

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

MONITOREO Y CONTROL REMOTO DE SENSORES POR MEDIO DE TARJETA OPEN SOURCE ORANGE PI

Julián Andrés Valencia Ramírez

Programa Académico
Tecnología en Electrónica

Director
Alexander Arias Londoño
MSc. IEo.

INSTITUTO TECNOLÓGICO METROPOLITANO

Medellín

Octubre de 2018

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

Para el desarrollo de este proyecto tuve la oportunidad de estar en el semillero MMCC (Modelamiento matemático y computo científico), participe en varias clases reuniones , foros y proyectos basados en IoT.

Realice prácticas de laboratorio en el laboratorio de electrónica como asistente de la ingeniera Marisol Estrada, haciendo mantenimiento a distintos equipos electrónicos y encargado de las dos aulas habilitadas para clases y trabajo en tiempo libre asesorando a estudiantes y demás gestiones. Por ultimo y lo principal es mi proyecto titulado Monitoreo y control remoto de sensores por medio de tarjeta open source orange pi , donde se presenta una implementación para solucionar problemas de adquisicion de datos por medio de mini-ordenadores de bajo costo y de código abierto para el internet de las cosas. Se muestra la adquisición de tres sensores: sensor de movimiento, y sensor de temperatura y humedad. Para ello se configuro la tarjeta de desarrollo Orange Pi con el sistema operativo Armbian, luego se instala un ambiente para manejo de dependencias virtuales, en este caso se hace para configurar el sistema como IoT (Internet of Things). Flask es un microframework para Python, y aprovechamos esta conexión para realizar la programación de las GPIO desde Python. En este caso la tarjeta Orange Pi funciona como servidor para que un usuario desde la web pueda conectarse y hacer monitoreo y control de variables. Por medio de la pagina web el usuario controla cargas de potencia activadas por rele. El sistema hace control de temperatura y activa un ventilador cuando sobrepase los 23°C, y se desactiva por debajo de esta temperatura.

Palabras clave: Orange pi, IoT, Python, sensor , pir , temperatura , humedad , control , monitoreo , rele , camara , web , bombillo , ventilador , script , html , servidor , ip.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

Primeramente muchas gracias a DIOS y a mi familia como lo es mi esposa Lady Bedoya y mi hija Paulina Valencia por ser el motor de mi vida e inspiración para seguir adelante en todos nuestros proyectos. Agradezco mucho al ITM como institución quien me dio la oportunidad de ingresar a estudiar esta interesante carrera tecnológica, por medio de la cual obtuve bastantes conocimientos de electrónica . Gracias a todos los maestros que hicieron parte de esta carrera ; gracias al profesor asesor Alexander Arias por guiarme de la mejor manera para el desarrollo y ejecución de este proyecto. Infinitas gracias a todos los compañeros del semillero MMCC por apoyarme en este desarrollo.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

GPIO General Purpose Input / Output, Entrada / Salida de Propósito General

IP protocolo internet

PIR (Passive Infrared) o pasivo Infrarrojo

SD SanDisk

MMCC Modelamiento matemático y computo científico

HTML HyperText Markup Language, es decir, Lenguaje de Formato de Documentos.

LCD Liquid Cristal Display, representación visual por cristal líquido

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Tabla de contenido

RESUMEN.....	2
RECONOCIMIENTOS	3
ACRÓNIMOS	4
TABLA DE FIGURAS	7
1. INTRODUCCIÓN.....	9
1.1. Objetivo General	9
1.2. Objetivos específicos.....	9
2. MARCO TEÓRICO	10
2.1 Clases de tarjetas Open Source.....	10
2.1.1 Arduino.....	10
2.1.2 Raspberry pi	11
2.1.3 Beagle bone.....	11
2.1.4 Orange pi.....	12
2.2.Arquitectura de la Orange Pi Pc Plus.....	13
2.3. Sistema operativo Armbian	15
2.4. GPIO regleta de expansión de 40 pines.....	16
2.5.Flask microframework (servidor en python).....	17
3. METODOLOGÍA	19
3.1. Instalación de armbian En la Orange Pi.....	19
3.1.1. Preparar la tarjeta tf micro sd	20
3.1.2. Conectar la tarjeta orange pi pc plus.....	22
3.1.3. Hacer un test de acceso con armbian	23
3.2. Instalacion de python con armbian.....	25

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.3. Instalacion de flask microframework	26
3.4. Monitoreo y control remoto de sensores por medio de tarjeta open sources orange pi	27
3.4.1. Activacion remota de un sistema de alarma por medio de un sensor de movimiento.	27
3.4.2 Visualizacion remota de valores reales de sensores de temperatura y humedad.....	29
3.4.3. Activacion de un relé remotamente.....	30
3.4.4. Control de temperatura.....	31
3.5 Codigo en Python para monitoreo y control de sensores.....	32
3.6 . Diseño de Aplicación Web en Microframework Flask.....	34
3.7. Sistema de vigilancia con web cam	35
3.8. XRDP Escritorio remoto en Armbian	39
Anexo A. Configuración paso a paso de la GPIO para encender un Led en la Orange Pi	48
Anexo B. Código completo de la aplicación en Python 3.0.....	52
Anexo C. Código aplicación web con bootstrp y html con css.	54
Anexo D. Participación en Evento de la Semana de Ingeniería del ITM 2018	57
Anexo E. Certificado de Participación en Evento de la Semana de Ingeniería del ITM 2018	62

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE FIGURAS

Figura 1. Símbolo de Arduino.	11
Figura 2. Página web de Raspberry Pi.	11
Figura 3. Tarjeta Beagle Bone.....	12
Figura 4. Simbolo Orange Pi y su S.O.....	12
Figura 5. Arquitectura frontal de la Orange PI.	13
Figura 6. Parte inferior Orange Pi.....	14
Figura 7.Ambian instalado.	16
Figura 8. GPIO de Orange Pi.	17
Figura 9. Elementos básicos para Instalacion.....	19
Figura 10. Opcion formateo SD.	21
Figura 11. Formateo tarjeta SD.	21
Figura 12. Abrir imagen descomprimida.	22
Figura 13. Conexion de perifericos.....	23
Figura 14. Primer acceso Armbian.....	24
Figura 15. S.O Armbian instalado.	25
Figura 16. Python 3.7 instalado.	25
Figura 17. Instalacion de Flask. 1.....	26
Figura 18.Codigo Hola Mundo.....	26
Figura 19. Ejecucion del programa.....	27
Figura 20. Sensor PIR desactivado en off	28
Figura 21. Sensor PIR IR utilizado.	28
Figura 22. Valores reales de sensor dig.....	29
Figura 23. Sensor DTH11.	29
Figura 24. Rele apagado en servidor.	30
Figura 25. Rele 5v 10A con optoacoplador.....	31
Figura 26. Valor de ventilador en ON.	31
Figura 27. Sensor DTH11 y Ventilador.....	32
Figura 28. Diagrama Driver de potencia.....	32
Figura 29. Script python linea1-38.	33
Figura 30. Script html linea 1-32.....	34

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Figura 31. Camara web 720p.....	35
Figura 32. Imagen 1 camara web.	39
Figura 33. Imagen 2 camara web.	39
Figura 34. Abrir Escritorio Remoto desde Windows.	40
Figura 35. XRDP escritorio remoto linux.	41
Figura 36. Escritorio Remoto de Armbian con Interfaz Gráfica.....	41
Figura 37. Diagrama del proyecto con los sensores y actuador conectados.	42
Figura 38. Interfaz de usuario en movil.....	43
Figura 39. Conexion para encender led.....	51
Figura 40. Poster proyecto de aula.....	58
Figura 41. Exposicion proyecto1.....	59
Figura 42. Exposicion proyecto 2.....	60
Figura 43. Exposicion proyecto 3.....	61
Figura 44. Certificado Participación en Evento.	62

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

La aplicación de los sensores aplicados en IoT a sistemas de alarmas es importante para el monitoreo remoto y el control de la activación de cargas. En este trabajo se mostrara el diseño de lot con un sensor de movimiento el cual muestra valores reales de temperatura y humedad. También podemos activar cierto dispositivo o máquina para que realice cierta función solo haciendo un clic desde el navegador del dispositivo móvil o de escritorio, este acciona un relé. Además, se puede tener control de un sistema, en este caso se controla la temperatura, puede ser en máquinas o dispositivos donde la temperatura es necesaria que sea estable para su buen comportamiento o estabilidad en el sistema. De igual manera se puede aplicar monitoreo y control al sistema de vigilancia con cámaras web, ya que esta también la podemos ver remotamente desde un servidor y se puede estar configurando constantemente por medio de comandos dependiendo de lo que se desee realizar, esto puede ser que grabe la cámara continuamente, que haga tomas por movimientos, que tome fotos, que grave videos y que la cámara se puede apagar y encender por medio de comandos en la consola remota.

1.1. Objetivo General

Desarrollar una aplicación IoT para monitoreo de sensores y control remoto por medio de un servidor en la tarjeta open source Orange Pi y visualización por medio de página web.

1.2. Objetivos específicos

- Implementar y configurar el sistema operativo armbian en la tarjeta open source ORANGE PI.
- Desarrollar una aplicación en Python para manejo del GPIO de la tarjeta Orange Pi.
- Implementar el microframework Flask para realizar web embebido en la tarjeta Orange Pi.
- Desarrollar una aplicación web con interfaz al framewor Flask para la adquisición de variables y control de cargas por medio de relé.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEÓRICO

En este capítulo primero conoceremos cuatro diferentes tipos de tarjetas de código abierto existente, se escoge por prestaciones y bajo costo. Como segundo punto en este capítulo esta la arquitectura de la tarjeta Orange Pi Pc Plus la cual contiene todas sus partes y conexiones. En tercer lugar se habla del sistema operativo Armbian, el cual fue necesario instalarlo para que este mini-ordenador trabaje correctamente. En cuarto lugar, esta el paso a paso de la instalación de las GPIO con Python y soportado con graficas. Por último se encuentra la instalación del microframework Flask que nos sirve como servidor para visualizar todo en una pagina web.

2.1 Clases de tarjetas Open Source

En este apartado se trabaja el estado del arte de cuatro tarjetas: Arduino ,Raspberry Pi , Beagle bone y la Orange Pi. A continuación se explicarán las características de cada una de ellas.

2.1.1 Arduino

Tiene la comunidad mas grande derivada de su filosofía open source, tanto en software como en hardware, además de que es muy fácil en curva de aprendizaje, el símbolo de Arduino se muestra en la Figura 1.

Arduino viene a satisfacer la demanda de los usuarios por hacer que el hardware se comuniquen e interactúen con varios dispositivos de manera sencilla, nos permite escalar con algo llamado shields, una características que permiten funcionalidades nada convencionales: comunicación wifi, sensores , GPS , SD Card reader y pantallas LCD (Arduino , 2018).

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



What is Arduino?

Figura 1. Símbolo de Arduino.

2.1.2 Raspberry pi

Es un ordenador completo con el tamaño mas reducido que se pueda concebir. Basta con conectarlo a un TV o monitor, usar los puertos usb para teclado y mouse para disfrutar de cualquiera de las distribuciones Linux disponibles para ella (Pi R. , 2018).

Gracias a sus GPIO (General Purpose Input/Output) abrió el camino de crear proyectos muy interesantes. La página web de la Raspberry se puede ver en la Figura 2.



Figura 2. Página web de Raspberry Pi.

2.1.3 Beagle bone

Beaglebone es una tarjeta de desarrollo de bajo costo desarrollada por la organización Beagleboard.org, la cual está orientada en estimular el uso de software y hardware open source, así como el conocimiento y el intercambio de ideas (beagleboard, 2018).

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Es una plataforma que corre bajo un sistema operativo Linux (cabe señalar que actualmente existen varias distribuciones de linux para las plataformas Beaglebone), y que cuenta con diversas entradas y salidas de propósito general las cuales cuentan con diversas funciones entre las cuales se encuentran (I/O Digitales, Entradas Analógicas, Salidas con PWM, soporte para I2 & SPI). Además cuenta con un puerto ethernet para la comunicación en red con otros dispositivos y un puerto USB 2.0 para la comunicación con otros dispositivos . La tarjeta Beagle Bone se muestra en Figura 3



Figura 3. Tarjeta Beagle Bone.

2.1.4 Orange pi

Es un computador de código abierto con todos los periféricos en una sola tarjeta , similar a la raspberry, puede funcionar con los sistemas operativos Android 4.4, Ubuntu, debian ,armbian . Utiliza el SoC Allwinner H3 y tiene 1GB de DDR3 de SDRAM y 8GB EMMC flash. Se puede construir un computador , un servidor inalámbrico, juegos. Se puede desarrollar con tecnología ; es una tarjeta mas complicada , con aplicaciones y proyectos mas profesionales (PI, 2018). Podemos observar su símbolo en la Figura 4.



Figura 4. Símbolo Orange Pi y su S.O.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Dadas las buenas prestaciones de cómputo y periféricos con sensores de la Orange Pi y su bajo costo, esta se escoge para desarrollar los proyectos de este trabajo de grado. Como esta se implementará en el transcurso del documento, en el ítem 2.2 y 2.3 se trabajan la arquitectura y sistema operativo para la Orange Pi respectivamente.

2.2.Arquitectura de la Orange Pi Pc Plus

A continuación veremos la placa madre, que es la tarjeta de circuito impreso de la Orange Pi Pc Plus a la que se conectan los componentes que constituyen este mini-ordenador. Tiene instalado tres puertos usb 2.0, un puerto ethernet de 100M, un sensor receptor infrarrojo, un conector hdmi, una antena wifi y entre otros una memoria flash de 8GB como se muestra en la Figura 5. (electronicas).

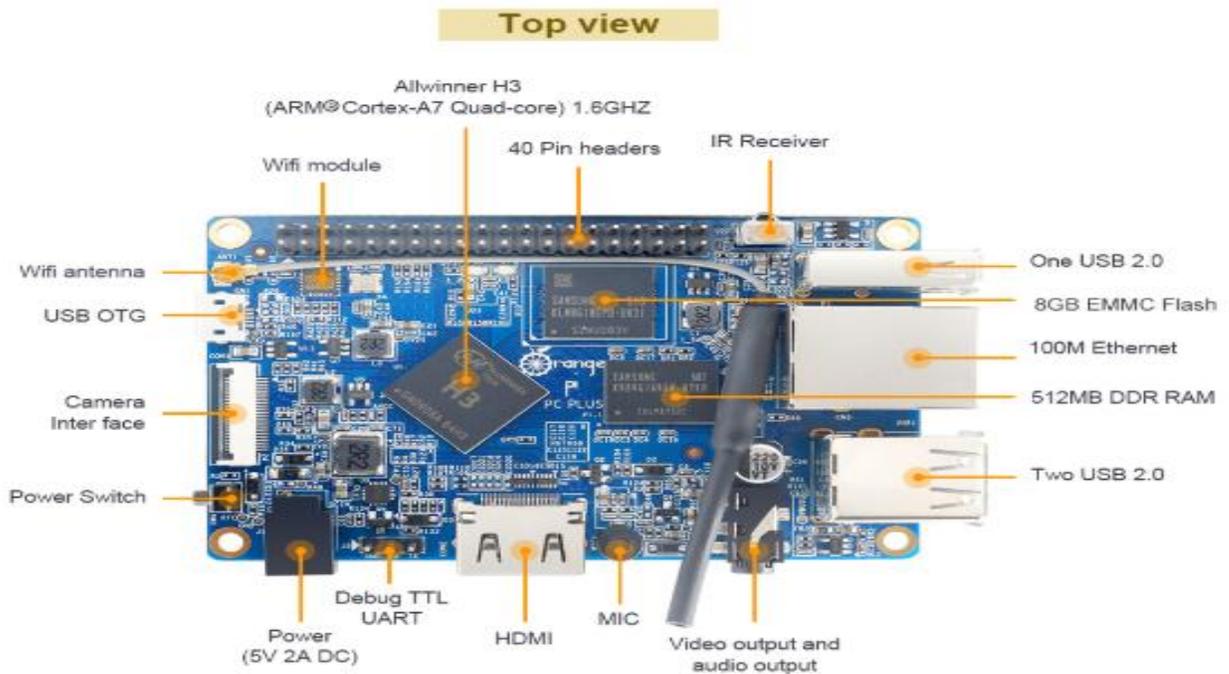


Figura 5. Arquitectura frontal de la Orange Pi.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En la parte inferior de la tarjeta se encuentra la TF card slot donde se inserta la tarjeta microSD y una memoria RAM DDR de 512MB como se observa en la Figura 6.

¡Error! No se encuentra el origen de la referencia.

Figura 6. Parte inferior Orange Pi.

¡Error! No se encuentra el origen de la referencia.

La tarjeta Orange Pi consta de una arquitectura de PC completa y adicionalmente presenta periféricos de sensores como audio e infrarrojos. Una descripción más detallada se presenta en la Tabla 1.

Tabla 1. Especificacion de hardware de la tarjeta Orange Pi.

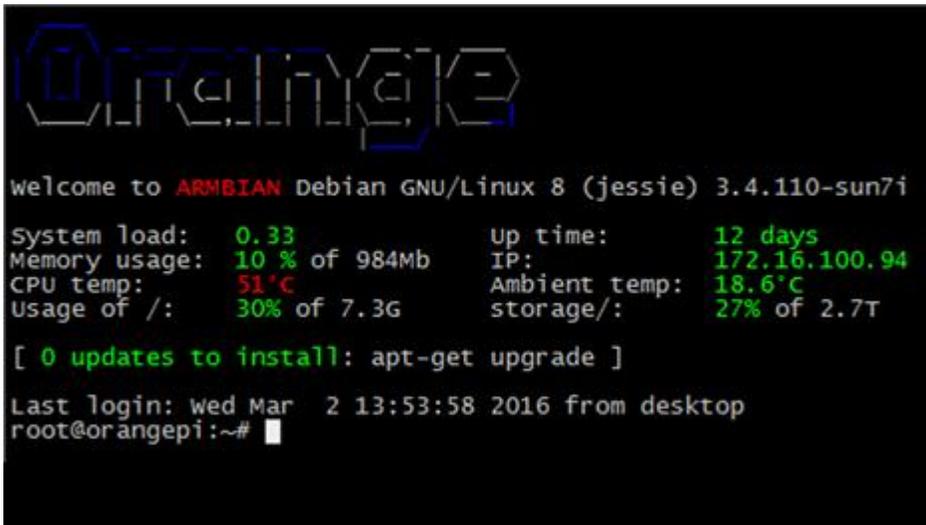
CPU	H3 QUAD CORE cortex A7 H.265/HEVC 4K
GPU	GPU Mall 400mp2 a 600MGZ
MEMORIA (SDRAM)	1GB DDR3 compatible con GPU
Almacenamiento a bordo	Tarjeta TF (Max.32GB) /Ranura para tarjeta MMC O 8GB EMMC Flask
RED a bordo	Ethernet 10/100 Rj 45 y antena WIFI
Entrada de Video	Un conector de entrada CSI cámara Admite la interfaz del sensor Cmos YUV 422 de 8 bits Admite el protocolo CCIR656 para NTSC y PAL Admite el sensor de camaras de pixeles SM Admite la solucion de captura de video hasta 1080P a 30 Fps
Entrada de audio	MIC
Salidas de video	Admite salida de video HDMI con HDCP Admite HDMI CEC Admite la función HDMI 30 CVBS integrado Admite salida simultanea de HDMI y CVBS
Salida de audio	Jack de 3.5mm y HDMI
Fuente de alimentacion	Entrada de cc. L a entrada USB OTG no suministra alimentacion
Puerto USB 2.0	Tres USB 2.0 HOST Y un USB 2.0 OTG
Botones	Boton de encendido
Perifericos de bajo nivel	Cabezal de 40 pines compatible con raspberry pi B+
PIN GPIO (1*3)	UART , suelo
LED	Led de alimentación y led de estado
Llave	Entrada IR, Power
Sistema operativo compatible	Android Lubuntu, Debian ,Raspbian imagen y Armbian

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.3. Sistema operativo Armbian

Armbian es un sistema operativo ligero orientado específicamente a su utilización en chipset ARM, el chipset que incorporan la mayoría de placas de desarrollo que tenemos hoy en día, en mi caso la Orange Pi Pc Plus. Armbian esta basada en dos distribuciones Linux de escritorio ampliamente conocidas, Debian Jessie y también Ubuntu Xenial. Para poner en marcha este sistema operativo , simplemente deberemos flashear la imagen del sistema operativo en la tarjeta micro SD o micro SD (soloelectronicos, 2018).

Cuando iniciamos sesión via SSH en el sistema operativo , nos aparecerá la temperatura actual de la CPU, la carga del sistema, el tiempo que lleva encendido, la dirección IP privada e incluso también el uso de los diferentes almacenamientos que tengamos en el miniordenador; como nos muestra la Figura 7.



```

ORANGE
welcome to ARMBIAN Debian GNU/Linux 8 (jessie) 3.4.110-sun7i
System load:  0.33      Up time:      12 days
Memory usage: 10 % of 984Mb  IP:           172.16.100.94
CPU temp:     51°C      Ambient temp: 18.6°C
Usage of /:   30% of 7.3G  storage/:     27% of 2.7T

[ 0 updates to install: apt-get upgrade ]
Last login: wed Mar  2 13:53:58 2016 from desktop
root@orangepi:~#

```

Figura 7. Armbian instalado.

2.4. GPIO regleta de expansión de 40 pines

Una de las principales características de la Orange Pi es su conector GPIO (*General Purpose Input / Output, Entrada / Salida de Propósito General*). Esta placa tiene un conector de 40 pines (de forma muy similar al IDE de las placas base de ordenador) que sirve para manejar señales de entrada / salida, al que se puede conectar hardware externo (Pérez Santos, 2018).

Además de suministrar energía (3,3 V, 5 V y Masa), todos los pines GPIO pueden utilizarse como entradas o salidas digitales. Con estos pines podemos leer valores de entrada (por ejemplo, sensores

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

de temperatura, de humedad, de aceleración, ...) así como enviar señales de salida (para control de motores, señalización en leds o displays...)

Absolutamente todos los pines tienen niveles lógicos de 3.3V y no son 5V por lo que los niveles de salida son entre 0 y 3.3Vdc y las entradas nunca deben ser superiores a 3.3V. Si desea conectar una salida de 5V a una entrada Pi, habrá que utilizar un convertidor de niveles.

Para observar lecturas de los pines (entrada de datos digital) o escribir en ellos (salida de datos digital) se pueden utilizar varios lenguajes como C o Python. Para poder usar estos lenguajes, debemos usar librerías específicas que traduzcan los comandos en datos reales tanto de entrada como de salida en el GPIO.

Como salidas digitales, puede escribir programas que asignan a un determinado pin un valor ALTO (HIGH) o BAJO (LOW). Si se asigna ALTO, se tendrán 3.3V entre dicho PIN y masa y si se establece en BAJO, tendremos un valor de 0V. Para encender un LED en uno de estos pines, necesitarás conectar una resistencia de 1kΩ en serie con el LED.

Si se utiliza un pin como entrada digital, entonces puedes conectar interruptores, pulsadores y sensores simples a un pin y luego puede comprobar si está abierto o cerrado.

Lo primero que debemos saber es la disposición de estos pines. En manual esquemático de diseño del hardware de las placas Orange Pi, como se muestra en la Figura 8.

ORANGE PI GPIO

VOLTAGE	3.3V	1	2	5V	VOLTAGE
I2C_0_SDA	PA12	3	4	5V	VOLTAGE
I2C_0_SCL	PA11	5	6	GND	REFERENCE
GPIO	PA6	7	8	PA13	UART_3_TX
REFERENCE	GND	9	10	PA14	UART_3_RX
UART_2_RX	PA1	11	12	PD14	GPIO
UART_2_TX	PA0	13	14	GND	REFERENCE
UART_2_CTS	PA3	15	16	PC4	GPIO
VOLTAGE	3.3V	17	18	PC7	GPIO
SPI_0_MOSI	PC0	19	20	GND	REFERENCE
SPI_0_MISO	PC1	21	22	PA2	UART_2_RTS
SPI_0_SCK	PC2	23	24	PC3	SPI_0_CS
REFERENCE	GND	25	26	PA21	GPIO
I2C_1_SDA	PA19	27	28	PA18	I2C_1_SCL
GPIO	PA7	29	30	GND	REFERENCE
GPIO	PA8	31	32	PG8	UART_1_RTS
GPIO	PA9	33	34	GND	REFERENCE
GPIO	PA10	35	36	PG9	UART_1_CTS
GPIO	PA20	37	38	PG6	UART_1_TX
REFERENCE	GND	39	40	PG7	UART_1_RX

5V	3.3V	GND	GPIO	UART	I2C	SPI
----	------	-----	------	------	-----	-----

Figura 8. GPIO de Orange Pi.

2.5.Flask microframework (servidor en python)

El servidor Flask está basado en Werkzeug, una librería en WSGI (Web Server Gateway Interface) que es la que se encarga de manejar las requests entre el cliente y el servidor. Y por otro lado JINJA 2, un amigable, moderno, seguro y popular lenguaje de templates utilizado para Python basado en el utilizado por Django. (Iomelisan, 2018)

Flask no es como los demás frameworks que cuentan con un sin fin de módulos ya integrados, en el caso de éste, siempre podrás integrar lo que desees, pero solo si yo así lo decido. En realidad, solamente contare con lo mínimo para poder desarrollar mi aplicación, lo demás quedará a decisión de los demás si agregarlo o no y dependerá siempre de la aplicación que esté desarrollando.

Esto nos permite no solamente tener control casi total del código nuestra aplicación, sino además tener código ligero, liberar las aplicaciones al mercado más rápidamente y por sobre todo, tener código fácil de mantener, algo que sin duda para el desarrollador es realmente importante.

Otro detalle sobre éste framework es la cantidad y calidad de su documentación, pues allí encontrarás cualquier cosa que estés buscando y también muy buenos ejemplos.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGÍA

En este capítulo se mostrará como se desarrollan las aplicaciones para el monitoreo y control de sensores y dispositivos para llegar a una aplicación IoT. El capítulo empieza mostrando la metodología de Instalación de Armbian, continua con la implementación de Python en Armbian, sigue con la implementación de un sistema de monitoreo y control con Python, continua con la configuración del microframework Flask y por último se muestra el sistema de monitoreo por medio de cámara web.

3.1. Instalación de armbian En la Orange Pi

Para empezar necesitamos tener un tarjeta Orange Pi Pc Plus en mi caso, una tarjeta SD 32G , un monitor lcd , un teclado , un mouse , un cable hdmi y una fuente de alimentación de 5vdc a 2A. En la Figura 9 se muestran los elementos principales para configurar un mini pc con la Orange Pi, estos son: Tarjeta micro SD, Tarjeta Orange Pi y monitor lcd.



a) Tarjeta micro SD

b) Tarjeta orange pi

c) Monitor LCD

Figura 9. Elementos básicos para Instalacion.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.1.1. Preparar la tarjeta tf micro sd

Para encender y que funcione la Orange Pi, hay que instalar un sistema operativo (OS) en una tarjeta MicroSD. Las instrucciones a continuación te enseñará cómo escribir una imagen de sistema operativo a la tarjeta MicroSD en Windows y Linux (orangePi, 2018).

Recomiendo empezar por Armbian porque es lo más estable. Descarga la distribución exacta para tu placa (muy importante que sea tu placa exactamente) .En Mi caso escogi DEBIAM JESSIE (escritorio).

Instalar imagen linux desde windows

- Insertar la tarjeta micro SD en el ordenador, en mi caso una clase 10 de 16GB.
- Formatear la tarjeta micro SD.
- Descargar la herramienta de formateo de tarjetas micro SD
https://www.sdcard.org/downloads/formatter_4/eula_windows/
- descomprimir el archivo descargado y ejecutar setup.exe para instalarla .
- en el botón "*opciones*" (option), ajuste la opción "*tipo de formato*" en *rapida* (quick), y la opción "*ajuste formato tamaño*" en *on*; como se observa en la Figura 10.

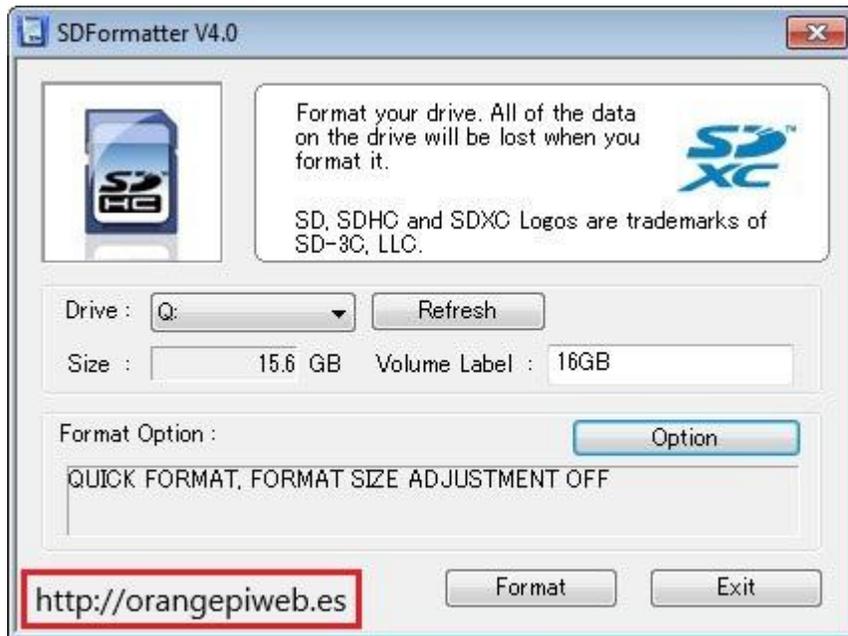


Figura 10. Opcion formateo SD.



Figura 11. Formateo tarjeta SD.

- Pulsa sobre el botón Format. La tarjeta quedará preparada para instalar un sistema operativo .

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Descargar una herramienta para grabar imágenes en una tarjeta MicroSD, como, por ejemplo, Win32diskimager en: <http://sourceforge.net/projects/win32diskimager/files/Archive/>
- Ejecuta el programa Win 32 Disk Imager

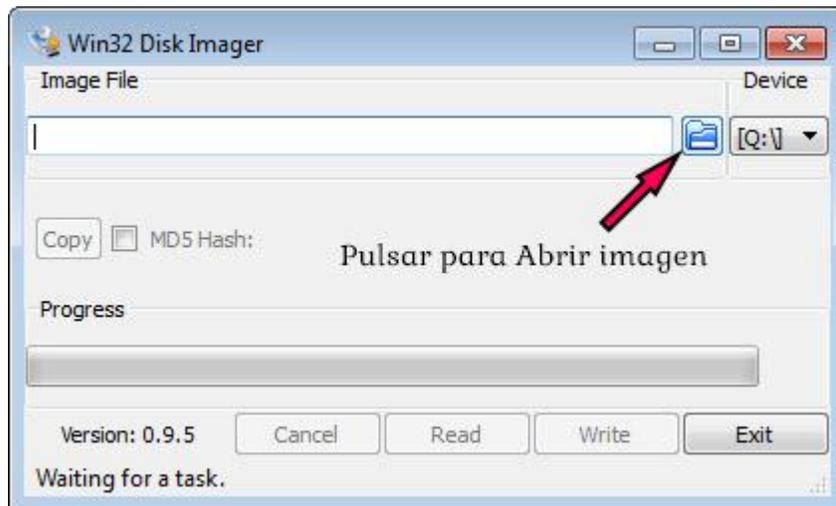


Figura 12. Abrir imagen descomprimida.

- Abre el archivo de imagen descomprimida como lo muestra la Figura 12.
- Comprueba que la tarjeta MicroSD corresponde a la que ha seleccionado la herramienta (Device).
- Haz clic en el botón "Write". Tardará un poco, así que ten paciencia hasta que la imagen se escriba completamente.
- La tarjeta ahora tiene un Sistema Operativo y está lista para arrancar la unidad Orange Pi.

3.1.2. Conectar la tarjeta orange pi pc plus

Para empezar tendremos que conectar cada uno de los diferentes periféricos, el teclado como el mouse entran a puertos usb, del puerto hdmi sale al monitor, en mi caso utilice un convertor de hdmi a vga porque mi monitor solo tiene como salida de video vga. La conexión del cable de red al puerto ethernet es opcional ya que esta tarjeta cuenta con antena wifi. Todas estas conexiones se muestran en la Figura 13.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

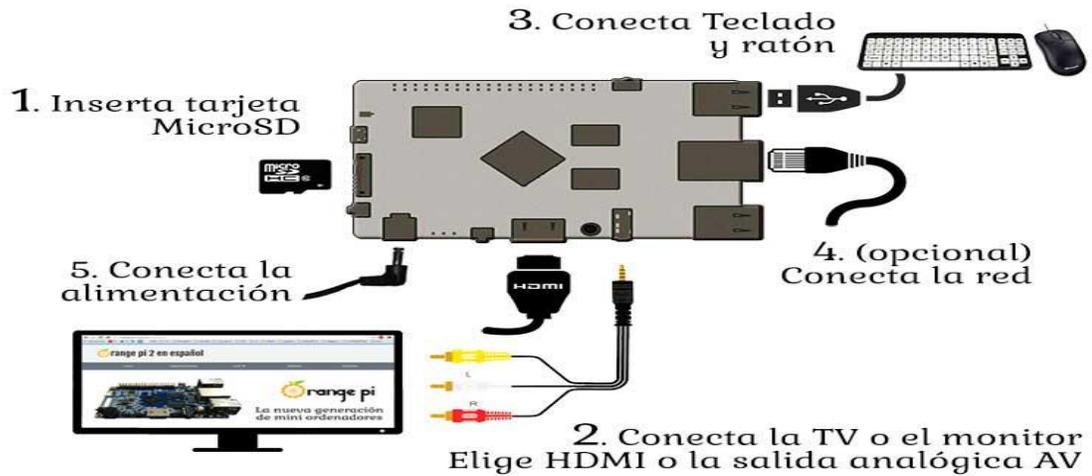


Figura 13. Conexión de periféricos.

3.1.3. Hacer un test de acceso con armbian

El primer acceso será en modo terminal. Lo primero que te pregunta es el nombre de usuario, y tendrás que acceder con el usuario root, más adelante crearemos otro con menos derechos y más seguro de usar. La clave es 1234. Te pide que cambies esa clave, así que tienes que escribir de nuevo la clave 1234 y luego la que quieras que sea definitiva para el usuario root dos veces. A continuación te pide un nombre de usuario que será el que uses normalmente. Introduce tu nombre de usuario y una clave, también dos veces, como se ilustra en la Figura 14.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

192.168.1.132 - PuTTY
login as: root
root@192.168.1.132's password: 1
You are required to change your password immediately (root enforced)

ORANGEPIPC

Welcome to ARMBIAN Debian GNU/Linux 8 (jessie) 3.4.112-sun8i
System load: 0.00 Up time: 12 min
Memory usage: 6 % of 1000Mb IP: 192.168.1.132
CPU temp: 45°C
Usage of /: 32% of 7.3G

New to Armbian? Check the documentation first: docs.armbian.com

Changing password for root.
(current) UNIX password: - Contraseña de root actual: 1234
Enter new UNIX password: - Nueva contraseña root (2 veces)
Retype new UNIX password:
Thank you for choosing Armbian! Support: www.armbian.com

Creating new account. Please provide a username (eg. your forename): opi
Adding user `opi' ...
Adding new group `opi' (1000) ...
Adding new user `opi' (1000) with group `opi' ...
Creating home directory `/home/opi' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for opi
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] Y

Dear , your account opi has been created and is sudo enabled.
Please use this account for your daily work from now on.

root@orangeipic:~# ls
Desktop Documents Downloads Music Pictures Public Templates Videos
root@orangeipic:~#

```

Figura 14. Primer acceso Armbian.

Lo que te pregunta a continuación son datos de usuario que puedes dejar en blanco y si quieres cambiar la resolución de tu monitor.

Ya con todo esto tenemos un sistema Linux activo y funcional, como nos muestra la Figura 15.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



Figura 15. S.O Armbian instalado.

3.2. Instalacion de python con armbian

Por defecto Armbian ya tiene instalado el PYTHON 2.7 y PYTHON 3.0 como se muestra en la Figura 16.

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)]
Type "help", "copyright", "credits" or "license" for more information.
>>> python3
```

Figura 16. Python 3.7 instalado.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.3. Instalacion de flask microframework

Para la instalación del framework debemos digitar en la terminal “pip install flask”.

Allí deberíamos ver un OUTPUT como nos muestra la Figura 17.

```
[root@localhost Flask]# pip install flask
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip install --user` instead.
Collecting flask
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
    100% |#####| 92kB 412kB/s
Collecting click>=2.0 (from flask)
  Downloading click-6.7-py2.py3-none-any.whl (71kB)
    100% |#####| 71kB 589kB/s
Collecting Jinja2>=2.4 (from flask)
  Downloading Jinja2-2.9.6-py2.py3-none-any.whl (340kB)
    100% |#####| 348kB 587kB/s
Collecting Werkzeug>=0.7 (from flask)
  Downloading Werkzeug-0.12.2-py2.py3-none-any.whl (312kB)
    100% |#####| 317kB 672kB/s
Collecting itsdangerous>=0.21 (from flask)
  Downloading itsdangerous-0.24.tar.gz (46kB)
    100% |#####| 51kB 795kB/s
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->flask)
  Downloading MarkupSafe-1.0.tar.gz
Installing collected packages: click, MarkupSafe, Jinja2, Werkzeug, itsdangerous, flask
Running setup.py install for MarkupSafe ... done
Running setup.py install for itsdangerous ... done
Successfully installed Jinja2-2.9.6 MarkupSafe-1.0 Werkzeug-0.12.2 click-6.7 flask-0.12.2 itsdangerous-0.24
```

Figura 17. Instalacion de Flask. 1

Para poder comenzar con nuestro proyecto, debemos crear una carpeta, en la cual tendremos todos nuestros archivos. En mi caso será por ejemplo “flask-project“. Allí, creamos un archivo llamado “app.py” y le colocamos el siguiente código dentro (GUIADEV, s.f.), como nos muestra la Figura 18.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hola_mundo():
    return 'Hello, World!'
```

Figura 18. Código Hola Mundo.

Ahora debemos exportar la aplicación y correrla, para luego poder verla desde nuestro navegador. Esto lo podemos hacer sencillamente con los siguientes comandos.

Nos paramos dentro del directorio donde tenemos el archivo app.py

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Corremos el comando “export FLASK_APP=app.py”.

Y ahora corremos la aplicación con “flask run”

Veremos que nos saldrá un output como nos muestra la Figura 19.

```
[root@localhost flask-project]# nano w app.py
[root@localhost flask-project]# export FLASK_APP=app.py
[root@localhost flask-project]# flask run
* Serving Flask app "app"
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figura 19. Ejecucion del programa.

Aquí nos muestra que nuestra aplicación se encuentra corriendo en la dirección ip con puerto “http://127.0.0.1:5000”. Por lo que para ver lo que hace, simplemente debemos acceder al navegador y cargar la dirección, y nos mostrara el mensaje “ hola mundo”.

3.4. Monitoreo y control remoto de sensores por medio de tarjeta open sources orange pi.

En este ítem se muestran los cinco proyectos desarrollados: Sensor de Movimiento, sensor de temperatura y humedad, activación de relé, control de temperatura y monitoreo con cámara web. En cada punto se muestran los elementos o sensores utilizado, su código en Python y su visualización desde una pagina web pedida al servidor.

3.4.1. Activacion remota de un sistema de alarma por medio de un sensor de movimiento.

En este ítem nos muestra por medio de un servidor en una pagina web la activación de un sensor de movimiento digital. Sí en este sensor ,esta el valor en alto nos muestra el mensaje “ Alarma ON “ y si esta el valor en bajo nos muestra el mensaje “ Alarma OFF “ , mostrándonos asi un mensaje de alarma en el servidor, como nos muestra la Figura 20.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

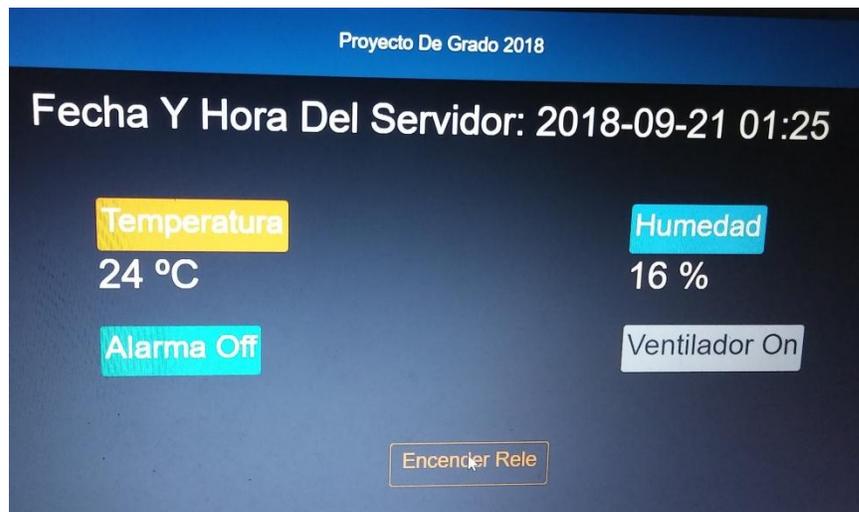


Figura 20. Sensor PIR desactivado en off

Se utilizo para este proyecto un sensor digital pir infrarojo (sensor de movimiento) Sensor ajustable que detecta movimiento al medir cambio en el infrarrojo emitido por los objetos circundantes, como se observa en Figura 21.



Figura 21. Sensor PIR IR utilizado.

Estas son sus características:

- Circuito de control.
- Sensibilidad y tiempo ajustable.
- Rango de voltaje: 4.5-20V

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Voltaje de salida: Niveles alto/bajo de señal 3.3V TTL.
- Distancia de detección: 3—7 metros (ajustable).
- Rango de detección:
- Tamaño: 32.2 mm x 24.3 mm x 18 mm (aprox)
- Tiempo detección: 5-200s(puede ser ajustado).

3.4.2 Visualización remota de valores reales de sensores de temperatura y humedad.

En este ítem nos muestra por medio de un servidor en una pagina web los valores reales y variables de de un sensor digital de temperatura y humedad como nos muestra la Figura 22.



Figura 22. Valores reales de sensor dig.

Se utiliza en sensor digital de temperatura y humedad DTH11en una sola interfaz, como se muestra en la Figura 23.



Figura 23. Sensor DTH11.

Estas son sus características:

- Voltaje de alimentación :5 V

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Rango de humedad: 20%~90%RH
- Error en la medida de Humedad: $\pm 5\%$ RH
- Rango de temperatura: 0~60°C
- Error en la temperatura: ± 2 °C

3.4.3. Activacion de un relé remotamente

En este ítem nos muestra por medio de un servidor en una pagina web la activación remota de un rele digital de 5VDC 10A con optocoplador incorporado.

Al dar click en el botón de Encender rele , este se activa , luego me aparece el mensaje apagar rele , le damos click y este se apaga como se muestra en la Figura 24.

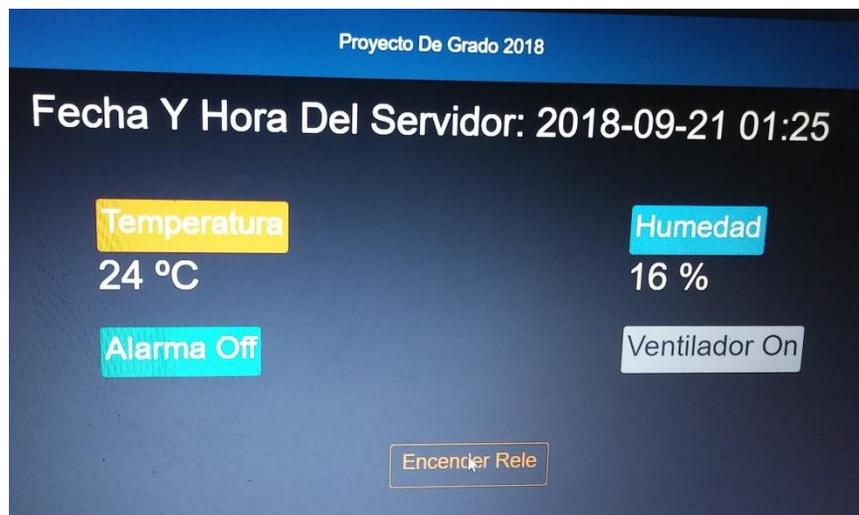


Figura 24. Rele apagado en servidor.

Se utiliza un rele de 5vdc a 30A con optoacoplador como se muestra en la Figura 25.

Estas son sus características:

- Cargas: AC250V/30A, DC30V/30A
- Voltaje de activación: 5V
- Corriente de carga: 30A
- Aislamiento por optoacoplador

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

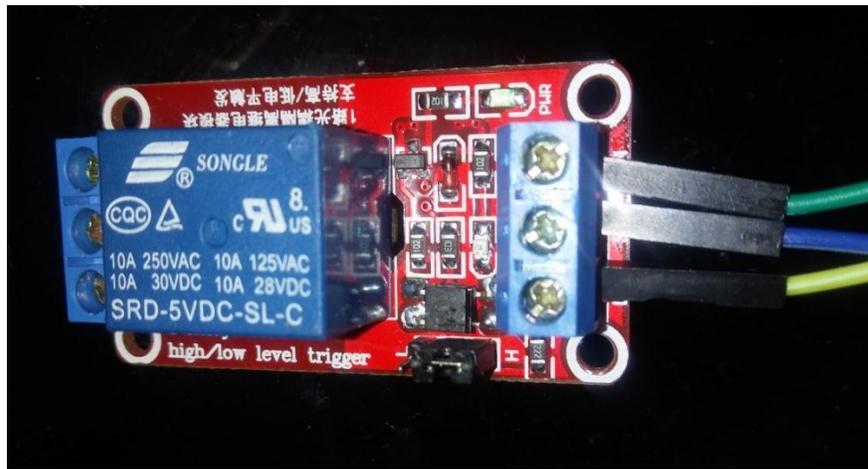


Figura 25. Relé 5v 10A con optoacoplador.

3.4.4. Control de temperatura

En este ítem nos muestra por medio de un servidor en una página web el estado de un ventilador. Este ventilador se enciende cuando el valor de temperatura supera los 23°C y se apaga cuando vuelva a un valor menor o igual a 23°C; como nos muestra la Figura 26.

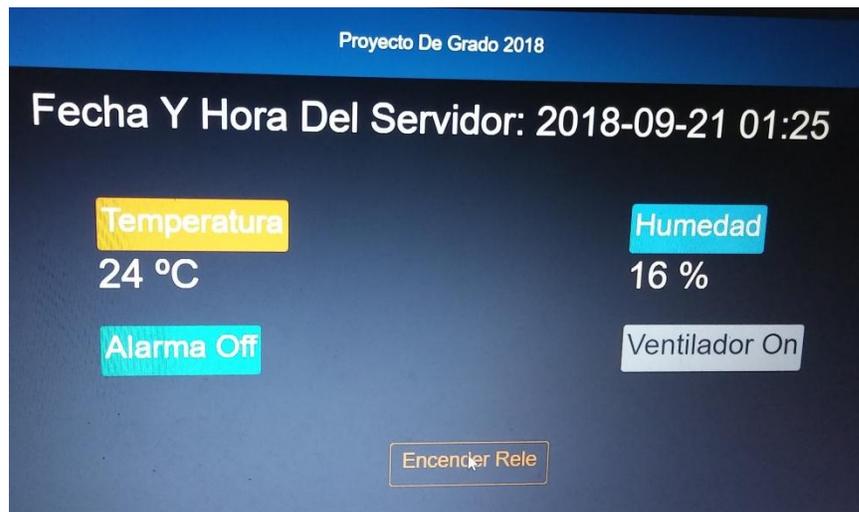


Figura 26. Valor de ventilador en ON.

Aquí se utilizó el mismo sensor anterior de temperatura y humedad y un cooler de 5vdc a 0.12A. Como se observa en la Figura 27.

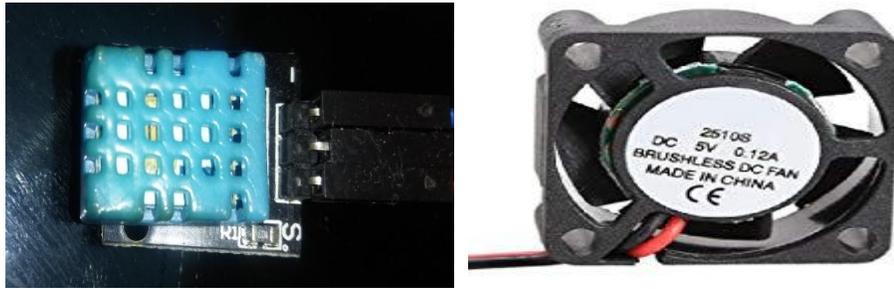


Figura 27. Sensor DTH11 y Ventilador.

Para este control de temperatura se implemento un circuito o driver de potencia que me activara el cooler cuando el sensor tomara un valor igual o superior a 23C°. Como se muestra en la Figura 28.

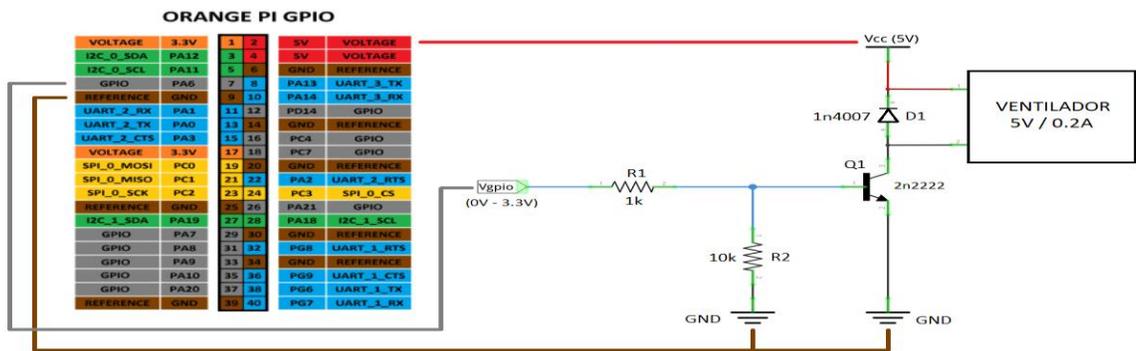


Figura 28. Diagrama Driver de potencia.

3.5 Código en Python para monitoreo y control de sensores

Para estos 4 items se realizo un solo script en PYTHON3 para que los 4 proyectos funcionen correctamente. A continuación se muestra el código del primer bloque en editor sublime3 como nos muestra en la figura Figura 29. El código restante en Python se encuentra en Anexo B.

Como van a observar es un código que consta de 134 líneas , les dare una explicación resumida del primer bloque el cual es una parte muy importante del programa.

Como vemos se hace una importación de la clase render_template de la librería flask (servidor) , tambien se importa la clase gpio y port de la librería pyA20, y se importan dos librerías , la del sensor de temperatura y humedad y la que se usa para extraer la fecha y hora de la tarjeta Orange Pi. Luego se le asignan todos los pines a usar en la tarjeta Orange Pi.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Luego se realiza una inicialización de la librería gpio (constructor), a continuación se instancia el objeto de la clase DHT de la librería dht. Se configuran los pines de cada sensor tanto de salida como de entrada , y se instancia el framework flask del servidor.

```

1  from flask import Flask, render_template
2  from pyA20.gpio import gpio
3  from pyA20.gpio import port
4  import dht
5  import datetime
6
7  DHT11_PIN = port.PA6
8  rele = port.PA1
9  pirPin = port.PA12
10 ventilador = port.PA2
11
12 gpio.init()
13 DHT11_instance = dht.DHT11(pin=DHT11_PIN)
14 gpio.setcfg(rele, gpio.OUTPUT)
15 gpio.setcfg(pirPin, gpio.INPUT)
16 gpio.setcfg(ventilador, gpio.OUTPUT)
17
18 app = Flask(__name__) #instancia del framework flask
19
20 pins = {
21     port.PA1: {'name': 'Rele', 'state': gpio.LOW}
22 }
23
24 for pin in pins:
25     gpio.setcfg(pin, gpio.OUTPUT)
26     gpio.output(pin, gpio.LOW)
27
28
29
30 @app.route("/")
31 def main():
32     global temp
33     global hum
34     global ventilador_estado
35
36
37     for PIN in pins:
38         pins[PIN]['state'] = gpio.input(PIN)
39

```

Figura 29. Script python linea1-38.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.6 . Diseño de Aplicación Web en Microframework Flask

Para la visualización de la página web desde el servidor para los botones y objetos de texto con los valores correspondientes de los sensores, se realizó un código en HTML con hoja de estilos mostrado en la Figura 30. El código de la aplicación completa se encuentra en el Anexo C.

```

1 <!DOCTYPE html>
2 <head>
3   <title>ProyectoDeGrado2018</title>
4   <meta charset="utf-8">
5   <link rel="stylesheet" type="text/css" href="https://bootswatch.com/4/darkly/
bootstrap.min.css">
6   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js" integrity=
"sha384-0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxs1yVq0tnepnHVP9aJ7xS" crossorigin="
anonymous"></script>
7 </head>
8
9 <style type="text/css">
10
11   body{font-family: Arial; box-sizing: border-box;}
12
13   .navbar-header
14   {
15     float: left;
16     padding: 15px;
17     text-align: center;
18     width: 100%;
19   }
20
21   .navbar-brand
22   {
23     float:none;
24   }
25
26   div#body
27   {
28     margin: auto;
29     margin-top: 60px;
30     width: 700px;
31     height: 70px;
32   }

```

Figura 30. Script html linea 1-32.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.7. Sistema de vigilancia con web cam

En este proceso , nos vamos a dar cuenta como por medio de comandos en nuestra terminal para escritura de comandos, podemos configurar una cámara web para que nos de visual en directo y grabe por movimiento o la configuración por fotos o continua.

La cámara web utilizada fue una Logitech como 720p la podemos ver en la Figura 31.



Figura 31. Camara web 720p.

Las características de la cámara son:

- Resolución máx.: 720p/30 fps
- Tipo de enfoque: foco fijo
- Tecnología de lente: estándar
- Micrófono integrado: mono
- Campo visual: 60°
- Clip universal para monitores, pantallas LCD o laptops
- Longitud de cable: 1.5 mt.

Todo se configura directamente desde la terminal emulador de Armbian. (Pi O. , 2018). Primero vamos a comprobar que la cámara está instalada, reconocida y funcionando con el comando lsub, como se muestra a continuación:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

pi@orangepi ~ $ lsusb

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 003: ID 0ac8:3420 Hewlett Packard
Bus 001 Device 002: ID 1a40:0101 Terminus Technology Inc. 4-Port HUB

pi@orangepi ~ $ ls /dev/video*
/dev/video0

```

En este paso así comprobamos que todo funciona correctamente. Si no tiene el Mplayer instalado , entonces se debe instalar con el comando apt-get como se muestra a continuación:

```

sudo apt-get install mplayer

```

Se ejecuta mplayer para usar la Webcam y capturar vídeo:

```

sudo mplayer tv://

```

Deberá abrirse una ventana con la captura de vídeo en directo.

Ya se comprobo que la cámara funciona y es capaz de capturar vídeo. Vamos ya por la detección de movimiento. Necesitamos instalar un software llamado Motion desarrollado por Kenneth Lavrsen (Lavrsen, 2018).

```

sudo apt-get install motion

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Cuando termine de instalar el software motion, hay que hacer algunos cambios. Abre el fichero *motion.conf* con tu editor preferido:

```
sudo nano /etc/motion/motion.conf
```

Busca las siguientes líneas y cambia:

```
"Daemon = off" por "on"
```

Se copia el directorio donde se grabarán las imágenes:

```
target_dir /home/orangepi/motion
```

Este directorio debe estar creado y tener todos los derechos de escritura, así que créalo si no existe y dale permisos totales. Desde el directorio */home/orangepi/*:

```
sudo mkdir motion
sudo chmod 777 motion
```

Para tener acceso desde el exterior tenemos que cambiar varias líneas. La primera es para acceder desde fuera de localhost y las siguientes para establecer un usuario y contraseña por seguridad; por lo cual lo hice y solo yo puedo acceder a ella en el servidor.

```
stream_localhost off
stream_auth_method 1
# cambia usuario:password por los de tu elección
stream_authentication usuario:password
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Desde el puerto 8080 se puede ver y editar la configuración del programa, pero sólo se podrá editar desde local. En este archivo están casi todos los parámetros de configuración del programa. Guarda y sal del editor.

Para asegurarnos que el servicio de detección arranca como un servicio de Linux automáticamente cuando se reinicia el sistema, necesitamos hacer otro cambio de configuración:

```
sudo nano /etc/default/motion
```

y cambiamos:

```
"start_motion_daemon=no" por "yes"
```

Guarda y sal del editor. Para terminar, lanzamos el programa:

```
sudo service motion start
```

y comprobamos que podemos ver la imagen accediendo al servidor local: en mi caso es 192.168.1.58:8081

A continuación adjunto dos imágenes de la cámara como se muestra en la Figura 32 y Figura 33 de las pruebas hechas en la oficina L304 en Fraternidad con el profesor Alexander Arias.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

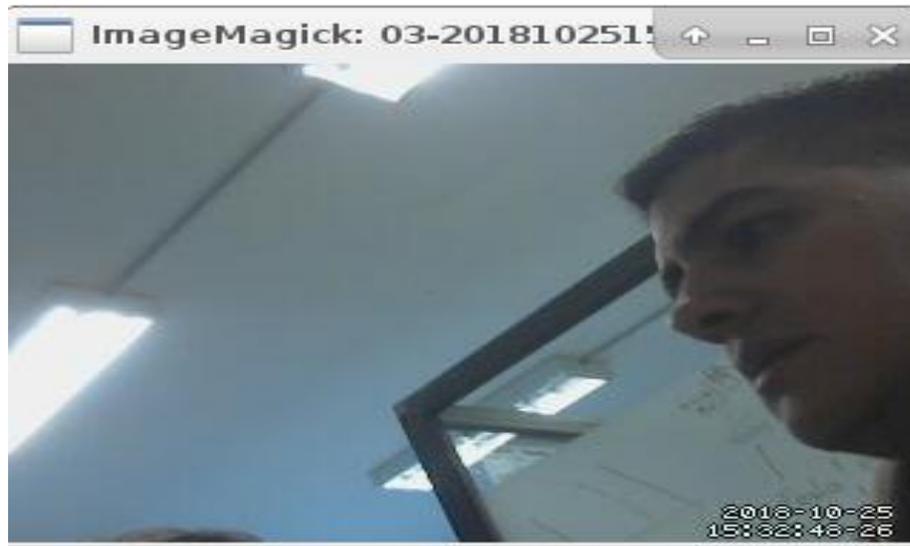


Figura 32. Imagen 1 camara web.

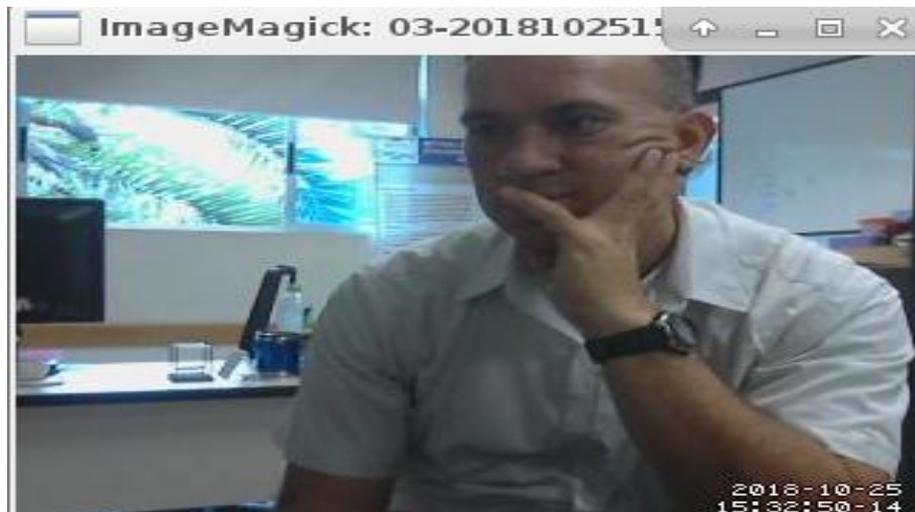


Figura 33. Imagen 2 camara web.

3.8. XRDP Escritorio remoto en Armbian

Las aplicación XRDP se utiliza con el fin de hacer conexiones remotas sin necesidad de conectar los periféricos de pantalla, teclado y mouse a la tarjeta Orange Pi y mas bien por medio de la antena

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

WIFI obtener una IP publica y abrirla remotamente. Esta aplicación nos permite manipular el escrito remoto de la interfaz gráfica de Armbian (neutrinolabs, 2018).

Remote desktop protocol también llamado RDP , Remote desktop services, terminal services o conexión de escritorio es un protocolo propietario desarrollado por Microsoft que permite conectarse y acceder a un ordenador de forma remota desde otro ordenador local. RDP se implementa con arquitectura cliente -servidor , por lo que necesita un servidor RDP que escuche en el puerto 3389 TCP por defecto y que atienda las peticiones de los clientes ; un cliente RDP que permita conectar con la maquina remota con la que corre el servidor RDP y en Linux existe esta implementación de servidor RDP llamada XRDP. (videlcloud, 2018)

Si abrimos el escritorio remoto de Windows como se ve en la Figura 34 , digitamos la dirección IP en este caso de la tarjeta orange pi. Despues se abre una coexión con XRDP como muestra la Figura 35. Allí digitamos el respectivo usuario y contraseña, para abrirnos el escritorio remoto con interfaz gráfica de Armbian Figura 36.

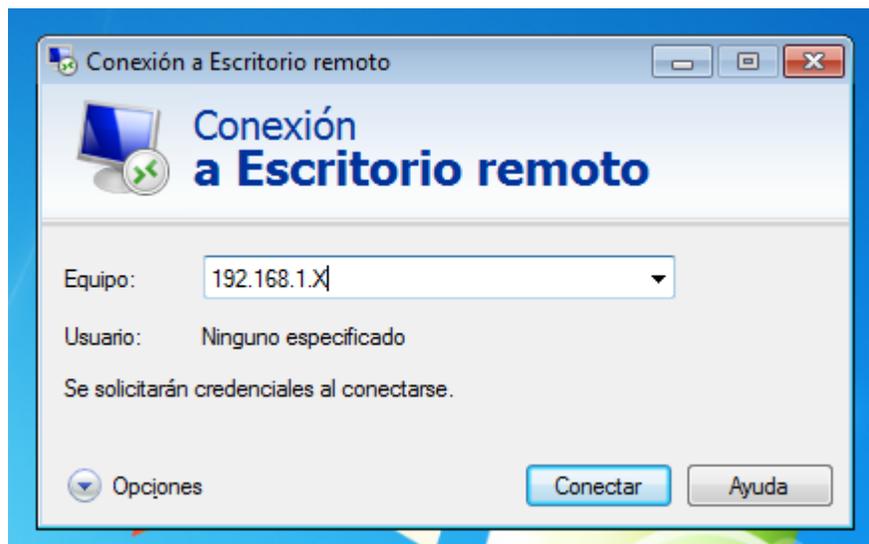


Figura 34. Abrir Escritorio Remoto desde Windows.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

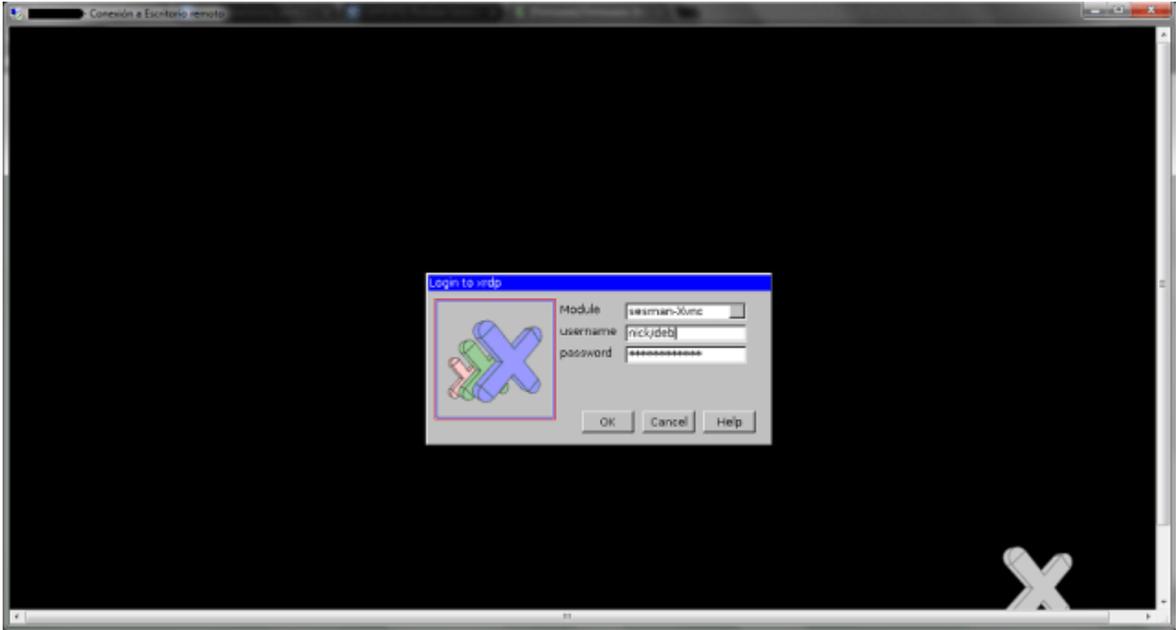


Figura 35. XRD escritorio remoto linux.

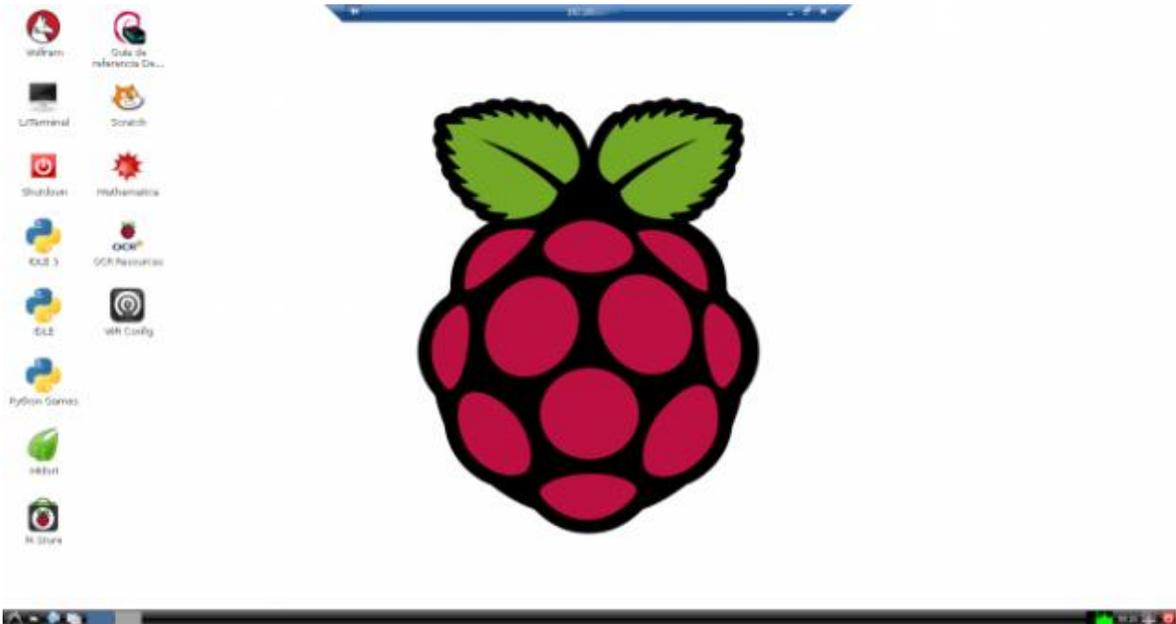


Figura 36. Escritorio Remoto de Armbian con Interfaz Gráfica.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESULTADOS Y DISCUSIÓN

En el proyecto se configuraron y programaron cada uno de los sensores y actuadores a la tarjeta Orange Pi, como el PIR, el sensor de temperatura y humedad, el rele con optoacoplador y el circuito o driver de potencia en cual activa el cooler. Todo este sistema se conecta a la web por medio de wifi y se puede visualizar tanto en PC como en teléfono móvil, como se muestra en la **¡Error! No se encuentra el origen de la referencia..**

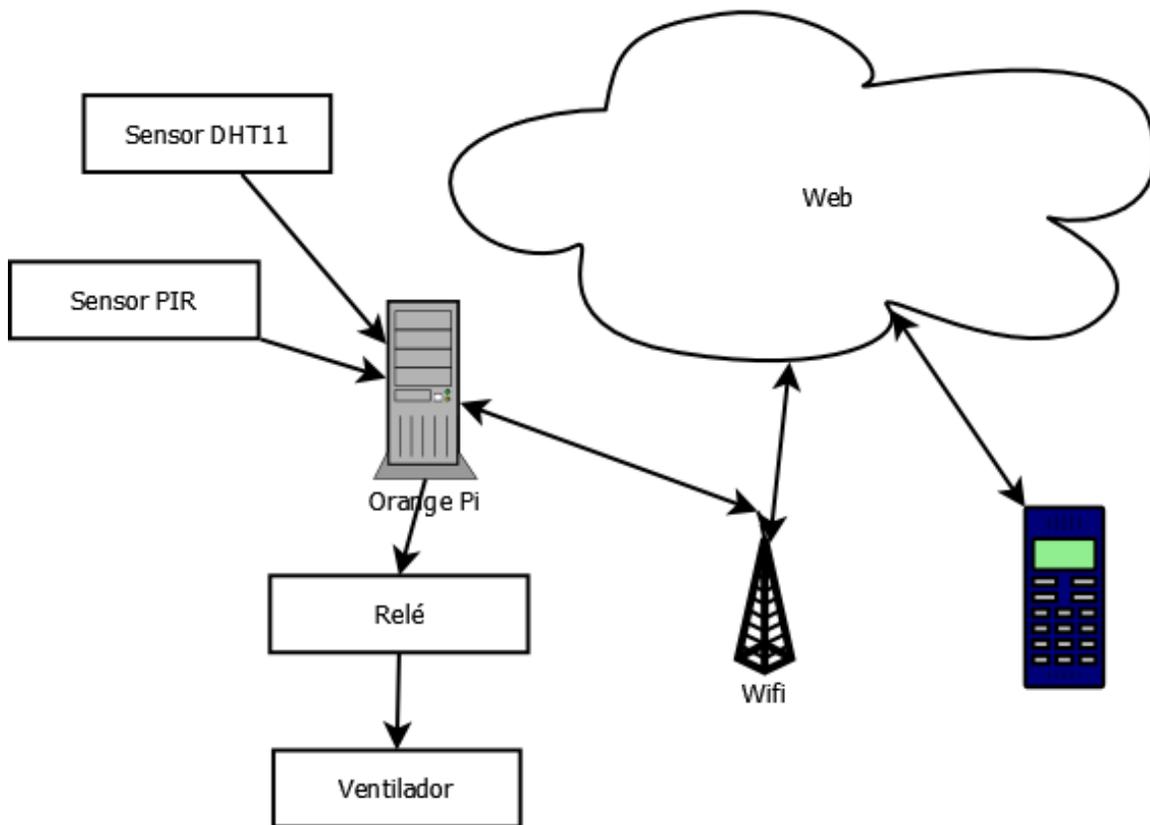


Figura 37. Diagrama del proyecto con los sensores y actuador conectados.

Se implemento el monitoreo y control remoto desde la web, lo cual configuro un sistema IoT. La interfaz de usuario se muestra en la Figura 38 ejecutada desde un celular. Se empleo la conexión de la tarjeta por medio de Wifi, también se puede emplear la conexión Ethernet por cable. Se identifica la dirección IP que adquiere la tarjeta por medio de la Wifi donde estemos; luego se digita

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

en el escritorio remoto. La reconfiguración y desarrollo de aplicaciones se hace por medio de escritorio remoto XRDP de Linux y Python 3. Se implemento un driver de potencia para la activación del ventilador en el control de temperatura.

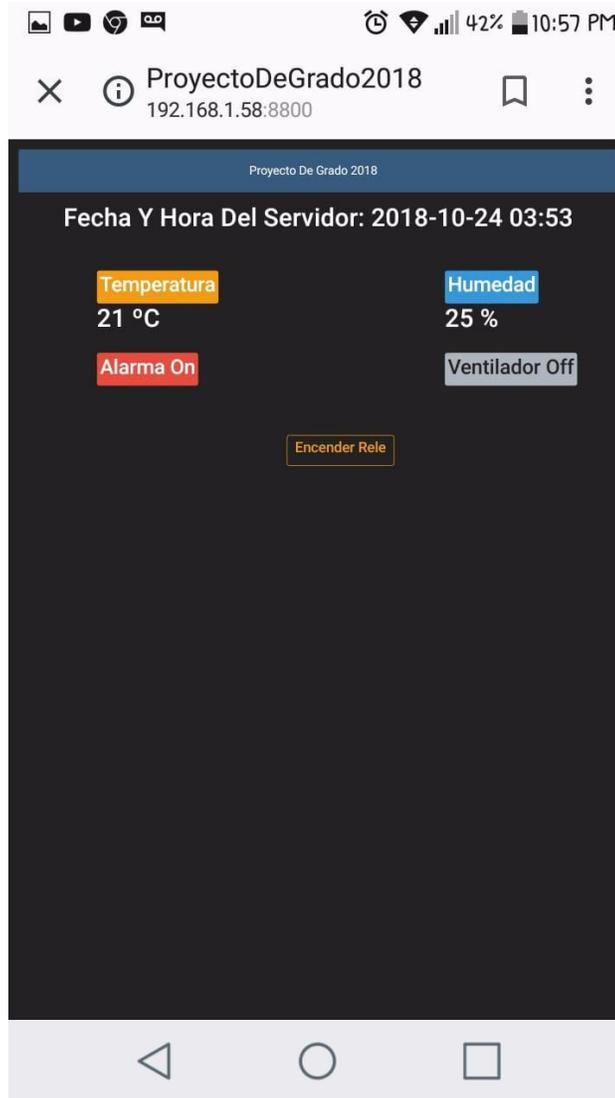


Figura 38. Interfaz de usuario en movil.

En el diseño de la pagina web , esta interfaz es la adecuada debido a la cantidad minima de sensores y variables para visualizar y controlar, igualmente se puede mejorar implementando otros colores , tamaño , estilo de letra y fondo de estos iconos. Al sistema también se les puede añadir mas cantidad de sensores de toda clase como de proximidad, de posición,etc.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

A este tipo de rele se le puede instalar cualquier dispositivo o maquina porque tiene un rango amplio de voltaje y corriente, tales como un bombillo , un motor paso a paso,etc y al activarlo va entrar en funcionamiento y se puede controlar activando o desactivándolo desde el servidor con un solo klik.

Con respecto al control de temperatura , se le puede adaptar un driver de una potencia mas alta para que por medio de un ventilador mas potente genere mas viento frio y se pueda establecer mas rápido el control de temperatura ; también el código se puede modificar y se puede dejar otro valor o rango de temperatura en °C para que controle el sistema de refrigeración o calor de cierta maquina o dispositivo a automatizar.

Esta es una explicación de como se interactua en la pagina web cuando estos sensores están en funcionamiento:

Cuando el sensor PIR Infrarojo detecta algún movimiento , aparecerá la alarma en ON, de lo contrario permanecerá en OFF. El sensor de temperatura y humedad siempre mostrara valores reales de temperatura en C° y humedad en %. Cuando el control de temperatura se active al superar el valor mayor o igual a 23C° , el ventilador queda encendido en ON , de lo contrario quedara en OFF. Y la activación del rele se realiza dando un clic en el icono encender rele , inmediatamente aparecerá en el icono apagar rele para poderlo desactivar.

Con respecto a la cámara , esta no es si no conectarla y digitar en el navegador la dirección ip en mi caso 192.168.1.58:8081 con puerto 8081 y ella inmediatamente me pedira un usuario y contraseña y abrirá la imagen en el servidor, y empezara a grabar cuando detecte un movimiento.

Al proyecto se le adapto una maqueta con la ubicación de todos los sensores y se expuso en la semana de la ingeniería logrando asi una exposición con concetos de IoT a evaluadores y estudiantes

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

- Se aplico IoT (Internet of Things), y se logro tener visualización en una pagina web del estado y control de todos los sensores conectados y programados en la tarjeta mini-ordenadora Orange Pi por medio de un servidor, en este caso Flask de Python.
- Se implemento y se configuro el sistema operativo Armbian de Linux , ya que es compatible con la tarjeta en mi caso la Orange Pi Pc Plus; ya que este sistema operativo es ligero orientado específicamente a su utilización en chipset ARM, el chipset que incorporan la mayoría de placas de desarrollo que tenemos hoy en día. Se logra flashear la imagen del sistema operativo en la tarjeta micro SD, obteniendo asi el sistema operativo instalado y funcionando correctamente
- Se utilizo e instalo el microframework Flask , servidor de Python ya que este nos enfoca en proporcionar lo mínimo necesario para que podamos poner a funcionar una aplicación básica en cuestión de minutos. Y lo mas interesante es que incluye un servidor web de desarrollo para poder probar aplicaciones sin tener que instalar un servidor web.
- Por medio de este servidor , identificamos primero que todo la dirección IP que tenga la tarjeta, le añadimos el puerto , lo digitamos en el navegador y obtenemos como producto final una pagina web con interfaz flask para adquisición de variables y control de cargas.

IoT trabaja con la interconexión digital de los objetos cotidianos con internet , conformado de múltiples tecnologías como sensores de los cuales podemos utilizar mucha mas variedad de ellos que nos permiten conectar el mundo físico con el digital, computadores o en este caso mini-ordenadores que permiten procesar esa información y plataformas web donde se procesan y almacenan los datos, en este trabajo se implemento el minipc con la Tarjeta Orange Pi.

Para aplicaciones industriales se puede implementar una infraestructura de red inteligente mejorando asi las operaciones en una empresa, aumentando la seguridad, protección y productividad . Además, permite obtener una perspectiva valiosa de datos para optimizar la automatización.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En general, la conexión de productos por medio de IoT, ofrece la posibilidad de tener información directamente desde los usuarios lo que es invaluable para su mejora continua y trabajo en tiempo real.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

- Arduino. (2018). *Arduino*. Obtenido de <https://www.arduino.cc/en/Guide/Introduction>
- beagleboard. (2018). Obtenido de beagleboard.org:
<https://mecatronicauasp.wordpress.com/2014/02/28/introduccion-a-beaglebone/>
- electronicas, D. (s.f.). *Didacticas electronicas*. Obtenido de <https://www.didacticaselectronicas.com/index.php/sistemas-de-desarrollo/orange-pi/tarjetas/mini-pc-orange-pi-pc-plus-orange-pi-pc-plus-tarjeta-raspberry-pi-detail>
- GUIADEV. (s.f.). *GUIADEV*. Obtenido de <https://guiadev.com/tutorial-flask-microframework-python-parte-01/>
- Lavrsen, K. (2018). *Kenneth Jahn Lavrsen's Home Page*. Obtenido de <https://www.lavrsen.dk/>
- lomelisan. (2018). *lomelisan*. Obtenido de <https://lomelisan.wordpress.com/2015/05/29/mega-tutorial-de-flask-python-1era-parte-hola-mundo/>
- neutrinolabs. (2018). *An open source remote desktop protocol(rdp) server*. Obtenido de <http://www.xrdp.org/>
- orangepi. (2018). *ORANGEPI*. Obtenido de <https://orangepiweb.es/>
- Pérez Santos, J. (2018). *Orangepiweb*. Obtenido de <https://orangepiweb.es/gpio.php>
- PI, O. (2018). Obtenido de ORANGE PI: <http://www.orangepi.org/>
- Pi, O. (2018). *ORANGEPI*. Obtenido de <https://orangepiweb.es/vigilancia.php>
- Pi, R. (2018). *Raspberry Pi*. Obtenido de <https://www.raspberrypi.org/>
- soloelectronicos. (2018). *soloelectronicos*. Obtenido de <http://soloelectronicos.com/2017/08/08/como-instalar-armbian-en-orange-pi-pc/>
- videlcloud. (2018). *XRDP*. Obtenido de <https://videlcloud.wordpress.com/2017/10/21/escritorio-remoto-en-linux-fedora-con-xrdp/>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ANEXOS

Para que los que lean este texto tengan la oportunidad de trabajar o adaptar sus propios códigos y que realicen proyectos con la Orange Pi, se incluye en el Anexo A la configuración paso a paso de las GPIO de la Orange Pi, en el Anexo B el código de Python con microframework Flask y en el Anexo C la aplicación web con html. Los otros anexos son evidencias de la participación en la semana de la Ingeniería y el certificado.

Anexo A. Configuración paso a paso de la GPIO para encender un Led en la Orange Pi

La librería para instalarlo con Python es, o se puede clonar con: `sudo git clone https://github.com/duxingkei33/orangepi_PC_gpio_pyH3.git`

Accede a la carpeta recién creada e instala con:

```
sudo python setup.py install
```

Importante: Si te sale un error sobre python.h

```
fatal error: Python.h No such file or directory
#include "Phyton.h"
...
```

tienes que instalar el paquete python-dev.

```
sudo apt-get install python-dev
```

Después vuelve a instalar.

```
sudo python setup.py install
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Ahora para probar que todo funciona, este es un ejercicio sencillo de la activación de un led. Crea un archivo donde escribiremos un script en python con sudo nano test.py y escribe el siguiente código.

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

import os
import sys

if not os.getegid() == 0:
    sys.exit('Debes ejecutar este Script como SU')

from time import sleep
from pyA20.gpio import gpio
from pyA20.gpio import port

led = port.PA20

gpio.init()
gpio.setcfg(led, gpio.OUTPUT)

try:
    print ("Pulsa CTRL+C para salir")
    while True:
        gpio.output(led, 1)
        print "led set 1 \r\n"
        sleep(2)
        gpio.output(led, 0)
        print "led set 0 \r\n"
        sleep(2)
```

```
    gpio.output(led, 1)
    sleep(0.1)
    gpio.output(led, 0)
    sleep(0.1)
    sleep(0.6)

except KeyboardInterrupt:
    print ("Saliendo.")

/env python

import os
import sys

if not os.getegid() == 0:
    sys.exit('Debes ejecutar este Script como SU')

from time import sleep
from pyA20.gpio import gpio
from pyA20.gpio import port

led = port.PA20

gpio.init()
gpio.setcfg(led, gpio.OUTPUT)

try:
    print ("Pulsa CTRL+C para salir")
    while True:
        gpio.output(led, 1)
```

```

print "led set 1 \r\n"
sleep(2)
gpio.output(led, 0)
print "led set 0 \r\n"
sleep(2)
gpio.output(led, 1)
sleep(0.1)
gpio.output(led, 0)
sleep(0.1)
sleep(0.6)

except KeyboardInterrupt:
    print ("Saliendo.")

```

Ahora con una resistencia, 2 cables y un diodo led, conecta como en la Figura 39.

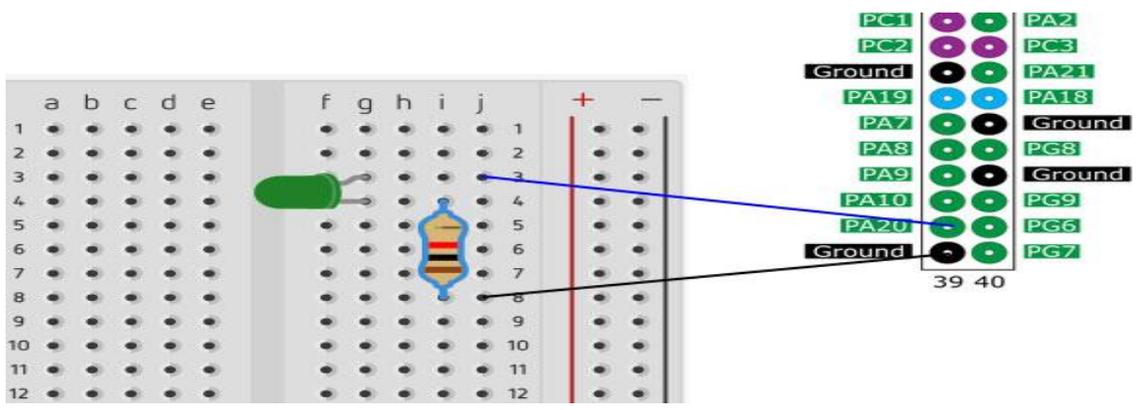


Figura 39. Conexion para encender led.

Ejecuta sudo python test.py y el led debe parpadear además de mostrar el estado por pantalla.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Anexo B. Código completo de la aplicación en Python 3.0

```

from flask import Flask, render_template
from pyA20.gpio import gpio
from pyA20.gpio import port
import dht
import datetime

DHT11_PIN = port.PA6
rele = port.PA1
pirPin = port.PA12
ventilador = port.PA2

gpio.init()
DHT11_instance = dht.DHT11(pin=DHT11_PIN)
gpio.setcfg(rele, gpio.OUTPUT)
gpio.setcfg(pirPin, gpio.INPUT)
gpio.setcfg(ventilador, gpio.OUTPUT)

app = Flask(__name__) #instancia del framework flask

pins = {
    port.PA1: {'name': 'Rele', 'state': gpio.LOW}
}

for pin in pins:
    gpio.setcfg(pin, gpio.OUTPUT)
    gpio.output(pin, gpio.LOW)

@app.route("/")
def main():
    global temp
    global hum
    global ventilador_estado

    for PIN in pins:
        pins[PIN]['state'] = gpio.input(PIN)

    now = datetime.datetime.now()
    time_string = now.strftime("%Y-%m-%d %H:%M")
    dht11_result = DHT11_instance.read()

    if dht11_result.is_valid():
        temp = dht11_result.temperature
        hum = dht11_result.humidity
    else:
        dht11_result = DHT11_instance.read()

    #global temp
    #global hum

```

```
alarma status = gpio.input(pirPin)

if temp > 23:
    gpio.output(ventilador, gpio.HIGH)
    ventilador_estado = True
else:
    gpio.output(ventilador, gpio.LOW)
    ventilador_estado = False

#global ventilador_estado

template_data = {
    'pins': pins,
    'time': time_string,
    'temperatura': temp,
    'humedad': hum,
    'pir': alarma_status,
    'status_fan': ventilador_estado
}

return render_template('botontemp.html', **template_data)

@app.route("/<change_pin>/<action>")
def action(change_pin, action):
    global temp
    global hum
    global ventilador_estado

    now = datetime.datetime.now()
    timeString = now.strftime("%Y-%m-%d %H:%M")
    DHT11_result = DHT11_instance.read()

    change_pin = int(change_pin)

    #deviceName = pins[change_pin]['name']

    if action == "on":
        gpio.output(change_pin, gpio.HIGH)

    if action == "off":
        gpio.output(change_pin, gpio.LOW)

    for pin in pins:
        pins[pin]['state'] = gpio.input(pin)

    if DHT11_result.is_valid():
        temp = DHT11_result.temperature
        hum = DHT11_result.humidity
    else:
        DHT11_result = DHT11_instance.read()

#global temp
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

#global hum

if temp > 23:
    gpio.output(ventilador, gpio.HIGH)
    ventilador_estado = True
else:
    gpio.output(ventilador, gpio.LOW)
    ventilador_estado = False

#global ventilador_estado

alarma_status = gpio.input(pirPin)

templateData = {
    'pins': pins,
    'time': timeString,
    'temperatura': temp,
    'humedad': hum,
    'pir': alarma_status,
    'status_fan': ventilador_estado
}

return render_template('botontemp.html', **templateData)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8800, debug=True)

```

Anexo C. Código aplicación web con bootstrap y html con css.

```

<!DOCTYPE html>
<head>
  <title>ProyectoDeGrado2018</title>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css"
href="https://bootswatch.com/4/darkly/bootstrap.min.css">
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"
integrity="sha384-
0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELQxss1yVqOtnepnHVP9aJ7xS"
crossorigin="anonymous"></script>
</head>

<style type="text/css">

  body{font-family: Arial; box-sizing: border-box;}

  .navbar-header
  {

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

float: left;
padding: 15px;
text-align: center;
width: 100%;
}

.navbar-brand
{
float:none;
}

div#body
{
margin: auto;
margin-top: 60px;
width: 700px;
height: 70px;
}
div#body2
{
margin: auto;
width: 100px;
height: 70px;
}

div#temperature-box
{
width: 150px;
height: 50px;
float: left;
}

div#rele-box
{
width: center;
height: 50px;
margin-top: 60px;
float: center;
}

div#humidity-box
{
width: 150px;
height: 50px;
float: right;
}

div#textos
{
text-align: center;
}

</style>

```

```

<script type = "text/javascript">
    setTimeout(function(){
        location = ''
    },1500)
</script>

<body>
    <nav class="navbar navbar-default " role="navigation" >
        <div class="navbar-header navbar-expand-lg navbar-dark bg-
primary">
            <a class="navbar-brand" href="#">Proyecto De Grado
2018</a>
        </div>
    </nav>

    <div id="textos">
        <h2>Fecha Y Hora Del Servidor: {{ time }}</h2>
    </div>

    <div id="body">
        <div id="temperature-box">
            <span class="badge badge-warning"> <h3>Temperatura </h3>
</span>
            <h2> {{ temperatura }} °C</h2>
        </div>

        <div id="humidity-box">
            <span class="badge badge-info"> <h3>Humedad </h3> </span>
            <h2> {{ humedad }} %</h2>
        </div>
    </div>

    <div id="body">
        <div id="temperature-box">

            {% if pir == true %}
            <span class="badge badge-danger"> <h3>Alarma On </h3> </span>

            {% else %}

            <span class="badge badge-success"> <h3>Alarma Off </h3>
</span>

            {% endif %}
        </div>

        <div id="humidity-box">

            {% if status_fan == true %}
            <span class="badge badge-dark"> <h3>Ventilador On </h3>
</span>

            {% else %}

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

    <span class="badge badge-dark"> <h3>Ventilador Off </h3>
</span>

    {% endif %}
</div>

</div>

{% for pin in pins %}

<div id="body2">
  <div id="rele-box">

    {% if pins[pin].state == true %}
    <a href="/{{pin}}/off" class="btn btn-warning"
role="button"><h4>Apagar Rele</h4></a>

    {% else %}
    <a href="/{{pin}}/on" class="btn btn-outline-warning"
role="button"><h4>Encender Rele</h4></a>
    {% endif %}
  </div>
</div>

{% endfor %}

</body>
</html>

```

Anexo D. Participación en Evento de la Semana de Ingeniería del ITM 2018

Se participo en la semana de la ingenieria, el dia 6 de noviembre del 2018 , se expone el proyecto ante los evaluadores y estudiantes, este fue el poster, como lo muestra la Figura 40.



Figura 41. Exposicion proyecto1.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



Figura 42. Exposición proyecto 2.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

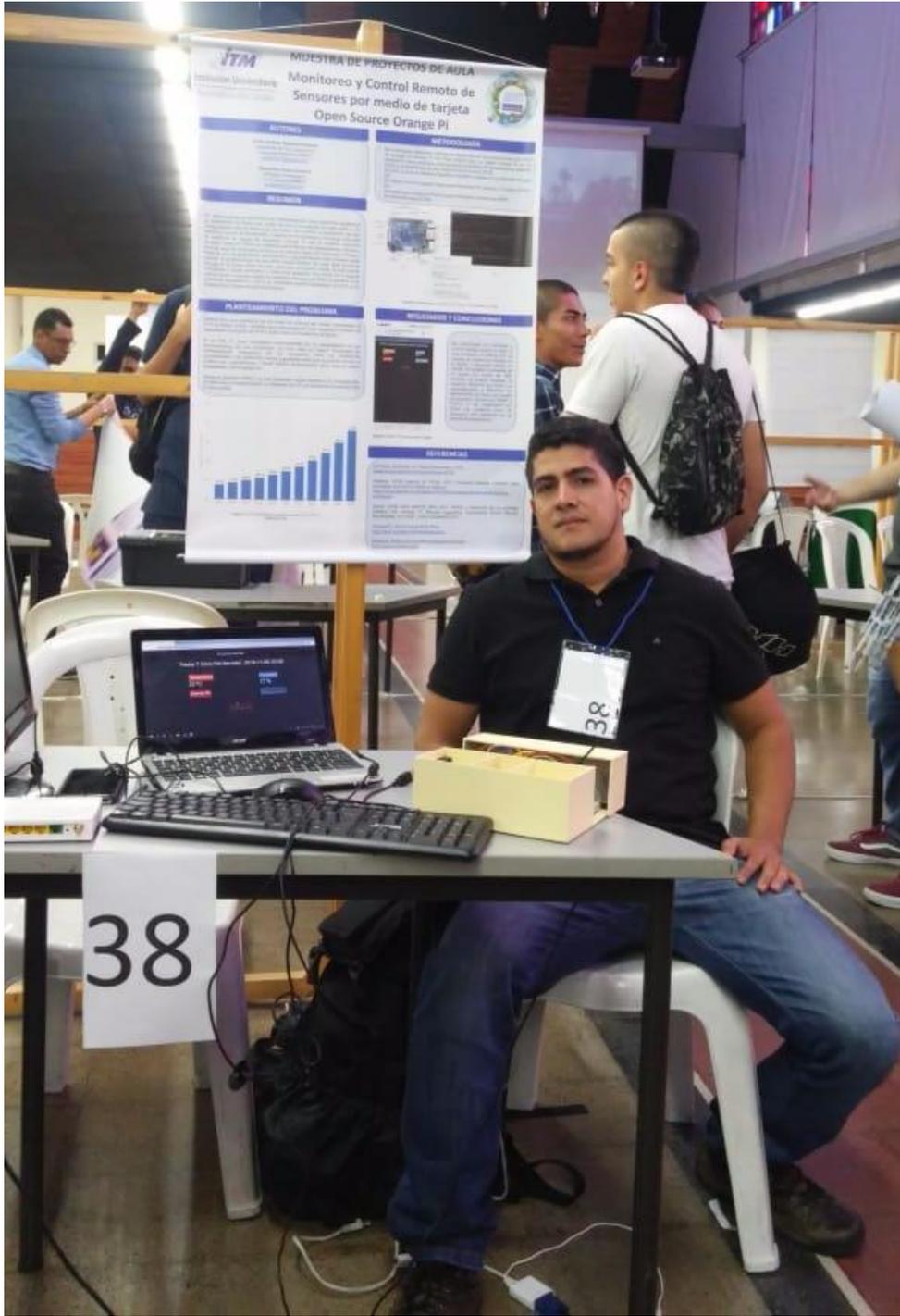


Figura 43. Exposición proyecto 3.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Anexo E. Certificado de Participación en Evento de la Semana de Ingeniería del ITM 2018



Figura 44. Certificado Participación en Evento.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES FIRMA ASESOR	_____ _____ _____ _____
FECHA ENTREGA: _____	

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD	_____
RECHAZADO _____	ACEPTADO _____
CON MODIFICACIONES _____	
ACTA NO. _____	
FECHA ENTREGA: _____	

FIRMA CONSEJO DE FACULTAD	_____
ACTA NO. _____	