso del hipervínculo `login.php`, en donde se hizo uso de la función `password_hash()` para lograr que nuestra contraseña sea mas difícil de descifrar mediante algún algoritmo, y darle un poco de seguridad. Es por ello que en el código se ve `'$2y$10$9Q/zHvx/ybqqvDZBJbE7tevrLeYYiVrECR/voIOv00G8bgp5ZQAE.'` lo cual indica el uso de la función anterior para resguardar la contraseña de acceso.

Cuadro 3.6. Comprobación del Formulario de Ingreso en PHP

```

1 <?php
2 $email = $_POST['email'];
3 $password = $_POST['password'];
4 if (($email == 'pi3@ula.com') && (password_verify($password,
5     '$2y$10$9Q/zHvx/ybqqvDZBJbE7tevrLeYYiVrECR/voIOv00G8bgp5ZQAE.'))
6     {
7     $_SESSION['email'] = $email;
8     header("HTTP/1.1 302 Moved Temporarily");
9     header("Location: principal.php");
10    }
11    else
12    {
13    echo 'El email o password es incorrecto,
14    <a href="index.php">vuelva a intentarlo</a>.<br/>';
15    }
16    ?>

```



Figura 3.9. Vista del Menú Principal

En cuanto a los clientes, se creó un formulario sencillo para poder cargar información a la base de datos como se puede ver en la Fig. 3.10 la cual se ejecuta mediante un código

en *HTML*. De la misma manera con los planes que se quieren ofertar al usuario, es decir se pueden ofrecer planes de duración predeterminada por el usuario.



Figura 3.10. Vista del Menú de Clientes

Por último la parte de publicidad que mediante el formulario de la Fig. 3.11, ejecutado mediante un código *HTML* y *PHP* del Cuadro 3.7, nos permite ingresar el contenido en formato *MP4* a la base de datos así como el cliente relacionado con dicho contenido, y la duración del mismo.



Figura 3.11. Vista del Formulario de Publicidad

Cuadro 3.7. Formulario de Ingreso de Publicidad en HTML

```

1 <form method="post" action="publicidad_agregar_subir.php"
  enctype="multipart/form-data">
2 <legend class="text-left header">Seleccione la publicidad:</
  legend>
3 <div class="col-md-8">
4   <input title="Introduzca plan" name="plan" type="text" class="
  form-control" required />
5 </div>
6 <div class="col-md-8">
7   <input title="Introduzca inicio" id="fname" name="inicio" type=
  "text" class="form-control required />
8 </div>
9 <div class="col-md-8">
10  <input title="Introduzca fin" id="fname" name="fin" type="text"
  class="form-control" required />
11 </div>
12 <div class="col-md-8">
13  <input title="nro cliente" id="fname" name="client" type="text"
  class="form-control" required />
14 </div>
15 <div class="col-md-8">
16  <input type="file" name="fileToUpload" id="fileToUpload" title=
  "Examinar" />
17 </div>
18 <div class="col-md-12 text-center">
19  <button type="submit" class="btn btn-primary btn-lg">Subir
  Publicidad</button>
20 </div>
21 </form>

```

El código en PHP a través del cual se carga la información al servidor se describe mediante el Cuadro 3.8, donde se indica a la carpeta a donde se va a subir la información, el tamaño del archivo, el tipo de archivo, entre otros.

Cuadro 3.8. Ejecución del Formulario de Ingreso de Publicidad en PHP

```

1  <?php
2  if (isset($_POST['plan']))
3      $plan = $_POST ["plan"];
4  if (isset($_POST['inicio']))
5      $inicio = $_POST ["inicio"];
6  if (isset($_POST['fin']))
7      $fin = $_POST ["fin"];
8  if (isset($_POST['client']))
9      $client = $_POST ["client"];
10 $target_dir = "UPLOADS/";
11 $target_file = $target_dir . basename($_FILES["fileToUpload"]["
    name"]);
12 $uploadOk = 1;
13 $imageFileType = pathinfo($target_file, PATHINFO_EXTENSION);
14 if ($_FILES["fileToUpload"]["size"] > 10000000) {
15     $_SESSION['mensaje'] = "Sorry, your file is too large.";
16     $uploadOk = 0;}
17 if ($imageFileType != "mp4" && $imageFileType != "png" &) {
18     $_SESSION['mensaje'] = "Sorry, only PNG, MP4
19     PNG & GIF files are allowed.";
20     $uploadOk = 0;}
21 if ($uploadOk == 0)
22     {
23     $_SESSION['mensaje'] = "Sorry, your file was not uploaded.";}
24 else
25     {
26     if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
27         $target_file))
28     {
29     $ruta = '/var/www/html/PHP/' . $target_file;
30     $sql = "INSERT INTO publicidad (plan_id, publi_ruta,
31     publi_inicio, publi_fin, client_id)
32     VALUES ( '$plan', '$ruta', '$inicio', '$fin', '$client')";
33     }
34 }
?>

```

De la misma manera fueron creadas las demás secciones del servidor, como el apartado de planes y a su vez cada opción de modificar o actualizar datos. También las consultas a

través del servidor en donde se accedía a la base de datos como se observa en la Fig. 3.12



[Sistema de Gestión de Publicidad]

Agregar Actualizar Listar Buscar Cliente

Lista de Publicidades

Identificación de la Publicidad	Tipo de plan (meses)	Ruta de la publicidad	Fecha de Inicio de Publicidad	Fecha de Inicio de Publicidad	Identificación del Cliente Asociado
1	3	/var/www/html/PHP/UPLOADS/v005.mp4	2017-06-01 00:00:00	2017-10-01 00:00:00	1
2	1	/var/www/html/PHP/UPLOADS/v006.mp4	2017-07-01 00:00:00	2017-08-01 00:00:00	2
3	2	/var/www/html/PHP/UPLOADS/v001.mp4	2017-07-01 00:00:00	2017-09-01 00:00:00	3
4	3	/var/www/html/PHP/UPLOADS/v003.mp4	2017-07-01 00:00:00	2017-10-01 00:00:00	4
5	1	/var/www/html/PHP/UPLOADS/v009.mp4	2017-07-01 00:00:00	2017-08-01 00:00:00	1
6	2	/var/www/html/PHP/UPLOADS/ULA-logo-279x300.png	2017-07-01 00:00:00	2017-08-01 00:00:00	2
7	2	/var/www/html/PHP/UPLOADS/Screenshot from 2017-05-22 17:17:1	2017-06-01 00:00:00	2017-09-01 00:00:00	1

Figura 3.12. Vista del Formulario de Publicidad

3.2.3 Programa de Reproducción en Lenguaje C

En esta etapa procedemos a la creación del programa de reproducción, en donde se delimitará nuestra forma de ejecución del mismo y se configurará su funcionamiento. Este programa consiste en hacer funcionar la pantalla de forma automática, para ello se va a definir un ciclo de trabajo, es decir, el funcionamiento de la pantalla se hará en base a doce (12) horas continuas y en este caso se va a ofrecer al usuario un máximo de sesenta (60) reproducciones, de veinte (20) segundos de duración cada una, es decir, en (12) horas se logrará reproducir 2160 *spot* publicitarios de veinte (20) segundos. Lo cual nos permite que solo podamos tener por día un máximo de treinta y seis (36) clientes por día.

La programación de esta etapa se lleva a cabo haciendo uso del editor de texto *Vim* y del compilador interno *gcc* de *Linux*. Por su parte el uso de las librerías del Cuadro 3.9

Cuadro 3.9. Librerías de C

```
1 #include <my_global.h>
2 #include <mysql.h>
3 #include <string.h>
4 #include <time.h>
5 #include "i2c.h"
6 #include "rtc.h"
```

El programa se busca la creación de un vector de treinta y seis (36) posiciones el cual se ejecute de forma seguida sesenta (60) veces y con eso se garantice al usuario que su publicidad será promocionada durante 12 horas. La forma en como se llena el vector asigna a cada posición una ruta del destino del archivo que se desea reproducir, generando así una única posición para cada ruta y permitiendo la alternancia de cada contenido. Para que el programa pueda obtener la ruta debe hacer una consulta a la base de datos, que es donde se ha almacenado en una cadena de caracteres cada una de las direcciones del archivo de destino que se agregó mediante el servidor web a la base de datos. Al obtener la ruta, el programa puede contar cuantas publicidades tiene destinadas para la fecha, ya que cada consulta tiene un identificador de fecha, permitiendo que se programen a futuro las publicidades, es decir, cada día en el que funcione la pantalla el programa verificará la fecha y seleccionará las publicidades que estén dentro de ese día.

El código descrito a continuación se encuentra dentro un bucle principal como el del Cuadro 3.10

Cuadro 3.10. Bucle Principal

```
1 int main()
2 {
3
4 }
```

En este programa también será necesaria la conexión a la base de datos y todo el código

que describe el funcionamiento se muestra en el siguiente Cuadro 3.11

Cuadro 3.11. Conexión a la Base de Datos

```
1  MYSQL *conn;
2  MYSQL_RES *res;
3  MYSQL_ROW row;
4  char server[11] = "localhost";
5  char user[11] = "root";
6  char password[11] = "heisemberg";
7  char database[11] = "myDB";
8  conn = mysql_init(NULL);
9  if (mysql_real_connect(conn, server, user, password,
10 database, 0, NULL, 0) == NULL)
11 {
12 fprintf(stderr, "%s\n", mysql_error(conn));
13 exit(1);
14 }
```

Es momento de hablar del control de la hora y de la fecha, ya que nuestro programa de reproducción se rige estrictamente por consultas específicas a la base de datos, es decir, al cargar algún contenido que se quiera reproducir, junto a el se adjunta en el formulario de cargar la publicidad, una fecha de inicio y una fecha de finalización de las reproducciones. Entonces es donde necesitamos complementar la Raspberry Pi con el modulo RTC, ya que la Raspberry Pi por si sola no dispone de un modulo de Reloj de Tiempo Real, siendo su única fuente de actualización de hora y fecha dependiente la conexión a alguna red que proporcione acceso a internet. Para ello se decidió agregar el modulo RTC al proyecto y así poder tener de manera independiente un control de la hora para cada consulta que se necesite.

Para la configuración del modulo RTC Arduino DS3231, se hizo uso de las herramientas de Linux `rtc-tools`, con las cuales se logro su puesta en funcionamiento sin ningún problema. Por su parte las rutinas utilizadas para el funcionamiento del Reloj de Tiempo Real fueron creadas a partir de códigos usados en microprocesadores, pero con algunas

variaciones de configuración. En el programa principal la consulta al RTC será una de las primeras cosas que se ejecuten para lograr tener siempre configurada la fecha y hora de nuestra consulta, destacan el uso de las sentencias que se muestran en el Cuadro 3.12. Donde `rtc_init()` representa la inicialización del módulo, `rtc_get_time` hará una consulta de la hora y `rtc_get_date` realizará una consulta de la fecha.

Cuadro 3.12. Sentencias para Consulta del RTC

```

1  rtc_init();
2  rtc_get_time(&h, &m, &s);
3  rtc_get_date(&D, &M, &Y, &W);

```

Cuadro 3.13. Código Para Reproducción en Caso de Falla Eléctrica

```

1  rtc_get_time(&h, &m, &s);
2  v = (h * 60 + m) * 60;
3  if(v < 28800)
4  continue;
5  if(v != 28800){
6    vr = v - 28800;
7    vr = vr / 720;
8    vr = 60 - vr;
9  }
10 else
11  vr = 60;

```

Es importante destacar que se ha previsto en el programa de reproducción en C, una rutina que permita recuperar las reproducciones a cualquier hora en el caso de alguna falla eléctrica. Y su funcionamiento parte de lograr contar cada segundo de reproducción desde que la pantalla empiece a reproducir contenido hasta que cumpla su ciclo de doce (12) horas de funcionamiento. De esta forma al momento de consultar la hora, en caso de que no sea la hora fijada para su puesta en funcionamiento en una jornada normal y sin interrupciones, la pantalla logre identificar en que segundo de la reproducción se encuentra

y así recalcular los segundos restantes para culminar la jornada. El código del Cuadro 3.13 muestra la rutina empleada para la ejecución anterior descrita.

Para realizar una consulta a la base de datos haremos uso del Cuadro 3.14, donde el sistema a su vez será capaz de identificar la fecha actual mediante las sentencias **SELECT**, **FROM**, **NOW()**, **BETWEEN**, **AND** para poder comprobar que el contenido a reproducir y también delimitar el número de clientes por día, haciendo uso de la información suministrada a la base de datos, la cual permita al operador hacer todas las consideraciones pertinentes y dar con el funcionamiento adecuando del prototipo.

Cuadro 3.14. Consulta a la Base de Datos

```

1  if (mysql_query(conn,"SELECT publi_ruta FROM publi WHERE now()
    BETWEEN publi_inicio AND publi_fin"))
2  {
3    fprintf(stderr, "%s\n", mysql_error(conn));
4    exit(1);
5  }
```

La siguiente etapa representa el llenado del vector de treinta y seis (36) posiciones, en donde luego de realizar la consulta a la base de datos, determinaremos mediante `mysql_num_rows`, el número de filas que poseen en su casilla alguna información, en nuestro caso la ruta del archivo que deseamos reproducir, para luego hacer un cálculo sencillo en donde se pueda distribuir toda la publicidad de manera uniforme, cuidando los espacios en blanco o vacíos en caso de que no se llene completamente el vector, y asignando a este espacio una publicidad predeterminada, que en nuestro caso será algo alusivo al proyecto. El Cuadro 3.15 muestra en detalle el funcionamiento de lo descrito anteriormente. Como también la reproducción de dicho vector por sesenta (60) veces y así cumplir con la transmisión de 2160 spot publicitarios.

Cuadro 3.15. Llenado del Vector de 36 posiciones, Gestión de Filas Vacías y Reproducción de 60 Veces

```

1  i = 0;
2  res = mysql_store_result(conn);
3  n = mysql_num_rows(res);
4  a = (36 - n) / n;
5  printf("n:%d a:%d \n", n, a);
6  j = 0;
7  for(i = 0; i < n; i++){
8      row = mysql_fetch_row(res);
9      strcpy(publicidad[j], "/var/www/html/PHP/UPLOADS/");
10     strcat(publicidad[j], (const char *)row[0]);
11     j++;
12     for(b = 0; b < a; b++){
13         strcpy(publicidad[j], "/home/pi/PHP/video_publicitario.jpg");
14         j++;
15     }
16 }
17 printf("%d\n", j);
18 if(j < 36){
19     for(i = j; i < 36; i++){
20         strcpy(publicidad[i], "/home/pi/PHP/video_publicitario.jpg");
21     }
22 }

```

La reproducción del vector vendrá dada por la función *system* como se aprecia en el Cuadro 3.16

Cuadro 3.16. Función de Reproducción de Contenido

```

1 strcpy(cadena, "cvlc -V X11 -f -play-and-exit -no-video-title ");
2 for(i = 0; i < vr; i++)
3 {
4     for(j = 0; j < 36; j++)
5     {
6         strcat(cadena, publicidad[j]);
7         strcat(cadena, " ");
8     }
9 }
10 system(cadena);

```

Culminamos este capítulo con la comprobación de el código de programación anterior descrito, el cual lo pudimos compilar y verificar su funcionamiento haciendo uso del compilador `gcc` como se muestra en el Cuadro 3.17 en el cual se ejecuta el archivo llamado `program_repro.c`

Cuadro 3.17. Compilar `program_repro.c`

```

~$ gcc -o program_repro -O3 -g -Wall -I/usr/local/include -I/usr/include/mysql -Winline -
pipe -L/usr/local/lib -lmysqlclient -lpthread program_repro.c

```

Y se ejecuta mediante el Cuadro 3.18

Cuadro 3.18. Ejecución `program_repro.c`

```

~$ ./program_repro

```

CAPÍTULO 4

ANÁLISIS DE RESULTADOS

En este capítulo se expondrán los resultados obtenidos en la simulación e implementación y se describirá el funcionamiento final del prototipo junto a todas las partes que lo conforman.

4.1 Análisis de Resultados

En el desarrollo de este proyecto fue fundamental los conocimientos y destrezas de programación, ya que es un trabajo mayormente lógico el que se requiere para lograr la integración de cada una de sus partes. Si bien no representó ninguna limitación, es esencial conocer sobre sistemas operativos *Linux*, puesto que la *Raspberry Pi 3* se maneja en ese entorno, y representa una potente herramienta el uso del *software libre*, debido a su amplia configuración y adaptación al proyecto, es por ello que pudimos usar en su totalidad la configuración necesaria para la elaboración del prototipo.

La creación de la base de datos fue una pieza determinante en el desarrollo del prototipo funcional, ya que se logró almacenar los datos necesarios para darle soporte a la información que a futuro se iba a necesitar. Siendo de gran ayuda conocer las sentencias *SQL* que permitirían mediante un código *PHP* darle solución a problemas de conexión, elaboración y consulta de datos. Brindando a su vez flexibilidad para ingresar cualquier tipo de dato, como cadenas de caracteres o archivos en formatos como *MP4* y *PNG*.

Siendo el *servidor web* el puente entre la base de datos y el *panel LCD*, se pudo dar un aporte extra a la creación de un entorno gráfico para la gestión de todos los datos,

de manera de acceder a un sitio de forma remota, donde cualquier usuario pueda agregar y editar de forma sencilla la base de datos antes descrita. Su programación no fue una limitación, mas aún, bajo la premisa de herramientas de desarrollo libres, se utilizó las plantillas de *Bootstrap* para darle rápidamente forma al servidor, que no es mas que una hoja de estilos. En esta etapa se fusionaron los códigos *PHP* para gestionar cualquier información de modificación e ingreso a la base de datos y poder integrar de manera efectiva todo el funcionamiento gráfico para una mejor adaptación al usuario.

Por su parte, era necesario un programa, que haciendo uso de otras funciones, gestionara la información alojada en la base de datos, es aquí donde nace el programa de reproducción en *C*. Que básicamente hace uso de funciones y librerías esenciales, primero de conexión a la base de datos y posteriormente de ejecución de un algoritmo para llenar un vector con una cadena de caracteres y posteriormente reproducirlo. Donde además se hace uso, de las funciones *System* para poder indicarle a la *Raspberry Pi 3* que a través de su reproductor predeterminado ejecutara las acciones pertinentes.

La culminación de esta etapa viene marcada por el uso del modulo RTC, el cual nos proporciona el elemento fundamental para poder realizar cada consulta a la base de datos en una fecha y hora específica, ya que cada uno de los contenidos que se cargaron serian consultados en un orden y fecha determinada. Su configuración y puesta en funcionamiento represento tarea sencilla, ya que la *Raspberry Pi* a través de sus pines multipropósito logra conectarse facilmente al modulo.

Para el total cumplimiento de los objetivos, en cuanto a la autonomía de la pantalla inteligente, se logro la creación y puesta en funcionamiento de un algoritmo con el cual la pantalla pueda ser encendida a cualquier hora y logre calcular en que instante específico debe continuar su reproducción, por alguna falla de electricidad o algún otro problema técnico que se pueda presentar. Sumado a esto, se pudo tomar previsión en cuanto a la forma en que se logra cargar el contenido que se desea reproducir, es decir, debido a las posibilidades de conexión inalámbrica de la *Raspberry Pi*, la información se podía hacer

llegar si la pantalla estaba conectada a una red interna del lugar en donde se usara, con solo tener su dirección IP, lo cual reflejaba ser algo bastante simple. Pero si se desconoce que en el lugar en el que va a funcionar la pantalla dispone o no de una red inalámbrica, se logro configurar la tarjeta Raspberry Pi para que funcionara como una red independiente a la cual se le pueda acceder y conectar como un router cualquiera y de esta forma lograr cargar información en el rango o alcance que la Raspberry Pi disponga.

Su funcionamiento es altamente configurable, pudiendo predeterminedar mas horas de trabajo, cantidad de contenido, duración de cada spot publicitario, entre muchas otras funciones. Por lo que es adaptable a cualquier ambiente en donde se requiera una publicidad o algún tipo de información en específico.

4.2 Prototipo Final

Siendo la pantalla LCD, la Raspberry Pi 3 y el controlador PCB800099-RTD2662 componentes electrónicos de un tamaño pequeño, se lograron integrar en una carcasa que simplifique su tamaño final. En donde resalta que solo tendremos una conexión a la alimentación de corriente continua. El resto de conexiones se lograrán de forma inalámbrica a través de la red. Comparado con una pantalla publicitaria real se logra simplificar en su totalidad el tamaño, puesto que la pantalla LCD es mucho mas liviana que un televisor, a su vez la tarjeta *Raspberry Pi* es muy portable, de hecho en cuanto a tamaño y potencia supera facilmente a una computadora convencional. Un ejemplo de esto lo podemos observar en la Figura 4.1 en donde se muestra una pantalla publicitaria en un centro comercial el cual refleja ser de un tamaño y peso considerables.

Por su parte el funcionamiento lógico no representó ninguna falla, en principio se logró ejecutar cada parte del código sin ningún problema, logrando configurar cualquier etapa del proyecto sin mayor dificultad. Fue necesaria la instalación de *Apache* y *MySQL* para poder realizar todas las pruebas, y realizar algunas modificaciones en el archivo `php.ini`

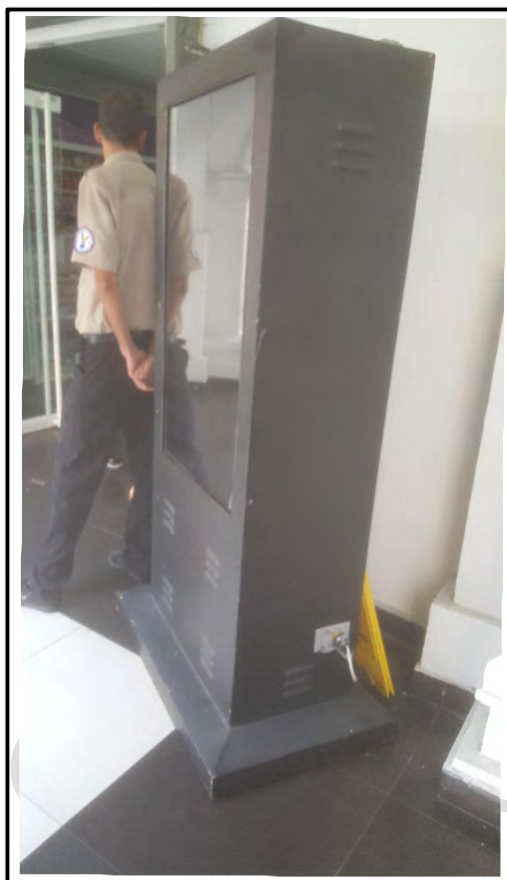


Figura 4.1. Pantalla Convencional

para lograr cargar archivos de mayor tamaño al servidor.

La reproducción de contenido se logro haciendo uso del programa VLC, en el cual se encontró una limitación en cuanto a la resolución del contenido a reproducir, ya que si los videos publicitarios tenían diferente resolución, haciendo un salto de pantalla el cual era algo molesto, pero con un ajuste en su configuración se limito al reproductor a que no hiciera cambios en la reproducción del contenido. Por su parte la Raspberry Pi puede funcionar con o sin interfaz gráfica, es decir en modo texto. El VLC es un programa de reproducción que presento alguna falla al momento de ejecutarlo sin interfaz, por lo que se configuro para que se ejecutara desde el inicio pero luego de que la Raspberry Pi cargara

su sistema operativo con toda su interfaz.

Haciendo uso del RTC, se logra configurar el funcionamiento de la pantalla para que trabaje de manera continua por lapsos de tiempo deseados, en nuestro caso doce (12) horas. La velocidad con la que se carga a través del servidor web los videos en la base de datos se hace de manera instantánea, por lo que el tiempo de carga de archivos no representa ninguna limitación.

Se puede hacer un análisis económico rápido y comprender el potencial de este prototipo, ya que se logra ahorrar hasta un 60 %, por ejemplo: si hacemos la comparativa de un sistema actual de pantalla publicitaria los costos aproximados seria del televisor de 40 Pulgadas seria de 260\$, una computadora convencional cuesta 100\$ y en los cables para conexiones serian casi 40\$. Lo que da un total de 400\$. En contra parte la adquisición de una tarjeta Raspberry Pi 3 tiene un costo de 30\$, sumado a una pantalla LCD de 40 Pulgadas son 100\$ adicionales, la interfaz electrónica de comunicación cuesta 25\$ y un gasto cercano a los 20\$ para la construcción de una carcasa. Lo que da un total de 175\$.

Es decir implica un ahorro de 225\$, lo que representa un monto bastante significativo que se debe considerar para la ejecución de algún proyecto.

Por ultimo el prototipo final se muestra en las Figuras 4.2 y 4.3, en el cual, la carcasa definitiva se logro diseñar con aluminio y algunas otras partes metálicas recicladas de monitores de escritorio.



Figura 4.2. Carcasa del Prototipo de Pantalla Inteligente Final 1



Figura 4.3. Carcasa del Prototipo de Pantalla Inteligente Final 2

CONCLUSIONES

- El uso de la Raspberry Pi 3 representa un gran avance en el desarrollo del proyecto debido a todas las ventajas de programación y la gran adaptación que esto brinda a cualquier proyecto. Permitiendo que se puedan ajustar a las necesidades reales del usuario y operador
- Se logró desarrollar el prototipo del sistema físico, el cual de manera simplificada cumplía con la principal propuesta de ergonomía y simplicidad, que mediante cada una de sus partes, brinda una sencilla instalación y posterior uso.
- Gracias al uso del software libre se logró estudiar cada parte del proyecto de forma vigorosa y eficaz, ya que existe una gran comunidad de desarrolladores dispuestos a contribuir con la solución de cualquier problema, pudiendo así, lograr dominar por completo las funcionalidades de la Raspberry Pi 3 y la programación de la parte lógica para el cumplimiento de objetivos.
- Fue determinante lograr familiarizarse con los lenguajes de programación requeridos. Esto llevó a poder desarrollar la parte lógica y gráfica con mayor destreza para integrar cada parte del proyecto. A su vez la gran cantidad de información digital disponible sobre el tema, hace que sea mas sencilla la solución de cualquier inconveniente. El uso de un IDE o entorno de desarrollo integrado (*Integrated Development Environment*) facilitó la elaboración de código, con sus funciones de auto completado y sintaxis.
- Resulta muy factible desde el punto de vista económico la implementación de este tipo de proyectos, puesto que se logra ahorrar mas de un 60 % del presupuesto, com-

parándolo con la adquisición de televisores, computadoras, entre otros dispositivos, que se suelen utilizar para la elaboración de una pantalla publicitaria convencional.

- El acceso de forma remota, que nos permite la Raspberry Pi 3 mediante su funcionalidad de conexión inalámbrica a la pantalla LCD, es sin duda muy relevante, puesto que desde cualquier lugar podemos ingresar, consultar y modificar la base de datos. Volviendo el prototipo ergonómico y flexible

www.bdigital.ula.ve

RECOMENDACIONES

Podemos destacar mejoras que pueden ser funcionales a este prototipo como la inclusión de nuevas tecnologías y sustitución de algunos componentes alternativos:

- La sustitución de la Raspberry Pi 3 por una nueva computadora de placa reducida de la misma fundación pero un modelo diferente, la Raspberry Pi Zero Wireless. La cual representa un tercio del costo de la Raspberry Pi 3, lo que sin duda representa un mayor ahorro

En el aspecto académico a fin de promover mas propuestas de este tipo, para lograr el cumplimiento de objetivos en materias de pregrado en la escuela de Ingeniería Eléctrica, y lograr estimular al estudiante a su futura experiencia laboral, es necesaria la inclusión de las siguientes recomendaciones:

- Rescatar la importancia de la programación digital para al elaboración de proyectos funcionales, que puedan estar ligados a pequeños emprendedores, competencias tecnológicas en diferentes universidades nacionales y cumplimiento de objetivos en materias afines a la carrera
- Implementar el uso de herramientas de software libre en los laboratorios de escuela, a fin de promover su gran potencial y que así se contribuya con el desarrollo de proyectos a cualquier nivel.
- Estimular el desarrollo de prototipos funcionales para la exploración de nuevas tecnologías.

Bibliografía

- [1] Portal de Tecnología. (2014). Definición de Computadora [En línea]. Disponible en: <https://www.mastermagazine.info/termino/4384.php>
- [2] Diccionario de la lengua Española. (2002, Agosto 10). Hardward [En línea]. Disponible en: <http://dle.rae.es/?id=K1Wwkf7>
- [3] F. Da Cruz. (2001, Enero). Computer Generations [En línea]. Disponible en: <http://www.columbia.edu/cu/computinghistory/generations.html>
- [4] Raspberry Pi Foundation. (2011, Octubre 06). FAQs [En línea]. Disponible en: <https://www.raspberrypi.org/help/faqs/>
- [5] R. Cellan-Jones. (2011, Mayo 5). A 15 pound computer to inspire young programmers [En línea]. Disponible en: http://www.bbc.co.uk/blogs/thereporters/rorycellanjones/2011/05/a_15_computer_to_inspire_young.html
- [6] R. Lawler. (2012, Febrero 29). Raspberry Pi credit-card sized Linux PCs are on sale now, \$25 Model A gets a RAM bump [En línea]. Disponible en: <https://www.engadget.com/2012/02/29/raspberry-pi-credit-card-sized-linux-pcs-are-on-sale-now-25-mo/>
- [7] L. Upton. (2013, Febrero 04). Today RB Foundation [En línea]. Disponible en: <https://www.raspberrypi.org/blog/model-a-now->
- [8] Universidad Rey Juan Carlos. (2011, Marzo 06). Buses del Sistema [En línea]. Disponible en: http://dac.escet.urjc.es/docencia/AC/Buses_mod.pdf

- [9] A. Cañas. (2007, Junio 15). Entrada/Salida [En línea]. Disponible en: https://swad.ugr.es/tmp/descarga/1%20Entrada-Salida/1%20Transparencias/Transparencias_E-S_2010-2011.pdf
- [10] Universidad de Azuay. (2011, Marzo 09). Equipos de comunicaciones [En línea]. volume(issue). Disponible en: http://www.uazuay.edu.ec/estudios/sistemas/teleproceso/apuntes_1/equipos_com.html
- [11] Diccionario de la lengua española. (2005). Software [En línea]. Disponible en: <http://www.wordreference.com/definicion/software>
- [12] N. Owano. (2012, julio 18). Raspberry Pi gets customized OS called Raspbian [En línea]. Disponible en: <https://phys.org/news/2012-07-raspberry-pi-customized-os-raspbian.html>
- [13] L. Upton. (2013, junio 03). INTRODUCING THE NEW OUT OF BOX SOFTWARE (NOOBS) [En línea]. Disponible en: <https://www.raspberrypi.org/blog/introducing-noobs/>
- [14] M. Lutz. (2010). Learning Python, Fourth Edition [En línea]. Disponible en: <http://oreilly.com/catalog/9780596158071>
- [15] J. Chacón. "Diseño e Implementación de un Sistema de Domótica Usando una Raspberry PI B+", Trabajo de Grado, Universidad de Los Andes, Octubre 2015.
- [16] L. R. Araujo. (2014). Clases de Microprocesadores [En línea]. Disponible en: <http://mp.luis-araujo.net>
- [17] M. Chapple. (17, Octubre 2015). SQL Fundamentals [En línea]. Disponible en: <http://databases.about.com/od/sql/a/sqlfundamentals.htm>
- [18] PHP Group. (2013). Manual de PHP [En línea]. Disponible en: <http://ch1.php.net/manual/es/index.php>

- [19] Z. Urlocker. (2005, Diciembre 13). Google Runs MySQL [En línea]. Disponible en: http://zurlocker.typepad.com/theopenforce/2005/12/googles_use_of_.html
- [20] Netcraft. (2009, Febrero). Web Server Survey [En línea]. Disponible en: <https://news.netcraft.com/>
- [21] VideoLAN Team (2007, Julio 30). Intellectual Property [En línea]. Disponible en: https://wiki.videolan.org/Intellectual_Properties
- [22] Linux.org (2015, Noviembre 25). Linux Online - About the Linux Operating System [En línea]. Disponible en: <http://web.archive.org/web/http://www.linux.org/info>
- [23] Raspberry Foundation (2017, Julio 13). Raspberry Pi 3 [Imagen En línea]. Disponible en: <https://https://www.raspberrypi.org/blog>
- [24] Hardward NET. (2017, Julio 28). Partes de una Laptop [Imagen En línea]. Disponible en: <https://https://www.hardwardnet.co.seeders/parts>
- [25] Hardward NET. (2017, Julio 29). Partes de un Panel LCD [Imagen En línea]. Disponible en: <https://https://www.LCDcomponents.org/sites/parts/lcd>
- [26] RaspBerry Source. (2017, Julio 19). GPIO Headers RB Pi 3 [Imagen En línea]. Disponible en: <https://https://www.sourceraspberry.se.forms/info/gpio/pi3>
- [27] Maxim Integrated. (2017, Julio 29). Selecting a Backup Source for Real-Time Clocks [En línea]. Disponible en: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/3816>