Laboratorio Nacional de Informática Avanzada A.C.

Sistema IoT para monitoreo de servidores

Fecha de elaboración:	10/01/2022
REPORTE TÉCNICO	
Número de reporte:	1
Versión:	
Autor(es):	Erick Abimael Rodriguez Hernández y José Ángel López Reyes

Declaraciones

Responsable de preparación

Propósito

Patrocinadores

Rébsamen 80
Col. Centro CP 91000
Xalapa, Veracruz
www.lania.mx

Contenido

Lista de Tablas

Lista de ilustraciones

Introducción

El proyecto a desarrollarse continuación propone un dispositivo de monitoreo y configuración de los equipos de aire acondicionado instalados en la sala de servidores de manera automática o remota mediante el acceso a una página web. La iniciativa del proyecto ha sido planteada por el Dr. Juan Manuel Gutiérrez Moreno ante las complicaciones operativas que suele significar al personal del Instituto el accionamiento manual de los equipos ante cada desperfecto en la red de alimentación eléctrica pues, aunque los servidores arrancan de manera automática, los Minispit no cuentan con esta facilidad.

El proyecto realizado está enfocado en la resolución de esta problemática, se busca diseñar un sistema embebido domótico capaz de monitorear temperatura y cortes de corriente para tomar acciones de control automáticas sobre el equipo de climatización. Del mismo modo se implementará un servidor en el sistema para monitoreo y control remoto. A lo largo del documento se expondrán las etapas principales de desarrollo del proyecto y las tecnologías empleadas.

Marco teórico y/o referencial

LIRC

Es un popular paquete de software para distribuciones de LINUX o código abierto que permite decodificar y enviar señales infrarrojas utilizadas en controles remotos. Ofrece mayor flexibilidad y funcionalidad.

Comunicación por medio de tecnología infrarroja

Es un tipo de comunicación inalámbrica que utiliza dos LED´s infrarrojos uno que recibe y otro que transmite información. Difieren en potencia, tipo de lentilla, consumo de energía y frecuencia de operación. La parte más crítica de este sistema es la receptora, ya que tiene que ser capaz de discernir entre la señal a recibir y otros tipos de señales infrarrojas como la del sol.

Ilustración 1. Receptor Infrarrojo

Adafruit Python-DHT

Librería que permite leer los datos digitales proporcionados por un sensor de temperatura y humedad DHT en una placa Raspberry Pi

GPIO

El nombre significa “Entrada / Salida de uso general”, es un tipo de pin en un circuito integrado que no tiene una función específica, esto quiere decir, que es personalizable y puede ser controlada por programa.

Internet de las cosas (IoT)

Consiste en la relación inteligente de personas, procesos, datos y objetos mediante la transformación de la información que crea nuevas funcionalidades, mejores experiencias para personas o empresas permitiendo la convergencia, orquestación y visibilidad a través de sistemas que, anteriormente, eran dispares.

Ilustración 2.Internet de las cosas. Las nuevas tecnologías han tenido un gran impacto en diversos ámbitos de la sociedad

Flask

Es un micro-framework WSGI (Web Server Gataway Interface). Está diseñado para diseñar, desarrollar y simplificar la creación de aplicaciones WEB con el lenguaje Python.

Flask-MQTT

Este software es una extensión de Flask que permite facilitar la integración de un cliente MQTT en una aplicación WEB. Es un protocolo “Machine to machine” enfocado a IoT que permite enviar y recibir mensajes ligeros.

Bases de datos y lenguaje SQL

Las bases de datos recopilan de manera organizada la información o datos de un sistema o proceso y los almacena de forma electrónica en un sistema informático. Se guardan como estructuras de filas y columnas para aumentar la eficacia del procesamiento. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurado llamado SQL para manipular, gestionar, consultar, ordenar y definir los datos.

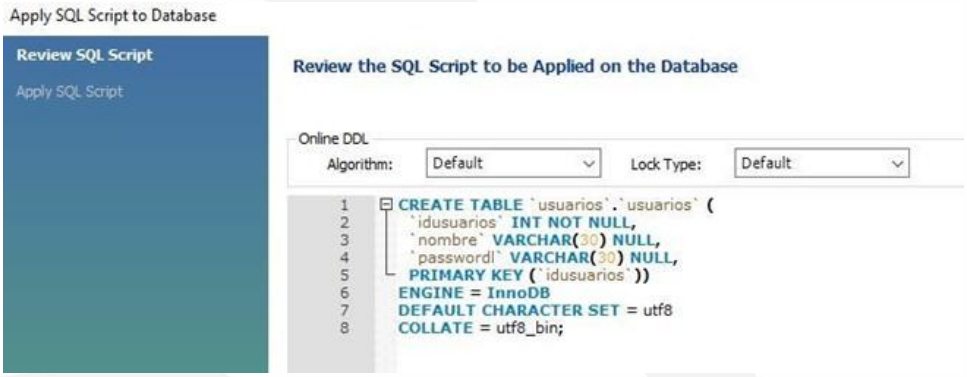


Ilustración 3.Creación de una base de datos.

Python-MySQL

Librería utilizada para insertar datos de un sistema en una base de datos utilizando lenguaje Python y estructurado SQL.

Mosquitto MQTT broker

Es un software de código abierto que implementa el protocolo MQTT, es ligero y conveniente para su use en computadoras de alto rendimiento, así como en servidores. Funciona con un protocolo de mensajes ligeros ideal para aplicaciones que involucren IoT.

Diseño de solución

Decodificación y codificación de señales infrarrojas

Según la documentación oficial de LIRC (https://www.lirc.org), entro de sus funciones se encuentra *mode2* la cual inicializa el modo de recepción y captura de señales infrarrojas. Se capturaron aproximadamente 130 archivos en los que cada uno corresponde a un "click" sobre algún botón. Se buscaba hacer una modificación de una de las configuraciones a la vez (por ejemplo, temperatura) manteniendo las otras constantes (modo, barrido, etc…) para poder diferenciar que bits la representan dentro de la trama general y poder decodificar la información.

Decodificación

Los datos en los archivos capturados representan los microsegundos transcurridos entre un cambio de estado de la señal emitida. La secuencia está compuesta por 4 lapsos de tiempo diferentes: el lapso de referencia indica el inicio y la mitad de la trama (≈4400), base de bit (≈500), indicador de estado de bit en 0(≈590) e indicador de estado de bit en 1 (≈1690). Al unir dos lapsos de tiempo contiguos (base e indicador) se decodifica la secuencia binaria.

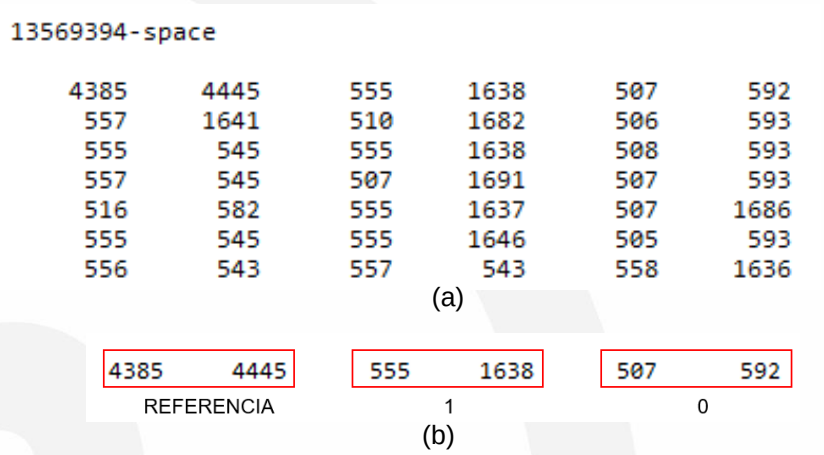


Ilustración 4.Codificación de tiempos en señales IR.
(a) Fragmento de un archivo capturado con LIRC
(b) Interpretación binaria del primer renglón de (a)

Utilizando Python se programó un algoritmo para la decodificación de todos los archivos capturados y se generó un listado de los códigos para su interpretación. La primera característica a denotarse es que a lo largo de la trama de bits en cada instrucción realmente se envina una secuencia principal de bits dos veces, el inicio y fin de cada secuencia esta delimitada por los periodos de tiempo de referencia, dichos periodos están representados por corchetes. Esto significa que, principalmente solo es necesario construir la mitad de la trama y replicarla para obtener el resto.

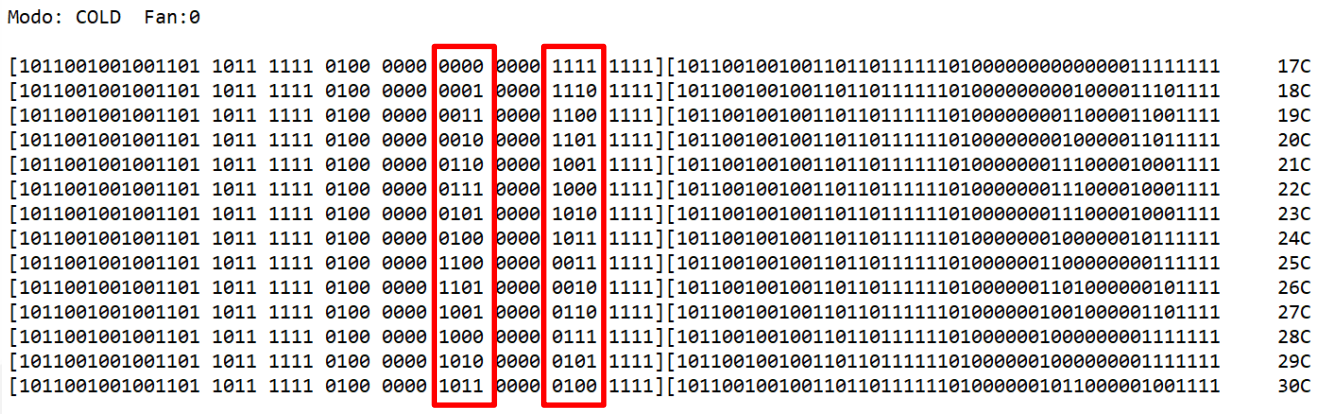


Ilustración 5.Códigos binarios de instrucciones variando solo la temperatura. La trama permanece constante con excepción de los bits señalados en rojo, también es apreciable como se repte la trama después de los corchetes intermedios

Análisis

El equipo utiliza patrones de codificación diferentes y para los requerimientos del proyecto solo se han interpretado 4 de ellos (se ha prescindido de la decodificación para los comandos de temporizadores pues no se consideraron relevantes). Descartando las referencias (representadas por corchetes) las tramas estaban compuestas por cadenas de 96 o 144 bits donde cada instrucción estaba codificada en paquetes de 4 bits.

- Apagar

La orden de 'OFF' es una cadena particular de 96 bits más referencias (corchetes). La acción de encendido se ejecuta con un código diferente.

[1011001001001101011111011100001001110000000011111][1011001001001101011111011100001001110000000011111

- SWING

Swing activa el barrido de la salida de aire del mini Split. Del mismo modo que 'OFF', swing cuenta con un código particular de 96 bits. En este caso la instrucción enciende o apaga esta funcionalidad según sea el caso.

[1011001001001101011010111001010011100000000011111][1011001001001101011010111001010011100000000011111

- Configuración de parámetros y encendido

La estructura de codificación principal configura: modo, temperatura y velocidad de ventilador, en caso de estar apagado el equipo se encenderá con los parámetros enviados. Cada configuración esta codificada en paquetes de 4 bits, a continuación, se muestra un desglose de la trama completa:

[1011001001001101 1011 1111 0100 0000 0010 1100 1101 0011][10110010010011010111111010000000010110011010011
A B C D E F G H I J

A.- secuencia de bits fija
B.- Velocidad de ventilador
C.- Bits fijos en 1
D.-Complemento de Velocidad de ventilador

E.- Bits fijos en 0
F.- Temperatura
G.- Modo
H.- Complemento de Temperatura
I.- Complemento de Modo
J.- Repetición de la cadena anterior de bits (A-I)

Tabla1. Códigos de temperatura para construcción de trama binaria

Temperatura (°C)	Código	Complemento
17	0000	1111
18	0001	1110
19	0011	1100
20	0010	1101
21	0110	1001
22	0111	1000
23	0101	1010
24	0100	1011
25	1100	0011
26	1101	0010
27	1001	0110
28	1000	0111
29	1010	0101
30	1011	0100

Tabla2. Códigos de modo para construcción de trama binaria

Modo	Código	Complemento
AUTO	1000	0111
DRY	0100	1011
COLD	0000	1111
HEAT	1100	0011

Tabla2. Códigos de velocidad de ventilador para construcción de trama binaria

Velocidad de Ventilador	Código	Complemento
1	1011	0100
2	1001	0110
3	0101	1010
4	0011	1100

- Sleep

Para esta instrucción únicamente se agrega una cadena de bits como base antes de la codificación de configuración de parámetros (Punto 3). La estructura de base no cambia para la habilitación o des habilitación de esta función si no que alternara entre ellas según sea el caso.

[10110010010011011110000000011111000000111111110 0] [10110010010 01101000111111110000000111000110 0111][10110010010011010001111111100000001110001100 0111
Base de 'Sleep' **Estructura de parámetros**

Codificación y emisión de comandos

Mediante un algoritmo desarrollado en Python se construyen las tramas de bits las tablas de codificación y las instrucciones deseadas para posteriormente reinterpretarlas en el listado de lapsos de tiempos de encendido y apagado del emisor infrarrojo. Con dicho listado se genera un archivo necesario para el uso de LIRC. LIRC no cuenta con una librería de Python enfocada a la ejecución de sus funciones desde este lenguaje de programación, debe ser ejecutado desde terminal por lo que se ha utilizado la librería 'OS' para ejecutar comandos a terminal desde Python.

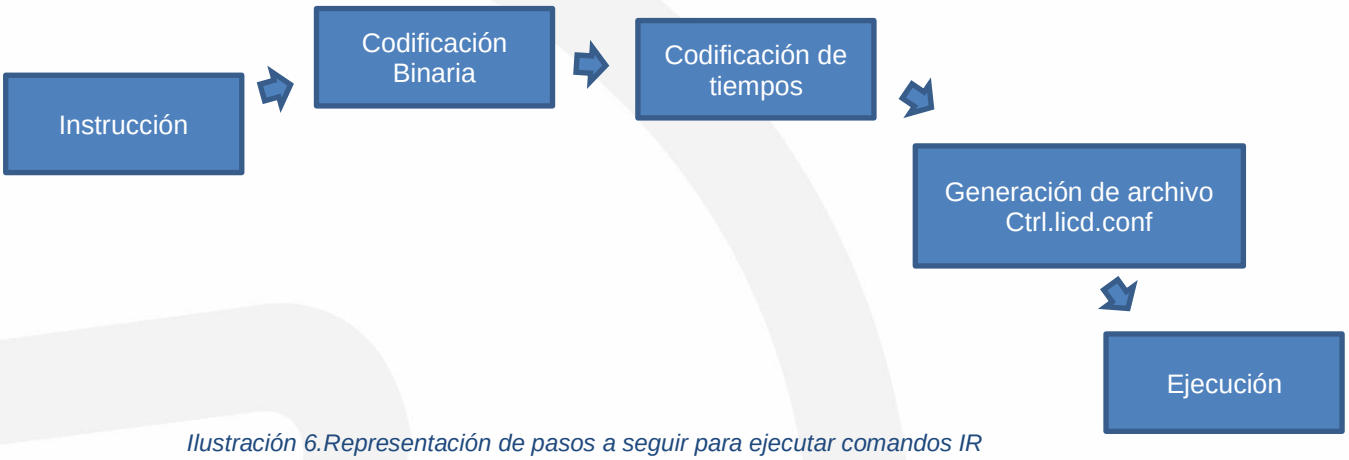


Ilustración 6.Representación de pasos a seguir para ejecutar comandos IR

Sensor de temperatura

Analizando las necesidades del proyecto se optó por el sensor DHT11 el cual mide temperatura y humedad. Este sensor cuenta con una precisión adecuada y compatibilidad con Raspberry al devolver valores digitales. El manejo de este sensor con Python es relativamente sencillo mediante la librería "Adafruit dht library" (https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/python-setup) mientras que las conexiones físicas son sencillas al operar con 3.3V y ser compatible con cualquiera de los puertos GPIO.

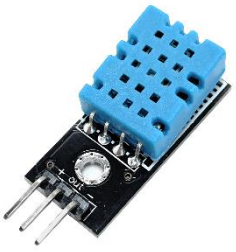


Ilustración 7.Sensor DHT11.

Sensor de Cortes de corriente Alterna

Para el proyecto se ha propuesto una implementación con el Sensor Sct-013, Este es un sensor de corriente por inducción electromagnética capaz de medir corrientes hasta de 30A (existen modelos de mayor Amperaje). El sensor cuenta con la ventaja de ser una alternativa no invasiva al circuito a monitorear por lo cual es más seguro y practico.

El tipo de señal devuelto por el transductor es de naturaleza analógica (0-1V) lo cual trae complicaciones al utilizarlo con Raspberry pues no cuenta con conversor analógico a digita (ADC) incorporado. Para solventar esta situación se ha diseñado un circuito acondicionador con comparadores que devuelve un valor binario si existe o no un flujo de corriente, esta alternativa nos impide conocer la corriente especifi ca que circula por el circuito sin embargo sí nos permite saber si existe algún corte en el suministro. No obstante, el acondicionador también cuenta con una salida amplificada al doble para facilitar su lectura en caso de ser necesario recibir el valor analógico en una modificación futura.

Ilustración 8.Sensor Sct-013.

La señal devuelta por el acondicionador se recibe por un sistema opto-acoplado con la terminal GPIO de Raspberry para protección de la misma. Esto también nos abre la posibilidad de emplear alguna otra alternativa para la detección de alimentación alterna siempre y cuando retorne una señal de 3 a 5V.

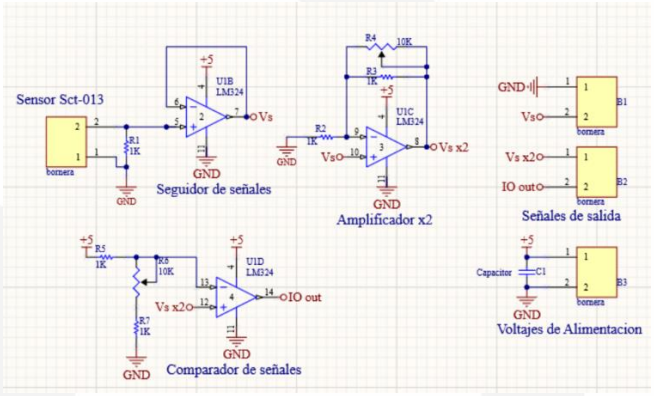


Ilustración 9.Diagrama eléctrico del acondicionador para el sensor Sct-013.

Integración de Hardware a Raspberry pi

Debido a los múltiples módulos y dispositivos a conectarse a raspberry pi se optó por el desarrollo de una tarjeta PCB. El circuito impreso cuenta con los circuitos necesarios para el manejo de hardware necesario para el proyecto las características de la placa se describen a continuación:

- Terminales de comunicación con Raspberry.
- Puerto de alimentación: se puede suministrar alimentación a Raspberry a través de este puerto en caso de ser necesario, es de vital importancia considerar que solo admite 5V.
- Múltiples puertos de suministro de 3V3 y 5V
- Salida transistrorada a 5V (libre).
- Led indicador.

- Emisor IR.
- Terminal de lectura opto-acoplada para sensor de corriente alterna.
- Terminal de conexión para sensor DHT11 a 3V3
- Terminal de conexión para sensor IR

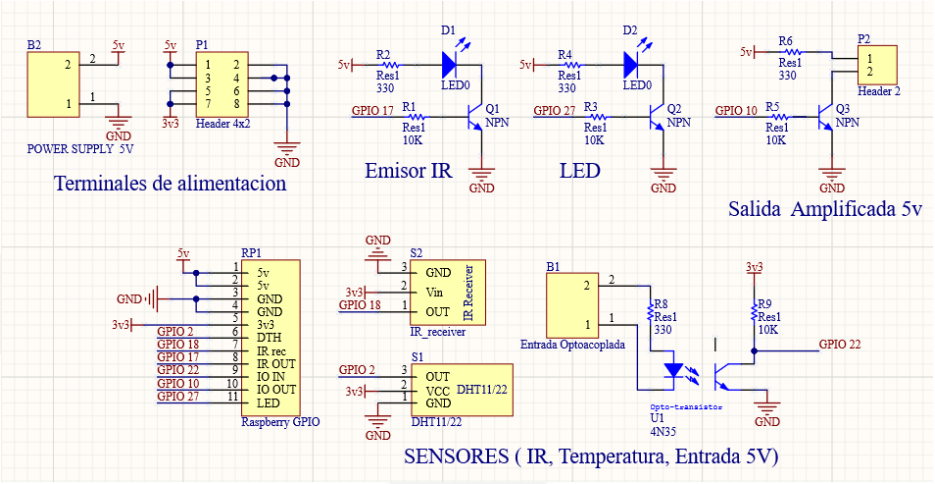


Ilustración 10. Diagrama eléctrico del módulo de integración de hardware a Raspberry.

Modulo MQTT con ESP32

Para una mayor fiabilidad en la medición de temperatura interna de la habitación es conveniente un muestreo en múltiples puntos de la misma, esto nos da una mayor comprensión de las variaciones de temperatura. La consideración del gradiente de temperatura toma mayor importancia conforme las dimensiones del espacio aumenten por lo cual entre mayor sea el espacio puede ser conveniente aumentar el número de puntos a muestrearse.
Basado en el microcontrolador ESP8266-0, el Módulo inalámbrico de muestreo está diseñado para medir la temperatura mediante el sensor DHT1 y transmitirla a un servidor MQTT montado internamente en la Raspberry pi a través de un topic especifico. La ventaja de esta implementación es la practicidad de las tecnologías inalámbricas y posibilidad de aumentar el número de dispositivos fácilmente.

Operación:
Buscando que el dispositivo sea practico y sencillo de configurar se han implementado dos modos de operación: Monitoreo y Configuración.

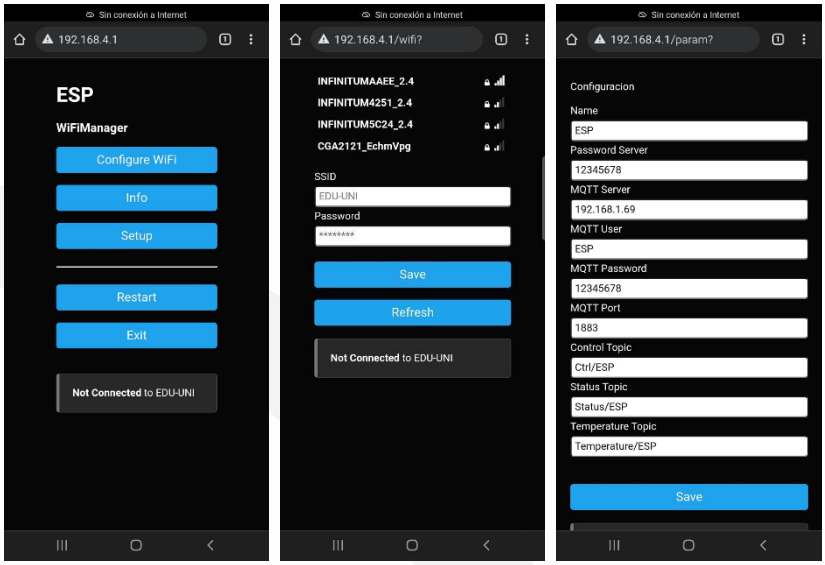
- Monitoreo:

El dispositivo al arrancar automáticamente intentará conectarse a una red Wifi y un broker MQTT, ambos definidos en su configuración interna en memoria. Una vez establecidas las conexiones, quedará a la espera de una instrucción ‘GET’ en el topic de control (‘ESP/control’ por defecto) a la cual responderá con una lectura de temperatura y humedad en el topic correspondiente (‘ESP/data’ por defecto). Adicional a esto el módulo publica actualizaciones de estado y cambios de modo en un topic de estado (‘ESP/Status’ por defecto).

- Configuración:

Al mantener presionado por 5 segundos es botón de configuración, el sistema se desconectará de la red e iniciará en modo de estación wifi donde por medio de una página en la IP “192.168.4.1” podremos acceder desde un buscador en algún dispositivo conectado a su red wifi. La página permitirá modificar los parámetros de operación:

nombre del dispositivo, red wifi, Host MQTT, usuario MQTT, contraseña MQTT y los topics MQTT con los que interactuará. En caso de no conseguir conectarse a la red wifi configurada, el modo de configuración iniciará de manera automática y permanecerá activo por un periodo de tiempo antes de reintentar conectarse a la red.



(a) (b) (c)

Ilustración 11.Paginas desplegadas por el servidor en modo 'configuración'. (a) es el menú principal del sistema desde donde se accede a la configuración de red wifi (b) o a la configuración de la conexión MQTT(c).

Diagrama eléctrico

la tarjeta de desarrollo ESP8266 01 cuenta con una disponibilidad de pines muy limitada sin embargo es más que suficiente para las aplicaciones y condiciones de trabajo a las que estará sujeto el sistema y además permite un diseño compacto. el circuito eléctrico esta dotado de un LED multipropósito, Botones de RESET y configuración, una terminal para el sensor de temperatura DHT11/22 y una terminal Libre capaz de actuar como entrada o salida digital en futuras actualizaciones. Una aplicación potencial muy interesante es la implementación de un emisor infrarrojo, esto para controlar múltiples equipos de climatización con estos módulos compactos gestionados desde una sola Raspberry Central.

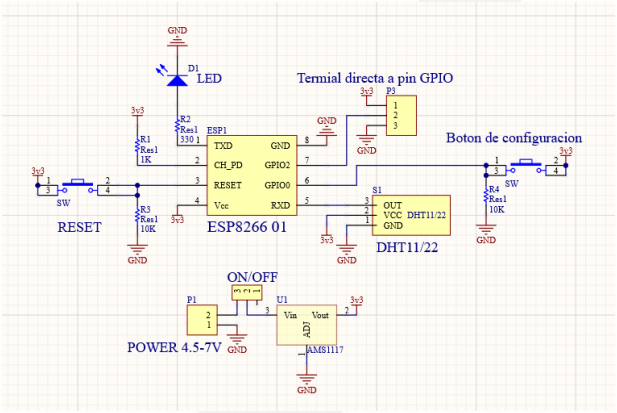


Ilustración 12. Diagrama eléctrico del módulo inalámbrico de muestreo.

Servidor y pagina web

API

La API se ha desarrollado en Python pues este lenguaje cuenta con un amplio catálogo y versatilidad para la implementación de todos los servicios necesarios a ser implementados. La implementación del sistema de manejo de la API fue entorno al micro framework FLASK pues permite crear aplicaciones web sencillas con gran agilidad al codificar. Aunado a esto se han desarrollado gran variedad de librerías y módulos optimizados para su compatibilidad con este framework minimalista. El esquema de la implementación es el siguiente:

Adicional a Flask se ha instalado dentro de Raspberry pi un Broker de "Mosquitto MQTT" para la comunicación con el (o los) modulo inalámbrico de muestreo mientras que la base de datos se administra desde el servidor principal de la institución.

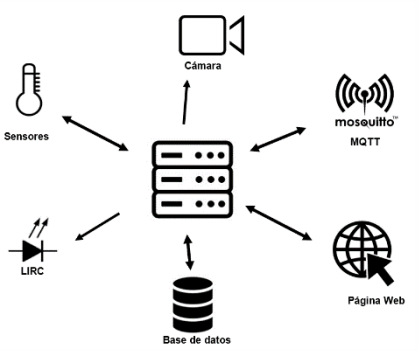


Ilustración 13. Diagrama ilustrativo de servicios empelados en el proyecto.

Página web

La página web está construida en una pantalla principal desde la cual podemos acceder a todo el sistema. Monitoreo en tiempo real de Temperatura y Humedad, estatus de Energía(Corriente Alterna), Streamig de video y configuración actual del equipo de aire acondicionado, así como los controles para la modificación de sus parámetros.

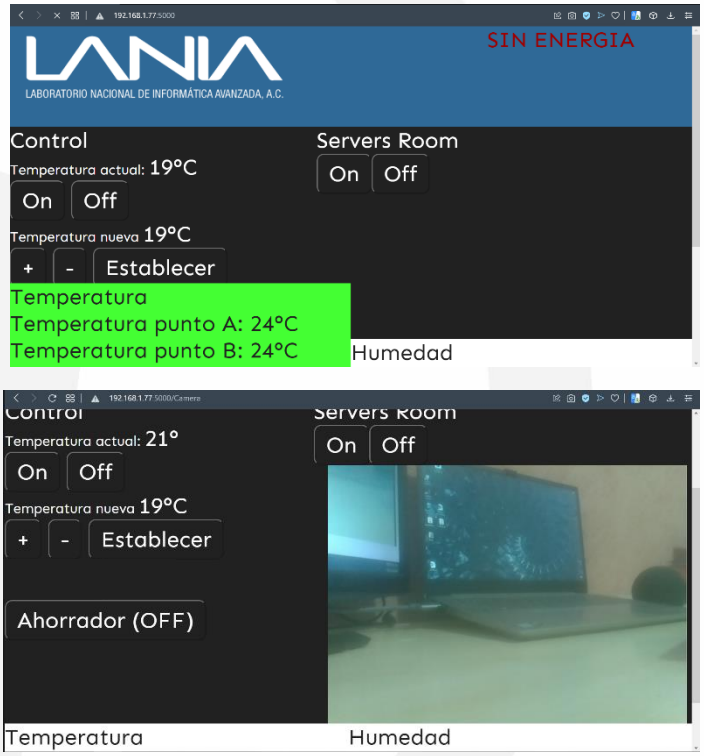


Ilustración 14. Capturas de la pagina web desplegada para el control del sistema. en la captura superior se encuentra inactivo el streaming de video mientras que en la inferior está activo.

Diseño de carcasa para Raspberry Pi

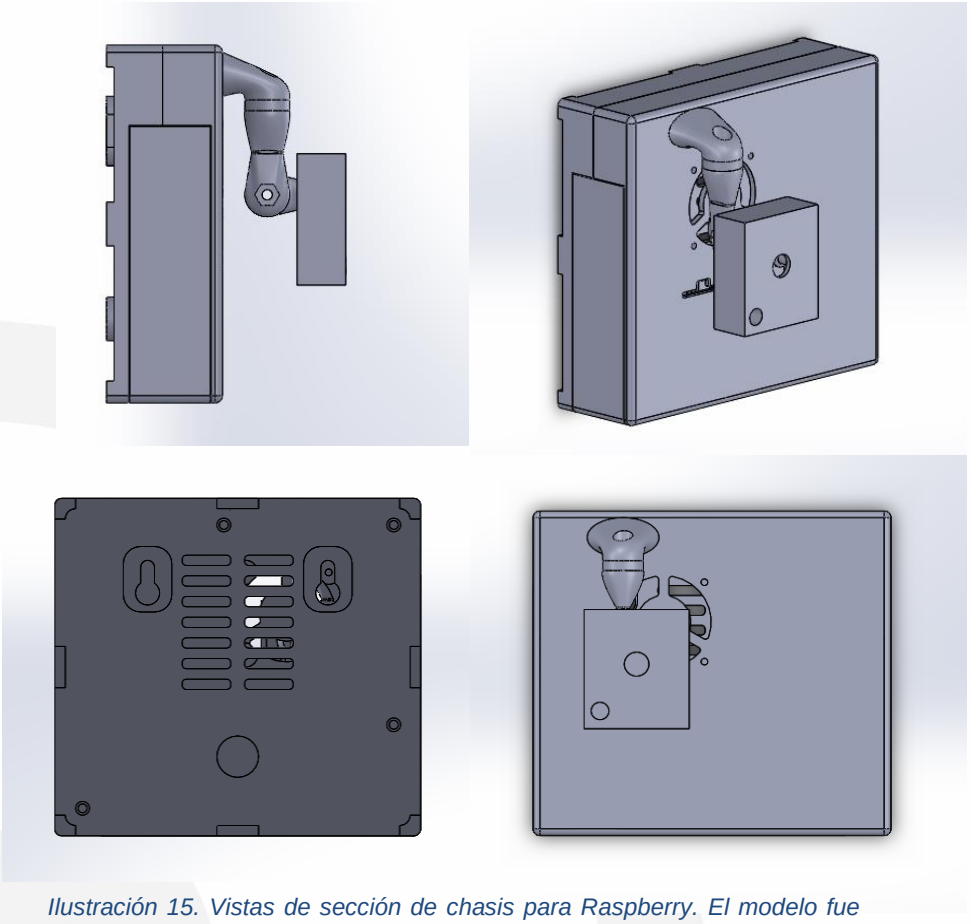


Ilustración 15. Vistas de sección de chasis para Raspberry. El modelo fue realizado en Solidworks

Presentación de resultados

El prototipo es capaz de solventar los requerimientos de la problemática. Después de pruebas en las instalaciones de LANIA se ha corroborado que el equipo es capaz de configurar y accionar los equipos de climatización además de que es capaz de inicializarlo automáticamente en caso de un corte de corriente

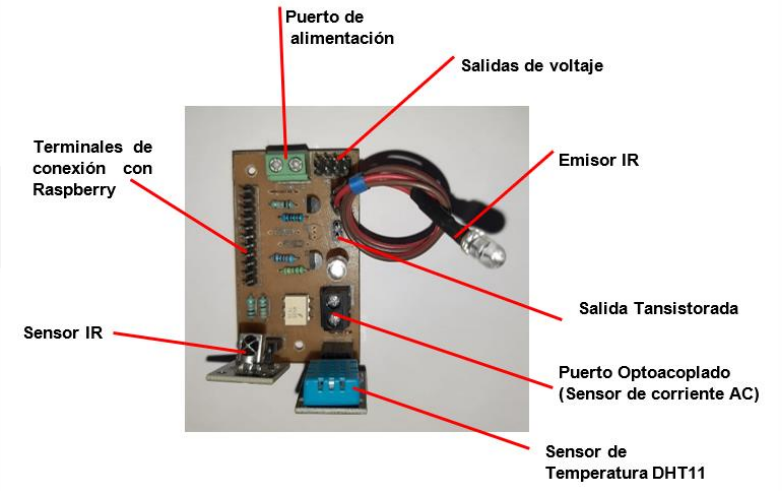


Ilustración 16. Foto del módulo de integración de hardware a Raspberry. Impreso por método de planchado.

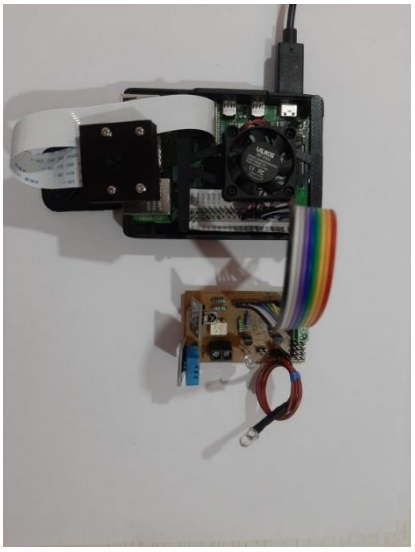


Ilustración 17. Conexión de Raspberry pi con hardware.

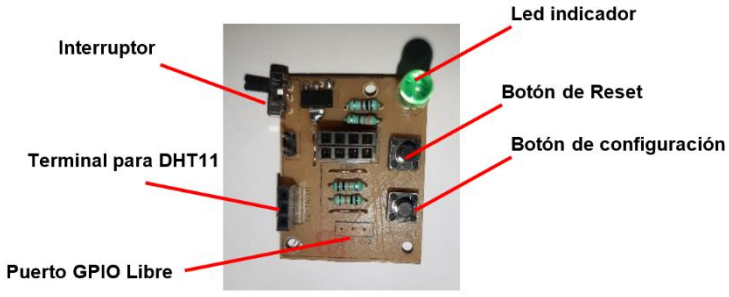


Ilustración 18. Foto del PCB del módulo de muestreo inalámbrico. Impreso por método de planchado.

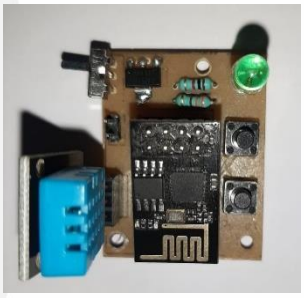


Ilustración 19. Foto del módulo de muestreo inalámbrico. Impreso por método de planchado.

Conclusiones

De acuerdo a las necesidades expresadas por el encargado de servicio social y la institución se desarrolló una propuesta y prototipado de dispositivos inspirados en las corrientes de tecnología emergentes de IoT capaces de brindar un control inteligente de dispositivos para solventar lo solicitado. Gracias a la investigación y prácticas que se realizaron a lo largo del periodo del servicio social pudimos tener un proyecto funcional y exitoso con oportunidad de evolución, así como una gran cantidad de conocimientos que pueden servirnos a lo largo de nuestra formación profesional.

Áreas de mejora

Todo proyecto en desarrollo cuenta con diversas oportunidades de mejora y este sistema no es una excepción. Los puntos más destacables de mejora de desarrollaran a continuación:

- Batería de respaldo propia.

  Actualmente se ha proyectado alimentar los dispositivos desde el sistema de respaldo del servidor. Una batería propia del sistema brindaría mayor fiabilidad e independencia a los dos dispositivos de monitoreo.
  Un acercamiento interesante para el desarrollo de esta mejora sería el uso de un circuito “Load-sharing” respaldado por un microcontrolador de bajo consumo gestionando la conexión, carga y descarga de la batería, esto para asegurar la durabilidad de esta.

- Múltiples emisores

  Dadas las necesidades actuales el sistema está preparado para gestionar solo un equipo de climatización no obstante en caso de requerirse podría modificarse para controlar más de uno. El módulo wifi está preparado para conectar un emisor infrarrojo sin embargo es necesario programar en el controlador las instrucciones de codificación necesarias. También sería necesario hacer las modificaciones correspondientes en la API.

Referencias bibliográficas

Christoph Bartelmus (26 de mayo de 2016) *LIRC Linux Infrared Remote Control*. https://www.lirc.org

Instructables circuits (S.F.) Reverse Engineering Air Conditioner IR Remote Control Protocol https://www.instructables.com/Reverse-engineering-of-an-Air-Conditioning-control/

Epitech. (8 de julio de 2021). *Epitech*. Obtenido de Epitech: https://www.epitech-it.es/flask-python/

Instituto Tecnologico de Aguascalientes. (2014). *Telecomunicaciones TICS*. Obtenido de Telecomunicaciones TICS: https://telecomunicaciones2.webnode.mx/unidad-6/a6-6-comunicacion-por-infrarrojo-/

Oracle. (s.f.). *Oracle*. Obtenido de Oracle: https://www.oracle.com/mx/database/what-is-database/

pue Academy. (s.f.). *pue*. Obtenido de https://www.pue.es/pue-academy/cisco-networking-academy/internet-de-las-cosas

Tech Lib. (s.f.). *Tech Lib*. Obtenido de Tech Lib: https://techlib.net/definition/gpio.html

Nuttall y Jones (S.F) *gpiozero* https://gpiozero.readthedocs.io/en/stable/#

Adafruit Industries (11 de junio de 2019) *Adafruit_Python_DHT* (Repositorio), https://github.com/adafruit/Adafruit_Python_DHT

Mönnich et al. (6 de abril 2021) *Flask. web development, one drop at a time*. https://flask.palletsprojects.com/en/2.0.x/

Stefan Lehmann (2017) *Welcome to Flask-MQTT's documentation!* https://flask-mqtt.readthedocs.io/en/latest/#

Eclipse Foundation (S.F.) *Eclipse Mosquitto™ An open source MQTT broker https://mosquitto.org*

Anexos

Cotización de material para desarrollo del proyecto

Proyecto: **Sistema IoT para monitoreo de servidores**

Alumnos: López Reyes José Ángel
Erick Abimael Rodriguez Hernández

Fecha de cotización: 1 de octubre del 2021

Prototipo de servidor basado en Raspberry pi:

Concepto	Costo (c/u)	unidades	Subtotal
Raspberry pi 3B plus kit • Tarjeta • Fuente de alimentación • Ventilador • Disipadores • Memoria microSD 32GB	$ 1800.00	1	$ 1800.00
Optoacoplador 4N35	$ 15.00	1	$ 15.00
Transistor 2n2222	$ 4.00	3	$ 12.00
Bornera de dos entradas	$ 5.00	2	$ 10.00
Resistencias varias	$ 0.50	10	$ 5.00
Tira Header macho	$ 10.00	2	$ 20.00
Tira Header hembra	$ 10.00	1	$ 10.00
Led Infrarrojo	$ 3.00	1	$ 3.00
Led verde 5mm	$ 2.00	1	$ 2.00
Receptor Infrarrojo	$ 45.00	1	$ 45.00
Sensor DHT11	$ 90.00	1	$ 90.00
Jumpers H-H 10 cm	$ 50.00	1	$ 50.00
Placa fenólica 10x10 cm	$ 50.00	1	$ 50.00
Tornillos milimétricos 3x6mm	$ 3.00	12	$ 36.00
Cámara Raspberry Pi 5mp V1.3 1080p 720p	$ 250.00	1	$ 250.00
TOTAL			$ 2398.00

Prototipo de módulo de monitoreo de temperatura Wifi basado en ESP8266

Concepto	Costo por unidad	unidades	Subtotal
ESP8266 01	$ 84.00	1	$ 84.00
Resistencias varias	$ 0.5	10	$ 5.00
Tira Header macho	$ 10.00	1	$ 10.00
Tira Header hembra	$ 10.00	1	$ 10.00
Led verde 5mm	$ 2.00	2	$ 4.00
Sensor DHT11	$ 90.00	1	$ 90.00
Placa fenólica 5x10 cm	$ 30.00	1	$ 30.00
Tornillos milimétricos 3x6mm	$ 3.00	12	$ 36.00
switch deslizable 3 pines para pcb	$ 30.00	1	$ 30.00
Boton para PCB	$ 3.00	2	$ 6.00
Tornillos milimétricos 3x6mm	$ 3.00	4	$ 12.00
Fuente de alimentación 5V 1a	$ 80.00	1	$ 80.00
Regulador de voltaje AMS1117 (5 piezas)	$ 60.00	1	$ 60.00
TOTAL			$ 457.00

* La Fabricación de las placas PCB se ha cotizado considerando técnicas de prototipado (planchado). En caso de preferirse una fabricación profesional se deberán reconsiderar algunos costos.
* La cotización no incluye costos referentes a impresión 3D de las carcasas necesarias para los prototipos, estos conceptos están por definirse.

LANIA
Laboratorio Nacional de Informática Avanzada A.C.

Rébsamen 80, esq. Circuito Presidentes
C.P. 91000 A.P. 696, Col. Centro
Xalapa, Veracruz, México
Tel. (52) (228) 8416100
Fax. (52) (228) 8416101