

**UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR**

**FACULTAD DE INGENIERIA Y GESTIÓN  
ESCUELA PROFESIONAL DE INGENIERIA ELECTRÓNICA Y  
TELECOMUNICACIONES**



**“DISEÑO DE UN SISTEMA DE ADMINISTRACIÓN Y  
AUTOMATIZACIÓN DE SURTIDORES DE COMBUSTIBLE LIQUIDO  
CON RASPBERRY PI3 PARA LA EMPRESA EMPEX”**

**TRABAJO DE SUFICIENCIA PROFESIONAL**  
Para optar el Título Profesional de

**INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**

**PRESENTADO POR EL BACHILLER**  
PASSUNI CORI, AKIN ABDEL

**Villa El Salvador**

**2017**

## DEDICATORIA

El presente trabajo está dedicado a mis padres que son la base de mi formación y a mi familia que siempre estuvo dándome su apoyo.

## AGRADECIMIENTO

Agradezco a mi asesores y a mis profesores de cursos que me apoyaron en la elaboración del presente trabajo y a mis primas.

## INDICE

1	CAPITULO I: PLANTEAMIENTO DEL PROBLEMA .....	12
1.1	DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA .....	12
1.2	JUSTIFICACION DEL PROBLEMA .....	17
1.3	DELIMITACIÓN DE LA INVESTIGACIÓN .....	18
1.3.1	Teórica .....	18
1.3.2	Espacial .....	18
1.3.3	Temporal .....	18
1.4	FORMULACION DEL PROBLEMA .....	18
1.4.1	PROBLEMA GENERAL .....	18
1.4.2	PROBLEMAS ESPECÍFICOS .....	18
1.5	OBJETIVOS.....	19
1.5.1	OBJETIVO GENERAL.....	19
1.5.2	OBJETIVOS ESPECÍFICOS.....	19
2	CAPITULO II: MARCO TEORICO.....	20
2.1	ANTECEDENTES .....	20
2.1.1	ANTECEDENTES NACIONALES .....	20
2.1.2	Antecedentes Internacionales.....	22
2.2	BASES TEÓRICAS .....	25
2.2.1	AUTOMATIZACIÓN .....	25
2.2.2	Raspberry Pi .....	30
2.2.3	ADMINISTRACIÓN.....	34
2.2.4	Surtidores de Combustibles .....	45
2.3	MARCO CONCEPTUAL.....	50

2.3.1	Micro controlador .....	50
2.3.2	Micro controlador ARM .....	50
2.3.3	Pines GPIO.....	51
2.3.4	Sensores .....	51
2.3.5	Encoder .....	51
2.3.6	Base de datos.....	52
2.3.7	Interfaz .....	52
2.3.8	RFID .....	53
2.3.9	Arduino Mega .....	53
2.3.10	Señales digitales.....	53
2.3.11	Sensor Fotoeléctrico .....	53
2.3.12	PIN .....	54
2.3.13	Librerías .....	54
2.3.14	Monitor Serial .....	54
2.3.15	Baudios .....	54
2.3.16	Seteado.....	55
2.3.17	Web.....	55
2.3.18	Comunicación Serial.....	55
2.3.19	Python .....	55
2.3.20	Pulsos.....	56
2.3.21	MySQL .....	56
2.3.22	PHPMYAdmin.....	56
3	CAPITULO III: DESARROLLO DE LA METODOLOGIA.....	57
3.1	DESCRIPCIÓN DEL PROYECTO .....	57
3.1.1	Adquisición de datos.....	59
3.1.2	Envío de datos y procesamiento de datos .....	70

3.2	Diseño de conexión eléctrica y empotramiento del sistema.....	77
3.2.1	Conexiones eléctricas.....	77
3.2.2	Estructura mecánica .....	80
3.2.3	Pruebas y Resultados .....	83
3.3	Análisis de costos de prototipo.....	88
3.4	Consumo de energía del sistema .....	88
3.5	Diagrama de gantt .....	90
4	Conclusiones.....	92
5	Recomendaciones .....	93
6	Bibliografía.....	94
7	Anexos .....	96

## LISTADO DE FIGURAS

<b>Figura 1.1.</b> Vehículo llegando al gasocentro.....	14
<b>Figura 1.2.</b> Vehículo de la Policía llenando combustible .....	14
<b>Figura 1.3,</b> Efectivo policial pagando con tarjeta multiflota.....	15
<b>Figura 1.4,</b> Vehículo particular llenando combustible.....	15
<b>Figura 1.5.</b> Efectivo policial recibiendo dinero del gasocentro .....	16
<b>Figura 2.1</b> Raspberry Pi3 tipo B, este es el dispositivo tecnológico adquirido para realizar todas las pruebas del proyecto, ahí se puede observar sus puertos y su forma física.....	30
<b>Figura 2.2</b> Raspberry pi y sus puertos. Descripción de los puertos de la tarjeta Raspberry Pi3 Modelo B .....	32
<b>Figura 2.3</b> Imagen de la instalación del S.O. del Raspberry Pi. Aquí se muestra la pantalla de instalación del S.O. Y se observan la selección de los diferentes sistemas disponibles para la tarjeta. Adaptado de la página Principal del Raspberry distribución .....	34
<b>Figura 2.4</b> Imagen de Tokheim 1990. Aquí se muestra el Tokleim del año 1990.....	48
<b>Figura 2.5</b> Imagen de Tokheim 1980. Aquí se muestra el Tokleim del año 1980.....	49
<b>Figura 2.6</b> Imagen de Tokheim 2000. Aquí se muestra el Tokleim del año 2000.....	49
<b>Figura 2.7</b> Imagen del Encoder. Aquí se muestra el Código de disco. ....	52
<b>Figura 3.1:</b> Diagrama de sistema de administración y automatización para surtidores de combustibles líquidos.....	58
<b>Figura 3.2:</b> Diagrama de adquisición de datos del surtidor y del RFID hacia el ARDUINO MEGA.....	59

<b>Figura 3.3:</b> Conexiones del Encoder con el Arduino Mega.....	60
<b>Figura 3.4:</b> Conexiones del módulo RDIF RC522 con Arduino Mega .....	62
<b>Figura 3.5:</b> Tabla de conexiones del módulo RFID RC522 tanto como para Arduino uno como para Arduino Mega .....	63
<b>Figura 3.6:</b> Control de Acceso a Arduino .....	65
<b>Figura 3.7:</b> Lectura serial de valores enviados por Arduino .....	70
<b>Figura 3.8:</b> Esquema de conexión desde el Arduino Mega hasta la nube.....	71
<b>Figura 3.9:</b> Instalación de librería serial para Python vía Putty desde pc .....	72
<b>Figura 3.10:</b> Creando nuevo archivo formato Python llamado datagaso.py.....	72
<b>Figura 3.11:</b> Comandos para la comunicación serial entre Arduino y Raspberry Pi 3 .....	73
<b>Figura 3.12:</b> Declaración de librerías a usar para crear una base de datos.....	74
<b>Figura 3.13:</b> Conexión a la base de datos .....	74
<b>Figura 3.14:</b> Comandos para importar Hora y Fecha de la PC .....	74
<b>Figura 3.15:</b> Comandos de declaración de variables a exportar a la base de datos.....	75
<b>Figura 3.16:</b> Base de datos y tabla de valores.....	75
<b>Figura 3.17:</b> Configuración de valores de la tabla “Gasolina” .....	76
<b>Figura 3.18:</b> Tabla de valores de la base de datos.....	76
<b>Figura 3.19:</b> Foto fuente de alimentación. ....	78
<b>Figura 3.20:</b> Foto de tubería conduit.....	79
<b>Figura 3.21:</b> Foto fuente de interruptor termostático 10A .....	79
<b>Figura 3.22:</b> Diseño de tapa de Protección con junta tórica.....	80
<b>Figura 3.23:</b> Diseño de la estructura de Base de Fierro .....	81
<b>Figura 3.24:</b> Placa de aluminio simulada en Solidworks™ dimensiones 25 x 25cm .....	81
<b>Figura 3.25:</b> Diseño de la estructura interna del equipo simulado en Solidworks™ .....	82



<b>Figura 3.26:</b> Diseño de la estructura completa del equipo simulado en Solidworks™ .....	83
<b>Figura 3.27:</b> Hora de inicio de prueba .....	84
<b>Figura 3.28:</b> Hora de envío de señal del encoder.....	85
<b>Figura 3.29:</b> Visualización de la base de datos desde una portátil en la misma red. ....	86
<b>Figura 3.30:</b> Visualización de la conexión de dispositivos del sistema .....	87
<b>Anexo 1- Foto:</b> Diseño de la estructura completa del equipo en físico. ....	96
<b>Anexo 2- Foto:</b> Diseño de la estructura completa del equipo en físico. ....	97
<b>Anexo 3- Foto:</b> La estructura completa del equipo en físico.....	98
<b>Anexo 4- Foto:</b> Conexiones eléctricas hacia el equipo. ....	99

## LISTADO DE TABLAS

Tabla 1. Tabla de costos del prototipo del sistema .....	88
Tabla 2. Consumo de energía del sistema .....	89
Tabla 3. Consumo de energía en reposo y en funcionamiento .....	90
Tabla 4. Diagrama de Gantt o cronograma de creación del sistema .....	91

# INTRODUCCION

El presente trabajo de investigación lleva por título “Diseño de un Sistema de Administración y Automatización para Surtidores de combustible líquido con Raspberry PI3 para la empresa Empex”, para optar el título de Ingeniero Electrónico y Telecomunicaciones, presentado por el alumno Akin Abdel Passuni Cori. El cual consiste en el diseño de un sistema electrónico para la mejora de la administración de surtidores de combustibles líquidos los cuales cuentan con un sistema mecánico eléctrico obsoleto, mediante el uso de herramientas electrónicas de software libre y de no muy alto costo, así como también el uso de herramientas de diseño mecánico eléctrico, que nos ayudaron poder elaborar una correcta instalación y posicionamiento del sistema.

En el proyecto se plasmaran en los siguientes tres capítulos, para la comprensión de su desarrollo:

En el primer capítulo trataremos sobre el planteamiento del problema, el cual nos situara en la necesidad de desarrollar este proyecto de investigación.

En el segundo capítulo desarrollaremos el marco teórico, donde encontraremos los antecedentes, que nos permitirán analizar investigaciones que tienen la misma visión sobre sus objetivos y por ende son una referencia para este proyecto.

En el tercer capítulo nos concentraremos en el desarrollo de la metodología y de la parte procedimental del proyecto de investigación, donde haremos la descripción técnica del proyecto y demostraremos su viabilidad.

**El autor**

# **1CAPITULO I: PLANTEAMIENTO DEL PROBLEMA**

## **1.1 DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA**

La industria es la base económica de nuestro país, por ello es necesario que su manejo sea el adecuado a fin de asegurar su desarrollo.

En la actualidad, La mala administración de procesos y el ineficiente control de ellos, son uno de los mayores problemas que enfrentan las empresas en el Perú, a esto se suma la poca capacitación de sus trabajadores y la mala práctica de parte de los mismos empleados en sus diferentes áreas.

Estos son algunos de los problemas más grandes que tienen las gasolineras y las empresas en las cuales se suministra combustible. Los empleados tapan los huecos económicos

y hasta logran lucrar con estos, dado que no hay una fuente confiable de información en donde se puede estar seguro de que la administración del combustible y/o las cuentas de dinero sean las correctas tanto en gasocentros como en las industrias en general.

En el caso de las empresas que necesitan de transporte para su correcta logística, se tergiversan los vales de combustible y su abastecimiento, haciendo que la fábrica tenga una pérdida considerable del recurso hidrocarburo y monetario.

En el caso de las empresas comercializadoras de combustible, esta pérdida económica, es causada directamente por el personal e indirectamente por el usuario, que busca generar una entrada monetaria extra de manera ilícita.

Si bien es necesario enfocarse en el capital humano para su mejora, es indispensable implementar y/o actualizar los sistemas electrónicos encargados de suministrar los datos para su correcto control. Tales como la Actualización digital de los surtidores y el mejoramiento del sistema de despacho de combustibles líquidos. Se podría explicar mejor la problemática con un caso en específico:

La modalidad de robo de combustible por la Policía Nacional del Perú.

En primera instancia, la Policía Nacional Peruana se abastecía en las gasolineras “publicas” y dinero en efectivo, pero dado a que se tergiversaban facturas por venta de combustible mayor a lo consumido y no había manera de cómo llevar un seguimiento de dichas compras de combustible, la PNP decidió darles tarjetas de crédito para el abastecimiento de combustible, a esa tarjeta normalmente se le denomina “Tarjeta multiflota”; con dicho sistema podían llevar un seguimiento a la compra del combustible, mas no a cuanto combustible se

abasteció en cada compra; entonces los policías al ver que ya no podían robar combustible por la modalidad de facturas infladas idearon otra manera de “coimear” y tener ingresos extras; crearon otra modalidad de robo el cual será explicado con las siguientes imágenes.



**Figura 1.1.** Vehículo llegando al gasocentro

*Fuente: Diario La república.*

Primero, el vehículo de la PNP llega o de otra forma nunca llega al gasocentro en el cual supuestamente va a abastecerse de combustible.



**Figura 1.2.** Vehículo de la Policía llenando combustible.

*Fuente: Diario La República*

El automóvil llena menos combustible de lo que fue asignado al patrullero por un superior o nunca llena el combustible ya asignado.



**Figura 1.3,** Efectivo policial pagando con tarjeta multiflota.

*Fuente: Diario La República.*

El efectivo abastece menos de lo establecido o nada. Mientras que el grífel, que es la persona encargada de abastecer, realizar la venta y llenar las facturas de venta, cobra como si hubiera llenado la dotación completa. El policía presenta su tarjeta multiflota al grifero, quien la pasa por el POS del establecimiento. Produciendo el robo.



**Figura 1.4,** Vehículo particular llenando combustible.

*Fuente: Diario La República.*

El combustible sobrante es revendido por el grifo a vehículos particulares los cuales para su venta no se requiere algún tipo de factura, generando un sobrante de dinero que llega al administrador.



**Figura 1.5.** Efectivo policial recibiendo dinero del gasocentro

*Fuente: Diario La República*

Por último los efectivos policiales regresan al gasocentro para recuperar parte de las ganancias generadas por el robo.

Con este ejemplo se pudo entender el gran problema sobre una mala administración en un gasocentro, que aun así se actualice con un sistema de pago por tarjeta los efectivos policiales y por qué no las empresas privadas logran generar ganancias ilícitas con el abastecimiento de combustible.



En las industrias un ejemplo claro de mala administración, es el abastecimiento de combustible a vehículos particulares los cuales no están autorizados para el abastecimiento de combustible dentro de la fábrica, un automóvil cualquiera se dirige hacia la estación de llenado de combustible y se abastece de él, luego como no hay un sistema fiable donde el jefe encargado de área pueda ver el consumo de combustible, este record de combustible faltante es sumado al consumo de otro vehículo autorizado por la fábrica, generando así una pérdida de combustible considerable a la fábrica, sin contar con la informalidad creada.

Los sistemas de control de combustible son normalmente importados, dado a su costo de envío y a su tecnología el costo se hace un poco elevado, su calibración y/o mantenimiento son igualmente de costosos ya que se tienen que reenviar el equipo hacia la fábrica de origen que está en otro país.

## **1.2 JUSTIFICACION DEL PROBLEMA**

Por lo mencionado en la sección anterior, es clara la necesidad del diseño de un sistema de administración y automatización para las empresas tanto en gasocentros como en las industrias en general. Para lograr ello, es factible implementar un Raspberry pi3 como controladora y procesadora de datos, como también una tarjeta Arduino para la recepción de señales y enviar estas señales hacia el Raspberry Pi 3 por un puerto serial, para así poder guardar estos datos y enviarlos a un servidor web, así como también es necesario implementar un control de los vehículos a los cuales están autorizados para ser abastecidos, para esto se utilizará la tecnología del RFID para la autenticación estos.

## **1.3 DELIMITACIÓN DE LA INVESTIGACIÓN**

### **1.3.1 Teórica:**

Fuente de información casi nula en estadísticas de pérdidas por mala administración de combustible en gasocentros e industrias; también falta de información sobre surtidores de combustible.

### **1.3.2 Espacial:**

El sistema está diseñado para lugares donde hay una correcta conectividad wifi.

### **1.3.3 Temporal:**

Comprende el período de: diciembre 2016 a junio 2017

## **1.4 FORMULACION DEL PROBLEMA**

### **1.4.1 PROBLEMA GENERAL**

¿Cómo diseñar un sistema de administración y automatización con Raspberry pi3 para surtidores de combustible líquido para la empresa Empex?

### **1.4.2 PROBLEMAS ESPECÍFICOS.**

- ¿Cómo obtener una correcta comunicación entre el Raspberry Pi3 y el Arduino para así poder enviar datos hacia el servidor web?
- ¿Cómo desarrollar la simulación del diseño del sistema de administración y automatización?

- ¿Cómo obtener los datos del surtidor de combustible correctamente y procesarlos con el Raspberry pi3?
- ¿Cómo crear una base de datos en computadora el cual recepcione los datos enviados por el surtidor mediante el Arduino?

## **1.5 OBJETIVOS.**

### **1.5.1 OBJETIVO GENERAL**

Diseñar un sistema de administración y automatización con Raspberry Pi 3 para surtidores de combustible líquido para la empresa Empex

### **1.5.2 OBJETIVOS ESPECÍFICOS.**

- Obtener una correcta comunicación entre el Raspberry pi3 y el Arduino para así poder enviar datos hacia el servidor web.
- Desarrollar la simulación del diseño del sistema de administración y automatización
- Obtener los datos del surtidor de combustible correctamente y procesarlos con el Raspberry pi3
- Crear una base de datos en computadora el cual recepcione los datos enviados por el surtidor mediante el Arduino

## **2 CAPITULO II: MARCO TEORICO**

### **2.1 ANTECEDENTES**

#### **2.1.1 ANTECEDENTES NACIONALES:**

Como antecedente en el ámbito nacional, tenemos los siguientes proyectos de tesis más resaltantes de los cuales se mencionan trabajos exitosos usando algunos componentes de la familia de las microcomputadoras y microcontroladores, como también de sistemas electrónicos que ayuden de cierto tipo a la administración de recursos.

Guerra (2013) En su Tesis para obtener el título profesional de Ingeniero de Telecomunicaciones: *Diseño de un sistema de control domótico y video vigilancia supervisado por un teléfono móvil*. En la Pontificia universidad Católica del Perú - PUCP. De donde se puede deducir lo siguiente:

La microcomputadora Raspberry Pi puede y es usada como ordenador también a su vez como microcontrolador, teniendo así un sistema operativo como es el Rasberryan y puertos de entrada y salida de señales (GPIO), lo que le permitió en su particular diseño, poder obtener datos de sensores y cámaras para así poder procesar las señales tan bien como lo hace un microcontrolador, y así también pudo comunicar y almacenar los datos de esas señales para poder comunicarlas vía Ethernet, lo que hizo al Raspberry Pi una herramienta ideal para su proyecto.

El lenguaje Python es entre varios, uno de los mejores lenguajes de programación, el cual facilita poder comunicar los sensores con el sistema operativo y así tener un código de programación ligero e igual de fiable que otros lenguajes de programación.

De su investigación se deduce que al ser una tarjeta electrónica ligera y pequeña comparado con una tarjeta de computadora, esta necesita de un almacenamiento externo, el cual se suministra mediante tarjetas micro SD; lo que hace que se tenga que hacer un análisis de la velocidad de almacenamiento a la cual será sometida la tarjeta Raspberry Pi para poder elegir una correcta memoria micro SD, que pueda cumplir con los requerimientos para poder tener un correcto funcionamiento del Raspberry Pi.

Ramos, J, (2017). En su Tesis para obtener el título profesional de Ingeniero Electrónico:

*Diseño e implementación de un prototipo de medición de potencial espontaneo para el monitoreo del volcán Ubinas.* En la Universidad Nacional Del Altiplano – UNA. Del cual se puede concluir lo siguiente:

- a) La microcomputadora Raspberry Pi 3 puede ser usada también como servidor web, así como además de microcontrolador y Computadora, usando el sistema operativo Raspbian, entre otros.
- b) El sistema operativo Raspbian al estar basado en Linux, es un sistema de software libre, por lo tanto, es de fácil instalación y acceso a descargas de programas completamente gratis.
- c) El sistema Raspbian usa comandos de Linux así como el famosos súper usuario (-sudo), para poder acceder a cualquier archivo del sistema.
- d) En aplicaciones de servidores, el Raspberry, es configurado muy similarmente a una consola con sistema Linux, lo que lo hace muy versátil y de conocimiento global.

### **2.1.2 Antecedentes Internacionales.**

Herrera (2014), En su Tesis para obtener el título profesional de Ingeniero Electrónico y de Telecomunicaciones: *Estudio, diseño e implementación de un prototipo de Entrenador de módulo Raspberry Pi*, en la Universidad Tecnológica de Israel. La cual llegó a las siguientes principales conclusiones:

- a) Se estudió el manejo de la placa Raspberry Pi, tanto para cargar el sistema operativo en la memoria SD, y se determinó que el lenguaje Python es el mejor recurso como plataforma de programación.
- b) Se comprobó con la implementación del entrenador que al realizar diseños electrónicos, se redujo el tiempo de ensamble, debido a que las conexiones de polarización las trae consigo el entrenador y se tendrá sólo que conectar las entradas o salidas de las respectivas zonas al Raspberry.
- c) Se diseñó una placa de circuito impreso con el software Proteus que genera los archivos Gerber que sirven para las máquinas CNC que fabrican circuitos impresos al por mayor, esto es útil ya que la idea de la empresa es fabricar placas del entrenador al por mayor.
- d) El departamento de investigación de la empresa Laservisión Cía. Ltda. Validó el correcto funcionamiento del entrenador mediante la prueba de cada una de las zonas y determinó que es el equipo adecuado para el desarrollo de proyectos con Raspberry Pi.

Gonzales (2015), En su Tesis para obtener el Grado de Ingeniero en Informática: *Red de Sensores – Internet de las cosas*, en la Universidad Politécnica de Valencia en la Escuela Técnica Superior de Ingeniería Informática. El cual nos estableció los siguientes puntos, para desarrollar el presente proyecto:

- a) El uso de un sistema sencillo, pero funcional con miras a ampliarse fácilmente, visto que la mayoría de sensores tienen un funcionamiento similar.
- b) La infinidad de opciones que nos proporcionan los servicios web REST son diversos, más aun con la facilidad y acceso permite la adaptabilidad del mismo en diferentes proyectos.
- c) Es sumamente satisfactorio el desarrollo con ARDUINO, el cual permite mejoras en los circuitos progresivamente, como en los sensores.
- d) Por otro lado el uso del Raspberry Pi fue sumamente agradable de usar ya que su comportamiento está orientado como una Pc con Linux.
- e) Por ende consiguió estudiar e implementar los nodos que formaran parte de la red, de tal forma que lean una serie de sensores y ofrecerán los resultados a los usuarios. Incluyendo la comunicación por Internet y así ofrecer los datos obtenidos por los sensores, todo a través de una interfaz para el usuario que será implementada.



## **2.2 BASES TEÓRICAS**

### **2.2.1 AUTOMATIZACIÓN**

#### **2.2.1.1 CONCEPTO DE AUTOMATIZACIÓN**

Según la Real Academia de las Ciencias Físicas y Exactas (2002) define la automática como el conjunto de métodos y procedimientos para la sustitución del operario en tareas físicas y mentales previamente programadas. De esta definición original se desprende la definición de la automatización como la aplicación de la automática al control de procesos industriales. Por ende automatiza es convertir procesos manuales en procesos más eficientes utilizando herramientas tecnológicas.

#### **2.2.1.2 IMPORTANCIA DE LA AUTOMATIZACIÓN.**

La importancia de la automatización radica en que reduce los gastos de mano de obra directa en un porcentaje alto según el grado de automatización, aumenta la calidad de producción ya que un gran porcentaje de las maquinas son más precisas que el humano. Además gracias a su uniformidad de trabajo de obtiene plazos de entrega más exactos en las industrias.

Así como también, se reduce las incidencias laborales puesto que las máquinas automáticas realizan todo tipo de trabajos perjudiciales para el hombre.

### 2.2.1.3 TIPOS DE AUTOMATIZACIÓN

Almazán (2006) Afirma que existen cinco formas de automatizar en la industria moderna, de modo que se deberá analizar cada situación a fin de decidir correctamente el esquema más adecuado.

Los tipos de automatización son:

El Control Automático de Procesos, se refiere usualmente al manejo de procesos caracterizados de diversos tipos de cambios (generalmente químicos y físicos); un ejemplo de esto lo podría ser el proceso de refinación de petróleo.

El Proceso Electrónico de Datos frecuentemente es relacionado con los sistemas de información, centros de cómputo, etc. Sin embargo en la actualidad también se considera dentro de esto la obtención, análisis y registros de datos a través de interfaces y computadores.

La Automatización Fija, es aquella asociada al empleo de sistemas lógicos tales como: los sistemas de relevadores y compuertas lógicas; sin embargo estos sistemas se han ido flexibilizando al introducir algunos elementos de programación como en el caso de los (PLC'S) O Controladores Lógicos Programables.

Control numérico computarizado. Este tipo de control se ha aplicado con éxito a Máquinas de Herramientas de Control Numérico (MHCN). Entre las MHCN podemos mencionar:

- Fresadoras CNC.
- Tornos CNC.

- Máquinas de Electroerosionado
- Máquinas de Corte por Hilo, etc.

El mayor grado de flexibilidad en cuanto a automatización se refiere es el de los Robots industriales que en forma más genérica se les denomina como "Celdas de Manufactura Flexible".

Por su parte la automatización flexible es más adecuada para un rango de producción medio. Estos sistemas flexibles poseen características de la automatización fija y de la automatización programada.

Los sistemas flexibles suelen estar constituidos por una serie de estaciones de trabajo interconectadas entre sí por sistemas de almacenamiento y manipulación de materiales, controlados en su conjunto por una computadora.

#### **2.2.14 DEFINICIÓN DE SISTEMAS AUTOMATIZADOS**

Según Pere (2012), los sistemas automatizados son sistemas hechos por el hombre que interactúan con o son controlados por una o más computadoras (p.56).

Aunque hay diferentes tipos de sistemas automatizados, todos tienden a tener componentes en común:

- a) El hardware de la computadora: los procesadores, los discos, terminales, impresora, unidades de cinta magnética, etcétera.

- b) El software de la computadora: Los programas de sistemas tales como sistemas operativos, sistemas de base de datos, programas de control de telecomunicaciones, etcétera.
- c) Las personas: los que operan el sistema, los que proveen su material de entrada y consumen su material de salida, y los que proveen actividades de procesamiento manual en un sistema.
- d) Los datos: la información que el sistema recuerda.
- e) Los procedimientos: las políticas formales e instrucciones de operación del sistema.

Una división categórica de los sistemas automatizados es la siguiente:

**Sistemas en línea:** es aquel que acepta material de entrada directamente del área donde se creó. También es sistema en el que el material de salida, o resultado de la computación, se devuelve directamente a donde es requerido.

**Sistemas de tiempo real:** puede definirse como aquel que controla un ambiente recibiendo datos, procesándolos y devolviéndolos con la suficiente rapidez como para influir en dicho ambiente en ese momento.

**Sistemas de apoyo a decisiones:** Estos sistemas computacionales no toman decisiones por sí mismos, sino ayudan a los administradores, y a otros profesionistas "trabajadores del conocimiento" de una organización a tomar decisiones inteligentes y documentadas acerca de los diversos aspectos de la operación.

**Sistemas basados en el conocimiento:** Estos sistemas contienen grandes cantidades de diversos conocimientos que emplean en el desempeño de una tarea dada. Los sistemas expertos son una especie de sistemas basados en el conocimiento, aunque ambos términos a menudo se utilizan indistintamente.

Muchas organizaciones que desarrollan sistemas únicos, el enfoque ascendente presenta un gran número de dificultades serias:

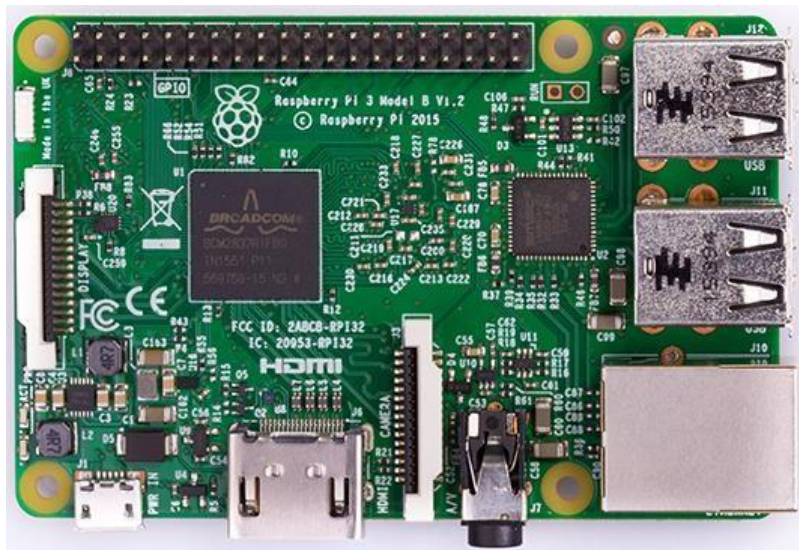
- Nada está hecho hasta que todo esté terminado.
- Las fallas más triviales se encuentran al comienzo del período de prueba y las más graves al final.
- La eliminación de fallas suele ser extremadamente difícil durante las últimas etapas de prueba del sistema.
- La necesidad de prueba con la computadora aumenta exponencialmente durante las etapas finales de prueba.

La segunda debilidad más importante del ciclo de vida de un proyecto clásico es su insistencia en que las fases se sucedan secuencialmente. Queremos esto es una tendencia natural humana: deseamos decir que hemos terminado la fase de análisis del sistema y que nunca tendremos que volver a preocuparnos por ella. El único problema del progreso ordenado es que no es nada realista. Por ejemplo, durante el período que transcurre para desarrollar el sistema pueden cambiar ciertos aspectos del ambiente del usuario (la economía, la competencia, los reglamentos gubernamentales que afectan a las actividades del usuario).

## 2.2.2 Raspberry Pi

### 2.2.2.1 Definición

Para Gonzales (2015) El Raspberry pi es un integrado de placa reducida, o (placa única), de bajo costo desarrollado en el Reino Unido por la fundación “Raspberry Pi”, con el objetivo de estimular la enseñanza de ciencias de la computadora en las escuelas y universidades (p.32).



*Figura 2.1 Raspberry Pi3 tipo B, este es el dispositivo tecnológico adquirido para realizar todas las pruebas del proyecto, ahí se puede observar sus puertos y su forma física.*

*Fuente: [www.raspberrypi.org](http://www.raspberrypi.org)*

El diseño incluye un chip Broadcom BCM2837, que contiene un procesador central (CPU) ARMv8 a 1.2GHz (el firmware incluye unos modos “Turbo” para que el usuario pueda hacerle overlock de hasta 1.4 GHz sin perder la garantía), un procesador gráfico (GPU) Video Core IV, y 1GB de memoria RAM. El diseño no incluye un disco duro ni unidad de estado

sólido, ya que usa una tarjeta SD para el almacenamiento permanente y el arranque del sistema operativo. Tampoco incluye fuente de alimentación ni carcasa.

La fundación da soporte para las descargas de las distribuciones para la arquitectura ARM, Raspbian (derivada de Debían), RISC OS 5, Arch Linux ARM(derivado de Arch Linux) y Pidora (derivado de Fedora) y promueve principalmente el aprendizaje del lenguaje de programación Python, y otros lenguajes como Ti y Basic, C y Perl.

### **2.2.2.2 Característica y especificaciones Técnicas del Raspberry Pi**

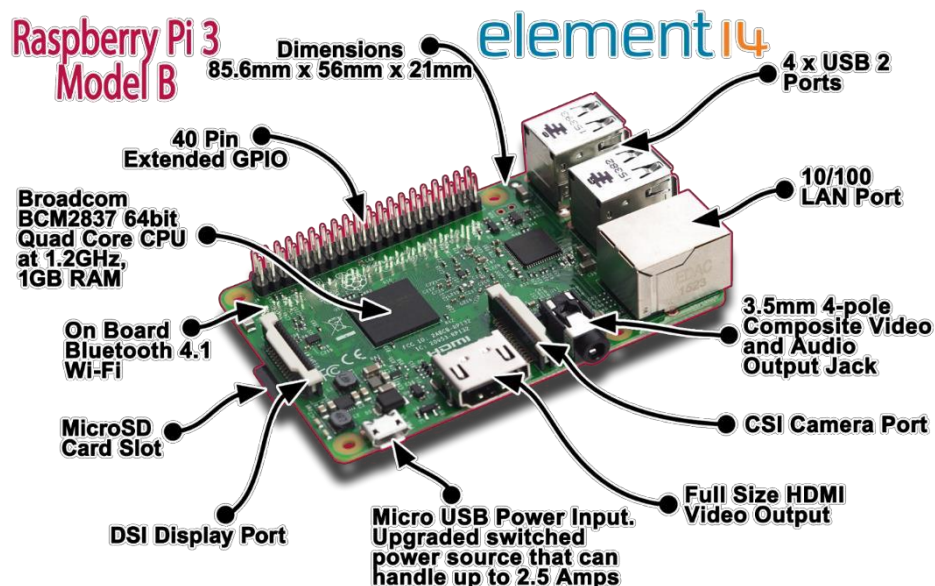
Para Israel(2014), el Raspberry pi3 Modelo C cuenta con conectores más que suficientes para permitir que el usuario pueda tener diferentes opciones de uso y así tiene un conector HDMI para alta definición como de un puerto de salida RCA, además su alimentación es tan sencilla como conectarse al adaptador micro-USB de un teléfono (p.17).

Aquí se presentan sus especificaciones técnicas:

- Procesador Broadcom de 1.2 GHz ARMv8 BCM2837 con FPU y Videocore 4 GPU.
- BCM43438 WiFi en tarjeta.
- GPU proporciona una tecnología Open GL ES 2.0, hardware acelerado OpenVG y admite imágenes de alta resolución 1080p30 H.264.
- GPU tiene una capacidad de 1 Gpixel/s, 1,5 Gtexel/s o 24 GFLOPs con filtrado e infraestructura DMA.
- 1GB de RAM (Modelo B).
- Arranca desde tarjeta SD, ejecutando una versión del sistema operativo Linux
- Conector hembra Ethernet 10 / 100 Base T (modelo B).

- Conector hembra de vídeo HDMI.
- conectores hembra USB 2.0 (modelo B).
- Conector hembra de vídeo compuesto RCA.
- Ranura para tarjeta SD.
- Alimentación a través de conector micro USB.
- Conector de audio Jack 3,5 mm.
- Conector hembra para buses serie y GPIO.
- Puerto para conector JTAG.
- Conector para videocámara HD Raspberry Pi (775-7731).
- Dimensiones: 86 x 56 x 20 mm (modelo B) 86 x 56 x 17 mm (modelo A)

En la imagen que se muestra a continuación, se observa cada uno de las partes, puerto y ranuras de la tarjeta.



*Figura 2.2 Raspberry pi y sus puertos. Descripción de los puertos de la tarjeta Raspberry Pi3*

*Modelo B.*

*Fuente: www.element14.com*



### 2.2.2.3 Sistema Operativo

El sistema operativo del Raspberry pi trabaja en entorno de Linux y contiene algunas distribuciones que se pueden utilizar para el funcionamiento de la tarjeta. Aquí se ve una lista de los diferentes sistemas:

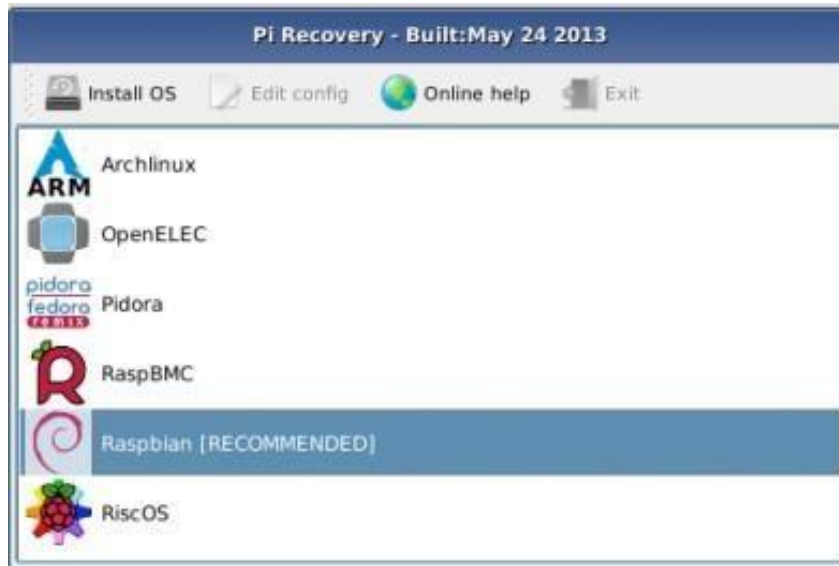
- Aros
- Linux
- Androide 98
- Arch Linux ARM
- Debian Whezzy
- Firefox OS
- Gento o Linux 99
- Pidora
- Etc.

El sistema operativo se trata de un Software basado en Linux, así que existen numerosos variantes de este. El núcleo principal es Raspbian, pero existen numerosas alternativas, como RISC OS, PIDORA, Snappy, Etc.

Además se puede descargar el paquete de NOOBS (New of the Box Software) que contiene todo lo necesario para comenzar a usar la Raspberry pi desde el primer momento y esto se lo descarga desde la página oficial del mismo. (Web oficial Raspberry pi).

Así se dice que el nuevo Raspberry pi 2 será compatible con Windows 10, con una versión creada por Microsoft que será gratuita para los desarrolladores.

Se muestra la figura 2.3 el arranque de instalación del sistema de instalación Noob para Raspberry pi, en el cual se visualiza los diferentes sistemas operativos que se pueden instalar.



*Figura 2.3 Imagen de la instalación del S.O. del Raspberry Pi. Aquí se muestra la pantalla de instalación del S.O. Y se observan la selección de los diferentes sistemas disponibles para la tarjeta. Adaptado de la página Principal del Raspberry distribución*

## **2.2.3 ADMINISTRACIÓN**

### **2.2.3.1 GENERALIDADES.**

Establecemos que la administración es universal, como necesidad, como disciplina, como proceso, como conjunto de técnicas y herramientas son necesarias estudiar con toda la seriedad y a toda profundidad.

### **2.2.3.2 DEFINICIONES DE ADMINISTRACIÓN.**

Administración es la ciencia, técnica o arte que por medio de los recursos humanos, materiales, y técnicos, pretende el logro óptimo de los objetivos mediante el menor esfuerzo para lograr una mayor utilidad (Stoner, 1996, p. 7)

Dentro de la definición anterior, se destacan los medios por los cuales el concepto de Administración es una ciencia. En el Recurso Humano se persigue la satisfacción de objetivos organizacionales a través de un “Esfuerzo Humano Combinado”, como los Recursos Materiales también son importantes para lograr los objetivos de la industria, también el Recurso Técnico; factor clave para el desarrollo organizacional; persigue la tecnificación en el desarrollo de las actividades dentro de la empresa. Donde el “Proceso de conseguir que se realicen las actividades eficientemente con otras personas y por medio de ellas” (Robbins, Coulter, 2005, p.5).

La eficiencia dentro de las actividades, exige responsabilidades para emprender acciones que permitan a las personas, realizar sus mejores contribuciones al cumplimiento de metas.

La Administración es una de las actividades más importantes, pues a medida en que la sociedad empezó a depender del esfuerzo grupal y que muchos grupos organizados tendieron a crecer; la tarea de los administradores se volvió más importante, puesto que la administración persigue la productividad, lo que implica eficiencia y eficacia.

“Es el proceso de hacer que las actividades se cumplan con eficiencia y eficacia por medio de otras personas.” (Robbins, Coulter, 2005, p.5).

Lograr la consecución de los objetivos, o más bien dicho el logro de los efectos deseados en cualquier actividad, es el término que define la eficacia. Para poder lograr que las

personas alcancen su mayor productividad, es necesario guiarlos por el camino que conlleve a los objetivos de la organización, al mismo tiempo efectuar las tareas con la menor cantidad de recursos, es lo que genera la eficiencia dentro de la empresa.

### **2.2.3.3 IMPORTANCIA DE LA ADMINISTRACIÓN.**

La Administración es importante porque ayuda a las empresas ya sean grandes, medianas o pequeñas a ser eficientes y eficaces en el uso y manejo de los recursos.

Es necesario considerar cada uno de los elementos, maquinaria, mercado, mano de obra, entre otras, para una mayor productividad, pues actualmente es de mucha importancia que la empresa sea muy productiva en el desarrollo de sus actividades, al mismo tiempo mejorar a la calidad de la administración.

### **2.2.3.4 CARACTERÍSTICAS DE LA ADMINISTRACIÓN.**

El propósito de la Administración se refiere a algo específico, como un objetivo en donde los administradores centran su atención y esfuerzo, en producir acciones exitosas, en la dirección de individuos con propósitos comunes.

La administración se logra mediante el esfuerzo de todos los que participan en las acciones u objetivos que persigue una determinada organización. Para participar en la administración, se requiere abandonar la tendencia a ejecutar todo por uno mismo y hacer que las tareas se cumplan en conjunto, es decir mediante los esfuerzos de los miembros de un determinado grupo.

### **Universalidad.**

Para Ñahui (2015) Esto nos indica que la administración se da dondequiera que exista una organización, porque en esta debe de existir siempre una coordinación sistemática de todos los medios que la componen (p.33).

El fenómeno administrativo se da donde quiera que existe un organismo social, porque en él tiene siempre que existir coordinación sistemática de medios. La administración se da por lo mismo en el estado, en el ejército, en la empresa, en una sociedad religiosa entre otras. Y los elementos esenciales en todas esas clases de administración serán los mismos, aunque lógicamente existan variantes accidentales.

### **Su especificidad.**

Aunque la administración va siempre acompañada de otros fenómenos (fusiones económicas, contables, productivas, mecánicas, jurídicas, entre otras), el fenómeno administrativo es específico y distinto a los que acompañan.

### **Su unidad temporal.**

Aunque se distingan etapas, fases y elementos del fenómeno administrativo, y por lo mismo en todo momento de la vida de una empresa se están dando, en mayor o menor grado, todos o la mayor parte de los elementos administrativos.

### **Su unidad jerárquica.**

Todos cuantos tienen carácter de jefes en un organismo social, participan en distintos grados u modalidades de la misma administración. Así, en una empresa forman un solo cuerpo administrativo, desde el Gerente General, hasta el último mayordomo.

### **2.2.3.5 PRINCIPIOS GENERALES DE LA ADMINISTRACIÓN.**

La aplicación de los principios de administración está orientada a la simplificación del trabajo de la empresa, lo cual incluye que se resuelva el trabajo en algo más sencillo para las personas.

Se han planteado muchos principios de la Administración General, cada autor ha expuesto su propio enfoque; sin embargo, a continuación se presentan los Principios Generales de Administración.

#### **Realizar las tareas con el menor esfuerzo.**

Todas las actividades que se señalan deben de obtenerse con efectividad y eficiencia, entendiendo por efectividad el logro del objetivo sin valuar y eficiencia como una relación de los recursos empleados con respecto a los resultados obtenidos.

#### **Racionalidad.**

Todo tiene que proveerse o planificar para saber qué es lo que puede incidir, sobre lo que prioritariamente queremos y cuanto podemos obtener en base a lo que tenemos.

#### **Obtención del máximo provecho.**

Obtener lo más, con el aprovechamiento de todos los recursos: de tiempo, dinero, esfuerzos, riesgo y otros.

#### **Obtención de productividad.**

Es el coeficiente obtenido del producto total por uno de los factores de la producción, o el total de insumos invertidos.

### **Reducción de Costos.**

Busca la efectividad para lograr rebajar los precios, a través de la reducción del valor del costo de la producción del bien o servicio.

### **Rentabilidad.**

Es la relación entre el valor de la producción (ventas totales) menos los gastos incurridos en ella, por una parte del capital empleado por la otra, se le conoce como el rendimiento patrimonial.

### **Economicidad.**

Reducción de tiempo, espacio, esfuerzo y recursos, entre otros.

### **Buen ambiente en el trabajo.**

Relaciones de prosperidad y satisfacción en el desarrollo del trabajo.

El uso de los principios de la Administración tiene como finalidad simplificar el trabajo administrativo, Henry Fayol enumeró catorce principios basados en su experiencia que pueden resumirse así:

- a. División del trabajo.
- b. Autoridad y Responsabilidad.
- c. Disciplina.
- d. Unidad de mando.
- e. Unidad de dirección.
- f. Subordinación del interés particular al general.
- g. Remuneración del personal.
- h. Centralización.
- i. Jerarquía de Autoridad.

- j. Orden.
- k. Equidad.
- l. Estabilidad del personal.
- m. Iniciativa.
- n. Espíritu de Grupo.

### **2.2.3.6 AREAS FUNCIONALES DE LAS ORGANIZACIONES.**

Las áreas funcionales de la empresa están íntimamente relacionadas entre sí, por lo que cada una de ellas, depende del accionar de las demás.

Las áreas funcionales están bajo el proceso administrativo, como generador de la actividad de cada una de ellas y el total de la empresa. Entre ellas se incluyen:

- **Función de Mercadeo.**
- **Función de Producción**
- **Función Financiera**
- **Función Administración**

### **2.2.3.7 PROCESO ADMINISTRATIVO.**

Todo proceso administrativo desde el punto de vista conceptual o metodológico, es con el fin de aplicar y comprender mejor la administración, ya que separa los elementos que en una circunstancia, se pueden dominar e integrar para dar resultados positivos dentro de una organización o empresa.

Las etapas que comprende el proceso administrativo son la base fundamental para el logro de las metas a corto, mediano y largo plazo; teniendo como fin, permitir que la



administración pueda guiar a la empresa a la realización de los objetivos que se hubieren fijado o que se propongan, logrando de esa forma los objetivos trazados, aquella está compuesta de diferentes etapas básicas a través de las cuales se realiza la administración. Es aplicable a los administradores en todos los niveles organizacionales en cualquier tipo de organización. Las fases del Proceso Administrativo son las siguientes:

**a) Planeación.**

Esta fase del proceso predetermina el curso de acción a seguir, permite decidir qué hacer antes de hacerlo, como base para prever y manejar el futuro; en consecuencia, planeación es sistematizar por adelantado los objetivos, políticas, programas, proyectos, planes de acción, entre otras. Tiene por objeto asegurar la supervivencia empresarial y neutralizar la incertidumbre.

**Definición de Planeación.**

“Es el proceso de mirar hacia el futuro para determinar el curso de acción que la empresa deberá seguir para alcanzar los objetivos” (Debes del Pino, 1982, p.20)

“Es el proceso en que se establecen las metas y las directrices apropiadas para el logro de metas” (Stone, James, 1989, p. 54)

**Importancia de la Planeación.**

La planeación es importante porque conduce las principales actividades organizativas, de liderazgo y control, teniendo claro que todo intento de control sin planes carece de sentido, porque la organización no tendrá manera de saber si efectivamente se dirige a dónde quiere ir, sin antes saber a dónde quiere ir, por lo que se explica lo importante que es la planeación.

## **b) Organización.**

La organización reúne todos los recursos básicos en forma ordenada y acomodada a las personas, en un esquema aceptable que pueda desempeñar las actividades requeridas. La organización une a las personas en tareas interrelacionadas. Está diseñada para que apoye a las personas a que trabajen en conjunto de forma eficaz, con el objetivo de alcanzar las metas de la empresa.

### **Definición de Organización.**

“Es el proceso de determinar cuáles son las actividades a realizar, quién las llevará a cabo, como deben agruparse éstas, quién informa a quién, y donde se tomarán las decisiones”.

### **Importancia de la Organización.**

La importancia de la organización radica en el hecho de ser un medio de ordenación indispensable para lograr la efectividad con el esfuerzo en conjunto, donde se coordinan las acciones de muchos individuos, de modo que la actuación es satisfactoria, ya que el propósito de la organización es ayudar a que los objetivos tengan significado y contribuir a la eficiencia organizacional.

## **c) Dirección.**

La dirección comprende la influencia del administrador en la realización de los planes, obteniendo de una manera significativa una respuesta “positiva” de parte de los empleados mediante la comunicación, la supervisión y la motivación.

Se puede mencionar que la dirección influye de una manera directa sobre las actividades de un individuo o grupo, en los esfuerzos que se realicen, encaminados al logro de metas en una situación determinada.

### **Definición de Dirección.**

“La Dirección es el proceso que consiste en influir en las personas para que contribuyan al cumplimiento de las metas organizacionales y grupales” (Koontz, 1998, p.24).

Cabe mencionar que influir en la personas para que contribuyan al cumplimiento de las metas dentro de la organización, es necesario incluir factores como la motivación a los subordinados, la conducción de otros, la selección canales de comunicación más efectivos y la resolución de conflictos.

### **Importancia de la Dirección.**

La Dirección es muy importante porque contribuye al cumplimiento de las metas organizacionales y grupales, por medio de la influencia hacia los individuos, aplicando liderazgo, motivación y formas de comunicación eficaz y oportuna por parte de los administradores, encaminándose al logro de los objetivos en la empresa.

El reto de la planificación y organización no solamente es crear un plan estratégico razonable y una estructura organizativa ideal para alcanzar metas, se trata de traducir dichas acciones en actos y, para sostenerlo, los gerentes deben de tener la voluntad para alentar y apoyar a las personas que lleven los planes a la práctica y que trabajen dentro de la estructura organizativa.

#### **d) Control.**

El Control es la última fase del proceso administrativo mediante el cual se miden los resultados obtenidos, con respecto a los que se esperaba en los planes elaborados para determinado propósito. El control permite visualizar y analizar las desviaciones o diferencias, con el objeto de informar oportunamente a la dirección para tomar las medidas correctivas.

Esta fase del proceso es esencial para todos los administradores, aunque el alcance del control varía de un administrador a otro, todos tienen las responsabilidades de ejecutar planes, de manera que el control es una de las funciones administrativas más soportadas en cualquier nivel.

#### **Definición de Control.**

“Función Administrativa que consiste en medir y corregir el desempeño individual y organizacional para asegurar que los acontecimientos se adecuen a los planes” (Koontz, 1998, p.24).

Esta definición explica la importancia de determinar lo que se está llevando a cabo en una empresa, a fin de establecer las medidas correctivas necesarias para evitar desviaciones en la ejecución de los planes.

#### **Importancia del Control.**

Para facilitar de manera eficiente el logro de los objetivos es necesario Planear, crear una estructura Organizacional y Dirigir, así como motivar a los empleados. No obstante no hay garantía; es ahí donde el control es importante para verificar las actividades y asegurarse que se estén llevando a cabo como se planearon y así corregir cualquier desviación importante.

#### 2.2.4 Surtidores de Combustibles

Como definición, podemos establecer en cuanto a la etimología de esta construcción, es:

“Surtidor” deriva del verbo surtir y este se origina en el latín. Proviene del latín vulgar *surtus* que es una contracción del participio *surrectus*, del verbo *surgo*, *surgis*, *surgere*, *surrexi*, *surrectum* que significa surgir, alzarse, aparecer.

“De” que nace del latín. Deriva de idéntica preposición y cuyo significado es punto de partida o separación.

“Combustible” que también proviene del latín. Procede del verbo *combuo*, *comburis*, *comburere*, *combussi*, *combustum* cuyo concepto es quemar.

Los Surtidores de Combustible son considerados como dispositivo para la venta de gasolina, gas-oil, etc., con destino al consumo en los vehículos automóviles.

En los primeros años de siglo, la venta de esos productos se efectuaba a través de las entidades responsables del automovilismo (automóvil club, etc.) e incluso en droguerías, farmacias y establecimientos análogos mediante latas. En cualquier caso, se evitaba la proximidad de estos locales de distribución a los centros de las poblaciones, por el peligro de explosión que aquella forma de manejo suponía. Después de la primera guerra mundial comenzaron a utilizarse para uso público las primeras bombas de gasolina, de accionamiento

manual (mediante manivela o dispositivo análogo). Ello permitió ya la generalización del depósito único, que pronto sería subterráneo y protegido.

Sin embargo, la difusión a gran escala de esos dispositivos no se alcanzó hasta finales de los años treinta, con la introducción de los modelos de medición automática con bomba eléctrica. En la posguerra se produjo otra innovación: la bomba calculadora, que podía indicar la cantidad y el precio del carburante suministrado, cuyo funcionamiento se producía solamente al llevar el registrador a cero. Los surtidores modernos miden y suministran simultáneamente el carburante.

La gasolina o producto de que se trate pasa a través de una bomba rotatoria, un separador de aire, un contador y un indicador de flujo. El llenado del depósito del vehículo se efectúa mediante una pistola de caudal regulable.

En los años setenta se han ido introduciendo en diversos países los surtidores automáticos, que van equipados con un cuadro de mandos. Éste va provisto de un cajetín donde se coloca el billete, que es sometido a un dispositivo fotoeléctrico que controla sus dimensiones, transparencia, trama, etc., y, si es falso, lo rechaza. En caso positivo, ya se puede pulsar el botón del producto deseado (gasolina normal, super, etc.) y se inicia el suministro, que puede detenerse mediante interruptor antes de que finalice, recuperando el cambio de dinero. Naturalmente, el sistema está protegido con dispositivos antirrobo especiales.

Un surtidor tiene tres partes:

#### A. - los tanques de almacenamiento del combustible

Tanto la nafta como el gasoil e inclusive el gas, se almacenan bajo tierra en depósitos subterráneos. Cada uno tiene una capacidad de varios miles de litros. Normalmente existen dos tanques de este tipo por estación de servicio.

Para mover el combustible desde ese lugar subterráneo hasta el nivel de la calle adonde se dispensa el combustible, se debe recurrir a una de estos tipos de bombas:

-una bomba sumergible que se halla dentro del líquido y que tiene una hélice que impulsa el combustible hacia arriba. Las palas de la hélice giran con un motor y mueven el líquido.

-una bomba de succión que mueve el líquido partiendo del principio de la presión desigual. Se inserta un tubo en el agua. Y mediante un motor que se encuentra encima del nivel del combustible elimina el aire de la tubería con el fin de disminuir la presión del aire encima del combustible, permitiendo cuando dicha presión baja, que el combustible suba a la superficie.

B.- Válvula de retención. Cuando el bombeo del combustible se ha completado y el motor de la bomba se apaga, el gasoil o la nafta que ha quedado en la tubería no vuelve a caer en el tanque. En cambio queda atrapada gracias a la válvula de retención que se halla encima del combustible dentro del tubo y con un sello hermético.

C.- Medidor de flujo. El combustible que va hacia el dispensador pasa a través de una válvula reguladora que se encarga de medir la velocidad de flujo de dicho combustible. Esta acción la realiza a través de una membrana plástica que se comprime cada vez más estrechamente por el tubo a medida que el flujo de combustible aumenta, dejando un espacio suficiente para que pase la cantidad correcta a través de él. Si se ha predeterminado una cantidad de combustible para bombear, a medida que se aproxima el límite, el flujo se va reduciendo.

También este tubo tiene un medidor de flujo que es de hierro o aluminio y que tiene un rotor o engranajes por los que pasa el combustible. De manera que estos leen el flujo y pasan la información a la computadora que se halla en el dispensar mostrando la cantidad medida en décimas de litros.



**Figura 2.4** Imagen de Tokheim 1990. Aquí se muestra el Tokleim del año 1990.

Fuente: <http://tokheimprofleet.com/es/sobre-nosotros/historia/>





**Figura 2.5** Imagen de Tokheim 1980. Aquí se muestra el Tokleim del año 1980.

Fuente: <https://www.globalauctionguide.com/government-auction-s-243373.html>



**Figura 2.6** Imagen de Tokheim 2000. Aquí se muestra el Tokleim del año 2000.

Fuente: <http://tokheimfleet.com/es/dispenser/quantium-310/>

## **2.3 MARCO CONCEPTUAL**

### **2.3.1 Micro controlador**

Un microcontrolador es un circuito integrado diseñado para cumplir un número de funciones previamente programadas en su memoria, se compone de al menos tres elementos principales, aunque podría contar con otros:

Un procesador, que se encarga de realizar las operaciones lógicas e interpretar las instrucciones recibidas por el microcontrolador.

Una memoria, que se encarga de almacenar de manera temporal o permanente los datos para el procesamiento del microcontrolador.

Dispositivos de entrada/salida para recibir y enviar la información que procesa el microcontrolador

### **2.3.2 Micro controlador ARM**

Un micro controlador ARM o (Advanced Risc Machine) son también considerados como computadores con un número reducido de instrucciones como lo dice su nombre (RISC significa computador de instrucciones reducidas), y su tamaño reducido, bajo consumo y bajo consumo de potencia los convierte en una opción muy útil en soluciones de electrónica en hogares o en entornos móviles, y existen varios microcontroladores ARM que cuentan con sistemas operativos de libre distribución como lo es el Raspberry Pi, el cual cuenta con pines de entrada y salida para enviar y recibir señal de dispositivos externos.

### **2.3.3 Pines GPIO**

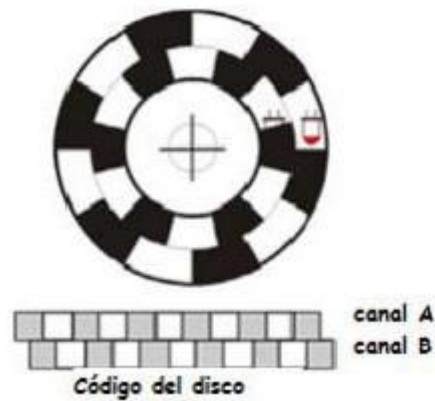
Los pines de entrada y salida de propósito general, o GPIO por sus siglas en inglés, son pines presentes en varios micro controladores ARM y que permiten su conexión con dispositivos externos ya sean de entrada o salida, para que estos puedan ser controlados por el micro controlador, algunos dispositivos que se pueden conectar usando estos pines son sensores, servos, relés, interruptores, entre muchos otros, la señal recibida por estos pines o enviada por estos pines es procesada en el micro controlador e interpretada de acuerdo a la programación que se les haya dado previamente.

### **2.3.4 Sensores**

Los sensores son dispositivos electrónicos que se encargan de detectar variaciones en las condiciones del ambiente que se busca estudiar, y ante una de estas variaciones, son capaces de enviar una señal de alerta al micro controlador pues estas variaciones son interpretadas como eventos, de forma que se pueda realizar una acción previamente programada, por ejemplo se tienen los sensores de movimiento, de humo, así como de fugas de gas, entre muchos otros.

### **2.3.5 Encoder**

El encoder es un transductor rotativo que transforma un movimiento angular en una serie de impulsos digitales. Se basa en la rotación de un disco graduado con un retículo radial formado por espacios opacos, alternados con espacios transparentes. Un sistema óptico de emisor receptor infrarrojo detecta el cambio en la superficie del disco, generando dos señales de cuadratura (desfasadas  $90^\circ$ ), las señales se identifican como A y B.



*Figura 2.7 Imagen del Encoder. Aquí se muestra el Código de disco.*

*Fuente: <http://setecindca.com/descargas/encoders/encoder-eltra.pdf>*

### **2.3.6 Base de datos**

Denominado 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 impresos 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.

### **2.3.7 Interfaz**

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.

### **2.3.8 RFID**

Denominada la identificación por radiofrecuencia (del inglés Radio Frequency Identification) es un sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas o transpondedores RFID.

### **2.3.9 Arduino Mega**

Arduino es una marca de microcontroladores mundialmente conocida por los amantes de la electrónica, la programación y la robótica. Es un proyecto Open Source que pone a disposición de sus usuarios una amplia gama de dispositivos basados en el microcontrolador AtMega. Es posible comprar una placa Arduino armada o conseguir las piezas para uno mismo desarrollar sus propios dispositivos.

### **2.3.10 Señales digitales**

La señal digital es un tipo de señal en que cada signo que codifica el contenido de la misma puede ser analizado en término de algunas magnitudes que representan valores discretos, en lugar de valores dentro de un cierto rango.

### **2.3.11 Sensor Fotoeléctrico**

Un sensor fotoeléctrico o fotocélula es un dispositivo electrónico que responde al cambio en la intensidad de la luz. Estos sensores requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor.

Todos los diferentes modos de sensado se basan en este principio de funcionamiento. Están diseñados especialmente para la detección, clasificación y posicionado de objetos; la detección de formas, colores y diferencias de superficie, incluso bajo condiciones ambientales extremas.

### **2.3.12 PIN**

En electrónica, se denomina pin, palabra inglesa que significa «clavija», a la terminal o patilla de cada uno de los contactos metálicos de un conector o de un componente fabricado de un material conductor de la electricidad.

### **2.3.13 Librerías**

En informática, una biblioteca (del inglés library) es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

### **2.3.14 Monitor Serial**

El Monitor Serial es una ventana emergente separada que actúa como un terminal independiente que se comunica recibiendo y enviando Datos Seriales.

### **2.3.15 Baudios**

La tasa de baudios (en inglés baud rate) —también conocida como baudaje— es el número de unidades de señal por segundo. Un baudio puede contener varios bits. Aunque a veces se confunden los baudios con los bits por segundo, son conceptos distintos.

### **2.3.16 Seteado**

Establecer la configuración de un programa o componente físico para que funcione correctamente.

### **2.3.17 Web**

Se considera Web al conjunto de información que se encuentra en una dirección determinada de internet.

### **2.3.18 Comunicación Serial**

Existen otras líneas disponibles para realizar handshaking, o intercambio de pulsos de sincronización, pero no son requeridas. Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad.

### **2.3.19 Python**

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

### **2.3.20 Pulsos**

El término pulso electromagnético o PEM (en inglés, EMP, de Electromagnetic Pulse) puede referirse a una emisión de energía electromagnética de alta intensidad en un breve período de tiempo.

### **2.3.21 MySQL**

MySQL es la base de datos de código abierto más popular del mundo. Con su rendimiento, confiabilidad y facilidad de uso comprobados, MySQL se ha convertido en la principal opción de base de datos para aplicaciones basadas en la Web, utilizada por propiedades web de alto perfil como Facebook, Twitter, YouTube, y los cinco principales sitios web. Además, es una alternativa extremadamente popular como base de datos integrada, distribuida por miles de ISV y OEM.

### **2.3.22 PHPMyAdmin**

PhpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas. Se encuentra disponible bajo la licencia GPL Versión 2.



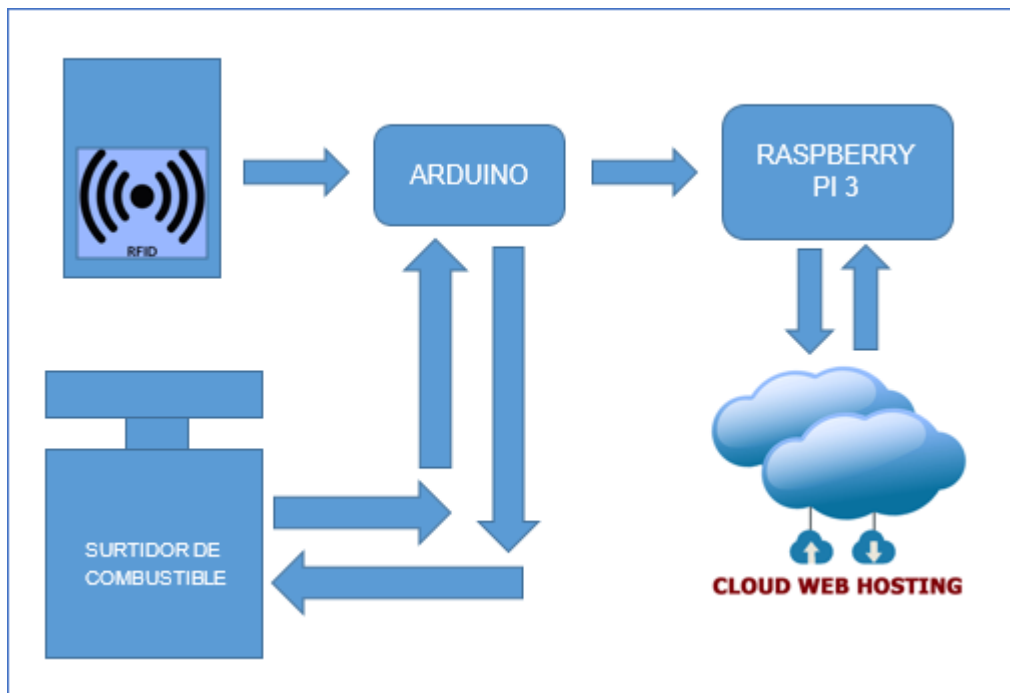
### **3 CAPITULO III: DESARROLLO DE LA METODOLOGIA**

#### **3.1 DESCRIPCIÓN DEL PROYECTO**

Este proyecto consiste en el diseño de un sistema electrónico de administración y automatización de surtidores de combustible, este sistema se encarga de adquirir datos de los surtidores de combustible antiguos para poder crear un sistema actual en el cual se muestran datos en un ordenador ya resumidos y de fácil entendimiento con la debida fecha y hora en el cual estos datos fueron generados. Además también este sistema tiene una tecnología de identificación mediante RFID el cual se utilizaría para un control de vehículos dentro de una empresa privada, identificando cuales motorizados pueden o no ser abastecidos de combustible.

Este proyecto puede ser dividido en 2 partes:

- a) Adquisición de datos del encoder, del surtidor de combustible y también del RFID; datos obtenidos por Arduino.
- b) Enviar datos del Arduino hacia el Raspberry para que puedan ser almacenados, procesados y poder enviarlos hacia una base de datos web.

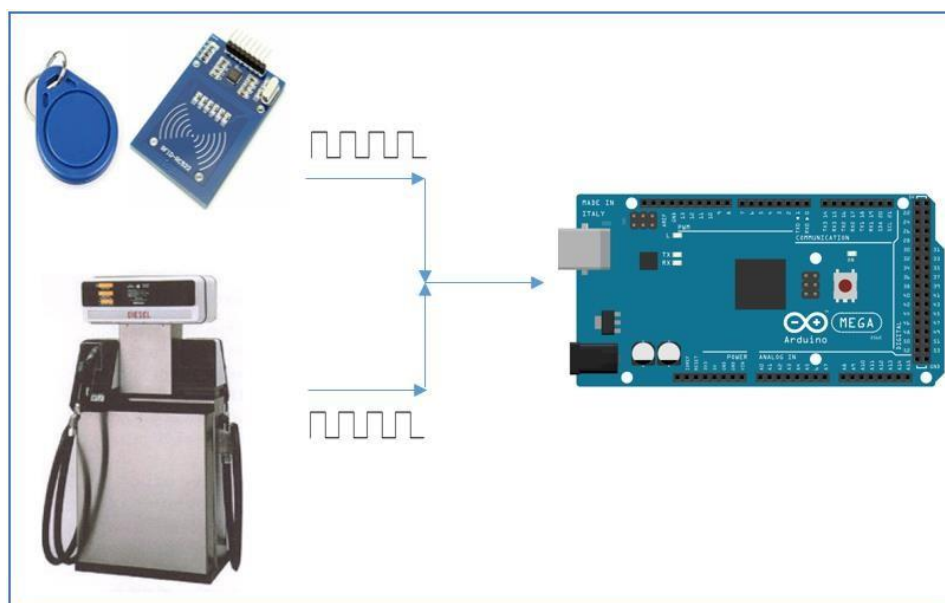


**Figura 3.1:** Diagrama de sistema de administración y automatización para surtidores de combustibles líquidos.

*Fuente: Elaboración propia.*

### 3.1.1 Adquisición de datos

Este es el primer paso del diseño, en esta parte tendremos que adquirir las señales digitales de parte del encoder del surtidor de combustible y del RFID, esto lo haremos con la tarjeta de desarrollo Arduino en su versión Arduino Mega, dada a su gran versatilidad y a su gran compatibilidad con los diferentes módulos y tipos señales.



**Figura 3.2:** Diagrama de adquisición de datos del surtidor y del RFID hacia el ARDUINO MEGA.

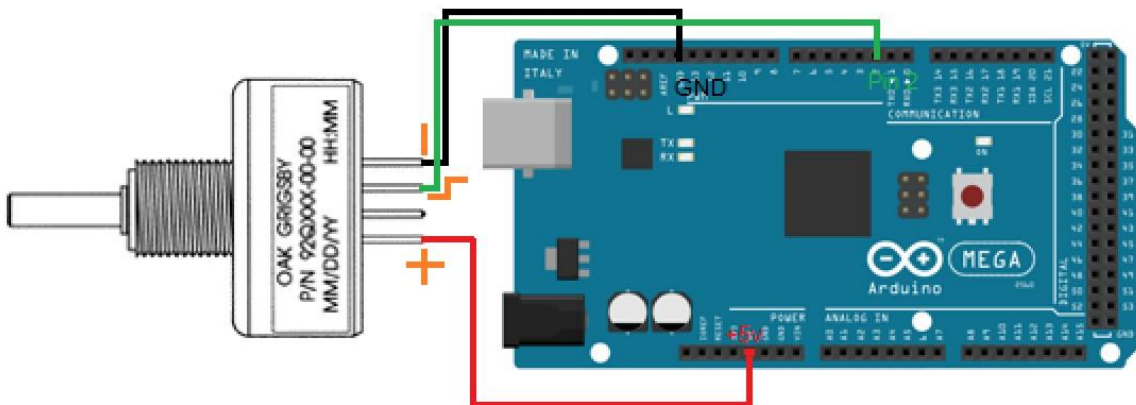
*Fuente:* Elaboración propia

#### 3.1.1.1 Lecturas de sensor encoder del surtidor:

Como ya se explicó en el capítulo anterior, el encoder es un sensor fotoeléctrico que genera pulsos digitales, este puede ser conectado en cualquiera de los pines del arduino en este caso vamos a usar el Pin 2. Dado que es un sensor que solo proporciona pulsos digitales, no es necesaria una librería especial para leer la señal digital. Entonces declaramos que el pin 2 va a ser el pin del encoder.

```
const int encoder = 2; //Pin de entrada de señal del encoder
```

El sensor encoder de los surtidores vienen en diferentes versiones, pero todos tienen en común el envío de señal de pulsos digitales, particularmente los surtidores tokheim usan los encoders Oak Grigsby, estos encoders tienen 4 pines, 2 de señal de pulsos, uno de señal de tierra (GND), y otro de alimentación, estos encoders funcionan igual que un botón Pull up común. Necesita de 5V de alimentación los cuales le puede suplir el Arduino sin problemas, Entonces la conexión quedó de esta manera.



**Figura 3.3:** Conexiones del Encoder con el Arduino Mega

**Fuente:** Elaboración propia.

Ahora se realiza el conteo de pulsos del encoder, el número de pulsos por galón, dado que cada surtidor tiene su proporción personal de pulsos por galón, este valor sería ya encontrado en la implementación del equipo y con unos cálculos matemáticos simples se hallaría el valor el cual multiplicado por el total de pulsos de 1 galón de combustible exacto.

Para esto se crea un bucle el cual lea el valor actual y lo compare con el valor anterior, si existe un cambio de estado de “1” a “0” o de “0” a “1”, entre a un “if”, que lo que suceda es

que si cambió a 1 el contador se suma, de caso contrario no ocurra ningún suceso, para podernos asesorar de que el Encoder esté dando señales correctamente se le pone un marcador, en este caso el led ya incluido en la placa Arduino Mega.

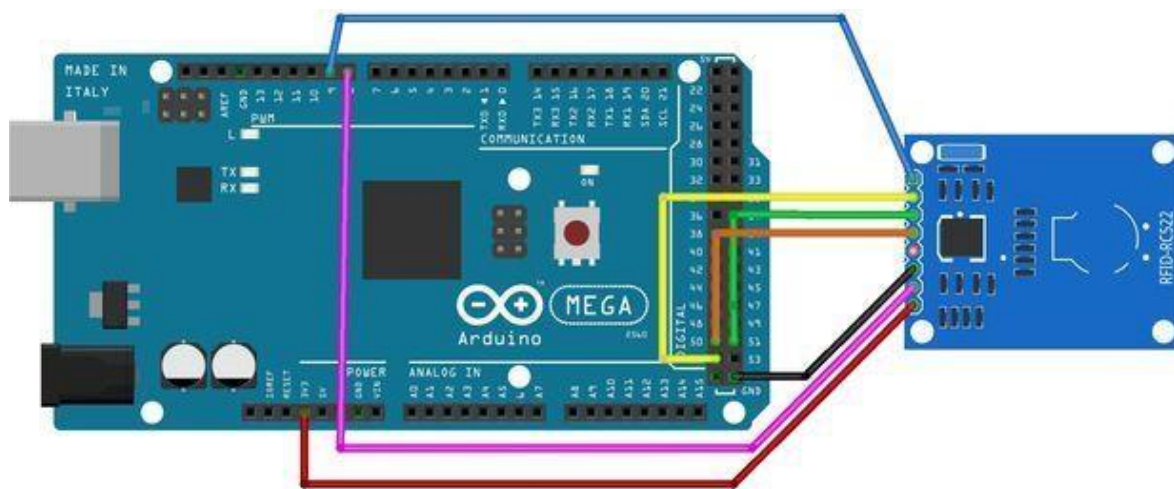
```
valorencoder=digitalRead(encoder); //Leemos el estado del pulsador

if(valorencoder!=estadoAnteriorencoder){ //Si hay un cambio de estado, entramos en el if
  if(valorencoder==1){
    contador++; //Aumentamos en una unidad la cuenta
    Serial.println(Galones); //Imprime el valor por consola
    digitalWrite(led,HIGH); //Encendemos el led
  }
  if(valorencoder==0){
    digitalWrite(led,LOW); //Apagamos el led
  }
}
estadoAnteriorencoder=valorencoder;
Galones=contador*(0.008333333333333333) ;
```

### 3.1.1.2 Lecturas del sensor RFID:

Como ya se examinó en el capítulo anterior, el sensor RFID sirve para la identificación de usuarios, dado que es un módulo más complejo, este va a necesitar pines especiales del Arduino Mega, dado que la comunicación no es la habitual.

En la FIGURA 3.4 se muestra las conexiones necesarias.



*Figura 3.4: Conexiones del módulo RDIF RC522 con Arduino Mega*

*FUENTE: Elaboración propia.*

Mayor detalle sobre las conexiones en la Figura 3.5:

Módulo RC522	Arduino Uno, Nano	Arduino Mega
SDA (SS)	10	53
SCK	13	52
MOSI	11	51
MISO	12	50
IRQ	No conectado	No conectado
GND	GND	GND
RST	9	9
3.3V	3.3V	3.3V

**Figura 3.5:** Tabla de conexiones del módulo RFID RC522 tanto como para Arduino uno como para Arduino Mega

Fuente: [www.naylampmechatronics.com](http://www.naylampmechatronics.com)

Para poder utilizar los correctos comandos de programación, se debe importar librerías para el RFID, estas se pueden encontrar en internet en la página oficial de Arduino [www.arduino.cc](http://www.arduino.cc), la instalación es de fácil entendimiento. Una vez ya instalada la librería, la mencionamos en nuestra programación principal.

```
#include <SPI.h>
#include <MFRC522.h>
```

Luego se define los puertos, en los cuales va conectado el RFID, y se crea un objeto para el RFID, para los 4 pines de conexión SDA, SCK, MOSI, MISO, solo es necesario declarar el ultimo pin ya que la librería del RFID por defecto los declara con sus pines correspondientes.

```

#define RST_PIN 9 //Pin 9 para el reset del RC522

#define SS_PIN 53 //Pin 53 para el SS (SDA) del RC522

MFRC522 mfrc522(SS_PIN, RST_PIN); ///Creamos el objeto para el RC522

```

Ahora declaramos las variables y los contadores.

```

float rnd; // random number

float Galones;

const int encoder = 2; //Pin de entrada de señal del encoder

const int led = 13; //Le ponemos nombre al pin 13

int contador = 0; //Variable para guardar la cuenta de pulsaciones

int estadoAnteriorencoder = 0; //Declaramos e inicializamos la variable

int valorencoder = 0; //Declaramos e inicializamos la variable

```

Iniciamos los periféricos en la función setup:

```

void setup() {
  Serial.begin(9600); //Iniciamos La comunicacion serial
  SPI.begin(); //Iniciamos el Bus SPI
  mfrc522.PCD_Init(); // Iniciamos el MFRC522
  Serial.println("Control de acceso:");
  // if analog input pin 5 is unconnected, random analog noise will cause the call to randomSeed() to
  generate different seed numbers each time the sketch runs
  randomSeed(analogRead(5));
}

```

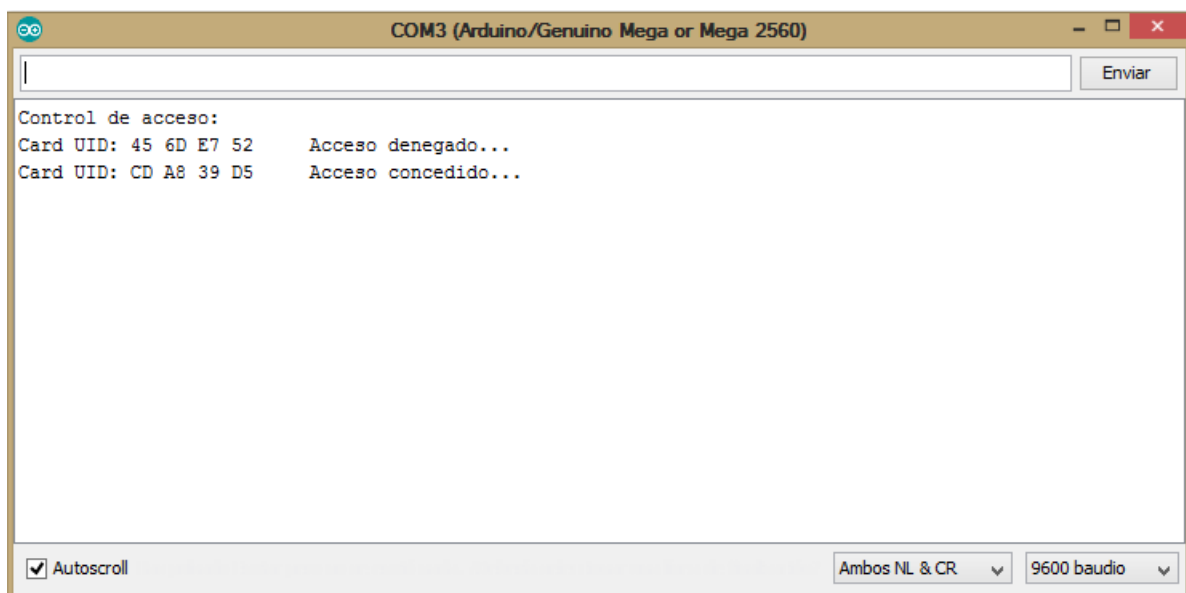


En esta parte declaramos los valores del RFID admitidos, estos valores son sexagesimales y se declaran cada tarjeta por separado.

```
byte ActualUID[4]; //almacenará el código del Tag leído  
byte Usuario1[4]= {0x45, 0x6D, 0xE6, 0x52} ; //código del usuario 1  
byte Usuario2[4]= {0xCD, 0xA8, 0x39, 0xD5} ; //código del usuario 2
```

Ahora se hace la secuencia central del control de acceso

Finalmente tenemos los resultados de la lectura del sensor RFID mediante el monitor serial.



**Figura 3.6:** Control de Acceso a Arduino

*Fuente: Elaboración propia.*

```

// Revisamos si hay nuevas tarjetas presentes
if( mfrc522.PICC_IsNewCardPresent()
{
//Seleccionamos una tarjeta
if( mfrc522.PICC_ReadCardSerial()
{
// Enviamos serialmente su UID
Serial.print(F("Card UID:"));
for( byte i = 0; i < mfrc522.uid.size; i++) {
Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
ActualUID[i]=mfrc522.uid.uidByte[i];
}
Serial.print(" ");
//comparamos los UID para determinar si es uno de nuestros usuarios
if(compareArray(ActualUID, Usuario1))
Serial.print("Acceso concedido...");
else if(compareArray(ActualUID, Usuario2))
Serial.print("Acceso concedido...");
else
Serial.print("Acceso denegado...");

// Terminamos la lectura de la tarjeta actual
mfrc522.PICC_HaltA();
}
}

```

### 3.1.1.3 Programa conjunto de lectura y envío de datos del Encorer y también del RFID:

Primero declaramos las librerías y definimos algunos parámetros que iremos a usar.

```

#include <SPI.h>
#include <MFRC522.h>
#include <TimerOne.h>

#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 53 //Pin 53 para el SS (SDA) del RC522
MFRC522 mfrc522(SS_PIN, RST_PIN); ///Creamos el objeto para el RC522

```

Luego definimos los caracteres según su tipología a usar:

```

float      rnd; // random number
float      Galones;
const int encoder = 2; //Pin de entrada de señal del encoder
const int led = 13; //Le ponemos nombre al pin 13
int contador = 0; //Variable para guardar la cuenta de pulsaciones
int askcontador = 0;
int estadoneutro = 0;
int estadoAnteriorencoder = 0; //Declaramos e inicializamos la variable
int valorencoder = 0; //Declaramos e inicializamos la variable

```

Creamos la función de inicio del microcontrolador declarando algunos parámetros como los baudios de comunicación serial:

```

void setup() {
  Timer1.initialize(1000000); // Dispara cada 1000 ms
  Timer1.attachInterrupt(ISR_ASK); // Activa la interrupcion y la asocia a ISR_ASK
  Serial.begin(9600); //Iniciamos La comunicacion serial
  SPI.begin(); //Iniciamos el Bus SPI
  mfrc522.PCD_Init(); // Iniciamos el MFRC522
  // Serial.println("Control de acceso:"); |
  randomSeed(analogRead(5));
}

```

Ahora creamos una función activada por interrupciones, esta función se activa cada 1 segundo, el tiempo ya está seteado en la función setup, esta función es utilizada para marcar el final del llenado de combustible y enviar los datos vía serial de los galones consumidos:



Leemos las señales del RFID y las enviamos vía serial:

```
// Revisamos si hay nuevas tarjetas presentes
if ( mfrc522.PICC_IsNewCardPresent() )
{
  //Seleccionamos una tarjeta
  if ( mfrc522.PICC_ReadCardSerial() )
  {
    // Enviamos serialmente su UID
    //Serial.print(F("Card UID:"));
    for (byte i = 0; i < mfrc522.uid.size; i++) {
      Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
      Serial.print(mfrc522.uid.uidByte[i],HEX);
      ActualUID[i]=mfrc522.uid.uidByte[i];
    }
    Serial.print(" "); // comparamos los UID para determinar si es uno de nuestros usuarios
    if(compareArray(ActualUID,Usuario1))
      Serial.println("Acceso concedido...");

    else if(compareArray(ActualUID,Usuario2))
      Serial.println("Acceso concedido...");
    else
      Serial.println("Acceso denegado...");

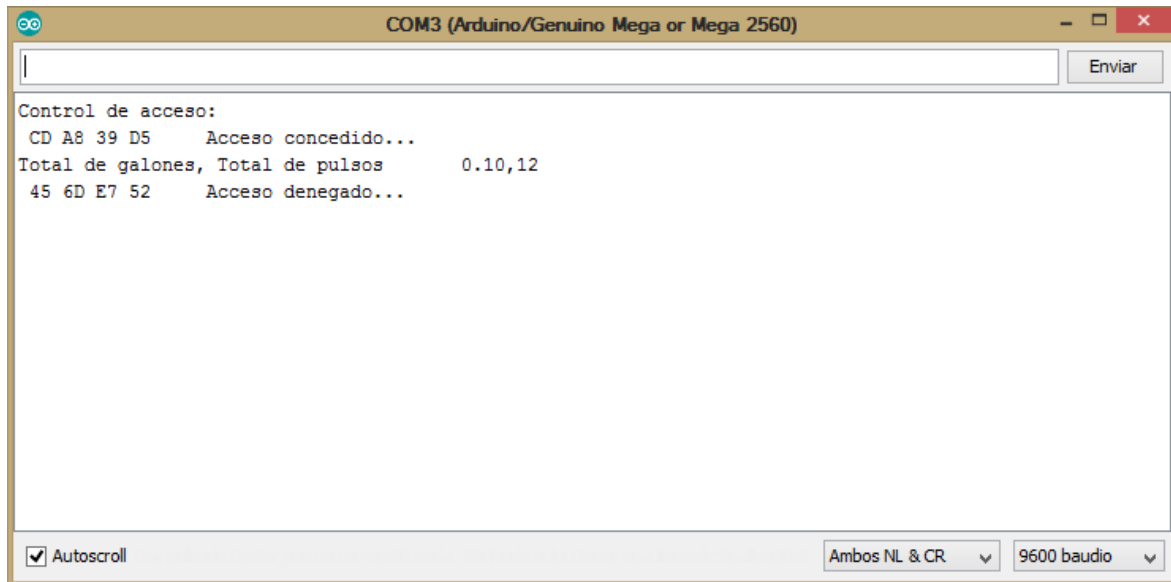
    //Terminamos la lectura de la tarjeta tarjeta actual

mfrc522.PICC_HaltA();
```

Ahora almacenamos el dato de la tarjeta actual para poder compararlo con tarjetas pre-guardadas, y así crear la autenticación.

```
byte ActualUID[4]; //almacenará el código del Tag leído
byte Usuario1[4]= {0x45, 0x6D, 0xE6, 0x52} ; //código del usuario 1
byte Usuario2[4]= {0xCD, 0xA8, 0x39, 0xD5} ; //código del usuario 2
```

La Figura 3.7 nos muestra el resultado de la programación, el control de acceso y el conteo de galones mediante el monitor serial del mismo arduino, en este caso se detectó el arduino en el puerto serial COM3.

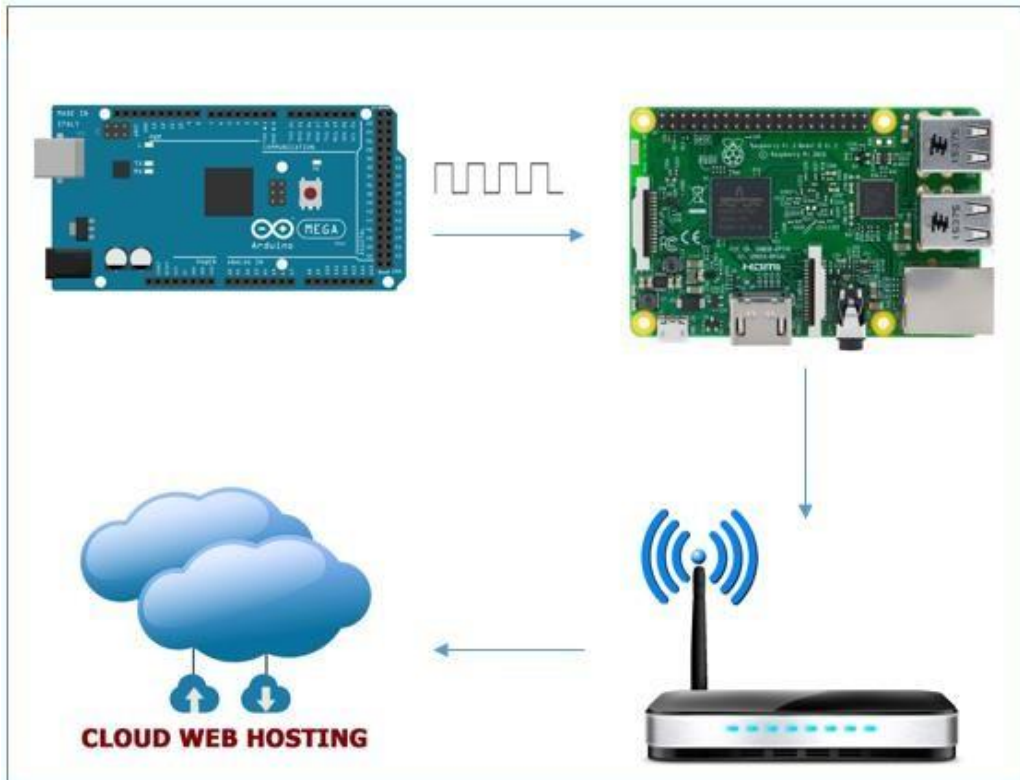


*Figura 3.7: Lectura serial de valores enviados por Arduino*

*Fuente: Elaboración propia.*

### **3.1.2 Envío de datos y procesamiento de datos:**

En esta parte del diseño el sistema se encarga de enviar hacia el Raspberry Pi los datos de las señales adquiridas por el Arduino, esta conexión entre Arduino y Raspberry Pi3 se hace mediante comunicación serial, ya una vez recibidos los datos, pueden guardarlos en una base de datos para que puedan ser enviados hacia un servidor web. Dado que el Raspberry es una microcomputadora, esta se encarga del servidor web. En la figura 3.9 se muestra como fue pensado el diseño, primero enviando datos del arduino al Raspberry pi 3, luego el Raspberry pi3 crea una base de datos, estos datos podrían ser enviados directamente a la web.



**Figura 3.8:** Esquema de conexión desde el Arduino Mega hasta la nube.

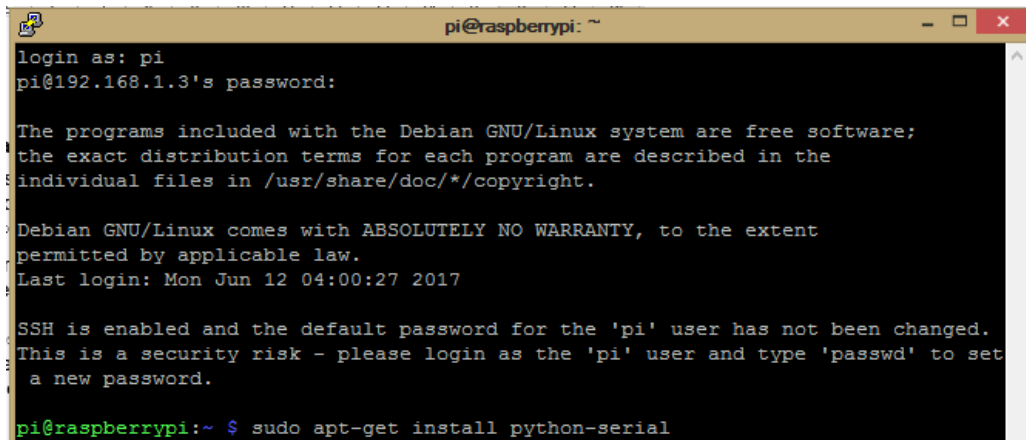
Fuente: Elaboración propia.

### 3.1.1.1 Recibiendo datos del arduino con Raspberry Pi 3

En esta parte del sistema, se hace la transferencia de datos de galones y pulsos, desde el Arduino hacia el Raspberry pi, esta transferencia de datos ha sido hecha por comunicación serial.

Para hacer esta transferencia se ha usado el lenguaje de programación Python, lenguaje de programación por defecto del Raspberry; usamos este lenguaje para crear un programa el cual recepcione los datos vía serial, para esto instalamos librerías antes de poder programar, la librería usada es “*Python-serial*”; en la figura 3.10 podemos ver el comando para la instalación de la librería, esto se hace desde la aplicación “terminal” del Raspberry pi o desde la PC usando

el programa Putty, el cual nos permite tener acceso al Raspberry Pi desde una PC en la misma área local.



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.1.3's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Jun 12 04:00:27 2017  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
pi@raspberrypi:~ $ sudo apt-get install python-serial
```

**Figura 3.9:** Instalación de librería serial para Python vía Putty desde pc.

*Fuente: Elaboración propia*

Una vez ya creada la librería procedemos a crear un archivo en blanco con formato “.py” que es el formato de Python, eso quiere decir que estaremos creando un archivo de programación Python. En la figura 3.10 podemos ver el comando para crear un archivo.



```
pi@raspberrypi:~ $ sudo nano datagaso.py
```

**Figura 3.10:** Creando nuevo archivo formato Python llamado datagaso.py

*Fuente: Elaboración propia*

Ya creado el archivo, se programó la comunicación Arduino-Raspberry; para esto se importó la librería serial, con líneas de comando, se declaró el puerto de entrada de la comunicación serial así como se decodificaron los valores enviados por el Arduino, separando así las 2 variables de la misma línea de comando recibida, como vemos en la figura 3.11.



```
import serial
ser = serial.Serial('/dev/ttyACM0',9600)
startTime =datetime.now()
line=ser.readline().rstrip()
galones,pulsos=line.split(",")
```

*Figura 3.11: Comandos para la comunicación serial entre Arduino y Raspberry Pi 3.*

*Fuente: Elaboración propia*

Con estos comando se obtuvieron las variables, galones y pulsos por separado, ya obtenido estas variables por separado, se pudo maniobrar estas y formar una base de datos.

### **3.1.1.2 Almacenando las variables en una base de datos.**

Una vez que ya se reconocieron los valores de galones y pulsos enviados por el Arduino, se pudo crear una base de datos para estos, con fecha y hora.

Para poder crear una base de datos, primero se tuvo que instalar las librerías de Python “Python-MySQLdb”, y también los programas “MySQL”, “Apache”, “phpMyadmin”.

Una vez instaladas las librerías se prosiguió a crear la base de datos también a crear los arreglos para que las variables “galones” y “pulsos” sean guardadas correctamente. Para esto modificamos el archivo ya creado de comunicación serial “datagaso.py”.

En primera instancia importamos las librerías necesarias para crear la base de datos basada en MySQL como se muestra en la figura 3.12.

```
import serial
import time
import MySQLdb as mdb
from time import strftime
from datetime import datetime, time
```

**Figura 3.12:** Declaración de librerías a usar para crear una base de datos.

*Fuente: Elaboración propia*

Luego se procedió a la conexión a la base de datos de MySQL. Esto hizo que se puedan enviar las variables a la base de datos en MySQL. El usuario por defecto es “root”, la contraseña es “passuni12” y la base de datos creada fue “Akin”. La imagen siguiente muestra cómo se debe programar la conexión a la base de datos MySQL.

```
con = mdb.connect('localhost','root','passuni12','Akin')
```

**Figura 3.13:** Conexión a la base de datos.

*Fuente: Elaboración propia*

Si estos datos de autenticación no están correctamente puestos el programa no se va a conectar a la base de datos.

Se procedió a importar la Fecha y Hora, para saber cuándo fue abastecido el combustible como muestra la figura 3.14 a continuación.

```
now = datetime.now()
elapsedTime = now-startTime
elapsedSeconds = (elapsedTime.microseconds+(elapsedTime.days*24*3600+elapsedTime.seconds)*10**6)/10**6
```

**Figura 3.14:** Comandos para importar Hora y Fecha de la PC

*Fuente: Elaboración propia*

Por último se programó los valores que se van a exportar a la base de datos, estas variables serán almacenados en la tabla “Gasolina” como muestra la figura 3.15 a continuación.

```

with con:
    cursor = con.cursor()
    cursor.execute("""INSERT INTO Gasolina VALUES(%s,%s,%s,%s) """, (now.strftime("%Y-%m-%d %H:%M:%S"), elapsedSeconds, galones, pulsos))
    con.commit()
    cursor.close()

```

**Figura 3.15:** Comandos de declaración de variables a exportar a la base de datos.

*Fuente: Elaboración propia*

### 3.1.1.3 Configuración de base de datos vía phpMyadmin

En este apartado, se explica la configuración de la base de datos vía web, se configura los nombres de la tabla y el tipo de carácter que es cada uno.

Se creó una nueva base de datos llamada “Akin”, esta base de datos fue mencionada en la figura 3.13; debió tener el mismo nombre para que las variables se direccionen hacia esa base de datos. Luego se creó la tabla “Gasolina”, mencionada en la línea de comando de declaración de variables (figura 3.15), al igual que la base de datos, debió mencionar la tabla con el mismo nombre para que las variables se direccionen, en la figura 3.16 se muestra ya creación de la base de datos y la tabla.

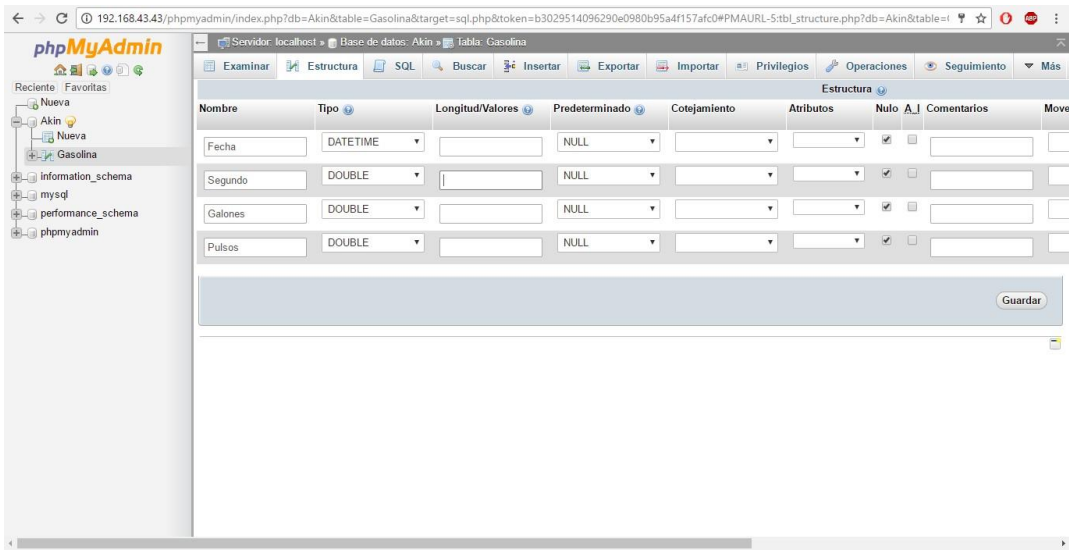


**Figura 3.16:** Base de datos y tabla de valores.

*Fuente: Elaboración propia*

Luego se configuro las variables de la tabla, según el orden que fueron mencionadas en la línea de comandos de la figura 3.16. Primero la fecha luego el segundo, los galones y los pulsos.

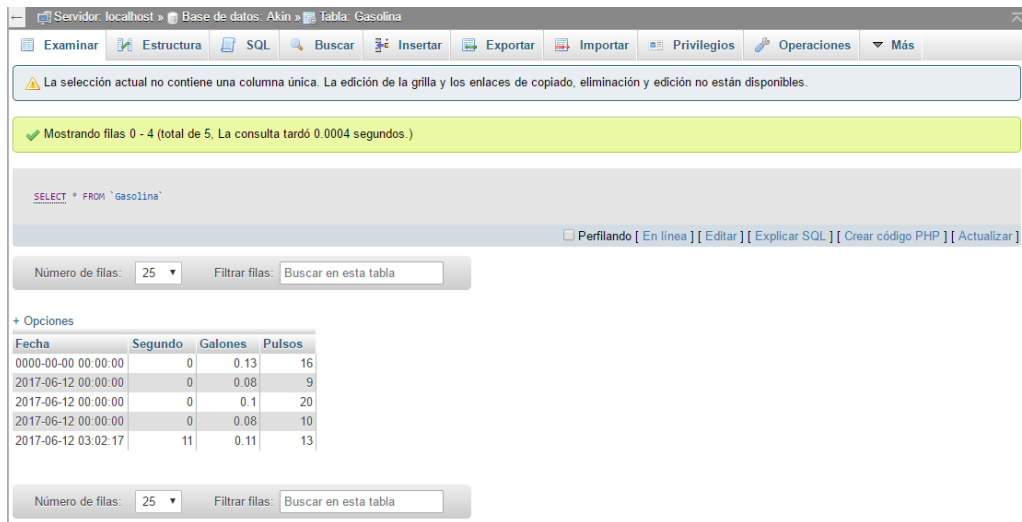
Esta configuración la muestra la figura 3.17



**Figura 3.17:** Configuración de valores de la tabla “Gasolina”

Fuente: Elaboración propia

Luego se hicieron las pruebas del sistema y se pudo observó cómo efectivamente se guardaron las variables en las tablas.



**Figura 3.18:** Tabla de valores de la base de datos.

Fuente: Elaboración propia

### **3.2 Diseño de conexión eléctrica y empotramiento del sistema:**

En esta parte del proyecto se realizó el correspondiente diseño mecánico eléctrico del sistema, para esto se utilizó el programa de diseño en 3D Solidworks™; se realizó un estudio del caso de empotramiento del sistema en gasolineras e industrias privadas, en el cual se concluyó que estos equipos deben seguir ciertas normas de seguridad como la normativa técnica peruana (NTP) y (NFPA), dado que están cerca de sustancias inflamables como los combustibles.

Por lo tanto se diseñó el sistema con las peculiaridades del caso, que se desarrollaron en dos fases descritas a continuación:

- a) Conexiones eléctricas.
- b) Estructura mecánica.

#### **3.2.1 Conexiones eléctricas:**

##### **3.2.1.1 Fuente de alimentación:**

Dado que el sistema es electrónico, este funciona a bajo voltaje, por lo tanto necesitaríamos de una fuente de alimentación regulada, el cual convierta el voltaje de la línea de 220V, a un voltaje de 5V que es lo requerido por el sistema electrónico. Para este diseño se utilizó una fuente conocida comercialmente como fuente para LEDs, el cual nos brinda el voltaje requerido con una fiabilidad alta.



**Figura 3.19:** Foto fuente de alimentación.

*Fuente: Elaboración propia*

### **3.2.1.2 Cableado eléctrico:**

Como ya se mencionó el sistema será instalado en un ambiente peligroso e inflamable, usualmente en la intemperie, por lo tanto se requiere un cableado con tubería y sello antiexplosivo Conduit de 1" de diámetro (figura 3.20), en el cual se ubica el conductor de alimentación tipo THW-90°C calibre 12 AWG, conductor el cual debe ser alimentado con una tensión de 220 V desde el tablero eléctrico y a su vez la señal del sensor encoder del surtidor.



**Figura 3.20:** Foto de tubería conduit.

Fuente: <http://technogroup.com.pe/aplicaciones-tuberias-conduit-y-accesorios.html>

El sistema también requiere un interruptor termomagnético 10A como protección a una elevación de corriente como muestra la figura 3.21.



**Figura 3.21:** Foto fuente de interruptor termostático 10A.

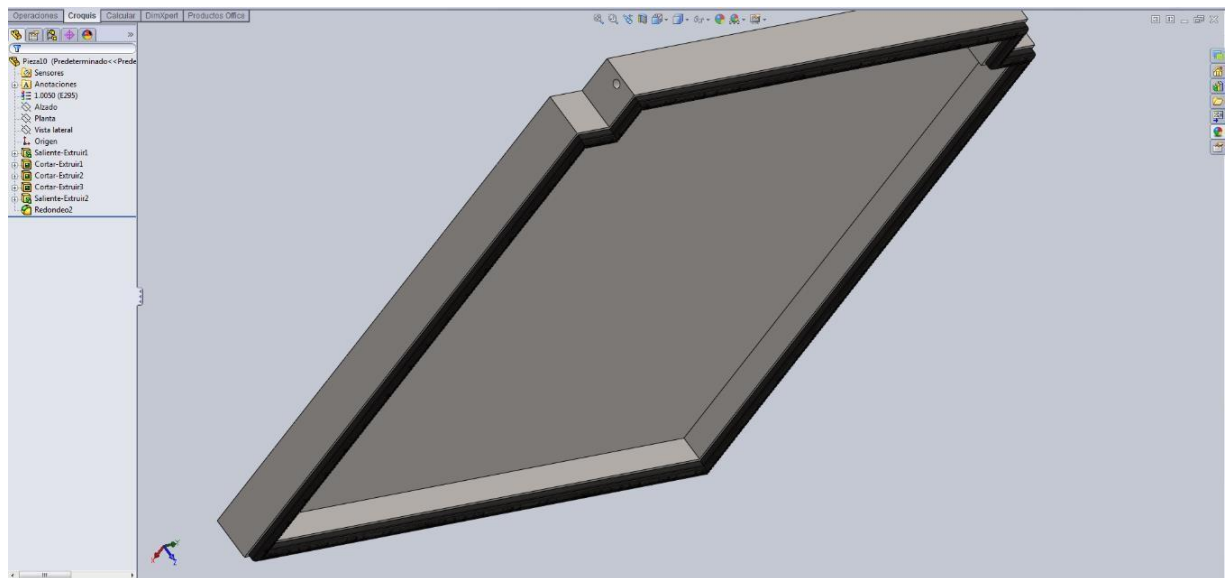
Fuente: [http://listado.mercadolibre.com.ar/interruptor-termomagnetico-10a-440v-bipolar\\_DisplayType\\_G](http://listado.mercadolibre.com.ar/interruptor-termomagnetico-10a-440v-bipolar_DisplayType_G)

### 3.2.2 Estructura mecánica:

#### 3.2.2.1 Estructura Externa:

En la elaboración de la estructura, se tuvo en cuenta el índice de protección IP 5 como dicta la normativa técnica peruana (NTP), lo cual nos indica que esta estructura debe ser resistente al polvo, a los chorros de agua y a los choques mecánicos medios.

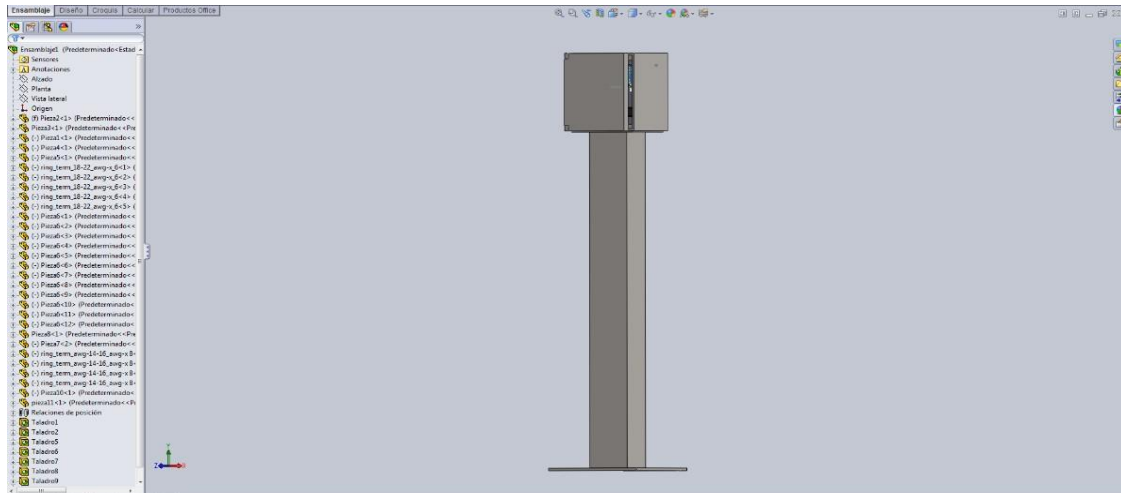
La cual nos obligó a que el diseño de la estructura tenga que tener ciertas particularidades, como lo es una junta tórica para la protección contra el agua y el polvo, base rígida para los choques mecánicos, y de material ferroso.



*Figura 3.22: Diseño de tapa de Protección con junta tórica.*

*Fuente: Elaboración propia*





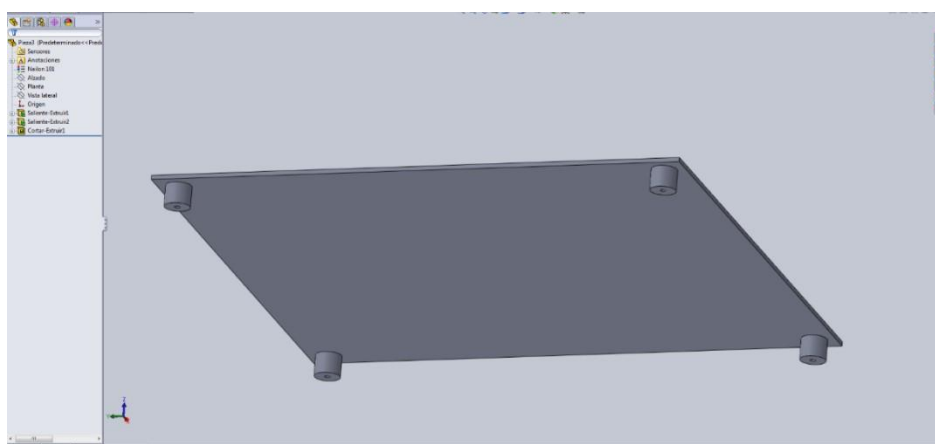
**Figura 3.23:** Diseño de la estructura de Base de Hierro

Fuente: Elaboración propia

### 3.2.2.2 Estructura interna:

La estructura interna consta de los elementos principales del sistema electrónico como también de un aislamiento de aluminio, este diseño está basado en un tablero eléctrico, en el cual están implementados los siguientes componentes:

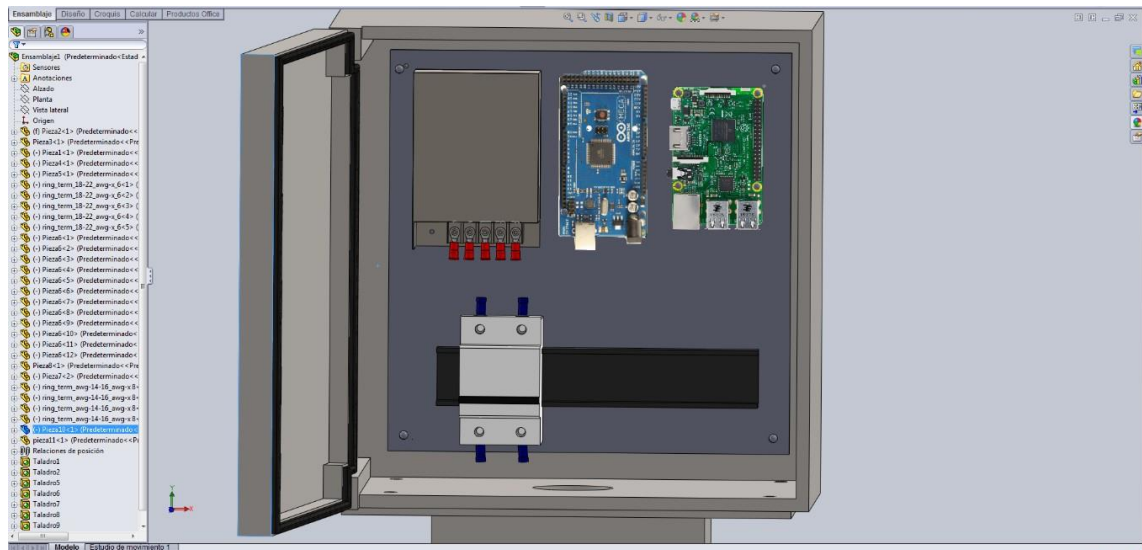
- Plancha de aluminio: Este elemento ha sido diseñado para ensamblarse dentro de la estructura interna para el aislamiento entre las paredes de hierro y los componentes electrónicos.



**Figura 3.24:** Placa de aluminio simulada en Solidworks™ dimensiones 25 x 25cm.

Fuente: Elaboración propia

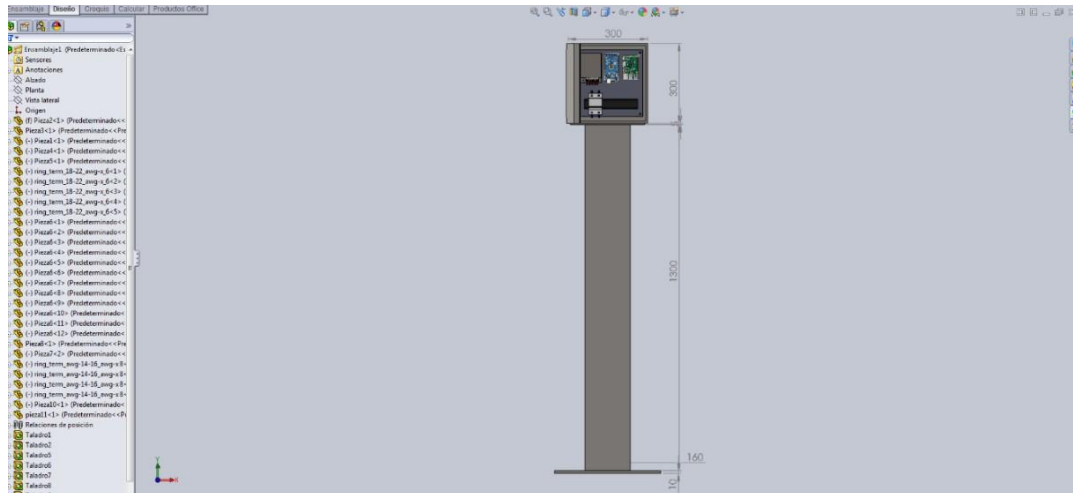
- Fuente regulada de 5V
- Interruptor termomagnético
- Riel
- Arduino Mega
- Raspberry Pi 3



**Figura 3.25:** Diseño de la estructura interna del equipo simulado en Solidworks™

Fuente: Elaboración propia

Se procedió a ensamblar las partes de la estructura interna y externa, mostrando así el diseño completo de la estructura en la cual serán instalados los componentes electrónicos y el cableado eléctrico, esta estructura en general cumple con las normas técnicas peruanas; la estructura del sistema estaría diseñada como se muestra en la figura 3.26.



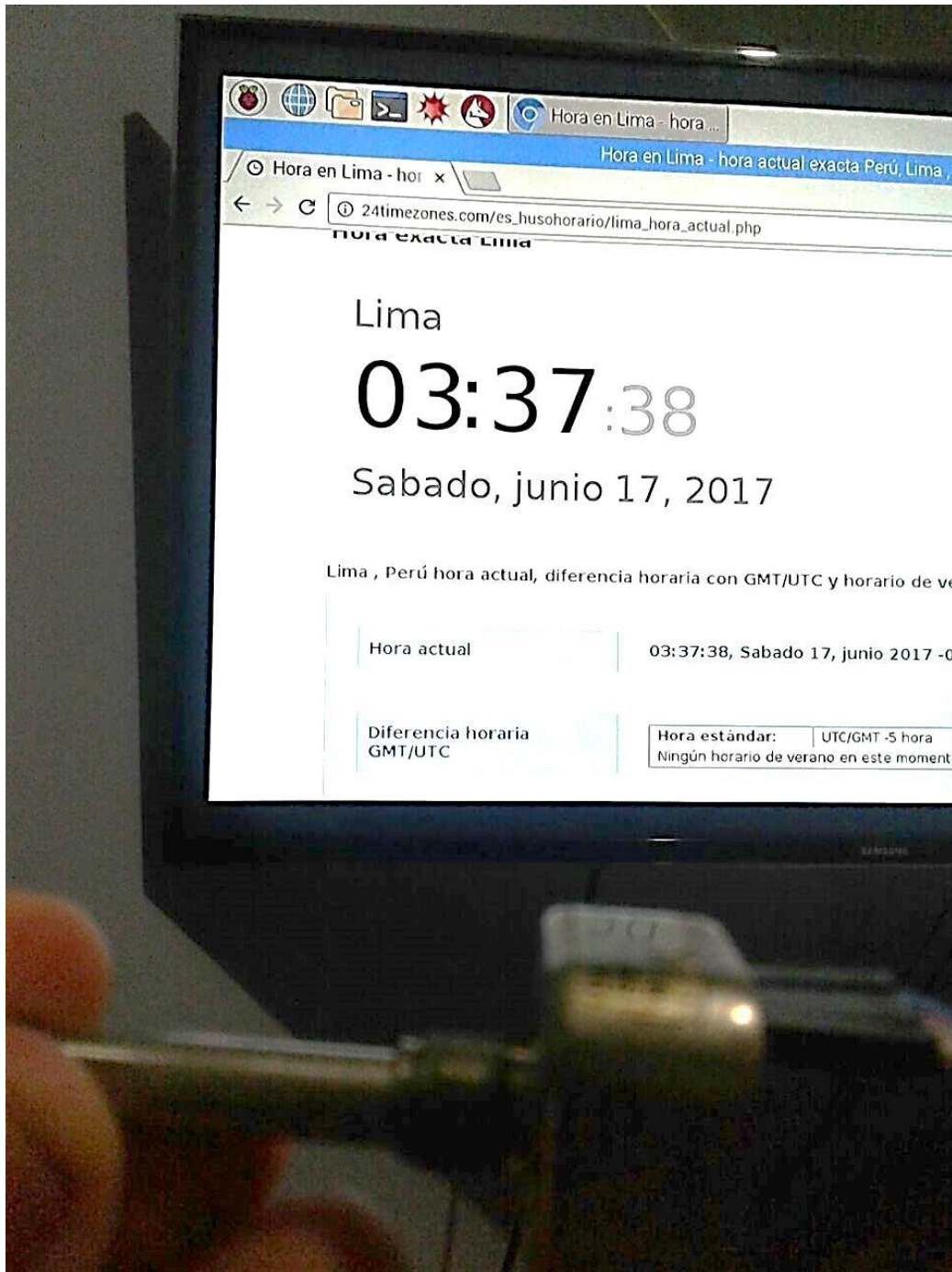
**Figura 3.26:** Diseño de la estructura completa del equipo simulado en Solidworks™

*Fuente: Elaboración propia*

### 3.2.3 Pruebas y Resultados

Se realizaron pruebas del sistema con fecha y hora a tiempo real de la recepción de señales y la base de datos. El surtidor de combustible es simulado por el sensor encoder el cual es el mismo sensor utilizado en los surtidos de combustibles. Al cual se le dieron giros manualmente simulando la acción real del medidor volumétrico de combustible.

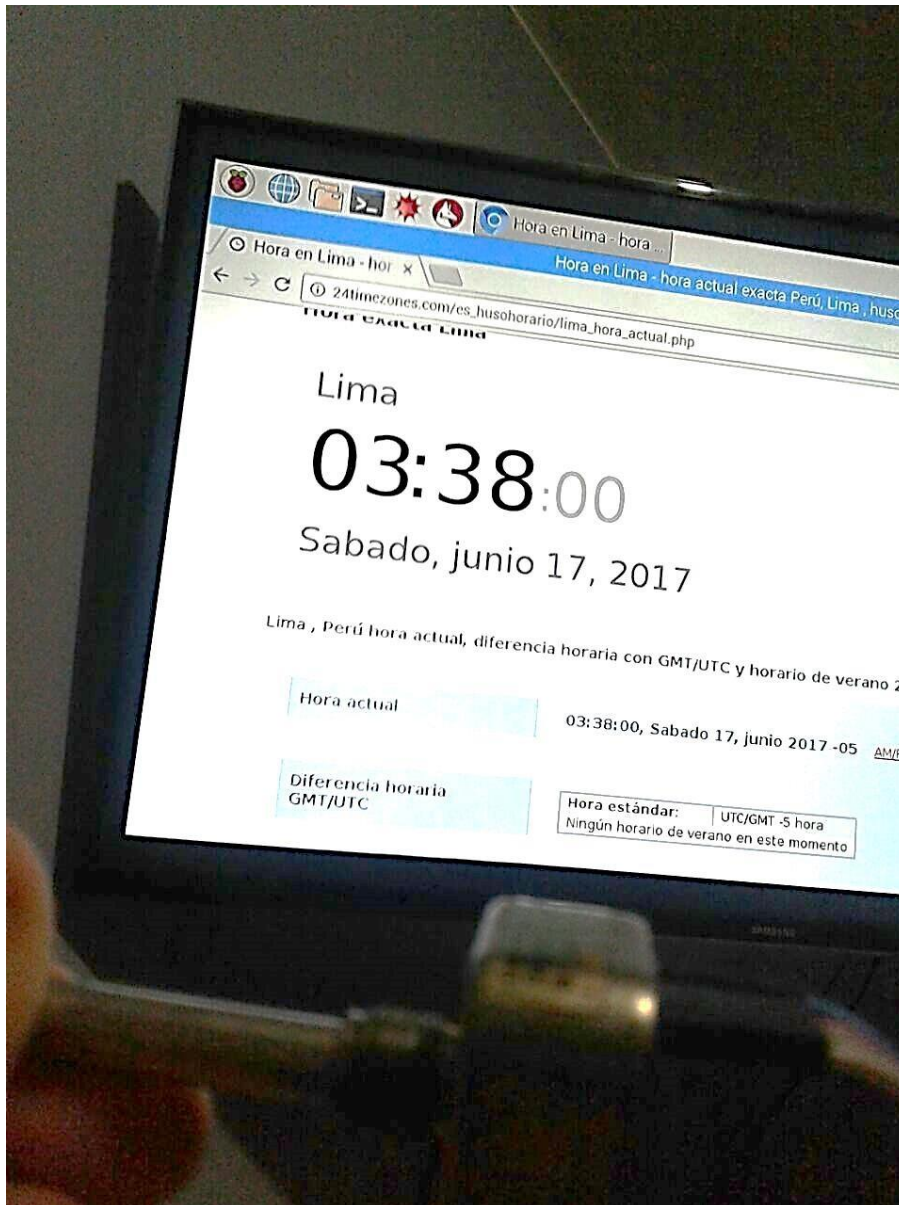
Se hizo una prueba moviendo manualmente el encoder, donde se dieron vueltas al encoder desde las 3:37:38 am hasta las 3:37:55 am. Como se muestra en las figuras 3.27 y 3.28.



**Figura 3.27:** Hora de inicio de prueba.

*Fuente:* Elaboración propia

A continuación se muestra la simulación y la hora de envío de datos. Como podemos observar en la Figura 3.28.



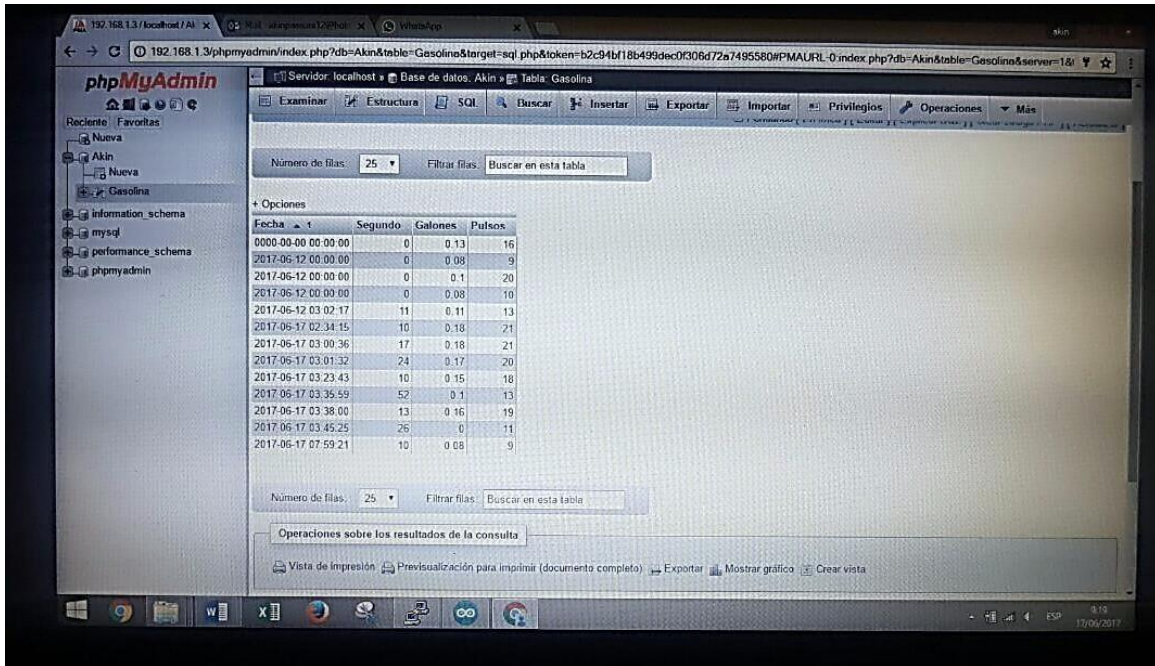
**Figura 3.28:** Hora de envío de señal del encoder.

*Fuente:* Elaboración propia

Ya terminado la prueba del encoder, el sistema logro guardar los datos automáticamente en la base de datos web, esta prueba de visualización de datos vía web, se hizo con una computadora conectada a la misma red Ethernet que el Raspberry Pi3 en el cual está configurado la base de datos, demostrando así los que datos fueron guardados exitosamente a

la hora exacta en la cual se hizo la prueba y estos también pudieron ser visualizados correctamente (figura 3.29) en una matriz en la cual nos indica:

- Fecha y hora en el cual el surtidor abasteció combustible.
- Galones suministrados por el surtidor de combustible.
- Pulsos del sensor encoder instalado en el surtidor.



Fecha	Segundo	Galones	Pulsos
0000-00-00 00:00:00	0	0.13	16
2017-06-12 00:00:00	0	0.08	9
2017-06-12 00:00:00	0	0.1	20
2017-06-12 00:00:00	0	0.08	10
2017-06-12 03:02:17	11	0.11	13
2017-06-17 02:34:15	10	0.18	21
2017-06-17 03:00:36	17	0.18	21
2017-06-17 03:01:32	24	0.17	20
2017-06-17 03:23:43	10	0.15	18
2017-06-17 03:35:59	52	0.1	13
2017-06-17 03:38:00	13	0.16	19
2017-06-17 03:45:25	26	0	11
2017-06-17 07:59:21	10	0.08	9

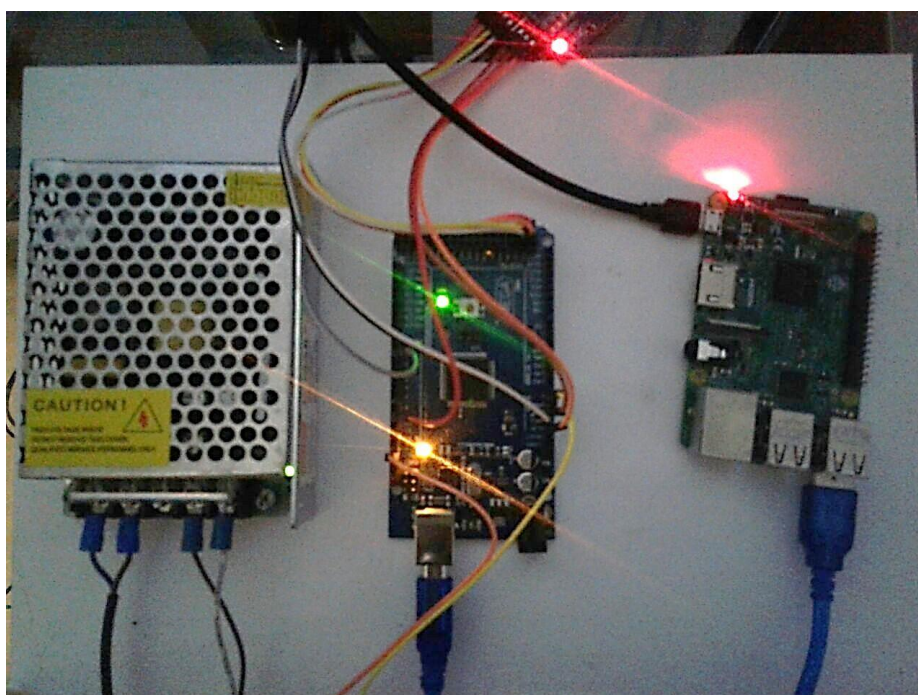
**Figura 3.29:** Visualización de la base de datos desde una portátil en la misma red.

*Fuente:* Elaboración propia

Las pruebas se realizaron con éxito cumpliendo los objetivos planteados. Demostrando que el sistema funciona correctamente.

Finalmente se hizo la prueba de conexión de los dispositivos electrónicos, demostrando que estos funcionan correctamente.

Esta prueba se hizo con la fuente de alimentación ya mencionada en el índice de conexiones eléctricas conectada a la línea eléctrica de 220V, no se necesitó de fuentes de alimentación externas a esta, se comprobó así que los dispositivos electrónicos funcionan correctamente con las conexiones diseñadas para este sistema, estando a la izquierda la fuente de alimentación, al centro la tarjeta electrónica Arduino Mega y en su derecha la microcomputadora Raspberry Pi3 como se muestra en la figura 3.30.



**Figura 3.30:** Visualización de la conexión de dispositivos del sistema.

*Fuente: Elaboración propia*

### 3.3 Análisis de costos de prototipo

Para el análisis de costos se tomó en cuenta los precios de mercado peruano, ya que la importación si bien puede ser más económica es más riesgosa.

Equipo	Cantidad	Costo	
Fuente de alimentación 220VAC - 5VDC	1	S/.	45.00
Arduino Mega 2560	1	S/.	80.00
Raspberry Pi3	1	S/.	250.00
Interruptor termomagnético ABB	1	S/.	40.00
Estructura metálica de empotramiento	1	S/.	1,900.00
Costo de instalación	1	S/.	2,664.00
Costo de Mantenimiento (Anual)	1	S/.	1,665.00
	Total	S/.	6,644.00

Tabla 1. Tabla de costos del prototipo del sistema

De la tabla se concluye que el prototipo inicial del sistema es mucho más económico que cualquier sistema de igual funcionalidad del mercado, dado a que en este sistema se usó las herramientas de Open software, para poder competir con los equipos del mercado actual.

Esta tabla de costos cuenta también con la estructura metálica, cosa que otros fabricantes no ofrecen a los clientes en sus ofertas de venta.

### 3.4 Consumo de energía del sistema

En este punto se tomaron las mediciones de corriente a la entrada de cada elemento electrónico del sistema con un multímetro, primero se toma cuando el sistema está en reposo y luego cuando el sistema está en pleno funcionamiento.



Equipo	Consumo en reposo (mA)	Consumo en funcionamiento (mA)	Voltaje de alimentación (V)	Potencia total en reposo (Wh)	Potencia total en funcionamiento (Wh)
Arduino Mega 2560	93	159	5 VDC	0.465	0.795
Raspberry pi 3B	1500	2000	5 VDC	7.5	10
Fuente de alimentación	1.590909091	49.10433884	220 VAC	8.037954545	10.80295455
Encoder	0	0	5 VDC	0	0
RFID	13	26	3.3 VDC	0.0429	0.0858

Tabla 2. Consumo de energía del sistema

Como se observa en la tabla el consumo del sistema está dado en Watts hora, el consumo total del sistema está dado por la fuente de alimentación dado que este suministra energía a los demás equipos, por lo cual el sistema alcanzaría un máximo de consumo hora de 10.80 Whatts/hora.

Para ver mejor la diferencia de cuanto consume el sistema en reposo y de cuanto consume en funcionamiento tenemos la siguiente gráfica.

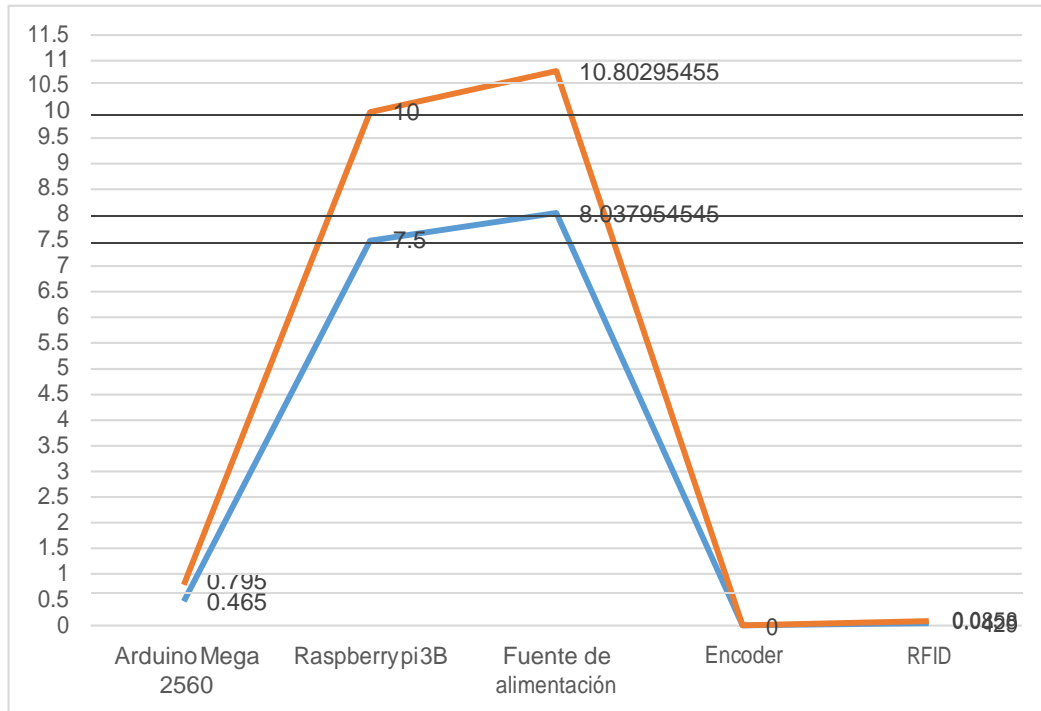


Tabla 3. Consumo de energía en reposo y en funcionamiento.

### 3.5 Diagrama de gantt

El sistema se elaboró desde su primera fase el 15 de diciembre y según los cálculos debería estar en prueba hasta el 20 de septiembre del 2017, para hacer los ajustes de un posible prototipo final.

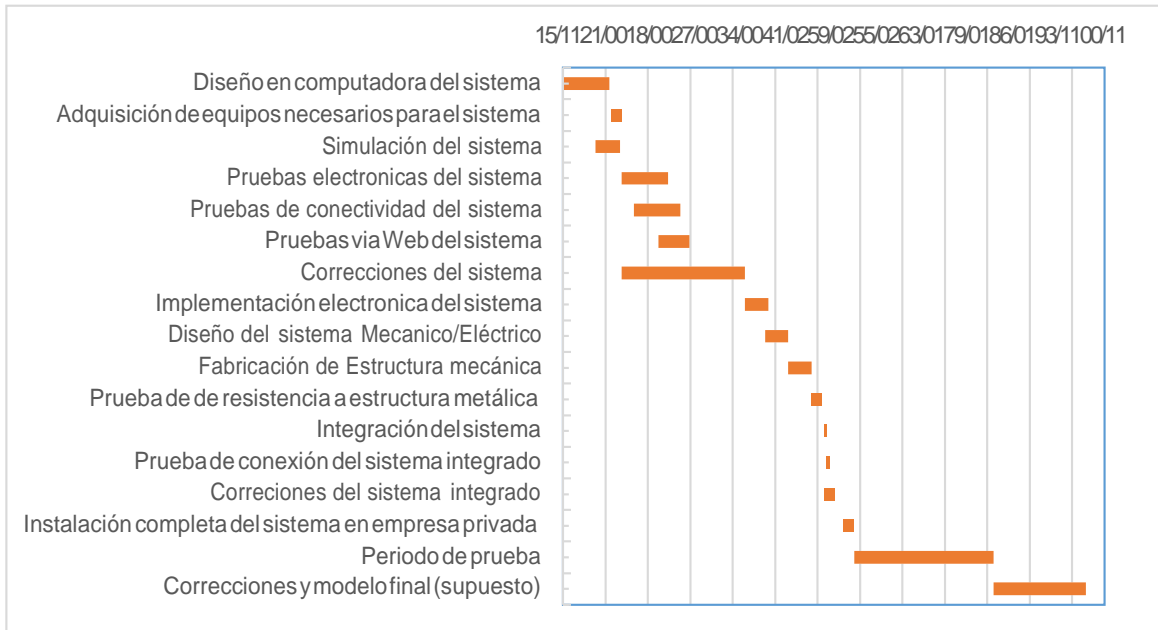


Tabla 4. Diagrama de Gantt o cronograma de creación del sistema

Como se ve en la tabla 3, el cronograma está pensado desde el inicio de la creación del sistema hasta un posible modelo final ya para poder ser patentado, el sistema el sistema necesita un periodo de prueba de por lo menos 3 meses, los cuales están realizados en una empresa privada donde se abastece combustible, y para el final de este año debería haber un prototipo del sistema ya más conciso y seguro.

## 4 Conclusiones

Después del desarrollo del presente proyecto de investigación hemos llegado a las siguientes conclusiones:

1. Se diseñó un sistema de administración y automatización de surtidores de combustible para la empresa Empex usando como herramienta principal el Raspberry Pi3, que consiste en un mejoramiento de tecnología con fin de optimizar la administración de las empresas que abastezcan combustible líquido mediante surtidores de combustible, pudiendo solucionar uno de los problemas principales de estas empresas.
2. Existen escasos antecedentes sobre los diseños para mejorar los surtidores de combustible orientado a la automatización digital.
3. Es importante tener conocimientos de las herramientas electrónicas para el almacenamiento de información fiable, esto ayuda a la correcta toma de decisiones, en cualquier empresa que su nueva visión sea la automatización.
4. Se ha diseñado también la estructura mecánica para la correcta instalación y protección del Sistema de acuerdo a las normativas y estándares de seguridad normalizadas por el estado peruano.
5. Se logró el objetivo buscado en la investigación, ya que este proyecto es viable y busca solucionar el problema desde un sistema diseñado en nuestro país.

## 5 Recomendaciones

Para el complemento de esta tesina tenemos una serie de Recomendaciones que serían factibles de aplicación:

1. Se recomienda hacer un plano base del área en el cual va a ser instalado el sistema con su debida estructura, ya que en caso contrario podría haber accidentes mayores y pérdidas de recursos económicos, el espacio recomendado es un metro de distancia entre el surtidor de combustible y la estructura del sistema.
2. Es recomendable tomar las correctas precauciones de la alimentación eléctrica del sistema ya que la fuente diseñada para este caso es un regulador de 220VAC a 5VDC.
3. Es recomendable que el sistema esté operando con la estructura de protección correctamente cerrada.
4. Es recomendable encender el equipo solo si el sistema está correctamente conectado con todas las medidas de seguridad dadas por la normativa técnica peruana.
5. Finalmente tener en cuenta que el sistema tiene una comunicación vía Wi-fi por lo que se deberá tomar todas las precauciones del caso, como es la distancia desde el sistema hacia una Red Ethernet, dado que la base de datos se guarda en el mismo sistema, al haber un fallo en el sistema wi-fi, estos datos se volverán a actualizar una vez haya reanudado la conexión correctamente, por lo que el sistema es a prueba de interferencias.

## 6 Bibliografía

- Andrade Rodriguez, J. W. (Octubre de 2001). Tesis Modernización de Estaciones de Servicio de Combustible mediante controladores logicos programables. . *Modernización de Estaciones de Servicio de Combustible mediante controladores logicos programables*. Quito, Ecuador: Escuela Politecnica Nacional.
- Gonzales Daza, E. (2015). Tesis Red de Sensores - Internet de las cosas. *Red de Sensores - Internet de las cosas*. Valencia, España: Universidad Politecnica de Valencia.
- González García, I. T. (2015). Tesis Diseño e Implementación de Sistema Interactivo de informacion de Docentes, con Raspeberry Pi. *Diseño e Implementación de Sistema Interactivo de informacion de Docentes, con Raspeberry Pi*. Guayaquil, Quito: Universidad Politecnica Sede Guayaquil.
- Guerra Ruiz, F. (Noviembre de 2013). Tesis Diseño de un Sistema de Control Domótico y Video Vigilancia Supervisado por un teléfono movil. *Diseño de un Sistema de Control Domótico y Video Vigilancia Supervisado por un teléfono movil*. Lima, Lima, Lima: PUCP.
- Herrera Acosta, E. (2014). Tesis Estudio, diseño e implementación de un prototipo de entrenador de módulo Raspeberry Pi. *Estudio, diseño e implementación de un prototipo de entrenador de módulo Raspeberry Pi*. Quito, Ecuador: Universidad Tecnologica Israel.
- Llontop Diaz, G. C. (2015). Tesis Propuesta de diseño para implementar de un servidor void con asterisk y raspeberry Pi en una oficina de Villa Maria del Triunfo. *Propuesta de diseño para implementar de un servidor void con asterisk y raspeberry Pi en una oficina de Villa Maria del Triunfo*. Villa el Salvador, Lima, Lima: UNTELS.
- Morris, E., Diaz, J., Marco, E., & Montenegro, C. (2010). *Comercializacion de Combustibles modelo de solucion tecnologica*. Lima: Esan Ediciones.

Ñahue Niquen, J. L. (2015). Tesis Modelo de abastecimiento en el Sector Público Peruano. *Modelo de abastecimiento en el Sector Público Peruano*. Lima, Lima, Peru: Universidad Nacional Mayor de San Marcos.

Ramos Figueroa, J. A. (2017). Tesis Diseño e Implementación de un prototipo de medición de potencial espontaneo para el monitoreo de volcan. *Diseño e Implementación de un prototipo de medición de potencial espontaneo para el monitoreo de volcan*. Piura, Piura, Peru: Universidad de Piura.

Real Academia de Ciencias. (23 de Mayo de 2017). *Real Academia de Ciencias*. Obtenido de Real Academia de Ciencias: [http://www.rac.es/1/1\\_1.php](http://www.rac.es/1/1_1.php)

Robbins, S., & Coulter, M. (2005). *Administración*. Mexico: Pearson Educación de México, S.A.

Fuentes:

[8] Almazán Blanca. (2008, Abril 9). *Automatización y robótica para la producción*.

Recuperado de <https://www.gestiopolis.com/automatizacion-robotica-produccion/>

[9] <http://www.itlp.edu.mx/publica/tutoriales/INTERPRETACION> consultada 30-05-2017

## 7 Anexos



*Anexo 1- Foto: Diseño de la estructura completa del equipo en físico.*

*Fuente: Elaboración propia*





*Anexo 2- Foto: Diseño de la estructura completa del equipo en físico.*

*Fuente: Elaboración propia*



*Anexo 3- Foto: La estructura completa del equipo en físico.*

*Fuente: Elaboración propia*



*Anexo 4- Foto: Conexiones eléctricas hacia el equipo.*

*Fuente: Elaboración propia*

## Anexo 5

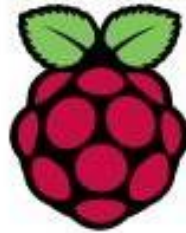
### Especificaciones técnicas del Arduino Mega:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

**Anexo 6**

Datasheet del Raspberry Pi3

**DATASHEET**



**Raspberry Pi Compute Module (CM1)**

**Raspberry Pi Compute Module 3 (CM3)**

**Raspberry Pi Compute Module 3 Lite (CM3L)**

**Version 1.0, October 2016**

**Copyright 2016 Raspberry Pi (Trading) Ltd. All rights reserved.**



## 2 Features

### 2.1 Hardware

- Low cost
- Low power
- High availability
- High reliability
  - Tested over millions of Raspberry Pis Produced to date
  - Module IO pins have 35u hard gold plating

### 2.2 Peripherals

- 48x GPIO
- 2x I2C
- 2x SPI
- 2x UART
- 2x SD/SDIO
- 1x HDMI 1.3a
- 1x USB2 HOST/OTG
- 1x DPI (Parallel RGB Display)
- 1x NAND interface (SMI)
- 1x 4-lane CSI Camera Interface (up to 1Gbps per lane)
- 1x 2-lane CSI Camera Interface (up to 1Gbps per lane)
- 1x 4-lane DSI Display Interface (up to 1Gbps per lane)
- 1x 2-lane DSI Display Interface (up to 1Gbps per lane)

### 2.3 Software

- ARMv6 (CM1) or ARMv7 (CM3, CM3L) Instruction Set
- Mature and stable Linux software stack
  - Latest Linux Kernel support
  - Many drivers upstreamed
  - Stable and well supported userland
  - Full availability of GPU functions using standard APIs



### 3 Block Diagram

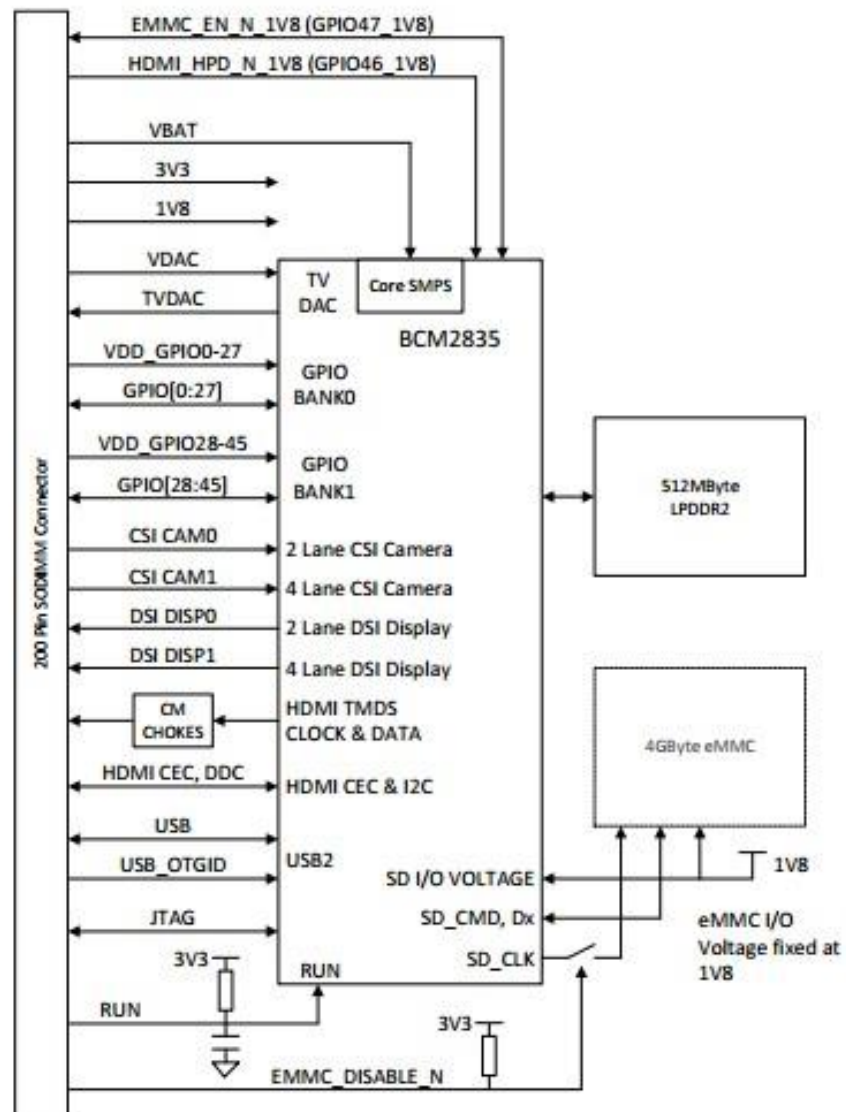


Figure 1: CM1 Block Diagram

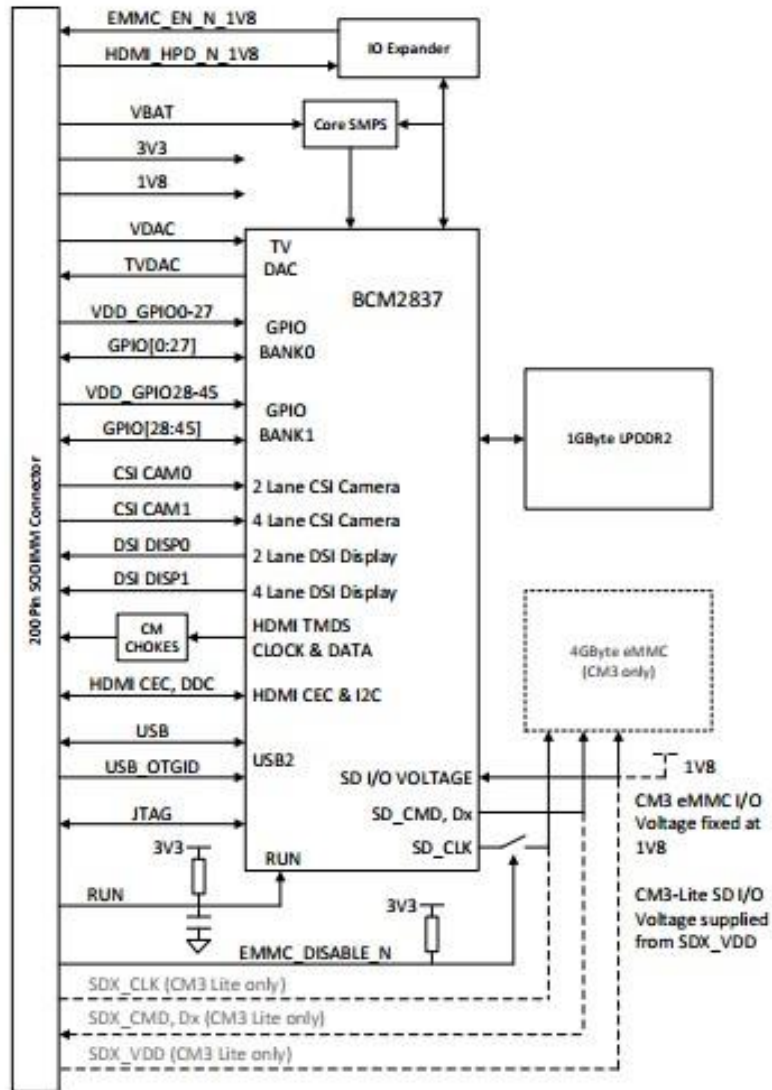


Figure 2: CM3/CM3L Block Diagram





## 4 Mechanical Specification

The Compute Modules conform to JEDEC MO-224 mechanical specification for 200 pin DDR2 (1.8V) SODIMM modules (with the exception that the CM3, CM3L modules are 31mm in height rather than 30mm of CM1) and therefore should work with the many DDR2 SODIMM sockets available on the market. (Please note that the pinout of the Compute Module is not the same as a DDR2 SODIMM module; they are not electrically compatible.)

The SODIMM form factor was chosen as a way to provide the 200 pin connections using a standard, readily available and low cost connector compatible with low cost PCB manufacture.

The maximum component height on the underside of the Compute Module is 1.2mm.

The maximum component height on the top side of the Compute Module is 1.5mm.

The Compute Module PCB thickness is 1.0mm +/- 0.1mm.

Note that the location and arrangement of components on the Compute Module may change slightly over time due to revisions for cost and manufacturing considerations; however, maximum component heights and PCB thickness will be kept as specified.

Figure 3 gives the CM1 mechanical dimensions. Figure 4 gives the CM3 and CM3L mechanical dimensions.

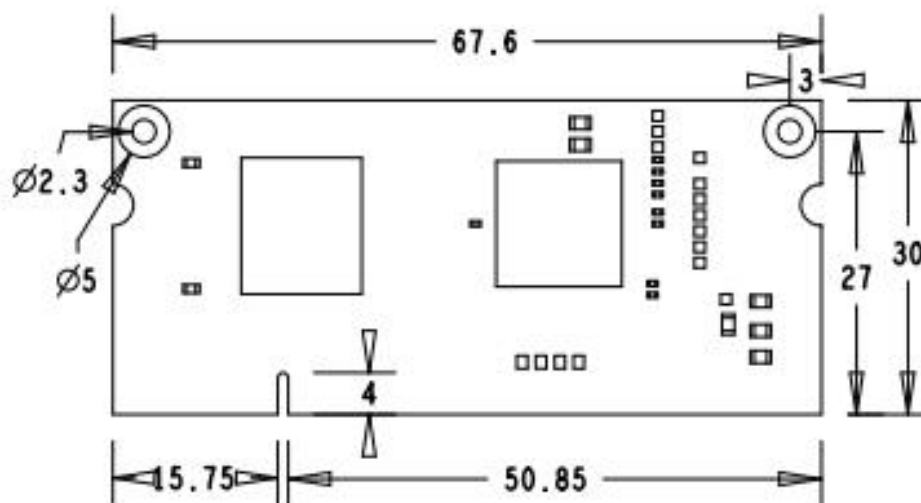


Figure 3: CM1 Mechanical Dimensions





Pin Name	DIR	Voltage Ref	PDN <sup>a</sup> State	If Unused	Description/Notes
<b><i>RUN and Boot Control (see text for usage guide)</i></b>					
RUN	1	3V3 <sup>b</sup>	Pull High	Leave open	Has internal 10k pull up
EMMC_DISABLE_N	1	3V3 <sup>b</sup>	Pull High	Leave open	Has internal 10k pull up
EMMC_EN_N_1V8	0	1V8	Pull High	Leave open	Has internal 2k2 pull up
<b><i>GPIO</i></b>					
GPIO[27:0]	I/O	GPIO0-27_VDD	Pull or Hi-Z <sup>c</sup>	Leave open	GPIO Bank 0
GPIO[45:28]	I/O	GPIO28-45_VDD	Pull or Hi-Z <sup>c</sup>	Leave open	GPIO Bank 1
<b><i>Primary SD Interface<sup>d,e</sup></i></b>					
SDX_CLK	0	SDX_VDD	Pull High	Leave open	Primary SD interface CLK
SDX_CMD	I/O	SDX_VDD	Pull High	Leave open	Primary SD interface CMD
SDX_Dx	I/O	SDX_VDD	Pull High	Leave open	Primary SD interface DATA
<b><i>USB Interface</i></b>					
USB_Dx	I/O	-	Z	Leave open	Serial interface
USB_OTGID	1	3V3		Tie to GND	OTG pin detect
<b><i>HDMI Interface</i></b>					
HDMI_SCL	I/O	3V3 <sup>b</sup>	Z <sup>f</sup>	Leave open	DDC Clock (5.5V tolerant)
HDMI_SDA	I/O	3V3 <sup>b</sup>	Z <sup>f</sup>	Leave open	DDC Data (5.5V tolerant)
HDMI_CEC	I/O	3V3	Z	Leave open	CEC (has internal 27k pull up)
HDMI_CLKx	0	-	Z	Leave open	HDMI serial clock
HDMI_Dx	0	-	Z	Leave open	HDMI serial data
HDMI_HPD_N_1V8	1	1V8	Pull High	Leave open	HDMI hotplug detect
<b><i>CAM0 (CSI0) 2-lane Interface</i></b>					
CAM0_Cx	1	-	Z	Leave open	Serial clock
CAM0_Dx	1	-	Z	Leave open	Serial data
<b><i>CAM1 (CSI1) 4-lane Interface</i></b>					
CAM1_Cx	1	-	Z	Leave open	Serial clock
CAM1_Dx	1	-	Z	Leave open	Serial data
<b><i>DSI0 (Display 0) 2-lane Interface</i></b>					
DSI0_Cx	0	-	Z	Leave open	Serial clock
DSI0_Dx	0	-	Z	Leave open	Serial data
<b><i>DSI1 (Display 1) 4-lane Interface</i></b>					
DSI1_Cx	0	-	Z	Leave open	Serial clock
DSI1_Dx	0	-	Z	Leave open	Serial data
<b><i>TV Out</i></b>					
TVDAC	0	-	Z	Leave open	Composite video DAC output
<b><i>JTAG Interface</i></b>					
TMS	1	3V3	Z	Leave open	Has internal 50k pull up
TRST_N	1	3V3	Z	Leave open	Has internal 50k pull up
TCK	1	3V3	Z	Leave open	Has internal 50k pull up
TDI	1	3V3	Z	Leave open	Has internal 50k pull up
TDO	0	3V3	0	Leave open	Has internal 50k pull up



## 6 Electrical Specification

**Caution!** Stresses above those listed in Table 4 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Symbol	Parameter	Minimum	Maximum	Unit
V <sub>BAT</sub>	Core SMPS Supply	-0.5	6.0	V
3V3	3V3 Supply Voltage	-0.5	4.10	V
1V8	1V8 Supply Voltage	-0.5	2.10	V
VDAC	TV DAC Supply	-0.5	4.10	V
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	-0.5	4.10	V
GPIO28-45_VDD	GPIO28-27 I/O Supply Voltage	-0.5	4.10	V
SDX_VDD	Primary SD/eMMC Supply Voltage	-0.5	4.10	V

Table 4: Absolute Maximum Ratings

DC Characteristics are defined in Table 5

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
$V_{IL}$	Input low voltage <sup>a</sup>	VDD <sub>IO</sub> = 1.8V	-	-	0.6	V
		VDD <sub>IO</sub> = 2.7V	-	-	0.8	V
$V_{IH}$	Input high voltage <sup>a</sup>	VDD <sub>IO</sub> = 1.8V	1.0	-	-	V
		VDD <sub>IO</sub> = 2.7V	1.3	-	-	V
$I_{IL}$	Input leakage current	TA = +85°C	-	-	5	μA
$C_{IN}$	Input capacitance	-	-	5	-	pF
$V_{OL}$	Output low voltage <sup>b</sup>	VDD <sub>IO</sub> = 1.8V, I <sub>OL</sub> = -2mA	-	-	0.2	V
		VDD <sub>IO</sub> = 2.7V, I <sub>OL</sub> = -2mA	-	-	0.15	V
$V_{OH}$	Output high voltage <sup>b</sup>	VDD <sub>IO</sub> = 1.8V, I <sub>OH</sub> = 2mA	1.6	-	-	V
		VDD <sub>IO</sub> = 2.7V, I <sub>OH</sub> = 2mA	2.5	-	-	V
$I_{OL}$	Output low current <sup>c</sup>	VDD <sub>IO</sub> = 1.8V, VO = 0.4V	12	-	-	mA
		VDD <sub>IO</sub> = 2.7V, VO = 0.4V	17	-	-	mA
$I_{OH}$	Output high current <sup>c</sup>	VDD <sub>IO</sub> = 1.8V, VO = 1.4V	10	-	-	mA
		VDD <sub>IO</sub> = 2.7V, VO = 2.3V	16	-	-	mA
$R_{PU}$	Pullup resistor	-	50	-	65	kΩ
$R_{PD}$	Pulldown resistor	-	50	-	65	kΩ



AC Characteristics are defined in Table 6 and Fig. 5.

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	$t_{rise}$	10-90% rise time <sup>a</sup>	-	1.6	-	ns
Digital outputs	$t_{fall}$	90-10% fall time <sup>a</sup>	-	1.7	-	ns
GPCLK	$t_{JOSC}$	Oscillator-derived GPCLK cycle-cycle jitter (RMS)	-	-	20	ps
GPCLK	$t_{JPLL}$	PLL-derived GPCLK cycle-cycle jitter (RMS)	-	-	48	ps

<sup>a</sup> Default drive strength, CL = 5pF, VDD\_IOx = 3.3V

Table 6: Digital I/O Pin AC Characteristics



Figure 5: Digital IO Characteristics

## 7 Power Supplies

The Compute Module has six separate supplies that must be present and powered at all times; you cannot leave any of them unpowered, even if a specific interface or GPIO bank is unused. The six supplies are as follows:

1. VBAT is used to power the BCM283x processor core. It feeds the SMPS that generates the chip core voltage.
2. 3V3 powers various BCM283x PHYs, IO and the eMMC Flash.
3. 1V8 powers various BCM283x PHYs, IO and SDRAM.
4. VDAC powers the composite (TV-out) DAC.
5. GPIO0-27\_VREF powers the GPIO 0-27 IO bank.
6. GPIO28-45\_VREF powers the GPIO 28-45 IO bank.



Supply	Description	Minimum	Typical	Maximum	Unit
VBAT	Core SMPS Supply	2.5	-	5.0 + 5%	V
3V3	3V3 Supply Voltage	3.3 - 5%	3.3	3.3 + 5%	V
1V8	1V8 Supply Voltage	1.8 - 5%	1.8	1.8 + 5%	V
VDAC	TV DAC Supply <sup>a</sup>	2.5 - 5%	2.8	3.3 + 5%	V
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	V
GPIO28-45_VDD	GPIO28-27 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	V
SDX_VDD	Primary SD/eMMC Supply Voltage	1.8 - 5%	-	3.3 + 5%	V

<sup>a</sup> Requires a clean 2.5-2.8V supply if TV DAC is used, else connect to 3V3

Table 7: Power Supply Operating Ranges

## 7.1 Supply Sequencing

Supplies should be staggered so that the highest voltage comes up first, then the remaining voltages in descending order. This is to avoid forward biasing internal (on-chip) diodes between supplies, and causing latch-up. Alternatively supplies can be synchronised to come up at exactly the same time as long as at no point a lower voltage supply rail voltage exceeds a higher voltage supply rail voltage.

## 7.2 Power Requirements

Exact power requirements will be heavily dependent upon the individual use case. If an on-chip subsystem is unused, it is usually in a low power state or completely turned off. For instance, if your application does not use 3D graphics then a large part of the core digital logic will never turn on and need power. This is also the case for camera and display interfaces, HDMI, USB interfaces, video encoders and decoders, and so on.

Powerchain design is critical for stable and reliable operation of the Compute Module. We strongly recommend that designers spend time measuring and verifying power requirements for their particular use case and application, as well as paying careful attention to power supply sequencing and maximum supply voltage tolerance.

Table 8 specifies the recommended minimum power supply outputs required to power the Compute Module.



Supply	Minimum Requirement	Unit
VBAT (CM1)	2000 <sup>a</sup>	mW
VBAT (CM3,3L)	3500 <sup>a</sup>	mW
3V3	250	mA
1V8	250	mA
VDAC	25	mA
GPIO0-27_VDD	50 <sup>b</sup>	mA
GPIO28-45_VDD	50 <sup>b</sup>	mA
SDX_VDD	50 <sup>b</sup>	mA

<sup>a</sup> Recommended minimum. Actual power drawn is very dependent on use-case

<sup>b</sup> Each GPIO can supply up to 16mA, aggregate current per bank must not exceed 50mA

Table 8: Minimum Power Supply Requirements

## 8 Booting

The 4GB eMMC Flash device on CM3 is directly connected to the primary BCM2837 SD/eMMC interface. These connections are not accessible on the module pins. On CM3L this SD interface is available on the SDX<sub>n</sub> pins.

When initially powered on, or after the RUN pin has been held low and then released, the BCM2837 will try to access the primary SD/eMMC interface. It will then look for a file called bootcode.bin on the primary partition (which must be FAT) to start booting the system. If it cannot access the SD/eMMC device or the boot code cannot be found, it will fall back to waiting for boot code to be written to it over USB; in other words, its USB port is in slave mode waiting to accept boot code from a suitable host.

A USB boot tool is available on Github which allows a host PC running Linux to write the BCM2837 boot code over USB to the module. That boot code then runs and provides access to the SD/eMMC as a USB mass storage device, which can then be read and written using the host PC. Note that a Raspberry Pi can be used as the host machine. For those using Windows a precompiled and packaged tool is available. For more information see [here](#).

The Compute Module has a pin called EMMC\_DISABLE\_N which when shorted to GND will disable the SD/eMMC interface (by physically disconnecting the SD\_CMD pin), forcing BCM2837 to boot from USB. Note that when the eMMC is disabled in this way, it takes a couple of seconds from powering up for the processor to stop attempting to talk to the SD/eMMC device and fall back to booting from USB.

Note that once booted over USB, BCM2837 needs to re-enable the SD/eMMC device (by releasing EMMC\_DISABLE\_N) to allow access to it as mass storage. It expects to be able to do this by driving the EMMC\_EN\_N\_1V8 pin LOW, which at boot is initially an input with a pull up to 1V8. If an end user wishes to add the ability to access the SD/eMMC over USB in their product, similar circuitry to that used on the Compute Module IO Board to enable/disable the USB boot and SD/eMMC must be used; that is, EMMC\_DISABLE\_N pulled low via MOSFET(s) and released again by MOSFET, with the gate controlled by EMMC\_EN\_N\_1V8. Ensure you use MOSFETs suitable for switching at 1.8V (i.e. use a



### 9.1.1 GPIO Alternate Functions

GPIO	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
	Pull						
0	High	SDA0	SA5	PCLK	•	•	•
1	High	SCL0	SA4	DE	•	•	•
2	High	SDA1	SA3	LCD_VSYNC	•	•	•
3	High	SCL1	SA2	LCD_HSYNC	•	•	•
4	High	GPCLK0	SA1	DPL_D0	•	•	ARM_TDI
5	High	GPCLK1	SA0	DPL_D1	•	•	ARM_TDO
6	High	GPCLK2	SOE_N	DPL_D2	•	•	ARM_RTCK
7	High	SPI0_CE1_N	SWE_N	DPL_D3	•	•	•
8	High	SPI0_CE0_N	SD0	DPL_D4	•	•	•
9	Low	SPI0_MISO	SD1	DPL_D5	•	•	•
10	Low	SPI0_MOSI	SD2	DPL_D6	•	•	•
11	Low	SPI0_SCLK	SD3	DPL_D7	•	•	•
12	Low	PWM0	SD4	DPL_D8	•	•	ARM_TMS
13	Low	PWM1	SD5	DPL_D9	•	•	ARM_TCK
14	Low	TXD0	SD6	DPL_D10	•	•	TXD1
15	Low	RXD0	SD7	DPL_D11	•	•	RXD1
16	Low	FL0	SD8	DPL_D12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPL_D13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPL_D14	•	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPL_D15	•	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPL_D16	•	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPL_D17	•	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPL_D18	SD1_CLK	ARM_TRST	•
23	Low	SD0_CMD	SD15	DPL_D19	SD1_CMD	ARM_RTCK	•
24	Low	SD0_DAT0	SD16	DPL_D20	SD1_DAT0	ARM_TDO	•
25	Low	SD0_DAT1	SD17	DPL_D21	SD1_DAT1	ARM_TCK	•
26	Low	SD0_DAT2	TE0	DPL_D22	SD1_DAT2	ARM_TDI	•
27	Low	SD0_DAT3	TE1	DPL_D23	SD1_DAT3	ARM_TMS	•

Table 9: GPIO Bank0 Alternate Functions





GPIO	Default						
	Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
28	None	SDA0	SA5	PCM_CLK	FL0	-	-
29	None	SCL0	SA4	PCM_FS	FL1	-	-
30	Low	TE0	SA3	PCMLDIN	CTS0	-	CTS1
31	Low	FL0	SA2	PCMLDOUT	RTS0	-	RTS1
32	Low	GPCLK0	SA1	RING_OCLK	TXD0	-	TXD1
33	Low	FL1	SA0	TE1	RXD0	-	RXD1
34	High	GPCLK0	SOE_N	TE2	SD1_CLK	-	-
35	High	SPI0_CE1_N	SWE_N	-	SD1_CMD	-	-
36	High	SPI0_CE0_N	SD0	TXD0	SD1_DAT0	-	-
37	Low	SPI0_MISO	SD1	RXD0	SD1_DAT1	-	-
38	Low	SPI0_MOSI	SD2	RTS0	SD1_DAT2	-	-
39	Low	SPI0_SCLK	SD3	CTS0	SD1_DAT3	-	-
40	Low	PWM0	SD4	-	SD1_DAT4	SPI2_MISO	TXD1
41	Low	PWM1	SD5	TE0	SD1_DAT5	SPI2_MOSI	RXD1
42	Low	GPCLK1	SD6	TE1	SD1_DAT6	SPI2_SCLK	RTS1
43	Low	GPCLK2	SD7	TE2	SD1_DAT7	SPI2_CE0_N	CTS1
44	None	GPCLK1	SDA0	SDA1	TE0	SPI2_CE1_N	-
45	None	PWM1	SCL0	SCL1	TE1	SPI2_CE2_N	-

Table 10: GPIO Bank1 Alternate Functions

Table 9 and Table 10 detail the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in the Broadcom Peripherals Specification document and have Linux drivers available.

### 9.1.2 Secondary Memory Interface (SMI)

The SMI peripheral is an asynchronous NAND type bus supporting Intel mode80 type transfers at 8 or 16 bit widths and available in the ALT1 positions on GPIO banks 0 and 1 (see Table 9 and Table 10). It is not publicly documented in the Broadcom Peripherals Specification but a Linux driver is available in the Raspberry Pi Github Linux repository (`bcm2835_smi.c` in `linux/drivers/misc`).



### 9.1.4 SD/SDIO Interface

The BCM283x supports two SD card interfaces, SD0 and SD1.

The first (SD0) is a proprietary Broadcom controller that does not support SDIO and is the primary interface used to boot and talk to the eMMC or SDX\_x signals.

The second interface (SD1) is standards compliant and can interface to SD, SDIO and eMMC devices; for example on a Raspberry Pi 3 it is used to talk to the on-board BCM43438 WiFi device in SDIO mode.

Both interfaces can support speeds up to 50MHz single ended (SD High Speed Mode).

## 9.2 CSI (MIPI Serial Camera)

Currently the CSI interface is not openly documented and only CSI camera sensors supported by the official Raspberry Pi firmware will work with this interface. Supported sensors are the OmniVision OV5647 and Sony IMX219.

It is recommended to attach other cameras via USB.

## 9.3 DSI (MIPI Serial Display)

Currently the DSI interface is not openly documented and only DSI displays supported by the official Raspberry Pi firmware will work with this interface.

Displays can also be added via the parallel DPI interface which is available as a GPIO alternate function - see Table 9 and Section 9.1.3

## 9.4 USB

The BCM283x USB port is On-The-Go (OTG) capable. If using either as a fixed slave or fixed master, please tie the USB\_OTGID pin to ground.

The USB port (Pins USB\_DP and USB\_DM) must be routed as 90 ohm differential PCB traces.

Note that the port is capable of being used as a true OTG port however there is no official documentation. Some users have had success making this work.

## 9.5 HDMI

BCM283x supports HDMI V1.3a.

It is recommended that users follow a similar arrangement to the Compute Module IO Board circuitry for HDMI output.

The HDMI CK\_P/N (clock) and D0-D2\_P/N (data) pins must each be routed as matched length 100 ohm differential PCB traces. It is also important to make sure that each differential pair is closely phase matched. Finally, keep HDMI traces well away from other noise sources and as short as possible.

Failure to observe these design rules is likely to result in EMC failure.



## 9.6 Composite (TV Out)

The TVDAC pin can be used to output composite video (PAL or NTSC). Please route this signal away from noise sources and use a 75 ohm PCB trace.

Note that the TV DAC is powered from the VDAC supply which must be a clean supply of 2.5-2.8V. It is recommended users generate this supply from 3V3 using a low noise LDO.

If the TVDAC output is not used VDAC can be connected to 3V3, but it must be powered even if the TV-out functionality is unused.

## 10 Thermals

The BCM283x SoC employs DVFS (Dynamic Voltage and Frequency Scaling) on the core voltage. When the processor is idle (low CPU utilisation), it will reduce the core frequency and voltage to reduce current draw and heat output. When the core utilisation exceeds a certain threshold the core voltage is increased and the core frequency is boosted to the maximum working frequency. The voltage and frequency are throttled back when the CPU load reduces back to an 'idle' level OR when the silicon temperature as measured by the on-chip temperature sensor exceeds 85C (thermal throttling).

A designer must pay careful attention to the thermal design of products using the CM3/CM3L so that performance is not artificially curtailed due to the processor thermal throttling, as the Quad ARM complex in the BCM2837 can generate significant heat output.

### 10.1 Temperature Range

The operating temperature range of the module is set by the lowest maximum and highest minimum of any of the components used.

The eMMC and LPDDR2 have the narrowest range, these are rated for -25 to +80 degrees Celsius. Therefore the nominal range for the CM3 and CM3L is -25C to +80C.

However, this range is the maximum for the silicon die; therefore, users would have to take into account the heat generated when in use and make sure this does not cause the temperature to exceed 80 degrees Celsius.

## 11 Availability

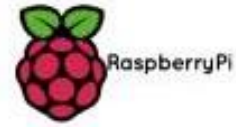
Raspberry Pi guarantee availability of CM1, CM3 and CM3 Lite until at least January 2023.

## 12 Support

For support please see the hardware documentation section of the Raspberry Pi website and post questions to the Raspberry Pi forum.

## Anexo 7

### Especificaciones Técnicas del Raspberry Pi3



#### Raspberry Pi 3 Model B



RASPERRYPI-MODB-1GB



RPI-MODB-16GB-NOOBS

#### Technical Specification:

- Broadcom BCM2837 64bit ARMv7 Quad Core Processor powered Single Board Computer running at 1.2GHz
- 1GB RAM
- BCM43143 WiFi on board
- Bluetooth Low Energy (BLE) on board
- 40pin extended GPIO
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source (now supports up to 2.4 Amps)
- Expected to have the same form factor has the Pi 2 Model B, however the LEDs will change position

## Anexo 8

### Especificaciones técnicas del encoder Oak Grigsby



## Specifications

### Electrical Specifications

Parameter	Minimum	Typical	Maximum	Units
Vcc Range	4.75		5.25	V
Supply Current @ 5V (Output Low) No Load		30		mA
VOH	see note 3			
VOL @ 16mA (Vcc = 4.75V)		290	700	mV
Pull-Up Resistor (Internal)	7.5	10.5	13.5	KOhm
Output Rise Time (CL-15pF)		500		nS
Output Fall Time (CL-15pF)		14		nS
Output: 2-bit gray code, Channel A leads Channel B by 90° with clockwise rotation				

### Environmental Specifications

Parameter	Minimum	Typical	Maximum	Units
Operating Temperature: Vcc = 5V	-40	25	85	°C
Storage Temperature Range: -55°C to +105°C, per MIL STD 202F Method 107 Test Condition A				
Humidity: 85% R.H. @ 40°C, 240 hours, per MIL STD 103B Test Condition A				

## Mechanical Specifications

Vibration: Harmonic motion with amplitude of 15g, varied from 10 to 2000 Hz for 12 hours, per MIL STD 204 Test Method B

Shock: 100g for 6 ms half sine wave with velocity change of 12.3 ft/s, per MIL STD 213 Test Condition C

Rotational Torque: Sleeve bearing: < 1 in. oz. max., Ball bearing: < .25 in. oz. max., higher for sealed unit.

Operating Speed: Sleeve bearing: 200 RPM max., Ball bearing: 3,000 RPM max.

Shaft Push In Force: 50 lbs. max.

Shaft Pull Out Force: 25 lbs. max.

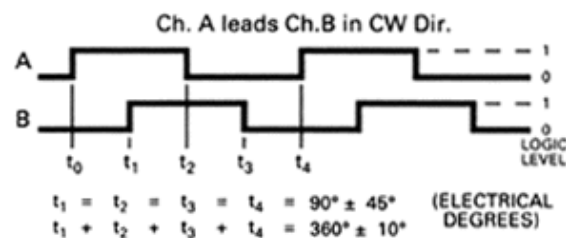
Bushing Mounting Torque: 10 in. lb. max

Terminal Strength: 3 lbs. max. applied perpendicular to the terminals

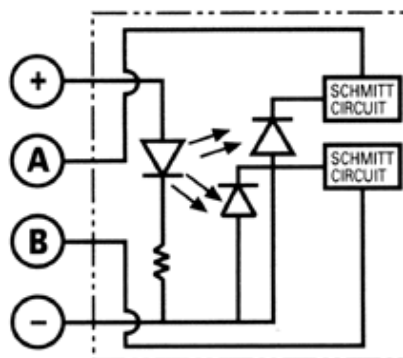
Rotational Speed, Rotational Life and Shaft Side Load Rating:  
(Load placed at end of standard shaft)

Series Number	Rotational Life	Rotational Speed Rating	Side Load Rating
90	5,000,000 +	200 rpm	0.25 lb.
92	100,000,000 +	3,000 rpm	0.25 lb.
94	100,000,000 +	3,000 rpm	5.0 lb.

### Two Channel Waveform



### Electrical Schematic



## Anexo 9

### Especificaciones Técnicas del RFID RC522

- o Nombre del módulo: MF522-ED
- o Corriente de trabajo: 3.3V 13-26mA / DC
- o Corriente de espera: 3,3 V 10-13mA / DC
- o actual para dormir: <80uA
- o pico de corriente: <30mA
- o Frecuencia de trabajo: 13,56 MHz
- o distancia de lectura de tarjeta: 0 ~ 60 mm (tarjeta Mifare1)
- o Protocolo: SPI
- o velocidad de comunicación de datos: Máximo de 10 Mbit / s
- o Tipos de tarjetas compatibles: Mifare1 S50, S70 Mifare1, Mifare Ultralight, Mifare Pro, Mifare DESFire
- o Dimensiones: 40 mm x 60 mm
- o Ambiente
- o Temperatura de trabajo: -20-80 grado
- o Temperatura de almacenamiento: -40-85 grado
- o Humedad: humedad relevante 5% -95%
- o Velocidad máxima SPI: 10 Mbit / s