DISEÑO DE UN SISTEMA DE ADQUISICIÓN Y VISUALIZACIÓN DE DATOS BASADO EN LA PLATAFORMA DE SISTEMAS EMBEBIDOS RASPBERRY PI

JEYSSON EMILIO LIMA GUAQUETA

PEDRO LEONARDO OSPINA FUENTES

UNIVERSIDAD SANTO TOMAS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTA D.C
2018

DISEÑO DE UN SISTEMA DE ADQUISICIÓN Y VISUALIZACIÓN DE DATOS BASADO EN LA PLATAFORMA DE SISTEMAS EMBEBIDOS RASPBERRY PI

JEYSSON EMILIO LIMA GUAQUETA

PEDRO LEONARDO OSPINA FUENTES

EDUARD GALVIS RESTREPO, PhD
ASESOR

UNIVERSIDAD SANTO TOMAS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTA D.C
2018

Agradecimientos

A Dios que fue mi fuerza y esperanza en los sueños que me propuse, mis padres, Rosaura y Pablo por brindarme su apoyo hasta el final y ser únicos en su forma de amarme. A mi hermana por siempre apoyarme sin importar la distancia. A Alejandra porque tu compañía me hizo seguir adelante. A mis amigos, que fueron incondicionales todo el tiempo. Y, mi tutor de proyecto que siempre brindo una mano y acompañamiento para llegar al final.

Jeysson.

En principio y sobre todo quiero darle gracias a Dios por darme vida y un gran carácter para poder superar todas las dificultades. Gracias a mis amados Padres, Pedro Nel y Luz Mery porque gracias a su formación y valores me han dado herramientas para avanzar en la vida, animándome insistentemente a seguir mis sueños apoyándome en toda mi formación profesional. A mis hermanas quienes han sido mi apoyo incondicional y parte fundamental de mi vida para hacerme reír, llorar, y ver dificultades desde otra perspectiva, finalmente gracias a todos mis maestros por su valioso aporte a mi formación académica.

Pedro.

NOTA DE ACEPTACIÓN

Aprobado por el comité de grado de la facultad de ingeniería electrónica en cumplimiento de los requisitos exigidos por la Universidad Santo Tomas para optar al título de Ingeniería Electrónica.

_____

EDUARD GALVIS RESTREPO, PhD
ASESOR DE PROYECTO

_____

DARIO ALEJANDRO SEGURA TORRES, MSc
FIRMA DE JURADO

_____

CAROLINA HIGUERA ARIAS, MEng
FIRMA DE JURADO

Bogotá, mayo de 2018

CONTENIDO

LISTA DE IMAGENES

LISTA DE TABLAS

LISTA DE ANEXOS

1. PRESENTACION DEL PROYECTO

1.1 INTRODUCCIÓN

El internet de las cosas es un tema en estudio en la actualidad, que consiste en la interconectividad de objetos cotidianos con el internet, esto con el fin de dotar nuevas funcionalidades que permitan controlar o administrar de forma remota dichos objetos. Gracias a esto se crean nuevos usos y aplicaciones en todos los ámbitos de la vida humana, desde la vivienda, ciudad, salud, industria consumo, etc. [61].

El presente proyecto busca fomentar e incentivar el uso de software libre y aplicaciones del IoT (Internet of Things de sus siglas en inglés). Esto con el objetivo de aportar nuevos métodos de implementación e integración en sistemas embebidos para el internet de las cosas, permitiendo la creación de nuevas estrategias de bajo costo y fomentar procesos óptimos y portables para el tratamiento y la adquisición de datos a partir de sensores por medio de software libre. Por tanto, se planteó realizar un aplicativo que permita a partir de estudios de ergonomía ambiental determinar el confort térmico de un lugar determinado, donde se plantea utilizar métodos de estudio en los que señalan zonas de bienestar a partir de humedad relativa y temperatura del aire.

La estructura de este documento está dividida por seis capítulos siendo el primero, la presentación del proyecto, que está conformada por subcapítulos que contienen, una introducción que presenta un breve resumen sobre el planteamiento y desarrollo del proyecto, antecedentes que se mencionaran trabajos relacionados a este proyecto, el problema que este proyecto busca solucionar, la justificación del proyecto en donde se va a argumentar la importancia del confort térmico en el internet de las cosas, objetivos que consiste en la presentación del objetivo general

y específicos. El segundo capítulo es el marco conceptual, que explica los conceptos necesarios para comprender el proyecto, referentes al confort térmico, el internet de las cosas, sistemas embebidos, el lenguaje de programación Python y por último la interface de aplicación en la nube. El tercer capítulo es el diseño y ejecución del proyecto donde expone los requerimientos y desarrollo para la interfaz gráfica del proyecto. El cuarto capítulo explica los resultados obtenidos, mencionando las ventajas y aportes de este proyecto para estudios en ergonomía ambiental. El quinto capítulo es el impacto social, con el que se argumentan los usos en los que se puede aplicar este proyecto. El sexto capítulo presenta las conclusiones que surgen a raíz de este proyecto, como también las propuestas y sugerencias a trabajo futuro. A su vez, en el final de este documento se presenta bibliografía y anexos del proyecto.

1.2 ANTECEDENTES

El internet de las cosas ha sido un concepto de estudio en la última década a nivel global, debido a la gran diversidad de campos que abarca dentro del sector salud, industrial, ambiental y control, lo que incentivo en la universidad Santo Tomás a que se iniciara una serie de investigaciones a partir del IoT, generando propuestas en domótica y protocolos unificados [1], investigaciones del panorama frente a la aplicación del internet de las cosas [2] y aplicaciones de cómputo móvil para el monitoreo de la diabetes [3]. El concepto del Internet de las cosas ha iniciado una serie de tendencias en temas de investigación para la educación superior en Colombia y a nivel internacional orientada a fomentar la creación de nuevas aplicaciones, como garantizar la seguridad y privacidad en IoT [4], mejorar la velocidad de tomas de datos dentro de la industria [5] y aplicaciones en el sector salud [6].

El concepto del IoT es un tema de investigación que tomo importancia hace menos de 6 años, que busca la optimización y mejoramiento de procesos en diferentes campos relacionados con el internet de las cosas, como la arquitectura de software en objetos inteligentes [7], captación de energía en el IoT y el manejo de datos provenientes de sensores [8].

La necesidad de mejorar los sistemas y buscar su optimización, ha generado la tendencia de utilizar plataformas de desarrollo basadas en dispositivos embebidos. Esta tendencia busca que el IoT permita que un sistema sea portátil para suplir necesidades como:

- La seguridad de un automóvil [9].
- La supervisión en carga en zonas no residenciales [10]
- Automatización de máquinas usando un Raspberry Pi [11].
- La aplicabilidad de productos de bajo costo para dispositivos del IoT [12].

Las aplicaciones del IoT siguen en expansión constante. Recientemente se han realizado diversas investigaciones y proyectos utilizando sistemas embebidos como la Raspberry PI. Permitiendo aumentar las posibilidades de iniciar en estudios de domótica, como la utilización de Tecnicas Smart Home por medio de la Raspberry Pi usando el IoT [13], supervisión de seguridad de manera remota en una habitación [16], también en estudios de monitoreo de energía se han desarrollado medidores automatizados que utiliza Wi-FI de la Raspberry Pi [14], en estudios ambientales y de agricultura como la detección de la contaminación atmosférica utilizando la Raspberry Pi con notificación basa en la IoT [62], sistemas para la vigilancia del clima urbano [15], , sistema de monitoreo de agricultura de precisión [17] y aplicaciones para la creación de un robot familiar [18].

En estudios de ergonomía ambiental existen proyectos enfocados al confort lumínico como la evaluación del rendimiento de la luz del día de un espacio de oficinas comerciales en clima cálido y árido para mejorar las condiciones de confort visual [63], Medición de niveles de confort de apartamentos usando sensores IoT [64], Con el fin de crear un camino que posibilite la ampliación de aplicativos para estos tipos de sistemas de monitoreo, se decidió enfocarse en variables de temperatura y humedad relativa que son un factor de estudio en diferentes campos como del confort térmico adaptativo [19], influencia en el tiempo de secado en hojas [20], relación de la temperatura y humedad con respecto a crecimiento de plantas [21], modelado de distribución de la temperatura y humedad en granos almacenados en silos [22].

1.3 PROBLEMA

En la actualidad existe una gran cantidad de estudios que buscan el control o el monitoreo de las cosas que los rodean, para lograrlo es importante tener conocimientos de estrategias que permitan captar las variables de forma remota y en tiempo real [23]. Por esta razón se necesita la creación de proyectos que encaminen a la adquisición, tratamiento y monitoreo de variables provenientes del entorno que nos rodea y a su vez tener la disposición de datos a través del internet [24].

La visualización de datos a través del internet ha sido implementada de diferentes maneras, lo que permitió que gracias al concepto de la IoT aportara beneficios, por medio de aplicaciones que utilizan diferentes ámbitos de interés; sociales, educación, comunicación, empresas, ciencias y gobierno [26]. Sin embargo, posee puntos de trascendencia que siguen sin ser utilizados y/o profundizados como en la ergonomía ambiental, por esto se crean oportunidades para la creación de nuevos estudios y soluciones a problemas que en tiempo real permitan mejorar servicios no solo relacionado con la ergonomía si no también con la industria, gobierno, medio ambiente, energía, entre otros [27].

La ergonomía ambiental es una rama de la ergonomía que se dedica al estudio y análisis de condiciones externas al ser humano que influyen en su vida diaria. Dentro de estos factores ambientales se encuentran: niveles térmicos, niveles de ruido, ventilación, iluminación, entre otros. Por lo que estudiarlos ayuda a mejorar las condiciones del confort en la vida cotidiana [59]. Uno de los principales aspectos en el que se usa el internet de las cosas, es el monitoreo de variables en tiempo real lo cual con ayuda de estudios de ergonomía ambiental se pueda determinar el confort de los usuarios, según la Real Academia Española define a este término como: “Bienestar con comodidad material”, aunque es un término subjetivo es crucial para los diseñadores, ya que un producto que ofrece mayor confort será mejor valorado por el usuario [58]. En la actualidad no existen estudios de confort

térmico que formen parte del IoT, con el fin de monitorear variables de humedad relativa y temperatura del aire de forma remota.

1.4 JUSTIFICACIÓN

La tendencia del internet como medio de visualización y monitoreo de datos, es un tema en crecimiento continuo en la última década, con el avance de nuevos dispositivos más portables y económicos, se está dando la implementación de sistemas embebidos de bajo costo en diferentes ámbitos de interés [26]. Por tal razón se busca desarrollar un sistema de visualización y monitoreo de datos que permita agregar valor a los servicios existentes.

Gracias al concepto de la IoT se busca vencer la barrera de distancia del dispositivo con respecto al usuario y la visualización de la información [29]. La ruptura de esta barrera permite captar mediciones en sensores instalados a grandes distancias (temperatura, presión, humedad, calidad de aire, etc.) [30].

Acorde a las tendencias actuales, este tipo de sistemas exigen plataformas versátiles orientadas al prototipo rápido. Con el fin de utilizar plataformas y tarjetas de desarrollo que cuentan con la capacidad de albergar un sistema operativo, lo que permite contar con controladores de video, puertos de comunicación USB y medios de conectividad como tarjetas de comunicación tipo Wi-Fi.

Un ejemplo de estas plataformas es del tipo hardware abierto denominadas comercialmente Raspberry Pi. Esta referencia fue creada por la unión de diversos grupos de investigación para suplir necesidades relacionadas con el desarrollo de aplicaciones portátiles orientadas a la adquisición y monitoreo de datos de forma remota usando conectividad Wi-Fi [31].

La Raspberry Pi ofrece una gran cantidad de herramientas, debido a la ventaja de poder albergar un sistema operativo embebido, lo que facilita el desarrollo de programas y aplicativos, en diferentes entornos de programación. Al ser un recurso comercial, está posibilitando el desarrollo de proyectos en diferentes ámbitos de estudios [31]. La empresa Raspberry ofrece un sistema operativo embebido (Raspbian) para la tarjeta de desarrollo, lo que permite ofrecer diferentes

herramientas de programación para la creación de aplicaciones en la plataforma Raspberry Pi [65]. Uno de los principales recursos de programación que ofrece es el lenguaje de programación Python, con este lenguaje se puede hacer uso de librerías para el manejo de datos, desarrollo de interfaz gráfica, intercambio de datos por puerto USB, conectividad con tarjetas de red y desarrollos de página WEB.

Consecuentemente, con el fin de incentivar un sistema de basado en software libre, para el monitoreo de variables que permitan determinar el Confort térmico según estudios realizados por el IDEAM (Instituto de Hidrología, Meteorología y Estudios Ambientales). El presente trabajo de grado plantea desarrollar un prototipo de sistema de adquisición de datos en la plataforma Raspberry Pi, y por medio del lenguaje de programación “Python” se plantea la creación de una interfaz gráfica de usuario, que permita monitorear y subir datos a la nube, a su vez determinar el confort térmico de su entorno. Por lo que se buscar dar un valor agregado a estudios de ergonomía ambiental para aplicaciones bajo el enfoque del IoT.

1.5 OBJETIVOS

1.5.1 OBJETIVO PRINCIPAL

Diseñar e implementar un sistema de adquisición y visualización de datos que permita visualizar al usuario el confort térmico de su entorno a partir de datos de temperatura y humedad relativa.

1.5.2 OBJETIVOS ESPECIFICOS

- Diseñar e implementar una interfaz gráfica basada en el lenguaje de programación “Python” para la visualización en la adquisición de variables de humedad y temperatura.
- Determinar la sensación de confort térmico, considerando variables de humedad relativa y temperatura del aire a partir de estudios realizados por el IDEAM (Instituto de Hidrología, Meteorología y Estudios Ambientales)..
- Implementar estrategias de filtrado digital para la eliminación de ruidos provenientes de los sensores de temperatura y humedad.

2. MARCO CONCEPTUAL

Para el desarrollo de este proyecto se hace necesario una breve aclaración de diferentes conceptos, que permitan ampliar el conocimiento básico requerido para este proyecto.

2.1 CONFORT TERMICO

El confort térmico conocido también como confort climático es un estudio, donde entran elementos meteorológicos básicos como la temperatura, la humedad, el viento, la radiación solar y su variabilidad a través del día y del año [48], El confort térmico posee diferentes definiciones dependiendo de la norma, según la norma técnica colombiana 5316 la define como: condición de la mente que expresa satisfacción con el ambiente térmico [49], el Instituto Nacional Estadounidense de Estándares, conocido como ANSI (por sus siglas en inglés) según la norma ASHRAE 55 define el confort térmico como: esa condición de la mente que expresa satisfacción con el ambiente térmico y es evaluado por medición subjetiva [50], y por ultimo según la norma ISO 7730 lo define como: esa condición de mente en la que se expresa la satisfacción con el ambiente térmico [51]. Cada una de estas definiciones son aceptadas, sin embargo, es difícil su traducción en parámetros físicos cuantificables. Básicamente y en términos generales, el hombre califica un ambiente confortable, si ningún tipo de incomodidad térmica está presente. La primera condición de confort es la neutralidad térmica, lo que significa que la persona no se siente demasiado acalorada ni demasiado fría [52].

Existe una gran cantidad de métodos y cálculos para poder estimar el grado de confort térmico, tales como cálculos del poder de refrigeración del cuerpo humano,

cálculo de la sensación térmica experimentada, cálculo de temperatura efectiva o equivalente, todo esto mediante el uso de gráficas, tablas y formulas.

Con el fin de incentivar estudios nacionales para la determinación del confort térmico se escogió la nota técnica del IDEAM “Metodología para el cálculo del confort climático en Colombia” [48]. Gracias a que el cálculo que ofrece esta nota técnica solo maneja variables cuantificables como la humedad relativa, temperatura del aire y velocidad del aire, facilita la adquisición de este tipo de datos por medio de sensores.

La nota técnica menciona que el estudio del confort humano, tiene como punto de inicio el confort climático, debido a que Colombia es un país tropical, la vida y actividades de sus ciudadanos están relacionados con el clima. Algunos factores climáticos condicionan la eficiencia de sus actividades, que puede actuar de forma beneficiara o adversamente en todas sus actividades y el funcionamiento general de su organismo.

La fórmula que se adoptó para este estudio realizado por la IDEAM, se basa en la fórmula de poder de refrigeración de Leonardo Hill y Morikofer-Davos, con algunas modificaciones: por lo que se obtiene un índice de confort (IC), en lugar del poder de refrigeración. En segundo lugar, se incluyó el parámetro de humedad relativa. En tercer lugar, los valores base para cada uno de los parámetros fueron modificados hasta conseguir que los resultados fueran adecuadas a las condiciones climáticas del país. Teniendo en cuenta la temperatura con la altura ya que es en este país el relieve es un factor importante.

La comprobación de este estudio fue realizado por el IDEAM por medio de pequeñas encuestas sobre la sensación que ha experimentado una muestra variada de diferentes personas ante diferentes puntos del país.

La fórmula que se adoptó y se ajustó para la determinación del Índice de Confort, crea dos variantes en la formula, un considerando el viento y la otra no. Por lo cual plantean las siguientes formulas:

Considerando el viento:

$$IC = (36.5 - Ts)\left(0.05 + 0.04\sqrt{v} + \frac{h}{250}\right) \quad (1)$$

Para elevaciones inferiores a 1000 metros

$$IC = (34.5 - Ts)\left(0.05 + 0.06\sqrt{v} + \frac{h}{180}\right) \quad (2)$$

Para elevaciones entre 1000 a 2000 metros

$$IC = (33.5 - Ts)\left(0.05 + 0.18\sqrt{v} + \frac{h}{160}\right) \quad (3)$$

Para elevaciones superiores a 2000 metros

Donde:

IC = Índice de confort

Ts = Temperatura del aire en grados centígrados (°C)

v = Velocidad del Viento en metros por segundo (m/s)

h = Humedad relativa en porcentaje (%)

Sin tener en cuenta el viento:

$$IC = (36.5 - Ts)\left(0.05 + \frac{h}{250}\right) \quad (4)$$

Para elevaciones inferiores a 1000 metros

$$IC = (36.5 - Ts)\left(0.05 + \frac{h}{180}\right) \quad (5)$$

Para elevaciones entre 1000 a 2000 metros

$$IC = (36.5 - Ts)\left(0.05 + \frac{h}{160}\right) \quad (6)$$

Para elevaciones superiores a 2000 metros

Teniendo en cuenta las condiciones del país se estableció una nueva clasificación bioclimática así:

Índice de Confort (IC)	**Sensación Experimentada**
0 a 3	Muy Caluroso
3.1 a 5	Caluroso
5.1 a 7	Cálido
7.1 a 11	Agradable
11.1 a 13	Algo frío
13.1 a 15	Frio
Más de 15	Muy frío

Tabla 1. Tabla para la determinación de la Sensación Experimentada [48].

A partir de los estudios realizados por el IDEAM, se realizó un análisis para determinar los rangos máximos y mínimos de humedad relativa y temperatura del aire, que se encuentran en Colombia. Por medio de la página web que ofrece el IDEAM, se realizó el análisis de los mapas climáticos de humedad relativa y temperatura media en el país, teniendo en cuenta que estos estudios fueron realizados hasta el año 2014. Los mapas climáticos se pueden apreciar en la figura 1, figura 2 y figura 3.

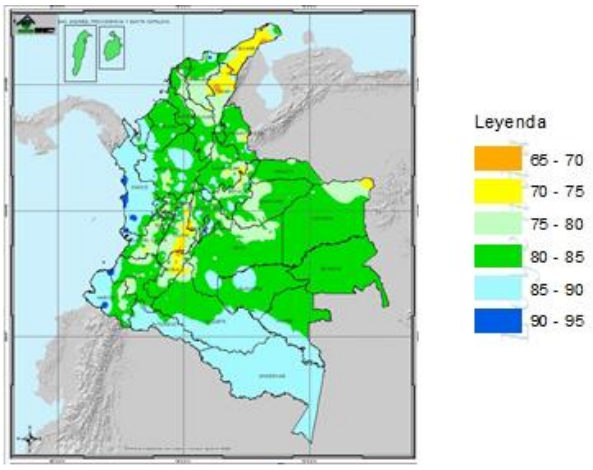


Figura 1. Mapa climático de Humedad Relativa de Colombia
Promedio Anual (1981-2010)

En la figura 1, se pueden ver los diferentes rangos de humedad relativa que posee Colombia en las ultimas 3 décadas, con el fin de determinar un rango de trabajo para la determinación de un sensor de humedad relativa, se tomaron los máximos y mínimos a partir del mapa climático, para este caso se necesita un sensor que trabaje en rangos de 65% a 95% de humedad relativa. El mapa anterior estará disponible en el Anexo A.

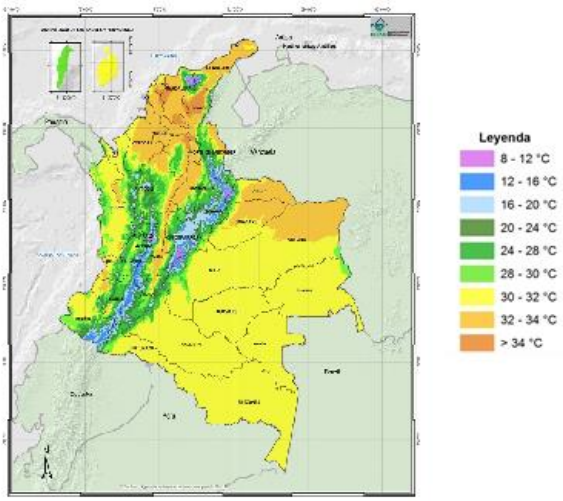


Figura 2. Mapa climático de temperatura máxima media de Colombia (2014)

La determinación de los rangos de temperatura del aire se dividió en dos análisis diferentes, el primero es el de temperatura máxima media de Colombia del año 2014 que se puede apreciar en la figura 2, a partir de este mapa se puede visualizar que llego a rangos superiores a 35°C. Este mapa estará disponible en el Anexo B.

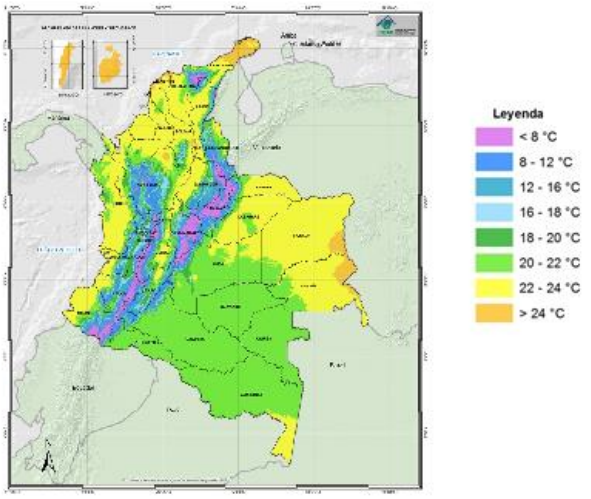


Figura 3. Mapa de temperatura mínima media de Colombia (2014)

La temperatura mínima media de Colombia del año 2014 llego a temperaturas inferiores de 8°C en diferentes zonas del país, a partir de estos datos y el análisis realizados en los mapas se llegó a la conclusión que se requiere un sensor de temperatura que pueda trabajar en rangos menores de 8°C y a temperaturas mayores de 35°C.

Al final de este análisis se determinó que es necesario contar con un sensor de humedad relativa que trabaje en rangos entre 65% a 95 % y un sensor de temperatura del aire que trabaje en rangos menores a 8°C y superiores a 35°C. Al tener en cuenta los rangos de humedad relativa y temperatura del aire se generaron graficas de la Ecuación 4, Ecuación 5 y Ecuación 6 por medio del software Matlab. Correspondiente a las diferentes altitudes que menciona la nota técnica del IDEAM. Estas graficas serán presentadas en el Anexo D.

2.1.1 SENSOR DE TEMPERATURA Y HUMEDAD RELATIVA

Para la selección del sensor de humedad relativa y temperatura del aire, se tuvieron en cuenta los siguientes requisitos:

Requisitos del Sensor de Temperatura y Humedad Relativa	
1	Un sensor comercial que permita la captación de humedad relativa y temperatura del aire
2	Debe ser un sensor que cumpla con los siguientes Rangos: Humedad relativa: 65% a 95% Temperatura del Aire: Inferiores a 8°C y superiores a 35°C
3	Debe ser un sensor comercial de bajo costo

Tabla 2. Requisitos del sensor de temperatura y humedad relativa

Según los requisitos estipulados en la tabla 2. Se encontró tres diferentes sensores comerciales acordes a las necesidades de este proyecto, los cuales son el sensor DHT11, DHT22 y SHT10. De estas tres opciones se escogió el DHT11 ya que en comparación con el DHT22 y el SHT10, el DHT11 es un sensor de rango menor pero que cumple con los rangos de trabajo requeridos para las variables de temperatura y humedad, su rango de precisión es aceptable para este proyecto, es más pequeño y de menor costo. Las características de estos sensores se pueden ver en la Tabla 3.

Parámetro	DHT 11	DHT22	SHT10
Alimentación	3 VDC≤ VCC ≤ 5 VDC	3 VDC≤ VCC ≤ 5 VDC	2.5 VDC≤ VCC ≤ 5.5 VDC
Rango de Medición de Temperatura	0 a 50°C	-40°C a 80°C	-40°C a 123.8°C
Precisión de medición de Temperatura	± 2.0 °C	± 5.0 °C±1ºC (condiciones adversas)	0,3ºC
Rango de medición de humedad	20 % A 80% RH	0 % A 99.9% RH	0 % A 100% RH
Precisión de medición de humedad	4% RH	±2%RH	±2%RH
Tiempo de Censado	2 Segundos	2 segundos	1 Segundo
Tamaño	12 x 15.5 x 5.5mm	14 x 18 x 5.5mm	78.74 x 40.64 x 5.08 inches
Precio	$4.82	$12.49	$13.99

Tabla 3. Características del Sensor DHT11, DHT22 y SHT10[57].

El módulo DHT11 es un sensor de temperatura y humedad relativa del aire de bajo costo, consiste en un sensor capacitivo de humedad y un termistor para medir el aire circundante, internamente posee un convertidor análogo digital que envía una trama de datos de 40 bits en la que los primeros 32 bits son los datos de temperatura

y humedad, los 8 bits restantes son bits de paridad que confirman que no hay datos corruptos esto facilita la comunicación con cualquier micro controlador, por lo que da una ventaja frente a los sensores de tipo analógico que presentan fluctuaciones de voltaje que posibilitan error en la lectura de datos [43].

Figura 4. Sensor de Temperatura y Humedad Relativa DHT11 [56].

2.2 INTERNET DE LAS COSAS

El IoT es el concepto de la interconexión de dispositivos tales como, computadoras, teléfonos móviles, electrodomésticos, que están conectados a internet [44]. Este tema ésta cambiando el enfoque para la utilización de identificadores únicos para diversos tipos de objetos, impulsando un avance en el intercambio de información de sensores en ambientes heterogéneos, que a su vez permite innovar y confiar datos en el IoT buscando la seguridad y privacidad de estos objetos en su entorno como también protección sobre la información [45].

Estas características van acompañadas de una tecnología apropiada para el objeto determinado, por ejemplo, puede estar identificado con códigos de dos dimensiones o marcas de agua digitales probablemente basados en el protocolo IPv6, los cuales mandan información a una base de datos por medio de internet u otra red convergente, convirtiendo estos objetos estáticos en objetos dinámicos y haciéndolos parte del internet de las cosas [46]. En pocas palabras el IoT es lo que

se consigue cuando se conecta a internet objetos que no son operados por los seres humanos [47].

El internet de todo por sus siglas en inglés (Internet of Everything), es un término que está tomando gran importancia, debido a que hace referencia a la conexión inteligente no solo de las cosas, sino también de personas, procesos y datos. Lo cual está creando la oportunidad de que algunos objetos se les agregue cierto nivel de inteligencia para que pueda funcionar de manera más convergente [58]. Gracias a esto grandes empresas han comenzado una carrera para invertir en la investigación referente a el internet de las cosas, con el fin de implementar estos servicios en hogares y empresas, lo cual permite que este concepto sea mejor valorado a largo plazo.

2.3 SISTEMAS EMBEBIDOS

Un sistema embebido es el conjunto de software y hardware integrados para la ejecución de una función dedicada [32], esta definición es usada en la actualidad. Otra definición encontrada es “Un Sistema Embebido es un sistema informático distinto de las computadoras de escritorio” [33]. Estos sistemas poseen características comunes que los hacen un sistema embebido tales como: Funcionamiento único, tiempo real, algoritmos complejos, multi-velocidad [34].

2.3.1 TARJETA DE DESARROLLO RASPBERRY PI

La Raspberry Pi es un sistema embebido que ofrece ventajas como: su reducido tamaño, una buena capacidad de proceso y su bajo costo en el mercado, una Raspberry Pi 3 actualmente tiene un valor de 40 USD. Es un proyecto desarrollado en Reino Unido por la fundación Raspberry Pi, con el fin de promover la producción digital a nivel mundial.

La tarjeta de desarrollo posee como sistema operativo una distribución de Linux adaptada de la versión Debían, que fue optimizada para el hardware de Raspberry Pi tomando el nombre de Raspbian, el cual tiene instalado asistentes matemáticos, herramientas de programación y una arquitectura modular y abierta para construir sistemas electrónicos [35].

La Raspberry Pi 3 es la tercera versión de tarjetas lanzadas al mercado en febrero de 2016 permitiendo la renovación de procesador, inclusión de módulo de comunicación Wifi y Bluethooth sin necesidad de adaptadores. Se trata de una placa base que consta de procesador, CPU, RAM, conexiones inalámbricas, almacenamiento, GPIO y puertos, donde sus especificaciones se representan en la tabla 4 [36]:

Especificaciones	
Procesador	Broadcom BCM2837
CPU	4xARM Cortex-A53 1.2GHz
GPU	Broadcom Video Core IV
RAM	1GB LPDDR2 (900Mhz)
Conexión	10/100 Ethernet, 2.4Ghz 802.11n Wireless LAN
Bluetooth	Bluetooth 4.1 Classic, Bluetooth Low Energy (BLE)
Almacenamiento	Puerto Micro SD
GPIO	40 pines de propósito general, el cual disponemos de 26 pines para entradas y salidas digitales.
Puertos	HDMI, 3.5mm análogo audio-video Jack 4 puertos USB 2.0, Interfaz de Camara Serial(CSI), Interfaz de Display Serial(DSI)

Tabla 4. Especificaciones Raspberry Pi 3 Modelo B [37].

Dentro de la placa madre se encuentran pines específicos que permiten la fácil comunicación con otros tipos de dispositivos o herramientas de control, como: puertos de PWM (Pulse width modulation) dedicados en el hardware a 4 pines externos, bus serial de interfaz periférico (SPI), protocolo I^2C (I2C) y comunicación serial [37].

Especificaciones	
PWM	• PWM por software disponible en todos los pines • Hardware PWM disponible en los pines GPIO12, GPIO13, GPIO18, GPIO19
SPI	• SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7) • SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
I^2C	• Data: (GPIO2); Clock (GPIO3) • EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
SERIAL	• TX (GPIO14); RX (GPIO15)

Tabla 5. Distribución de pines para propósitos específicos [37].

La Raspberry Pi 3 Modelo B Versión 1.2, Posee 40 pines de entrada y salida como puertos de propósito general (GPIO de sus siglas en ingles *General Purpose Input/Output*). Lo cual permite configurar cualquier pin como entrada o salida, permitiendo una amplia gama de propósitos. Teniendo en cuenta que el nivel alto es de 3.3V y no son tolerantes a tensiones a 5V y su nivel bajo es de 0V. A partir de la figura 5 se puede ver la distribución de pines de este sistema embebido [37].

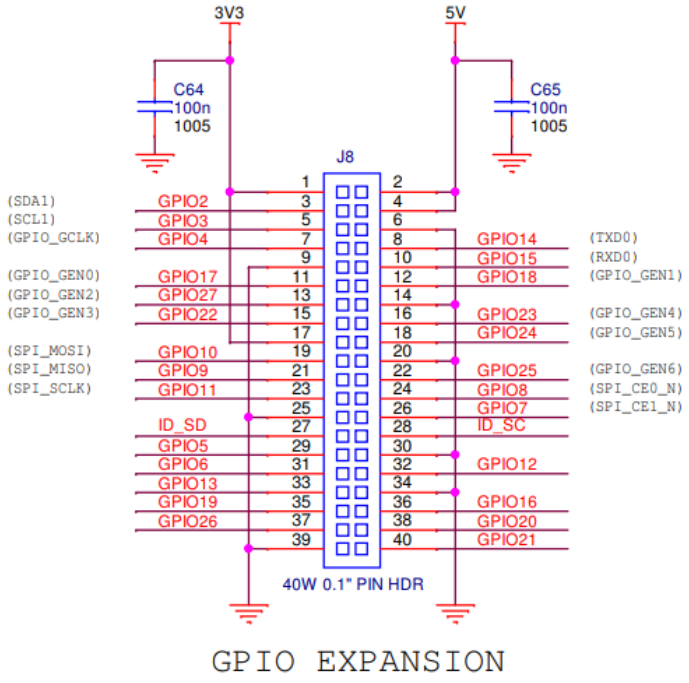


Figura 5. Esquemático de Puertos GPIO Raspberry Pi 3 Modelo B V1.2 [37]

2.3.1.1 SISTEMA OPERATIVO RASPBIAN GNU/LINUX

Un sistema operativo consiste en el software que administra una computadora y permite ejecutar aplicaciones en él, ha sido uno de los avances tecnológicos más importantes en el siglo XXI. Uno de los sistemas operativos más populares son aquellos basados en las plataforma Linux, el desarrollo de sistemas operativos de software abierto conocido popularmente como “Open Source” es un ejemplo de cómo los proyectos colaborativos igualan a compañías dedicadas al desarrollo de Software [38], uno de estos ejemplos es el sistema operativo Raspbian, consiste en un sistema operativo libre basado en Debian Jessie optimizado para el hardware de Raspberry Pi, es un sistema operativo embebido lanzado en junio de 2012, que ofrece una gran cantidad de recursos como: más de 35.000 paquetes de software

que permiten, actualizar, instalar y desinstalar de forma sencilla, software de desarrollo pre instalados en el sistema operativo [39]. La distribución de escritorio que posee este sistema operativo usa LXDE y Midori como navegador web. Además, contiene herramientas de desarrollo como IDLE para los lenguajes de programación Python y Scratch. Es importante tener en cuenta que Raspbian no está afiliado a la fundación Raspberry, es un software libre y financiado por la comunidad.

El sistema operativo Raspbian posee una distribución GNU\ Linux, lo cual permite que todo su software este en código abierto lo que crea la posibilidad de ser recompilado en la misma Raspberry PI, también ofrece repositorio que permiten a los usuarios la opción de descargar programas como si se tratase de una distribución de Linux, tomando la ventaja de una computadora de escritorio de bajo costo. Raspbian permite adquirir datos provenientes de transductores comerciales por medio del uso de librerías aportadas en la comunidad. Se denomina transductor en general a todo dispositivo que convierte una señal de una forma física en una señal correspondiente, pero de otra forma física distinta [40]. Por tanto, un sensor es un dispositivo que a partir de la energía del medio donde se mide da una señal de salida traducible que es función de la variable de medida [41],

2.4 LENGUAJE DE PROGRAMACIÓN PHYTON

El lenguaje de programación “Python” fue creado por Guido Van Rossun. Este lenguaje nace debido a su motivación y experiencia en lenguaje ABC. Debido a la necesidad de un lenguaje orientado a objetos de fácil implementación, y que permitiera tratar diversas tareas dentro de la programación, es creado el lenguaje de programación Python a finales de los años ochenta [42].

Phyton es un lenguaje de programación que permite un amplio conjunto de herramientas de codificación, es modular y con un grupo de bibliotecas que ayuda a ampliar rápidamente su desarrollo. Python posee una licencia de código abierto lo que favores a los programadores un extenso dominio en aplicaciones como desarrollo web e internet, educación, científico y numérico y aplicación en negocios [42]. Python es un lenguaje de programación interpretado, permite a diferencia de otros lenguajes implementar librerías estándar en el lenguaje C, lo que hace que sus funciones sean bastante eficientes al momento de compilarse [60].

Python es un lenguaje de programación orientada a objetos, que puede ser soportado sobre cualquier plataforma de sistema operativo o servidor red, Python no necesita compilar el código fuente para poder ser ejecutado, este lenguaje permite la facilidad de desarrollar programas en menor tiempo debido a su sencillez en la sintaxis. Otra gran ventaja que posee es la disposición de librerías incorporadas en el propio lenguaje para el tratamiento de archivos, caracteres, importación de programas e incluso sistemas de red. Debido a que es un lenguaje de alto nivel permite contener de forma implícita algunas estructuras de datos como: listas, conjuntos y tareas complejas en pocas líneas de código de manera legible [60].

2.5 INTERFACE DE APLICACIÓN EN LA NUBE

Thingspeak es una interfaz de programación de aplicaciones API (por sus siglas en inglés Application Programming Interface) específica que permite el manejo de datos, esto quiere decir que su trabajo primario es recopilar, almacenar, analizar y visualizar la información suministrada por los usuarios. Una gran ventaja de esta API es su código abierto. El tiempo de actualización de Thingspeak es de 15 segundos [53]. La API de publicación y consulta de datos está basada en REST lo cual permite recibir datos en formato XML, JSON o CSV [54].

Thingspeak es una interfaz de programación desarrollada por Mathworks que permite enviar datos a la nube [53], esta plataforma es comúnmente usada para análisis del IoT, debido a que ofrece suscripciones de forma gratuita, sus usuarios tienen la posibilidad de crear canales privados o públicos que permitan la lectura o el almacenamiento de datos.

Para contextualizar la inscripción y el uso de esta plataforma se explicará paso a paso la creación de una cuenta en Thingspeak:

Paso 1: Se ingresa por medio de cualquier navegador de internet a la página web: del anexo E, para iniciar el proceso de registro se selecciona la opción de “Sign Up”. Vista en la figura 6.

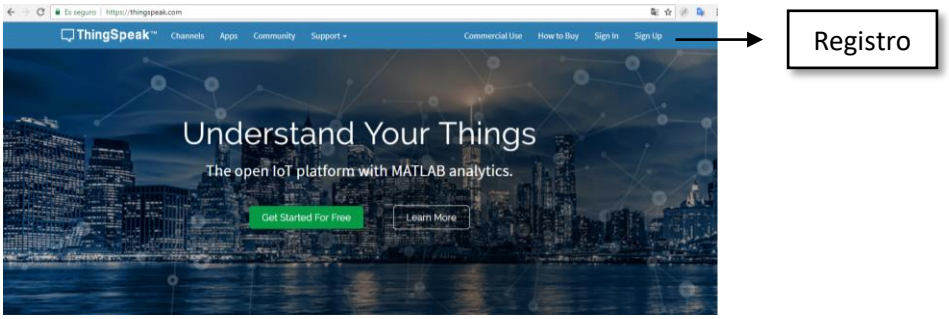


Figura 6. Página de inicio web de Thingspeak.

Paso 2: Para ingresar en la plataforma de Thingspeak es necesario ingresar con una cuenta de Mathworks existente o en el caso contrario crearla.

Paso 3: Thingspeak ofrece dos tipos de cuenta en su dominio, La primera es la cuenta Estudiantil que tiene un costo de USD 45,00 precio / unidad, ofreciendo como beneficio recibir 33 millones de datos al año y la actualización del canal cada segundo. La segunda es la opción gratuita, aun que posee recursos menores como: capacidad de almacenamiento de 3 millones datos al año y la actualización del canal se hace cada 15 segundos.

	GRATIS Para pequeños proyectos no comerciales	ESTUDIANTE Para estudiantes en instituciones que otorgan títulos [1]
Escalable para proyectos más grandes	✖ No. El uso anual está limitado.	✔
Cantidad de mensajes	3 millones / año (~ 8.200 / día) [2]	33 millones / año por unidad (~ 90.000 / día por unidad) [2]
Límite de intervalo de actualización de mensaje	Cada 15 segundos	Cada segundo
MATLAB Compute Timeout	20 segundos	20 segundos
Número de suscripciones MQTT simultáneas	Limitado a 3	50 por unidad
Compartir canales privados	Limitado a 3 acciones	Ilimitado
Soporte técnico	Foro	Foro

Figura 7. Cuentas que ofrece Thingspeak

Paso 4: En el momento de inscribirse en Thingspeak, se iniciará el proceso para la creación de un canal de almacenamiento de datos, lo primero es registrar un nombre y de ser requerido, una descripción sobre el propósito del canal. Thingspeak administra el almacenamiento de datos por medio de campos. Por cada canal se tiene un máximo de 8 campos de almacenamiento. Los campos de registro que solicita Thingspeak se pueden apreciar en la figura 8.

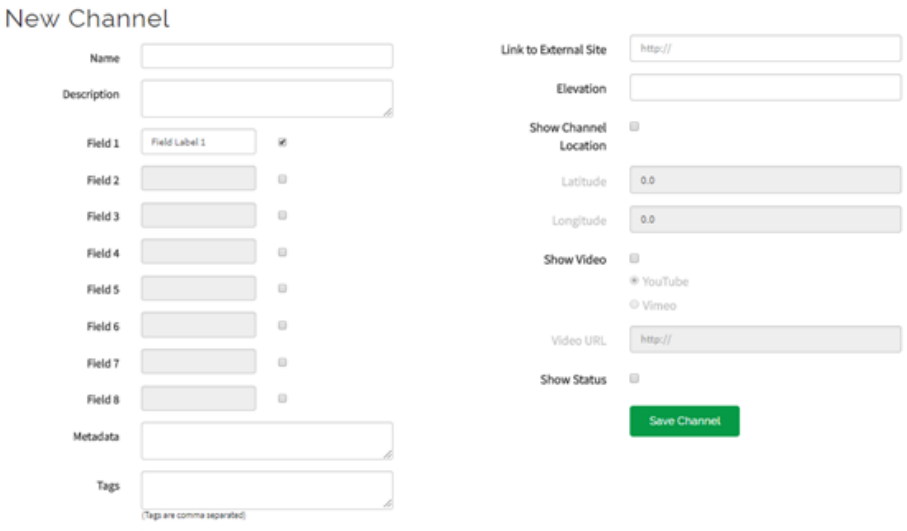


Figura 8. Registro de datos para la creación de un canal en Thingspeak.

Al registrar el nombre y seleccionar cuantos campos de almacenamiento se van a utilizar, presionaremos la opción en “Save channel” y automáticamente Thingspeak genera el nuevo canal.

Paso 5: Thingspeak ofrece la opción de visualizar el canal de manera privada o pública, es decir si el administrador prefiere que los datos almacenados en los campos puedan ser vistos de forma pública o privada. Al tener creado el canal Thingspeak ofrece dos API’s Key (Llave de interfaz de programación de aplicaciones) por cada canal, la primera es “Write API’s Key” que es básicamente una clave que permite la escritura de datos en los campos del canal y la segunda es de “Read API’s Key” que permite adquirir datos almacenados en los campos del canal. Thingspeak ofrece la facilidad de regenerar las API’s Key si el administrador considera que alguna de las dos claves ha sido vista o comprometida. En la figura 9 se puede apreciar la visualización de API’s Key en Thingspeak.

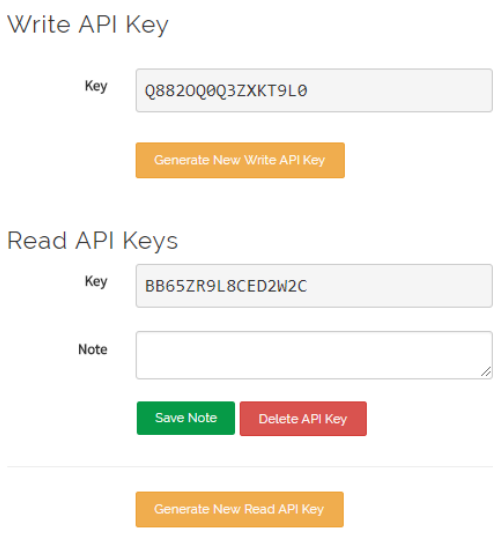


Figura 9. Visualización de API's Key en Thingspeak

3. DISEÑO Y EJECUCIÓN DEL PROYECTO

Para la etapa de diseño y ejecución se ha dividido en cuatro etapas, la primera consta en los requerimientos del programa, donde contiene la planeación de la interfaz de usuario en Python, la segunda es el planteamiento del sistema el cual explica que elementos que intervienen para la formación del proyecto, la tercera parte consiste en el desarrollo del programa, en esta parte se explicará el desarrollo que ha tenido la aplicación y el desarrollo final del programa para subir los datos del sensor DTH11 y determinar el confort térmico, y como cuarta y última parte consta de los parámetros para detección de fallos del aplicativo y que errores se tuvieron en cuenta para hacerlo más estable.

3.1 REQUERIMIENTOS DEL PROGRAMA

Para el desarrollo de este proyecto se plantea un programa que permita adquirir, graficar y procesar datos por medio de una interfaz gráfica utilizando una tarjeta de desarrollo Raspberry Pi, con la finalidad de determinar la sensación de confort térmico en su entorno. Por lo tanto, se plantean requerimientos que busquen cumplir las necesidades del proyecto priorizando los aspectos más importantes que debe tener el aplicativo como: cumplimiento de los objetivos del proyecto, estabilidad en el programa, comodidad visual en la interfaz gráfica, fácil uso para los usuarios y control de errores. Estos requerimientos fueron planteados a partir de recomendaciones y estudios para el desarrollo de software [66].

Para el desarrollo del aplicativo se plantea los siguientes requerimientos básicos en la tabla 6.

Requerimientos	
1	El aplicativo debe constar con una interfaz gráfica estable, que grafique con respecto al número de muestras tomadas y permita visualizar las variables de humedad y temperatura en tiempo real.
2	El aplicativo debe contar con controles de Inicio y parada para la interfaz del programa y a partir de los estudios realizados por la IDEAM determinar la sensación de confort térmico experimentada.
3	El usuario podrá manejar el parámetros de control en el programa, permitiendo escoger los pines de entrada para el DHT11, la altitud en la cual se va a hacer la prueba y si lo desea puede ingresar el API KEY de escritura al cual quiere que se envíen los datos de humedad relativa y temperatura del aire.
4	Debe ser una interfaz gráfica de fácil manejo para los usuarios, básica y que permita conocer el: confort térmico, humedad relativa y temperatura aire en el entorno.
5	La interfaz gráfica debe tener manejo de control de errores que permita al aplicativo informar el fallo al usuario para la toma de decisiones.

Tabla 6. Requerimientos del programa.

Para el desarrollo de este aplicativo se realiza el diagrama de flujo que permitirá una descripción de los componentes generalizados en el software. El diagrama de flujo inicia desde el momento en el que se abre la interfaz gráfica desarrollada en Python, este bloque inicializa las variables que se usan en el programa como la visualización de la interfaz para el usuario, luego el programa espera la decisión del usuario, presentando las siguientes opciones:

- Iniciar proceso (Este bloque constará de: adquisición, cálculo del índice del confort térmico, gráfica, visualización de variables y subida de datos a Thingspeak).
- La segunda opción es al bloque de Configuración, siempre se tendrá la opción de acceso en cualquier momento del programa (Permite la inicialización de parámetros o modificación de los mismos).
- La tercera opción es la toma de decisión de cerrar el programa (Teniendo en cuenta que esto debe hacerlo en cualquier momento o proceso en el que este el programa).
- La cuarta opción es la interfaz consta de un subproceso el cual Reinicia la gráfica: permite que la interfaz reinicialice las gráficas de Humedad relativa y Temperatura, cada vez que el usuario lo desee.

Por último, el subproceso más importante es el control de errores en diferentes casos, el cual permite informa el error al usuario para la toma de decisiones o corrección en el sistema. El diagrama de flujo está representado en la Figura 10.

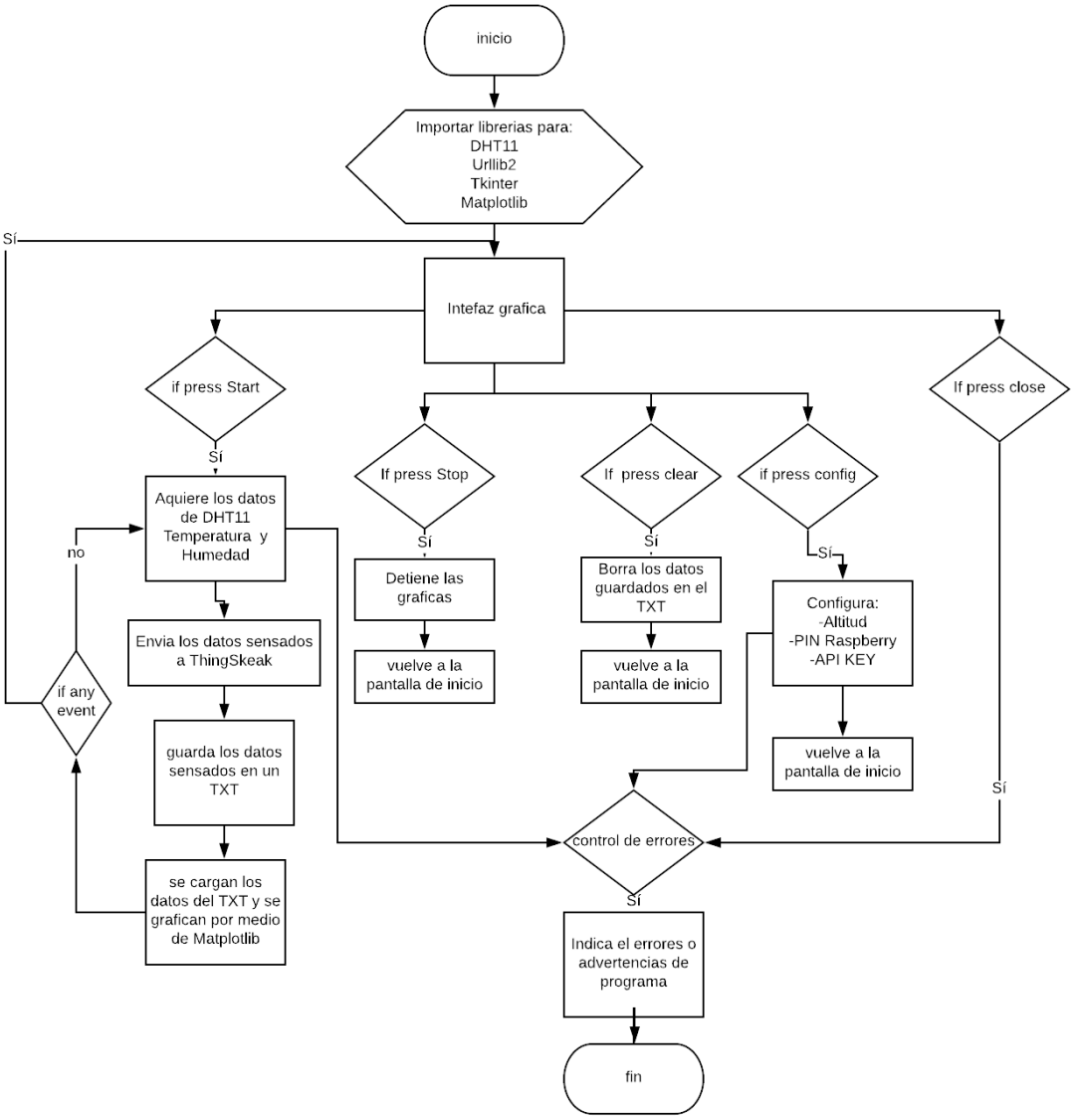


Figura 10. Diagrama de flujo general del programa

3.2 PLANTEAMIENTO DEL SISTEMA

Se escogió el lenguaje de programación Python debido a su fácil manejo y la viabilidad que tiene en el momento de migrar entre diferentes sistemas operativos, permitiendo que cualquier desarrollador de software tenga la facilidad de modificar

programas en diferentes plataformas. Con el fin de promover un sistema de bajo costo se busca realizar un aplicativo sobre el sistema operativo embebido Raspbian, debido a que es un desarrollo software libre, en el cual permite la creación y el desarrollo de una interfaz gráfica para una pantalla táctil de 7” conectada a una Raspberry Pi, con el propósito de adquirir variables de humedad relativa y temperatura a partir de un sensor DTH11, y poder visualizar la sensación de confort térmico a partir de los datos de humedad relativa y temperatura del aire, las cuales serán enviadas al aplicativo web Thingspeak.

Al subir datos a la nube provenientes de sensores se está incluyendo este proyecto que forme parte del internet de las cosas, con el fin de incentivar el desarrollo de aplicativos con interfaz gráfica de usuario sobre la Raspberry Pi. Se busca demostrar que este tipo de aplicativos pueden capturar variables en tiempo real y ser procesadas para su uso, permitiendo a diferentes usuarios ver estas variables de forma remota a través de otros tipos de dispositivos.

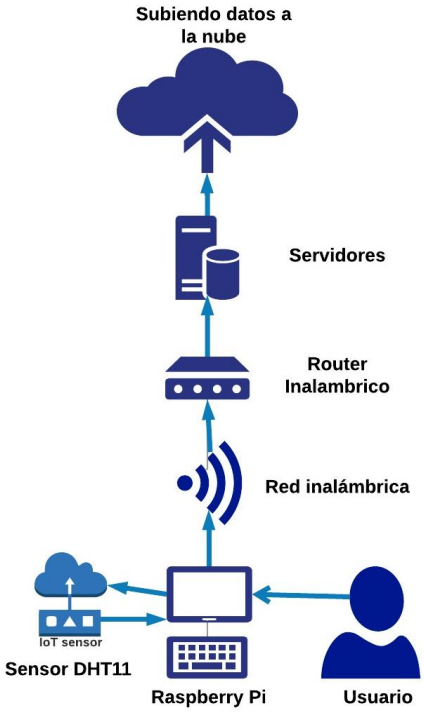


Figura 11. Diagrama de Red Genérico del Proyecto

Actualmente Python posee diferentes tipos de librerías para el desarrollo de interfaz gráfica de usuario (GUI), el cual se diferencia por los diferentes conjuntos de elementos gráficos y de control sobre el comportamiento de la interface, debido a esta necesidad en específico se escogió una librería incorporada en Python llamada Tkinter que facilita a los programadores desarrollar una interfaz gráfica de usuario, ya que esta ofrece una completa documentación de su manejo. En relación con esta librería se pueda implementar el control de ventanas emergentes, visualización de variables en tiempo real, graficación de datos, como también la viabilidad de cambiar diferentes parámetros que requiera el programa, como por ejemplo la modificación de la API'S key de escritura de Thingspeak. De este modo Tkinter ofrece recursos para las necesidades que pueda presentar el programa.

3.3 DESARROLLO DEL PROGRAMA

Para el inicio en el desarrollo de este aplicativo es necesario aclarar que existen una gran cantidad de editores de texto que permiten la construcción de código como facilitadores para el lenguaje Python. Sin embargo, para el desarrollo de este proyecto se ha utilizado el editor de texto que viene incorporado en Raspberry Pi, denominado nano, este es un editor que señala y resalta palabras propias de Python únicamente y es manejado directamente por consola.

Para la construcción de la interfaz gráfica es necesario tener en cuenta que Tkinter es una biblioteca, que posee integrada una gran cantidad de recursos que facilitan el diseño, así mismo que su desarrollo en el código es secuencial y maneja jerarquías para el control de ventanas. Además de Tkinter es necesario importar e instalar bibliotecas externas que no posee la Raspberry Pi, como la librería Adafruit_DHT la cual provee el protocolo de comunicación I^2C para la captación de datos provenientes del sensor DHT11, urllib2 una librería que facilita el manejo y

comunicación a paginas por medio de HTML, y matplotlib esta es la que permite agregar el desarrollo de gráficas y la animación en tiempo real.

Para el desarrollo de la interfaz gráfica del programa se inició con una versión de prueba, la cual adquiere las variables de humedad relativa y temperatura del aire por medio del sensor DHT11, que son graficadas con respecto al número de muestras respectivamente. Las variables obtenidas a su vez son enviadas al aplicativo web ThingSpeak, esto con el fin de asegurar un respaldo de los datos que son censados. La versión de prueba fue desarrollada con el fin de ver el tipo de interfaz que se puede crear y como se puede mejorar para ser adaptable ante diferentes usuarios con condiciones de entorno distintas ejemplo (Trabajo, estudio, hogar). Así mismo ver el tiempo de muestreo mínimo con el cual se puede realizar la aplicación, para este caso son 2 segundos ya que este es el tiempo de muestreo mínimo del sensor DHT11. La interfaz de prueba se puede apreciar en la Figura 12.

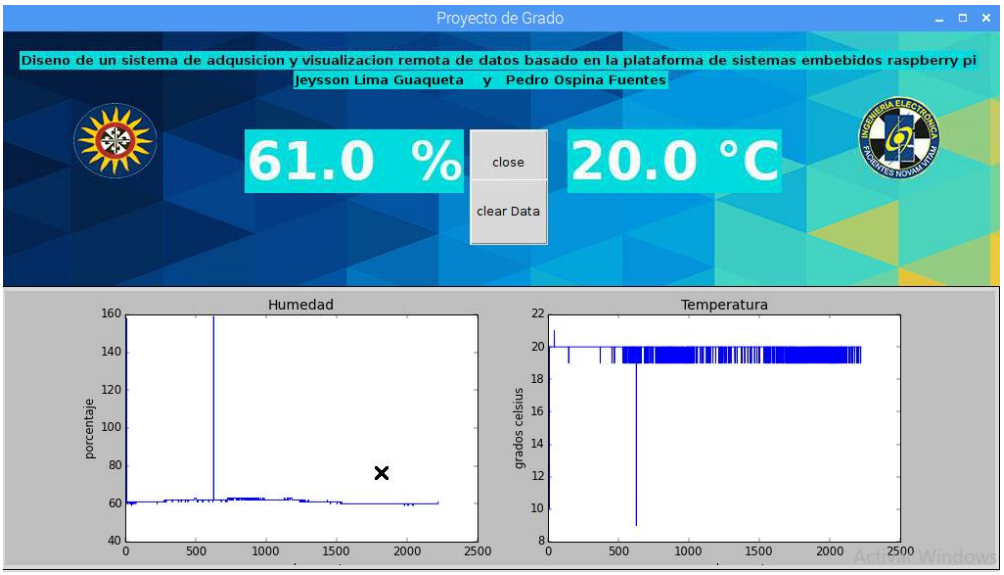


Figura 12. Interfaz de Usuario Versión Beta.

El segundo aspecto que se desarrollo fue la creación de un canal en Thingspeak, para el caso de este proyecto se utilizó una suscripción gratuita debido a que es una

versión prototipo, que busca reducir costos para la inclusión de estudios como el confort térmico en el concepto de la IoT.

Para enviar los datos adquiridos de los sensores a Thingspeak, es necesario el uso de la librería Urllib2, este módulo permite que Python busque por medio del URL la página a la cual va a ser enviardos los datos, por esta razón es importante tener en cuenta la clave API's Key de escritura, debido que permite acceder y almacenar en los campos del canal. De igual forma es necesario resaltar que para este proyecto solo se utilizaron 2 campos que representa los datos de humedad relativa y temperatura del aire. La implementación de esta librería se puede apreciar en la figura 13.

```
baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI
        f = urllib2.urlopen(baseURL + "&field1=%s&field2=%s" % (h, t))
```

Figura 13.Implementación de la librería Urllib2 en Python

Al tener previsto el diseño desarrollado en la primera versión del programa, se plantea el cumplimiento de los requerimientos que hacen falta para el aplicativo, uno de ellos es la opción que se le ofrece al usuario para configurar algunos parámetros del aplicativo, como por ejemplo: escoger el puerto de propósito general para conectar el sensor DHT11, tener la opción de modificar la API key de escritura de Thingspeak y tener la opción de modificar el nivel de altitud en el que se va a hacer la medida, teniendo en cuenta lo visto en el capítulo anterior a partir de las ecuaciones del índice de confort 4, 5, 6. Posteriormente darle un valor agregado al aplicativo al hacer uso de los recursos ofrecidos por Tkinter como el de ventanas emergentes.

El segundo aspecto que hace falta agregar a partir de los requerimientos es la visualización de la sensación del confort térmico, según el estudio realizado por el IDEAM se plantean dos tipos formulas diferentes, una que considera la velocidad del aire y otra que no. Para el caso de este proyecto se utilizarán las formulas 4, 5,

y 6 de la sección 2.1, en el cual se determina el índice de confort térmico a partir de tres variables, la humedad relativa, la temperatura del aire, y el nivel de altitud. A partir del resultado obtenido en el cálculo del índice de confort se plantea visualizar la clasificación de sensación térmica.

Posteriormente al plantear los requerimientos que faltaban en la versión beta se inicia el desarrollo del aplicativo final. Como presentación se creó una ventana emergente temporal que muestra el título del proyecto, los integrantes y el escudo de la Universidad por 5 segundos, después de iniciar el programa de la Interfaz gráfica de usuario como se puede apreciar en la figura 14.

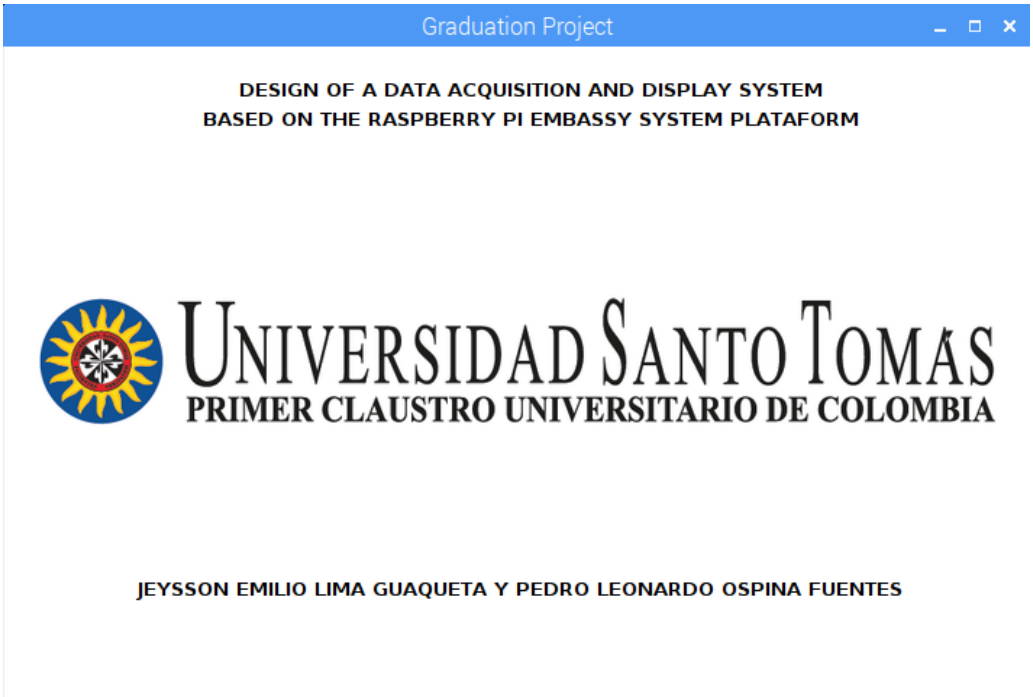


Figura 14. Ventana de Presentación

Como pantalla de inicio se volvió a distribuir y se organizó la interfaz de usuario, de tal manera que sea agradable visualmente y sencilla en su manejo. La clasificación bioclimática del confort térmico es visualizada en la parte central de la pantalla resaltado de azul, la temperatura del aire y la humedad relativa son colocados sobre su grafica respectivamente.

Se crearon 5 botones para la interfaz. El primer botón denominado Start, da inicio al muestreo, graficación, determinación del confort térmico y él envió de datos a Thingspeak. El segundo botón es Stop, este genera un evento que para toda la secuencia del programa y hace que la interfaz gráfica de usuario espere a un nuevo evento. El tercer botón es Clear, este reinicializa las gráficas de la interfaz, borrando los datos que se adquirieron hasta ese momento y empezando de cero el valor de las muestras. El cuarto botón es Config este botón me permite activar una ventana emergente que administra las variables que puede modificar el usuario, como: el nivel de altitud de la medición en metros, el número de pin por el cual se va a conectar la señal de comunicación del DHT11 y por último la dirección del API en ThingSpeak que si lo desea el usuario puede ingresar. El quinto botón es aquel que cierra todos los procesos y finaliza el programa. La ventana principal de usuario se puede ver en la figura 15.

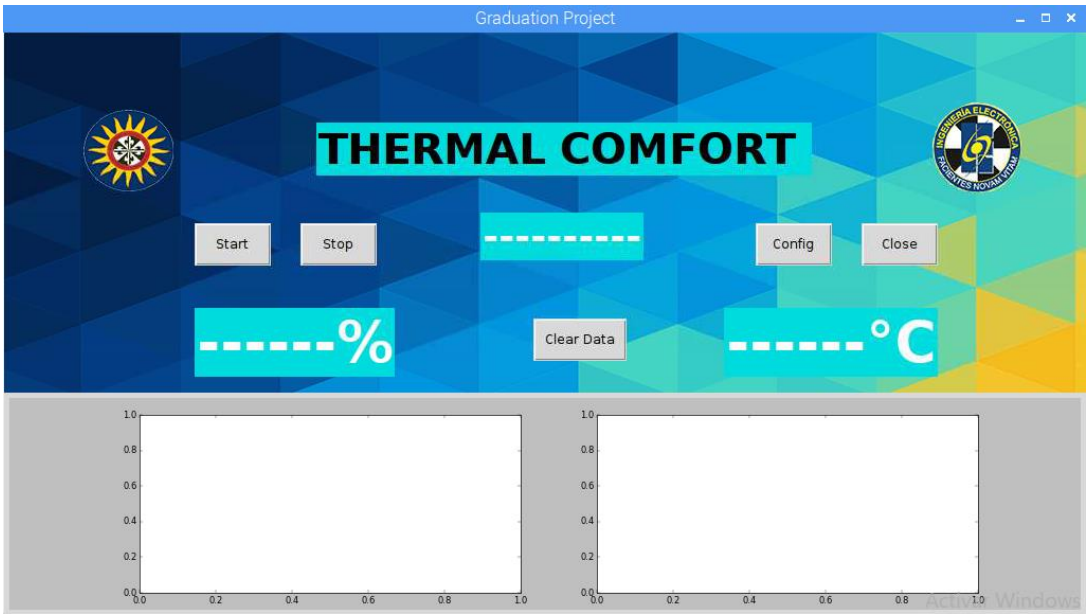


Figura 15. Interfaz gráfica de usuario principal

En el momento en que se genera el evento de Configuración, detendrá los demás procesos en el aplicativo, automáticamente la interfaz gráfica de usuario hereda el control a esta ventana de configuración, con el fin de solo permitir la modificación

de los parámetros que se encuentran en ella, únicamente se podrá salir de ella dándole clic en el botón ok o en el sub botón 'x'. La ventana de configuración posee dos eventos de botón, el primero es verify pin que permite la verificación del GPIO por el cual se comunicara con el sensor DHT11, el segundo botón es verify HTML que consiste en la verificación del API Key ingresado, en cual determina si es posible enviar datos a Thingspeak. La ventana de configuración se puede apreciar en la figura 16.

Figura 16. Ventana de Configuración del aplicativo

Como requerimientos para garantizar el correcto funcionamiento y estabilidad del programa en su entorno visual se agregaron ventanas emergentes que permitan informar al usuario las acciones o fallas que pueda presentar el programa.

Un requerimiento necesario en el programa fue debido al botón de 'x' del aplicativo ubicado en la parte superior derecha, puesto que este permitía destruir la interfaz gráfica de usuario, pero no cerraba el proceso general que se estaba realizando. Debido a esto se solucionó generando un evento que permitiera preguntarle al usuario si estaba seguro de salir, si es afirmativo, cerraba la aplicación general, si

cancelaba la acción retornaba a la ventana principal del programa. La ventana emergente se puede apreciar en la figura 17.

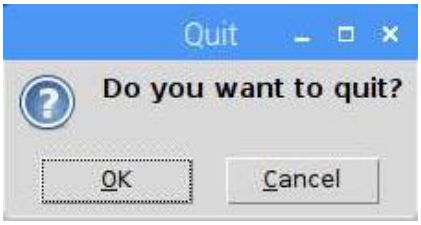


Figura 17. Ventana de Dialogo si desea cerrar el programa.

El segundo requerimiento necesario implementado en el programa fue debido a errores de conexión en la red Wifi, es decir si la conexión a internet no era del todo estable o presentaba caídas momentáneas el aplicativo se bloqueaba debido a este fallo, como solución se planteó por medio de una ventana emergente informar al usuario que existió un error de red y retorna la interfaz a modo de espera. La ventana emergente se puede apreciar en la figura 18.

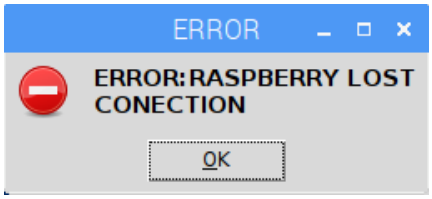


Figura 18. Ventana de Perdida de conexión.

3.4 MANEJO DE CONTROL DE ERRORES

Como manejo de control de errores se catalogaron en dos las posibles causas de falla en el aplicativo, La primera son fallas internas que son provocadas internamente en el sistema de adquisición de datos, y el segundo tipo de fallas son externas por mal manejo en el programa o problemas externos.

3.4.1 MANEJO DE FALLAS INTERNAS

Como fallas internas que puede presentar el programa se encontraron variables atípicas que captura el sensor DHT11. Este tipo de variables se puede reconocer en la figura 10, debido a que son datos que salen fuera del rango promedio respecto a las variables de humedad relativa y temperatura del aire. Normalmente es un error típico que para sensores que son sometidos a trabajos de larga duración. Como solución a este problema se planteó la implementación de un filtro pasa bajo digital con una brecha que no supere 10 unidades sobre el valor promedio. Esto asegurando que no permitirá graficar y enviar a Thingspeak, estos datos erróneos.

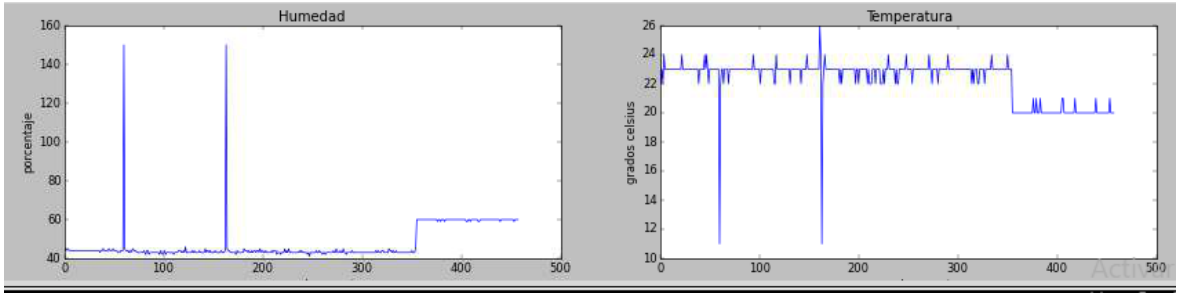


Figura 19. Datos muestreados sin filtro digital

Para finalizar con lo propuesto de la implementación de un filtro digital, se planteó el siguiente diagrama de flujo que se puede apreciar en la figura 20. El diagrama funciona de la siguiente manera: Primero se iniciará almacenando 10 datos adquiridos de humedad y temperatura a medida que trabaja el sensor, como paso a seguir se aplica la media a estos 10 últimos datos adquiridos y se almacenara como “Promedio”, luego se planteó una “brecha” que no supere 10 unidades, puesto que se determinó que no existe cambios abruptos en el medio que lleguen a esta escala. En el momento que se reciba el siguiente dato, el filtro lo capturara y almacenara este valor como “Dato presente”, posteriormente sumara los datos de “Promedio” y la “brecha” almacenándolo como valor “Limite”. Por consiguiente, se aplicó una condición que comparara el “Dato presente” con “Limite” en donde tiene dos casos: Si el “Dato presente” es mayor al límite entonces omita este valor y

asigne el valor anterior, caso contrario si el “dato presente” es menor entonces almacene el dato y siga muestreando.

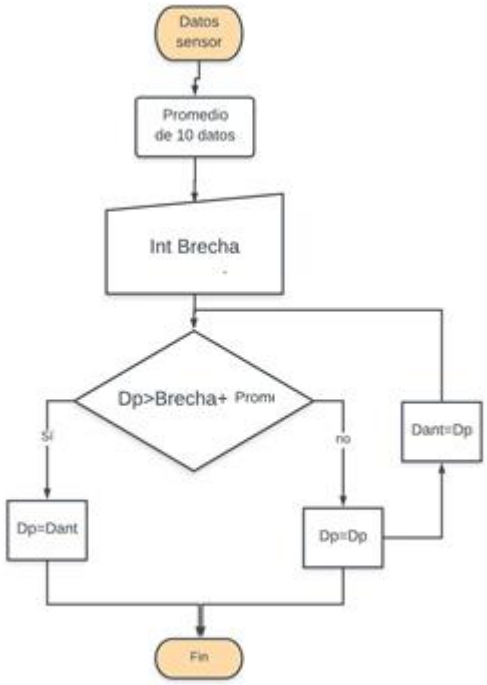


Figura 20. Diagrama de flujo del filtro digital pasa bajo implementado.

Por último, se implementó el filtro pasa bajo en el programa, para evidenciar el correcto funcionamiento del filtro se inició una prueba de muestreo de datos, que se puede evidencia en la figura 21. En comparación con la figura 19 ya no se presentan evidencia de datos atípicos captados por el sensor DHT11.

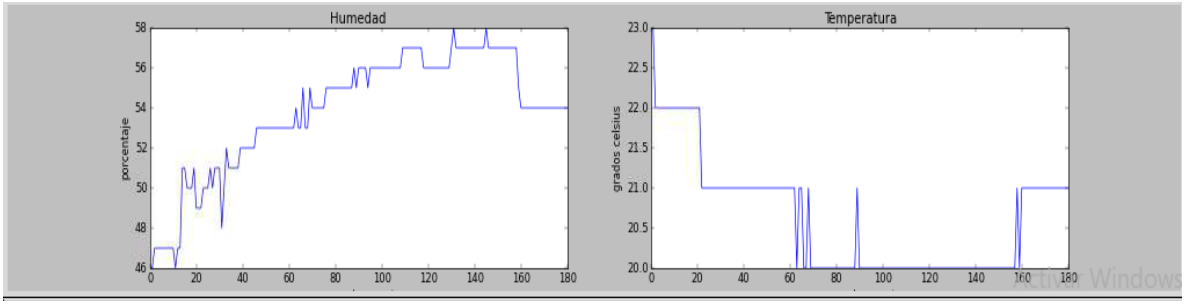


Figura 21. Datos muestreados con filtro digital

3.4.2 MANEJO DE FALLAS EXTERNAS

Como manejo de fallas externas se determinó que son aquellas que pueden ser generadas por el usuario, estas pueden ser provocadas en el momento de controlar o ingresar parámetros en la interfaz gráfica. Por tanto, como requerimiento para asegurar un control de errores, se crearon ventanas emergentes que informan al usuario si alguna acción ha provocado un fallo en el programa.

El primer control de error es debido a que la tarjeta Raspberry Pi no pueda comunicarse con el sensor DHT11, sea provocado por que el sensor no está conectado o es desconectado en el momento que se están adquiriendo datos, Para este caso fue implementado una ventana de información advirtiendo que el sensor no se está adquiriendo datos del sensor. La ventana generada se puede apreciar en la figura 22.

Figura 22. Ventana de Error, Sensor no conectado

El segundo control de error puede ser producido debido a que el usuario no confirma la inicialización o configuración de parámetros para la ejecución del aplicativo, generando una ventana emergente donde solicita que confirme los parámetros de la ventana de configuración. Esta ventana solo se aparecerá si el usuario intenta iniciar el programa por primera vez. La ventana de información generada se puede ver en la figura 23.

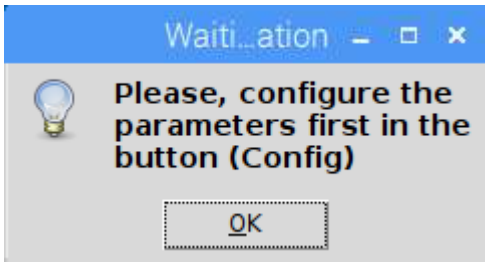


Figura 23. Ventana de información para actualizar parámetros de inicio

El tercer control de error es considerado en la ventana de configuración, y puede ser generado en la casilla Pin GPIO for DHT11, debido a que el usuario puede ingresar cualquier valor numérico en esta casilla. Con el fin de advertir al usuario que la Raspberry solo posee puertos de propósito general del 2 al 27, se creó una ventana emergente que informa que el número de PIN es incorrecto. La ventana generada se puede apreciar en la figura 24.

Figura 24. Ventana de Error, Puerto GPIO es incorrecto

El cuarto error puede ser generado en la ventana de configuración. El parámetro de modificación que permite escoger el puerto GPIO que adquiere los parámetros de humedad y temperatura, permite el ingreso de diferentes tipos de caracteres; sin embargo, para controlar que solo permita el ingreso números enteros, se creó una ventana emergente que informa al usuario que ha cometido un error en el momento de ingresar la información. La visualización de esta ventana se puede ver en la figura 16.

Figura 25. Ventana de Error Valor ingresado no es correcto.

En el dado caso que el valor ingresado en la casilla Pin GPIO for DHT11 le informa al usuario que es correcto el pin seleccionado para proceder con la configuración. La ventana de información generada se puede apreciar en la figura 17.

Figura 26. Ventana de información, A sido seleccionado el GPIO

El quinto error puede ser generado en la ventana de configuración, este fue desarrollado con el fin de confirmar que la API Key ingresada por el usuario existe y tiene accesibilidad para recibir datos. En el dado caso que la API Key no es válida se genera una ventana emergente informando que no se pudo conectar con la pagina el canal de almacenamiento que ingreso. La ventana emergente generada se puede ver en la figura18.

Figura 27. Ventana de Error. API Key ingresado no es correcto

Si el valor ingresado en la casilla es correcto, la aplicación informa al usuario que el API Key es válido y se pudo comunicar con el HTML de Thingspeak de tal manera que se pueden aceptar los parámetros ingresados e iniciar el muestreo de datos en la ventana principal. La ventana generada que confirma la comunicación con Thingspeak se puede apreciar en la figura 19.

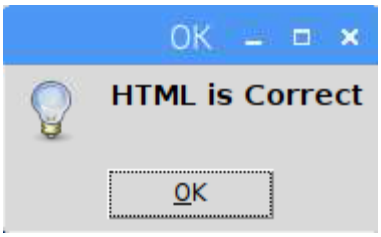


Figura 28. Ventana de Información, API ingresado es válido.

4. RESULTADOS DEL PROYECTO

Como resultados del proyecto se hace énfasis al funcionamiento final del aplicativo explicando su conexión y visualización de las pruebas tanto en el aplicativo como en Thingspeak, cuánto tiempo puede trabajar el aplicativo y la facilidad de usarlo en entornos cotidianos.

4.1 FUNCIONAMIENTO

En la prueba de funcionamiento, se implementó el sistema general planteado en el proyecto para utilizar el aplicativo, de tal manera que utilizando la tarjeta Raspberry Pi 3 se conectó una pantalla Touch de 7'' un teclado inalámbrico portable y el sensor de temperatura DHT11, lo que permite la portabilidad del sistema. La conexión general se puede ver en la figura 29.

Figura 29. Implementación general.

Para el funcionamiento del sistema de este proyecto se requiere la conexión a una toma de alimentación monofásica para energizar la Raspberry Pi 3 y una conexión a internet vía wifi o por cable de Red Ethernet.

Las pruebas realizadas se centraron en entornos comunes residenciales de un ser humano, con el fin de comprobar que el sistema implementado cumple con los requerimientos planteados en el desarrollo del proyecto. Las pruebas realizadas para este proyecto se realizaron en diferentes sitios de un hogar para ver el comportamiento en tiempo real. Es importante aclarar que los datos de humedad relativa y temperatura del aire captados en el momento de muestreo, son utilizados para determinar el índice de confort, este índice no es visualizado en la aplicación final, únicamente es visualizado la sensación térmica mencionada en la Tabla 1.

Una de las pruebas que se realizaron en una Sala de estar en horarios de la mañana consistió, en comparar las gráficas obtenidas en el aplicativo y las que eran graficadas en la página Thingspeak, con el fin de comprobar el correcto funcionamiento de almacenamiento de la interfaz y la transferencia de datos al canal de Thingspeak, la comparación de estas graficas se pueden ver en la figura 30 y figura 31.

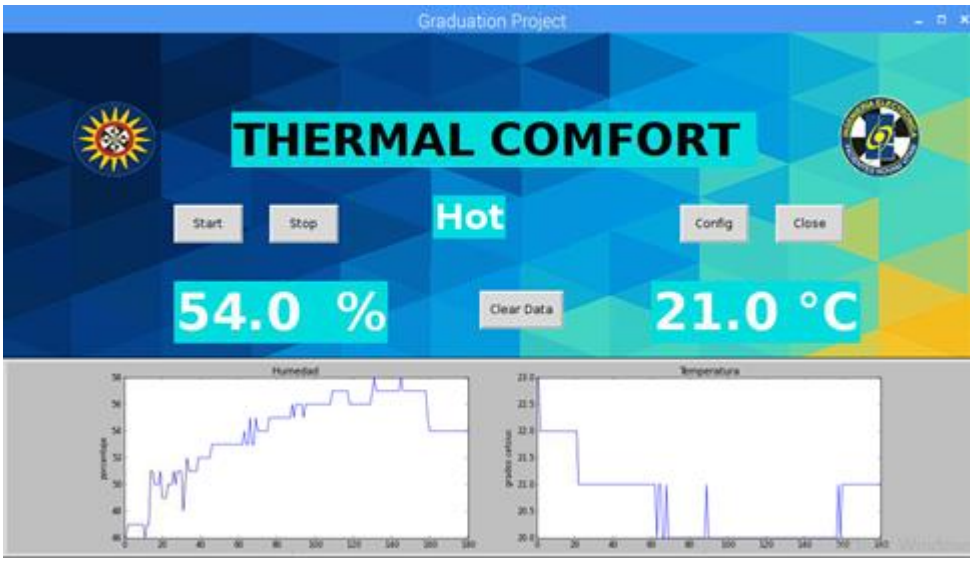


Figura 30. Interfaz de usuario en modo de funcionamiento

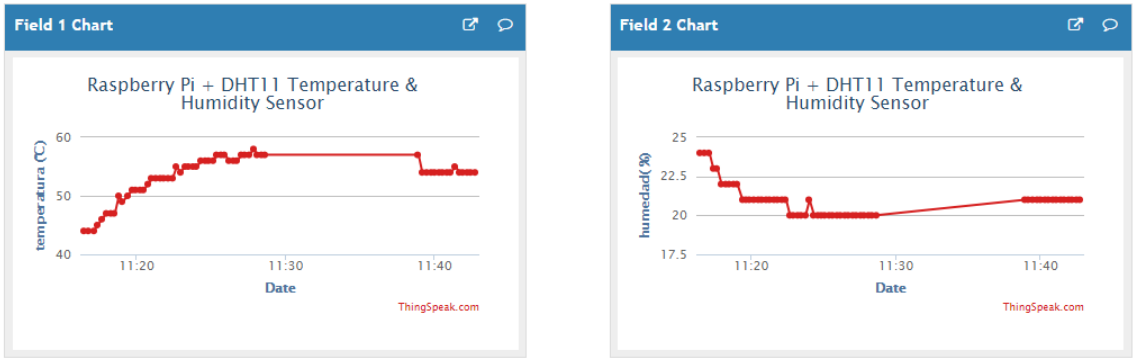


Figura 31. Datos de humedad y temperatura visto desde Thingspeak

Para ver los datos de humedad relativa y temperatura del aire almacenadas en el canal de Thingspeak creado para este proyecto, se pueden visualizar en la dirección web del canal del proyecto en el Anexo F. Es importante tener en cuenta que el canal de Thingspeak se están actualizando cada 15 segundos y la interfaz de usuario envía datos cada 2 segundos, por lo que produce que su visualización no sea simultánea.

Thingspeak permite almacenar hasta 67 datos en cada uno de los campos lo cual permite visualizar datos en un intervalo de 18 minutos aproximadamente, por lo que cabe aclarar que como requerimiento de la interfaz de usuario es que permite graficar los datos obtenidos desde el momento de inicio de la adquisición de datos hasta el momento que se para o se cierra el programa.

La segunda prueba se realizó en un cuarto residencial esto con el objetivo de comprobar el correcto funcionamiento del filtro implementado en el aplicativo, se realizó una prueba que consiste en hacer trabajar el programa en largos ciclos de tiempo, por el cual se sometió a una prueba por más de 5 horas con el fin de comprobar que no existan datos atípicos captados en la graficación de los datos de humedad relativa y temperatura del aire. Las gráficas obtenidas se pueden apreciar en la figura 32.

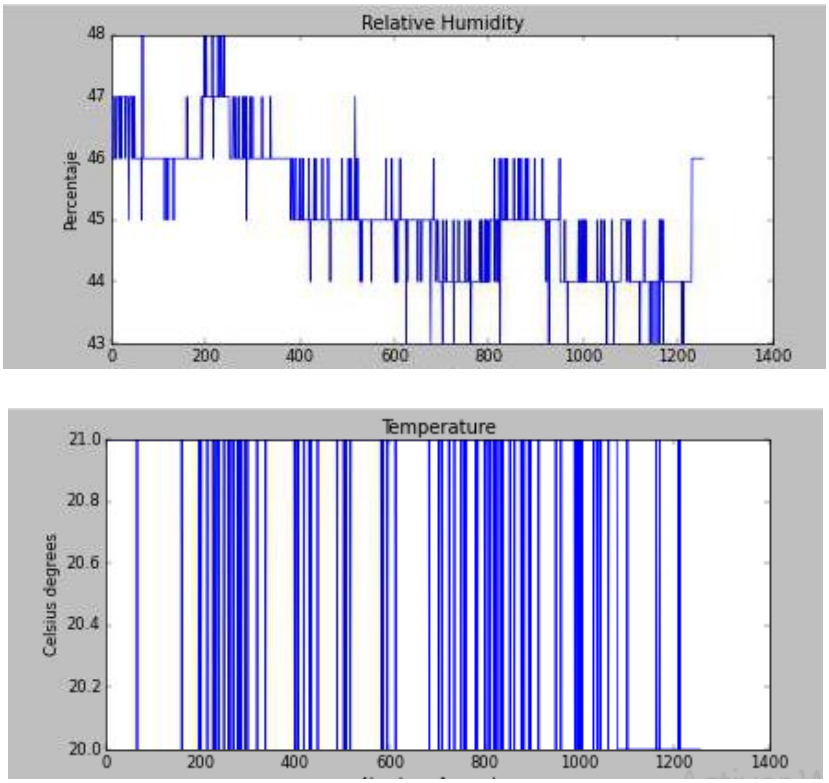


Figura 32. Graficacion de datos en un ciclo de 5 horas.

El aplicativo puede funcionar 24 horas al día siempre y cuando exista una conexión estable a internet, debido a que si se pierde la comunicación con Thingspeak el programa automáticamente detendrá el proceso de adquisición y procesamiento de datos y esperara a que el usuario vuelva a dar inicio al proceso, como se mencionó en la explicación de la figura 18. El código generado en este proyecto se podrá ver en el Anexo G de este documento permitiendo para futuros proyectos el acceso para realizar modificaciones o mejoras.

5. IMPACTO SOCIAL

Este proyecto busca sentar las bases y abrir caminos en campos de estudio que aún no forman parte del concepto de la IoT. Por tal razón se escogió el tema de la ergonomía ambiental para determinar el confort térmico. Gracias a investigaciones realizadas por el IDEAM ahora es posible estimar la sensación térmica experimentada en un país tropical como Colombia, se escogió este estudio con el fin de apoyar investigaciones realizadas en el país. Como estrategia tecnológica este proyecto plantea un sistema de adquisición de datos para la determinación del confort térmico promoviendo la solución de problemas a partir de herramientas de software y hardware libre, para el desarrollo de aplicaciones basadas en investigaciones nacionales.

El grado de bienestar de las personas y su calidad de vida pueden ser determinados, debido a que el confort térmico es un estudio que permite catalogar la sensación térmica del ser humano. Teniendo en cuenta estos tipos de estudios ahora es posible plantear el mejoramiento de espacios cotidianos, a partir de mediciones cuantificables como el índice de confort que puede ser calculado a partir de datos de temperatura y humedad. Por medio de este proyecto se busca incentivar el desarrollo en sistemas basados en la tarjeta Raspberry Pi, ya que es una herramienta que puede utilizarse para la solución de diferentes problemas que requieran conectividad y muestreo en tiempo real, ahora es posible fomentar la inclusión de estudios que requieran ser portables para la implementación de diferentes entornos, para este caso la determinación de la sensación de confort térmico en espacios cotidianos.

Gracias a estos tipos de desarrollo ahora es posible para aquellas empresas y/o personas que busquen innovar en diversas áreas de ingenierías la implementación de este tipo de sistema para poder argumentar el planteamiento de un problema, como por ejemplo la deficiencia de ventilación en entornos de trabajo, esto por medio del cálculo del confort térmico permite argumentar si una zona de trabajo no

es apta y para el caso de un ingeniero pueda plantear una solución a este problema. En otro caso se puede utilizar para determinar el confort térmico de estudiantes en espacios académicos, con la finalidad de poder argumentar si los espacios de estudios suministrados por la institución académica, permiten tener un ambiente apto para el aprendizaje.

6. CONCLUSIONES Y TRABAJO FUTURO

La finalización de este proyecto permite concluir que por medio de datos de temperatura del aire y humedad relativa capturados por un sensor DHT11, es posible estimar la sensación térmica experimentada en el entorno. Por el cual gracias a la tarjeta de desarrollo Raspberry Pi es posible generar aplicativos con interfaz gráfica de usuario para la adquisición de datos en tiempo real, gracias a esto es posible el procesamiento de estos datos, para ser almacenados en la nube y adicionalmente permitir determinar el índice de confort térmico en un espacio cerrado.

Este proyecto permite evidenciar que la utilización de herramientas de software libre junto con la tarjeta de desarrollo Raspberry Pi, permite crear un sistema de adquisición de datos, que incentive la generación de proyectos para la implementación de estudios en diferentes campos, que requieran monitoreo de datos de forma remota o local como uso profesional.

El proyecto realizado se desarrolló de forma básica y sencilla, permitiendo ser accesible a modificaciones, cambios o mejoras en el momento de ser incorporado para la creación de otros trabajos de grado.

El presente proyecto permite la utilización de nuevas tecnologías a aplicaciones relacionadas con la climatología para la toma de datos del confort térmico en diferentes zonas del país, como también permite para estudios futuros realizar una medición de impacto en respuesta al ambiente en cuanto al comportamiento humano o a la eficiencia en diferentes tipos de entornos como; trabajo, estudiantil, hogar, sistemas públicos, etc.

Por otro lado, los resultados del presente trabajo demuestra la facilitad en la toma de datos de temperatura ambiente y humedad relativa en otros tipos de entornos, en cuanto a la agronomía para desarrollar futuros proyectos en la toma de decisiones y regulación en ambientes controlados de cultivos, la zoología en

criaderos de animales de granja y en sistemas de monitoreo para el control de temperatura y humedad en animales de cautiverio, la industria para sistemas de producción que necesiten toma de decisiones en tiempo real, adquisición de variables de forma remota o una solución de basado en software libre que permita suplir las necesidades que se presenten posibilitando el desarrollo de proyectos en el contexto del paradigma de industria 4.0, la cual es una tendencia actual de la automatización de los procesos y redes industriales [55].

REFERENCIAS

- [1]Gaviria Cuevas, C., Ordoñez Oliveros, J., Montenegro Narváez, C., & Universidad Santo Tomás Facultad Ingeniería de Telecomunicaciones. Especialización en Gestión de Redes de Datos. (2014). Estudio comparativo del internet de las cosas frente a los protocolos tradicionales de la domótica y propuesta de un protocolo unificado.
- [2] Pinzón Niño, D., Zona Ortiz, Universidad Santo Tomás Facultad de Ingeniería de Telecomunicaciones. (2016). Panorama de aplicación de internet de las cosas (IoT).
- [3] Colmenares-Guillen, E., Carrillo, M., Edilberto, R., & Nino, H. (2015). Aplicación de computo móvil y pervasivo para el monitoreo no invasivo de la diabetes en tiempo real. 11(33), 130.
- [4] Sánchez-Alcón, J., López-Santidrián, L., & Martínez, J. (2015). Solución para garantizar la privacidad en internet de las cosas. El Profesional De La Información, 24(1), 62-70.
- [5] Cortés -, Javier. (2015). Internet Industrial de las Cosas. Bit, (199), 46-48.
- [6] Sevilla, Ana, Castaño, María, & Cervantes, Elisa. (2011). Internet de las Cosas y Salud. Bit, (187), 49-52.
- [7] Segura, A. (2016). Arquitectura de Software de Referencia para Objetos Inteligentes en Internet de las Cosas. Revista Latinoamericana De Ingeniería De Software, 4(2), 73.
- [8] ArakistainMarkina, I. (2015). Engergy Harvesting: Captando Energía para el internet de las cosas y los sensores que las vigilan, Dyna Ingeniería e Industria 90(3), 300-306.
- [9] V. K. Sehgal, S. Mehrotra and H. Marwah, (2016). Car security using Internet of Things. *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, India. 1-5.
- [10] P. K. Rahulkrishna, R. Eshwari, N. J. S. Harsha and R. Hegde. (2016). Design and development of remote load monitoring suitable for non-residential loads through wireless data transmission - *IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Mangalore. 35-40.
- [11] V. Sandeep, K. L. Gopal, S. Naveen, A. Amudhan and L. S. Kumar.(2015). Globally accessible machine automation using Raspberry pi based on Internet of Things, *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi. 1144-1147.
- [12] S. J. Johnston, M. Apetroaie-Cristea, M. Scott and S. J. Cox. (2016). Applicability of commodity, low cost, single board computers for Internet of Things devices, *IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA. 141-146.

- [13] (2015). A Smart Home Automation Technique with Raspberry Pi using IoT, International Conference on Smart Sensors and Systems (IC-SSS)
- [14] (2016). Automated Energy Meter Using Wi-Fi Enabled Raspberry Pi, IEEE International Conference On Recent Trends In Electronics Information Communication Technology, India.
- [15] (2016). IoT Based Urban Climate Monitoring using Raspberry Pi, International Conference on Communication and Signal Processing India. 1-978
- [16] Daniela Eduardo Martínez (2016), Sistema de domótica para control y supervisión de una habitación de manera remota, Bogotá.
- [17] K. O. Flores, I. M. Butaslac, J. E. M. Gonzales, S. M. G. Dumlao and R. S. J. Reyes, (2016). Precision agriculture monitoring system using wireless sensor network and Raspberry Pi local server. I*EEE Region 10 Conference (TENCON)*, Singapore. 3018-3021.
- [18] X. Q. Li, X. Ding, Y. Zhang, Z. P. Sun and H. W. Zhao.(2016). IoT Family Robot Based on Raspberry Pi, International Conference on Information System and Artificial Intelligence *(ISAI)*, Hong Kong. 622-625.
- [19] Alfonso Godoy Muñoz, Universidad Politécnica de Cataluña (2012) Confort térmico adaptativo.
- [20] Iosu Otazu Larrasoaña (2010) Influencia de temperatura y tiempo de secado en calidad de las hojas. Calidad de las hojas en Cymbopogon Citratus.
- [21] Soto F; Morales, D.(1996) Relación de la temperatura del aire, la humedad relativa y la radiación global con el crecimiento de plántulas de cafeto.
- [22] Abalone, R. M., et al. (2006) Modelización de la distribución de la temperatura y humedad en granos almacenados en silos. Mecánica Computacional, vol. 25. 233-247.
- *[23] R. Bhilare and S. Mali.(2015)* IoT based smart home with real time E-metering using E-controller, Annual IEEE India Conference (INDICON)*, New Delhi. 1-6*
- *[24] P. P. Ray.* (2015) Internet of Things based smart measurement and monitoring of wood Equilibrium Moisture Content, *International Conference on Smart Sensors and Systems (IC-SSS)*, Bangalore. 1-5.
- [25] A. Khanna and R. Tomar,(2016) IoT based interactive shopping ecosystem, 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, India. 40-45.*
- *[26] O. Bates and A. Friday.(2017).Beyond Data in the Smart City: Repurposing Existing Campus IoT, in* IEEE Pervasive Computing*, vol. 16, no. 2. 54-60.*
- [27] Pinzón Niño, David Leonardo. (2016). Universidad Santo Tomas Panorama de aplicación de internet de las cosas (IoT).
- *[28]* Gonzalez, K., Urrego, D., & Gordillo, W. (2016). Estudio sobre Computadores de Placa Reducida Raspberry Pi Modelo B y Cubieboard2 en la creación de proyectos con base tecnológica de gran impacto social. *ENGI Revista Electrónica de la Facultad de Ingeniería*, *3*(1).

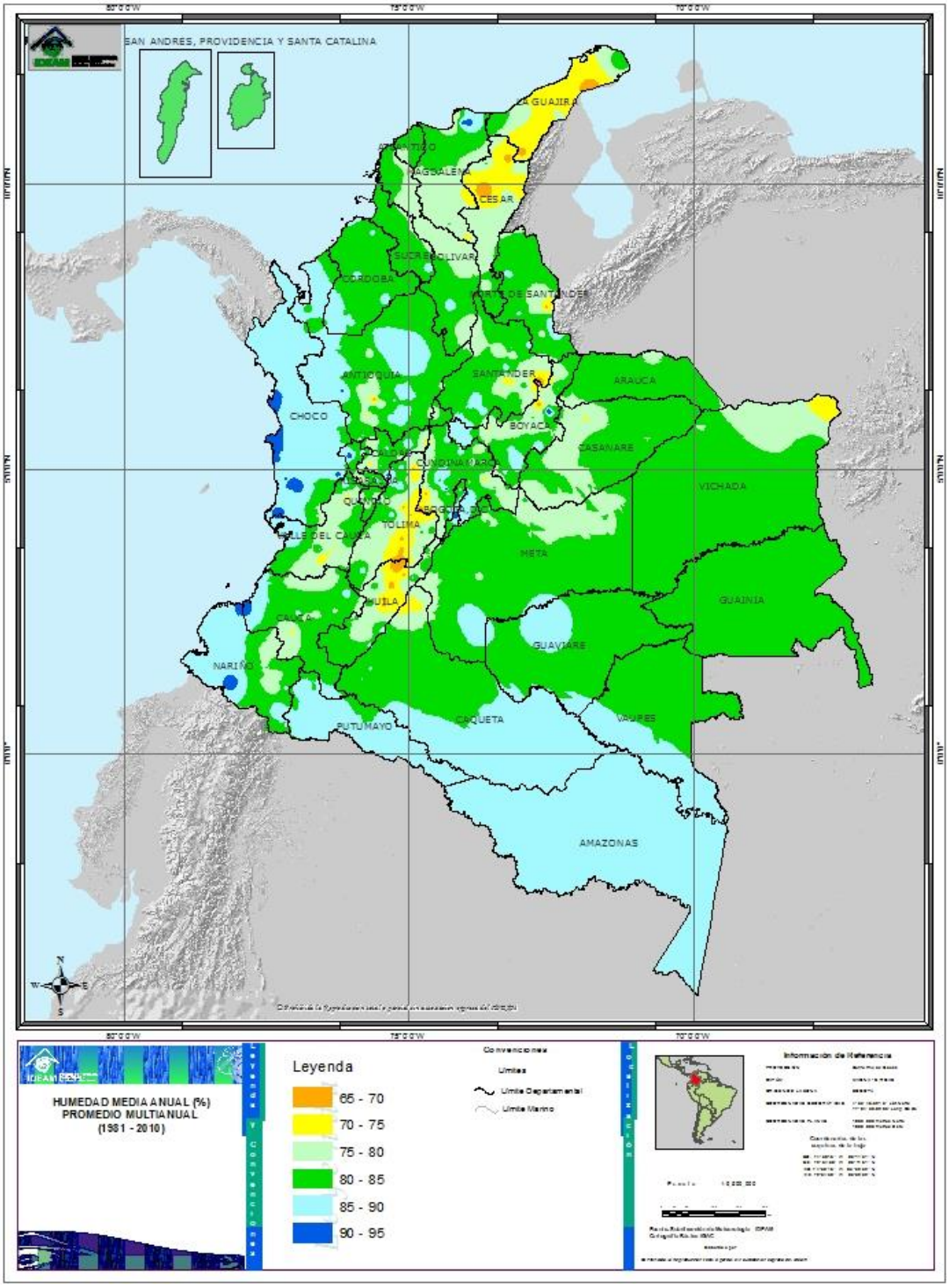
- [29]S. N. Jyothi and K. V. Vardhan.(2016). Design and implementation of real time security surveillance system using IoT, International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore. 1-5.*
- *[30]* E. Laftchiev and D. Nikovski*,(2016). An IoT system to estimate personal thermal comfort.* IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA,672-677.-K. Laubhan, K. Talaat, S. Riehl, T. Morelli, A. Abdelgawad and K. Yelamarthi. (2016)* A four-layer wireless sensor network framework for IoT applications," *IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Abu Dhabi. 1-4
- [31] Dennis, A. K. (2013). Raspberry Pi home automation with Arduino. Packt Publishing Ltd
- [32] Hugues, J. H. J. D. A. (2013). Embedded Systems. Somerset: John Wiley & Sons, Incorporated.Pag(1-2)
- [33] Kothari, D. S. K. S. (2011). Embedded Systems. Daryaganj: New Age International.Pag (1)
- [34] Kothari, D. S. K. S. (2011). Embedded Systems. Daryaganj: New Age International.Pag(3)
- [35] Dennis, A. K. (2013). *Raspberry Pi Super Cluster*. Packt Publishing Ltd.
- [36] Hossain, M. I., & Ghose, R. (2017). *IoT Based Information Collection System of Victims From Crash Sites* (Doctoral dissertation, East West University).
- [37] Rasbian.Org (2017) GPIO- Raspberry Pi Documentation N.p.2017. Web. 21 Mar. 2017
- [38] Cox, T. (2014). Raspberry Pi Cookbook for Python Programmers. Olton: Packt Publishing.
- [39] Rasbian.Org (2017). Raspbian- Raspberry Pi Documentation N.p.2017. Web. 21 Mar. 2017
- [40] Ramon Pallas,(2013),Sensores y Acondionadores de Señales, Marcombo. Pag (2)
- [41] Ramon Pallas,(2013),Sensores y Acondionadores de Señales, Marcombo. Pag (3)
- [42] Chazallet, S. (2016). *Python 3: los fundamentos del lenguaje*. Ediciones ENI.
- [43] Erazo, P., Jennyfer, K., Hervas, P., & Carlos, A. (2014). Sistema de detección de incendios forestales mediante redes sensoriales inalámbricas (Zigbee).
- [44] Hersent, O. B. D. E. O. (2011). Internet of Things. Pag(1)
- [45] Vermesan, O. F. P. (2013). Internet of Things. Pag(2)
- [46] Ortiz, L., & Nelson, R. (2017). Internet de las cosas.
- [47] Waher, P. (2015). Learning Internet of Things. Olton Birmingham.Pag(19)
- [48] Olga C Gonzales.(1998) Nota tecnica IDEAM. Metodoloía para el Calculo del Confort Climático en Colombia. Pag (5)
- [49] NTC 5316. (2008). Condiciones ambientales térmicas de inmuebles para personas. Colombia: ICONTEC, 2004. 1p

- [50] ANSI/ASHRAE. (2010). Thermal Environmental Conditions for Human Occupancy. Definitions.
- [51] UNE-EN ISO 7730 (2006) Ergonomía del ambiente térmico. Determinación analítica e interpretación del bienestar térmico mediante el cálculo de los índices PMV y PPD y los criterios de bienestar térmico local (ISO 7730:2005).
- [52] Godoy Muñoz, A. (2012). *El Confort térmico adaptativo. Aplicación en la edificación en España* (Master's thesis, Universitat Politècnica de Catalunya).
- [53] Gualoto, T., & Esteban, M. (2018). *Diseño de una red WSN para el monitoreo óptimo de personas en el hospital Eugenio Espejo de Quito* (Bachelor's thesis, Quito: Universidad de las Américas, 2018).
- [54] ThingSpeak. (2018) Internet of things – ThingSpeak. https://www.thingspeak.com
- [55] Catalán, C., Serna, F., & Blesa, A. (2015, July). Industria 4.0 en el Grado de Ingeniería Electrónica y Automática. In *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática*(pp. 327-332). Universitat Oberta La Salle
- [56] León Mendoza, D. A., & Castro Chávez. (2018) Diseño e implementación de un módulo de prácticas de domótica aplicando electrónica a bajo costo para el laboratorio de electrónica del colegio técnico Febres Cordero.
- [57] Temperature and Humidity module, DHT11 Product manual, Aosong (Guangzhou) Electronics Co.Ltd.
- [58] Zito, M. (2018). La sustentabilidad de Internet de las Cosas. *Cuadernos del Centro de Estudios en Diseño y Comunicación. Ensayos*, (70), 1-3.
- [59] Geraldo, A. P., & Paniza, G. M. (2014). ERGONOMÍA AMBIENTAL: Iluminación y confort térmico en trabajadores de oficinas con pantalla de visualización de datos. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, *1*(2).
- [60] Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, *20*(2), 1-12.
- [61] Loureiro Garrido. (2015) R. Estudio Plataformas IoT.
- [62] M. V. C. Caya, A. P. Babila, A. M. M. Bais, S. J. V. Im and R. Maramba, "Air pollution and particulate matter detector using raspberry Pi with IoT based notification," , Manila, 2017, pp. 1-4.
- [63] R. Saraf and R. G. Bhavani, "Assessment of daylight performance of a commercial office space in hot, arid climate for enhanced visual comfort conditions," Kollam, India, 2017, pp. 1-6.
- [64] Y. Obuchi, T. Yamasaki, K. Aizawa, S. Toriumi and M. Hayashi, "Measurement and evaluation of comfort levels of apartments using IoT sensors," *2018 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2018, pp. 1-6.
- [65] Harrington, W. (2015). *Learning Raspbian*. Packt Publishing Ltd.

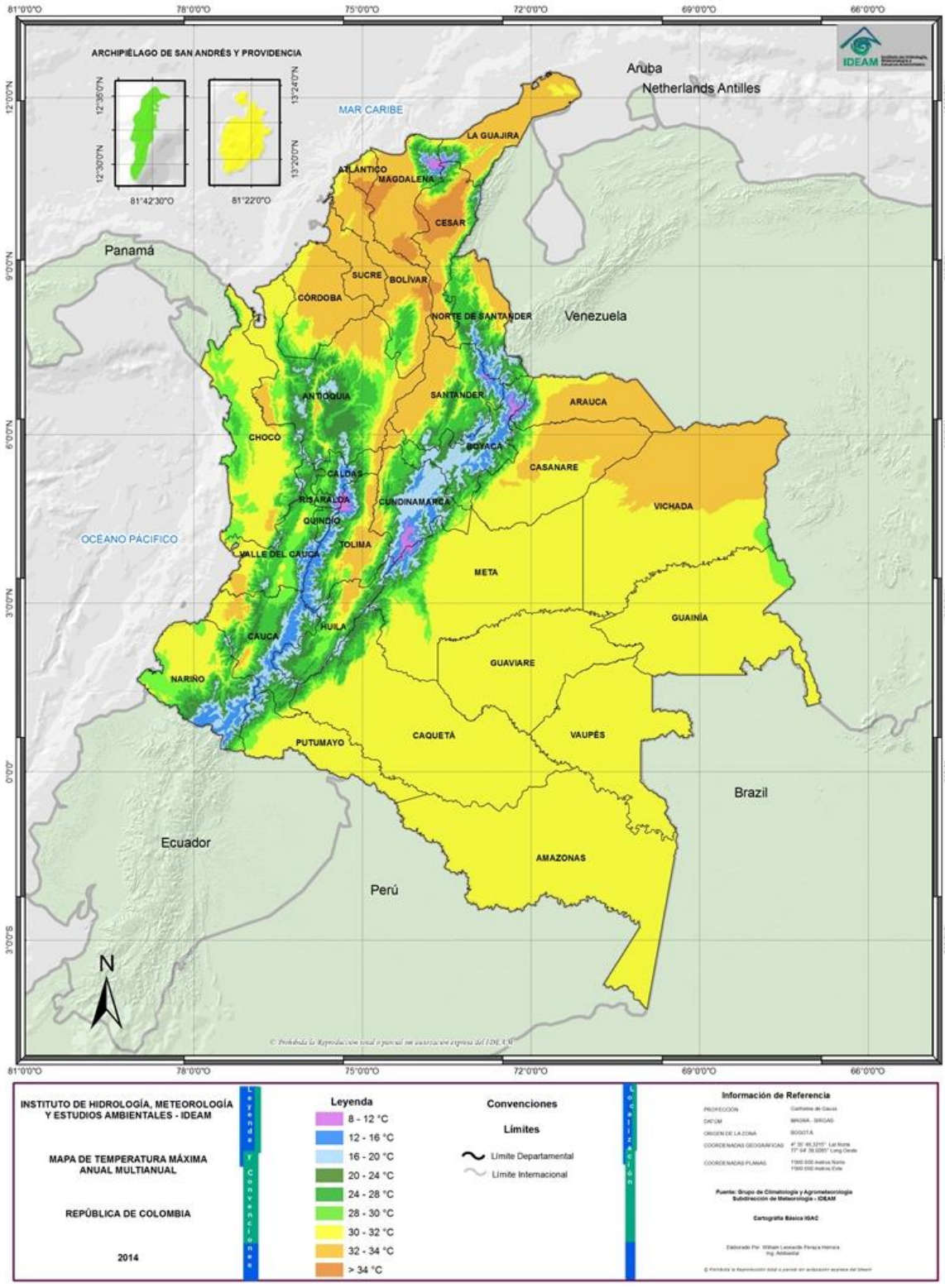
- [66] Arias Chaves, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, *6*(10).

ANEXOS

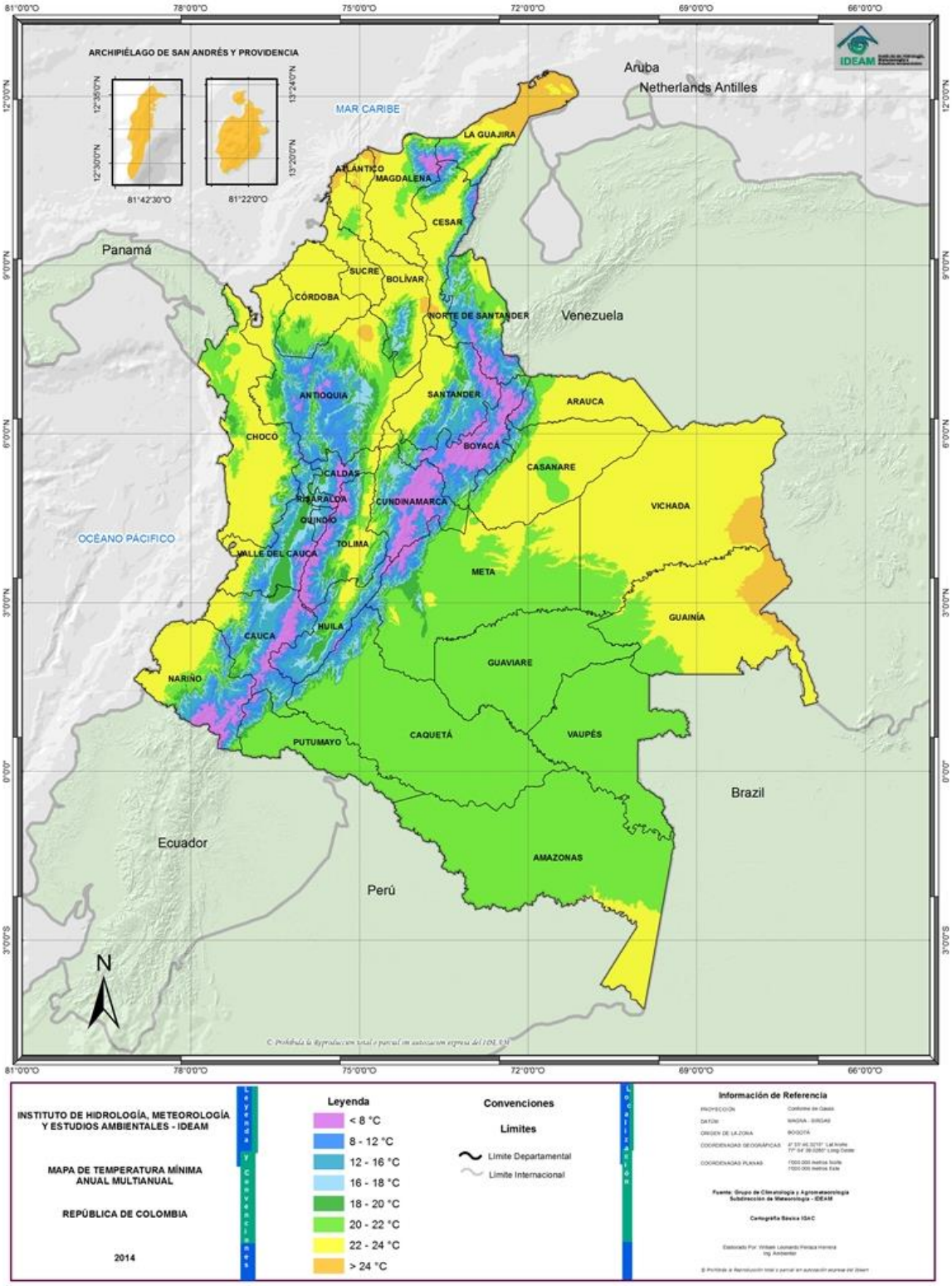
Anexo A Mapa Climático de Humedad Relativa en Colombia (1981-2010)



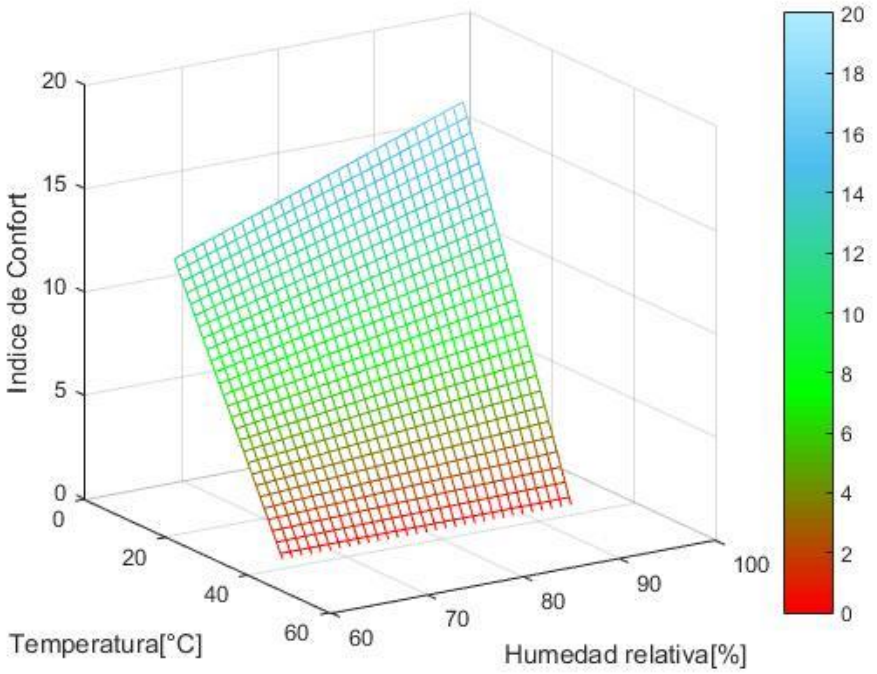
Anexo B. Mapa climático de temperatura máxima media de Colombia (2014)



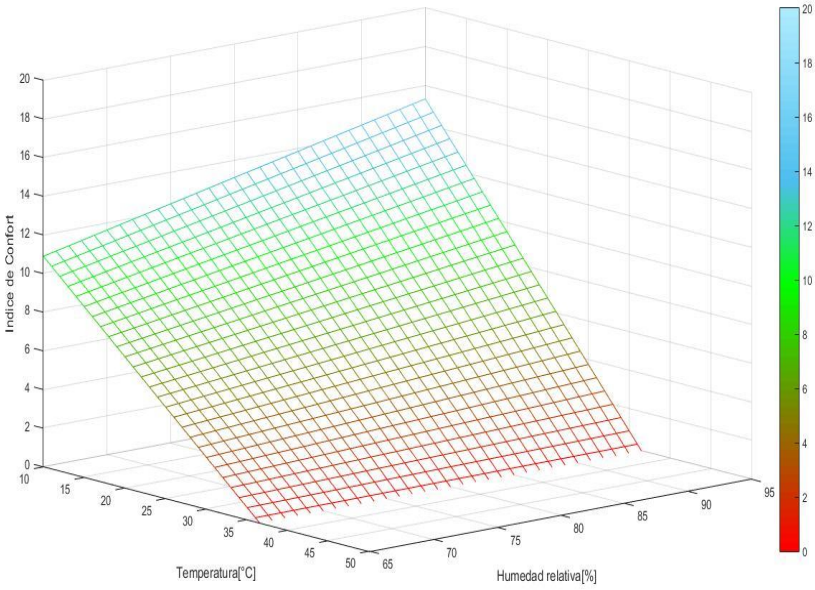
Anexo C. Mapa climático de temperatura media mínima de Colombia (2014)



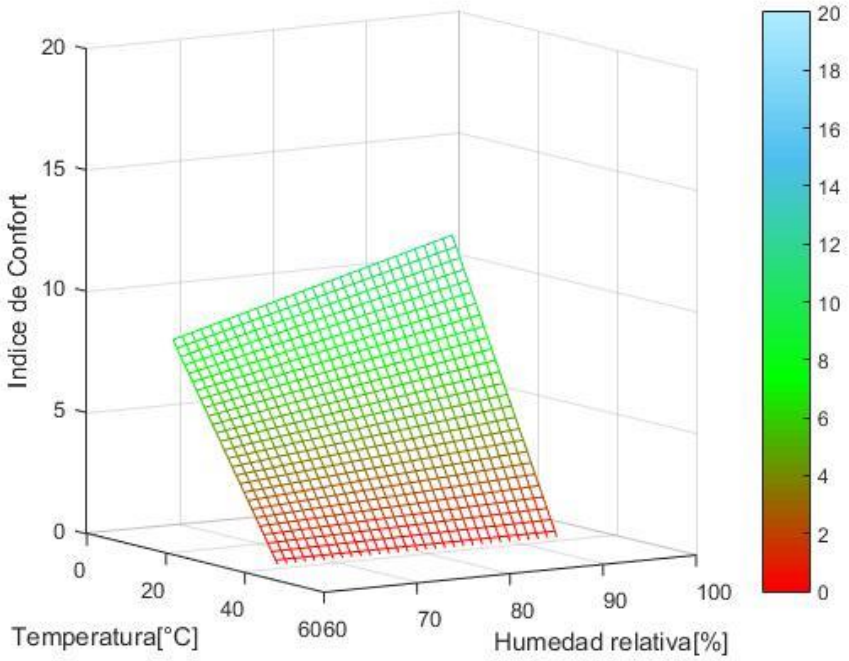
Anexo D. Graficas del Índice de confort con respecto a humedad relativa y temperatura del aire.



Para elevaciones superiores a 2000 metros



Para elevaciones de 1000 a 2000 metros



Para elevaciones Inferiores a 1000 metros

Anexo E. Dirección de página web de Thingspeak

https://thingspeak.com/

Anexo F. Dirección HTML del canal de Thingspeak

https://thingspeak.com/channels/528504

Anexo G. Código general del aplicativo

```
# -*- coding: utf-8 -*-
 #---------------------------Librerias----------------------------
import sys
import matplotlib
matplotlib.use("TkAgg")
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from Tkinter import *
import ttk
import tkMessageBox
from math import *
```

```
import math
import Tkinter as tk
import Adafruit_DHT as dht
import threading
import tkFont
import ImageTk
import RPi.GPIO as GPIO
import urllib2
import time
import matplotlib.pyplot as pyplot
import matplotlib.animation as animation
from matplotlib.figure import Figure
#-----------------------------------------------------------------
# Varible global para el API
myAPI = ""
#----------- Control de Errores para cerrar Programa----------
def seguroclosex():
        if tkMessageBox.askokcancel("Quit","Do you want to quit?"):
                root.quit()
#---------- Control de Errores  reconoce si es un----------
#------caracter y cumple con los pines digitales externos---------
#-------------------de la Raspberry------------------------------
def VerifPIN():
        try:
                integer_result = int(PIN.get())
        except:
                tkMessageBox.showerror("ERROR","ERROR NO ES UN NUMERO VALIDO ")
        else:
                if(PIN.get()>=2 and PIN.get()<=27):
                        tkMessageBox.showinfo("OK","Selected GPIO")
                else:
```

```
                                tkMessageBox.showerror("Error","GPIO Incorrect")
#----------Control de Errrores verifica si el APIKEY ------------
#----------------------es correcto-------------------------------
def VerifHTML():
        try:
                myAPI=  MYAPY.get()
                baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI
          f = urllib2.urlopen(baseURL + "&field1=0&field2=0")
        except IOError:
                tkMessageBox.showerror("ERROR","ERROR:HTML NOT CORRECT ")
        else:
                tkMessageBox.showinfo("OK","HTML Correct")
#---------------- Genera la ventana emergente de configurar------------
def configurar():
        global INIC
        global ALTITUD
        global PIN
        global MYAPY
        INIC.set(0)
        INCONFIG.set(1)
        ALTITUD.set(160)
        PIN.set(23)
        MYAPY.set("9R90SOJNATQI8H63")

        ventanaconfig= tk.Toplevel(root)
        ventanaconfig.title("Configuracion")
        ventanaconfig.geometry('600x300')
    ventanaconfig.focus_set()# provoca que la ventana tome el focus
    ventanaconfig.grab_set()# Deshabilita las otras ventanas hasta que esta sea destruida
    ventanaconfig.transient(master=root)#Aparece al frente de la ventana padre
```

```
        boton = Button(ventanaconfig,text="Aceptar",command=ventanaconfig.destroy,height
=2,width=6).place(x=270,y=250)

        boton1 = Button(ventanaconfig,text="Verificar PIN",command=VerifPIN,height
=2,width=10).place(x=385,y=80)

        boton2 = Button(ventanaconfig,text="Verificar HTML",command=VerifHTML,height
=2,width=10).place(x=385,y=185)



        a=tk.Radiobutton(ventanaconfig, text="Inferiores a 1000
metros",variable=ALTITUD,value=250).place(x=40,y=90)#grid(row=0,sticky=W)

        b=tk.Radiobutton(ventanaconfig, text="De 1000 metros a 2000
metros",variable=ALTITUD,value=180).place(x=40,y=130)#grid(row=0,sticky=W)

        c=tk.Radiobutton(ventanaconfig, text="Superiores a 2000
metros",variable=ALTITUD,value=160).place(x=40,y=170)#grid(row=0,sticky=W)



        titulo = Label(ventanaconfig, text ="Altitud de la Prueba
",font=("Helvetica",12,"bold")).place(x=40, y=65)

        titulo1 = Label(ventanaconfig, text ="Pin GPIO IN ",font=("Helvetica",12,"bold")).place(x=350,
y=30)

        titulo2 = Label(ventanaconfig, text ="APIKEY for
Thingspeak",font=("Helvetica",12,"bold")).place(x=350, y=135)




        GPIO= ttk.Entry(ventanaconfig,textvariable=PIN).place(x=350,y=50)

        MYAPYF= ttk.Entry(ventanaconfig,textvariable=MYAPY).place(x=350,y=155)


#-------Limpia y reinicializa las graficas de la interfaz de usuario principal ------------

def clea():

        saveFile=open('/home/pi/Desktop/processing_load_data/temperatura.txt','w')

        saveFile.write('\n')

        graphData = open('/home/pi/Desktop/processing_load_data/temperatura.txt','r').read()

        lines = graphData.split('\n')

        xValues = []

        yValues = []

        for line in lines:

                if len(line) > 1:
```

```
                        x, y = line.split(',')
                        xValues.append(x)
                        yValues.append(y)
        ax1.clear()
        ax2.clear()
        ax2.set_title('Temperatura')
        ax1.set_title('Humedad')
        ax2.set_xlabel('numero de muestras')
        ax2.set_ylabel('grados celsius')
        ax1.set_xlabel('numero de muestras')
        ax1.set_ylabel('porcentaje')
        ax1.plot(yValues)
        ax2.plot(xValues)
        canvas.show()
#------------------------Genera el evento de salir de la aplicacion----------------
def exit():
    root.quit()
#---------------------- Informa que es necesario inicializar-----------------------
#-----------------------------las variables por primera vez--------------------------
def Iniciar():
        global INIC
        global INCONFIG
        if INCONFIG.get()!=1:
                tkMessageBox.showinfo("Waiting for the first configuration","Please, configure the
parameters first in the button (Config)")
        else:
                INIC.set(1)
#----------------Para el programa general del aplicativo------------------
def Parar():
        global INIC
        INIC.set(0)
#--------------------Inicia el proceso general del aplicativo------------------
```

```
def readSensor():
        global INIC
        global CONFORT
        global ALTITUD
        global PIN
        global MYAPY
        if  INIC.get() == 1:
        h,t = dht.read_retry(dht.DHT11,PIN.get())
                if h is not None and t is not None:
                        temp = "%.1f" %t
                temperature.set(temp+" °C")
                hum = "%.1f" %h
                        myAPI= MYAPY.get()
                humidity.set(hum+"  %")
                        CONFORT.set((36.5-t)*(0.05+(h/ALTITUD.get())))
                        if CONFORT.get()>=0 and CONFORT.get()<=3:
                                INDICECONFORT.set("Muy Caluroso")
                        elif CONFORT.get()>=3.1 and CONFORT.get()<=5:
                        INDICECONFORT.set("Caluroso")
                        elif CONFORT.get()>=5.1 and CONFORT.get()<=7:
                        INDICECONFORT.set("Cálido")
                        elif CONFORT.get()>=7.1 and CONFORT.get()<=11:
                        INDICECONFORT.set("Agradable")
                        elif CONFORT.get()>=11.1 and CONFORT.get()<=13:
                        INDICECONFORT.set("Algo Frio")
                        elif CONFORT.get()>=13.1 and CONFORT.get()<=15:
                        INDICECONFORT.set("Frio")
                        elif CONFORT.get()>=15:
                        INDICECONFORT.set("Muy Frio")
                        else:
                        INDICECONFORT.set("Error")
```

```
baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI
        f = urllib2.urlopen(baseURL + "&field1=%s&field2=%s" % (h, t))
saveFile=open('/home/pi/Desktop/processing_load_data/temperatura.txt','a')
saveFile.write('\n')
saveFile.write(temp)
saveFile.write(',')
saveFile.write(hum)
saveFile.write('\n')
saveFile.close()
f.close()
graphData = open('/home/pi/Desktop/processing_load_data/temperatura.txt','r').read()
lines = graphData.split('\n')
xValues = []
        yValues = []
for line in lines:
        if len(line) > 1:
                x, y = line.split(',')
        xValues.append(x)
                yValues.append(y)
        ax1.clear()
        ax2.clear()
        ax2.set_title('Temperatura')
ax1.set_title('Humedad')
ax2.set_xlabel('numero de muestras')
ax2.set_ylabel('grados celsius')
ax1.set_xlabel('numero de muestras')
ax1.set_ylabel('porcentaje')
ax1.plot(yValues)
ax2.plot(xValues)
canvas.show()
```

```
                else:
                        tkMessageBox.showerror("Error Sensor","Sensor Not Conect")
                        INIC.set(0)

        root.after(2000,readSensor)

#------------------------Iniciamos la ventana principal--------------------------
root = tk.Tk()
#------------------------Variable Globales de la Interfaz grafica----------------------------
INIC = IntVar()
PORTADA= IntVar()
CONFORT= IntVar()
ALTITUD= IntVar()
INCONFIG= IntVar()
PIN=IntVar()
#-------------------------Inicializacion de parametros del GUI--------------------
root.geometry('1200x550')
image = PhotoImage(file="fondo2.gif")
background=Label(root, image=image)
background.place(x=0,y=0,relwidth=1, relheight=1)
fig = pyplot.figure(figsize=(2,4),dpi=50)
root.title('Proyecto de Grado')
#----------------- Creacion de sub Graficas para humedad y temperatura------------------
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(122)

canvas = FigureCanvasTkAgg(fig,root)
canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH)
```

```
temperature = StringVar()
temperature.set("------"+"°C")
humidity = StringVar()
humidity.set("------"+"%")
INDICECONFORT= StringVar()
INDICECONFORT.set("--------")
MYAPY = StringVar()


botonstart = Button(root,text="Start",command=Iniciar,height =2,width=6).place(x=180,y=180)
botonstop = Button(root,text="Stop",command=Parar,height =2,width=6).place(x=280,y=180)
botonclose = Button(root,text="Close",command=exit,height =2,width=6).place(x=810,y=180)
botonclear = Button(root,text="Clear Data",command=clea,height =2,width=8).place(x=500,y=270)
botonconfig = Button(root,text="Config",command=configurar,height =2,width=6).place(x=710,y=180)

temperatureLabel = Label(root, fg="white", background="#00dbde", textvariable=temperature,
font=("Helvetica", 40,"bold"))
temperatureLabel.place(x=680, y=260)
humidityLabel = Label(root, fg="white", background="#00dbde", textvariable=humidity, font=("Helvetica",
40,"bold"))
humidityLabel.place(x=180, y=260)
confortLabel = Label(root, fg="white", background="#00dbde", textvariable=INDICECONFORT,
font=("Helvetica", 35,"bold"))
confortLabel.place(x=450, y=160)

titulo = Label(root, text ="CONFORT TERMICO
",font=("Helvetica",30,"bold"),background="#00dbde").place(x=295, y=85)

#------ Crea una ventana emergente temporal----------
if PORTADA.get()!=1:
        windowport = tk.Toplevel(root)
        windowport.title("Proyecto de Grado")
    windowport.geometry('708x449')
```

```
        Escudo = PhotoImage(file="506812.png")
        background1 = tk.Label(windowport, image=Escudo)
        background1.place(x=0,y=0,relwidth=1, relheight=1)
        windowport.focus_set()# provoca que la ventana tome el focus
        windowport.grab_set()# Deshabilita las otras ventanas hasta que esta sea destruida
        windowport.transient(master=root)#Aparece al frente de la ventana padre
        titulo = tk.Label(windowport, text ="DISEÑO DE UN SISTEMA DE ADQUISICION Y
VISUALIZACION REMOTA DE DATOS ",font=("Helvetica",10,"bold"),background="#ffffff").place(x=80,
y=20)
        titulo2 = tk.Label(windowport, text =" BASADO EN LA PLATAFORMA DE SISTEMAS
EMBEBIDOS RASPBERRY PI",font=("Helvetica",10,"bold"),background="#ffffff").place(x=100, y=40)
        titulointegrante= tk.Label(windowport, text="JEYSSON EMILIO LIMA GUAQUETA Y PEDRO
LEONARDO OSPINA FUENTES",font=("Helvetica",10,"bold"),background="#ffffff").place(x=90, y=360)
        windowport.after(5000,windowport.destroy)
        windowport.protocol("WM_DELETE_WINDOW",seguroclosex)

        PORTADA.set(1)
#.....Genera eventos principales, cierre, retorna a
#.....readsensor luego vuelve y retorna a mainloop.....
root.protocol("WM_DELETE_WINDOW",seguroclosex)
root.after(2000,readSensor)
root.mainloop()o ot.mainloop()
```