



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

DESARROLLO E IMPLEMENTACIÓN DE INTERNET INDUSTRIAL DE LAS COSAS APLICADO AL LABORATORIO DE PLC'S DE LA FACULTAD DE INGENIERÍA INDUSTRIAL EN PROCESOS DE AUTOMATIZACIÓN DE LA UNIVERSIDAD TÉCNICA DE AMBATO.

TANNYA LUCIA GALLO CASTILLO

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

**MAGISTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN
INDUSTRIAL**

Riobamba - Ecuador

Octubre – 2018



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad **Proyectos de Investigación y Desarrollo**, titulado: “DESARROLLO E IMPLEMENTACIÓN DE INTERNET INDUSTRIAL DE LAS COSAS APLICADO AL LABORATORIO DE PLC’S DE LA FACULTAD DE INGENIERÍA INDUSTRIAL EN PROCESOS DE AUTOMATIZACIÓN DE LA UNIVERSIDAD TÉCNICA DE AMBATO”, de responsabilidad de la Ing. Tannya Lucia Gallo Castillo ha sido prolijamente revisado y se autoriza su presentación.

Tribunal:

ING. OSWALDO MARTÍNEZ GUASHIMA, M.Sc.

PRESIDENTE (DELEGADO)

FIRMA

ING. FERNANDO SAÁ TAPIA; M.Sc.

TUTOR

FIRMA

ING. JULIO CAJAMARCA VILLA; M.Sc.

MIEMBRO

FIRMA

ING. DAVID RIVAS LALALEO; M.Sc.

MIEMBRO

FIRMA

Riobamba, Octubre - 2018

DERECHOS INTELECTUALES

Yo, Tannya Lucia Gallo Castillo, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en el **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

TANNYA LUCIA GALLO CASTILLO

N° Cédula: 1804154233

DEDICATORIA

A mi esposo por su devoción y entrega, brindándome el apoyo, comprensión en el momento oportuno le dedico este trabajo, manifestando así que mis éxitos son los suyos.

A mi hermana Daniela que es la luz de mi vida, a ti con mucho cariño te dedico mi proyecto demostrándote así que las metas se cumplen, se alcanzan con dedicación, esfuerzo y la bendición de Dios siempre.

Lucia Gallo Castillo

AGRADECIMIENTO

Agradezco al todopoderoso Dios y a la Virgen María por ser el cimiento firme de un interminable cumplimiento de logros y metas, a mi esposo David por ser quien logra hacer de mí una persona mejor cada día con su amor y apoyo incondicional, a mis padres Luis y Yolanda, hermanos Daniela, Félix y Norma, mis sobrinas Angélica, Amanda y a mis suegros Rene y Silvia por sus palabras de aliento y buenos deseos.

Un agradecimiento de todo corazón al Ing. Fernando Saá, al Ing. Julio Cajamarca y al Ing. David Rivas por su activa participación en el proyecto.

Lucia Gallo Castillo

TABLA DE CONTENIDO

RESUMEN.....	x
ABSTRACT.....	xi
CAPÍTULO I	
1. INTRODUCCIÓN	1
1.1. Planteamiento del problema.....	1
1.2. Justificación	2
1.3. Objetivos	3
1.3.1. <i>Objetivo General</i>	3
1.3.2. <i>Objetivos Específicos</i>	3
1.4. Hipótesis.....	4
CAPÍTULO II	
2. MARCO DE REFERENCIA	5
2.1. Introducción.....	5
2.2. Estudio del Arte.....	6
2.3. Fundamentación Teórica	7
2.3.1. <i>Internet of things (IoT)</i>	7
2.3.2. <i>Ventajas del Internet de las cosas</i>	10
2.3.3. <i>Internet Industrial de las cosas(IIoT) y la Industria 4.0</i>	11
2.3.4. <i>Máquina a Máquina (M2M)</i>	12
2.3.5. <i>Protocolo de comunicación M2M</i>	13
2.3.6. <i>Message Queuing Telemetry Transport (MQTT)</i>	15
2.3.7. <i>Aplicaciones de la comunicación M2M para aplicaciones industriales</i>	16
2.3.8. <i>Raspberry Pi</i>	17
CAPÍTULO III	
3. METODOLOGÍA INVESTIGATIVA.....	19
3.1. Tipos de Investigación.....	19
3.2. Recolección de Información	19
3.3. Procesamiento y Análisis de datos	20
3.4. Desarrollo del Proyecto.....	22

CAPÍTULO IV

4. PRUEBAS Y RESULTADOS	23
4.1. Verificación de los Hipótesis.....	29

CAPÍTULO V

5. PROPUESTA DE SOLUCIÓN	31
5.1. Introducción.....	31
5.2. Propuesta de Solución.....	31
5.2.1. <i>Requerimientos de Hardware</i>	32
5.2.2. <i>Requerimientos de Software</i>	32
5.3. Configuración del Plc Siemens 1200.....	33
5.4. Programación en Python	34
5.5. Software de prueba Mosquito para MQTT	37
5.6. Base de datos en PhpMyAdmin	39
5.7. Página Web en PHP	42

CONCLUSIONES.....	43
--------------------------	-----------

RECOMENDACIONES.....	44
-----------------------------	-----------

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-2: Comparación de la Industria 4.0 e IIoT.....	12
Tabla 2-2: Especificaciones técnicas de la Raspberry Pi.....	18
Tabla 1-3: Aspectos específicos.....	20
Tabla 2-3: Aspectos Generales	21
Tabla 1-5: Requerimientos de Hardware	32
Tabla 2-5: Requerimientos de Software.....	33

ÍNDICE DE FIGURAS

Figura 1-2: Arquitectura de M2M	14
Figura 2-1: Arquitectura de red de M2M para aplicaciones industriales.....	17
Figura 3-1: Dispositivo Raspberry Pi	17
Figura 1-4: Página Web de los usuarios para acceder al sistema.....	23
Figura 2-4: Datos para acceder al sistema.	23
Figura 3-4: Datos analógicos de la salida del PLC.....	24
Figura 4-4: Datos digitales de las salidas del PLC.	24
Figura 5-4: Configuración de datos de temperatura adquiridos con el PLC.....	25
Figura 6-4: Configuraciones de conexión MQTT	26
Figura 7-4: Métricas y datos históricos de la tasa de mensajerías en ejecución.	26
Figura 8-4: Lista de paquetes capturados utilizando el programa Wireshark.....	27
Figura 9-4: Datos estadísticos de los paquetes enviados utilizando el protocolo MQTT	28
Figura 10-4: Comparación de length vs time de paquetes enviados con MQTT	28
Figura 11-4: Latencia de los diferentes servidores por un medio físico de fibra y ADSL.....	29
Figura 12-4: Latencia medida en América, España y Europa.....	30
Figura 1-5: Diagrama de bloques.....	32
Figura 2-5: Dirección Ip del Plc Siemens 1200.....	33
Figura 3-5: Encapsulación del protocolo S7.....	35
Figura 4-5: Código en Python utilizando el protocolo S7	37
Figura 5-5: Probando MQTT con Mosquitto.....	39
Figura 6-5: Base de datos para almacenar las salidas del PLC.....	41
Figura 7-5: Base de datos para almacenar las entradas del PLC	41
Figura 8-5: Base de datos de los usuarios para acceder al sistema.	42

RESUMEN

El objetivo fue el desarrollo e implementación del Internet Industrial de las cosas (IIoT) aplicado al laboratorio de PLC's de la Facultad de Ingeniería Industrial en Procesos de Automatización de la Universidad Técnica de Ambato, nace por la necesidad del acceso a la información, de conectar dispositivos industriales al internet y las nuevas tecnologías que introducen las soluciones para mejorar la calidad de los procesos esta propuesta alcanzó la implementación del IIoT aplicado al laboratorio de procesos. El proyecto consistió en realizar una página web donde se monitorea los datos y estados de las salidas y entradas del PLC Siemens 1200, se realizó una comunicación con la raspberry pi que contiene todos los softwares utilizados para acceder a la información requerida del PLC, para el manejo de datos se utilizó uno de los protocolo del IIoT denominado Message Queuing Telemetry Transport (MQTT) el cual permite enviar y almacenar en una base de datos creada en PhpMyAdmin, la interfaz de presentación se realizó en Hypertext Preprocessor (PHP), que permite crear páginas web para el monitoreo remoto de datos en cualquier dispositivo inteligente que pueda acceder a internet, el sistema mostró una latencia del 0,6 milisegundos concluyendo de esta manera que la obtención y monitorización de datos es accesible y se aplica a procesos de automatización industrial de manera segura, confiable y con calidad de servicio. Se recomienda utilizar software abiertos para el desarrollo de la aplicación.

Palabras Claves: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <AUTOMATIZACIÓN INDUSTRIAL>, <PYTHON (SOFTWARE)>, < MESSAGE QUEUING TELEMETRY TRANSPORT (MQTT)>, <INTERNET INDUSTRIAL DE LAS COSAS (IIoT)>, <RASPBERRY (SOFTWARE - HARDWARE)>, <PHP (SOFTWARE)>.

ABSTRACT

The objective was the development and implementation of the Industrial Internet of Things (IIoT) applied to the PLC laboratory of the Faculty of Industrial Engineering in Automation Processes of the Technical University of Ambato, born from the need of access to information, of connecting industrial devices to the internet and the new technologies that introduce the solutions to improve the quality of the processes. This proposal reached the implementation of the IIoT applied to the process laboratory. The project consisted of making a web page where the data and states of the outputs and inputs of the Siemens 1200 PLC, a communication was made with the raspberry pi that contains all the softwares used to access the required information of the PLC, for the handling of data one of the IIoT protocol called Message Queuing Telemetry Transport (MQTT) which allows to send and store in a database created in PhpMyAdmin, the presentation interface was made in Hypertext Preprocessor (PHP), which allows to create web pages for remote data monitoring in any intelligent device that can access the internet, the system showed a latency of 0,6 milliseconds concluding in this way that obtaining and data monitoring is accessible and applied to industrial automation processes in a safe, reliable and quality of service manner. It is recommended to use open software for the development of the application.

Key words: < TECHNOLOGY AND SCIENCE OF ENGINEERING>, <INDUSTRIAL AUTOMATION>, <PYTHON (SOFTWARE)>, < MESSAGE QUEUING TELEMETRY TRANSPORT (MQTT)>, <INDUSTRIAL INTERNET OF THINGS (IIoT)>, <RASPBERRY (SOFTWARE - HARDWARE)>, <PHP (SOFTWARE)>.

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Planteamiento del problema

Las industrias de la próxima generación requieren mejoras, a fin de maximizar la rentabilidad: más flexibilidad en sus diseños, más información sobre el proceso y los productos manufacturados, más inteligencia en el procesamiento de datos adquiridos y una integración efectiva de la experiencia humano/máquina. Sin embargo, a medida que la nueva tecnología se introduce en el sector industrial, algunas reglas deben ser respetadas. La primera y más importante es que la producción no puede detenerse; las nuevas tecnologías deben ser compatibles con los sistemas antiguos, y la interoperabilidad entre los proveedores debe ser facilitado. Además, las soluciones deben ofrecer un medio para apoyar la operación en tiempo real y la ciberseguridad.

Hoy en día el difundido acceso a los datos y la integración eficiente de los datos distribuidos entre software y hardware se convierte en un problema común para las universidades y la industria. Esta investigación pone de relieve las principales preocupaciones en la construcción de un sistema distribuido de manera sistemática, proporciona un mecanismo de comunicación M2M (máquina a máquina) orientado al almacenamiento y monitoreo de los datos con una arquitectura de referencia del sistema industrial y da una prueba de concepto de aplicación y evaluación experimental.

La incorporación de tecnologías de la información suscitadas en Latinoamérica se da en la rama de la producción agrícola a través del IoT (Internet de las cosas) con un impacto importante, tanto económico como medio ambiental particularmente citadas en redes de sensores inalámbricos un ejemplo palpable de la aplicación de esta tecnología se da en la agricultura en Uruguay quienes desarrollaron un proyecto denominado “Monitoreo microclimático en plantaciones cítricas para alertas de helada y riego de precisión, y monitoreo de plagas mediante la adquisición de imágenes” el cual describe las plataformas de hardware y software utilizando el concepto del IIoT(SASE, 2015).

En Ecuador muchas de las empresas de producción y servicios petroleros tienen presencia de monitoreo remoto en sus líneas de producción, por lo tanto, la presente investigación busca

correlacionar todos estos factores innovadores para aplicaciones IIoT en el ya nombrado sistema industrial.

1.2. Justificación

El Instituto Fraunhofer de Ingeniería y automatización industrial en Alemania proyecta que la industria 4.0 y su concepto del IIoT puede provocar un salto en la productividad del 20 % al 30% para el año 2025. Sin embargo, el sector industrial necesita cambios progresivos y amigables con tecnologías y soluciones.

El IIoT es una tecnología que combina técnicas industriales como lo son sensores inteligentes y dispositivos inteligentes en una red. Esta tecnología permite memorizar, analizar e integrar los datos de todos los sensores inteligentes a un único espacio de trabajo.

El IIoT tendrán un gran impacto en la economía de las industrias individuales y de todos los países que lo adopten. Además, aumentará la productividad y el crecimiento del producto nacional bruto, un impacto positivo sobre las condiciones de trabajo y el crecimiento profesional de los empleados. El modelo de servicio de economía, que se ha creado en el proceso de esta transición, a partir de la digitalización de la producción, el intercambio de datos entre los distintos asociados y análisis de grandes volúmenes de datos.

La industria 4.0 y el IIoT ayuda a mejorar la eficiencia de producción y abastecimiento mediante la automatización y el intercambio de datos entre los sistemas de fabricación y logística, conectar dispositivos industriales con tecnologías de información y multiplataformas, significa dar un gran paso en la Industria 4.0 y avanzar así hacia el denominado IIoT logrando uno de los objetivos de las smart factories que es la interoperabilidad. (Semle, 2016)

El proyecto basa su desarrollo en diseñar e implementar una aplicación del IIoT en un sistema de automatización industrial que permitan el monitoreo del sistema de forma alámbrica e inalámbrica, basándose en un diseño simple, genérico y extensible para la fácil identificación e integración de los equipos en procesos de automatización.

Además, los componentes que posee el proyecto corresponden a sistemas embebidos para los elementos dedicados al monitoreo del proceso, o computadores de placa reducida para alojar servidores que registren y permitan la interacción con el usuario. Debido al tipo de componentes que integra el sistema permite reducir costos relacionados al hardware y software debido a su

masividad. Además, esto permite que tengan un mayor desarrollo y soporte, debido a múltiples usuarios que ocupan estos componentes.

El sistema propuesto está enfocado en plantas de pequeña escala o para laboratorios docentes o de investigación, ya que el costo de instalación y operación de sistemas de control convencionales son muy elevados haciendo aceptable y fiable este tipo de tecnología.

Actualmente existen estudios que permiten innovar el IIoT mas no se conoce hasta el momento trabajos específicos relacionados al tema propuesto, la transmisión de datos desde dispositivos industriales como son los controladores lógicos programables hacia la nube no ha sido desarrollado completamente, es por ello que este tipo de investigación se convierte en un aporte importante a los objetivos que persigue la nueva era industrial, denominada la Industria 4.0 beneficiando a todos aquellos usuarios inmersos en procesos industriales que buscan elevar sus estándares de calidad mediante la aplicación del IIoT.

1.3. Objetivos

1.3.1. *Objetivo General*

Desarrollar e implementar el Internet Industrial de las Cosas aplicado al laboratorio de PLC's de la facultad de Ingeniería Industrial en Procesos de Automatización de la Universidad Técnica de Ambato.

1.3.2. *Objetivos Específicos*

- Desarrollar una aplicación que permita monitorear las salidas del sistema, basándose en un diseño genérico y extensible para la identificación de datos y estados que posee el elemento de control inmerso en un proceso de automatización industrial.
- Implementar la aplicación de IIoT en un proceso de automatización industrial con la finalidad de integrar el concepto de Industria 4.0 en un sistema genérico que utiliza protocolos del IIoT.
- Evaluar experimentalmente la aplicación del IIoT con un dispositivo PLC, para realizar un monitoreo remoto desde una multiplataforma utilizando un computador de placa reducida.

1.4. Hipótesis General

Una posible solución a los problemas planteados y que permita resolver la inmersión del IIoT en procesos de automatización industrial, surge como propuesta el desarrollo e implementación del IIoT aplicado al laboratorio de PLC's de la facultad de Ingeniería Industrial en Procesos de Automatización de la Universidad Técnica de Ambato, lo que lleva a este trabajo a plantear la siguiente hipótesis:

¿El desarrollo del IIoT para integración de datos a nivel de procesos industriales basado en el protocolo MQTT permite tener tiempo óptimo de envío con respecto a la longitud de paquetes.?

CAPÍTULO II

2. MARCO DE REFERENCIA

Este capítulo trata acerca de la contextualización de los temas y subtemas que abarca el proyecto.

2.1. Introducción

Mediante que el IIoT evoluciona, cada vez más dispositivos de adquisición de datos están integrados dentro de las fábricas, silos, instalaciones de agricultura, instalaciones industriales, automóviles y muchas otras aplicaciones.

La aceptación generalizada de IIoT ha cambiado el modo manual de recopilación de datos utilizados en tecnología de información tradicionales. En particular, IIoT puede registrar diversos parámetros del proceso de producción de manera automática, precisa y oportuna. Sólo la producción industrial tradicional se percata de la comunicación de la máquina a través de la tecnología M2M, pero puede lograr conexiones perfectas entre las personas, las máquinas y los objetos físicos. Sin embargo, en el entorno de, la función y el rendimiento de los dispositivos de comunicación son muy diferentes. Por ejemplo, algunas aplicaciones requieren un alto rendimiento en tiempo real, mientras que otras no, mientras que algunas de las tareas de la aplicación se realizan periódicamente y otros son desencadenados por eventos. Estas características aumentan la dificultad de la aplicación práctica de IIoT.

La comunicación máquina-a-máquina es una tecnología clave para las futuras aplicaciones del IIoT. Desempeña un papel importante en la conectividad y la integración de máquinas computadorizadas, tales como sensores, actuadores, controladores y robots. Las exigencias en cuanto a flexibilidad, eficiencia y compatibilidad multiplataforma de la intercomunicación entre el módulo de máquinas conectadas plantean retos para el M2M.

El desarrollo e implementación de los sistemas que componen el IIoT podría representar una enorme inversión no sólo en el presente sino también para las próximas décadas debido a intercambiar una antigua tecnología por una actual y a la vanguardia demandara recursos infinitos. Para satisfacer las necesidades de hoy y del mañana, una de las maneras mediante la implementación de una red de sistemas lo suficientemente flexible como para evolucionar y adaptarse. Esto requiere un enfoque basado en una plataforma de arquitectura con hardware

único que se implementa a través de múltiples aplicaciones, lo suficientemente flexible como para extraer una cantidad sustancial de la complejidad del hardware. Las plataformas que son escogidos por los diseñadores del sistema deben estar basados en un sistema operativo amigable para que puedan ser bien abastecido y correctamente configurado, para autenticar y autorizar a los usuarios para mantener la integridad del sistema y maximizar la disponibilidad del sistema.

2.2. Estudio del Arte

La principal aplicación de IIoT se tiene en las cadenas de suministro donde se utiliza la tecnología de identificación por radiofrecuencia (RFID) con mayor énfasis. Por eso mismo IoT tiene un rol tan importante en las industrias logísticas. En estas se utiliza IoT para monitorizar el movimiento de objetos desde su origen hasta su destino a través de la cadena de suministro para identificar productos y obtener información acerca de su almacenamiento, sin embargo, al incluir el uso de sensores se podría obtener más datos del entorno, como la localización y condiciones del ambiente (Gómez, 2017).

Existen dos enfoques para implementar IIoT en la comunidad mundial: centrada e integrada. El enfoque centrado en el desarrollo consiste en un único vector de la fijación de metas y el enfoque integrado se centra en la investigación de aplicaciones IIoT. Sin embargo, ambas implementaciones resuelven problemas comunes como el desarrollo de tecnologías básicas dentro de la visión de la industria 4.0.(Meng, Wu, Member, Muvianto, & Gray, 2016)

La industria 4.0 está conectada al internet para el intercambio de información, la supervisión de la producción y la gestión de la industria. Es el cuarto tipo de revolución industrial donde los componentes de una industria se comunican unos con otros para lograr objetivos como el de aumentar la eficiencia, la producción y obtener beneficios.(Wan et al., 2016)

El escenario final para la Industria 4.0 puede ser imaginado como fábricas donde la producción está completamente automatizada y controlada por la electrónica inteligente que funciona de manera autónoma e incluso puede reconfigurar a sí mismos, a fin de adaptarse al entorno o a cualquier cambio o anomalía basándose en fiabilidad, definida como la capacidad de ofrecer un servicio que puede legítimamente ser de confianza.(Rong, Vanan, & Phillips, 2016).

Para los trabajos futuros se puede mencionar los protocolos de comunicación del IIoT que todavía no cuentan con niveles óptimos de encriptación que permitan guardar la confidencialidad de los datos, la heterogeneidad de los sistemas los hace vulnerables y la

calidad de conexión reduce la fiabilidad de los mismos. Es necesario establecer protocolos de seguridad y protección de acceso a la información.

2.3. Fundamentación Teórica

2.3.1. *IoT (Internet de las cosas)*

El IoT se trata de una red que interconecta objetos físicos valiéndose del internet. Los mencionados objetos se valen de sistemas embebidos, o lo que es lo mismo, hardware especializado que le permite no solo la conectividad a internet, sino que además programa eventos específicos en función de las tareas que le sean dictadas remotamente.

El IoT potencia objetos que antiguamente se conectaban mediante circuito cerrado, como comunicadores, cámaras, sensores, y demás permitiéndoles comunicarse globalmente mediante el uso de la red de redes.

No hay un tipo específico de objetos conectados a IoT. En lugar de eso se les puede clasificar como objetos que funcionan como sensores y objetos que realizan acciones activas.

En cualquier caso, el principio es el mismo y la clave es la operación remota. Cada uno de los objetos conectados a internet tiene una IP específica y mediante esa dirección se puede acceder para recibir instrucciones. Así mismo, puede contactar con un servidor externo y enviar los datos que recepte (Torres, 2014).

Áreas de Aplicación

El IoT puede aplicarse a varios ámbitos pues la cantidad de usos que pueda darse a los productos y servicios depende de la creatividad e ingenio de los desarrolladores. La factibilidad de implementar hoy en día los ejemplos a ser mencionados a continuación en las distintas áreas puede ser mayor o menor pero todos ellos constituyen formas en las que eventualmente se puede usar esta plataforma (Figuerola, 2014).

Industria y Comercio

En lo que se refiere a la producción y distribución de bienes y servicios el IoT puede utilizarse para:

Optimizar la cadena de producción: Sensores ubicados en las distintas máquinas involucradas en el proceso de producción, pueden informar si se requiere alguna revisión y ocurre alguna falla. Esto permite realizar un proceso de mantenimiento más efectivo lo cual reduce costos y permite un nivel mayor de automatización al disminuir la intervención de actividad humana.

Facilitar el proceso de inventario: Si cada objeto en stock cuenta con un mecanismo que le permite conectarse a la red pueden realizarse controles más precisos sobre la cantidad, los detalles y la ubicación exacta de cada uno de ellos dentro de cada almacén y/o tienda de manera más automatizada y actualizada reduciendo el margen de errores humanos que suelen ocurrir en esta tarea. Esto puede repercutir favorablemente en los tiempos de búsqueda y aprovechamiento del espacio físico.

Mejorar los mecanismos de distribución y abastecimiento: El control sobre los mecanismos de producción e inventario a su vez posibilita hacer más eficiente el sistema de abastecimiento permitiendo elaborar pedidos precisos y en tiempos convenientes. La distribución de los productos a la vez puede coordinarse con ayuda de la conexión de los distintos productos o cargamentos que permitan operaciones de logística mejor adecuadas.

Campos de aplicación de internet de las cosas

Las aplicaciones de IoT actualmente se enfocan en las Smart Home, la industria 4.0, llamada IIoT, los municipios o poblados agrupados en el concepto Smart Town (pueblos inteligentes), las ciudades también llamadas Smart Cities (ciudades inteligentes) y el medio ambiente que se sitúa bajo el concepto Smart Earth (planeta inteligente). Cada una de estas divisiones tiene sus propias aplicaciones, aunque todas se basen en IoT. Las Smart home (hogares inteligentes) buscan priorizar la automatización, la industria mejorar el proceso industrial, los pueblos el mantener su identidad cultural, las ciudades su habitabilidad y el medioambiente la comunicación con los edificios y la naturaleza. Con base a esto, la finalidad es crear un mundo inteligente, conducente al Smart World (mundo inteligente). El dominio de las áreas de aplicación para el IoT está limitado sólo por el tiempo y la imaginación.

El interés en IoT sigue creciendo en diferentes tipos de industria como es en la agricultura, la alimenticia, la farmacéutica y la minera, entre muchas otras. En estos casos, la tecnología que más ha sido utilizada es la RFID, empleada para mejorar la eficiencia de las cadenas de suministro y mejorar la información obtenida durante todo el proceso en tiempo real. Existe gran cantidad de aplicaciones en el IIoT, ejemplo de ello un sistema para poder localizar el

transporte y monitorizarlo, permitiendo predecir su localización futura y el tráfico que tendrá en la carretera.

Con base en el desarrollo de la tecnología IoT para aplicaciones industriales, ha llamado la atención la implementación de sensores y las ventajas de la inteligencia de detección, con las cuales se ha logrado un eficiente monitoreo y control en la reducción de costos y el consumo de energía de las industrias de producción de bienes y/o servicio. De la misma manera con la ayuda de sensores y dispositivos inalámbricos incorporados en máquinas y sistemas industriales, el mantenimiento de estas máquinas y sistemas es controlable y automatizable(Gómez, 2017).

Tecnología

Varios de los sensores necesarios para los objetos del IoT ya existen actualmente: sensores de luminosidad, humedad, contaminación, signos vitales, movimiento, cámaras, sistema de posicionamiento global (GPS), etc. Sin embargo, los desarrollos tecnológicos aún no se detienen en varias de estas áreas con el fin de mejorar su potencia y su precisión, reducir su tamaño y coste, etc. Las plataformas de software y hardware deben permitir la extensión y actualizaciones preferentemente remotas. De este modo sus funcionalidades o características no quedan limitadas y pueden adaptarse a nuevas necesidades y condiciones. Un aspecto importante en este apartado es la seguridad: si se descubren vulnerabilidades o se desarrollan alternativas más eficientes de mecanismos de seguridad estos deben poder implementarse. Los objetos no deben correr el riesgo de quedar obsoletos por no admitir actualizaciones (Figuerola, 2014).

Red y Datos

Las estimaciones sobre la cantidad de objetos conectados al IoT supera ampliamente a los objetos actualmente conectados a la red. La empresa de investigación tecnológica Gartner estima 26 mil millones de objetos van a ser conectados para el año 2020. ABI Research pone esa cifra a 30 mil millones mientras que Cisco estima 50 mil millones. Incluso la menor de estas cifras supone varios retos relacionados con la infraestructura del internet actual:

Implementación de IPV6 (Protocolo de Internet versión 6). Como es bien sabido uno de los principales problemas con el internet hoy en día es que las direcciones del actual protocolo IPV4(Protocolo de Internet versión 4) se agotaron. Si bien este problema se ha “solucionado” en el internet actual con la implementación de redes privadas, esta no es una solución conveniente para el IoT, IPV6 facilita la administración gracias a sus capacidades de autoconfiguración. Además, no siempre se tratan de objetos destinados a ser utilizados en ambientes informáticos,

deben ser lo suficientemente amigables para que una persona sin los conocimientos pueda ponerlos en funcionamiento sin mucha dificultad. Existirán tantos objetos que se necesitan más direcciones de internet. Con IPV6 se tienen 2^{128} direcciones únicas posibles, lo cual hace prácticamente imposible su agotamiento. Su uso es idóneo para el IoT ya que contemplan la gran cantidad de objetos conectados en simultaneo y todos aquellos que puedan fabricarse a lo largo del tiempo (Figuerola, 2014).

La gran cantidad de objetos conectados a el IoT y la gran cantidad de aplicaciones que se le pueden dar a los mismos generaran un gran volumen de datos constantemente. Prácticamente todo podría constituir una fuente de datos. Pero para que estos datos puedan tener algún valor, es decir, se puedan transformar en información útil y valiosa se requieren de mecanismos adecuados para procesarlos y almacenarlos. El big data (datos masivos) es justamente una tecnología ideada para la recolección y procesamiento de grandes volúmenes de información de tipo variado y no estructurada que se generan a gran velocidad. Existen hoy en día frameworks para Big Data como Hadoop que utilizan procesamiento distribuido.

Acceso a internet

Cada objeto conectado al IoT necesita un mecanismo de conexión a la red para cumplir su propósito. Esto implica que las compañías proveedoras de este servicio deben ajustarse a las nuevas necesidades y ofrecer planes o mecanismos con el costo e infraestructura adecuados para que el público en general perciba que están aportando un valor tangible.

2.3.2. Ventajas del Internet de las cosas

El control electrónico apoyado por la información aportada por sensores (monitorización), algo como aumentar la velocidad de giro de un ventilador cuando se detecta que sube la temperatura, ya era una realidad tecnológica y comercial mucho antes de que la IoT llegara para conectar los dispositivos a internet. Aún siendo una aportación muy interesante, la verdadera ventaja tampoco es la posibilidad de usar la red (seguramente la web) para conocer el estado de los dispositivos y controlarlos en remoto. Una de las ventajas más importantes de la integración de sensores en el IoT radica en el gran volumen de datos que pueden obtenerse para conocer un entorno de forma (estadísticamente) más precisa tanto de forma directa (big data) como indirecta (minería de datos).

En ese sentido, el alcance de la IoT se potencia extraordinariamente si se relaciona con las tecnologías de la inteligencia artificial, que pueden utilizar esta información para predecir y

tomar decisiones automáticas. Si se prefiere este otro enfoque, también se puede defender que una de las vías más importantes de obtención de datos para estas tecnologías (inteligencia artificial, big data, minería de datos...), que serían entonces las protagonistas, es la IoT (Victor Ventura, 2017).

2.3.3. *Internet Industrial de las cosas (IIoT) y la industria 4.0*

La industria 4.0 es la cuarta revolución industrial que es la informatización de los sistemas de fabricación. Tiene cyber-sistemas físicos que son la combinación de softwares y los activos de producción. Incluye la automatización industrial, internet de las cosas, intercambio de datos y cloud computing. Posee principalmente los siguientes componentes:

- *Interoperabilidad*: la capacidad de las máquinas, sensores y de las personas de comunicarse entre sí por medio de la IIoT y el Internet de las personas (IoP).
- *Transparencia*: La capacidad de los datos para hacer una versión virtual del mundo físico, mediante la mejora de modelos computadorizados con la información recopilada físicamente.
- *Orientación técnica*: La capacidad de los sistemas digitales para apoyar a la gente por el aprendizaje a partir de la información recopilada y tomar decisiones y cuidar de los problemas críticos de forma independiente. Es también la capacidad de digitalizar los sistemas para realizar tareas que no son adecuadas o peligrosa para las personas.
- *Opciones independientes*: la capacidad de los sistemas digitales para tomar decisiones por sí mismo y trabajar de forma independiente (Jayaram, 2016).

IIoT se centra en la automatización industrial, dispositivos de comunicación, flujo de datos, administración de dispositivos, integración y análisis predictivo.

La industria 4.0 puede ser implementado en la cadena de suministro global junto con IIoT, IoP, sensores y almacenamiento en la nube considerándose como aplicaciones del IoT en las industrias. La industria 4.0 se centra en la coordinación de la cadena de suministros, robots, sistemas integrados y automatización. La comparación entre estos dos conceptos se muestra en la Tabla 1-2.

Tabla 1-2: Comparación de Industria 4.0 e IIoT.

Comparación de la Industria 4.0 y IIoT	Conceptos Industriales	
	Industria 4.0	IIoT
Objetivos Tecnológicos	Automatización, sistemas embebidos, robótica	Automatización industrial, comunicación de dispositivos, administración de dispositivos, integración de dispositivos, análisis predictivo, flujo de datos
Área De Aplicación	Hardware	Hardware, Software integrados
El tamaño de la Industria	Ideal para las pequeñas y medianas industrias	Ideal para todas las industrias
Área De Optimización	Producción	Bienes o activos

Fuente: (Jayaram, 2016).

Realizado por: (Gallo,2018)

2.3.4. *Máquina a máquina (M2M)*

La tecnología de comunicación M2M lleva con nosotros desde hace más de una década en forma de gestión de flotas, redes de sensores y otras aplicaciones industriales, pero su progreso ha sido lento y se han hecho eco de él sobre todo las publicaciones especializadas. En la actualidad existen indicios de que la M2M está a punto de generalizarse y convertirse en el elemento que definirá la siguiente generación de numerosos productos y servicios centrados en el consumidor.

Ahora que las redes inalámbricas son fiables y accesibles en casi todo el mundo, nos encontramos con que los emprendedores de distintos sectores están participando activamente en la aplicación de la tecnología inalámbrica a sus propios productos y servicios. Este nuevo papel de M2M está siendo impulsado por una avalancha de factores, entre los que se incluyen la necesidad que tienen los operadores inalámbricos de expandirse más allá de las comunicaciones de voz, las grandes reducciones en el coste de los chipsets inalámbricos y la conectividad inalámbrica, las regulaciones gubernamentales centradas en el ahorro de energía y la seguridad pública, y una demanda sin precedentes por parte de los consumidores de cualquier cosa que sea móvil, según indica la rápida aceptación del smartphone y las tabletas inalámbricas.

Dos de las nuevas aplicaciones más importantes para M2M son la gestión de la energía. Esta iniciativa se ve en parte impulsada por las regulaciones gubernamentales. Empresas de suministros de red de todo el mundo están instalando redes inteligentes que conectan con contadores inteligentes para mejorar la eficiencia energética, y se están imponiendo sistemas de llamada de emergencia (como el sistema eCall en Europa) para agilizar la respuesta de emergencia en caso de accidente y ofrecer otras ventajas de seguridad pública. Sin embargo, la investigación de Accenture (empresa multinacional dedicada a la prestación de servicios de consultoría) prevé que estos importantes proyectos gubernamentales pronto quedarán eclipsados por una oportunidad de mercado mucho más amplia impulsada por los consumidores. El contador inteligente es solo el primer paso hacia la automatización del hogar que, con el tiempo, abarcará casi todos los aparatos, dispositivos de entretenimiento y otros artículos domésticos.

En resumen, muchas organizaciones de diferentes sectores se están dando cuenta de la importancia de M2M para sus propios negocios. No obstante, como observamos en nuestra investigación, la incorporación de la conectividad a un producto tradicional supone una complicación importante tanto desde una perspectiva técnica como desde el punto de vista de las empresas. El desarrollo de una solución de M2M efectiva requiere experiencia en tecnología de comunicaciones, hardware, aplicaciones de software, integración e implantación, y por lo general supone trabajar con múltiples agentes para obtener capacidades especializadas. Se necesita un examen de arriba abajo del proceso de negocio, ya que la incorporación de la conectividad suele desencadenar una transición desde un enfoque centrado en el producto hacia un modelo de negocio de servicio continuo que genera ingresos constantes a través de suscripciones o de ventas de productos complementarios.

2.3.5. *Protocolos de comunicación M2M*

Cada entorno del IIoT necesita un protocolo con capacidades diferentes, lo que hace realmente específico de qué protocolo tiene que ser usado para un determinado número de nodos, el alcance de la comunicación, los requisitos de potencia y fiabilidad. En las investigaciones sobre desarrollo del IIoT de hoy, los principales protocolos que se adaptan a cada entorno son MQTT y Constrained Application Protocol (CoAP). Estos protocolos son ligeros en términos de operación y transferencia de datos por lo tanto toman los mercados del IIoT.

Un ejemplo de aplicaciones utilizando los protocolos del IIoT son las redes de sensores inalámbricos que están siendo implementadas a gran escala para monitorear diferentes parámetros como temperatura, humedad, etc. Para asegurar que el mensaje/paquete sea

entregado o no, CoAP cuenta con un mecanismo de re-transmisión, mientras que MQTT confía en el protocolo de control de transmisión (TCP). En las zonas remotas que carecen de conectividad por cable o red, la única tecnología de acceso a internet disponible es la conexión inalámbrica o por satélite, que implican pérdidas y retrasos significativos.

Con este aumento de los dispositivos deben ser clasificados en función de diversos parámetros como el consumo de energía, el requisito de ancho de banda, factor de forma del dispositivo en sí. Cada dispositivo actuará como un nodo que recopila datos del entorno y que transmite a través de la red. Los protocolos utilizados para tales comunicaciones de nodo a nodo en el IIoT se conocen como protocolos M2M. La Figura 1-2 muestra la arquitectura de las aplicaciones M2M y sus protocolos.

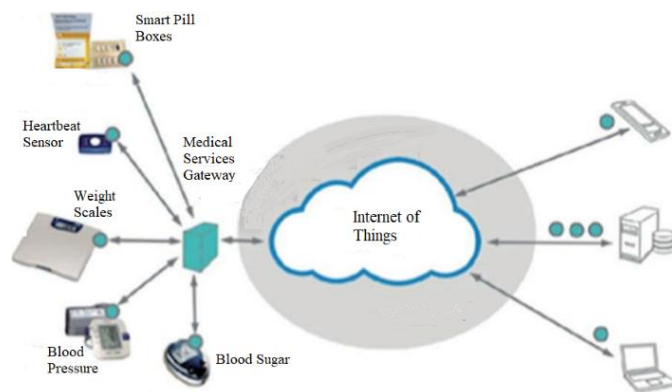


Figura 1-2: Arquitectura de M2M

Fuente: (Thota & Kim, 2016)

La oportunidad de M2M es dinámica y está creciendo rápidamente. Actualmente está proporcionando conectividad para una variedad de dispositivos como sensores, dispositivos móviles, sistemas de monitorización de la salud, medidores, etc. Esto se debe a que los objetos cotidianos que estamos usando están siendo actualizados, abordados, reconocidos y controlados a través de redes. Los protocolos que permiten realizar la comunicación M2M son:

- MQTT (Message Queuing Telemetry Transport).
- CoAP (Constrained Application Protocol).

Ambos protocolos mencionados anteriormente son estándares abiertos y son más adecuados para entornos restringidos que el protocolo de transferencia de hipertexto (HTTP). Estos protocolos tienen amplia gama de implementaciones para diferentes requisitos en el campo del IIoT, así como también ambos proporcionan mecanismos para la comunicación asíncrona y se ejecutan utilizando direcciones IP.

2.3.6. *Message Queuing Telemetry Transport (MQTT)*

En su totalidad, la señales que ingresan a los sistemas de control son desconocidas, aunque se puede tomar en cuenta que son aleatorias y es imposible expresarlo analíticamente. Únicamente en ciertos casos especiales es posible conocer dicha naturaleza de la señal y puede ser expresable mediante curvas o analíticamente, como es el caso de los controladores para herramientas de corte; en la mayoría del análisis y/o diseño de sistemas de control, hay que evaluar el comportamiento de diversos sistemas de control, esta base se la desarrolla especificando señales de entrada de prueba frente a señales de respuesta de varios sistemas. Mayormente los criterios para el diseño están basados en las señales anteriormente mencionadas o en respuestas al cambio en las condiciones iniciales, la utilización de señales de prueba han de justificar la existencia de una reciprocidad entre las características del sistema como respuesta a una entrada común de prueba y su capacidad de trabajar con señales reales (Ogata, Ingeniería de Control, 2010).

MQTT usa un tópico para definir un canal de comunicación entre los equipos. Aquellos que deseen recibir datos necesitan suscribirse bajo ese tópico en el bróker y así se los relacionará con un canal de comunicación, mientras que los dispositivos que deseen enviar datos necesitarán publicarlos bajo el tópico que relaciona al canal de comunicación deseado. En este protocolo, cuando un nuevo dato está disponible, su bróker simplemente envía los nuevos datos a cualquier dispositivo suscrito en éste (Osathanunkul, Hantrakul, Pramokchon, Khoenkaw, & Tantitharanukul, 2017).

A diferencia de otros protocolos, éste no necesita enviar mensajes constantes de solicitud para refrescar nuevos datos, por tanto, maneja una carga inferior de paquetes lo que lo hace más rápido.

Este protocolo está diseñado para entornos donde se tiene recursos limitados y responde a las siguientes necesidades:

- Está especialmente adaptado para utilizar un ancho de banda mínimo.
- Es ideal para utilizar redes inalámbricas.
- Consume muy poca energía.
- Posibilita un tiempo de respuesta superior al resto de protocolos web actuales.
- Permite una gran fiabilidad.
- Requiere pocos recursos procesadores y memorias.

2.3.7. *Aplicaciones de la comunicación M2M para aplicaciones industriales*

En comparación con las demás comunicaciones de internet, M2M ofrece más nodos de comunicación, menor ancho de banda, procesamiento en tiempo real, y muchos nodos se requieren para el uso eficiente de la energía. Dado que los datos son la preocupación más importante en los sistemas M2M, la recolección discreta, la transmisión fiable y el uso eficaz de los datos generados por la máquina son las tareas principales que deben realizarse (Thota & Kim, 2016). Los sistemas M2M son altamente dependientes de la infraestructura de la tecnología de la información (TI) para la recopilación de datos, el intercambio y la toma de decisiones. En términos de comunicación de datos para aplicaciones industriales, las especificidades de la comunicación M2M son:

- La heterogeneidad en las plataformas de hardware y software.
- Notificación frecuente y pares de observación de la máquina.
- Colaboración entre los equipos de automatización.
- Manejo de eventos en tiempo real y procesamiento de datos.
- Datos de diferentes formato y tamaño para transmitir.

Por lo tanto, la comunicación entre las máquinas conectadas, se espera que sea:

- Plataforma cruzada interoperable con normas básicas.
- Descubrimiento de la máquina y la capacidad de presencia.
- Interacción flexible de datos y notificación de eventos.
- Fiable, eficaz y rápido en la velocidad.
- Capacidad de recuperación y manejo de falla.

Dado que cada vez más dispositivos informáticos como nodos de sensores, lectores RFID, teléfonos inteligentes y ordenadores portátiles están integrados en la infraestructura de TI, la heterogeneidad en el hardware y el software y la distribución de estos dispositivos son problemas comunes para esta área de investigación en particular. Por lo tanto, se esperan soluciones genéricas y multiplataforma para una eficiente interacción de datos entre máquinas.

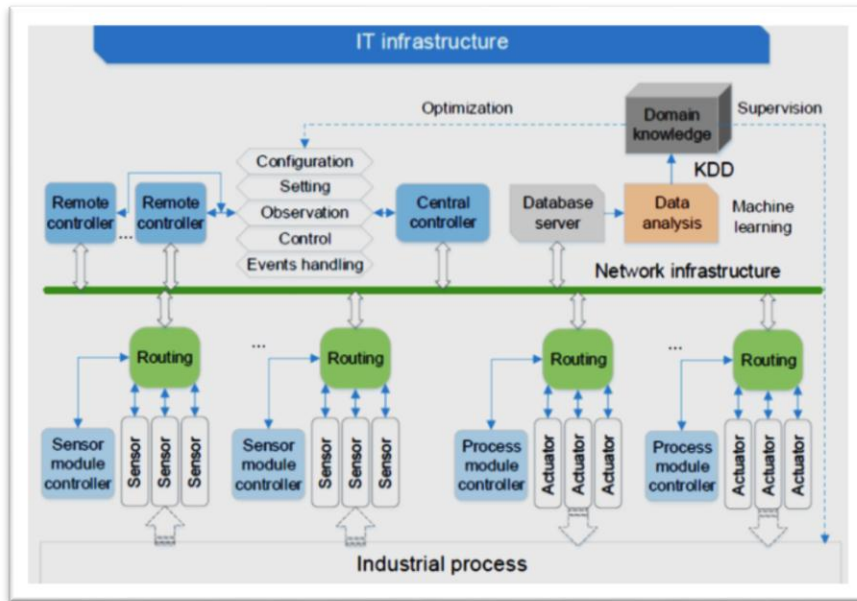


Figura 2-2: Arquitectura de red de M2M para aplicaciones industriales.

Fuente: (Ghanbari, Laya, & Markendahl, 2016)

2.3.8. *Raspberry Pi*

Es un ordenador de bajo costo. Consta de una placa base sobre la que se monta un procesador, un chip gráfico y memoria RAM. Se puede considerar un ordenador completo. La diferencia con un ordenador normal es que no incluye cable de alimentación, la caja ni el disco duro, para el que se utiliza una tarjeta SD ver Figura 3-2. Las especificaciones técnicas del modelo que se utiliza se visualizan en la Tabla 2-2.

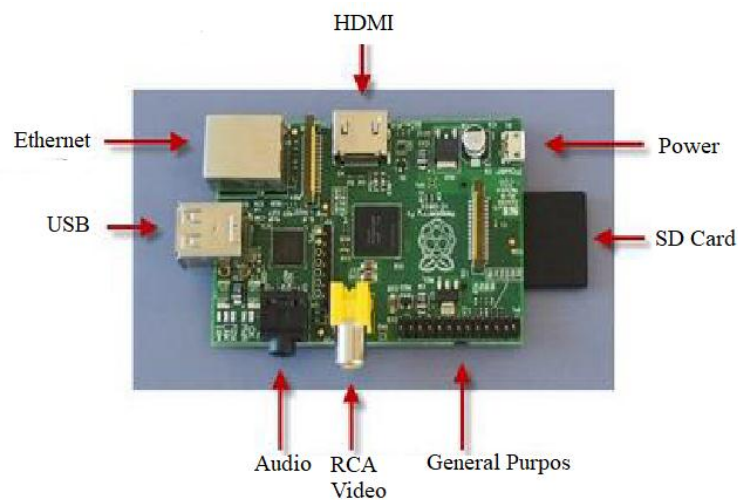


Figura 3-2: Dispositivo Raspberry Pi

Fuente:(Humano, 2012).

Tabla 2-2: Especificaciones técnicas Raspberry Pi

CPU	ARM1176JZF-S a 700 MHz (familia ARM11)
Memoria (SDRAM)	512 MB (compartidos con la GPU) desde el 15 de octubre de 2012
Almacenamiento integrado	SD / MMC / ranura para SDIO
Conectividad de red	10/100 Ethernet (RJ-45) via hub USB
Consumo energético	700 mA, (3.5 W)
Fuente de alimentación	5 V vía Micro USB o GPIO header
Dimensiones	85.60mm × 53.98mm
Sistemas operativos soportados	Debian (Raspbian), Fedora, Arch Linux, Slackware Linux, RISC OS

Fuente: (Humano, 2012)

Debido a sus características físicas mostradas en la Tabla 2-2, la raspberry pi es una herramienta que posee la tecnología utilizada para la generación, adquisición, almacenamiento y análisis de datos y que permitiera la conexión entre ellos, independiente del hardware. De esta manera se determinó utilizar este microcontrolador de placa reducida por tener una amplia gama de aplicaciones generalmente orientadas al uso del software y también tomando en cuenta que tiene la capacidad de alojar varios lenguajes de programación.

CAPÍTULO III

3. METODOLOGÍA INVESTIGATIVA

Este capítulo trata sobre la metodología investigativa que se realizó para obtener los resultados del proyecto.

3.1. Tipo de Investigación

- Finalidad o propósito: la investigación es de tipo aplicada por que se resuelve una problemática en base a los objetivos del proyecto.
- Alcance: el proyecto se desarrolla en un tiempo determinado ya que se tuvo un cronograma de actividades a realizarse.
- Amplitud: la investigación es de tipo microsociológica por que el proyecto posee variables que están relacionados a un grupo determinado de estudio que esta la automatización industrial.
- Profundidad: la investigación es de tipo descriptiva por que el proyecto analiza e implementa una nueva tendencia que implica al IIoT y la Industria 4.0.
- Fuentes: el proyecto posee fuentes primarias, debido a que los contenidos del tema de investigación se realizaron consultando en libros de electrónica, revistas y publicaciones de internet referentes al IIoT y sus protocolos de comunicación.
- Lugar que se desarrolla: la investigación es de campo, para lo cual se utilizó el laboratorio y se comprueba el uso de los protocolos que existen para la comunicación de IIoT mediante la aplicación en un dispositivo de control PLC y el uso de sistemas alternativos.
- Naturaleza: la investigación es de tipo documental y experimental, porque el proyecto se desarrolla a partir de documentos científicos y documentales, se verifica en el laboratorio su funcionamiento.

3.2. Recolección de Información

Fuentes, técnicas e instrumentos.

Para el presente proyecto se recopiló información de libros, internet, revistas científicas, trabajos de investigación, papers, patentes, guía del tutor y los miembros para el desarrollo del proyecto, etc.

3.3. Procesamiento y Análisis de datos

Una vez que se ha obtenido la información necesaria acerca de la investigación, esta fue procesada de manera ordenada y sistemática, de acuerdo con la matriz de consistencia de la Tabla 1-3 y Tabla 2-3.

Tabla 1-3: Aspectos específicos

Variable Dependiente: Internet Industrial de las cosas y la Industria 4.0			
Conceptualización	Dimensión	Indicadores	Técnicas e Instrumentos
Internet Industrial de las cosas (IIoT) se centra en la automatización industrial, dispositivos de comunicación, flujo de datos, administración de dispositivos, integración y análisis predictivo. La industria 4.0 puede ser implementado en la cadena de suministro global junto con IIoT, IoP, sensores y almacenamiento en el internet	Flujo de datos, administración de dispositivos, integración	Transmisión y Almacenamiento de datos	Investigación, revisión, bibliografía y desarrollo de aplicaciones con multiplataforma
Variable Independiente: Monitoreo de controladores lógicos programables (PLC) desde una página web			
Conceptualización	Dimensión	Indicadores	Técnicas e Instrumentos
Un controlador Lógico Programable es un dispositivo industrial que programa, procesa y guarda información para la automatización de procesos industriales los datos que en sí guarda serán monitoreados desde una página web	Monitoreo de datos Diseño de páginas web	Almacenamiento y lectura de datos	Investigación, revisión, bibliografía y desarrollo de aplicaciones con multiplataforma

Realizado por: Lucia Gallo, 2018

Tabla 2-3: Aspectos generales

MATRIZ DE CONSISTENCIA						
TITULO: DESARROLLO E IMPLEMENTACIÓN DE INTERNET INDUSTRIAL DE LAS COSAS APLICADO AL LABORATORIO DE PLC's DE LA FACULTAD DE INGENIERIA INDUSTRIAL EN PROCESOS DE AUTOMATIZACION DE LA UNIVERSIDAD TECNICA DE AMBATO						
AUTOR: LUCIA GALLO						
FORMULACIÓN DEL PROBLEMA	OBJETIVOS	HIPÓTESIS GENERAL	VARIABLES	INDIADORES	TÉCNICAS	INSTRUMENTOS
¿Cómo se realiza una comunicación del internet industrial de las cosas(IIoT) en un sistema de automatización industrial?	Desarrollar e implementar el internet industrial de las cosas aplicado al laboratorio de PLC's de la facultad de ingeniería industrial en procesos de automatización de la Universidad Técnica de Ambato	¿El desarrollo del IIoT para integración de datos a nivel de procesos industriales basado en el protocolo MQTT permite tener tiempos óptimos de envío de paquetes de datos?	V.D: Internet Industrial de las cosas y la Industria 4.0 V.I: Monitoreo de controladores Lógicos Programables (PLC) desde una página web.	Ventajas y desventajas	Investigación bibliográfica	Internet Libros Bases de datos científicas

Realizado por: Lucía Gallo.2018

3.4. Desarrollo del Proyecto

Para el desarrollo de la investigación se efectuó los siguientes pasos:

- Describir el concepto del IIoT sus protocolos, comunicación M2M e Industria 4.0
- Citar las aplicaciones del IIoT para sistemas de automatización Industrial.
- Ilustrar los diferentes sistemas y dispositivos que se pueden conectar al internet mediante el IIoT.
- Determinar los requerimientos de hardware y software que el desarrollo de proyecto emplea en su ejecución.
- Establecer la comunicación que permita enlazar la Raspberry pi con el dispositivo de control el PLC Siemens 1200.
- Construir la simulación de la adquisición de datos y estados utilizando una base de datos que se encuentra alojado en la Raspberry.
- Desarrollar la implementación de la plataforma para el monitoreo de los datos contenidos en el dispositivo de control PLC mediante un servidor alojado en la Raspberry.
- Evaluar los ensayos de funcionamiento de la plataforma para el monitoreo de datos y estados que posee el PLC ya desde la página web.

CAPÍTULO IV

4. PRUEBAS Y RESULTADOS

Los resultados obtenidos fueron las páginas web que se programaron utilizando PHP como se puede ver en el **Anexo C**, se ingresa a cualquier navegador con la dirección 192.168.1.4/inicio.php la misma que corresponde a la validación del sistema donde se pide el usuario y la contraseña como se muestra en la Figura 1-4:

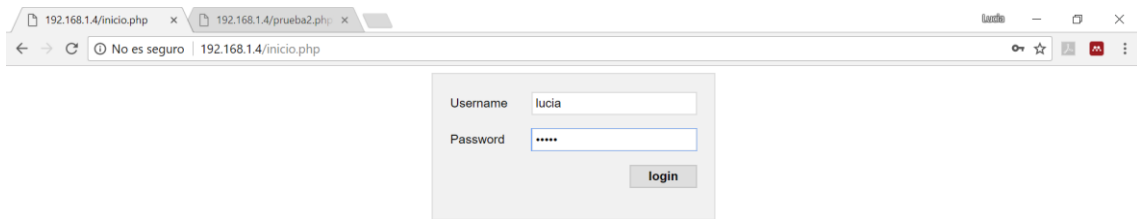


Figura 1-4: Página Web de los usuarios para acceder al sistema.

Realizado por: Lucia Gallo.2018

Cuando se ingresa al sistema mediante la validación, la siguiente página web que aparece es la que se muestra en la Figura 2-4, en donde se selecciona el tipo de datos que se quiere revisar como son los 'Datos analógicos' y los 'Datos digitales' respectivamente.

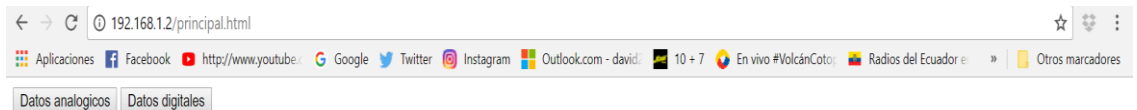


Figura 2-4: Datos para acceder al sistema.

Realizado por: Lucia Gallo.2018

La siguiente página web muestra la selección de los datos analógicos del sistema, para esto se colocó un sensor de temperatura (LM55) en una de las entradas analógicas del PLC Siemens 1200 denominada IW66, se configuró y se escaló para que los datos adquiridos se muestren en grados Celsius, como se representa a continuación; el número de identificación los 42 datos leídos, la temperatura de 33 grados Celsius la fecha y hora que fue adquirido el dato visualizado en la Figura 3-4:

42	33	2017-09-30 00:07:24
ID	Temperatura	fecha

Figura 3-4: Datos analógicos de la entrada del PLC.

Realizado por: Lucia Gallo.2018

La última página web muestra los datos digitales de los estados de las salidas del PLC Siemens 1200, la primera columna muestra el número de identificación ‘ID’ de los datos leídos, las 8 salidas denominadas Q0.0, Q0.1, Q0.2, Q0.3, Q0.4, Q0.5, Q0.6, Q0.7, ubicadas en la segunda columna de manera ordenada y en la tercera columna da los estados que se encuentran dichas salidas siendo en ‘1’ el estado de encendido o habilitado y ‘0’ el estado de apagado o inhabilitado, como se observa en la Figura 4-4:

ID	Q0.	estado
33	0	1
34	1	1
35	2	0
36	3	0
37	4	0
38	5	0
39	6	1
40	7	1

[logout](#)

Figura 4-4: Datos digitales de las salidas del PLC.

Realizado por: Lucia Gallo.2018

Prueba de entrega de paquetes de datos utilizando el protocolo MQTT

Con el propósito de validar el tiempo de respuesta (latencia) del protocolo MQTT configurado en este proyecto vs la configuración del envío de datos que se realizó utilizando un simulador específico del uso del protocolo MQTT que de manera gratuita está disponible en la página de la empresa HiveMQ, este software facilita la entrega de datos y da gráficas de métricas que ayudan analizar el tiempo de respuesta, la empresa siendo unas de las pioneras en el desarrollo del

protocolo MQTT maneja óptimos tiempos de respuesta ya que se especializan en esta rama y utilizan la última versión del protocolo, para el proyecto se utiliza el software whireshark.

La empresa HiveMQ fue creada como el único punto de contacto para todas las comunicaciones de dispositivos en la red utilizando el protocolo del IoT denominado MQTT, responsable de manejar, asegurar y procesar todos los datos entrantes y salientes. Son un equipo de expertos cuando se trata de comunicación de dispositivos con MQTT. Participan activamente en el avance de MQTT como miembro del comité de estandarización en OASIS. Son parte de la primera especificación MQTT abierta y ahora trabajan para la próxima versión del protocolo. Esta empresa posee un sitio online donde utilizan el protocolo MQTT para envío de datos ya sean estados u otros datos como temperatura, etc. Los mismos que hay que configurarlos como se muestra a continuación en la Figura 5-4, la cual muestras los datos suscritos a tópicos o temas que se desea publicar y la generación de datos de temperatura adquiridas por el PLC Siemens 1200 en la página denominada MQTT Dashboard.

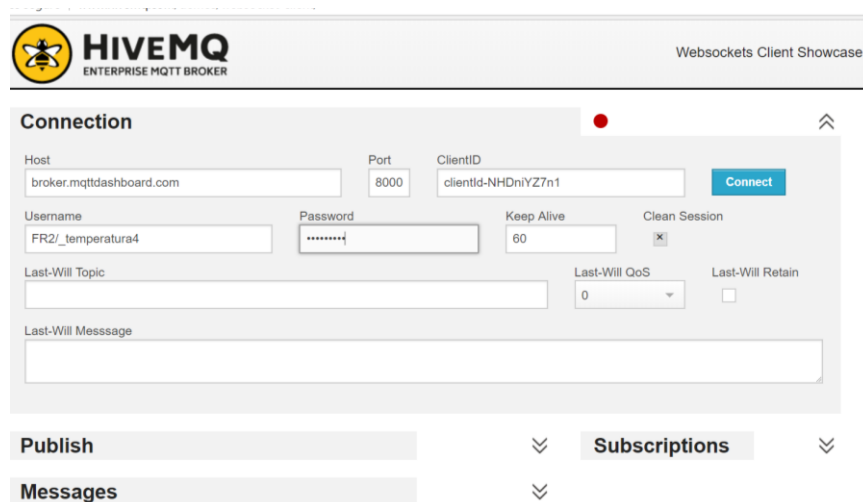


Figura 5-4: Configuración de datos de temperatura adquiridos con el PLC

Fuente:(HIVEMQ, 2012)

En la siguiente Figura 6-4, se visualiza los datos de temperatura adquiridos del PLC Siemens 1200 y enviados con MQTT los cuales utilizan el bróker HiveMQ MQTT. El mismo que puede ser usado por cualquier cliente MQTT o biblioteca para publicar en el bróker ya mencionado.

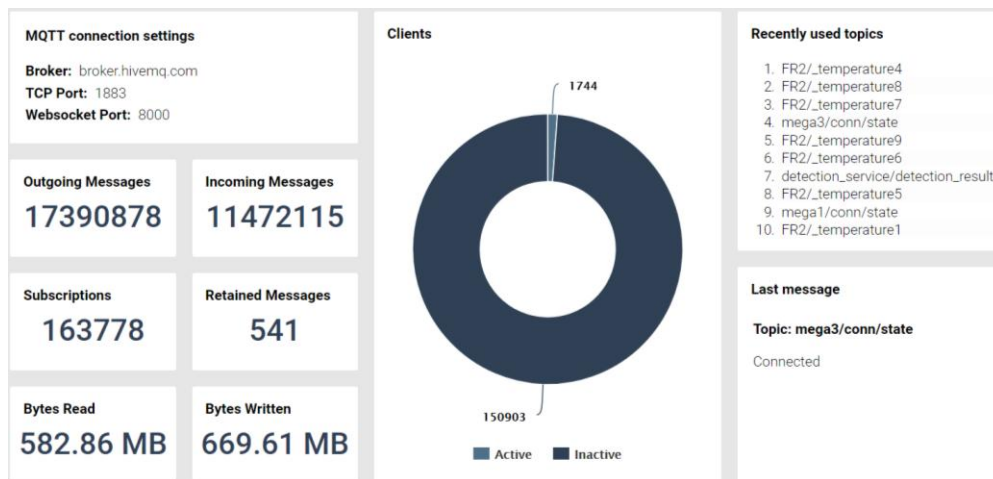


Figura 6-4: Configuraciones de conexión MQTT

Fuente:(HIVEMQ, 2012)

Para proporcionar una base sólida para el monitoreo continuo, HiveMQ contiene una gran cantidad de métricas que permiten una visión profunda del estado actual y la carga de datos que se envían por los clientes al bróker en el caso del proyecto se envían datos de temperatura adquiridos por el PLC Siemens 1200 ya que estos datos ocupan más longitud en tanto al tamaño de los paquetes de datos.

En la página web de la empresa hay un extracto de algunas de esas métricas y sus datos históricos de uno de los corredores actualmente en ejecución incluidos los datos del proyecto.

Los datos que se envía a esta página utilizan el protocolo MQTT y se puede monitorear, longitud de paquete y el tiempo de ejecución que se muestra en la Figura 7-4, se puede discriminar los datos del proyecto debido a que se ocupa la página como cliente del bróker.

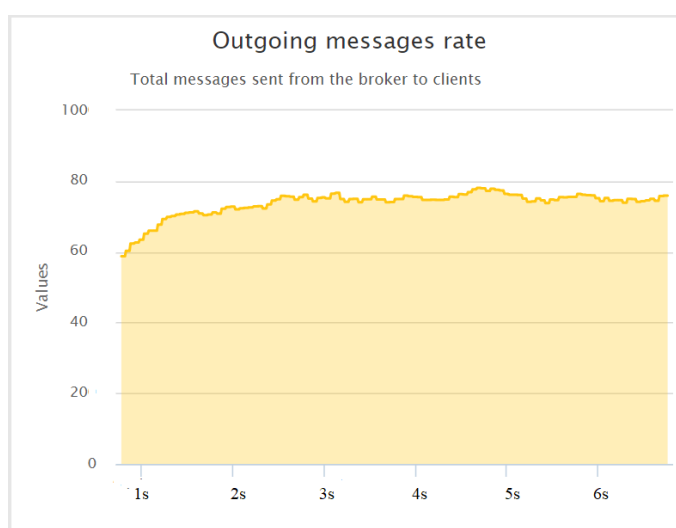


Figura 7-4: Métricas y datos históricos de la tasa de mensajerías en ejecución.

Fuente: (HIVEMQ, 2012).

En la Figura 7-4 se puede observar que el tiempo de retardo entre paquetes de datos de temperatura enviados a la página de HiveMQ, del sistema como cliente del bróker se encuentra en el rango de 0,11 y 0,98 milisegundos, siendo 0,54 milisegundos el tiempo promedio de retardo entre los datagramas enviados concepto que se conoce como latencia de la red.

Las pruebas del sistema se realizan probando enviar los paquetes de datos utilizando el protocolo MQTT y haciendo la comparación entre el tamaño del paquete y el tiempo que se demora en ejecutarse la acción eso en redes se denomina latencia así por ejemplo, de acuerdo a la estructura del protocolo conectamos al comando luego espera la respuesta de reconocimiento se publica el mensaje y finalmente se desconecta esperando un nuevo paquete que ejecutaría la misma acción repetitivamente a continuación se muestra la Figura 8-4 en donde se capturan el tráfico de la red:

No.	Time	Source	Destination	Protoco	Length	Info
7	0.277489	192.168.1.3	192.168.1.2	MQTT	80	Connect Command
8	0.278097	192.168.1.2	192.168.1.3	MQTT	70	Connect Ack
10	0.279837	192.168.1.3	192.168.1.2	MQTT	76	Publish Message
11	0.280207	192.168.1.3	192.168.1.2	MQTT	68	Disconnect Req
18	1.285854	192.168.1.3	192.168.1.2	MQTT	80	Connect Command
19	1.286312	192.168.1.2	192.168.1.3	MQTT	70	Connect Ack
21	1.288094	192.168.1.3	192.168.1.2	MQTT	76	Publish Message
22	1.288511	192.168.1.3	192.168.1.2	MQTT	68	Disconnect Req
32	2.294058	192.168.1.3	192.168.1.2	MQTT	80	Connect Command
33	2.294586	192.168.1.2	192.168.1.3	MQTT	70	Connect Ack
35	2.296383	192.168.1.3	192.168.1.2	MQTT	76	Publish Message
36	2.296797	192.168.1.3	192.168.1.2	MQTT	68	Disconnect Req
51	3.302318	192.168.1.3	192.168.1.2	MQTT	80	Connect Command
52	3.303028	192.168.1.2	192.168.1.3	MQTT	70	Connect Ack
54	3.304922	192.168.1.3	192.168.1.2	MQTT	77	Publish Message
55	3.305297	192.168.1.3	192.168.1.2	MQTT	68	Disconnect Req
63	4.310960	192.168.1.3	192.168.1.2	MQTT	80	Connect Command
64	4.311443	192.168.1.2	192.168.1.3	MQTT	70	Connect Ack

**LISTA DE PAQUETES
CAPTURADOS**

> Frame 7: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) > Ethernet II, Src: Raspberr_a5:b5:ac (b8:27:eb:a5:b5:ac), Dst: Cats_13:06:0b (00:08:15:13:06:0b) > Internet Protocol Version 4, Src: 192.168.1.3, Dst: 192.168.1.2 > Transmission Control Protocol, Src Port: 44851, Dst Port: 1883, Seq: 1, Ack: 1, Len: 14 > MQ Telemetry Transport Protocol, Connect Command						
---	--	--	--	--	--	--

DETALLE DEL PAQUETE

```

0000  00 08 15 13 06 0b b8 27  eb a5 b5 ac 08 00 45 00  .....E.
0010  00 42 01 66 40 00 40 06  b5 fa c0 a8 01 03 c0 a8  .B.f@.
0020  01 02 af 33 07 5b a4 b1  a1 f7 b9 ef da c8 80 18  ...3.[.
0030  00 e5 2d 0f 00 00 01 01  08 0a 00 15 a0 92 95 97  .....
0040  47 3a 10 0c 00 04 4d 51  54 54 04 02 00 3c 00 00  G.....MQ TT...

```

BYTE DEL PAQUETE / TRAMA

Figura 8-4: Lista de paquetes capturados utilizando el programa Wireshark.

Realizado por: Lucia Gallo.2018

La transmisión se realiza con el protocolo MQTT con detalles de los paquetes enviados y la trama que contiene, cada vez que se transmite un paquete como se observa en la Figura 8-4 anterior a continuación se establece un registro de datos enviados y luego se realizan operaciones de estadísticas para poder evaluar el comportamiento del tamaño de los paquetes enviados respecto al tiempo que demora su ejecución es decir la latencia de la red, obteniendo el siguiente resultado como se muestra en la Figura 9-4:

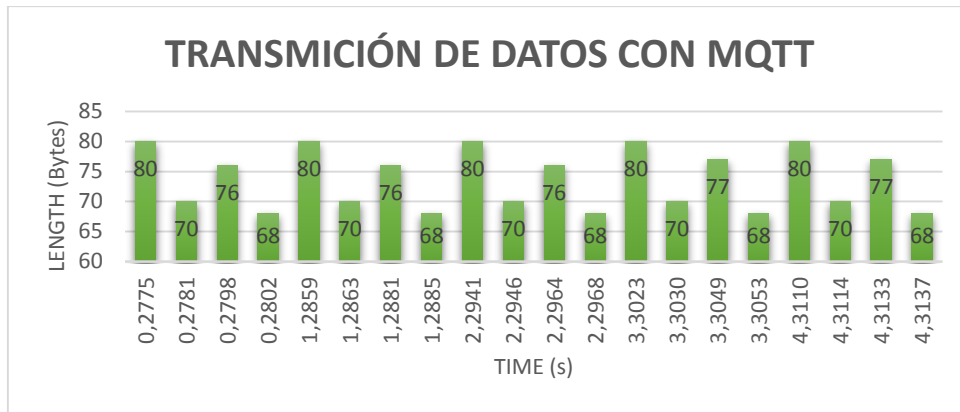


Figura 9-4: Datos estadísticos de los paquetes enviados utilizando el protocolo MQTT.

Realizado por: Lucia Gallo.2018

En la Figura 9-4 se puede observar que el tiempo de retardo entre paquetes de datos de temperatura y estados de las salidas digitales del PLC Siemens 1200 se encuentra en el rango de 0,4 milisegundos y 1 segundo, siendo 1 segundo el tiempo máximo de retardo entre los datagramas enviados concepto que se conoce como latencia de la red. A continuación, se muestra los datos estadísticos de la comparativa entre los envíos de paquetes de datos de la misma longitud, pero el tiempo de respuesta o latencia de la red en el sistema es; 0,6 milisegundos y para HiveMQ es de 0,54 milisegundos en promedio, por lo tanto, el rendimiento es aceptable en los dos casos respecto al tráfico de red en el sistema.

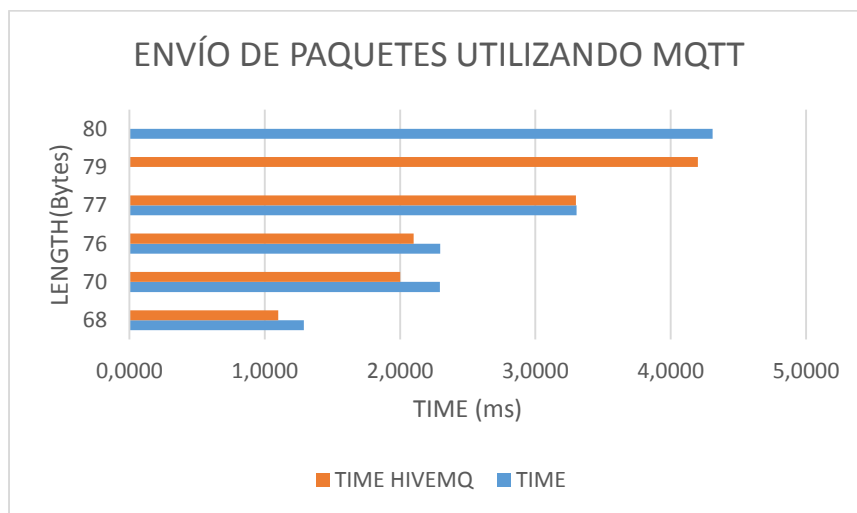


Figura 10-4: Comparación de la longitud vs tiempo de paquetes enviados con MQTT.

Realizado por: Lucia Gallo.2018

4.1. Verificación de la Hipótesis

La tesis planteó como hipótesis de partida, el siguiente enunciado ¿El desarrollo del IIoT para integración de datos a nivel de procesos industriales basado en el protocolo MQTT permite tener tiempo óptimo de envío con respecto a la longitud de paquetes?.

Para poder analizar el tiempo óptimo de envío de paquetes se recurre a la latencia que es el tiempo que tarda en transmitirse un paquete dentro de la red, y es un factor clave en las conexiones a Internet.

Factores que generan latencia

Para reducir la latencia tenemos que fijarnos en los factores que pueden influir negativamente en nuestra conexión. Los más comunes son los siguientes:

- La tecnología de acceso a Internet: Asymmetric Digital Subscriber Line (ADSL) o Fibra.
- La distancia entre los dos puntos que quieran establecer la comunicación, y las redes o saltos intermedios por los que tengan que pasar los paquetes.
- Capacidad del dispositivo desde el que nos conectamos (ordenador, portátil, tablet, móvil o consola) y la carga del servidor al que nos estamos conectando.

Se ha mejorado mucho la tecnología que influye en la latencia. En el ADSL, unido a la mejora de la velocidad, se han ido eliminando puntos intermedios que generaban retrasos en la señal debido a que era necesario eliminar ruidos. En la actualidad, el menor ping que se puede conseguir en ADSL es gracias a la tecnología fastpath.

Los gráficos que a continuación se presentan son datos de latencia para los diferentes servidores utilizando un medio físico de fibra y ADSL:

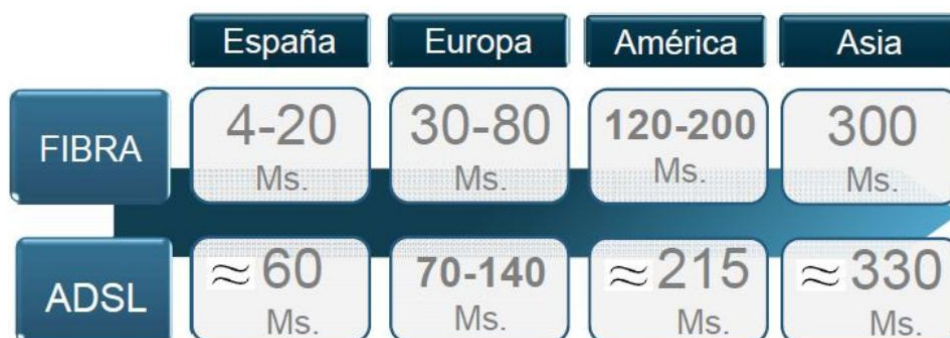


Figura 11-4: Latencia de los diferentes servidores por un medio físico de fibra y ADSL

Fuente: (Sanz, 2005)

Otros datos tomados de los servidores de América, España y UE se tienen los siguientes datos de latencia.

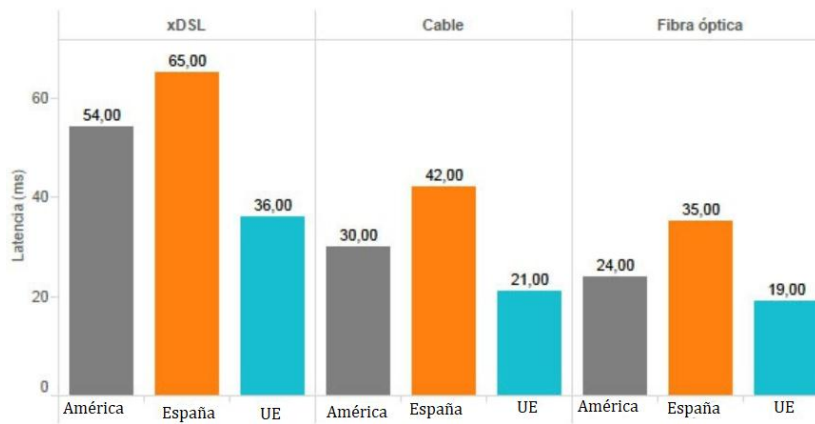


Figura 12-4: Latencia medida en América, España y Europa.

Fuente:(Sanz, Valero, & Gómez, 2016).

Como consecuencia de las pruebas realizadas y alcanzadas en este proyecto se puede inferir que de acuerdo a los resultados obtenidos en el análisis del tráfico de la red utilizando el software wireshark para el proyecto y MQTT Dashboard de la empresa HiveMQ son: 0,6 milisegundos y 0,54 milisegundos en promedio respectivamente, por lo tanto, el rendimiento y retardo de la red en el proyecto está por debajo de los valores antes mencionados en la información de latencia que se viene manejando de acuerdo al medio físico en fibra óptica que se maneja en un rango de 24 milisegundos y 200 milisegundos, con un promedio de 112 milisegundos concluyendo que la latencia para los datos del proyecto es mínima y que no está en el rango, esto se debe a que los datos son estados y temperaturas en un tamaño de bytes, probando así que el protocolo MQTT ocupa una mínimo ancho de banda y que la latencia es buena debido a esta característica.

CAPÍTULO V

5. PROPUESTA DE SOLUCIÓN

Este capítulo trata de la propuesta que se realizó en base a la contextualización de conceptos, dando la solución al tema del proyecto alcanzando los objetivos y comprobando la hipótesis dada.

5.1. Introducción

En la presente se describe la solución sobre la aplicación del IIoT para el laboratorio del PLC's de la Universidad Técnica de Ambato se propone que mediante la ayuda de un dispositivo de placa reducida en donde previamente se instaló una serie de software utilizados para permitir la comunicación entre este dispositivo y el PLC Siemens 1200 se recoge datos y estados los mismo que son almacenados en una base de datos y presentados una página web que muestra esta información disponible a cualquier dispositivo que se encuentre en la red creada para mostrar esta solución, para esto se utiliza protocolos, librerías y soluciones de administración de datos que corresponde al desarrollo del IIoT.

5.2. Propuesta de Solución

La programación que se realiza para la conexión entre el PLC Siemens 1200 y la Raspberry Pi fue mediante el uso del software PYTHON donde se utiliza la librería SNAP7 que permite adquirir los datos del PLC los cuales son enviados utilizando el protocolo del IIoT denominado MQTT, luego se utiliza el software PhpMyAdmin para crear una base de datos donde se almacena los estados y datos, por último el software PHP que mediante páginas web permite presentar los datos y estados del PLC, para monitorear desde cualquier dispositivo inteligente que se conecte a la red wifi configurada previamente, el diagrama de bloque que se muestra a continuación en la Figura 1-5 contiene la estructura del proceso que se llevara a cabo.

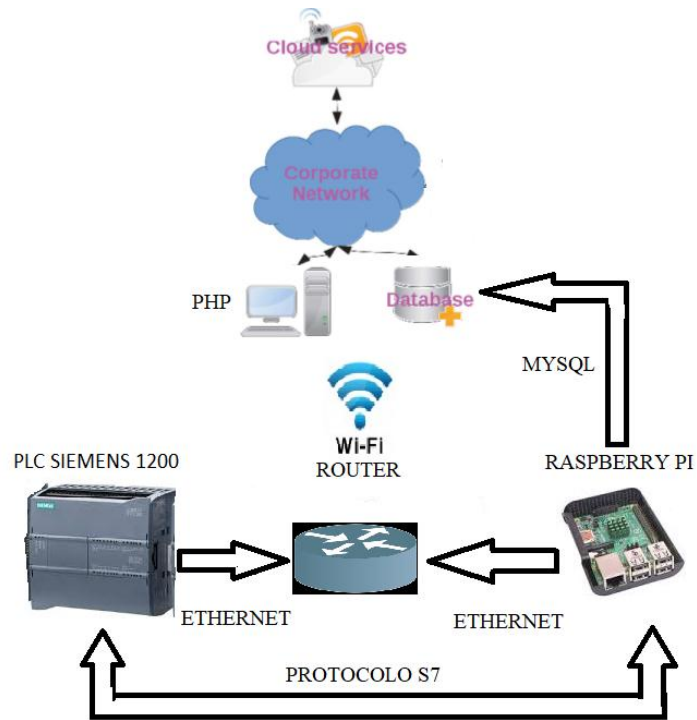


Figura 1-5: Diagrama de bloques.

Realizado por: Lucía Gallo. 2018

5.2.1. *Requerimientos de Hardware*

En la Tabla 1-5 se enumeran los requerimientos de hardware del sistema.

Tabla 1-5: Requerimientos de Hardware

Requerimiento	Material Utilizado
PLC	PLC Siemens S7 1200
Raspberry pi3	Dispositivo Embebido mini computadora Costo: 60 USD.
Router (WiFi)	Dispositivo de Red
PC	Intel (R) Core (TM) i5-4210U CPU @ 2.4 GHz
Fuente de alimentación	Fuente de DC, 1000 mA, 5V.

Realizado por: Lucia Gallo. 2018

5.2.2. *Requerimientos de Software*

En la Tabla 2-5 se enumeran los requerimientos de hardware del sistema.

Tabla 2-5: Requerimientos de Software

Requerimiento	Material Utilizado
Software para Monitoreo	PHP
Software de Programación de comunicación y adquisición	Phyton 2
Software de la Raspberry Pi3	OS Debian
Software para Prueba MQTT	Mosquitto y PAHO
Software para base de Datos	Php My Admin

Realizado por: Lucia Gallo 2018

5.3. Configuración del PLC Siemens 1200

El PLC Siemens 1200 posee una CPU que procesa y controla cambios de entradas/salidas de acuerdo con la lógica del programa que el usuario implemente, se puede incluir lógica booleana, temporizadores, conteo, operaciones matemáticas, adquisición de datos analógicos y las comunicaciones con otros dispositivos de tipo industrial.

Para la configuración del módulo ethernet del PLC Siemens 1200 únicamente se debe confirmar que la dirección IP sea coherente con la red que se encuentren los demás dispositivos en este caso se asignó la 192.168.1.2 como se muestra en la siguiente Figura 2-5:

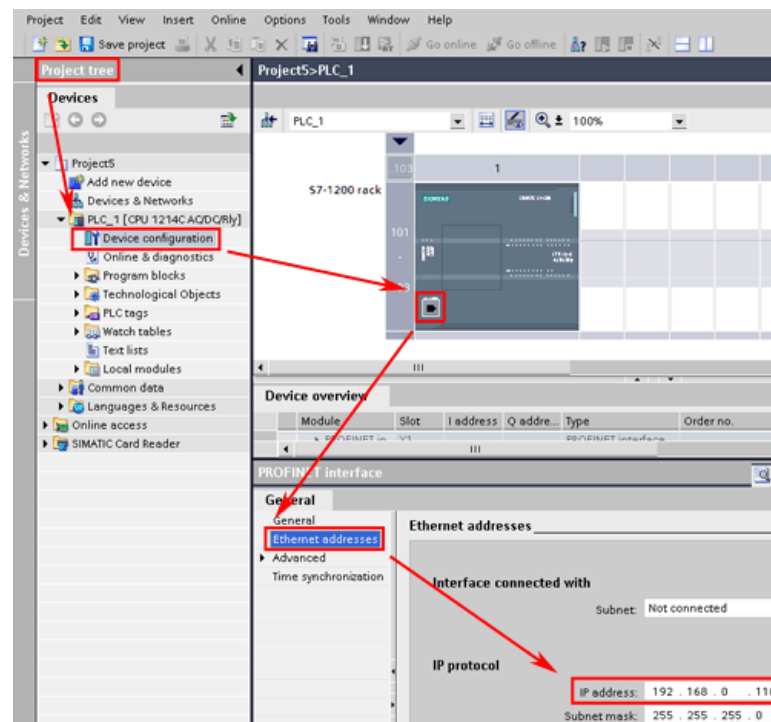


Figura 2-5: Dirección Ip del PLC Siemens 1200.

Realizado por: Lucía Gallo. 2018

El dispositivo raspberry al estar conectado al router el mismo le asignara una dirección IP dentro de la red.

Los datos que contiene el PLC en sus entradas tanto analógicas y salidas digitales van a ser mostradas en la página web que se desarrollo en la plataforma Php.

5.4. Programación en Python

Python es un lenguaje de programación poderoso cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

Python contiene una gran cantidad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas comunes sin necesidad de tener que programarlas desde cero.

Las librerías pueden ayudar a hacer varias cosas como expresiones regulares, generación de documentos, evaluación de unidades, pruebas, procesos, bases de datos, navegadores web y otras funciones dependientes del Sistema.

Python se utilizó para instalar la librería SNAP7 su código se muestra en el **Anexo A**, el cual se necesitó para la comunicación entre el PLC Siemens 1200 y la raspberry denominado el protocolo Ethernet S7.

Comunicación del PLC Siemens 1200 con el Protocolo S7.

El protocolo S7 es el backbone de las comunicaciones de Siemens, su implementación de Ethernet se basa en ISO (Interconexión del sistema abierto) TCP (Protocolo de control de transmisión) (RFC1006) por su diseño que está orientado a bloques.

Cada bloque se denomina PDU (Unidad de datos de protocolo), su longitud máxima depende del CP (Proceso de comunicación) y se negocia durante la conexión.

El protocolo S7 está basado en funciones y comandos, es decir, cada transmisión contiene un comando y una respuesta al sistema como se puede ver en la programación del **Anexo A**. Si el

tamaño del paquete de un comando no cabe en una PDU, entonces debe dividirse en más PDU posteriores.

Cada comando está estructurado con los siguientes elementos:

- Un encabezado.
- Un conjunto de parámetros.
- Datos de parámetros.
- Un bloque de datos.

Los primeros dos elementos están siempre presentes, los otros son opcionales.

El protocolo S7, ISO TCP y TCP / IP siguen la regla de encapsulación que consiste en que cada telegrama es la parte de "payload" del protocolo subyacente su estructura se muestra en la Figura 3-5:

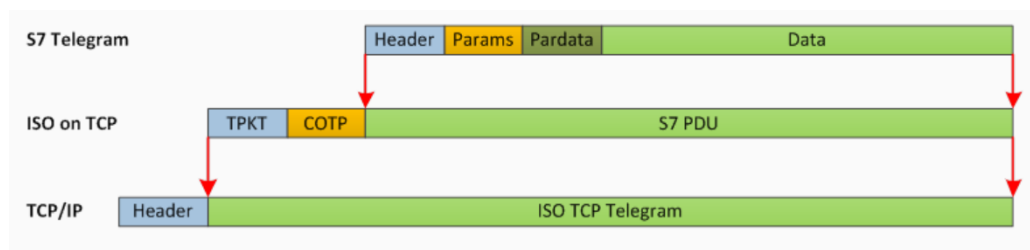


Figura 3-5: Encapsulación del protocolo S7

Fuente:(Github, 2014)

La biblioteca S7 está diseñada teniendo en cuenta las grandes transferencias de datos industriales de tiempo crítico que involucran redes con PLC's y posee 3 tipos de configuraciones: client, server y partner que cumplen tareas específicas como son:

- Client solo puede realizar consultas.
- Server solo puede responder.
- Partner puede hablar por su propia iniciativa.

Para la aplicación de IIot que se está manejando, la comunicación S7 es de tipo Client la cual expone tres características interesantes: PDU Independence, SmartConnect y Asynchronous data.

PDU Independence

Esta característica se refiere a que cada paquete de datos que se intercambian con un PLC debe caber en una PDU, cuyo tamaño es fijo y varía desde 240 hasta 960 bytes.

Todas las funciones de S7 ocultan completamente este concepto, los datos que puede transferir en una sola llamada dependen únicamente del tamaño de los datos disponibles.

Si este tamaño de datos excede el tamaño de la PDU, el paquete se divide automáticamente en más transferencias posteriores.

SmartConnect

La función SmartConnect se basa en el principio de dar “ping” si podemos hacerlo entres dos dispositivos la conexión será exitosa, en S7 evita el tiempo de espera de la conexión TCP cuando un PLC está apagado o el cable de red no está conectado.

A diferencia del tiempo de espera de la conexión TCP, el tiempo de ping es fijo.

Asynchronous Data

En esta característica el trabajo de protocolo S7 consiste en:

- Preparación de datos.
- Transmisión de datos.
- Esperando la respuesta.
- Descodificación del telegrama de respuesta.

Cada bloque transmitido se denomina PDU, que es el bloque más grande que se puede gestionar por transmisión beneficiándose de la ejecución asincrónica.

Para la conexión del PLC Siemens 1200 de CPU 1214 C AC/DC/Rly se usa la configuración Client con la siguiente librería que contiene el código base que será ejecutado en la raspberry en el programador python. (Phython-snap7, 2013) que contiene la programación de la comunicación como se puede observar en el **Anexo A**.

PLC = snap7.client.Client

(librería importada a Python)

El programa está estructurado como se muestra en la siguiente Figura 4-5 donde se encuentra el código en python el mismo que ayuda a la conexión y comunicación con el PLC Siemens 1200 y la raspberry pi con el código que se puede observar en el **Anexo A**, además ingresa los datos para el almacenamiento en la base de datos que nos permitirá presentar en una página para monitorear.

```

s71200.py x prueba.py x sqttest.py x untitled.php x pruebapaho.py x
1  from time import sleep
2  import paho.mqtt.publish as publish
3  import snap7
4  from snap7.util import *
5  import struct
6  import time
7  import mysql.connector
8  plc = snap7.client.Client()
9  plc.connect("192.168.1.3",0,1)
10 area=0x82
11 area1=0x83
12 start=0
13 length=1
14 bit=0
15
16 basdat = mysql.connector.connect(user='root', password='0503029589', host='127.0.0.1' , database='prueba')
17 cursor = basdat.cursor()
18
19 for x in range(0,8):
20     byte=plc.read_area(area,0,start,length)
21     byte1=plc.read_area(area1,0,start,4)
22     print 'Q0.'+str(x),get_bool(byte,0,x)
23     print "Mw1",get_int(byte1,1)
24     d=get_bool(byte,0,x)
25     d1=get_int(byte1,1)
26
27     valor = ("INSERT INTO sensor_temp(temperatura,fecha) VALUES("+str(d1)+",now())")
28     valor1 = ("INSERT INTO Digital_out(Q,estado) VALUES("+str(x)+","+str(d)+")")
29     cursor.execute(valor)
30     cursor.execute(valor1)
31     basdat.commit()
32
33     publish.single("ENVIO",x,hostname="192.168.1.5")
34     publish.single("ENVIO",d1,hostname="192.168.1.5")
35     sleep(1)
36 plc.disconnect()
37 cursor.close()
38 basdat.close()
39

```

Figura 4-5: Código en Python utilizando el protocolo S7

Realizado por: Lucia Gallo. 2018

5.5. Software de prueba Mosquitto para MQTT

PYTHON para MQTT

La biblioteca de Python de Paho surgió porque es la biblioteca de python para MQTT, se ha convertido en python puro, compatible con python 2.7 y 3.x.

La biblioteca implementa una clase de cliente que se puede usar para agregar compatibilidad MQTT a su programa python creando instancias del cliente o heredando con su propia clase. También proporciona algunas funciones de ayuda para que la publicación de mensajes de una toma sea extremadamente fácil.

MQTT es el protocolo M2M del futuro. Es ideal para el mundo de los dispositivos conectados "IoT". Su diseño minimalista lo hace perfecto para sistemas incorporados, teléfonos móviles y otras aplicaciones sensibles a la memoria y al ancho de banda.

Las colas de mensajes proporcionan un protocolo de comunicaciones asíncronas, el remitente y el receptor del mensaje no necesitan interactuar con la cola de mensajes al mismo tiempo. Los mensajes colocados en la cola se almacenan para que el destinatario los recupera o hasta que los mensajes caduquen. MQTT y Mosquitto son de buen uso para aplicaciones sensibles al ancho de banda.

El protocolo MQTT, proporciona métodos ligeros para llevar a cabo mensajes utilizando un modelo de cola de mensajes de publicación / suscripción.

MQTT Publicación / Suscripción.

El protocolo MQTT se basa en el principio de publicar mensajes y suscribirse a temas, o "pub / sub". Varios clientes se conectan con un intermediario y se suscriben a los temas que les interesan. Los clientes también se conectan con el intermediario y publican mensajes en los temas. Muchos clientes pueden suscribirse a los mismos temas y hacer con la información cuando lo deseen. El intermediario y MQTT actúan como una interfaz simple y común para todo lo que se conecta al internet.

Instalación

Para la instalación importando la librería siguiente llamada desde Python:

```
Import paho.mqtt.publish as publish
```

Una opción para la mensajería MQTT en raspberry Pi es simplemente instalar mosquitto que es un software de prueba en donde se envía mensajes con dicho protocolo con estructura mosquitto_pub y mosquitto_sub para comunicarse con el servidor MQTT.

MOSQUITTO

Para alojar un intermediario, el MQTT Mosquitto de código abierto es una excelente opción. Este broker MQTT totalmente compatible también viene con utilidades de línea de comandos para publicar y suscribirse.

Por defecto, mosquitto se ejecuta en el puerto 1883 con el código del servidor ejecutándose en modo detallado, abra dos avisos de comando más. Vamos a usar mosquitto_pub y mosquitto_sub para enviarnos mensajes de prueba.

Ahora que se tiene en funcionamiento el broker mosquitto, se puede comunicar con el dispositivo como se realiza la prueba a continuación en la Figura 5-5 que muestra que el protocolo MQTT se está ejecutando en python enviando los datos del PLC Siemens 1200.

```
Command Prompt - mosquitto_sub -d -h 192.168.1.5 -t "ENVIO"
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Lu Gallo>cd C:\Program Files (x86)\mosquitto

C:\Program Files (x86)\mosquitto>mosquitto_sub -d -h 192.168.1.5 -t "ENVIO"
'mosquitto_sub' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files (x86)\mosquitto>mosquitto_sub -d -h 192.168.1.5 -t "ENVIO"
Client mosqsub/10864-LAPTOP-BV sending CONNECT
Client mosqsub/10864-LAPTOP-BV received CONNACK
Client mosqsub/10864-LAPTOP-BV sending SUBSCRIBE (Mid: 1, Topic: ENVIO, QoS: 0)
Client mosqsub/10864-LAPTOP-BV received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/10864-LAPTOP-BV received PUBLISH (d0, q0, r0, m0, 'ENVIO', ... (4 bytes))
True
Client mosqsub/10864-LAPTOP-BV received PUBLISH (d0, q0, r0, m0, 'ENVIO', ... (2 bytes))
28
Client mosqsub/10864-LAPTOP-BV received PUBLISH (d0, q0, r0, m0, 'ENVIO', ... (2 bytes))
42
Client mosqsub/10864-LAPTOP-BV received PUBLISH (d0, q0, r0, m0, 'ENVIO', ... (4 bytes))
True
Client mosqsub/10864-LAPTOP-BV received PUBLISH (d0, q0, r0, m0, 'ENVIO', ... (2 bytes))
43
Client mosqsub/10864-LAPTOP-BV received PUBLISH (d0, q0, r0, m0, 'ENVIO', ... (5 bytes))
False
Client mosqsub/10864-LAPTOP-BV received PUBLISH (d0, q0, r0, m0, 'ENVIO', ... (2 bytes))
43
Client mosqsub/10864-LAPTOP-BV sending PINGREQ
Client mosqsub/10864-LAPTOP-BV received PINGRESP
Client mosqsub/10864-LAPTOP-BV sending PINGREQ
Client mosqsub/10864-LAPTOP-BV received PINGRESP
Client mosqsub/10864-LAPTOP-BV sending PINGREQ
Client mosqsub/10864-LAPTOP-BV received PINGRESP
```

Figura 5-5: Probando MQTT con Mosquitto

Realizado por: Lucia Gallo. 2018

5.6. Base de datos en PhpMyAdmin

PhpMyAdmin es una aplicación PHP que permite gestionar servidores de bases de datos MySQL o MariaDB desde cualquier navegador Web. Es una herramienta ideal para permitir a desarrolladores el acceso y creación de bases de datos para una aplicación Web, ya que provee una vista conveniente y práctica a un motor de bases de datos MySQL, especialmente a aquellos usuarios no familiarizados con el cliente de línea de comandos mysql.

SERVIDOR WEB – phpMyAdmin

Para administrar una base de datos MySQL lo más sencillo es utilizar phpMyAdmin, pues tiene una interfaz web con mucha ayuda y fácil de utilizar el enlace siguiente corresponde a la librería que se importa a Python para la instalación.

```
sudo apt-get install libapache2-mod-auth-mysql php5-mysql phpMyAdmin
```

Durante la instalación se preguntará qué tipo de servidor se tiene, se selecciona Apache, después se confirma si se quiere configurar la base de datos con un 'Ok', preguntara por la contraseña de MySQL (previamente configurada) y a continuación por una contraseña para phpMyAdmin (recomendable que sea diferente, pero si es para pruebas se puede usar la misma). Al finalizar se edita el siguiente fichero:

```
sudo nano /etc/php5/apache2/php.ini
```

El siguiente código se puede escribir en cualquier sitio, pero para ser coherentes bajaremos unos dos tercios de la web hasta encontrar la sección "Dynamic Extensions" y escribiremos en una nueva línea después del título:

```
extension=mysql.so
```

Se guarda el fichero con los cambios efectuados y se visita la siguiente web (usando la IP que tenga la raspberry pi): <http://192.168.1.4/phpmyadmin/>

En la web del phpMyAdmin el usuario será root y la contraseña la que se haya elegido cuando se instala.

Creación de tablas en phpMyAdmin

Las entradas y salidas que se puede leer en el PLC Siemens 1200 son de tipo analógica y digital se creó dos tablas en PhpMyAdmin en este caso Digital_out y sensor_temp que almacenan los datos y estados del sistema, adicional se configuró una tercera base de datos denominada "Usuarios" en donde se manejan la validación del sistema.

Para la primera tabla de la base de datos denominada 'Digital_out', se crea tres columnas: la primera de nombre 'ID' que es la numeración ordenada de los datos, la segunda de nombre 'Q' que representa las 7 salidas digitales del PLC Siemens 1200 y se van incrementado de acuerdo a como se va entregando los datos por el sistema y la última columna de nombre 'Estado' que muestra el estado de las salidas de PLC siendo '1' el de encendido o habilitado y '0' el de apagado o deshabilitado, los datos se van almacenado en orden como se observan en el **Anexo B**, y la estructura se ve a continuación en la Figura 6-5:

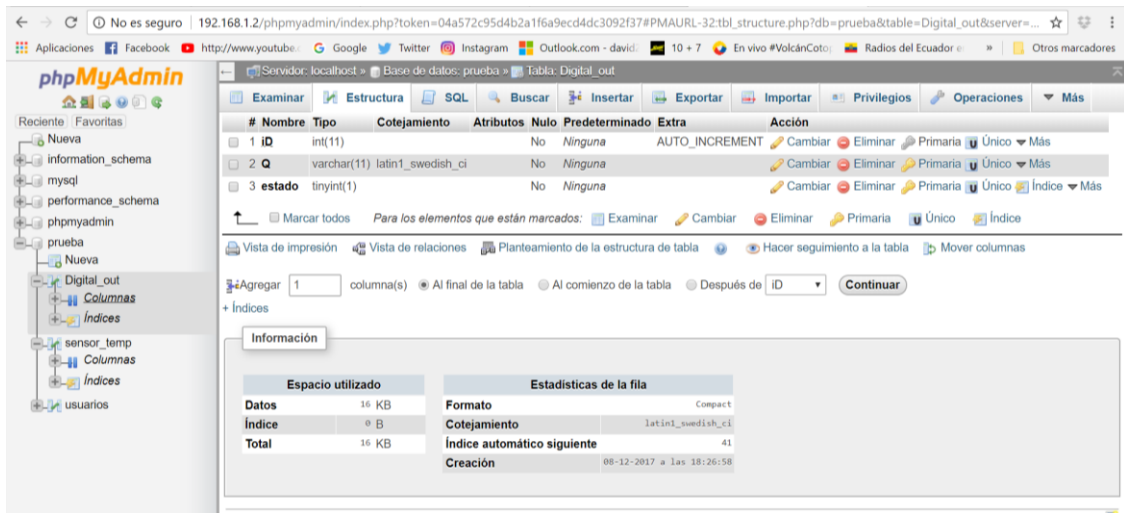


Figura 6-5: Base de datos para almacenar las salidas del PLC.

Realizado por: Lucia Gallo.2018

Para la segunda tabla de la base de datos denominada 'sensor_temp', se crea tres columnas: la primera de nombre 'ID' que es la numeración ordenada de los datos, la segunda de nombre 'temperatura' que simula la temperatura de un sensor colocado en una de las entradas analógicas del PLC Siemens 1200 y la última columna de nombre 'Fecha' que muestra la fecha y hora que se va adquiriendo los datos que se van almacenado en orden como se observan en el **Anexo B**, y la estructura se ve a continuación en la Figura 7-5:

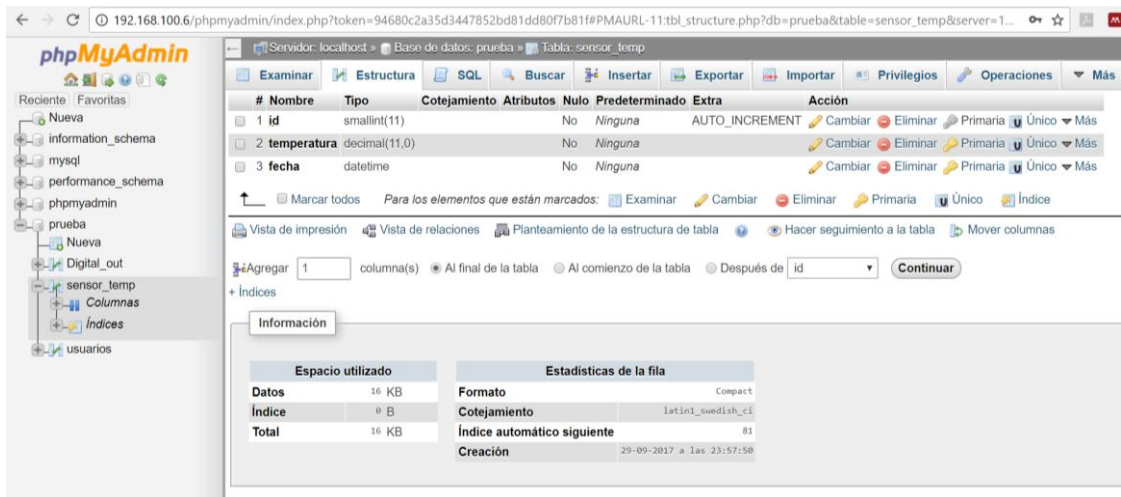


Figura 7-5: Base de datos para almacenar las entradas del PLC.

Realizado por: Lucia Gallo.2018

La seguridad del sistema esta dada por una tercera base de datos la cual contiene nombres de usuarios que pueden acceder al sistema de monitoreo de los datos y estados del PLC Siemens 1200, la estructura de la tabla se muestra en la Figura 8-5:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
1	idusuario	int(11)			No	Ninguna	AUTO_INCREMENT	Cambiar Eliminar Primaria Único Más
2	usuario	varchar(20)	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Primaria Único Más
3	password	varchar(10)	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Primaria Único Más

Espacio utilizado		Estadísticas de la fila	
Datos	16 KB	Formato	Compact
Índice	0 B	Cotejamiento	latin1_swedish_ci
Total	16 KB	Índice automático siguiente	3
		Creación	30-10-2017 a las 12:44:00

Figura 8-5: Base de datos de los usuarios para acceder al sistema.

Realizado por: Lucía Gallo.2018

5.7. *Página Web en PHP*

SERVIDOR WEB – PHP

PHP es un programador para servidores que tiene la capacidad de poder ejecutar contenido dinámico.

En cualquier servidor se podrá entrar con la siguiente dirección: <http://192.168.1.4/phpinfo.php> (usar la IP que corresponda) se ve una página web con la información que se necesita para la realización de las páginas web que se utiliza para la monitorización de los datos del PLC Siemens 1200.

La implementación del proyecto se realizo en el laboratorio y las evidencias se muestran en el **Anexo D**.

CONCLUSIONES

A continuación, se presenta las conclusiones obtenidas durante el desarrollo del proyecto, la cual expone la implementación del Internet Industrial de las cosas aplicado para el laboratorio de PLC's de la facultad de Ingeniería Industrial en Procesos de Automatización de la Universidad Técnica de Ambato.

- Se establece la implementación del proyecto que se llevó a cabo de una manera experimental, el monitoreo de las entradas analógicas y salidas digitales del PLC Siemens 1200 mediante la comunicación con la Raspberry pi, quien se encargó del almacenamiento de los datos mediante una base de datos los mismo que se mostraron en las páginas web, estableciendo así un acceso de manera virtual para visualizar los datos con un monitoreo remoto para los dispositivos inteligentes que se pueden conectar a la red establecida por el router del sistema.
- Se ejecutó la aplicación que permitió monitorear las entradas analógicas y salidas digitales del PLC Siemens 1200, basándose en un diseño genérico y extendible para eso se realizó una revisión de las plataformas para el desarrollo del Internet Industrial de las cosas, así como las compatibilidades de hardware y software conducentes a una integración.
- El proyecto alcanzó a integrar la arquitectura propuesta en el concepto de industria 4.0 y el IIoT, con protocolos del Internet Industrial de las cosas denominado el MQTT, en la mayoría de los casos resulta más económico implementar la tecnología y la arquitectura de forma simultánea en comparación a cuando se implementa la tecnología a una arquitectura ya establecida.
- De acuerdo con los resultados estadísticos del envío de paquetes de datos utilizando el protocolo del Internet Industrial de las cosas denominado MQTT, se analizó la latencia de la red comprobando que la empresa HiveMQ posee 0,54 milisegundos frente al sistema con 0,6 milisegundos, el rendimiento y retardo de la red en el proyecto está por debajo de los valores de las figuras 19-4 y 20-4, que tienen la información de latencia que se viene manejando de acuerdo al medio físico en fibra óptica que se maneja en un rango de 24 milisegundos y 200 milisegundos, con un promedio de 112 milisegundos concluyendo que la latencia para los datos del proyecto es mínima y que no está en el rango esto se debe a que los datos son estados y temperaturas en un tamaño de bytes, probando así que el protocolo MQTT ocupa una mínimo ancho de banda y que la latencia es buena debido a esta característica.

RECOMENDACIONES

- Para obtener mejores resultados en tanto a la comunicación del sistema se debe cargar adecuadamente las librerías en el programador de Python esto debido a que algunas versiones no son completas.
- En este proyecto se necesita 32 Gb de capacidad mínima de la tarjeta de memoria para la Raspberry o se recomienda una de mayor capacidad.
- Al momento de utilizar los equipos tomar las seguridades necesarias ya que se debe tener en cuenta cuales son las fuentes de alimentación necesarias, que corrientes maneja cada equipo y proceder con cuidado cuando se manipula la tarjeta de entradas analógicas.
- Cuando se necesite reensamblar la aplicación SNAP7 tener cuidado ya que se necesita modifica archivos del sistema de la Raspberry.
- Esta solución se podría implementar cuando se tiene recursos limitados y en procesos de difícil acceso o remotos para el operador ya que el consumo de energía es bajo.

BIBLIOGRAFÍA

- Astarloa, Bidarte, Jimenez, Zuloaga, & Lazaro,** (2016) Astarloa, A., Bidarte, U., Jimenez, J., Zuloaga, A., & Lazaro, J. (2016). Intelligent gateway for industry 4.0-compliant production. *IECON Proceedings (Industrial Electronics Conference)*, 4902–4907. <https://doi.org/10.1109/IECON.2016.7793890>
- Figuerola, N.** (2014). Internet de las cosas. *PMQuality*, 1–27. <https://doi.org/10.1016/j.cirp.2011.03.145>
- Ghanbari, A., Laya, A., & Markendahl, J.** (2016). Value Creation and Coopetition in M2M Ecosystem - The Case of Smart City.
- Github.** (2014). Simply Automationized: Raspberry Pi getting data from a S7-1200 PLC. Retrieved December 12, 2017, from <http://simplyautomationized.blogspot.com/2014/12/raspberry-pi-getting-data-from-s7-1200.html>
- Gómez, L.** (2017). Internet de las Cosas. Privacidad y Seguridad., 10_114. Retrieved from <http://di002.edv.uniovi.es/~cueva/investigacion/tesis/Tesis-JoseIgnacio.pdf>
- HIVEMQ.** (2012). MQTT Dashboard. Retrieved May 24, 2018, from <http://www.mqtt-dashboard.com/index.html>
- Humano, D. E. C., Enfoque, U. N., Autor, A., Szabo, A., Titulaci, B., Econom, G. E. N., ... Mesonada, J.** (2012). Trabajo final de grado, 1–43.
- Jayaram, A.** (2016). Lean six sigma approach for global supply chain management using industry 4.0 and IIoT. *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 89–94. <https://doi.org/10.1109/IC3I.2016.7917940>
- Meng, Z., Wu, Z., Member, S., Muvianto, C., & Gray, J.** (2016). A Data-Oriented M2M Messaging Mechanism for Industrial IoT Applications, 4662(c). <https://doi.org/10.1109/JIOT.2016.2646375>
- Osathanukul, K., Hantrakul, K., Pramokchon, P., Khoenkaw, P., & Tantitharanukul, N.** (2017). Configurable Automatic Smart Urinal Flusher based on MQTT Protocol.
- Python-snap7.** (2013). Cliente - documentación de python-snap7 0.0rc0. Retrieved December 13, 2017, from <http://python-snap7.readthedocs.io/en/latest/client.html#snap7.client.Client>
- Rong, W., Vanan, G. T., & Phillips, M.** (2016). The internet of things (IoT) and transformation of the smart factory. *2016 International Electronics Symposium (IES)*, 399–402. <https://doi.org/10.1109/ELECSYM.2016.7861039>
- Sanz Javier, Claudio Valero, & David Gómez Bolaños.** (2016). Qué es la latencia, y cómo podemos mejorarla. Retrieved October 10, 2018, from <https://www.testvelocidad.es/2016/08/18/la-latencia-podemos-mejorarla/>

- Sanz Jose Luis.** (2005). ADSLZone - Portal sobre ADSL, fibra óptica y tecnología. Retrieved October 16, 2018, from <https://www.adslzone.net/>
- SASE.** (2015). Tutoriales del SASE2015 | Simposio Argentino de Sistemas Embebidos 2015. Retrieved February 15, 2018, from <http://www.sase.com.ar/2015/tutoriales2015/>
- Semle, P. A.** (2016). Protocolos IIoT para considerar, 32–35.
- Thota, P., & Kim, Y.** (2016). Implementation and Comparison of M2M Protocols for Internet of Things. <https://doi.org/10.1109/ACIT-CSII-BCD.2016.20>
- Torres.** (2014). ¿Qué es y cómo funciona el Internet de las cosas? Retrieved January 12, 2018, from <https://hipertextual.com/archivo/2014/10/internet-cosas/>
- Ventura Victor.** (2017). conceptos - polaridad.es. Retrieved January 16, 2018, from <https://polaridad.es/category/conceptos/>
- Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., & Vasilakos, A. V.** (2016). Software-Defined Industrial Internet of Things in the Context of Industry 4.0. *IEEE Sensors Journal*, 16(20), 7373–7380. <https://doi.org/10.1109/JSEN.2016.2565621>

ANEXOS

ANEXO A.- Código de la comunicación Snap7 cargada en la Raspberry pi.

```
1  """
2  Snap7 client used for connection to a siemens7 server.
3  """
4  import re
5  from ctypes import c_int, c_char_p, byref, sizeof, c_uint16, c_int32, c_byte
6  from ctypes import c_void_p
7
8  import logging
9
10 import snap7
11 from snap7 import six
12 from snap7.snap7types import S7Object, buffer_type, buffer_size, BlocksList
13 from snap7.snap7types import TS7BlockInfo, param_types, cpu_statuses
14
15 from snap7.common import check_error, load_library, ipv4
16 from snap7.snap7exceptions import Snap7Exception
17
18 logger = logging.getLogger(__name__)
19
20
21 def error_wrap(func):
22     """Parses a s7 error code returned the decorated function."""
23     def f(*args, **kw):
24         code = func(*args, **kw)
25         check_error(code, context="client")
26     return f
27
28
29 class Client(object):
30     """
31     A snap7 client
32     """
33     def __init__(self):
34         self.library = load_library()
35         self.pointer = False
36         self.create()
37
38     def create(self):
39         """
40         create a SNAP7 client.
41         """
42         logger.info("creating snap7 client")
43         self.library.Cli_Create.restype = c_void_p
44         self.pointer = S7Object(self.library.Cli_Create())
45
46     def destroy(self):
47         """
48         destroy a client.
49         """
50         logger.info("destroying snap7 client")
51         return self.library.Cli_Destroy(byref(self.pointer))
52
53     def plc_stop(self):
54         """
55         stops a client
56         """
57         logger.info("stopping plc")
58         return self.library.Cli_PlcStop(self.pointer)
59
60 ..
```

```

60     def plc_cold_start(self):
61         """
62         cold starts a client
63         """
64         logger.info("cold starting plc")
65         return self.library.Cli_PlcColdStart(self.pointer)
66
67     def plc_hot_start(self):
68         """
69         hot starts a client
70         """
71         logger.info("hot starting plc")
72         return self.library.Cli_PlcColdStart(self.pointer)
73
74     def get_cpu_state(self):
75         """
76         Retrieves CPU state from client
77         """
78         state = c_int(0)
79         self.library.Cli_GetPlcStatus(self.pointer, byref(state))
80
81         try:
82             status_string = cpu_statuses[state.value]
83         except KeyError:
84             status_string = None
85
86         if not status_string:
87             raise Snap7Exception("The cpu state (%s) is invalid" % state.value)
88
89         logging.debug("CPU state is %s" % status_string)
90         return status_string
91
92

```

```

92     @error_wrap
93     def disconnect(self):
94         """
95         disconnect a client.
96         """
97         logger.info("disconnecting snap7 client")
98         return self.library.Cli_Disconnect(self.pointer)
99
100    @error_wrap
101    def connect(self, address, rack, slot, tcpport=102):
102        """
103        Connect to a S7 server.
104
105        :param address: IP address of server
106        :param rack: rack on server
107        :param slot: slot on server.
108        """
109        logger.info("connecting to %s:%s rack %s slot %s" % (address, tcpport,
110                                                            rack, slot))
111
112        self.set_param(snap7.snap7types.RemotePort, tcpport)
113        return self.library.Cli_ConnectTo(
114            self.pointer, c_char_p(six.b(address)),
115            c_int(rack), c_int(slot))
116
117    def connect(self):
118        """To connect to these PLC you need to call set_connection_params() and then
119        connect(). S7-200/Logo PLCs requiring TSAPs
120        """
121        logger.debug("connecting using set_connection_params")
122        return self.library.Cli_Connect(self.pointer)
123

```

ANEXO B

Creación de la base de datos en el programa phpMyAdmin.

The screenshot shows the phpMyAdmin interface for the 'prueba' database. The 'Digital_out' table is selected, and the 'Estructura' (Structure) tab is active. The table structure is displayed as follows:

id	Q	estado
25	0	1
26	1	1
27	2	0
28	3	0
29	4	1
30	5	1
31	6	1
32	7	0
33	0	1
34	1	1
35	2	0
36	3	0

The screenshot shows the phpMyAdmin interface for the 'prueba' database. The 'sensor_temp' table is selected, and the 'Estructura' (Structure) tab is active. A message indicates that 24 rows are shown out of a total of 56. The table structure is displayed as follows:

id	temperatura	fecha
25	18	2017-12-13 10:57:25
26	18	2017-12-13 10:57:26
27	18	2017-12-13 10:57:27
28	18	2017-12-13 10:57:29
29	18	2017-12-13 10:57:30
30	18	2017-12-13 10:57:31
31	18	2017-12-13 10:57:32
32	18	2017-12-13 10:57:33
33	19	2017-12-13 11:04:43
34	19	2017-12-13 11:04:44

192.168.100.6/phpmyadmin/index.php?token=94680c2a35d3447852bd81dd807b81f#PMAURL-3:sql.php?db=prueba&table=usuarios&server=1&target=&toke...

Servidor: localhost » Base de datos: prueba » Tabla: usuarios

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Más

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0.0007 segundos.)

```
SELECT * FROM `usuarios`
```

Perfilando [En línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Número de filas: 25 Filtrar filas: Buscar en esta tabla

Ordenar según la clave: Ninguna

Opciones

	idusuario	usuario	password
<input type="checkbox"/>	1	david	david
<input type="checkbox"/>	2	lucia	lucia

↑ Marcar todos Para los elementos que están marcados: Cambiar Borrar Exportar

Número de filas: 25 Filtrar filas: Buscar en esta tabla

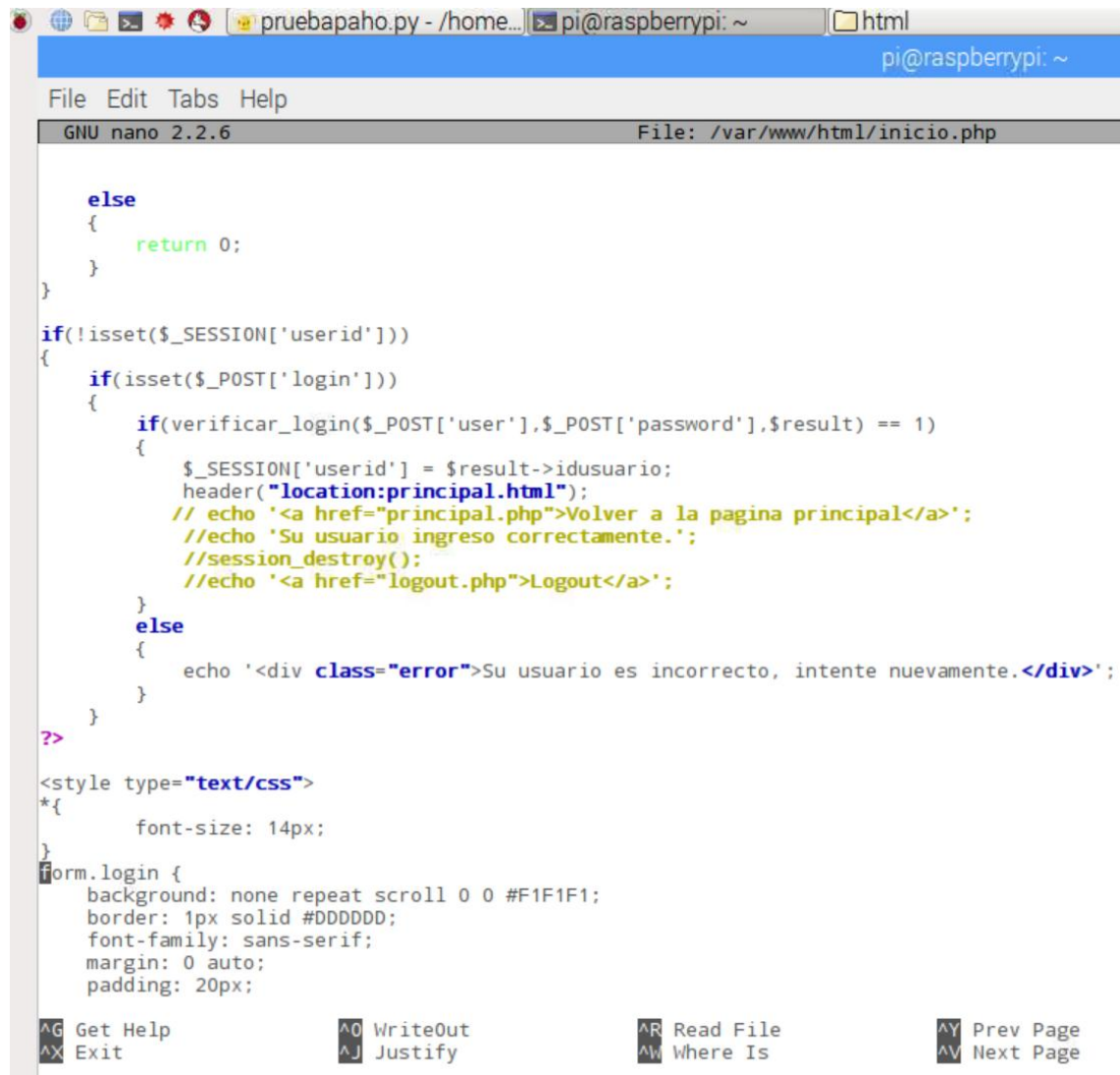
Operaciones sobre los resultados de la consulta

Vista de impresión Previsualización para imprimir (documento completo) Exportar Mostrar gráfico Crear vista

192.168.100.6/phpmyadmin/sql.php?server=1&db=prueba&table=usuarios&nos=0&token=94680c2a35d3447852bd81dd807b81f

ANEXO C

Programación de las páginas web utilizando el software PHP en la plataforma Python.



```
pruebapaho.py - /home... pi@raspberrypi: ~ | html
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /var/www/html/inicio.php

    else
    {
        return 0;
    }
}

if(!isset($_SESSION['userid']))
{
    if(isset($_POST['login']))
    {
        if(verificar_login($_POST['user'],$_POST['password'],$result) == 1)
        {
            $_SESSION['userid'] = $result->idusuario;
            header("location:principal.html");
            // echo '<a href="principal.php">Volver a la pagina principal</a>';
            //echo 'Su usuario ingreso correctamente.';
            //session_destroy();
            //echo '<a href="logout.php">Logout</a>';
        }
        else
        {
            echo '<div class="error">Su usuario es incorrecto, intente nuevamente.</div>';
        }
    }
}

?>

<style type="text/css">
*{
    font-size: 14px;
}
form.login {
background: none repeat scroll 0 0 #F1F1F1;
border: 1px solid #DDDDDD;
font-family: sans-serif;
margin: 0 auto;
padding: 20px;
}

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page
^X Exit          ^J Justify      ^W Where Is     ^V Next Page
```

```
pruebapaho.py - /home... pi@raspberrypi: ~ html
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /var/www/html/inicio.php
<?php
session_start();
include_once "conexion.php";

function verificar_login ($user,$password,&$result) {
    $sql = "SELECT * FROM usuarios WHERE usuario = '$user' and password = '$password'";
    $rec = mysql_query($sql);
    $count = 0;

    while($row = mysql_fetch_object($rec))
    {
        $count++;
        $result = $row;
    }

    if($count == 1)
    {
        return 1;
    }

    else
    {
        return 0;
    }
}

if(!isset($_SESSION['userid']))
{
    if(isset($_POST['login']))
    {
        if(verificar_login($_POST['user'],$_POST['password'],$result) == 1)
        {
            $_SESSION['userid'] = $result->idusuario;
            header("location:principal.html");
            // echo '<a href="principal.php">Volver a la pagina principal</a>';
            //echo 'Su usuario ingreso correctamente.';
            //session_destroy();
        }
    }
}

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page
^X Exit          ^J Justify     ^W Where Is    ^V Next Page
```

```
pruebasapaho.py - /home... pi@raspberrypi: ~ | html
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /var/www/html/inicio.php

}
form.login div {
  margin-bottom: 15px;
  overflow: hidden;
}
form.login div label {
  display: block;
  float: left;
  line-height: 25px;
}
form.login div input[type="text"], form.login div input[type="password"] {
  border: 1px solid #DCDCDC;
  float: right;
  padding: 4px;
}
form.login div input[type="submit"] {
  background: none repeat scroll 0 0 #DEDEDE;
  border: 1px solid #C6C6C6;
  float: right;
  font-weight: bold;
  padding: 4px 20px;
}
.error{
  color: red;
  font-weight: bold;
  margin: 10px;
  text-align: center;
}
</style>

<form action="" method="post" class="login">
  <div><label>Username</label><input name="user" type="text" ></div>
  <div><label>Password</label><input name="password" type="password"></div>
  <div><input name="login" type="submit" value="login"></div>
</form>
<?php
} else {

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page
^X Exit          ^J Justify      ^W Where Is    ^V Next Page
```



```
[pruebapaho.py - /hom...] pi@raspberrypi: ~ [html]
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /var/www/html/start.php

?php
// Conectando, seleccionando la base de datos
$link = mysql_connect('127.0.0.1', 'root', '0503029589')
    or die('No se pudo conectar: ' . mysql_error());
echo 'Connected successfully';
print "\n";
mysql_select_db('prueba') or die('No se pudo seleccionar la base de datos');

// Realizar una consulta MySQL
$query = 'SELECT * FROM sensor_temp';
// $query = 'ALTER TABLE Digital_out AUTO_INCREMENT = 0';
$result = mysql_query("SELECT id, Q, estado FROM Digital_out", $link) or die('Consulta fallida: ' . mysql_error());

// Imprimir los resultados en HTML
//echo "<table>\n";
//echo "<table border = '1'> \n";
//echo "<tr><td>ID</td> <td>Temperatura</td> <td>fecha</td> </tr> \n";
//echo "\n";
//print "<tr>\n";
//echo 'Su usuario ingreso correctamente.';
//echo '<a href="logout.php">Logout</a>';
echo '<a href="principal.html">Volver a la pagina principal</a>';
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    echo "<table border = '1'> \n";
    echo "<tr><td>ID</td> <td>Q.</td> <td>estado</td> </tr> \n";
    echo "<META HTTP-EQUIV='refresh' CONTENT='1'>";
    //echo "<tr><td>".$row["id"]."</td><td>".$row["temperatura"]."</td><td>".$row["fecha"]."</td></tr> \n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";
echo '<a href="logout.php">logout</a>';
//echo '<output type="button" onclick="location=inicio.php" a href="logout.php">Logout</a>';
//echo "
```

[Read 49 lines]

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text

```
[pruebapaho.py - /hom... pi@raspberrypi: ~ [html]
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /var/www/html/start.php

//echo 'Su usuario ingreso correctamente.';
//echo '<a href="logout.php">Logout</a>';
echo '<a href="principal.html">Volver a la pagina principal</a>';
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    echo "<table border ='1'> \n";
    echo "<tr><td>ID</td> <td>Q0.</td> <td>estado</td> </tr> \n";
    echo "<META HTTP-EQUIV='refresh' CONTENT='1'>";
    //echo "<tr><td>".$row["id"]."</td><td>".$row["temperatura"]."</td><td>".$row["fecha"]."</td></tr> \n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";
echo '<a href="logout.php">logout</a>';
//echo '<output type="button" onclick="location=inicio.php" a href="logout.php">Logout</a>';
//echo "
//    <input type="button"
//    onclick="location=inicio.php"
//    value="Retornar"
//    name="Retornar">
// "
//<input type="button" value="buscar" onclick="location=inicio.php"/>
// Liberar resultados
mysql_free_result($result);

// Cerrar la conexión
mysql_close($link);
?>
█

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page   ^K Cut
^X Exit          ^J Justify      ^W Where Is     ^V Next Page   ^U UnCu
```

```
pruebapaho.py - /home... pi@raspberrypi: ~
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /var/www/html/prueba2.php

<?php
// Conectando, seleccionando la base de datos
$link = mysql_connect('127.0.0.1', 'root', '0503029589')
    or die('No se pudo conectar: ' . mysql_error());
echo 'Connected successfully';
print "\n";
mysql_select_db('prueba') or die('No se pudo seleccionar la base de datos');

// Realizar una consulta MySQL
$query = 'SELECT * FROM sensor_temp';
$result = mysql_query("$query", $link) or die('Consulta fallida: ' . mysql_error());

// Imprimir los resultados en HTML
//echo "<table>\n";
//echo "<table border='1'> \n";
//echo "<tr><td>ID</td> <td>Temperatura</td> <td>fecha</td> </tr> \n";
//echo "\n"
print "<tr>\n";
//echo 'Su usuario ingreso correctamente.';
//echo '<a href="logout.php">Logout</a>';
echo '<a href="principal.html">Volver a la pagina principal</a>';
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    echo "<table border='1'> \n";
    echo "<tr><td>ID</td> <td>Temperatura</td> <td>fecha</td> </tr> \n";
    echo "<META HTTP-EQUIV='refresh' CONTENT='1'>";
    //echo "<tr><td>".$row["id"]."</td><td>".$row["temperatura"]."</td><td>".$row["fecha"]."</td></tr> \n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";

// Liberar resultados
mysql_free_result($result);

[ Read 40 lines ]
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text
^X Exit          ^J Justify     ^W Where Is    ^V Next Page    ^U UnCut Text
```

```
pruebapaho.py - /home... pi@raspberrypi: ~
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /var/www/html/prueba2.php

//echo '<a href="logout.php">Logout</a>';
echo '<a href="principal.html">Volver a la pagina principal</a>';
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    echo "<table border ='1'> \n";
    echo "<tr><td>ID</td> <td>Temperatura</td> <td>fecha</td> </tr> \n";
    echo "<META HTTP-EQUIV='refresh' CONTENT='1'>";
    //echo "<tr><td>".$row["id"]."</td><td>".$row["temperatura"]."</td><td>".$row["fecha"]."</td></tr> \n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";

// Liberar resultados
mysql_free_result($result);

// Cerrar la conexión
mysql_close($link);
?>
█

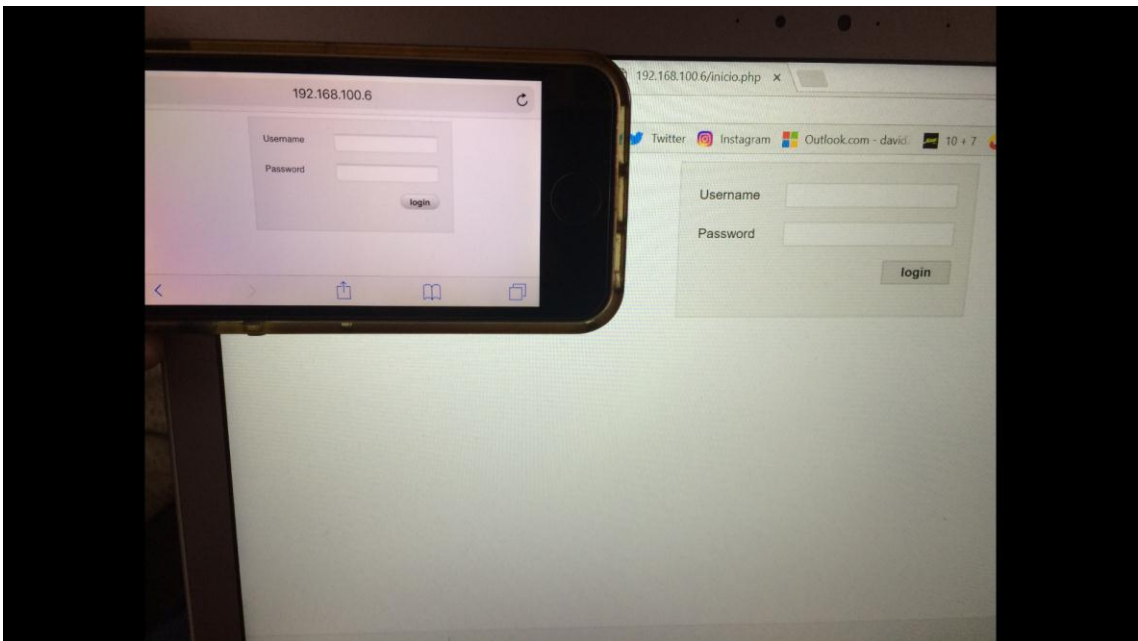
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut
^X Exit          ^J Justify      ^W Where Is    ^V Next Page    ^U UnCu
```

ANEXO D

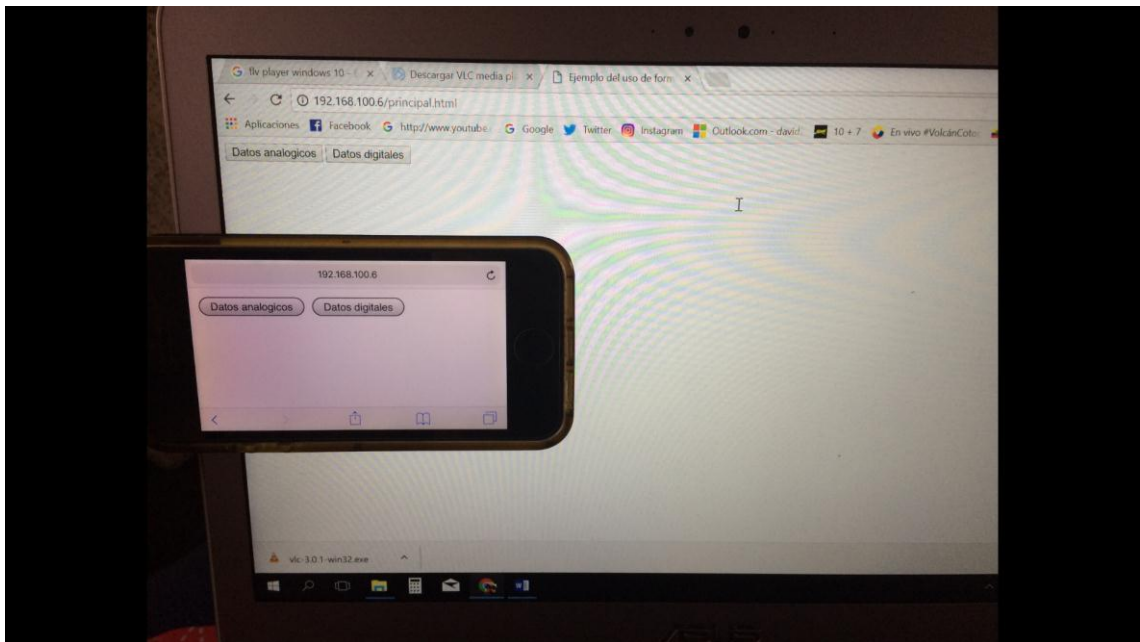
Implementación física del proyecto utilizando la Raspberry y el PLC Siemens 1200.



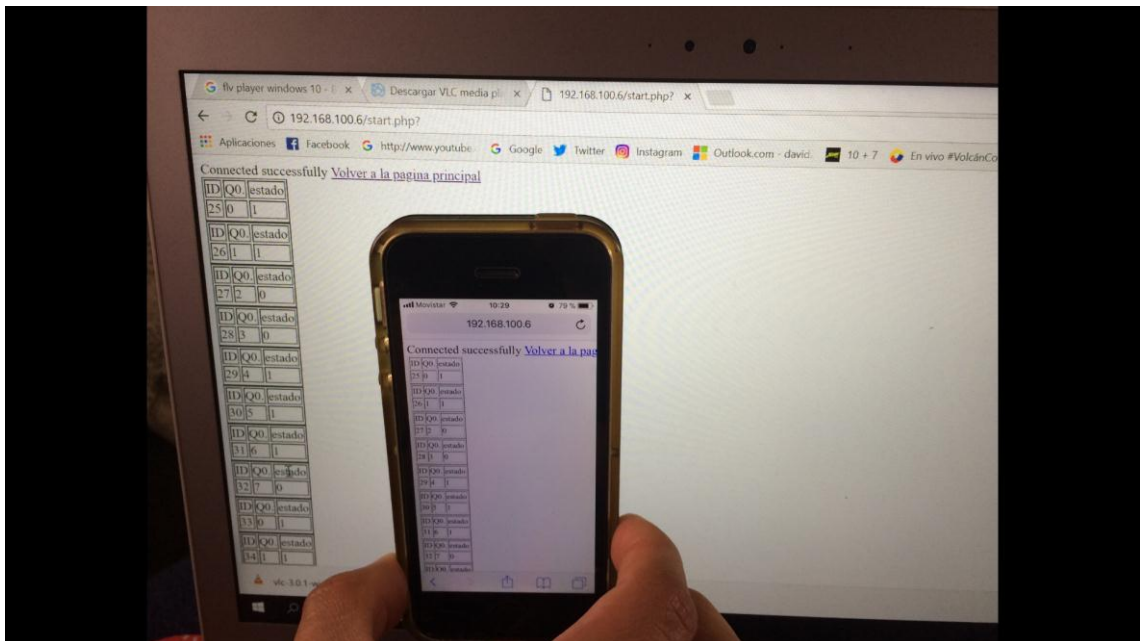
Página web donde se muestra la validación del proyecto, tanto en una computadora como en un smartphone.



Página web donde se muestra la selección de datos del proyecto tanto en una computadora con en un smartphone.



Página web donde se muestra la toma de datos digitales de las salidas digitales del PLC Siemens 1200 en tiempo real del proyecto, tanto en una computadora con en un smartphone.



Página web donde se muestra la toma de un sensor de temperatura colocada en una de las entradas analógicas del PLC Siemens 1200 en tiempo real del proyecto, tanto en una computadora como en un smartphone.

