



ESPE

**UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA**

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y

TELECOMUNICACIONES

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y

COMUNICACIONES DE DATOS

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO

DE INGENIERO ELECTRÓNICO EN REDES Y COMUNICACIÓN DE

DATOS

TEMA: “SISTEMA APLICADO AL CONTROL Y MONITOREO EN LAS

PYMES CON VOIP-GSM E IOT”

AUTOR: MORENO BONILLA, GREGORY ROBERTO

DIRECTOR: ING. ROMERO GALLARDO, CARLOS GABRIEL

SANGOLQUÍ

2019



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

CARRERA DE REDES Y COMUNICACIONES DE


DATOS

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**SISTEMA APLICADO AL CONTROL Y MONITOREO EN LAS PYMES CON VOIP-GSM E IOT**”, fue realizado por el señor **Gregory Roberto Moreno Bonilla** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente .

Sangolqui, 15 de enero del 2019

Firma:


.....
Ing. Carlos Gabriel Romero Gallardo

C.C.: 1712198066.....



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y

TELECOMUNICACIONES

CARRERA DE REDES Y COMUNICACIONES DE

DATOS

AUTORÍA DE RESPONSABILIDAD

Yo, **Gregory Roberto Moreno Bonilla**, declaro que el contenidos, ideas y criterios de trabajo de titulación: **Sistema aplicado al control y monitoreo en las Pymes con VOIP-GSM e IOT** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas-ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 15 de enero del 2019

Firma:

Gregory Roberto Moreno Bonilla

C.C.: 050200819-C.....



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
CARRERA DE REDES Y COMUNICACIONES DE
DATOS

AUTORIZACIÓN

Yo, **Gregory Roberto Moreno Bonilla** autorizo a la Universidad de las Fuerzas Armadas-ESPE, publicar el trabajo de titulación: **Sistema aplicado al control y monitoreo en las Pymes con VOIP-GSM e IOT** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 15 de enero del 2019

Firma:

Gregory Roberto Moreno Bonilla

C.C.: 0502008196.....

DEDICATORIA

Esta tesis está dedicada a mis padres Gregory y Elena quienes, con su apoyo incondicional, amor, paciencia y mucho esfuerzo han permitido que hoy cumpla una meta más, gracias por estar ahí en cada momento e inculcar en mí el ejemplo de perseverancia y lucha contra toda adversidad.

A mis hermanos Angie y Ronny por todo su cariño y siempre considerarme un ejemplo para sus vidas. Para mis dos sobrinos Fernandito y Ronnie que son el mejor regalo que la vida me ha podido dar y son el motor para siempre seguir adelante.

Finalmente quiero dedicar esta tesis a la memoria de mi Tío Oto, quien fue mi padrino y amigo desde la infancia y fue el pilar de nuestra familia inculcando los valores de amor y unidad.

AGRADECIMIENTO

Mi profunda gratitud a Dios que con su bendición me ha permitido salir victorioso de duras batallas sobre todo de salud.

Un agradecimiento a toda mi familia que siempre me apoya en cada decisión y camino que he decidido tomar en mi vida.

Un especial agradecimiento a Carito Franco que me ha acompañado incondicionalmente en todo este largo camino para cumplir la meta de graduarme.

De igual manera mis agradecimientos a la Universidad de Las Fuerzas Armadas ESPE y todos los profesores de los cuales me llevo la enseñanza y valiosos conocimientos que me han formado como un gran profesional y una persona que busca siempre el éxito y el desarrollo personal.

ÍNDICE DE CONTENIDOS

CERTIFICADO DEL DIRECTOR.....	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	viii
ÍNDICE DE FIGURAS	ix
SIGLAS Y ABREVIATURAS	xii
RESUMEN.....	xiv
ABSTRACT	xv
CAPÍTULO I.....	1
INTRODUCCIÓN	1
1.1. Antecedentes.....	1
1.2. Planteamiento del problema	11
1.3. Justificación	11
1.4. Objetivos.....	13
CAPÍTULO II	14
GENERALIDADES.....	14
2.1. Fundamentación teórica.....	14
2.2. Hipótesis	22
2.3. Variables de investigación.....	22

CAPÍTULO III	23
IMPLEMENTACIÓN DEL SISTEMA	23
3.1. Configuración de micro ordenador como servidor VoIP	23
3.2. Desarrollo de sistema de adquisición de señales analógicas	68
3.3. Desarrollo de algoritmos para la conversión datos a voz (DB-Rpi-RedCelular)	78
3.4. Desarrollo de algoritmos para la conversión datos a órdenes para microcontrol (RedCelular-DB-Rpi-Arduino).....	80
3.5. Desbloqueo de dispositivo de acceso a servicio VoIP a través de redes GSM	82
3.6. Implementación de algoritmos de conexión entre servidor VOIP y red GSM (GSM - BASE - VOZ - GSM).....	83
3.7. Diseño electrónico de sistema de adquisición.	85
3.8. Diagrama esquemático de topología implementada.....	86
3.9. Diseño de maqueta de experimentación	87
CAPÍTULO IV	90
PRUEBAS Y RESULTADOS DE LA INVESTIGACIÓN	90
4.1. Pruebas de funcionamiento del sistema.....	90
4.2. Análisis de resultados	100
CAPÍTULO V	101
CONCLUSIONES Y RECOMENDACIONES.....	101
REFERENCIAS BIBLIOGRAFICAS	104

ÍNDICE DE TABLAS

Tabla 1 <i>Librerías y lenguajes de AGI</i>	21
Tabla 2 <i>Características técnicas del módulo DTH22</i>	69

ÍNDICE DE FIGURAS

Figura 1 Arquitectura de Asterisk	17
Figura 2 Configuración del Sistema Operativo.....	24
Figura 3 Selección de módulos a instalar.....	26
Figura 4 Configuración del archivo “festival.scm”.....	33
Figura 5 Configuración del archivo “festival.conf”.....	34
Figura 6 Archivos disponibles en la carpeta voces	35
Figura 7 Conexión FTP mediante Filezilla	35
Figura 8 Modificación del archivo "siteinit.scm"	36
Figura 9 Modificación del archivo "phpagi.conf".....	36
Figura 10 Modificación del archivo "voices.scm"	37
Figura 11 Verificación de registro de los usuarios.....	41
Figura 12 Asignación de extensiones de los usuarios	45
Figura 13 SERVIDOR LAMP- Pagina por default de Apache2 Debian.....	49
Figura 14 SERVIDOR LAMP- Página de inicio de sesión de phpMyAdmin.....	50
Figura 15 Ingreso al base de datos de phpMyAdmin.....	52
Figura 16 Creación de la base de datos “variables”.....	53
Figura 17 Creación de tablas “datos”.....	53
Figura 18 Creación de variables en la tablas “datos”.....	54
Figura 19 Insertar valores en la tabla "datos" utilizando peticiones SQL.....	54
Figura 20 Valores de la tabla "datos".....	55
Figura 21 Esquema de conexión del Arduino	56
Figura 22 Programación del Arduino sin elementos físicos	56
Figura 23 Programación del Arduino con elementos físicos	57
Figura 24 Programación del archivo “Arduino_ReadData.py”	58
Figura 25 Ejecución del archivo "Arduino_ReadData.py".....	58
Figura 26 Conexión del Raspberry Pi con el adaptador RS-232	59
Figura 27 Puertos serial disponibles de la Raspberry Pi.....	59
Figura 28 Configuración avanzada de la Raspberry Pi.....	60
Figura 29 Deshabilitación del puerto serial.....	60

Figura 30 Mensaje de confirmación de deshabilitar el puerto	61
Figura 31 Mensaje de Aviso	61
Figura 32 Mensaje de Finalización del proceso	62
Figura 33 Estado de los puertos seriales de la Raspberry Pi.....	62
Figura 34 Conexión del puerto Serial de la Raspberry Pi.....	63
Figura 35 Estado de los puertos seriales de la Raspberry Pi.....	63
Figura 36 Ventanas de la comunicación serial.....	65
Figura 37 Módulo DTH22.....	68
Figura 38 Algoritmo de adquisición de humedad y temperatura.....	72
Figura 39 curva característica del sensor MQ6.....	73
Figura 40 Algoritmos de adquisición de GLP y CH4.....	74
Figura 41 Algoritmo de almacenamiento de datos analógicos	78
Figura 42 Algoritmo de conversión de texto a voz.....	80
Figura 43 Algoritmo de conversión datos a órdenes para microcontrol	82
Figura 44 Modem Huawei E160.....	83
Figura 45 Algoritmo de conexión VOIP y red GSM.....	84
Figura 46 Diagrama esquemático del prototipo de adquisición y control	85
Figura 47 Diagrama PCB de adaptación para la unidad de microcontrol.....	86
Figura 48 Diagrama de esquemático de la topología de control implementada	87
Figura 49 Diagrama de maqueta de pruebas y dimensionamiento a escala.....	88
Figura 50 Posición de elementos de control y adquisición.....	89
Figura 51 Prueba de conexión al servidor a través de terminal segura.....	90
Figura 52 Petición de almacenamiento de llave para conexión segura.....	91
Figura 53 Acceso a servidor implementado en la unidad de procesamiento utilizando usuario y contraseña.....	91
Figura 54 Petición de almacenamiento de llave para conexión segura.....	92
Figura 55 Comando de acceso a servicio de Voz IP en modo depuración	92
Figura 56 Petición de características pre-configuradas cargadas en el modem GSM	93
Figura 57 Respuesta del servidor de Voz IP sobre las características asignadas al modem GSM	93

Figura 58 Respuesta del servidor al recibir una llamada telefónica.....	94
Figura 59 Resultado del redireccionamiento de la llamada hacia el servicio de Voz IP	94
Figura 60 Respuesta del servidor generada por la herramienta de software PHP-AGI.....	95
Figura 61 Resultado de la respuesta de voz generada por el servidor y espera de señal de retorno emitida por el cliente.....	95
Figura 62 Adquisición de señal de control emitida por el cliente y actualización a través de la petición SQL generada por herramienta PHP-AGI.....	96
Figura 63 Prueba de acceso a gestor de base de datos	96
Figura 64 Actualización de estado de actuadores conectados a luminarias	97
Figura 65 Actualización de estado de actuadores conectados a luminarias y puertas	97
Figura 66 Actualización de estado de actuadores	98
Figura 67 Actualización de estado de actuadores conectados a luminarias.....	98
Figura 68 Actualización de estado de actuadores conectados a luminarias.....	99
Figura 69 Actualización en tabla de estados de actuadores con todas las señales de control activados	99

SIGLAS Y ABREVIATURAS

PyMe: Pequeña y mediana empresa.

GLP: *General Public License*

IoT: *Internet de las cosas*

VoIP: Voz sobre protocolo de internet

GSM: Global System for Mobile communications (Sistema Global para las comunicaciones Móviles),

SME: *Small and Medium Enterprise*

Wi-Fi: *Wireless Fidelity*

HTTP: *HyperText Transfer Protocol*

IP: *Internet protocol*

PBX: *Private Branch Exchange*

PSTN: *Public Service Telephony Network*

SIP: *Session Initiation Protocol*

AGI: *Asterisk Gateway Interface*

CSTR: *The Center for Speech Technology Research*

HTML: *HyperText Markup Language*

PHP: *HyperText Preprocessor*

LRC: *Least Cost Routing*

SSH: *Secure Shell*

TFTP: *Trivial File Transfer Protocol*

UDP: *User Datagram Protocol*

SQL: *Structured Query Language*

LAMP: *Linux Apache MySQL Perl, PHP, Python*

WWW: *World Wide Web*

LDR: *Light Dependent Resistor*

HR: *Humedad Relativa*

GND: *Ground*

VCC: *Voltaje en corriente continua*

RL: Load Resistor

RS: Sensor Resistor

PPR: Partes por millón

DB: Data Base

UART: Universal Asynchronous Receiver-Transmitter

IVR: Interactive Voice Response

USB: Universal Serial Bus

RESUMEN

El presente trabajo de investigación describe el diseño de un sistema electrónico de monitoreo y control de dispositivos cotidianos interconectados en red (Internet de las Cosas - IoT), destinado a pequeñas y medianas empresas (PyMES); vinculando un sistema de control basado en Voz sobre IP (VoIP) interconectado a las redes de telefonía móvil (GSM), de tal forma que permita el acceso hacia la información o el control de dispositivos cuando el usuario se encuentre dentro o fuera del perímetro de alcance de la intranet. La topología que describe el sistema se basa en un microordenador como nodo principal de la topología IoT; el nodo principal se encargará de la recepción de información de una tarjeta de adquisición de datos analógicos (Humedad, Temperatura y GLP) y procesarlos para su posterior consulta a través del servidor VoIP o a su vez, estará en la posibilidad de recibir señales de comando emitidas por un usuario de VoIP y convertirlas en señales de comando para controladores de dispositivos de uso regular (apertura de puertas, persianas o alumbrado). El control centralizado propuesto beneficiará al propietario de la PyME en la óptima gestión de la energía eléctrica invertida para la producción o el monitoreo constante de variables de las cuales depende el bienestar del producto final.

PALABRAS CLAVE:

- **SISTEMAS IOT**
- **VOZ SOBRE IP**
- **GSM**
- **RED DE TELEFONIA MOVIL**

ABSTRACT

This research work describes the design of an electronic monitoring and control system for daily devices interconnected in a network (Internet of Things - IoT), aimed at small and medium enterprises (SMEs); linking a control system based on Voice over IP (VoIP) interconnected to mobile telephony networks (GSM), in such a way as to allow access to information or control of devices when the user is inside or outside the perimeter of scope of the intranet. The topology that describes the system is based on a microcomputer as the main node of the IoT topology; the main node will be responsible for receiving information from an analog data acquisition card (humidity, temperature and LPG) and process them for later consultation through the VoIP server or, in turn, will be able to receive issued command signals by a VoIP user and convert them into command signals for controllers of regular use devices (opening of doors, blinds or lighting). The proposed centralized control will benefit the owner of the SME in the optimal management of the inverted electrical energy for the production or constant monitoring of variables on which the welfare of the final product depends.

KEYS WORDS:

- **IOT SYSTEMS**
- **VOICE OVER IP**
- **GSM**
- **MOBILE TELEPHONE NETWORK**

CAPÍTULO I

INTRODUCCIÓN

1.1. Antecedentes

De acuerdo a los avances de la tecnología y la gran demanda de los usuarios por la cantidad de información descargada de la nube ya sea datos, voz y video. Existen en la actualidad redes de comunicaciones que deben soportar estos requerimientos que andan en constante crecimiento y medios de transmisión que los soportan como las líneas de fibra óptica.

Las personas consideran que las conexiones a sistemas celulares, ISP, cámaras IP, Smart TV, sistemas domóticos e inmóticos para la conexión al internet para acceder a las configuraciones y funcionalidades que prestan estos dispositivos electrónicos para la comodidad del ser humano.

Con este enfoque nacen los sistemas IoT Internet de las cosas donde se gestiona o administra dispositivos de acceso a internet y controla las funciones para aplicaciones residenciales industriales entre otras desde un terminal remoto.

En si estos sistemas son subsistemas que permiten un acceso al internet para acceder a estos dispositivos dentro de una red de comunicaciones para configuración de los mismos en sistemas de seguridad, control de videovigilancia y la optimización de actividades diarias o cotidianas.

Estudio del Arte

En el año 2017 en Canonsburg, PA, USA, desarrollan el trabajo Energy Efficient IoT-Based Smart Home, por los autores Laila Salman, Safa Salman, Saeed Jahangirian, Mehdi Abraham, Fred German, Charlotte Blair, Peter Krenz. ANSYS Inc. En este proyecto se realizó con un enfoque hacia el control de temperatura en la cocina de un hogar, para poder demostrar la eficiencia de la energía que se podía obtener al controlar la calefacción y aire acondicionado de forma inteligente en este ambiente cuando el usuario se encuentre dentro o fuera del mismo, para lo cual se empleó un sensor de movimiento y una cámara perteneciente al sistema de seguridad de dicho hogar. El estudio se planteó en base a pruebas multifísicas realizadas en el ambiente, por los resultados arrojados de las pruebas realizadas se puede decir que se debe tener en cuenta este tipo de análisis para determinar el correcto enlace de los dispositivos dentro de un Smart Home, ya que de esto depende directamente el ahorro energético que se puede lograr.

La deficiencia principal de este estudio es el ambiente en el que se lo realizó, ya que al contar con una casa inteligente en su totalidad se pudo haber realizado el análisis multifísico en cada uno de los espacios en donde los usuarios presentan una mayor actividad para comparar los costes energéticos y la eficiencia del sistema. (Laila Salman, 2017)

En el 2018 en Ambato Ecuador se tiene un trabajo con el tema Approximation and Temperature Control System via an Actuator and a Cloud: An Application Based on the IoT for Smart Houses desarrollada por Pablo Palacios, Andrés Córdova

El trabajo se llevó a cabo tomando comparaciones de los valores de temperatura externa y posicionamiento y los valores umbrales de las mismas variables. Se empleó como equipos una Raspberry Pi 3, una máquina virtual alojada en la nube, y un celular con una aplicación GPS, elementos que fueron analizados para basar un sistema con inclusión de IoT para el control de electrodomésticos o dispositivos electrónicos a través de las variables anteriormente nombradas. Los resultados obtenidos arrojaron que la integración de los tres equipos seleccionados permite incluir las redes y tecnologías informáticas para mejorar la eficiencia de una casa inteligente y contribuyendo directamente a incrementar la calidad de vida del usuario. El sistema era diseñado para que solo la Raspberry Pi envíe la temperatura datos a la nube y lee una bandera de la nube para cambiar el actuador ENCENDIDO APAGADO. Además, el dispositivo móvil envía los datos de posición a la nube, liberando partes del sistema de procesamiento que podría hacerlo lento y menos eficiente. La capacidad del sistema realizado se puede incrementar aumentando el número de actuadores para el control de electrodomésticos y dispositivos electrónicos, adaptando sensores para la medición de distintas variables según se requiera. (Pablo Palacios, 2018)

En el mes de Agosto del año 2018, los estudiantes Pratibha D. y Mr. S. B, Patil del Departamento de Ingeniería en Electrónica y Comunicaciones de la ciudad de la India realizan un proyecto denominado: “Embedded based Industrial Parameters Monitoring and Controlling System using Internet of Things ” un sistema de control y monitoreo de parámetros industriales utilizando específicamente el internet de las cosas, con finalidad de obtener productos de alta calidad que se fabrican en dicha industria. Para el control y monitoreo del sistema utilizan comunicación GSM por su bajo costo y alta seguridad sin la necesidad de una gran infraestructura de hardware en las áreas de cobertura de GSM; los datos emitidos de los

sensores tanto de temperatura como de humo son recopilados y subidos a páginas web para tener el acceso y decisiones de estos desde cualquier parte del mundo, una vez recopilados los datos envían por medio de mensajes y correos electrónicos a la persona que se encuentra a cargo del departamento en caso de existir alguna emergencia. El proyecto ha sido de gran utilidad para el monitoreo continuo de aplicaciones industriales, así como también para el control de los sensores a larga distancia entre la industria y el usuario, y por medio de esta comunicación GSM se obtiene una conectividad flexible. Para obtener un mejor rendimiento en la productividad de la industria se puede adicionar más funciones y así poder llegar a tener un sistema inteligente de monitoreo y control total. (Pratibha D. Lagad, 2018)

Se publica en el 2017 en la India un artículo denominado Voice controlled home automation system using Natural Language Processing (NLP) and Internet of Things (IoT); en el Departamento de Ingeniería y Gestión de Tecnología Científica realizada por Paul Jasmin Rani, Jason Bakthakumar, B. Praveen Kumaar, U. Praveen Kumaar, Santhosh Kumar, donde hacen un sistema de automatización del hogar controlado por voz utilizando procesamiento de lenguaje natural e Internet de las Cosas (IoT), en donde controlan ventiladores, alarmas de puertas, luces y máquinas de café por medio del control de la voz del usuario, este comando de voz es interpretado por el móvil el cual usa el procesamiento de lenguaje natural, este dispositivo actúa como como consola central para cumplir la operación que es dictada por el usuario. El teléfono inteligente lo utilizan para conectar a este los dispositivos que se encuentran a su vez conectados a una placa Arduino a través del internet de las cosas, con ello proporcionan una solución automatizada en la que, los usuarios tengan más confianza para el control de varios electrodomésticos. Al utilizar comunicación inalámbrica wifi y teléfonos

inteligentes este proyecto tiene gran ventaja, ya que en la actualidad estos dispositivos abarcan casi el 95% de toda la población a nivel mundial. Para que este sistema sea más completo se podría enviar mensajes de texto al usuario en caso de existir emergencias, como incendios producidos por fuga de gas, robos, entre otros parámetros que puedan emitir los sensores previamente instalados. (P. J. Rani, 2017)

En el mismo país en el año 2016 se desarrolla un proyecto con el tema:” Design and development of an IOT based wearable device for the safety and security of women and girl children” , por Anand Jatti, Madhvi Kannan, R M Alisha; en el presente paper se trata sobre el desarrollo de un dispositivo portátil para la seguridad y protección de mujeres y niñas, esto se realiza a partir de señales fisiológicas obtenidas mediante una resistencia galvánica de la piel y la temperatura corporal, los datos de la posición del cuerpo se adquieren mediante un acelerómetro de triple eje, esto continua con la etapa de reconocimiento de actividad, en el cual se emplea un algoritmo de aprendizaje automático.

En cuanto a la monitorización de datos se lo realiza a través del envío inalámbrico de los datos obtenidos por el sensor a una nube de código abierto, el análisis de estos es analizados por el software MATLAB, si se presenta alguna anomalía en estas señales monitoreadas se envía una notificación de alerta a las personas designadas. Para el desarrollo de este proyecto se usó diferentes tipos de sensores como un sensor de temperatura, un acelerómetro (ADXL335E), sensor de resistencia de piel, dichos datos son procesados por un Arduino ATmega 328P, los cuales son enviados a ThingSpeak a través de un módulo Wi-Fi, el algoritmo empleado el Weka, este dispositivo tiene un valor aproximado de 855 dólares. (Anand Jatti, 2016)

En el 2013 en Corea del Sur de tiene un trabajo titulado:” Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone ” , por Rajeev Piyare , en este trabajo se presenta un sistema de monitoreo y control flexible y de bajo costo, el mismo que consta de un servidor micro web, con conectividad IP el mismo que es usado para acceder y controlar remotamente a través de un teléfono con sistema operativo Android, el mismo que no requiere de una PC dedicada como servidor puesto que usa un nuevo protocolo de comunicación.

Básicamente el sistema utiliza 3 partes, las cuales son el servidor web-micro, módulo de interfaz de Hardware y el paquete de software

En el primer punto la arquitectura utiliza servicio web basados en RESTful para la comunicación entre el usuario y el dispositivo doméstico, ya que el mismo usa XML y envía una solicitud HTTP POST, para la interfaz de la aplicación se utilizó lenguaje de programación JAVA mediante el Kit de desarrollo de software para Android (SDK), el mismo que consta de una interfaz intuitiva para el usuario, así como las indicaciones de como conectarse a la red. Para la implementación se hace uso de un Arduino Uno y una Shell Ethernet que permiten implementar el micro servidor para la puerta de enlace doméstica, el módulo ethernet ayuda a crear un puente para la conexión entre el proxy local y el home Gateway, para el control del interruptor de la luz, control de temperatura, así como electrodomésticos. (Piyare, 2013)

En 2018 en Gales, Reino Unido por James Hargreaves realiza un trabajo denominado: Security Awareness and Risks in IoT Devices: Gathering Information and Perspectives from University Students in South Wales, Cardiff.

La tecnología está integrada completamente en nuestro diario vivir y ha seguido avanzando

para acoplarse a las necesidades de la sociedad. En la actualidad, el crecimiento y la prevalencia de los dispositivos IoT en el hogar son particularmente sobresalientes mediante el uso de computadoras, teléfonos inteligentes y tabletas, por nombrar algunos. El punto en común que comparten todos estos dispositivos es la conectividad con una red para pasar y compartir información. Esto se puede hacer sin prisa o como una herramienta para fines comerciales y de estudio. Es importante reconocer que, como resultado del crecimiento de estas tecnologías, las amenazas son más inminentes que nunca y continúan desarrollándose a medida que el acceso a la información personal a través de la tecnología se generaliza. El propósito de esta disertación es determinar las especificidades del uso de los dispositivos IoT en los hogares para estudiantes en Cardiff, Gales del Sur. Se evaluarán las implicaciones de seguridad y la comprensión de la terminología clave de TI con el fin de evaluar las amenazas y la conciencia que tienen los estudiantes con respecto a las tecnologías destacadas. Identificar en qué posición se están metiendo los estudiantes cuando se trata de los riesgos de ser hackeado. (Hargreaves, 2018)

Además, en Canadá en el 2018 por Tam Thanh Doan desarrolla un Smart Home with Resilience Against Cloud Disconnection; es una casa inteligente está equipada con cientos de sensores que realizan mediciones y, en combinación con otras fuentes de datos (por ejemplo, datos de terceros), proporciona inteligencia que se utiliza para personalización y servicios automatizados, además de mejorar la eficiencia y comodidad para sus residentes. Las plataformas de hogares inteligentes de hoy son principalmente dispositivos basados en la nube, para que hogar detecten el entorno y envían los datos recopilados, directamente o a través de un hub, a la nube. Las nubes ejecutan diversas aplicaciones y análisis en los datos recopilados y generan comandos de acuerdo con las especificaciones de los usuarios, que se envían a los

actuadores para controlar el entorno. Centrándose en la dependencia de la nube de los hogares inteligentes, esta tesis hace dos contribuciones. Diseño de un sistema seguro de registro de eventos para una casa inteligente basada en host, SHEL. Un sistema de registro de eventos seguro y confiable es un componente esencial de los hogares inteligentes con una amplia gama de aplicaciones tales como detección de fallas, análisis forense y contabilidad. Proponemos un marco conceptual basado en el host para almacenar y procesar datos en hogares inteligentes, analizar los requisitos de seguridad de dichos entornos. Proporcionamos una descripción general de nuestra implementación de un sistema de registro de mensajes (eventos) para un hogar típico y presentamos la evaluación de eficiencia de nuestro diseño criptográfico. Proponer una casa inteligente resistente. Hacemos la siguiente pregunta: ¿qué pasa si la nube no está disponible? Esto puede suceder no solo por accidente o por causas naturales sino también debido a ataques dirigidos. La función del hub en esta configuración es el enlace de mensajes entre los dispositivos y la nube, mientras que la nube realiza los análisis, el cálculo y el control requeridos. Discutimos los posibles efectos de dicha falta de disponibilidad en las funcionalidades comúnmente disponibles en hogares inteligentes, incluidos los servicios relacionados con la seguridad y la protección, así como el apoyo a la salud y el bienestar de los usuarios domésticos. Proponemos la arquitectura de software de RES-Hub, un centro que puede proporcionar las funcionalidades requeridas cuando la nube no está disponible. Durante el funcionamiento normal del sistema, RES-Hub recibirá actualizaciones regulares de estado desde la nube y utilizará esta información para continuar proporcionando los servicios especificados por el usuario cuando detecta que la nube está inactiva. Describimos una arquitectura de software basada en IoTivity que se utiliza para implementar RES-Hub de una manera flexible y discutir nuestra implementación. (Doan, 2018)

En Argentina en el 2018 se realiza un trabajo Sistema de control aplicado a comunas rurales aisladas y su vinculación con Internet de las cosas (IoT) por Walter Daniel Etchart - Andres Ezequiel Martinovic; El mundo de las Comunidades Rurales Aisladas es muy amplio y diverso. Para los objetivos de esta tesina no resulta necesario entrar en el análisis de su diversidad, sino más bien en el de los elementos que las unen, fundamentalmente, la ausencia de infraestructuras básicas (recursos energéticos básicos), vital para alcanzar unos niveles de bienestar y desarrollo razonables. Esta problemática, en principio, es afrontada por organismos gubernamentales mediante el uso de equipos que utilizan energías renovables o alternativas. Dada la poca frecuencia con que se realizan las salidas por mantenimiento, muchas veces suele ocurrir que las baterías de los equipos se desgastan más rápido de lo previsto o que los equipos sufren fallas inesperadas.

En la actualidad, no existen mecanismos implementados para llevar un control más preciso de estos equipos, por lo que los mantenimientos son estipulados cada cierto intervalo regular de tiempo, basados en la experiencia de los técnicos. Sin embargo, si estos equipos contarán con la capacidad de informar su estado (por ejemplo, si pudieran transmitir la corriente que están generando) se podría determinar si el equipo está funcionando adecuadamente y así tener un control más preciso del funcionamiento de los mismos. Esta problemática permite tener una base interesante y comprometida socialmente para ahondar en conceptos en pleno auge como es el caso de Internet de las Cosas.

A partir de la problemática planteada y del concepto de partida mencionado, se satisface con el objetivo principal de este trabajo, el cual es:

Estudiar e investigar el concepto de IoT y de diferentes protocolos de comunicación entre sistemas, como marco de trabajo orientado a la colaboración y compartimiento de información

entre diferentes aplicaciones. Estudiar e investigar los distintos tipos de sensores aplicables a este proyecto. Estudiar e implementar el patrón de diseño publish/subscribe como herramienta que sirve para mejorar la eficiencia en la comunicación e interoperabilidad entre aplicaciones. (Martinovic & Etchart, 2018)

En el mismo país realizan un trabajo Dispositivo de asistencia de personas mediante monitoreo IoT por Waldo Valiente, Esteban Carnuccio, Mariano Volker, Graciela De Luca, Gerardo García, Daniel Giulianelli, Raúl Villca, Marcos Vittorio

Los objetos y las personas, gracias a la tecnología actual, pueden ser comunicados y utilizados para mejorar las condiciones de vida.

Este trabajo intenta hacer llegar la tecnología para el cuidado de las personas, realizando un aporte social, ya que se trata de desarrollar un dispositivo simple, amigable y de bajo costo.

Será simple debido a que solo realizará funciones para el monitoreo y comunicación del estado de la persona. Amigable, ya que será fácil de usar para el monitoreo y contará con una aplicación Android en un dispositivo del cuidador. Los datos obtenidos durante el monitoreo serán almacenados en un servidor en la nube pudiendo ser procesados y utilizados posteriormente de ser necesario. Contará con el sistema de caídas desarrollado por este equipo de investigación, adaptándolo a un dispositivo pequeño y con baterías de larga duración. De bajo costo, debido a una producción en serie mediante circuitos y software a medida. (Waldo Valiente, y otros, 2018)

1.2. Planteamiento del problema

Los sistemas electrónicos actuales trabajan de forma independiente y no de forma integral para poder operar de forma dinámica y se puede tener problemas al momento de obtener información de sus actuadores.

Además, no se tiene sistemas para saber del correcto funcionamiento de los sistemas integrados de comunicaciones entre los sensores, el sistema de control y operación.

El elevado costo de integrar sistemas de control de datos y operación en una PYMES en una red de comunicaciones o tener acceso desde un lugar remoto es un limitante al momento de implementarle.

1.3. Justificación

Según (ekos, 2016) las PyMES representan un 97% del total de empresas establecidas en Ecuador, y únicamente un 3% de la población empresarial corresponde a una gran empresa; no obstante, estas últimas acaparan el 75% de los ingresos económicos a nivel nacional y aunque exista mayor cantidad de PyMES solo se recauda el 25% del ingreso económico. De la cifra anteriormente analizada, un 39% de las PyMES se dedica a actividades del comercio, seguida de un 14% que se dedica actividades de manufactura.

Para la mayoría de las PyMES el adaptarse a tecnologías IoT parte de una pregunta muy simple “Como afectara al negocio” (ITUSER, Tecnología para tu empresa, 2016) y la respuesta radica en el tipo de negocio, infraestructura y presupuesto, aunque este último no

representa una barrera puesto que no representa una inversión excesiva ni mucho menos requiere modificación extensa en la infraestructura (Enterprise, 2017).

En la época actual todo tipo de información se considera como relevante y a razón de la universalización de la tecnología casi todo dispositivo está en la capacidad de almacenarla. La respuesta de qué hacer con la información depende del tipo de negocio, servicio o producto ofrezca la empresa y cuáles son los datos relacionados que de ser mejor gestionados potenciaran las operaciones o el producto final, como puede ser el caso de las manufacturas, alimentación, sanidad e inclusive seguridad. (ITUSER, 2017)

Ya sea optimizando su funcionamiento interno con la interconexión que ofrece para mejorar departamentos de seguridad o marketing, entre muchos otros, o tomando la ola del desarrollo y ofreciendo servicios enfocados a esto, el IoT revoluciona la industria en general y es importante mantenerse al pendiente de su desarrollo y adopción (Negocio, 2017).

A beneficio de la universalización de la tecnología y el desarrollo de nuevos y mejorados dispositivos a precios que no superan los 100 USD, hoy en día se han desarrollado proyectos disponibles para todo el mundo en los que únicamente requieren el dispositivo y descargar la imagen adecuada del sistema operativo, cargarla y configurarla en una tarjeta SD (E. Rico, 2017)

Si una empresa adoptase la topología IoT, según estudios realizados en domótica e inmótica los consumos energéticos podrían ser reducidos hasta en un 50% y se incrementaría la productividad reduciendo el tiempo que un operario requiere para el monitoreo y control

manual de las variables de entorno, permitiendo reorientar el tiempo ahorrado en mejorar los procesos productivos de la PyME (Benavides, 2018).

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un sistema aplicado al control y monitoreo en las PyMes con VoIP-GSM e IoT.

1.4.2. Objetivos Específicos

- Configurar un micro ordenador como servidor VoIP para la gestión de información analógica convertida en datos audibles.
- Desarrollar un sistema de adquisición de señales de humedad, temperatura y GLP y control de dispositivos cotidianos con capacidad de enviar o recibir información desde el servidor VoIP.
- Configurar un dispositivo de acceso a redes GSM para el redireccionamiento de llamadas entrantes y salientes hacia servicios VoIP.
- Implementación de sistema de control y monitoreo en entorno simulado a través de maquetaría.
- Análisis de resultados obtenidos

CAPÍTULO II

GENERALIDADES

2.1. Fundamentación teórica

2.1.1. Internet de las cosas

Dentro de la perspectiva de lo que es las ciudades inteligentes esta la transformación en donde se utilizan diferentes algoritmos para lograr este proceso, esto hace relación directa a lo que es la IoT en donde se la define como internet de las cosas, donde se menciona que una ciudad es inteligente cuando las inversiones en capital humano y social, en la infraestructura y calidad de vida está a través de un gobierno participativo. (SANTANA, 2018)

2.1.1.1. Topología entre otras cosas de IOT

Para tener acceso a VoIP se debe tomar en cuenta que entre las terminales de los usuarios debe existir un software compatible en el cual se permita crear una sesión IP para continuar con la conexión se digitaliza la voz, se la comprime para que esta ocupe un menor ancho de banda y se suprime pausas o silencios de la conversación tratando de minimizar el ruido de fondo de la transmisión para luego poder enviarla a través de la red como si fuera un flujo de datos. (Ana Lucia Calderon, 2006)

Dentro de la perspectiva de lo que es las ciudades inteligentes esta la transformación en donde se utilizan diferentes algoritmos para lograr este proceso, esto hace relación directa a lo que es la Iot en donde se la define como internet de las cosas, donde se menciona que una ciudad es inteligente cuando las inversiones en capital humano y social, en la infraestructura y calidad de vida está a través de un gobierno participativo. (Santana, 2018)

2.1.2. Voz sobre IP

Se denomina VoIP a la forma de transportar tráfico de voz en tiempo real mediante el uso de internet empleado el protocolo IP (internet protocol) el cual permitirá a personas y compañías dependiendo el hardware y el software que ocupen realizar llamadas telefónicas mediante la red de datos. (Ana Lucia Calderon, 2006)

Al término VoIP también se lo puede considerar como la definición del conjunto de recursos que intervienen para la transición de la voz a través de internet mediante el uso del protocolo IP (Castillo, 2013)

2.1.2.1. VoIP y GSM

Muchos países antes del 2005 iniciaron la implementación de unificación de comunicaciones y comenzaron a transmitir no solo datos por este protocolo si no otro tipo de información como: videos, imágenes y entre ellos la voz, dando a conocer lo que hoy denominamos como tecnología VoIP y los servicios de Telefonía IP. (LENIN, 2017)

El término IP, hace referencia al protocolo de internet, el cual también es utilizado en las redes locales de datos, donde las PYMES están optando pasar voz sobre IP, aprovechando

su infraestructura de red, como alternativa a las tradicionales centrales telefónicas (ZAMBRANO, 2014)

2.1.3. Asterisk

Asterisk es un software PBX, desarrollado por la empresa Digium. Asterisk funciona bajo plataforma Linux y otras plataformas Unix con o sin hardware conectado a la red pública de telefonía, PSTN (Public Service Telephony Network).

Asterisk permite agregar a la red en telefonía funcionalidades como:

- Conectar empleados trabajando desde casa para un PBX de la oficina sobre conexiones de banda ancha.
- Conectar oficinas en varias provincias sobre IP. Esto puede ser hecho por Internet o por una red IP privada.
- Dar a los funcionarios, buzón de voz, integrándose con una “web” y sus e-mails.
- Construir aplicaciones de respuesta automática por voz, que puede conectarlo a un sistema de pedidos, por ejemplo, o a otras aplicaciones internas.
- Dar acceso al PBX de la compañía para usuarios que viajan, conectando sobre la VPN de un aeropuerto o un hotel. (Goncalves, 2017)

Arquitectura de Asterisk

Asterisk se compone de muchos módulos diferentes (Ver figura 1), siendo posible elegir qué módulos cargar y configuración. Cada módulo que carga proporciona diferentes capacidades al sistema. Por ejemplo, un módulo podría permitir que su sistema Asterisk se comunice con líneas telefónicas analógicas, mientras que otro podría agregar capacidades de informe de llamadas. (Newton, 2014)

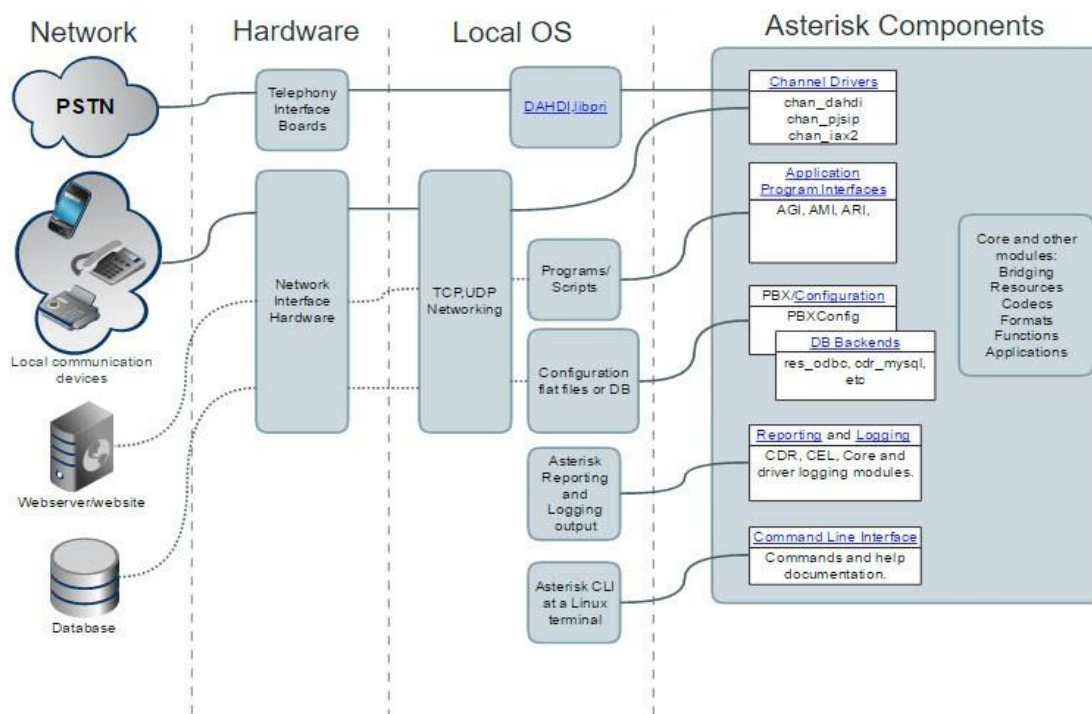


Figura 1 Arquitectura de Asterisk

El core de asterisk interactuar con muchos módulos llamados controladores de canal, proporcionando canales que siguen el plan de marcado de Asterisk para ejecutar el comportamiento programado y facilitar la comunicación entre dispositivos o programas fuera de Asterisk. Los canales a menudo usan la infraestructura de puente para interactuar con otros canales. (Newton, 2014)

El Core

El corazón de cualquier sistema Asterisk es el core, permitiendo leer los archivos de configuración, incluido el plan de marcado y carga todos los demás módulos, entre otras funciones. (Newton, 2014)

El core carga y construye el plan de marcado, que es la lógica de cualquier sistema Asterisk. El plan de marcado contiene una lista de instrucciones que Asterisk debe seguir para manejar llamadas entrantes y salientes en el sistema.

Módulos

Son archivos con una extensión de archivo .so, que se alojan en el directorio de módulos de Asterisk (que generalmente es /usr/lib/asterisk/modules). Cuando se inicia Asterisk, carga estos archivos y agrega su funcionalidad al sistema.

Los módulos de Asterisk que forman parte del core tienen un nombre de archivo que se parece a pbx_XXXXX.so. (Newton, 2014)

ejemplos de módulos de asterisk:

- chan_pjsip usa res_pjsip y muchos otros módulos res_pjsip para proporcionar una pila SIP para que los dispositivos SIP interactúen con Asterisk y entre ellos a través de Asterisk.
- app_voicemail proporciona funciones tradicionales de correo de voz de tipo PBX.
- app_confbridge proporciona puentes de conferencia con muchas características opcionales.
- res_agi proporciona Asterisk Gateway Interface, una API que permite el control de llamadas desde scripts y programas externos.

Llamadas

El objetivo principal de Asterisk es ser un motor para la construcción de sistemas de comunicación en tiempo real y aplicaciones.

Las llamadas en la terminología de telefonía generalmente se refieren a un teléfono que se comunica con (llama) a otro teléfono a través de un medio, como una línea PSTN. Sin embargo, en el caso de Asterisk, una llamada generalmente hace referencia a uno o más canales existentes en Asterisk.

Canales

Los canales son creados por Asterisk usando Channel Drivers, utilizando otros recursos en el sistema Asterisk para facilitar varios tipos de comunicación entre uno o más dispositivos. Por ejemplo, los canales SIP al pasar audio utilizarán el códec y los módulos de formato.

Dialplan

Es el principal método para dirigir el comportamiento de Asterisk, se encuentran como archivos de texto (por ejemplo, `extensions.conf`). Alternativamente, el dialplan podría leerse desde una base de datos, junto con otra configuración de módulo. Dialplan también puede llamar a través de otras interfaces como AGI para recibir instrucciones de control de llamadas desde scripts y programas externos.

2.1.4. Festival

Festival es un sistema general de síntesis de habla multilingüe desarrollado en CSTR. Ofrece un sistema de texto a voz completo con varias API, así como un entorno para el desarrollo y la investigación de técnicas de síntesis de voz.

El sistema está escrito en C++ y utiliza la Biblioteca de herramientas de voz de Edimburgo para la arquitectura de bajo nivel y tiene un intérprete de comandos basado en esquema (SIOD) para el control. La documentación se proporciona en el formato de información FSF que puede generar, un manual impreso, archivos de información y HTML.

El festival es software libre. Festival y las herramientas de voz se distribuyen bajo una licencia de tipo X11 que permite un uso comercial y no comercial sin restricciones. (Clark, 2004)

2.1.5. Asterisk Gateway Interface (AGI)

AGI es análogo a CGI en Apache. AGI proporciona una interfaz entre el plan de marcado de Asterisk y un programa externo que quiere manipular un canal en el plan de marcado. En general, la interfaz es sincrónica: acciones tomadas en un canal desde un bloque AGI y no regresan hasta que se completa la acción.

Existen tres tipos de AGIs:

- AGI Normal: llamado directamente en el servidor Asterisk.
- FastAGI: que llama al AGI en otro servidor a partir de una interfaz sockets.
- EAGI: da a la aplicación la posibilidad de acceder y controlar el canal de sonido más allá de la interacción con el plan de discado.
- DEADAGI: que da acceso a un mismo canal después de hangup().
- Bibliotecas y Frameworks AGI (Colp, 2018)

Tabla I*Librerías y lenguajes de AGI*

Nombre	Lenguaje	Protocolo
Adhearsion	Ruby	AMI/FastAGI
Asterisk-Java	Java	AMI/FastAGI
PAGI	PHP	AGI
PHPAGI	PHP	AGI
Panoramisk	Python+AsyncIO	AMI/FastAGI
Pyst2	Python	AMI/AGI
StarPy	Python+Twisted	AMI/FastAGI
Nanoagi	C++	AGI
AsterNET	.NET (C#/VB.net)	AMI/FastAGI
Ding-dong	node.js	AGI

2.1.6. GSM VoIP Gateway

Una puerta de enlace VoIP GSM permite el enrutamiento directo entre redes IP, digitales, analógicas y GSM. Con estos dispositivos (terminales celulares fijas) las empresas pueden reducir significativamente el dinero que gastan en telefonía, especialmente el dinero que gastan en llamadas de IP a GSM. La idea central detrás del ahorro de costos con VoIP GSM Gateways es Least Cost Routing (LCR).

Mediante el enrutamiento de menor costo, las puertas de enlace seleccionan la conexión telefónica más rentable. Controlan el número que se marca y la información de la tasa que se almacena en una tabla de enrutamiento interno. Debido a que varias tarjetas SIM y módulos GSM están integrados dentro de la pasarela GSM VOIP, es capaz de hacer llamadas GSM / GSM relativamente más baratas en lugar de costosas llamadas IP a GSM. (voip-info, s.f.)

2.2. Hipótesis

Sistema aplicado al control y monitoreo en las PyMes con VoIP-GSM e IoT.

2.3. Variables de investigación

Variable independiente

- Sistema de control y monitoreo en las PyMes

Variable dependiente

- VoIP-GSM e IoT

CAPÍTULO III

IMPLEMENTACIÓN DEL SISTEMA

3.1. Configuración de micro ordenador como servidor VoIP

3.1.1. Compilar versión estable de asterisk

INSTALACIÓN DE ASTERISK 13 EN RASPBERRY Pi 3

Como primer paso se debe instalar el sistema operativo de la Raspberry descargando la imagen de raspbian última versión en la página oficial, el sistema operativo a instalar se denomina Raspbian-Stretch-lite.

Se debe copiar el archivo en una tarjeta de memoria que posteriormente se insertará en la ranura de la raspberry. Para proceder con la instalación se debe conectar la raspberry a una pantalla, teclado y mouse. (admin, 2017)

Se debe cambiar la contraseña para personalizar la raspberry mediante el comando:

```
passwd
```

Luego hay que configurar una contraseña para el usuario root:

```
sudo passwd root
```

Se debe modificar la configuración de SSH

```
sudo nano /etc/ssh/sshd_config
```

Se debe modificar la siguiente línea:

```
PermitRootLogin without-password
```

Por el siguiente comando:

```
PermitRootLogin yes
```

Se debe guardar los cambios y activar y reiniciar el servicio:

```
systemctl enable ssh
```

```
systemctl start ssh
```

El siguiente paso es configurar una IP local estática que quede dentro del rango permitido:

```
nano /etc/dhcpd.conf
```

Al final del archivo se debe añadir la siguiente línea:

```
interface eth0
```

```
static ip_address=192.168.1.88/24
```

```
static routers=192.168.1.1
```

```
static domain_name_servers=192.168.1.1
```

La IP estática que se asigna es la 192.168.1.88 y el gateway es 192.168.1.1 Reiniciamos el servidor:

```
reboot
```

En este punto ya se puede acceder a la raspberry mediante comunicación ssh con la IP asignada y la contraseña personalizada.

La siguiente tarea es actualizar el sistema:

```
apt-get update
```

```
apt-get upgrade
```

Este comando permite la configuración más detallada del Sistema operativo:

```
raspi-config
```

```

1 Change User Password Change password for the current user
2 Hostname Set the visible name for this Pi on a network
3 Boot Options Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options Configure connections to peripherals
6 Overclock Configure overclocking for your Pi
7 Advanced Options Configure advanced settings
8 Update Update this tool to the latest version
9 About raspi-config Information about this configuration tool

<Select> <Finish>

```

Figura 2 Configuración del Sistema Operativo

Se debe configurar el nombre del dominio del servidor, la fecha y huso horario, por último, reiniciamos el sistema:

```
reboot
```

Se empieza con la instalación de Asterisk de la siguiente manera:

```
apt-get install libsqlite3-dev libncurses5-dev libxml2-dev libnewt-dev libssl-dev  
libiksemel-dev libgnutls28-dev libcurl4-openssl-dev libspandsp2 libspandsp-dev  
mysql-client libmariadb18 libmariadb-dev mariadb-server unixodbc-dev unixodbc  
libportaudio-dev libical-dev libneon27-dev libportaudio-dev libspeex-dev speex  
libvorbis-dev libsrtp0-dev iptables-persistent libsox2 libsox-dev  
apt-get install libuuid1 uuid uuid-dev libjansson4 libjansson-dev subversion sendmail  
sox ngrep
```

Se debe descargar la última versión de Asterisk 13

```
cd/usr/src/  
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-13.17.8.tar.gz
```

Se debe descomprimir y entrar a la siguiente carpeta:

```
tar -xf asterisk-13.17.2.tar.gz  
cd asterisk-13.17.2
```

Para instalar el Nuevo canal chan_pjsip:

```
./configure --libdir=/usr/lib --with-pjproject-bundled
```

Sin el siguiente canal chan_pjsip:

```
./configure --libdir=/usr/lib  
make menuconfig
```

Se debe seleccionar los módulos: Add-ons

```

--- Extended ---
XXX chan_mobile
[ ] chan_oooh323
[*] format_mp3
[*] res_config_mysql
--- Deprecated ---
[*] app_mysql
[*] cdr_mysql

```

Figura 3 Selección de módulos a instalar

Se requiere del menú Core Sound Package y el menú Extras Soun Packages. Se guarda los cambios con Save & Exit y se inicia la compilación de las fuentes:

```

contrib/scripts/get_mp3_source.sh
make
make install
make samples
make config

```

Para que el audio sea en español:

```

mkdir /var/lib/asterisk/sounds/es
cd /var/lib/asterisk/sounds/es
wget https://www.asterisksounds.org/sites/asterisksounds.org/files/sounds/es-AR-1.11.3.zip
wget https://www.asterisksounds.org/sites/asterisksounds.org/files/sounds/es-
AR/download/asterisk-sounds-extra-es-AR-1.11.3.zip
unzip asterisk-sounds-core-es-AR-1.11.3.zip
unzip asterisk-sounds-extra-es-AR-1.11.3.zip

```

Iniciar Asterisk:

```
systemctl start asterisk
```

Entrar en la consola:

```
asterisk -rvvvvv
```

Se debe salir de la consola mediante:

CLI>quit

Se para el servidor de Asterisk mediante:

systemctl stop asterisk

Crear una base de datos y tabla para guardar los registros de llamadas:

systemctl enable mariadb

systemctl start mariadb

Se debe crear la contraseña del usuario root de Mariadb

mysql_secure_installation

Se debe configurar una contraseña. Y accedemos con la contraseña y usuarios configurados para el usuario root (los **** equivalen al usuario que deseen crear):

*mysql -u root -p*****

Se crea la base de datos de la siguiente manera:

MariaDB [(none)]> create database asteriskcdr;

Crear Usuario con accesos privilegiados:

MariaDB [(none)]> Grant All Privileges on asteriskcdr. to asterisk@'localhost' identified by '****';*

Seleccionar la base de datos y crear una tabla:

```
MariaDB[(none)]> use asteriskcdr
MariaDB[(none)]> CREATE TABLE cdr (
  id bigint(20) NOT NULL auto_increment,
  calldate datetime NO NULL default '0000-00-00 00:00:00',
  clid varchar(80) NOT NULL default'',
  src varchar(80) NOT NULL default'',
  dst varchar(80) NOT NULL default'',
  dcontext varchar(80) NOT NULL default'',
  channel varchar(80) NOT NULL default'',
  dstchannel varchar(80) NOT NULL default'',
  lastapp varchar(80) NOT NULL default'',
  lastdata varchar(80) NOT NULL default'',
  duration int(11) NOT NULL default '0',
  billsec int(11) NOT NULL default '0',
  disposition varchar(45) NOT NULL default'',
  amaflags int(11) NOT NULL default '0',
  accountcode varchar(20) NOT NULL default'',
  peeraccount varchar(20) NOT NULL default'',
  uniqueid varchar(32) NOT NULL default'',
  linkedid varchar (80) NOT NULL default'',
  userfield varchar(255) NOT NULL default'',
  PRIMARY KEY ('id'),
  KEY callerid (clid)
);
```

Salir de la base de datos:

```
MariaDB[asteriskcdr]>quit
```

Se configura el archivo de asterisk

```
mv /etc/asterisk/cdr_mysql.conf /etc/asterisk/cdr_mysql.conf.old
nano /etc/asterisk/cdr_mysql.conf
```

Se deben insertar las siguientes líneas:

```
[global]
hostname=127.0.0.1
dbname=asteriskcdr
table=cdr
password=****
user=asterisk
```

Se guarda los cambios posteriormente se inicia el servidor Asterisk:

```
systemctl start asterisk
```

Entrar en la consola:

```
asterisk -rvvvvv}
```

Se debe averiguar si la conexión está funcionando:

```
CLI> cdr mysql status
```

Instalar el servidor DNS

```
apt-get install bind9
```

Se debe activar para que se inicie el sistema:

```
systemctl enable bind9
```

Se modifica solamente el archivo:

```
nano /etc/bind/named.conf.options
```

Se cambia este bloque:

```
//forwarders{
//0.0.0.0;
//};
```

Por lo siguiente:

```
forwarders {  
208.67.222.222;  
208.67.220.220;  
};
```

Para resolver los nombres de dominio se utiliza las dos IP de OpenDNS.

```
dnssec-validation auto;
```

Tiene que quedar:

```
dnssec-validation no;
```

Guardar los cambios e iniciar el servicio:

```
systemctl start bind9
```

Revisar el archivo de Log para comprobar errores.

```
tail -100f /var/log/syslog
```

```
nano /etc/resolvconf.conf
```

Se modifica las siguientes líneas:

```
#name_servers=127.0.0.1
```

Para que quede:

```
name_servers=127.0.0.1
```

Guardar y reiniciar el sistema

```
reboot
```

Se accede nuevamente y se comprueba que el sistema resuelve los nombres de dominio localmente:

```
nano /etc/resolv.conf
```

Debe encontrarse:

```
# Generated by resolvconf
```

```
nameserver 127.0.0.1
```

Cerrar y abrir el siguiente:

```
nano /etc/dhcpd.conf
```

Modificar la última línea

```
static domain_name_servers=192.168.1.1
```

Para que quede:

```
static domain_name_servers=127.0.0.1
```

Guardar los cambios

A continuación, se debe instalar el servidor TFTP

```
apt-get install atftpd
```

Crear una carpeta con servicios privilegiados donde se guardarán la configuración de los teléfonos

```
mkdir /tftpboot
```

```
chmod -R 777 /tftpboot
```

```
chown -R nobody /tftpboot
```

Se modifica la configuración del servidor mediante:

```
nano /etc/default/atftpd
```

Modificar la línea:

```
USE_INETD=true
```

Por:

```
USE_INETD=false
```

Modificar esta línea:

```
OPTIONS="—tftpd-timeout 300 —retry-timeout 5 —mcast-port 1758 —mcast-addr  
239.239.239.0-255 —mcast-ttl —maxthread 100 —verbose=5 /srv/tftp"
```

Por:

```
OPTIONS="—tftpd-timeout 300 —retry-timeout 5 —mcast-port 1758 —mcast-addr  
239.239.239.0-255 —mcast-ttl —maxthread 100 —verbose=5 /tftpboot —  
logfile=/var/log/atftp.log"
```

Guardar los cambios

```
touch /var/log/atftpd.log  
chmod 644 /var/log/atftpd.log
```

Si el servicio inetd está corriendo:

```
ps aux | grep inetd
```

Se detiene y se deshabilita por defecto:

```
systemctl stop inetd  
systemctl disable inetd
```

Iniciar el servidor TFTP

```
/etc/init.d/atftpd restart
```

Para averiguar si está corriendo correctamente:

```
netstat -lnd | grep ":69"  
ps aux | grep atftpd
```

3.1.2. Importación de librerías texto a voz

Festival es un sistema que convierte el texto a voz, que tiene como objetivo establecer una comunicación entre el ordenador y la persona por medio de audio. Es por ello que la instalación de Festival es necesaria para realizar consultas al servidor de Asterisk. (Vivanco, 2017)

Para lo cual se realiza el siguiente procedimiento:

- Primero se debe instalar Festiva

```
sudo apt-get install festival
```

- Posteriormente se ingresa a la carpeta donde se instaló Festival y se edita el archivo "festival.sem"

```
cd /usr/share/festival/  
sudo nano festival.sem
```


- En el archivo “festival.scm” se va a realizar la configuración de la vox predefinida de Festival, esto se lo realiza añadiendo las líneas de código al final del archivo, como se muestra en la figura 4

```
;; Enable access to localhost (needed by debian users)
(set! server_access_list '("localhost\\.localdomain" "localhost"))

(define (tts_textasterisk string mode)
  "(tts_textasterisk STRING MODE)
Apply tts to STRING. This function is specifically designed for
use in server mode so a single function call may synthesize the string.
This function name may be added to the server safe functions."
  (let ((wholeutt (utt.synth (eval (list 'Utterance 'Text string))))
        (utt.wave.resample wholeutt 8000)
        (utt.wave.rescale wholeutt 5)
        (utt.send.wave.client wholeutt)))
    (provide 'festival)
```

Figura 4 Configuración del archivo “festival.scm”

- Ahora se debe integrar Festival con Asterisk, esto se lo realiza ingresando a la carpeta indicada y editando el archivo “festival.conf” de la siguiente manera

```
cd /etc/asterisk
```

```
sudo nano festival.conf
```

- Una vez ingresado al archivo “festival.conf” se debe corroborar que las siguientes instrucciones estén activadas, como se muestra en la figura 5

```
[general]
```

```
host=localhost
```

```
port=1314
```

```

GNU nano 2.2.6          Fichero: festival.conf
;
; Festival Configuration
;
[general]
;
; Host which runs the festival server (default : localhost);
;
host=localhost
;
; Port on host where the festival server runs (default : 1314)
;
port=1314
;
; Use cache (yes, no - defaults to no)
;
usecache=yes
;
; If usecache=yes, a directory to store waveform cache files.
; The cache is never cleared (yet), so you must take care of cleaning it
; yourself (just delete any or all files from the cache).
; THIS DIRECTORY *MUST* EXIST and must be writable from the asterisk process.
; Defaults to /tmp/
;
cachedir=/var/lib/asterisk/festivalcache/
;
; Festival command to send to the server.
; Defaults to: (tts_textasterisk "%s" 'file')(quit)\n
; %s is replaced by the desired text to say. The command MUST end with a
; (quit) directive, or the cache handling mechanism will hang. Do not
; forget the \n at the end.
;
festivalcommand=(tts_textasterisk "%s" 'file)(quit)\n

```

Figura 5 Configuración del archivo “festival.conf”

- Por último para guardar los cambios, se debe ejecutar el festival

`usr/bin/festival -server > /dev/null 2>&1&`

Instalación de voces en español

El asterisk tiene instaladas las voces por defecto en inglés, para lo cual se va a instalar las voces en español para un fácil entendimiento del usuario. (Vivanco, 2017)

- Vamos a crear la carpeta “voces” en el directorio “cd/usr/src”, en el cual va a estar instalado las voces en español

`sudo mkdir voces`

- Ingresar a la carpeta voces para instalar los paquetes que contienen las voces en español

`sudo wget https://www.sinologic.net/voces/voipnovatos-core-sound-es-ulaw-1.4.tar.gz`

`sudo wget https://www.sinologic.net/voces/voipnovatos-extra-sound-es-ulaw-1.4.tar.gz`

- Ahora se procede a descomprimir los archivos

```
sudo tar zxvf voipnovatos-core-sound-es-ulaw-1.4.tar.gz
```

```
sudo tar zxvf voipnovatos-extra-sound-es-ulaw-1.4.tar.gz
```

- Ahora se puede observar los paquetes que se encuentran en la carpeta mediante el comando ls, como se puede observar en la figura 6

```
pi@raspberrypi:/usr/src/voces $ ls
dictate  followme  silence
digits  letters   voipnovatos-core-sounds-es-gsm-1.4.tar.gz
es       phonetic  voipnovatos-extra-sounds-es-gsm-1.4.tar.gz
```

Figura 6 Archivos disponibles en la carpeta voces

Festival con la voz en español

Una vez instalado festival en asterisk se puede dar cuenta, que no viene incluido voces en español y por ese motivo se escucha diferente (español con voz inglesa), además la calidad del sonido es mala. Para lo cual se procede a instalar las voces en español del proyecto de la junta de AndaLucía, los mismos que contienen dos paquetes con voz en español tanto masculina como femenina.

Para este proyecto se utiliza la voz femenina, mediante Filezilla se va a transferir el archivo “festvox-sflpc16k_1.0-1_all.deb” al directorio “cd/usr/src/” como se observa en la figura 7

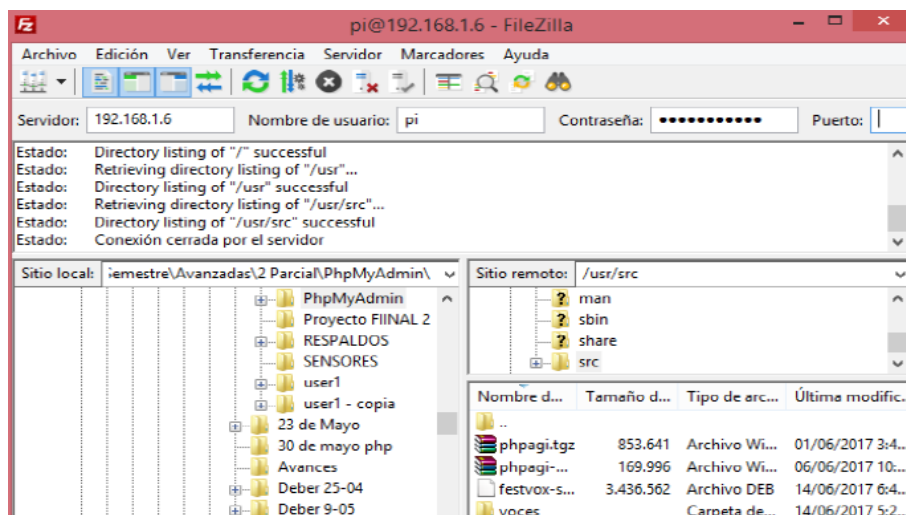


Figura 7 Conexión FTP mediante Filezilla

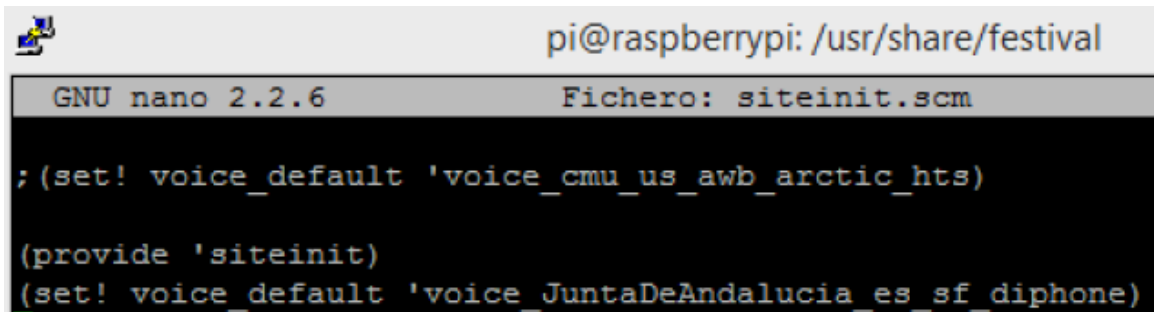
- Se ingresa al directorio “cd /usr/src/” para instalar la voz femenina mediante el siguiente comando

```
sudo dpkg -i festvox-sflpc16k_1.0-1_all.deb
```

- Posteriormente se configura la voz femenina como predeterminada, esto se logra editando el archivo “siteinit.scm” que se encuentra ubicado en “cd /usr/src/”

```
sudo nano siteinit.scm
```

- Para que festival utilice la voz femenina de forma predeterminada se agrega la siguiente línea al final:
(set! voice_default 'voice_JuntaDeAnadalucia_es_sf_diphone)’
como se puede observar en la figura 8

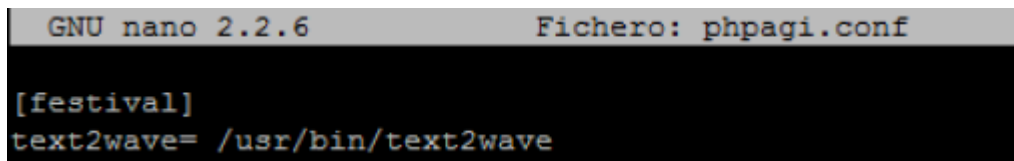


```
pi@raspberrypi: /usr/share/festival
GNU nano 2.2.6 Fichero: siteinit.scm
;(set! voice_default 'voice_cmu_us_awb_arctic_hts)
(provide 'siteinit)
(set! voice_default 'voice_JuntaDeAndalucia_es_sf_diphone)
```

Figura 8 Modificación del archivo "siteinit.scm"

- Así mismo se edita el archivo phpagi.conf que se encuentra en el directorio “cd /etc/asterisk” y se agrega la siguiente línea, como se observa en la figura 9

```
Text2wave= /usr/bin/text2wave
```



```
pi@raspberrypi: /etc/asterisk
GNU nano 2.2.6 Fichero: phpagi.conf
[festival]
text2wave= /usr/bin/text2wave
```

Figura 9 Modificación del archivo "phpagi.conf"

- También se modifica el archivo “voices.sm” que se encuentra en “cd /usr/share/festival/” para que festival pueda reconocer la voz femenina como la voz principal.

```
sudo nano voices.scm
```

- Aquí se cambia la línea de código “kal_diphone” por “JuntaDeAndalucia_es_sf_diphone” como se muestra en la figura 10

```

GNU nano 2.2.6                               Fichero: voces.scm
(list 'nitech_us_slt_arctic_hts
      'nitech_us_awb_arctic_hts
      'nitech_us_bdl_arctic_hts
      'nitech_us_clb_arctic_hts
      'nitech_us_jmk_arctic_hts
      'nitech_us_rms_arctic_hts
      'JuntaDeAndalucia_es_sf_diphone
      'ked_diphone

```

Figura 10 Modificación del archivo "voices.scm"

- Finalmente se inicia el servidor de Asterisk

```
sudo asterisk -rvvvvv
```

3.1.3. Configuración de suscriptores SIP

Creación De Cuentas SIP

Para lograr la comunicación entre teléfonos se debe crear y configurar cuentas SIP, para lo cual existen dos opciones para el controlador. La primera es sip.conf para chan_sip y la segunda es pjsip.conf para chan_pjsip/res_pjsip siendo res_pjsip el que proporciona la configuración. Cabe recalcar que chan_pjsip está disponible para asterisk 12 o versiones posteriores. Estos archivos suelen estar en la dirección /etc/asterisk/ (Newton, 2014)

- **Configuración de chan_sip (primera opción)**

Se abre el archivo sip.conf con el editor de texto de preferencia (nano, vi)

```
sudo nano sip.conf
```

Como se puede observar en la parte superior del archivo existen secciones como [general] y [authentication] los cuales controlan la funcionalidad del canal SIP.

Ahora se debe sacar una copia de respaldo, es decir, se copia el archivo sip.conf con el nombre sip.conf.sample y en el archivo sip.conf se precede a borrar todas las líneas. (Newton, 2014)

```
cp sip.conf sip.conf.sample
```

En los teléfonos agregamos los usuarios "Alice" y "Bob" para que podamos diferenciar los dispositivos.

Ahora se debe escribir el código de configuración en el archivo sip.conf.

```
sudo nano sip.conf
[general]
transport=udp
[friends_internal](!)
type=friend
host=dynamic
context=from-internal
disallow=all
allow=ulaw
[demo-alice](friends_internal)
secret= password1; poner una contraseña fuerte y única aquí en lugar de password1
[demo-bob](friends_internal)
secret=password2; poner una contraseña fuerte y única aquí en lugar de password2
```

A continuación, se va a describir cada uno de los elementos del archivo sip.conf

Transport=udp: establece que el transporte entre todas las cuentas chan_sip es mediante UDP

En la sección friends_internal

type=friend: se debe poner como tipo amigo para facilitar la comunicación entre los dispositivos.

host=dynamic: los dispositivos se deben registrar en asterisk para no establecer ip estática en el teléfono. (Newton, 2014)

Context=from-internal: cuando asterisk recibe una llamada, busca la extensión de la cual está recibiendo la llamada en el archivo extensions.conf

Disallow=all: no permite más códecs de los que están establecidos en “allow”

Allow=ulaw: solo permite el uso del códec ulaw

Secret: en esta sección se establece la contraseña para la conexión entre el dispositivo y asterisk.

Algo importante que hay que tomar en cuenta es que en la configuración de chan.sip se establece los usuarios con los que se deben autenticar los teléfonos, en este caso los usuarios serán “demo-alice” y el otro usuario “demo-bob”, cada uno con su respectiva contraseña (secret). (Newton, 2014)

➤ **Configuración de chan_pjsip (segunda opción)**

Al igual que el anterior método se realiza una copia del archivo pjsip.conf con el nombre pjsip.conf.sample y se edita el pjsip.conf y colocamos el siguiente código.

```

[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

; Plantillas para la configuración
[endpoint_internal](!)
type=endpoint
context=from-internal
disallow=all
allow=ulaw

[auth_userpass](!)
type=auth
auth_type=userpass

[aor_dynamic](!)
type=aor
max_contacts=1
; aquí se define los telefonos a conectar
[demo-alice](endpoint_internal)
auth=demo-alice
aors=demo-alice
[demo-alice](auth_userpass)
password=unsecurepassword ; contraseña de autenticación
username=demo-alice
[demo-alice](aor_dynamic)
[demo-bob](endpoint_internal)
auth=demo-bob
aors=demo-bob
[demo-bob](auth_userpass)
password=unsecurepassword ; contraseña de autenticación
username=demo-bob
[demo-bob](aor_dynamic)

```

La configuración de pjsip es más compleja que chan_sip debido a su arquitectura, ya que esta tiende a ser más desglosada.

En la sección **transport-udp** se especifica de la siguiente manera:

Type=transport: Esto define el tipo de transporte que utilizará las secciones

Protocol=udp: esto define que se va a usar el protocolo udp

`Bind=0.0.0.0`: se desea comunicarnos a través de las interfaces más apropiadas

En la sección **endpoint_internal** se especifica de la siguiente manera:

`Type=endpoint`: Esto establece un punto final para la configuración.

`Context=from-internal`: cuando asterisk recibe una llamada, busca la extensión de la cual está recibiendo la llamada en el archivo `extensions.conf`

`Disallow=all`: no permite más códecs de los que están establecidos en “allow”

`Allow=ulaw`: solo permite el uso del códec `ulaw`

En la sección **auth_userpass** se especifica de la siguiente manera:

`Type=auth`: aquí se describe las credenciales necesarias para la autenticación.

`Auth_type=userpass`: aquí se comunica a Asterisk que use las opciones de usuario y contraseña para la autenticación.

En la sección **aor_dynamic** se especifica de la siguiente manera:

`Type=aor`: aquí se describe la información de ubicación de registro para que asterisk sepa donde enviar la llamada

`Max_contacts=1` tiene un comportamiento similar al `host=dynamic` del `chan_sip`, es decir. establece el número máximo de registros aor, es decir una llamada por cada aor.

En la sección **demo-alice** y **demo-bob** se vincula a sus secciones de autor y autenticación, en la cual se debe establecer los nombres de usuario y contraseñas. (Newton, 2014)

Se procede a cargar la configuración

Aún falta varias configuraciones por realizar, sin embargo, algunas secciones de configuración no se pueden volver a cargar, es por ello que es más fácil reiniciar el servidor de asterisk de la siguiente manera

```
service asterisk restart
```

```
Asterisk -rx "core restart now"
```

Registro De Teléfonos Es Asterisk

Ahora se procede a configurar los teléfonos para que se comuniquen con Asterisk y para ello se deben registrar, dicho registro consiste en que el teléfono da su nombre de usuario y contraseña, espera alguna respuesta o llamada en una dirección ip.

A continuación, se va a listar algunas de las configuraciones necesarias para que el teléfono se comuniquen con Asterisk: (Newton, 2014)

Registrador/ Servidor de registro: el teléfono debe registrarse en esta ubicación del servidor.

Nombre del usuario SIP /Nombre de cuenta/Dirección de SIP: aquí se debe configurar el nombre del usuario, es decir en nuestro ejemplo “demo-alice” para un teléfono mientras que para el otro teléfono “demo-bob”

Autenticación del usuario SIP/Usuario Autenticado: Aquí será el mismo nombre del usuario anterior SIP.

Servidor proxy/Servidor proxy saliente: mediante este servidor, el teléfono se puede comunicar con llamadas externas

Para verificar que su teléfono se ha registrado correctamente se debe entrar al CLI mediante asterisk -rvvvv y presionar “sip show peers”, Si en la columna del host sale “sin especificar” el teléfono aún no se ha registrado, mientras que si sale una ip y en la columna de Dyn sale una “D” el teléfono está registrado correctamente, como se puede observar en la figura 11. El usuario “demo-alice” no está registrado correctamente mientras que el usuario “demo-bob” está correctamente registrado (Davenport, 2014)

```
server*CLI> sip show peers
Name/username      Host           Dyn NAT ACL Port      Status
demo-alice         (Unspecified) D      A  5060    Unmonitored
demo-bob           192.168.5.105 D      A  5060    Unmonitored
2 sip peers [Monitored: 0 online, 0 offline Unmonitored: 2 online, 0 offline]
```

Figura 11 Verificación de registro de los usuarios

3.1.4. Configuración de suscriptor GSM usando CHAN_DONGLE

INSTALACION DEL CONTROLADOR DE ASTERISK 11 CHANG DONGLE

1. **Primero se debe instalar los paquetes suficientes y necesarios** (MY TECHNOLOGY | REPUBLIC, 2015)

- libbluetooth-dev libncurses5 libncurses5-dev
- uuid-dev
- libxml2-dev
- git-core libtools
- automake make
- usb-modeswitch
- vim vim-common
- ntp
- asterisk-dev

- 2. Se debe dirigir al directorio preferido y se descarga el controlador que se va a utilizar y luego se extrae del archivo zip descargado con las siguientes líneas de comando:**

```
cd /home/user/download
wget http://www.mytechrepublic.com/wp-content/uploads/2016/03/asterisk-
chan-dongle-asterisk11.zip
unzip asterisk-chan-dongle-asterisk11.zip
cd asterisk-chan-dongle-asterisk11
```

- 3. Se crea el aclocal.m4 el cual es de generación automática.**

```
aclocal
autoconf
automake -a
```

- 4. Después de haber instalado los paquetes necesarios y realizado los pasos anteriores, se debe compilar el paquete usando ./configure como se muestra a continuación:**

```
./configure
make
make install
```

- 5. Posteriormente se copia el archivo dongle.conf que esta creado en la carpeta como se indica a continuación:**

```
cp etc/dongle.conf /etc/asterisk
chmod -R 0777 /etc/asterisk
```

- 6. Después se carga el módulo dongle.conf en asterisk:**

```
asterisk -rvvv
asterisk*CLI> module load chan_dongle.so
```

- 7. Se modifica el archivo dongle.conf y se cambia la configuración de su audio y sus datos de acuerdo con su USB como se indica a continuación:**

```
[dongle0]
exten=0912345678          %Se coloca la extensión del teléfono móvil
audio=/dev/ttyUSB1       %Puerto tty para conexión de audio
data=/dev/ttyUSB2        %Puerto tty para comandos AT, ningún valor por
                          defecto, o se puede usar imei y/o imsi que deben tener exactamente 15 dígitos.

imei=353558042801234
imsi=515050921461234
```

8. Se modifica el archivo /etc/udev/rules.d/92-dongle.rules y se coloca lo siguiente:

```
vim /etc/udev/rules.d/92-dongle.rules
KERNEL == "ttyUSB [012345]", OWNER = "root", GROUP = "root", MODE = "660"
```

9. Se edita el archivo extensions.conf que se encuentra en el directorio “etc/asterisk/extensions.conf” y se agrega un código para llamar al módulo chan dongle:

```
[general]
static=yes
writeprotect=no
clearglobalvars=no
[globals]          %Aquí van las variables globales

[incoming]          %Aviso de llamada entrante
exten => 09123456789,1,Log(NOTICE, Incoming call from
${CALLERID(all)})
exten => 09123456789,n,Dial(SIP/4001)
exten => 09123456789,n,Hangup()

[outgoing]          %Aviso de llamada saliente
exten => _X.,1,Log(NOTICE, Dialing out from ${CALLERID(all)} to
${EXTEN} through Dongle0 Provider)
exten => _X.,n,Dial(dongle/dongle0/${EXTEN},60)
exten => _X.,n,Playtones(congestion)
exten => _X.,n,Hangup()
exten => 4001,1,Dial(SIP/4001)
exten => 4001,n,Hangup()
exten => 4002,1,Dial(SIP/4002)
exten => 4002,n,Hangup()
```

10. Se edita el archivo sip.conf y se añade el número al cual se va a utilizar para nuestro hardphone o softphone:

```
[general]
disallow=all
allow=ulaw
allow=alaw
allow=gsm
qualify=yes
canreinvite=no
deny=0.0.0.0/0.0.0.0          %Sirve para denegar todo el tráfico y solo permitir la subred
valida de su VoIP
permit=192.31.255.0/255.255.255.0 %Sirve para asegurar que sea de su red y no de otra
```

```
[4001]
type=friend
callerid="ext4001"
context=outgoing           %Contexto de extensions.conf
host=dynamic
secret=secretpassword
[4002]
type=friend
callerid="ext4002"
context=outgoing           %Contexto de extensions.conf
host=dynamic
secret=secretpassword
```

11. Por último, se vuelve a cargar la configuración del plan de marcación de asterisk para que se guarden los cambios que se realizaron durante la configuración:

```
asterisk -rvvv
asterisk * CLI> dialplan reload
asterisk * CLI> sip reload
```

12. Ya se puede realizar una llamada de prueba al teléfono móvil configurado para comprobar el correcto funcionamiento de asterisk. (MY TECHNOLOGY | REPUBLIC, 2015)

3.1.5. Configuración de Dialplan

La extensión en asterisk representa una serie de acciones en el plan de marcado, como por ejemplo extensiones para el menú de asistencia automática (operadora)

Se debe abrir el archivo `extensions.conf` y se puede observar que se encuentra las secciones `[general]` y `[globals]` que son de propósito general. Las demás secciones son llamadas secciones de contexto, estas secciones le dicen a Asterisk que dirija la llamada al punto de destino, para ello agregamos el contexto `"[from-internal]"` al final del archivo. (Newton, Asterisk, 2014)

Nombrar los contextos en Dialplan

No necesariamente debe tener un nombre específico el contexto, es decir en vez de `internal` podría haberse llamado `contexto1`, al fin al cabo el comportamiento va a ser el mismo, para este ejemplo se lo llamará `"internal"` dado que las extensiones se marcarán dentro de la red.

Ahora se agrega el código al final de `extensions.conf`.

```
[from-internal]
extend=>6001,1,Dial,(SIP/demo-alice,20)
extend=>6002,1,Dial,(SIP/demo-bob,20)
```

Aquí se establece que el usuario `demo-alice` va a tener la extensión `6001`, la misma que tendría un tiempo de espera de `20` segundos, de igual manera al usuario `demo-bob` va a tener la extensión `6002` y también va a tener una espera de `20` segundos. (Newton, Asterisk, 2014)

Una vez guardados los cambios se debe verificar que Asterisk esté leyendo correctamente la configuración, esto se lo realiza en el CLI mediante el comando `dialplan show from-internal`, si la configuración está correctamente, debe aparecer en el CLI algo como muestra la figura 12

```
server*CLI> dialplan show from-internal
[ Context 'from-internal' created by 'pbx_config' ]
'6001' =>          1. Dial(SIP/demo-alice,20)           [pbx_config]
'6002' =>          1. Dial(SIP/demo-bob,20)           [pbx_config]

-- 2 extensions (2 priorities) in 1 context. --
```

Figura 12 Asignación de extensiones de los usuarios

3.1.6. Configuración de Asterisk Gateway Interface (AGI) para comunicación a base usando PHP

TRABAJAR CON ASTERISK AGI Y PHP

Se debe habilitar el módulo de MS SQL en PHP, como primer paso debe instalar PHP y MYSQL en centos. (Innovative tech pros, 2014)

Se debe crear un nuevo archivo dentro de la carpeta

```
/var/lib/asterisk/agi-bin.
```

En este caso se la va a denominar miapp.php

Se debe otorgar permisos mediante `chmod 777, chown asterisk; asterisk` y `chmod + x`, o no podrá ejecutarse el archivo.

Al inicio del archivo debe constar la siguiente línea:

```
#!/usr/bin/php -q
```

Todo código php debe tener este inicio y fin:

```
<?php  
?>
```

Se debe incluir los archivos de `phpagi.php`

```
$ agi = new AGI();  
$ agi-> answer ();
```

Estas son las líneas que requiere cualquier script, se deben colgar todas las llamadas para que no sobrecargue el sistema.

Pasos para configurar llamadas automáticas.

1. Use Flite (asterisk como motor de conversión de texto a voz)
2. Grabe previamente su voz con asterisk y reproduzca el archivo.
3. Use Flite para convertir un archivo de texto en archivo de sonido.

El código es el siguiente:

```
$ result= $agi->get_data('custom/please-enter-your-id', 3000, 15);  
$customerID = $result['result'];
```

Se crea una variable de matriz llamada `$result` que almacena los dígitos devuelto del comando `get_data` de `agi`. Entonces lo que hace este comando es reproducir un archivo de sonido y escucha los dígitos DTMF que presiona la persona que llama.

Entonces lo que tiene que hacer es consultar la base de datos `$customerID`, y en base a eso se le contestara por ejemplo el número de facturas a pagar.

La programación es la siguiente:

```
$myServer = "192.3.4.3";
$myUser = "sa";
$myPass = "ursapassword";
$myDB = "dbName";
$dbhandle = mssql_connect($myServer, $myUser, $myPass)
```

O también:

```
Die("Couldn't connect to SQL Server on $myServer");
$selected=mssql_select_db($myDB, $dbhandle)
```

O

```
Die("Couldn't open database $myDB");
While($numRows==0){
$result = $agi->get_data('custom/please-enter-your-ac-num,3000, 15);
$customerID = $result['result'];
$query = "SELECT*FROM tbl_customers WHERE
CustomerID='$customerID'";
$result = mssql_query($query);
$numRows = mssql_num_rows($result);
}
$query = "SELECT*FROM tbl_invoices WHERE
Status='Unpaid' AND CustomerID='$customerID'";
$result= mssql_query($query);
$numRows = mssql_num_rows($result);
If($numRows == 0){
$agi->exec("Flite", "\No tiene facturas pendientes\");
}

Else{
While($row=mssql_fetch_array($result))
{
$balance = $balance + $row["Balance"];
}
$agi->exec("Flite", "\Su saldo actual es $balance\");
}
Mssql_close($dbhandle);
$agi -> hangup($request['agi_channel']);
```

3.1.7. Configuración de gestor de base de datos phpmyadmin

INSTALACION DEL SERVIDOR LAMP EN UNA RASPBERRY PI

1. **Primero se debe actualizar el sistema operativo Raspbian, el cual es recomendado para Raspberry Pi (Richie, 2017)**

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

2. **Se debe esperar hasta que se instalen todas las actualizaciones del sistema operativo Raspbian para luego instalar los paquetes LAMP. El primer paquete que se va a instalar es Apache2:**

```
sudo apt-get install apache2 -y
```

3. **Luego se habilita el módulo MOD_REWRITE del servidor Apache:**

```
sudo a2enmod rewrite  
sudo service apache2 restart
```

4. **Ahora se modifica una configuración del archivo apache2.conf para permitir anulaciones de .htaccess, para ello se dirige al directorio /var/www y se abre el archivo de configuración con:**

```
sudo nano /etc/apache2/apache2.conf
```

5. **Posteriormente se modifica el archivo apache2.conf cambiando:**

```
AllowOverride None por AllowOverride All
```

6. **Para que se realicen los cambios se debe reiniciar el servidor Apache2 como se muestra:**

```
sudo service apache2 restart
```

7. **Una vez realizado todo lo anterior, se debe conocer la dirección IP de la Raspberry PI, para lo cual se debe realizar de la siguiente manera:**

```
ifconfig eth0
```

8. **La línea que muestra inet addr: 192.xxx.xxx.xxx es la cual indica la dirección IP**

```
192.168.0.104 Dirección IP de la Raspberry PI
```

9. **Se digita la dirección IP de nuestra Raspberry PI en otro computador que esté conectado a la misma red local, en la barra de direcciones del buscador, como se muestra a continuación:**

```
http://192.168.0.104  
Debe aparecer la página por default de APACHE2 DEBIAN
```




Figura 13 SERVIDOR LAMP- Pagina por default de Apache2 Debian

10. Ahora se debe instalar PHP con la siguiente línea de comando:

```
sudo apt-get install php libapache2-mod-php -y
```

11. Después de que se instalen todos los paquetes de PHP se debe verificar que funcione correctamente creando un archivo PHP en la dirección /var/www/html y se lo abre de la siguiente manera:

```
cd /var/www/html  
sudo nano index.php
```

12. En el nuevo archivo index.php se coloca el siguiente código:

```
<?php echo "HELLO WORD"; ?>  
Se guarda con Ctrl + O y se puede salir con Ctrl + X
```

13. Ahora se requiere eliminar el archivo que viene por defecto en APACHE2 DEBIAN, index.html y reiniciar APACHE:

```
sudo rm index.html  
sudo service apache2 restart
```

14. Nuevamente se ingresa la dirección IP en el navegador y debe mostrarse:

"HOLA MUNDO" en una página web en blanco

15. Ahora se debe instalar el motor de la base de datos que es MySQL con la siguiente línea de comando:

```
sudo apt-get install mysql-server php-mysql -y  
sudo service apache2 restart
```

16. Ahora se debe instalar el gestor de la base de datos phpMyAdmin como se muestra a continuación:

```
sudo apt-get install phpmyadmin y
```

Cabe recalcar que para ingresar en phpMyadmin se requiere introducir la contraseña.

17. Ahora se debe comprobar que el gestor de la base de datos phpMyAdmin está funcionando correctamente, para ello se ingresa la dirección IP de la Raspberry PI en la barra de direcciones del buscador web añadiendo phpmyadmin como se indica a continuación:

```
http://192.168.0.140/phpmyadmin/
```

Debe mostrarse la página de inicio de sesión de phpMyAdmin como se muestra en la siguiente imagen.

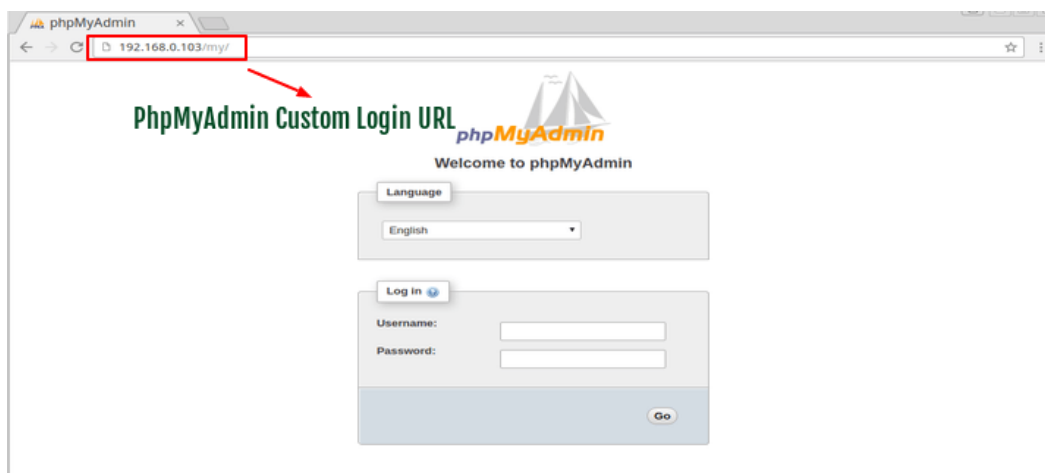


Figura 14 SERVIDOR LAMP- Página de inicio de sesión de phpMyAdmin

Ahora se debe iniciar sesión con el nombre de usuario y contraseña que genero durante la configuración de phpMyAdmin.

18. Instalar y configurar los servicios vsftpd de FTP y bloquear el usuario Raspberry PI en el directorio /var/www como se muestra a continuación:

```
sudo apt-get install vsftpd -y
```

19. Ingresar al archivo vsftpd.conf y comentar las siguientes líneas de código:

```
sudo nano /etc/vsftpd.conf
local_enable=YES
ssl_enable=NO
#local_enable=YES
#ssl_enable=NO
```

20. Agregar las siguientes líneas de código al final del archivo:

```
# CUSTOM
ssl_enable=YES
local_enable=YES
chroot_local_user=YES
local_root=/var/www
user_sub_token=pi
write_enable=YES
local_umask=002
allow_writeable_chroot=YES
ftpd_banner=Welcome to my Raspberry Pi FTP service.
```

21. Añadir el usuario creado al grupo de datos www y otorgar la propiedad de la carpeta /var/www al usuario y conceder los permisos de la siguiente manera:

```
sudo usermod -a -G www-data pi
sudo usermod -m -d /var/www pi
sudo chown -R www-data:www-data /var/www
sudo chmod -R 775 /var/www
```

22. Por último, reiniciar el servicio vsftpd de la siguiente manera:

```
sudo service vsftpd restart
```

23. Ahora el servidor FTP está listo para iniciar sesión, que puede ser mediante FileZilla y configurar una conexión FTP con las siguientes configuraciones: (Richie, 2017)

```
Host: 192.xxx.x.xxx (dirección IP de su Raspberry PI sin prefijo)
Puerto: 21
Protocolo: FTP (Protocolo de transferencia de archivos)
Cifrado: use FTP explícito sobre TLS si está disponible
Tipo de inicio de sesión: Normal (nombre de usuario y contraseña)
Nombre de usuario: pi
Contraseña: [contraseña del usuario]
```

3.1.8. Creación de base de datos para almacenamiento de información de variables y actuadores

Describir el procedimiento de la creación de la base de datos

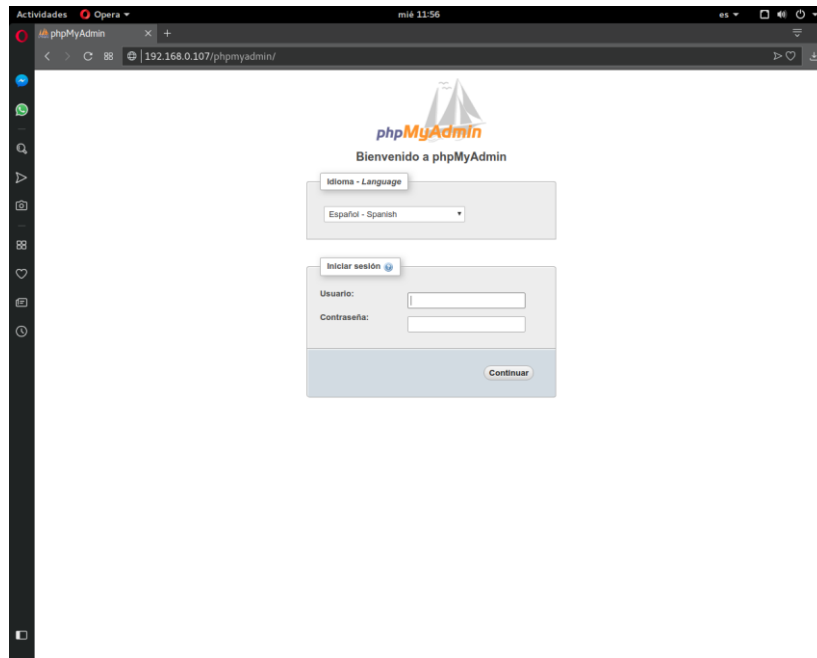


Figura 15 Ingreso al base de datos de phpMyAdmin

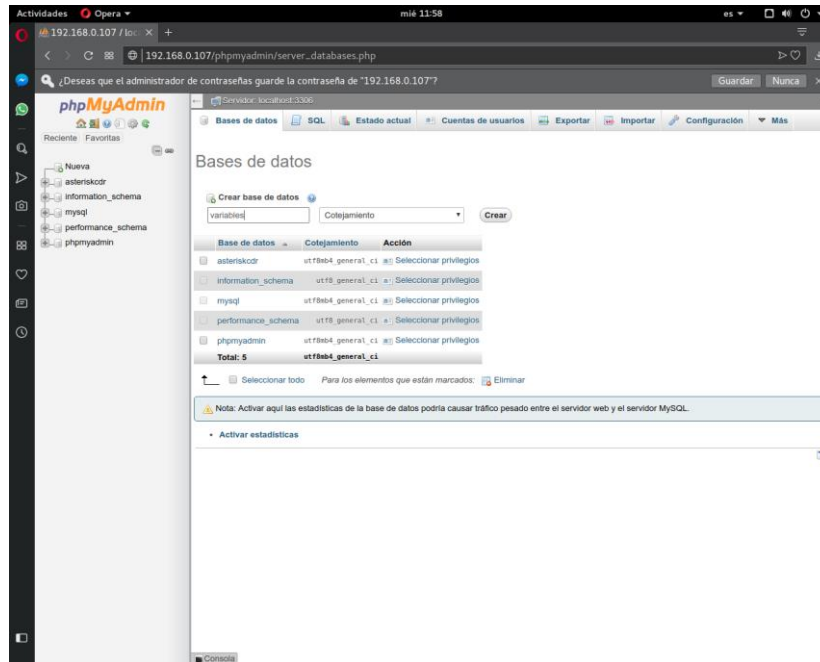


Figura 16 Creación de la base de datos “variables”

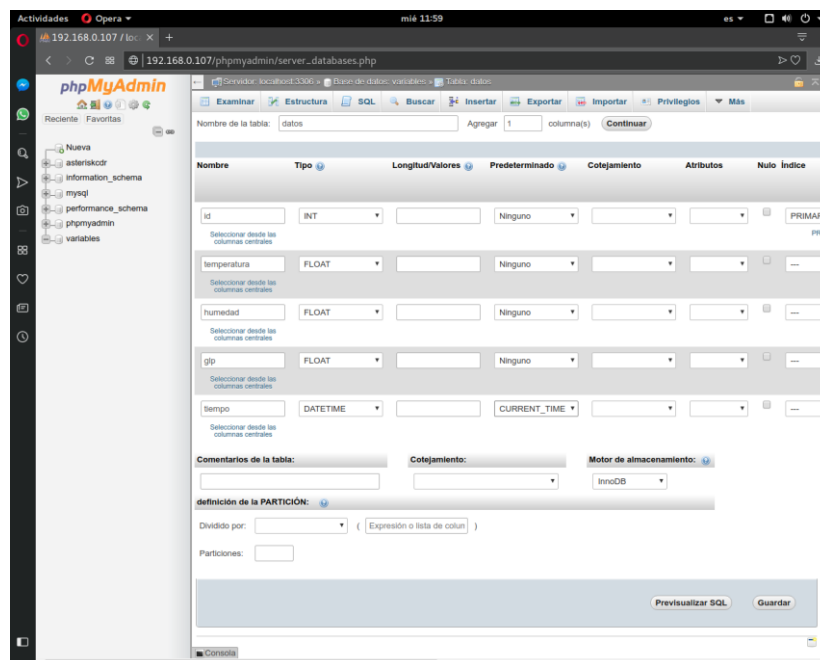


Figura 17 Creación de tablas “datos”

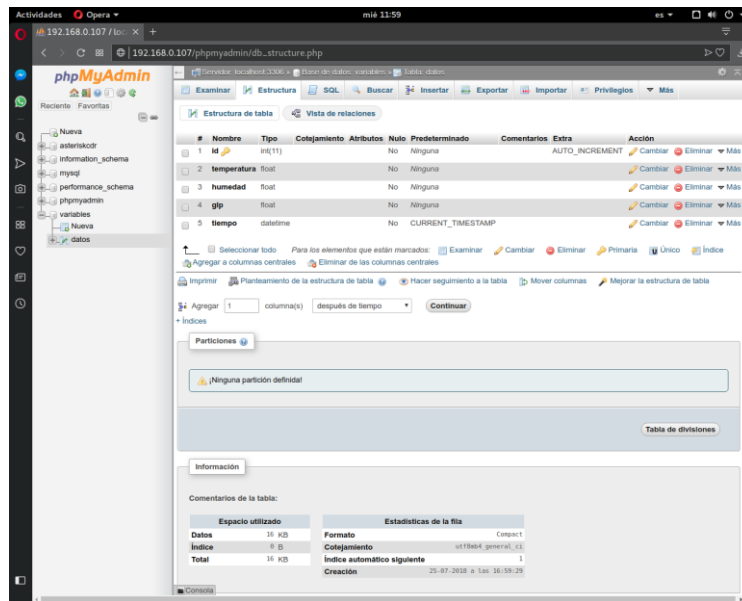


Figura 18 Creación de variables en la tablas “datos”

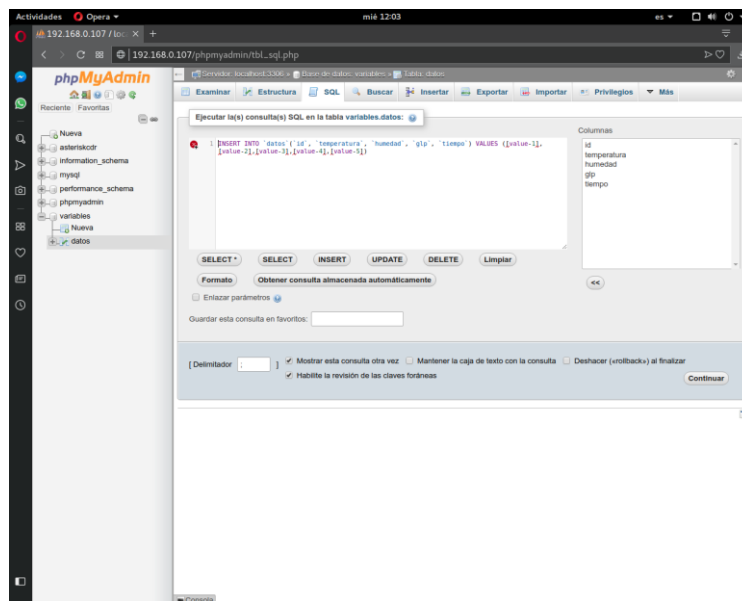


Figura 19 Insertar valores en la tabla "datos" utilizando peticiones SQL

Mostrando filas 0 - 4 (total de 5. La consulta tardó 0.0011 segundos.)

SELECT * FROM `datos`

Perfileando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

id	temperatura	humedad	gpp	tempo
1	12	21	12	2018-07-25 17:04:09
2	17.6	22.5	90	2018-07-25 17:06:09
3	17.6	22.5	90	2018-07-25 17:06:12
4	17.6	22.5	90	2018-07-25 17:06:15
5	17.6	22.5	90	2018-07-25 17:06:18

Operaciones sobre los resultados de la consulta

Guardar esta consulta en favoritos

Etiqueta: Permitir que todo usuario pueda acceder a este favorito

Guardar esta consulta en favoritos

Figura 20 Valores de la tabla "datos"

3.1.9. Configuración de entorno de comunicaciones entre puerto serial y base de datos utilizando Python

3.1.9.1. Serial

En esta ocasión se realizará la lectura de datos mediante Arduino, la misma que consiste en mandar una letra desde la Raspberry pi para saber el estado de datos como un potenciómetro, un botón o un ldr. (Geeky Theory, 2013)

- Si se envía una “P”, el arduino devuelve el valor del potenciómetro.
- Si se envía una “B”, el arduino devuelve el valor del botón.
- Si se envía una “L”, el arduino devuelve el valor del LDR.

El diagrama de conexión del Arduino es el siguiente:

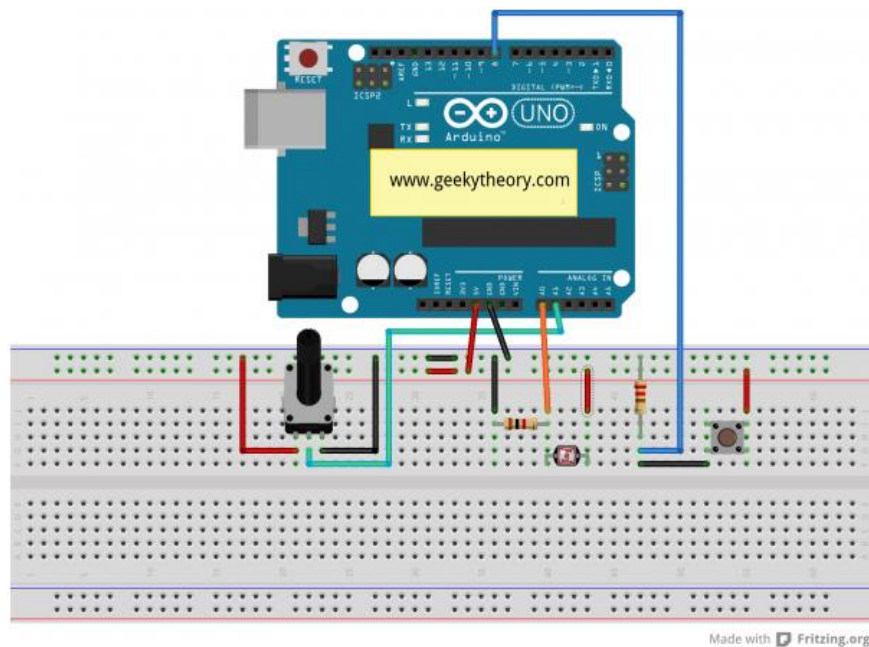


Figura 21 Esquema de conexión del Arduino

En caso de no tener los elementos (pulsador, potenciómetro y LDR), se puede implementar la programación que se muestra en la figura 22, en la cual los mensajes son mostrados por el monitor serial del software Arduino.

```

void setup () {
    Serial.begin(9600); //Inicializo el puerto serial a 9600 baudios
}
void loop () {
    if (Serial.available()) { //Si está disponible
        char c = Serial.read(); //Guardamos la lectura en una variable char
        if (c == 'H') { //Si es una 'H', enciendo el LED
            Serial.println("H' pulsada");
        } else if (c == 'L') { //Si es una 'L', apago el LED
            Serial.println("L' pressed");
        } else {
            Serial.println("Caracter no reconocido");
        }
    }
}

```

Figura 22 Programación del Arduino sin elementos físicos

Si se dispone de los elementos se puede armar el circuito anterior y se puede utilizar la siguiente programación:


```

int LDRPin = 0;
int valor;

int min = 0;
int max = 1023;

int boton = 8;

void setup () {
  Serial.begin(9600);
  pinMode( boton, INPUT );
}

void loop () {
  if(Serial.available()) {
    char c = Serial.read();
    if (c=='B') {
      valor = digitalRead(boton);
      if (valor == 0) {
        Serial.println("Boton no pulsado");
      } else {
        Serial.println("Boton pulsado");
      }
    } else if (c=='L') {
      valor = analogRead(LDRPin);
      valor = map(valor, min, max, 0, 100);
      Serial.println(valor);
    } else if (c=='P') {
      Serial.println(analogRead(1));
    }
  }
}

```

Figura 23 Programación del Arduino con elementos físicos

Mediante el código mostrado en la Figura 23, se puede obtener los valores e imprimirlo en el monitor serial del software Arduino de acuerdo al carácter ingresado.

Ahora se procede a instalar la librería phyton-serial la cual permite utilizar la conexión serial.

```
sudo apt-get install phyton-serial
```

A continuación, se debe crear un archivo en la Raspberry pi el cual permitirá enviar datos por puerto serial al Arduino, el mismo que tendrá el nombre de “Arduino_ReadData.py”.

```
sudo nano Arduino_ReadData.py
```

Aquí se agrega el código que se muestra en la figura 24

```

import serial
import time

arduino=serial.Serial('/dev/ttyACM0',baudrate=9600, timeout = 3.0)
arduino.open()
txt=''

while True:
    var = raw_input("Introducir un Comando: ")
    arduino.write(var)
    time.sleep(0.1)
    while arduino.inWaiting() > 0:
        txt += arduino.read(1)
        print txt
    txt = ''

arduino.close()

```

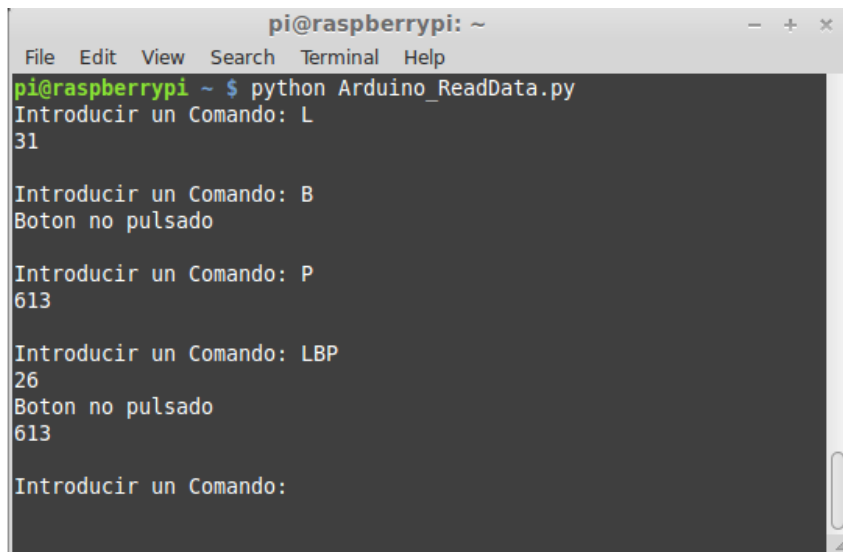
Figura 24 Programación del archivo "Arduino_ReadData.py"

En este código lo único que se realiza es que Arduino_ReadData.py espera a que Arduino se conecte con él, es decir, espera a que se envíe un dato.

Una vez cargado el programa de la figura 23 en el Arduino, se debe ejecutar el archivo de lectura, mediante el siguiente código.

python Arduino_ReadData.py

Se tiene como resultado una ventana tipo consola que requiere que se ingrese el dato.



```

pi@raspberrypi: ~
File Edit View Search Terminal Help
pi@raspberrypi ~ $ python Arduino_ReadData.py
Introducir un Comando: L
31

Introducir un Comando: B
Boton no pulsado

Introducir un Comando: P
613

Introducir un Comando: LBP
26
Boton no pulsado
613

Introducir un Comando:

```

Figura 25 Ejecución del archivo "Arduino_ReadData.py"

Lectura y Escritura desde el puerto serial de Raspberry Pi

En este caso se va a realizar una comunicación serial con el adaptador RS-232 y el adaptador USB-Serial, cabe recalcar que la Raspberry Pi viene configurado el puerto serial como entrada y salida por defecto. (emmeshop, 2015)

Los pines de la Raspberry pi a utilizar son (pin 4 Vcc, pin 6 Gnd, pin 8 Tx. Pin 10 Rx)



Figura 26 Conexión del Raspberry Pi con el adaptador RS-232

El siguiente comando permite saber que puertos seriales están disponibles

```
dmesg grep tty
```

Una vez que se ingresa el comando anterior, se obtiene una respuesta como se muestra en la figura 27

```
pi@raspberrypi ~ $ dmesg | grep tty
[ 0.000000] Kernel command line: dma.dmachans=0x7f35 bcm2708_fb.fbwidth=656 bcm2708_fb.fbheight=416 bcm2709.boardrev=0xa01041 bcm2709.serial=0x93f9c7f9 smsc95xx.macaddr=B8:27:EB:F9:C7:F9 bcm2708_fb.fbswap=1 bcm2709.disk_led_gpio=47 bcm2709.disk_led_active_low=0 sdhci-bcm2708.emmc_clock_freq=250000000 vc_mem.mem_base=0x3dc00000 vc_mem.mem_size=0x3f000000 dwc_otg.lpm_enable=0 console=tty1 console=ttyAMA0,115200 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
[ 0.001774] console [tty1] enabled
[ 0.749509] dev:f1: ttyAMA0 at MMIO 0x3f201000 (irq = 83, base_baud = 0) is a PL011 rev3
[ 1.268971] console [ttyAMA0] enabled
pi@raspberrypi ~ $
```

Figura 27 Puertos serial disponibles de la Raspberry Pi

Como se puede observar en la última línea el puerto ttyAMA0 se encuentra habilitado, por lo cual se procede a deshabilitarlo mediante el siguiente comando:

```
sudo raspi-config
```

De inmediato se abrirán ventanas de configuración como las que va a observar a continuación:
Ahora se debe ingresar a opciones avanzadas como se muestra en la figura 28.

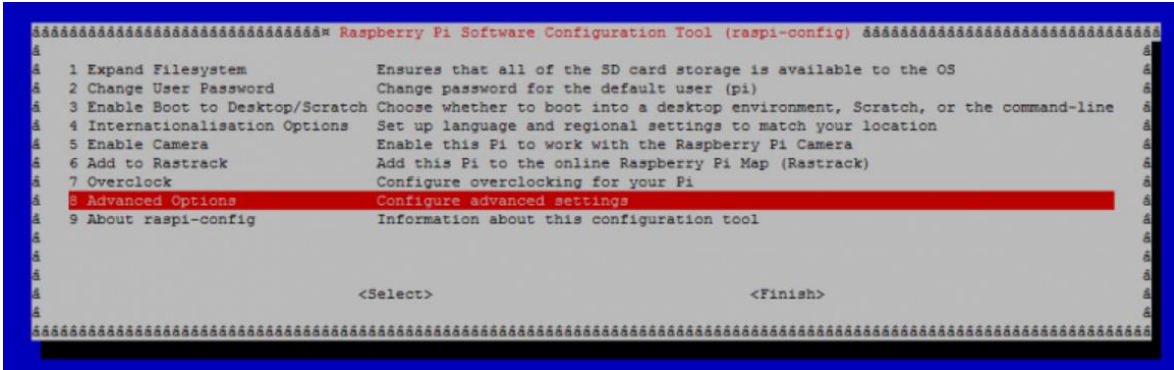


Figura 28 Configuración avanzada de la Raspberry Pi

Ahora se debe ingresar en A8 serial

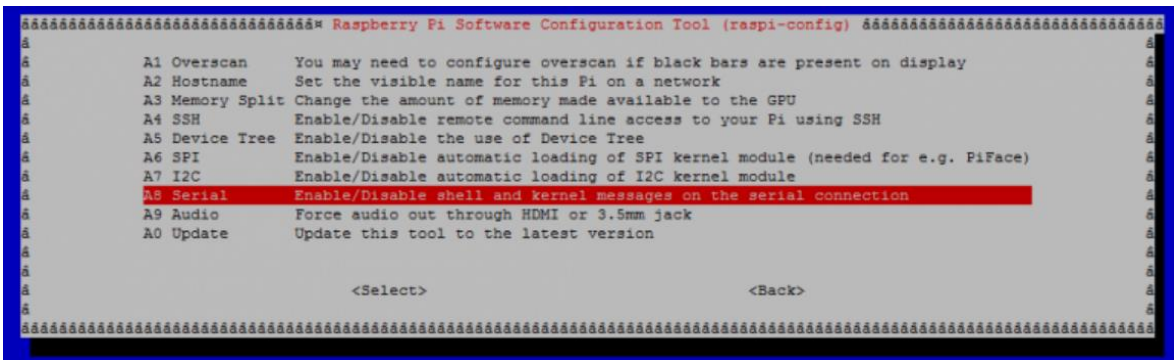


Figura 29 Deshabilitación del puerto serial

Ahora se debe deshabilitar el puerto serial

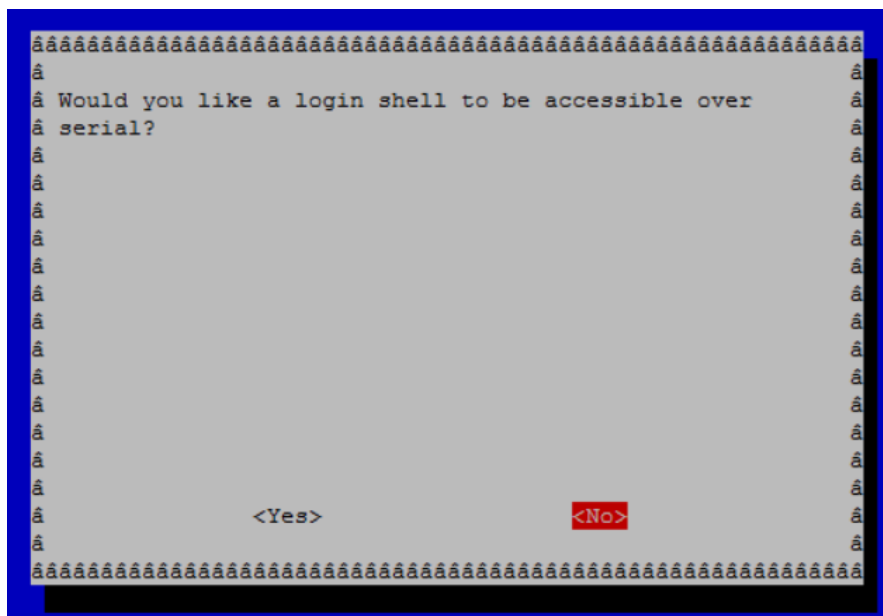


Figura 30 Mensaje de confirmación de deshabilitar el puerto

En esta sección se debe dar ok al mensaje de deshabilitación del puerto serial



Figura 31 Mensaje de Aviso

Se debe confirmar la deshabilitación del puerto serial.

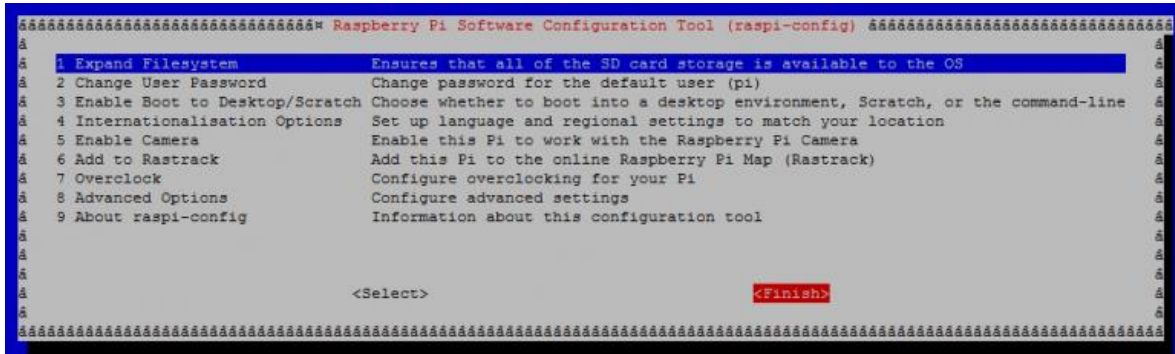


Figura 32 Mensaje de Finalización del proceso

Nuevamente se comprueba si el puerto esta deshabilitado, esto se lo realiza mediante el comando:

```
dmesg grep tty
```

Ahora la respuesta que se obtiene es una similar a la figura 33

```
pi@raspberrypi ~ $ dmesg | grep tty
[ 0.000000] Kernel command line: dma.dmachans=0x7f35 bcm2708_fb.fbwidth=656 bcm2708_fb.fbheight=416 bcm2709.boardrev=0xa01041 bcm2709.serial=0x93f9c7f9 smsc95xx.macaddr=B8:27:EB:F9:C7:F9 bcm2708_fb.fbswap=1 bcm2709.disk_led_gpio=47 bcm2709.disk_led_active_low=0 sdhci-bcm2708.emmc_clock_freq=250000000 vc_mem.mem_base=0x3dc00000 vc_mem.mem_size=0x3f000000 dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
[ 0.001769] console [tty1] enabled
[ 0.749438] dev:f1: ttyAMA0 at MMIO 0x3f201000 (irq = 83, base_baud = 0) is a PL011 rev3
pi@raspberrypi ~ $
```

Figura 33 Estado de los puertos seriales de la Raspberry Pi

Como se puede observar ya no aparece el puerto ttyAMA0 habilitado, es decir ahora si se puede utilizar, para lo cual se debe conectar el adaptador USB/serial, ahora se establece una comunicación entre los dos puertos.



Figura 34 Conexión del puerto Serial de la Raspberry Pi

Ahora se requiere saber que nombre le asigna la Raspberry Pi al puerto serial, para ello se digita el siguiente comando:

```
dmesg grep tty
```

Se logra tener una salida como la que indica la figura 35

```
pi@raspberrypi ~ $ dmesg | grep tty
[ 0.000000] Kernel command line: dma.dmachans=0x7f35 bcm2708_fb.fbwidth=656 bcm2708_fb.fbheight=416 bcm2709.boardrev=0xa01041 bcm2709.serial=0x93f9c7f9 smsc95xx.macaddr=B8:27:EB:F9:C7:F9 bcm2708_fb.fbswap=1 bcm2709.disk_led_gpio=47 bcm2709.disk_led_active_low=0 sdhci-bcm2708.emmc_clock_freq=250000000 vc_mem.mem_base=0x3dc00000 vc_mem.mem_size=0x3f000000 dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
[ 0.001769] console [tty1] enabled
[ 0.749438] dev:f1: ttyAMA0 at MMIO 0x3f201000 (irq = 83, base_baud = 0) is a PL011 rev3
[ 971.919417] usb 1-1.2: pl2303 converter now attached to ttyUSB0
pi@raspberrypi ~ $
```

Figura 35 Estado de los puertos seriales de la Raspberry Pi

Ahora se procede a crear dos archivos, el uno que permite escribir en el puerto ttyAMA0 y el otro va a permitir leer el puerto ttyUSB0, para ello se utiliza el siguiente comando:

```
sudo nano serial_write.py
```

En el cual se va a colocar el siguiente código:

```

#!/usr/bin/env python
import time
import serial
ser = serial.Serial(
port = '/dev/ttyAMA0',
baudrate = 9600,
parity = serial.PARITY_NONE,
stopbits = serial.STOPBITS_ONE,
bytesize = serial.EIGHTBITS,
timeout = 1
)
counter = 0
while 1:
    ser.write('Write counter: %d \n' %(counter))
    time.sleep(1)
    counter +=1

```

Serial_read.py

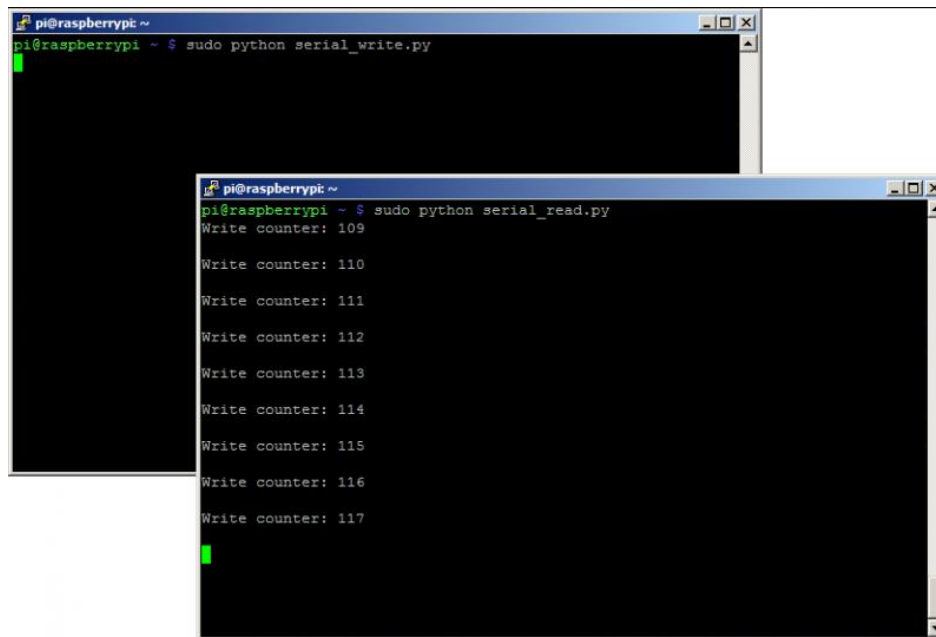
```

#!/usr/bin/env python

import time
import serial
ser=serial.Serial(
port='/dev/ttyUSB0',
baudrate = 9600,
parity = serial.PARITY_NONE,
stopbits = serial.STOPBITS_ONE,
bytesize = serial.EIGHTBITS,
timeout = 1
)
counter = 0
while 1:
    x=ser.readline()
    print x

```

Si se ejecuta los dos archivos “*serial_write.py*” y “*Serial_read.py*” se abrirá dos ventanas como las siguientes.



```
pi@raspberrypi ~  
pi@raspberrypi ~ $ sudo python serial_write.py  
  
pi@raspberrypi ~ $ sudo python serial_read.py  
Write counter: 109  
  
Write counter: 110  
  
Write counter: 111  
  
Write counter: 112  
  
Write counter: 113  
  
Write counter: 114  
  
Write counter: 115  
  
Write counter: 116  
  
Write counter: 117
```

Figura 36 Ventanas de la comunicación serial

3.1.9.2. Conexión a base de datos

Para realizar la conexión con la base de datos, se realiza el siguiente procedimiento: (Learn Python, s.f.)

1. Importar el conector:

Utilizar la librería **import MySQLdb**

2. Conectarse a la base de datos:

Declarar un objeto para establecer la conexión con la base de datos, utilizando la función **connect(parámetros...)**, conformada por los siguientes parámetros:

host

Nombre del host para conectarse. Predeterminado: usar localhost a través de un socket UNIX (cuando corresponda)

user

Usuario para autenticar como. Predeterminado: usuario actual.

passwd

Contraseña para autenticar con. Predeterminado: sin contraseña.

db

Base de datos a utilizar. Predeterminado: no hay base de datos predeterminada.

port

Puerto TCP del servidor MySQL. Predeterminado: puerto estándar (3306).

3. Utilizar un método cursor de conexión

Un cursor es una estructura de control que se usa para recorrer (y eventualmente procesar) los records de un result set.

Ejemplo para declarar un cursor:

```
cursor = db.cursor()
```

4. Realizar una consulta

En este apartado se realiza la conexión SQL.

Ejemplo que determina la versión de SQL:

```
cursor.execute("SELECT VERSION()")
```

5. Capturar los datos

Para obtener los datos de la base se utiliza las siguientes funciones:

```
fetchone()
```

Obtiene un solo elemento

```
fetchall()
```

Utilizar esta función para adquirir varios datos

6. Cerrar el cursor

Permite cerrar la conexión con el servidor

A continuación, se presenta un ejemplo completo de conexión a la base de datos, utilizando un script de Python:

```
#!/usr/bin/python
```

```
#Importar la librería que permite la conexión
```

```
import MySQLdb
```

```
# Establecer conexión la base de datos

db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )

# Preparar el objeto cursor

cursor = db.cursor()

# Ejecutar la consulta SQL

cursor.execute("SELECT VERSION()")

# Captar los datos

data = cursor.fetchone()

print "Database version : %s " % data

#cerrar conexión

db.close()
```

3.2. Desarrollo de sistema de adquisición de señales analógicas

3.2.1. Sensor de humedad y temperatura DTH22

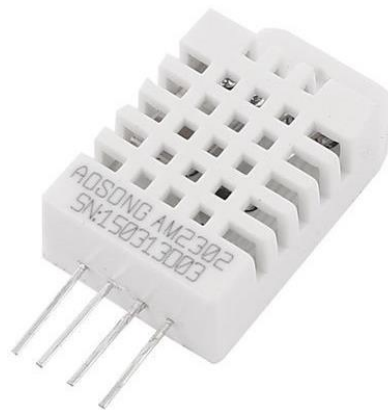


Figura 37 Módulo DTH22

En la tabla 2, se indica las características del sensor entre las que se cita:

- Salida de señal digital calibrada. (Liu, s.f.)
- Tecnología de detección, asegurando su fiabilidad y estabilidad.
- Tamaño pequeño de bajo consumo y larga distancia de transmisión (20m) habilitada.

Tabla2

Características técnicas del módulo DTH22

Modelo	DHT22
Fuente de alimentación	3.3-6V DC
Señal de salida	señal digital a través de un solo bus
Elemento sensor	Condensador de polímero
Rango de operación	humedad 0-100% HR; temperatura -40 ~ 80 grados celsius
Exactitud	humedad + -2% RH (Max + -5% RH); temperatura <+ - 0.5 grados Celsius
Resolución o sensibilidad	humedad 0.1% RH; temperatura 0.1 grados celsius
Repetibilidad	humedad + -1% RH; temperatura + -0.2 Celsius
Histéresis de humedad	+ -0.3% RH A largo plazo
Estabilidad	+ -0.5% RH / año
Período de detección	Promedio: 2s
Dimensiones	Tamaño pequeño 14*18*5.5mm; Tamaño grande 22*28*5mm

El sensor internamente se encuentra en la capacidad de proporcionar la información de las variables del ambiente, utilizando un bus de una sola vía que puede ser leído a través de un microcontrolador, mediante el uso de librerías *1-Wire* que habitualmente proporcionan fabricantes como MAXIM o DALLAS

La estructura del mensaje proporcionado por el sensor está descrita de la siguiente forma:

DATA=16 bits Humedad Relativa + 16 bits Temperatura + 8 bits check-sum

Es decir, cada paquete de información contiene una secuencia de bits que son entregado hacia el microcontrolador en el orden establecido con anterioridad. Por ejemplo:

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

Humedad Relativa + Temperatura + Check Sum

Para obtener la humedad relativa solo es necesario convertir del sistema binario al decimal y dividir el valor resultante para 10:

$$0000\ 0010\ 1000\ 1100 = 652$$

$$\mathbf{HR = 652 / 10 = 65.2\%}$$

Para el caso de la temperatura, ha de desarrollarse el mismo principio de cálculo para obtener el valor deseado. A diferencia de que, en el caso de que el valor del bit más significativo sea 1 lógico, el valor calculado estaría por debajo del cero (grados centígrados negativos).

$$0000\ 0001\ 0101\ 1111 = 351$$

$$\mathbf{T(\text{centígrados}) = 351/10 = 35.1^{\circ}\text{C}}$$

Finalmente, para el caso del check sum, habrá de realizarse una suma binaria Byte por Byte de la información proporcionada por el sensor que tiene que ser idéntico al valor adjunto como check sum y de esta manera validar el mensaje.

0000 0010+1000 1100+0000 0001+0101 1111 = 1110 1110 = Check Sum

3.2.2. Algoritmos de adquisición de humedad y temperatura

Para la obtención de los valores de temperatura y humedad en el caso de estudio, se ha utilizado una librería que se basa en el método anteriormente descrito. El procedimiento para adquirir la humedad y temperatura se basa en el algoritmo descrito en la Figura 38

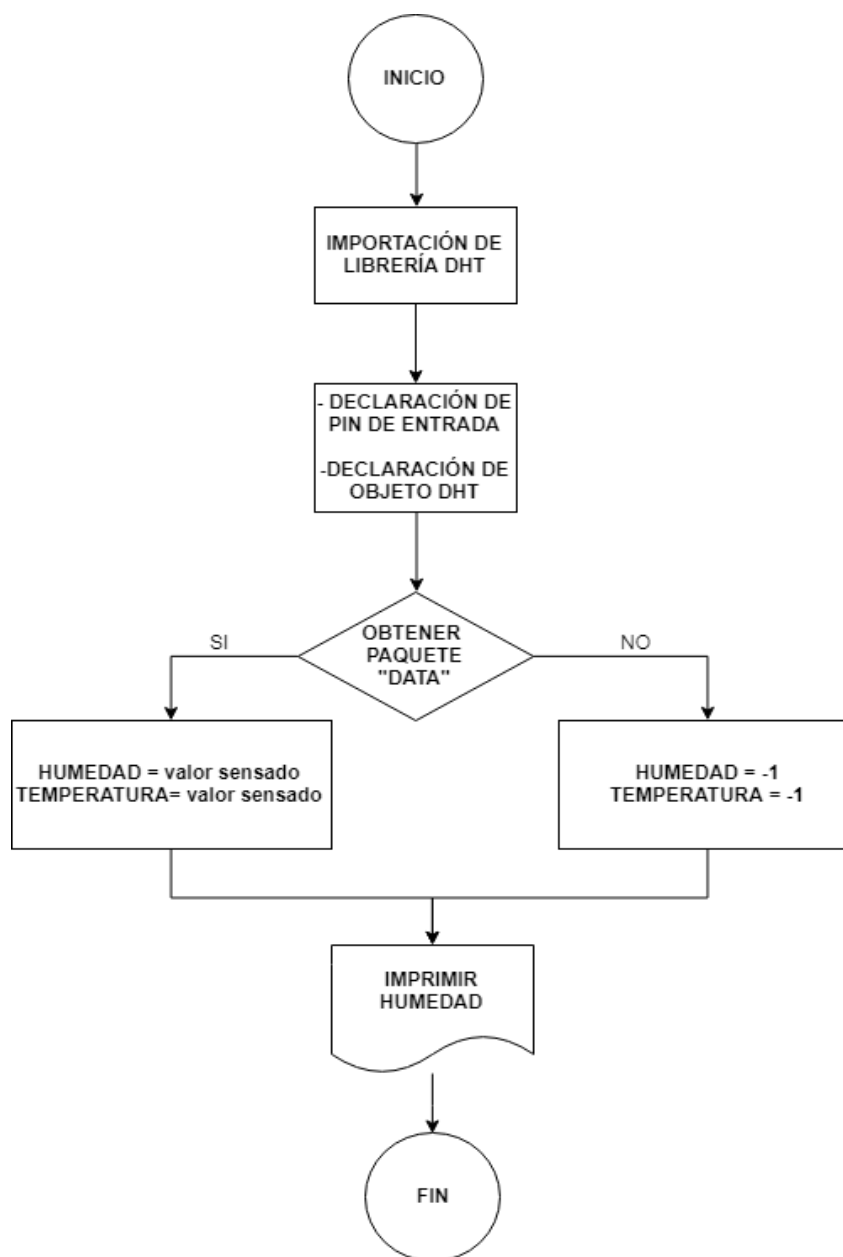


Figura 38 Algoritmo de adquisición de humedad y temperatura

3.2.3. Sensor de Gas Licuado de Petróleo

Utilizado en la detección de GLP, i-butano, propano, metano. La sensibilidad se calibra con un potenciómetro. (sparkfun, 2003)

Características:

- Alimentación: 5V
- Tipo de interfaz: Analógico.
- Pin Definición: 1-salida, 2 GND, 3-VCC
- Amplio alcance de detección.
- Respuesta rápida y alta sensibilidad.
- Baja sensibilidad al alcohol y al humo
- Circuito de accionamiento sencillo.
- Tamaño: 40x20mm.

La curva característica de sensado del dispositivo MQ-6 presentada en la figura 39, se comporta de forma relativamente lineal al situarlo en entornos saturados de GLP o CH₄, a tal punto que es posible obtener un valor resultante de la cantidad de partículas por millón del entorno a través procesamiento matemático

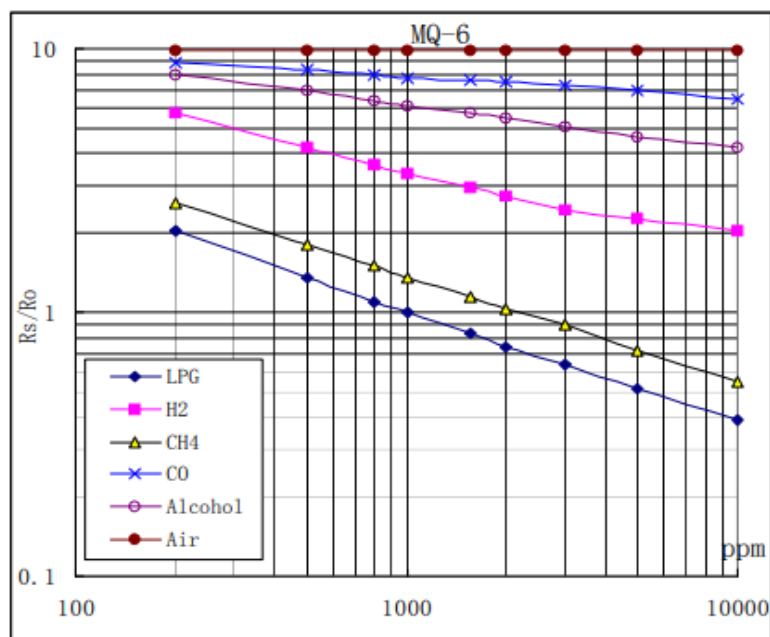


Figura 39 curva característica del sensor MQ6

Hay que tomar en cuenta que, según el fabricante, el valor de resistencia de MQ-6 es la diferencia entre varios tipos de concentración gases. Entonces, cuando se usan sus componentes, el ajuste de sensibilidad es muy necesario. Y es recomendable calibrar el detector para 1000 ppm de concentración de GLP en el aire y utilizar el valor de resistencia de carga (RL) de aproximadamente 20 K Ω (rango aceptable de 10 K Ω a 47 K Ω).

3.2.4. Algoritmos de adquisición de GLP y CH4

En cuanto a la obtención de la cantidad de partículas por millón se ha adaptado el algoritmo de cálculo (ver Figura 40) para que sea capaz de proporcionar valores precisos en base a la curva característica descrita en el párrafo anterior.

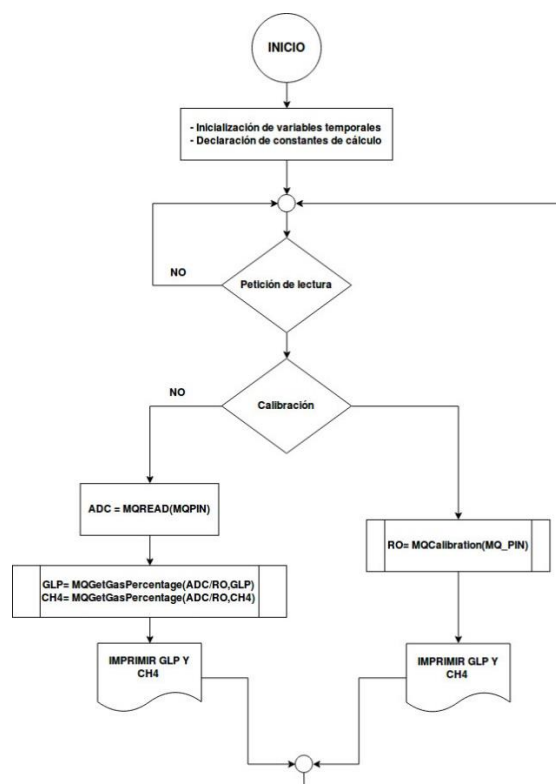


Figura 40 Algoritmos de adquisición de GLP y CH4

En cuanto a los métodos descritos en la figura 40, se componen en su mayoría del proceso matemático que requiere la señal analógica adquirida para obtener un valor resultante en ppm.

El método MQResistanceCalculation que obtiene la resistencia calculada del sensor está compuesta por:

- Entrada: raw_adc - valor bruto leído de adc, que representa el voltaje
- Salida: la resistencia calculada del sensor

Observaciones: El sensor y la resistencia de carga forman un divisor de voltaje. Dado el voltaje a través de la resistencia de carga y su resistencia, la resistencia del sensor podría ser derivado.

El método MQCalibration que contribuye en la calibración del sensor en un entorno libre de contaminación se compone de:

- Entrada: mq_pin - canal analógico al que se encuentra conectado el sensor
- Salida: Ro del sensor

Observaciones: Esta función supone que el sensor está en aire limpio. Utiliza MQResistanceCalculation para calcular la resistencia del sensor en aire limpio y luego lo divide con RO_CLEAN_AIR_FACTOR/10; esta última constante de división (10) difiere ligeramente entre sensores.

En cuanto al método MQRead está compuesto por:

- Entrada: mq_pin - canal analógico

- Salida: Rs del sensor

Observaciones: Esta función usa MQResistanceCalculation para calcular la resistencia del sensor (Rs). El Rs cambia a medida que el sensor está en la diferente concepción del gas objetivo

El método MQGetGasPercentage se encuentra compuesto por:

- Entrada: rs_ro_ratio - Rs dividido por Ro gas_id - tipo de gas objetivo
- Salida: ppm del gas objetivo

Observaciones: Esta función pasa diferentes curvas a la función MQGetPercentage que el ppm (partes por millón) del gas objetivo.

Finalmente, el método MQGetPercentage del cual depende el anterior método. Se encarga de relacionar la curva característica de determinado gas y proporcionar una respuesta matemática de tal forma que pueda ser entendida por el cliente que consuma la información.

Esta función se compone de:

- Entrada: rs_ro_ratio - Rs dividido por Ro pcurve - puntero a la curva del gas objetivo
- Salida: ppm del gas objetivo

Observaciones: al usar la pendiente y un punto de la línea. La x (valor logarítmico de ppm) de la línea podría derivarse si se proporciona y (rs_ro_ratio). Como es una coordenada logarítmica, potencia de 10 se usa para convertir el resultado en no logarítmico

3.2.5. Algoritmos de almacenamiento de variables analógicas (ARDUINO - Rpi - DB)

Una vez adquiridas, linealizadas y debidamente procesadas las variables de entorno a través de la unidad de microcontrol. Se ha considerado necesario el envío de los valores hacia una base de datos, con la finalidad de generar persistencia de la información y posibilitar el acceso a la misma en cualquier momento y desde cualquier aplicativo web por parte del cliente.

Por lo tanto, se han implementado algoritmos de envío de información a través del puerto UART de la unidad de microcontrol, receptorla en un puerto serial del ordenador que aloja el servidor central del sistema y procesarla para ser almacenada correctamente ordenada, a través de peticiones SQL en una base de datos. El algoritmo encargado de este procedimiento se describe como se presenta en la figura 41

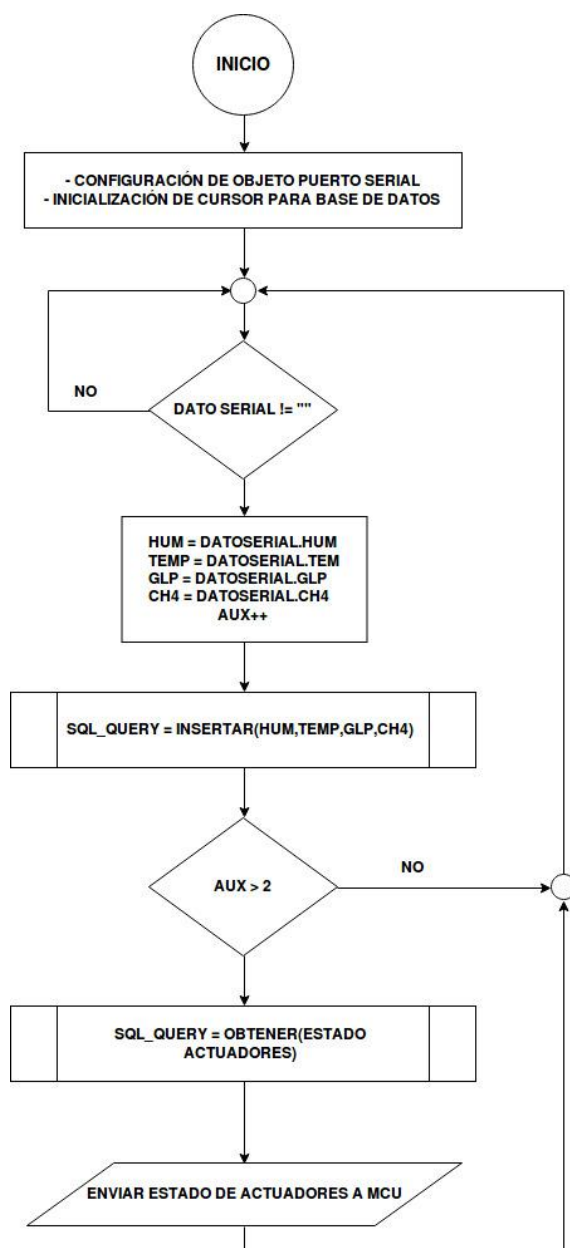


Figura 41 Algoritmo de almacenamiento de datos analógicos

3.3. Desarrollo de algoritmos para la conversión datos a voz (DB-Rpi-RedCelular)

Para el caso de estudio, al requerir que el sistema se encuentre en la capacidad de proporcionar información sobre las variables adquiridas por los sensores a través de la IVR; se ha considerado que es de vital importancia, implementar algoritmos de obtención de la

información almacenada en la base de datos en tiempo real (Ver figura 42) y convertirla en mensajes de voz para atender a las peticiones que un cliente realice a través de la red de telefonía celular. El proceso de obtención de las variables cumple una secuencia determinada de la siguiente forma:

- A. Adquirir la información almacenada en las tablas de variables de base de datos a través de herramienta de software PHP-AGI.
- B. Conversión de información de texto a datos a través de herramienta de software FESTIVAL
- C. Envío de información a través de la red IP vinculada a la red celular utilizando Asterisk y CHAN-Dongle.

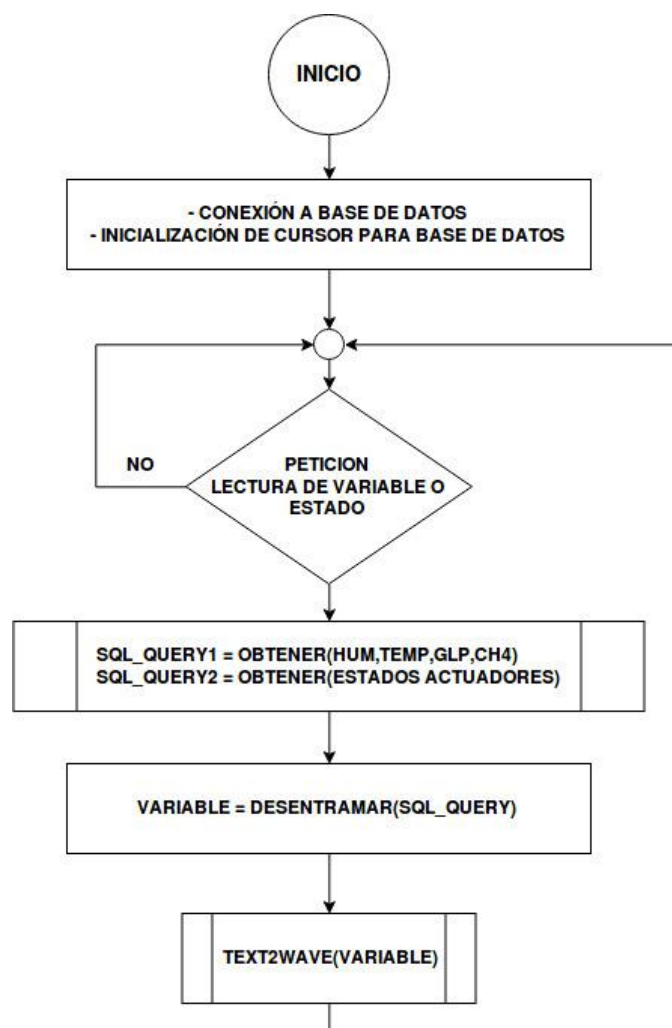


Figura 42 Algoritmo de conversión de texto a voz

3.4. Desarrollo de algoritmos para la conversión datos a órdenes para microcontrol (RedCelular-DB-Rpi-Arduino)

Concluyendo con el apartado de algoritmos, la etapa de adquisición de los datos de control enviados hacia la unidad de microcontrol para tareas de activación de periféricos. Se han desarrollado con el principio inverso utilizado para el almacenamiento de las variables, por las condiciones de software y hardware que limitan la posibilidad de compartir información de variables de forma autónoma. En tal

virtud, se ha utilizado a la base de datos como intermediaria para la tarea descrita siguiendo la siguiente secuencia:

- D. Adquirir la información emitida por el cliente a través de la herramienta PHP-AGI.
- E. Actualizar la información en tabla de la base de datos correspondiente a las señales de control.
- F. Verificar cambios en las variables de control a través de algoritmos python.
- G. Enviar la señal de cambio a la tarjeta de microcontrol a través del puerto UART.

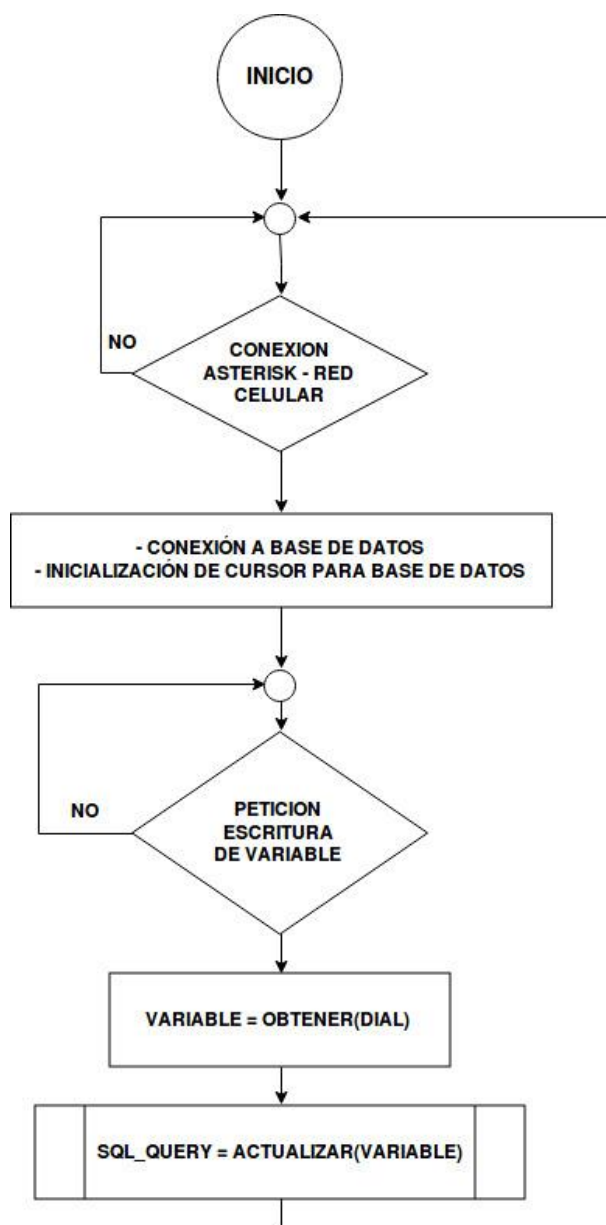


Figura 43 Algoritmo de conversión datos a órdenes para microcontrol

3.5. Desbloqueo de dispositivo de acceso a servicio VoIP a través de redes GSM

El dispositivo USB modem a desbloquear y que permita el acceso VOIP a través de la red GSM es Huawei E160, soportando alta velocidad de descarga.



Figura 44 Modem Huawei E160

Los pasos para desbloquear Huawei E160 Modem son:

1. Descarga el firmware de Huawei E160.
2. Descargar el software Universal Master Code.exe. Ingrese el número IMEI de 15 dígitos. El programa dará el código de desbloqueo y código Flash
3. Conectar el modem a la computadora.
4. Detectado el modem, ejecutar el archivo E160 Update.exe para actualizar el firmware, el software pedirá el código de desbloqueo y código Flash.

3.6. Implementación de algoritmos de conexión entre servidor VOIP y red GSM (GSM - BASE - VOZ - GSM)

El algoritmo de mayor relevancia en el sistema de control y monitoreo implementado (Ver figura 45), es el responsable de obtener los comandos enviados a través del terminal móvil del cliente (encargado del monitoreo y control), procesar el dato

comando y generar una petición SQL para adquirir o cambiar el estado de un actuador. En el caso de la obtención de la información, la respuesta de la petición SQL retorna un vector que contiene la información necesaria que ha de ser enviada al algoritmo de conversión de texto a voz descrito en anteriores apartados.

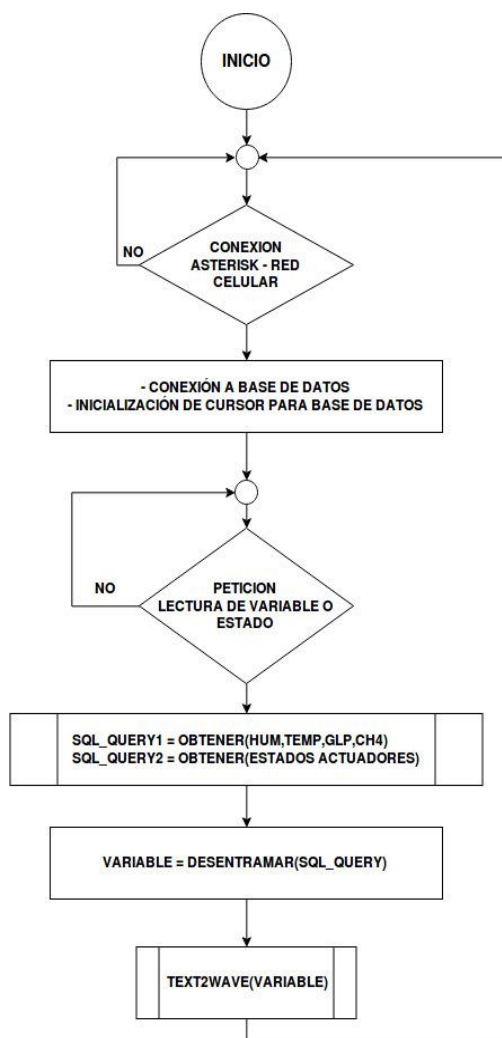


Figura 45 Algoritmo de conexión VOIP y red GSM

3.7. Diseño electrónico de sistema de adquisición.

Con la finalidad de desarrollar pruebas en un entorno real con condiciones muy cercanas a las que se puedan presentar dentro de un entorno industrial de una pequeña o mediana empresa, se ha diseñado un prototipo electrónico para validar el funcionamiento. En la figura. 46 se describe el sistema electrónico a manera de escudo de adaptación para la unidad de microcontrol (Arduino Mega R3) del prototipo implementado para el control de dispositivos y adquisición de las variables analógicas.

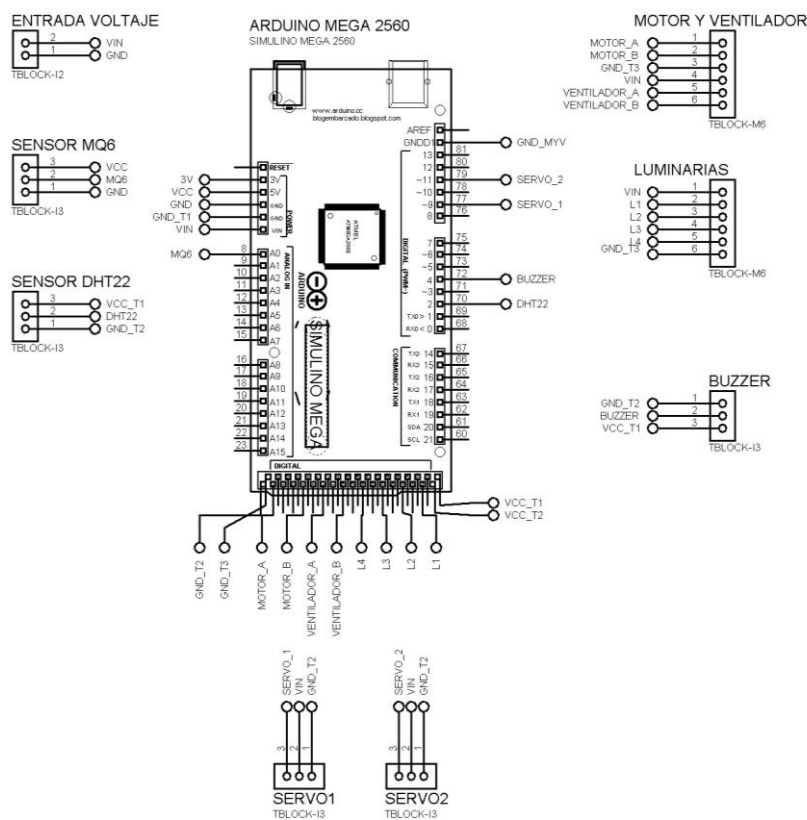


Figura 46 Diagrama esquemático del prototipo de adquisición y control

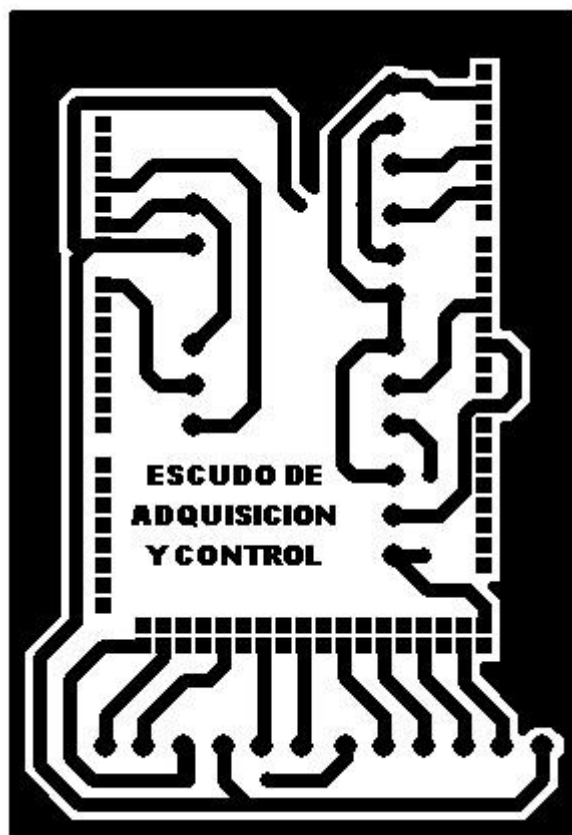


Figura 47 Diagrama PCB de adaptación para la unidad de microcontrol

3.8. Diagrama esquemático de topología implementada

El sistema aplicado al control y monitoreo en las PYMES a través de la vinculación de la tecnología de VOIP-GSM y la estructura de las redes IOT cumple una estructura descrita en la figura 48. De tal forma que un usuario se encuentre en la capacidad de acceder a una central de Voz sobre IP a través del modem GSM liberado y controlar los periféricos de la unidad de microcontrol vinculadas directamente los dispositivos electrónicos del sistema simulado; además de obtener en forma directa los valores procesados de las variables analógicas y darlas a conocer a través del enlace de voz que se ha generado entre la terminal del cliente y modem GSM enlazado a la central VoIP.

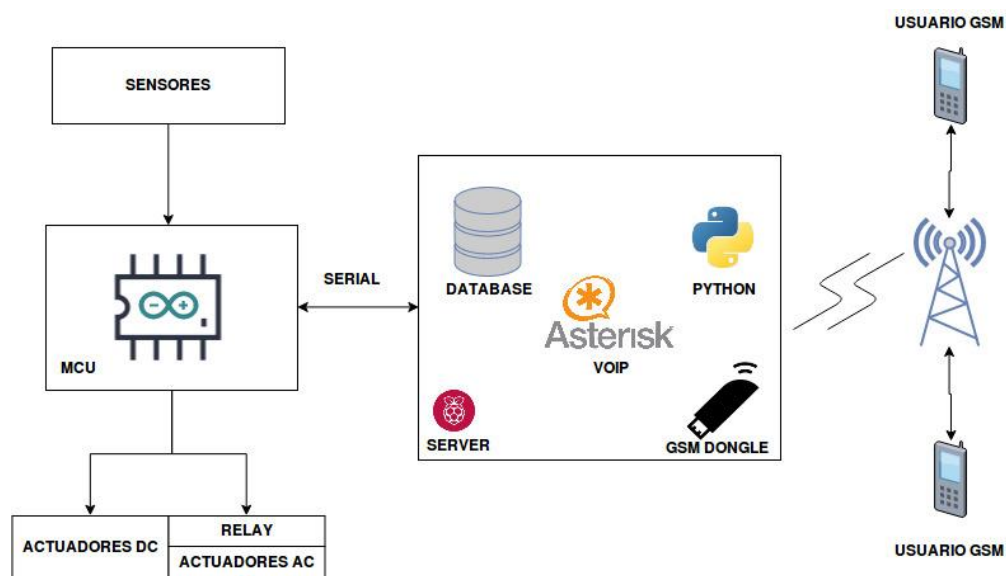


Figura 48 Diagrama de esquemático de la topología de control implementada

3.9. Diseño de maqueta de experimentación

El entorno de pruebas de desarrolló completamente sobre una maqueta de experimentación a escala, descrito en la figura 49 y fue desarrollado de tal forma que se asemeje a una mediana empresa que cuenta con varios ambientes de trabajo. Entre los ambientes considerados se encuentran:

- un taller en el que se exponen a gases como GLP o CH₄ y además requiera un control especial sobre la humedad y temperatura constante.
- Oficinas ejecutivas como despacho y recepción
- Un área de control en el que se encuentran los dispositivos reales encargados del sistema.
- Y servicios higiénicos que requieren de ventilación prolongada

Considerado dentro del sistema de control, a la iluminación y control de puertas principales.

Proporcionando un método confiable de la gestión energética que permita reducir costos o

a su vez proporcionar una herramienta de simulación de presencia o control remoto para micro empresarios sin requerir de sistemas de alto costo y software y hardware de uso comercial.

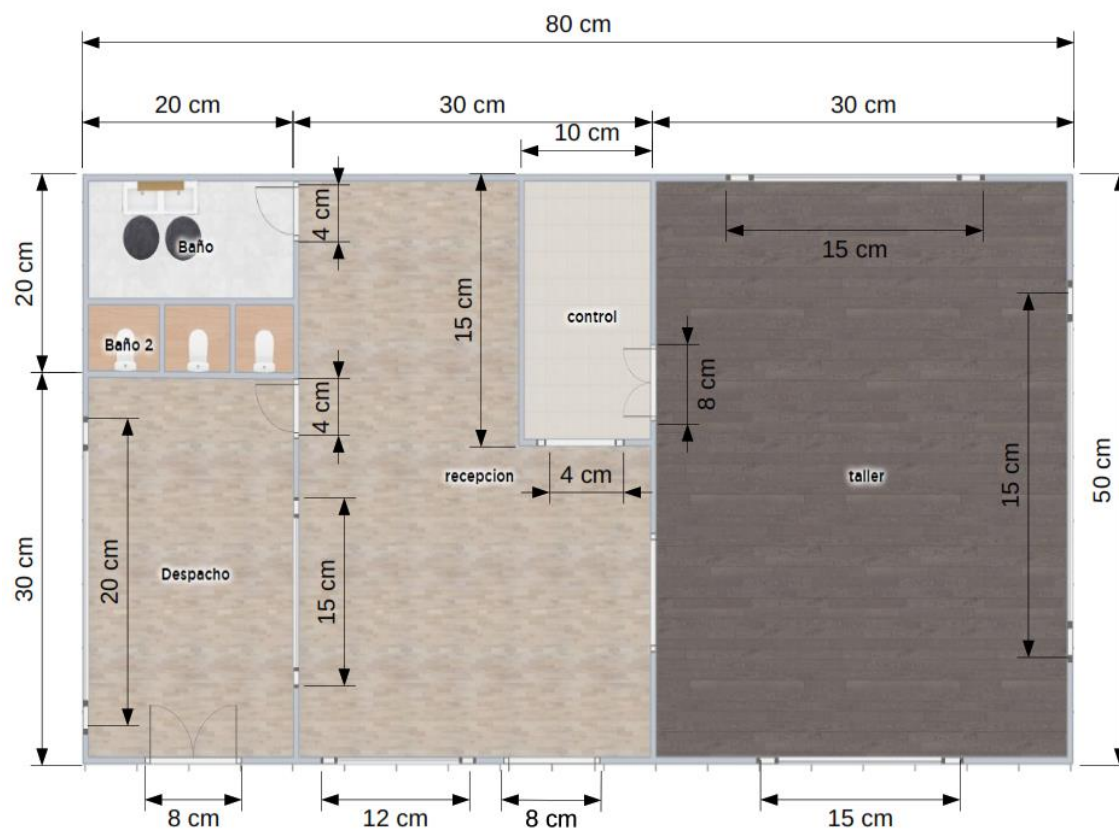


Figura 49 Diagrama de maquetación de pruebas y dimensionamiento a escala de una mediana empresa de referencia

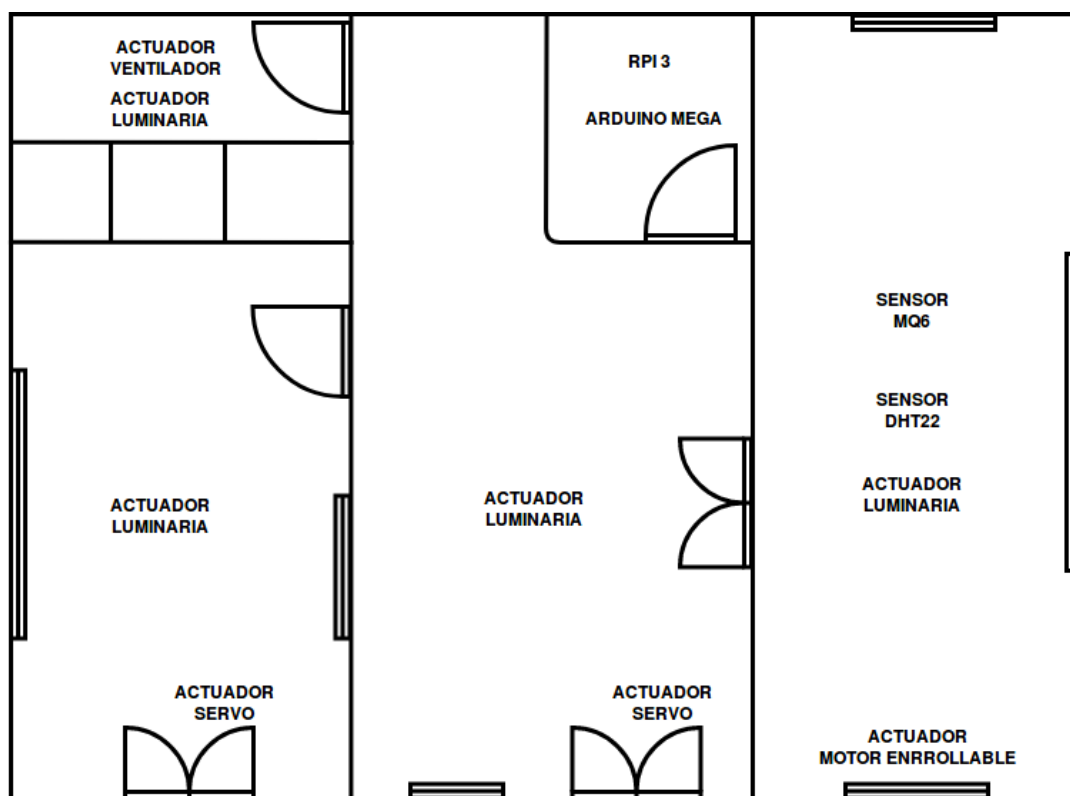


Figura 50 Posición de elementos de control y adquisición

CAPÍTULO IV

PRUEBAS Y RESULTADOS DE LA INVESTIGACIÓN

4.1. Pruebas de funcionamiento del sistema

Culminada la etapa de implementación, el sistema fue sometido a una serie de pruebas de la adquisición y control. En las siguientes figuras se describe el procedimiento de un ciclo de trabajo que permita comprobar el óptimo funcionamiento del sistema implementa; demostrando que puede ser replicado, adaptado y/o modificado para satisfacer las necesidades de una pequeña o mediana empresa de forma independiente de las actividades de la empresa.

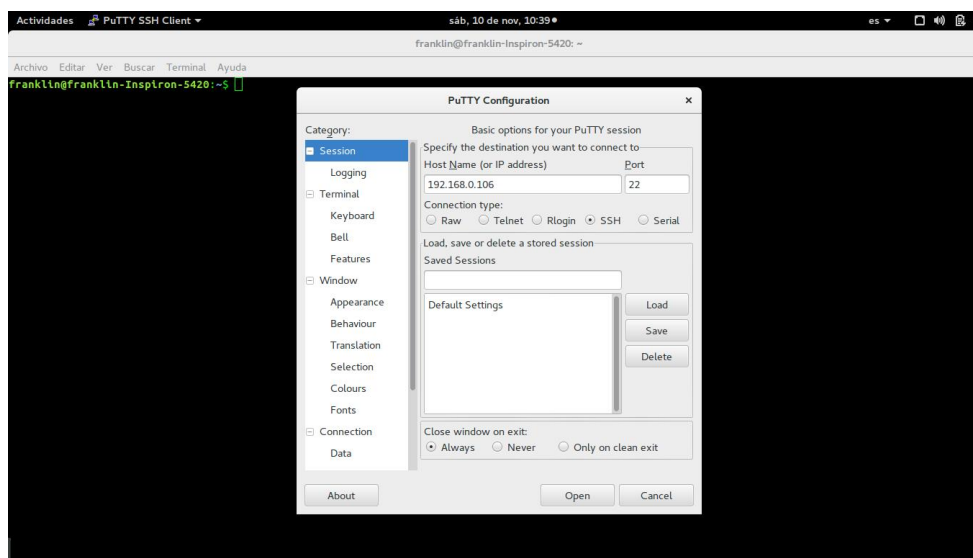


Figura 51 Prueba de conexión al servidor a través de terminal segura

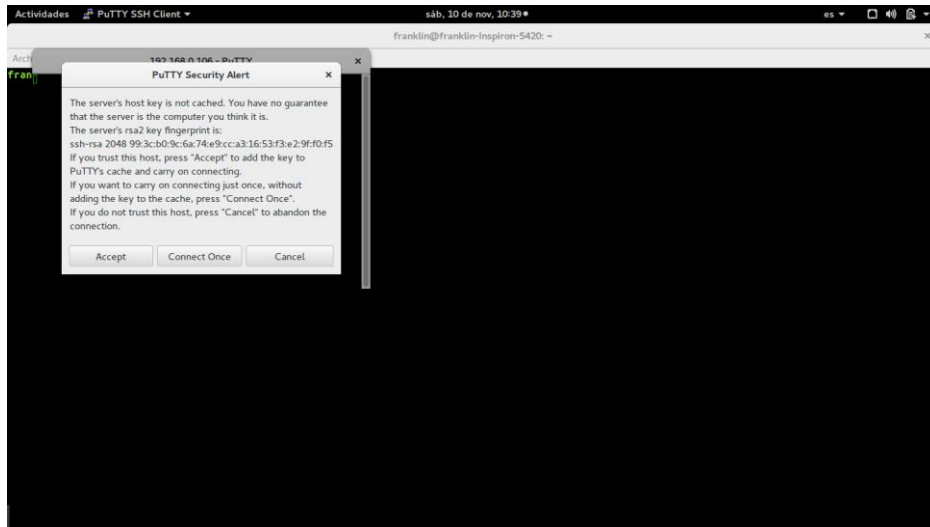


Figura 52 Petición de almacenamiento de llave para conexión segura

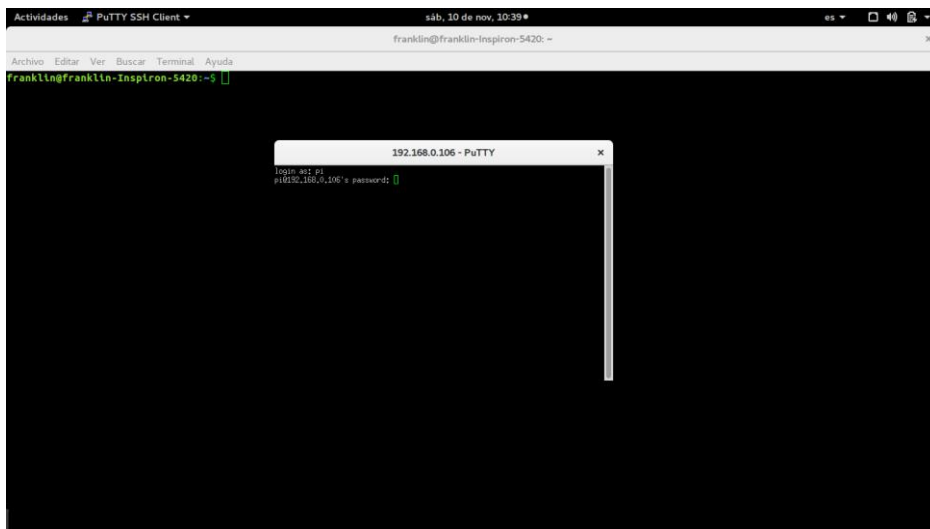
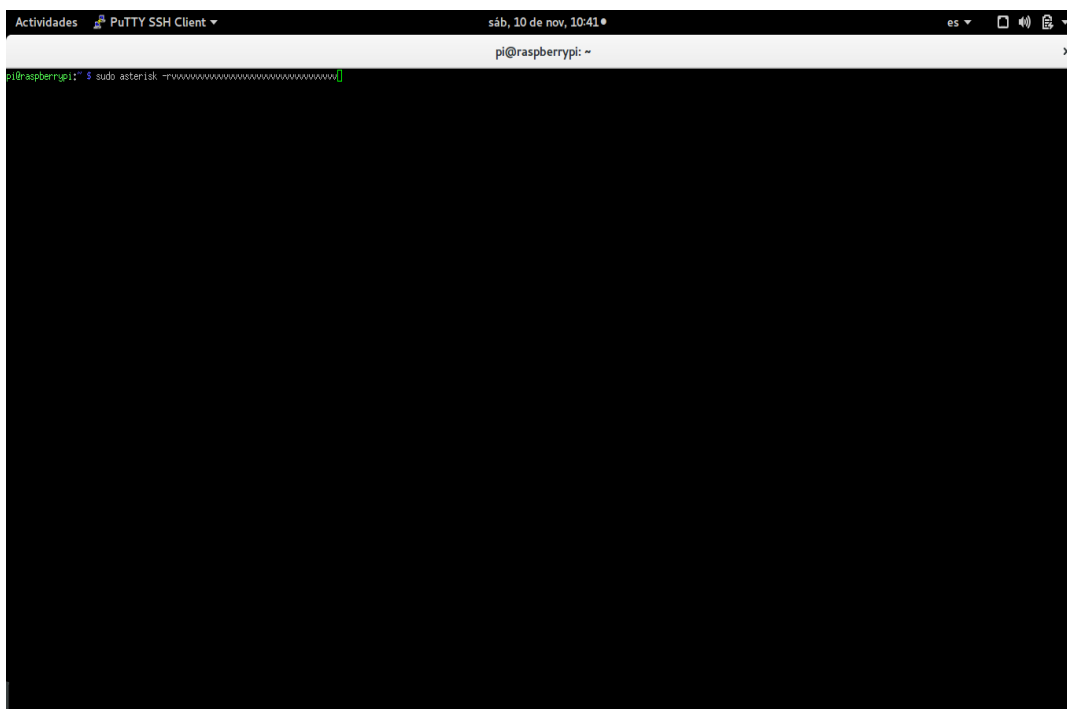


Figura 53 Acceso a servidor implementado en la unidad de procesamiento utilizando usuario y contraseña



A screenshot of a terminal window titled 'Putty SSH Client'. The window shows a terminal session on a Raspberry Pi. The prompt is 'pi@raspberrypi: ~/algoritmos'. The user has entered the command 'sudo systemctl start interfazhduno' and pressed Enter, resulting in a new prompt 'pi@raspberrypi: ~/algoritmos \$'.

Figura 54 Petición de almacenamiento de llave para conexión segura



A screenshot of a terminal window titled 'Putty SSH Client'. The window shows a terminal session on a Raspberry Pi. The prompt is 'pi@raspberrypi: ~'. The user has entered the command 'sudo asterisk -r' and pressed Enter, resulting in a new prompt 'pi@raspberrypi: ~\$'.

Figura 55 Comando de acceso a servicio de Voz IP en modo depuración

```

Actividades PuTTY SSH Client sáb, 10 de nov, 10:41 es
pi@raspberrypi: ~
pi@raspberrypi:~$ sudo asterisk -r
*****
Asterisk 13.22.0, Copyright (C) 1999 - 2014, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
*****
Connected to Asterisk 13.22.0 currently running on raspberrypi (pid = 856)
raspberrypi*CLI>

```

Figura 56 Petición de características pre-configuradas cargadas en el modem GSM

```

Actividades PuTTY SSH Client sáb, 10 de nov, 10:41 es
pi@raspberrypi: ~
pi@raspberrypi:~$ sudo asterisk -r
*****
Asterisk 13.22.0, Copyright (C) 1999 - 2014, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
*****
Connected to Asterisk 13.22.0 currently running on raspberrypi (pid = 856)
raspberrypi*CLI> dongle show devices
dongle0 0 Free 31 0 0 Claro E160 11,609,10,02,432 353871028642418 740010168429115 *59398253789
raspberrypi*CLI>

```

Figura 57 Respuesta del servidor de Voz IP sobre las características asignadas al modem GSM

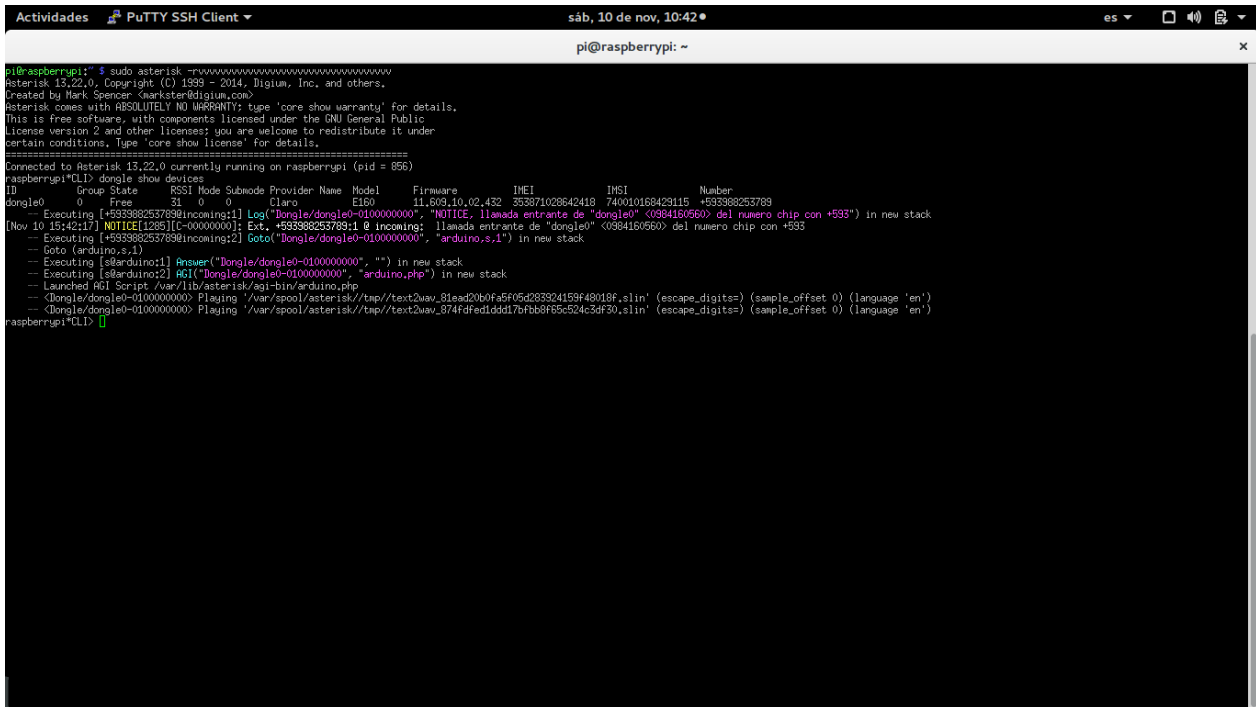


Figura 58 Respuesta del servidor al recibir una llamada telefónica

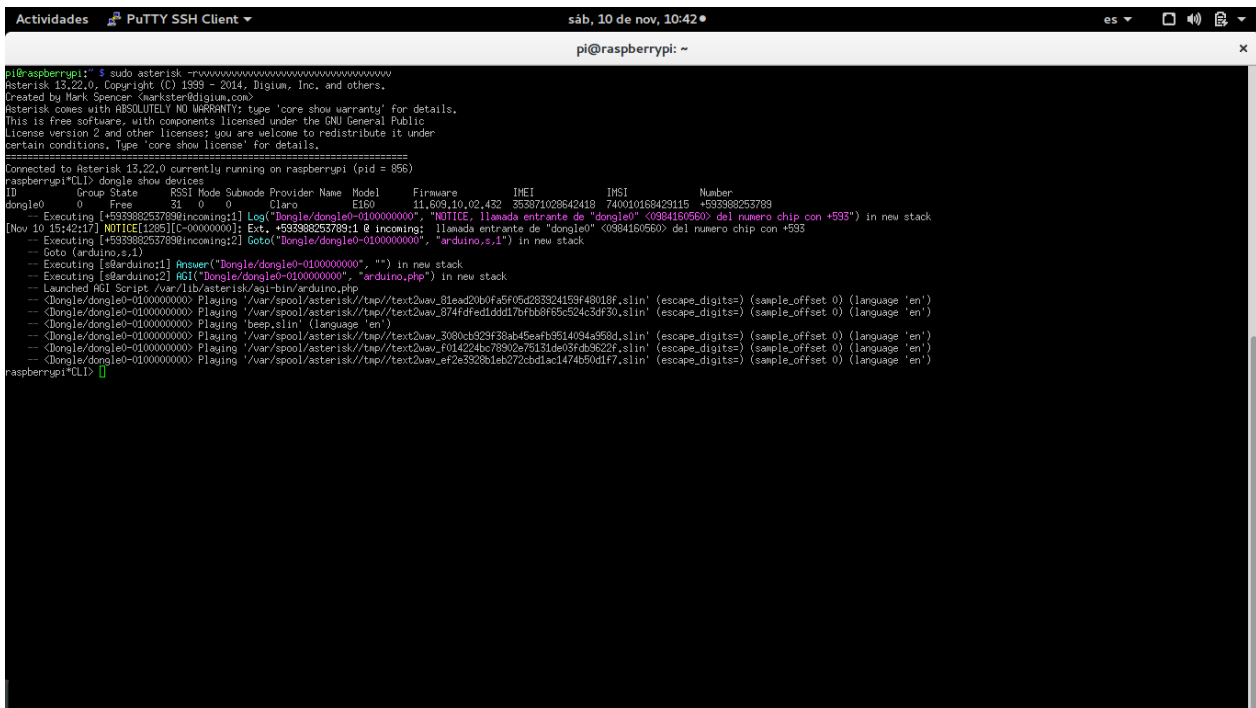


Figura 59 Resultado del redireccionamiento de la llamada hacia el servicio de Voz IP

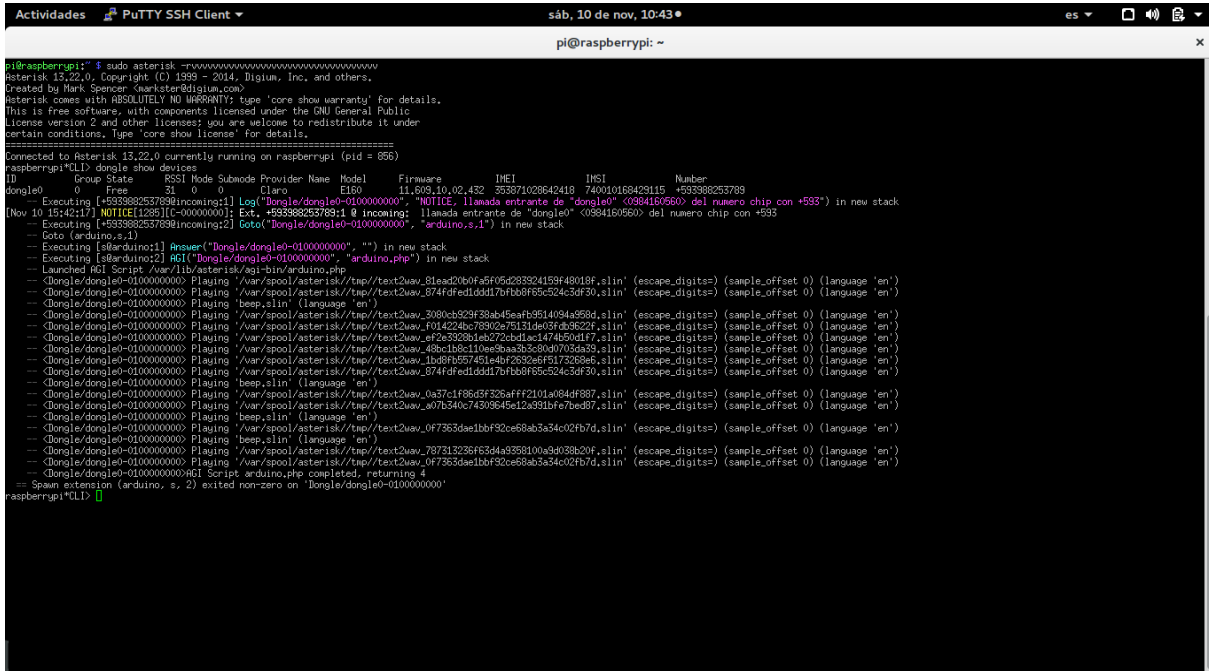


Figura 62 Adquisición de señal de control emitida por el cliente y actualización a través de la petición SQL generada por herramienta PHP-AGI

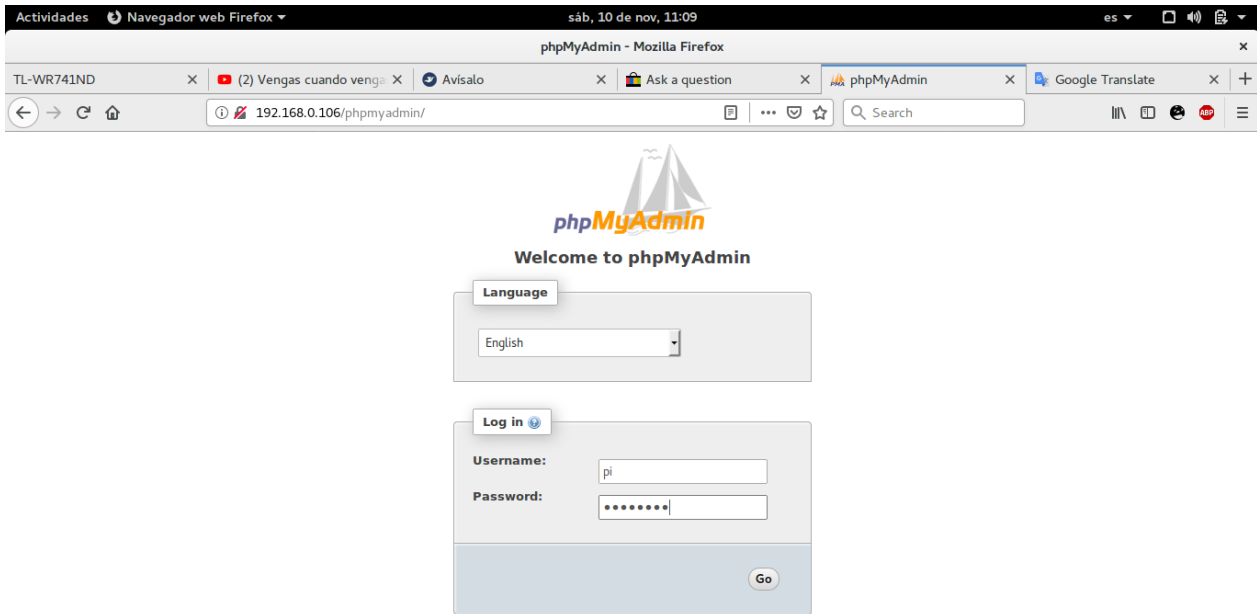


Figura 63 Prueba de acceso a gestor de base de datos

The screenshot shows the phpMyAdmin interface for the 'variables' database, specifically the 'actuaadores' table. The table structure is as follows:

a1	a2	a3	a4	puertaEn	puerta1	puerta2	ventilador	buzzer
1	0	0	0	0	0	0	0	0

The interface includes a navigation sidebar on the left, a top menu with options like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', and 'Operations'. A message at the top states: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' The SQL query shown is 'SELECT * FROM `actuaadores`'. Below the table, there are options for 'Show all', 'Number of rows' (set to 25), and a search filter. At the bottom, there are buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

Figura 64 Actualización de estado de actuadores conectados a luminarias

The screenshot shows the phpMyAdmin interface for the 'variables' database, specifically the 'actuaadores' table. The table structure is as follows:

a1	a2	a3	a4	puertaEn	puerta1	puerta2	ventilador	buzzer
1	0	0	0	1	1	1	0	0

The interface is similar to the previous screenshot, but the values for 'puertaEn', 'puerta1', and 'puerta2' have been updated to 1. The URL in the browser address bar is '192.168.0.106/phpmyadmin/sql.php?db=variables&table=actuaadores&token=6702b3a43c6eb4f1c4f948e12eb1bd39&pos=0'.

Figura 65 Actualización de estado de actuadores conectados a luminarias y puertas

The screenshot shows the phpMyAdmin interface for the 'variables' database, specifically the 'actuadores' table. The table structure is as follows:

a1	a2	a3	a4	puertaEn	puerta1	puerta2	ventilador	buzzer
0	1	1	1	1	0	0	0	0

The interface includes a navigation sidebar on the left, a top toolbar with options like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', and 'Operations'. A message at the top states: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' Below the table, there are options for 'Show all', 'Number of rows' (set to 25), and a search filter. At the bottom, there are buttons for 'Query results operations' (Print, Copy to clipboard, Export, Display chart, Create view) and 'Bookmark this SQL query'.

Figura 66 Actualización de estado de actuadores

The screenshot shows the phpMyAdmin interface for the 'variables' database, specifically the 'actuadores' table. The table structure is as follows:

a1	a2	a3	a4	puertaEn	puerta1	puerta2	ventilador	buzzer
0	0	1	1	0	0	0	0	0

The interface is identical to the previous screenshot, showing the same navigation sidebar, toolbar, and table structure. The query result in the table has been updated, with 'a1' and 'a2' now set to 0, and 'puertaEn' set to 0.

Figura 67 Actualización de estado de actuadores conectados a luminarias

The screenshot shows the phpMyAdmin interface in a Mozilla Firefox browser. The URL is `192.168.0.106/phpmyadmin/sql.php?db=variables&table=actuadores&token=`. The interface displays the 'actuadores' table with the following data:

a1	a2	a3	a4	puertaEn	puerta1	puerta2	ventilador	buzzer
0	0	0	1	0	0	0	0	0

The interface also shows a message: "Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available." and a query result: "Showing rows 0 - 0 (1 total, Query took 0.0019 seconds.)".

Figura 68 Actualización de estado de actuadores conectados a luminarias

The screenshot shows the phpMyAdmin interface in a Mozilla Firefox browser. The URL is `192.168.0.106/phpmyadmin/sql.php?db=variables&table=actuadores&token=`. The interface displays the 'actuadores' table with the following data:

a1	a2	a3	a4	puertaEn	puerta1	puerta2	ventilador	buzzer
0	0	0	0	0	0	0	0	0

The interface also shows a message: "Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available." and a query result: "Showing rows 0 - 0 (1 total, Query took 0.0013 seconds.)".

Figura 69 Actualización en tabla de estados de actuadores con todas las señales de control activados

4.2. Análisis de resultados

Una vez desarrolladas varias pruebas sobre el sistema implementado, ha sido posible comprobar el funcionamiento óptimo. Estableciendo que el sistema planteado, se encuentra en la capacidad de satisfacer las necesidades de una empresa de mediana magnitud.

El uso de dispositivos de fuente abierta ha permitido adaptar y vincular varios trabajos de investigación precursores sin verse limitados por el cubrir licencias de paga o acceder a software de alto costo para implementar un sistema de control y adquisición confiable. Además, se ha demostrado que no se requiere de hardware de altas prestaciones o una inversión alta para competir contra la capacidad de acceso a las nuevas tecnologías con las que hoy en día gozan las grandes empresas.

Permitiendo, también reducir costos en la gestión energética, precautelar la seguridad de empleados, controlar variables de entorno de vital importancia en caso de que la empresa lo requiera e incluso proporcionar una interfaz remota que puede ser controlada a través de una terminal telefónica desde cualquier parte del globo

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El uso de las nuevas tecnologías de comunicación permite extender las capacidades de la industria en la actualidad, dotando de herramientas que posibilitan el acceso remoto a la información y el control a través de interfaces desarrolladas para dispositivos inteligentes de uso diario; no obstante solo las grandes empresas han sido beneficiadas debido a la capacidad adquisitiva que es la principal limitante para que la pequeña o mediana empresa implemente algún tipo de plataforma de monitoreo remoto.

La tecnología de fuente abierta, desde su instauración hasta el año en curso, ha catapultado el desarrollo de decenas de proyectos a nivel mundial a tal punto de llegar a competir e inclusive a superar al software y hardware de uso comercial. Proporcionando libertad a desarrolladores, investigadores y pequeños y medianos emprendedores de incorporar, adaptar y/o mejorar una herramienta de uso general siempre y cuando se conserven los principios bajo los que han sido desarrollados en un principio sin la necesidad de cubrir los costos que representa utilizar herramientas comerciales.

De la implementación del sistema de adquisición y control a través de un entorno de comunicaciones VoIP, se ha concluido que la implementación de sistemas por módulos, disminuye el tiempo de operación, desarrollo, vinculación entre herramientas de software, hardware y detección de errores, debido a que una vez implementado cada módulo independiente, es posible vincular su funcionamiento entre dos o más módulos

e inclusive reutilizarlos corto, mediano o largo plazo e inclusive expandir el sistema dependiendo de los requerimientos que surgen en el trabajo diario y evolución continua de la pequeña o mediana empresa.

La topología de desarrollo adoptada para el proyecto de investigación, se encuentra en la capacidad no solo de satisfacer las demandas que se generen en las PyMES sino también de enfrentar la alta demanda tecnológica de una empresa con requerimientos superiores; no obstante, con el incremento de esta demanda, la cantidad de recursos para el procesamiento, capacidad de gestión de llamadas telefónicas y control de periféricos requerirá una mejora directamente proporcional en aspectos de hardware.

5.2 Recomendaciones

En cuanto a la selección del hardware comunicaciones GSM es recomendable documentar a fondo la compatibilidad entre el módulo de gestión CHAN_DONGLE y los modelos de modems GSM/USB puesto que ciertas versiones demasiado antiguas se encuentran descontinuadas, mientras que las más actuales requerirán de herramientas externas o inversión adicional para liberar la capacidad de transmisión y recepción de voz.

Se recomienda elegir distribuciones de software con soporte garantizado a mediano o largo plazo, debido a que ciertas versiones de software de código abierto se ven alteradas con las actualizaciones constantes o la falta de ellas, que tienden a generar conflictos en funcionamiento o de llegar el caso problemas inseguridad en la información que resultaría perjudicial para la empresa.

Para la vinculación entre el servicio de VoIP con el módulo de acceso a red GSM a través de módems USB, se recomienda compilar e instalar el código fuente de cada una de las herramientas de software; puesto que, además de agregar únicamente las características

que se adapten a las necesidades de la empresa, permite comprobar y corregir errores que suelen presentarse durante una instalación automática y debido a su transparencia hace imposible detectar la causa de los fallos durante la puesta en marcha de los servicios.

La mayoría de las causantes de los fallos durante el redireccionamiento interno de la llamadas entre abonados internos y externos, se presentan por no considerar el contexto con el que han sido creados. Por lo que se recomienda evitar el uso del canal pjsip al compilar cualquier versión del servicio de asterisk, utilizar contextos e identificadores diferente; además de considerar que las relaciones entre extensiones son asignadas manualmente por el administrador del sistema y no de forma automática por el servidor.

REFERENCIAS BIBLIOGRAFICAS

- (s.f.). Obtenido de voip-info: <https://www.voip-info.org/voip-gsm-gateways/admin>. (Octubre de 2017). Obtenido de VozToVoice: <https://www.voztovoice.org/?q=node/2211>
- Ana Lucia Calderon, M. d. (Septiembre de 2006). Diseño de una red VoIP utilizando el sistema operativo linux para su implementacion existente en ecorae. Quito, Pichincha, Ecuador: Escuela Politecnica Nacional.
- Anand Jatti, M. K. (2016). Design and Development of an IOT based. *IEEE International Conference On Recent Trends In Electronics Information Communication Technology*, 1108- 1112.
- Benavides, C. (2018). Obtenido de <http://revistaconstruir.com/como-optimizar-el-manejo-de-energia-de-su-edificio-a-traves-del-iot/>
- Castillo, D. E. (2013). Análisis y resultados sobre el uso de la telefonía ip en las pymes de cartagena, bolívar - colombia. Colombia.
- Clark, R. (2004). Obtenido de <http://www.cstr.ed.ac.uk/projects/festival/>
- Colp, J. C. (Octubre de 2018). Obtenido de <https://wiki.asterisk.org/wiki/pages/viewpage.action?pageId=32375589>
- Davenport, M. (23 de Diciembre de 2014). *Asterisk*. Recuperado el 21 de Noviembre de 2018, de <https://wiki.asterisk.org/wiki/display/AST/Registering+Phones+to+Asterisk>
- Doan, T. T. (2018). *Smart Home with Resilience Against Cloud Disconnection*. Quebec.
- E. Rico, E. (2017). Obtenido de <http://www.ermesh.com/raspberry-pi-3-internet-de-las-cosas/>
- ekos, R. (13 de Septiembre de 2016). Obtenido de Ekos: <http://www.ekosnegocios.com/negocios/verArticuloContenido.aspx?idArt=8091>
- emmeshop. (04 de 05 de 2015). *instructables*. Recuperado el 21 de 11 de 2018, de <https://www.instructables.com/id/Read-and-write-from-serial-port-with-Raspberry-Pi/>
- Enterprise, H.-p. (Mayo de 2017). Obtenido de Hewlet-packard Enterprise: <https://www.hpe.com/us/en/solutions/smb.html>
- Geeky Theory*. (2013). Obtenido de <https://geekytheory.com/arduino-raspberry-pi-lectura-de-datos>
- Goncalves, F. E. (2017). *Como construir y configurar un PBX con software libre Asterisk versión 1.4*. Janeiro.

- Hargreaves, J. (2018). *Security Awareness and Risks in IoT Devices: Gathering Information and Perspectives from University Students in South Wales, Cardiff*. Cardiff.
- Innovative tech pros.* (2014). Obtenido de <http://itpscorp.com/blog/working-with-asterisk-agi-and-php-3/>
- ITUSER. (21 de Noviembre de 2016). Obtenido de Tecnología para tu empresa: <https://tecnologiaparatuempresa.ituser.es/transformacion-digital/2016/11/aprovecha-los-beneficios-de-iot-para-tu-negocio>
- ITUSER. (20 de Enero de 2017). Obtenido de <https://tecnologiaparatuempresa.ituser.es/actualidad/2017/01/pymes-de-todas-las-industrias-pueden-aprovechar-internet-de-las-cosas>
- Laila Salman, S. S. (09 de 02 de 2017). *IEEE XPLORE*. Recuperado el 20 de 09 de 2018, de <https://ieeexplore.ieee.org/document/7845449/>
- Learn Python.* (s.f.). Obtenido de https://www.tutorialspoint.com/python/python_database_access.htm
- Lenin, R. D. (2017). Implementacion de software para la administracion de ancho de banda y acceso a contenidos en una pyme de Guayaquil: Universidad de Guayaquil.
- Liu, T. (s.f.). Obtenido de <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- Martinovic, A. E., & Etchart, W. D. (2018). *Sistema de control aplicado a comunas rurales aisladas y su vinculación con Internet de las cosas (IoT)*. Argentina.
- My Technology | Republic.* (03 de Diciembre de 2015). Recuperado el 21 de noviembre de 2018, de http://mytechrepublic.com/asterisk-11-usb-gsm-trunk-and-gateway-using-chan_dongle/
- My Technology | Republic.* (03 de Diciembre de 2015). Obtenido de http://mytechrepublic.com/asterisk-11-usb-gsm-trunk-and-gateway-using-chan_dongle/
- Negocio, D. (2017). Obtenido de <http://destinonegocio.com/emprendimiento/el-iot-debe-importarle-a-las-pymes/>
- Newton, R. (06 de Agosto de 2014). Obtenido de <https://wiki.asterisk.org/wiki/display/AST/Asterisk+Architecture%2C+The+Big+Picture>
- Newton, R. (23 de Abril de 2014). Obtenido de <https://wiki.asterisk.org/wiki/display/AST/Creating+SIP+Accounts>

- Newton, R. (23 de Diciembre de 2014). Obtenido de <https://wiki.asterisk.org/wiki/display/AST/Registering+Phones+to+Asterisk>
- Newton, R. (23 de Diciembre de 2014). *Asterisk*. Recuperado el 21 de Noviembre de 2018, de <https://wiki.asterisk.org/wiki/display/AST/Creating+Dialplan+Extensions>
- P. J. Rani, J. B. (2017). Voice controlled home automation system using Natural Language Processing (NLP) and Internet of Things (IoT). *hird International Conference on Science Technology Engineering & Management (ICONSTEM)*, 368-373.
- Pablo Palacios, A. C. (07 de 06 de 2018). *IEEE XPLORE*. Recuperado el 20 de 09 de 2018, de <https://ieeexplore.ieee.org/document/8372357/>
- Piyare, R. (11 de 02 de 2013). *International Journal of Internet of Things*. Recuperado el 22 de 09 de 2018, de <http://article.sapub.org/10.5923.j.ijit.20130201.02.html#Sec5>
- Pratibha D. Lagad, M. S. (2018). Embedded based Industrial Parameters Monitoring and Controlling System using Internet of Things. *International Journal of Advanced Research in Science, Engineering and Technology*, V(8), 1-7.
- Richie. (07 de Marzo de 2017). Obtenido de <https://pchelp.ricmedia.com/setup-lamp-server-raspberry-pi-3-complete-diy-guide/3/>
- Richie. (07 de Marzo de 2017). *Ricmedia Pc Help*. Recuperado el 21 de Noviembre de 2018, de <https://pchelp.ricmedia.com/setup-lamp-server-raspberry-pi-3-complete-diy-guide/3/>
- Santana, G. y. (Enero de 2018). */ri.uaemex.mx*. (Dr. Luis Enrique Diaz Sanchez) Recuperado el 15 de Agosto de 2018, de <http://ri.uaemex.mx/bitstream/handle/20.500.11799/68484/Reconocimiento%20por%20Biometria%20Facial%20para%20Aplicaciones%20en%20Ciudades%20Inteligentes.pdf?sequence=1>
- Santana, M. A. (2018). Reconocimiento Por Biometria Facial Para Aplicaciones En Ciudades Inteligentes. Mexico: Universidad Autonoma del Estado de Mexico.
- sparkfun*. (2003). Obtenido de <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-6.pdf>
- Vivanco, J. (Septiembre de 2017). Obtenido de http://repo.uta.edu.ec/bitstream/123456789/26367/1/Tesis_%20t1308ec.pdf
- Waldo Valiente, E., Carnuccio, M., Volker, G., De Luca, G., García, D., Giulianelli, R., & Villca, M. V. (2018). Dispositivo de asistencia de personas mediante monitoreo IoT. *XX Workshop*

de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste)., 5.

Zambrano, J. H. (2014). “Fundamentación de factibilidad y conveniencia en el diseño de una propuesta de un Sistema de comunicación, basada en una solución tecnológica Open Source para la Empresa Pública Municipal de Agua Potable y Alcantarillado de Portoviejo. Guayaquil: Universidad Católica Santiago De Guayaquil.