

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN FIEC**

**AVANCE DE PROYECTO DE SISTEMAS EMBEBIDOS**

**Home automation system based on Cyber-Physical system**

**Alexander Martínez**

**Tomas Guijo**

**PAR. 107**

**Introducción:**

Hoy en día debido al gran avance de la tecnología, los seres humanos buscamos automatizar procesos tanto en la industria como en nuestro hogar por consiguiente el siguiente proyecto trata de facilitar las acciones diarias que se realizan en el domicilio de manera automática. La herramienta que permite tener el control automático de las funciones de los elementos y electrodomésticos del hogar es la domótica doméstica. Cabe recalcar que el uso de las mismas, funciona gracias a que todos los objetos deben estar conectados a internet, de ahí también se relaciona el concepto de IoT (Internet of Things).

Por otra parte, los dispositivos Cyber Físicos, hace referencia a los sistemas que integran redes de capacidad de computación, de comunicación, inteligencia y autonomía integrada con procesos físicos. En pocas palabras, se hace notar a aparatos físicos capaces de ser autosuficientes enlazado a la red capaz de controlar todo lo que ocurre en el mundo real. Como punto importante, desempeñan un papel fundamental actualmente en aplicaciones, transportes, salud, energía hasta poder generar ciudades ciudades inteligentes. Realizando una mención importante para dimensionar el proyecto y aminorar costos se plantea usar tarjetas Rasberry PI y Arduino.

Como dato importante, se estimó que los dispositivos finales inmersos en IoT crecieron hasta 29,5 millones en 2020 en relación a 2014. Esto se debe a que el IoT, se está notando un crecimiento exponencial con respecto al desarrollo de internet, y los atributos clave son los siguientes: sensorizacion con relación al uso de sensores, eficiencia, en red haciendo énfasis a los equipos que se deben conectar a la nube, especialización para determinado tipo de funciones y ubicuidad que se puede colocar en cualquier tipo de partes.

La tecnología avanza día a día de una manera abrumadora, y con ella un sinnúmero de cosas que la acompañan, cada vez es más frecuente el tener una mayor cantidad de componentes y dispositivos en nuestro hogar que nos ayuden a cumplir con un cometido en específico (Huidobro, 2019), y es por esta misma razón es que el poder controlar todos y cada uno de ellos puede llegar a ser una tarea realmente ardua si se lo quisiese aplicar de manera manual (Ponce, 2020). Controlar y gestionar cada uno de ellos prácticamente a cada momento podría no ser la mejor manera de resolverlo.

Por ello surge la necesidad de poder gestionar cada componente de una manera mucho más óptima (Satish Palaniappan, 2015), y es precisamente esto lo que conlleva a utilizar los sistemas ciber físicos como herramienta crucial para crear sistemas automatizados que se encarguen de optimizar enormemente este cometido (Dr Syeda Gauhar Fatima, 2019), buscando siempre la manera más optima y eficaz, con el fin de llegar a los mejores resultados usando la menor cantidad de recursos posible.

Además, se prevé que para el 2025 las cifras de dispositivos que se conecten a la IoT habrán aumentado en una medida nunca vista (Bochman, 2017), es claro que esto representa diversas ventajas para el hogar, puesto que conlleva diferentes facilidades de uso, gestión, etc. (Carlos Cortés, 2014), pero asimismo no podemos ignorar el hecho de que con esto viene una gran cantidad de desventajas también, que tendrán que ver con la seguridad de al red, temor a la vigilancia y las preocupaciones que con lleva (Karen Rose, 2015), es por esto que se busca dar una solución robusta y eficiente con sistemas que no solo controlen y gestionen el uso de cada componente, si no también monitoricen el estado de cada uno y le den al usuario la seguridad y privacidad que debe tener.

**Trabajos Relacionados:**

Una de las características principales al momento de realizar el análisis y ejecución del proyecto es que las investigaciones únicamente se han limitado a realizar control básico con supervisión de autómatas programables en pocas palabras (PLCs) y una de las mayores limitantes debido a su diseño que no son capaces de administrar y solucionar problemas con respecto a la incertidumbre. Sin embargo al momento de fusionarlas con la utilización de redes FBs para los procesos que se desean ejecutar, surge como una alternativa viable debido a que pueden ser activados para realizar algoritmos internos de manera controlada, lo que facilitara la generación y cambio de planes de procesos para los rendimientos óptimos del sistema.

También realizando las verificaciones correspondientes uno de los hitos principales previamente para realizar dicho trabajo es definir las características y especificaciones generales para los CPS en tareas para monitorear con respecto a técnicas de computación. La eficacia de la estructura del sistema dependerá del despliegue y del desempeño de las funciones de análisis de información que se ingrese. El conocimiento del sistema puede incluir posibles patrones de trabajo, estado de máquina, fallos y actividades específicas; sin embargo es fundamental la programación de CPs debido a que tienen una alta prioridad con respecto a los modelos tradicionales, los que lo hacen superiores y novedosos que el resto.

Adicionalmente, el concepto de Sistemas Cyber Fisicos constituye una nueva disciplina de la ingeniería que exige sus propios modelos y métodos correspondientes; debido a ello el diseño de los mismos constituyen en subsistemas distribuidos a través de la red que utilizan modelos matemáticos; como dato importante un modelo completo se desarrolla con la inserción de su entorno a las acciones que desea ejecutar y al cálculo de embebidos, lo que indica que pueden ser probados de manera simulada lo que permite verificar el funcionamiento del mismo sin necesidad de realizar ningún tipo de gasto y validar las características de su comportamiento.

**Metodología experimental (End Device):**

**Objetivos:**

-Diseñar sistemas de automatización a través de las herramientas proporcionadas en el presente curso para ser aplicados en el ámbito de la domótica.

-Controlar diversos tipos de actuadores mediante sensores y sistemas embebidos que permitan un rendimiento óptimo para el sistema electrónico automático de una casa.

-Realizar el desarrollo de sistemas de monitoreo que permitan observar los estados en los que se encuentran los actuadores en la vivienda, brindando información acerca de los cambios de acuerdo a fecha y hora, que se han ido dando en todo el sistema.

**Materiales:**

Arduino uno ATmega 328p

Raspberry PI 3

**Actuadores:**

* Televisores (el principal y el que seria para el dormitorio).
* Aires acondicionados (el que estaría en la sala, y el del dormitorio).
* Alarmas
* Bomba de agua
* Luces Smartlight, tanto para el cuarto, sala y un garaje.

**Sensores:**

* DHT11-Sensor de temperatura y humedad
* TMP36-Sensor de temperatura
* LDR-Fotoresistor
* MK0434-Sensor de proximidad IR

Dado que el fin es implementar un sistema domótico basado en automatización por sistemas ciber físicos se ha pensado en emplear diferentes placas electrónicas conocidas para el cometido, con el Arduino uno Mega 328p configuraremos el respectivo control de cada actuador como se requiera, dependiendo de los datos recolectados de los sensores que serán utilizados respectivamente para cada actuador.

Tenemos el control del rociador contra incendios proporcionado por los datos que recolecte el sensor de temperatura y humedad DHT11, para dar seguridad al hogar ante cualquier catástrofe de este tipo, también se pretende lidiar con el control de la temperatura en el hogar, para lo cual se utilizara el sensor de temperatura TMP36, el cual obtendrá los datos que serán utilizados para aplicar el control respectivo mediante el código configurado en Arduino, que dependiendo de la temperatura hará que encienda el aire acondicionado o el calefactor, además de la caldera para temperar el agua caliente.

Para la red de iluminación de la casa se cuenta con luces smartlights, que serán controladas mediante el módulo de gestión (el Arduino) dependiendo de la hora que sea, por último, tenemos el motor de la puerta de garaje, que será el actuador accionado con ayuda del sensor de proximidad IR MK0434, puesto que cuando detecte al vehículo cerca se aplicaría la señal que hará que comience a iniciar el movimiento del motor.

En cuanto al software de los end device se utilizará la unidad de administración con ayuda de la placa de Raspberry PI para crear un menú que controle de manera eficaz la placa de Arduino que será el módulo que controlara todo el sistema.

En primera instancia se encuentran los sensores, puestos en sitios estratégicos para su correcto funcionamiento y estarán conectados a los puertos de entrada analógicos del Arduino, se utilizarán los puertos PD0 hasta PD3 para el ingreso de estos datos, por lo que serán configurados como entradas. Los puertos PB0 hasta PB5 serán configurados como salidas para posteriormente conectarse a los respectivos actuadores.

**Metodología experimental:**

Al momento del planteamiento del diagrama esquemático en Proteus se ha definido diferentes dispositivos que se podrán automatizar para que se realice el sistema domótico dentro de cualquier domicilio que se pueda implementar, en este caso se comenzó planteando un calefactor, aire acondicionado, luces Smart y un garaje. Debido a ello en primera instancia, hemos definido el uso del integrado de un ATMEGA328P para la conexión de los diferentes dispositivos como se puede observar en la ilustración 1. Por ello, se ha representado los mismos con un grupo de resistencias y leds para verificar la funcionabilidad de cada uno de ellos.

Conforme se fue realizando la implementación correspondiente, y añadiendo tanto la parte de MITAI2, Ubidots, Arduino y Raspbian se fue modificando el esquema quedando de la manera final como la ilustración 2; donde se agregó la conexión para Arduino y también con Raspberry para realizar la conexión serial con los puertos del dispositivo. En primer lugar, se comentara acerca de la conexión con respecto al arduino, al inicio se debe configurar las entradas correspondientes dentro del arduino con conexiones analógicas y digitales y también verificar en que Baud Rate se encuentra configurado los dispositivos, ya que si se encuentran diferentes nunca se ejecutara la conexión y no se podrá visualizar lo deseado en Node-Red.

Una vez establecida la validación de los parámetros y conexiones correspondientes, se procede a la conexión del Raspberry, pero antes que nada se tiene que instalar node-red dentro del raspberry para poder la visualización de los datos que se desean ejecutar. (Dicha configuración se encuentra dentro de los archivos adjuntos del documento). Por otra parte, el dispositivo Arduino también se debe programarlo en la parte de source code, se debe iniciarlo para poder ejecutar las funciones y que el virtual terminal muestre lo que se desea ejecutar. (Codigo indicado en los archivos adjuntos del documento).

Posterior a ello, se configura dentro del aplicativo de MIT para brindarle la funcionalidad deseada por medio de algún tipo de aplicación y poder manejar los dispositivos deseas a través de ella, esto quiere decir que por parte de los botones programados dentro del aplicativo se puede controlar el funcionamiento de nuestro hogar, específicamente de los dispositivos inteligentes que se ha usado. Adicionalmente también se ejecuta una interfaz gráfica.

Por último, se ejecuta la conexión con Ubidots generando la conexión de los puertos seriales, mediante el virtual serial port; se valida que no se encuentra bajo el mismo puerto trabajando, cabe recalcar que los puertos siempre van en pares por ello se debe tomar en cuenta que puertos se usaran para la conexión correspondiente, el código del arduino también es importante para que los gráficos que se deseen mostrar salgan de la manera correcta. En la ilustración 4 se determina la conexión de Ubidots con un teléfono móvil específico y las ilustraciones 5 y 6 se inicializa las variables que se desean mostrar y adicionalmente los cambios que surgen dentro del registro cuando van entrando valores al sistemas, de manera digital 1 cuando se encuentra encendido y 0 para cuando este apagado.

**Metodología experimental (Gateway):**

En cuanto al Gateway, podemos definir en nuestro sistema a la tarjeta raspbian como responsable de cumplir con el cometido, puesto que será el puente intermedio que procesara los datos que vendrán desde el mit ai2, que es la app que básicamente se encargaría de la función de controlador para las acciones de encendido y apagado de las luminarias inteligentes, y dichas ordenes iran al proteus pasando por el código en raspbian como interceptor de datos.

En cuanto a ubidots, utilizamos también la raspberry como medio de transmisión y validación de datos desde el proteus, puesto que, además de brindar seguridad a los mismos a través de encriptación en claves dentro de las señales activadoras para cada cometido dentro del código, cumple el propósito de enviar dicha información a la nube de ubidots, para poder monitorear el estado, cambios, y en si un historial que nos permite visualizar la fecha y hora de los cambios que se efectuaron en cuanto a las luminarias inteligentes, que en nuestro sistema están señaladas por las etiquetas correspondientes al patio, cuarto y garaje.

Además de eso tenemos también el envio de información por parte de la parte izquierda del sistema, la que se conforma por los elementos que vendrían a ser los electrodomésticos, los cimatizadores, la bomba de agua y la alarma, los cuales enviarían dicha información a través del serial COM20 (conectado al COM21), pasando a través de la raspberry, siendo interceptados e interpretados por el código de nodered, y a continuación enviados al sistema en el sitio web de node-red, el cual corresponde a la pagina de <http://localhost:1880>.

En caso de abrirlo desde una pc ajena a la raspberry pero que estuviese conectado a ella por SSH, la dirección cambiaria a la que pertenecería a la ip de raspberry, en nuestro caso esto es: <http://192.168.1.16:1880>, una vez enviados los datos serán mostrados a través de la interfaz grafica que otorga node-red a través del dashboard, facilitando en gran manera el monitoreo y revisión de los datos que corresponden a los actuadores mencionados con anterioridad.

Y así es como tendríamos un control, activación a distancia, automatizada, control y monitoreo en línea de los diferentes dispositivos configurados y agregados al sistema. Los sistemas utilizados de ubidots y node-red fueron especialmente importantes para este cometido, pues gracias a estos podríamos llevar una sencilla y optima gestión de dicha información incluso desde fuera de nuestra casa, tan solo teniendo que conectar nuestra laptop o celular a la web mencionada.

**Resultados Obtenidos:**

Para el Arduino atmega328p podemos tener en consideración varias cosas, en primer lugar, trabajara todo el tiempo, sensando y obteniendo los datos provenientes de otros sensores, la tarjeta raspberry, etc., pero aun así, dado que se trata de un hardware de bajo consumo, la cantidad de mW consumidos rondan los 100mW cuando se trabaja con el reloj a una frecuencia de 20Mhz, pero puesto que en nuestro sistema este valor esta configurado a 8 Mhz (podría ser incluso menos, dado que la aplicación utilizada lo permite) el consumo rondara un aproximado de 30 mW, dado que además tiene habilitados los puertos seriales, lo que aumenta en una cantidad ínfima el consumo total. En cuanto al % de CPU utilizado en nuestro sistema, éste ronda un promedio de 55% de uso.

El atmega328p tiene como memoria de programa una memoria flash de 32 KB, y como memoria de datos una memoria EEPROM de 1 KB y una memoria SRAM de 2 KB, de las cuales, en este sistema es utilizado en promedio 72% de memoria de programa (la cual fue depende prácticamente de la cantidad que ocupa todo el código implementado en la memoria flash) y 36% de memoria EEPROM y 59% de memoria SRAM, lo que significaría un 71% de memoria de datos aproximadamente.

Por otra parte, la tarjeta raspberry utilizada (la cual, como se especifico corresponde al modelo raspberry PI 3) se configuro en modo idle como espera para cuando no se esta recibiendo datos, puesto que podría llegar a ser un periodo realmente extenso, y además esta tarjeta consumo, en general, mas recursos que el atmega328p. Considerando el sistema implementado, tenemos que el consumo de energía ronda los 96 wH (aproximadamente 346 kJ al dia).

En cuanto al cpu y la memoria, el primero tiene una utilización en promedio de 1.42 mHz, que representa el 30% del total para el cpu, mientras que la memoria total es de 1.92 GB, esta siendo utilizada 0.82 GB, que se traduce en el 43% del total. Como vemos, en comparativa el Arduino tiene los recursos mucho mas saturados en comparación, puesto que la tarjeta raspberry tiene libre tanto el % del cpu total, como memori, quedando libre una cantidad considerable en relación con la ocupada.

Por último tenemos los sensores utilizados, el DHT11 funciona bajo la alimentación de 5v establecida, y, puesto que el consumo de corriente es de 2.5 mA, el consumo total de potencia rondaría los 12.5 mW. El siguiente sensor, el TMP36 tendria también 5v de alimentación, ya que es la que se ha dado en todo el sistema (soporta hasta 5.5), y su consumo máximo de corriente es de 50 uA, que se traduce en un consumo de potencia de 250 uW, o sea 0.25 mW; como podemos observar es una cantidad mucho menor al anterior. El siguiente en la lista es el fotoresistor, del cual podemos observar su consumo de potencia en su datasheet, siendo este 100 mW. Finalmente tenemos el sensor de proximidad IR MK0434, asimismo con una alimentación de 5v, y un consumo promedio aproximado de corriente de 35 mA, tenemos un total de 175 mW, siendo el que más consume dentro de nuestra lista de los sensores utilizados para el sistema desarrollado.

**Conclusiones:**

* Luego de realizar un análisis minucioso de los datos estadísticos arrojados al medir las variables de consumo tanto de memoria, cpu y potencia en general de nuestros dispositivos (tanto hardware como los sensores en sí) podemos observar que estos valores son relativamente altos, comparados con otros sistemas mas simples donde, por ejemplo, el arduino trabaja con una frecuencia de reloj mucho más baja (pudiendo ser desde 32 khz aproximadamente) o directamente con menos código, y es eso justamente lo interesante de esto, que aunque tenemos los recursos como tal disponibles, la optimización de código y herramientas a utilizar para que sean tanto suficientes para ejecutar el resultado como económicas, en términos de coste de todas nuestras variables asociadas, nos permitiría mejorar la utilización de recursos en general que consume este sistema, o cualquier otro.
* La seguridad siempre será, por mucho, lo mas importante dentro de un sistema de este tipo, es por esto que no lo podemos dejar de lado, mas aun si se trata de un sistema de domotica, donde practicamente se tiene el control de tu propia casa. Así que las medidas de seguridad aplicadas deberán ser muy rigurosas, aun después de que la implementación y la optimización como tal estén funcionales en nuestro sistema, se debe desarrollar un sistema de cifrado o código que permita diferenciar y reconocer al usuario, para evitar posible suplantación de identidad o un ataque MitM, siendo este desarrollado principalmente entre la comunicación de los end devices con el Gateway, donde en el escenario del sistema mencionado se encontrarian las tarjetas Arduino y raspberry.

**Esquemático en Proteus:**

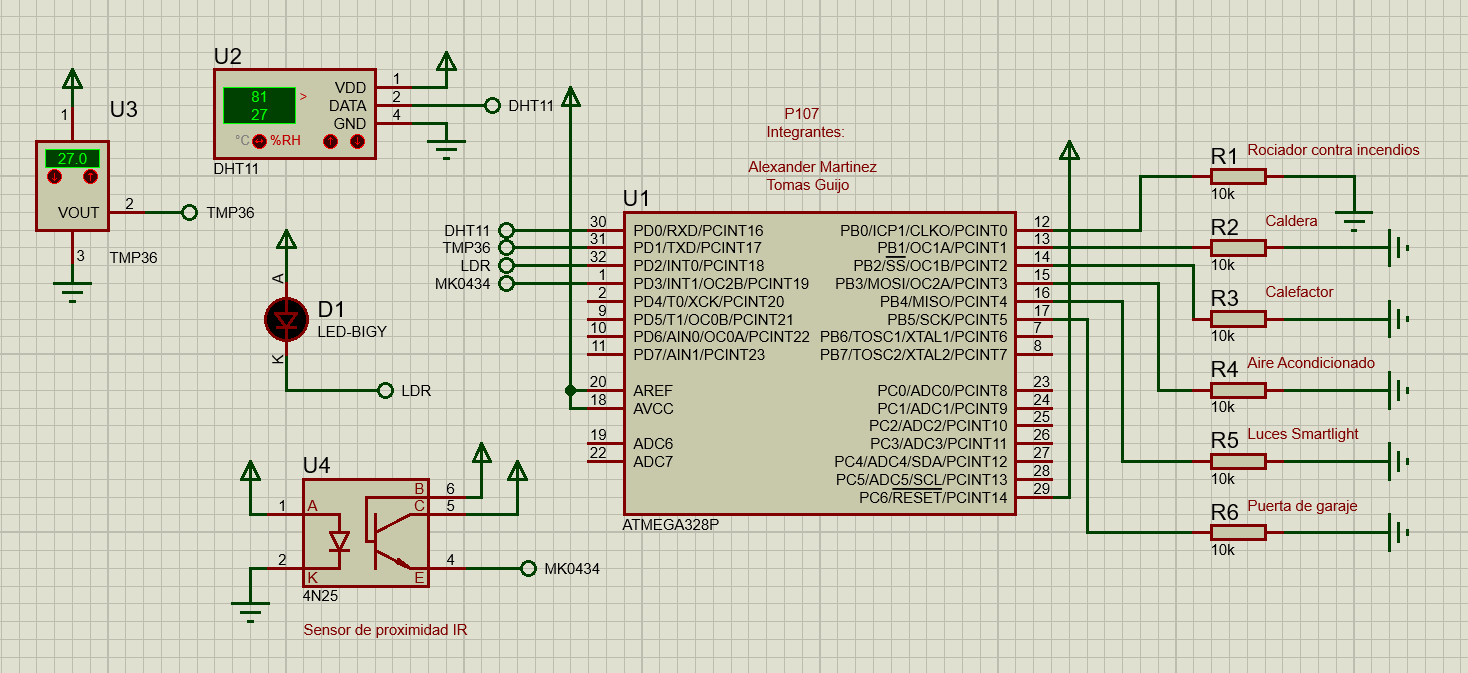


Ilustración 1. Esquemático inicial del proyecto

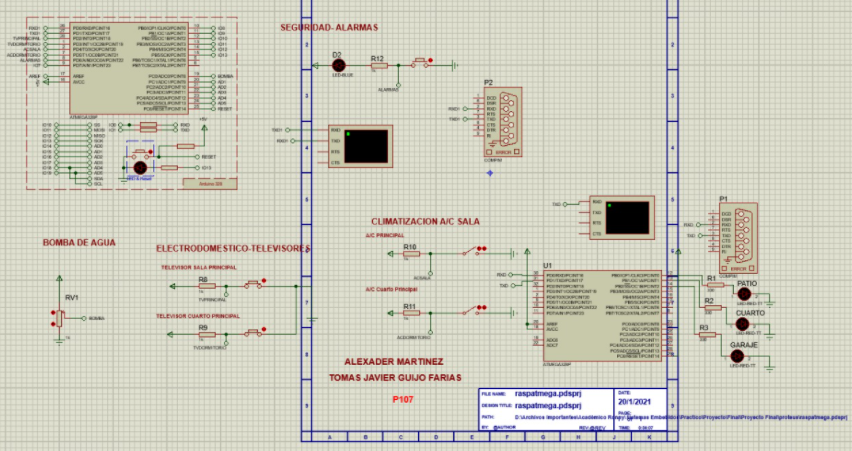


Ilustración 2. Esquemático final del proyecto

**Conexión con MIT:**

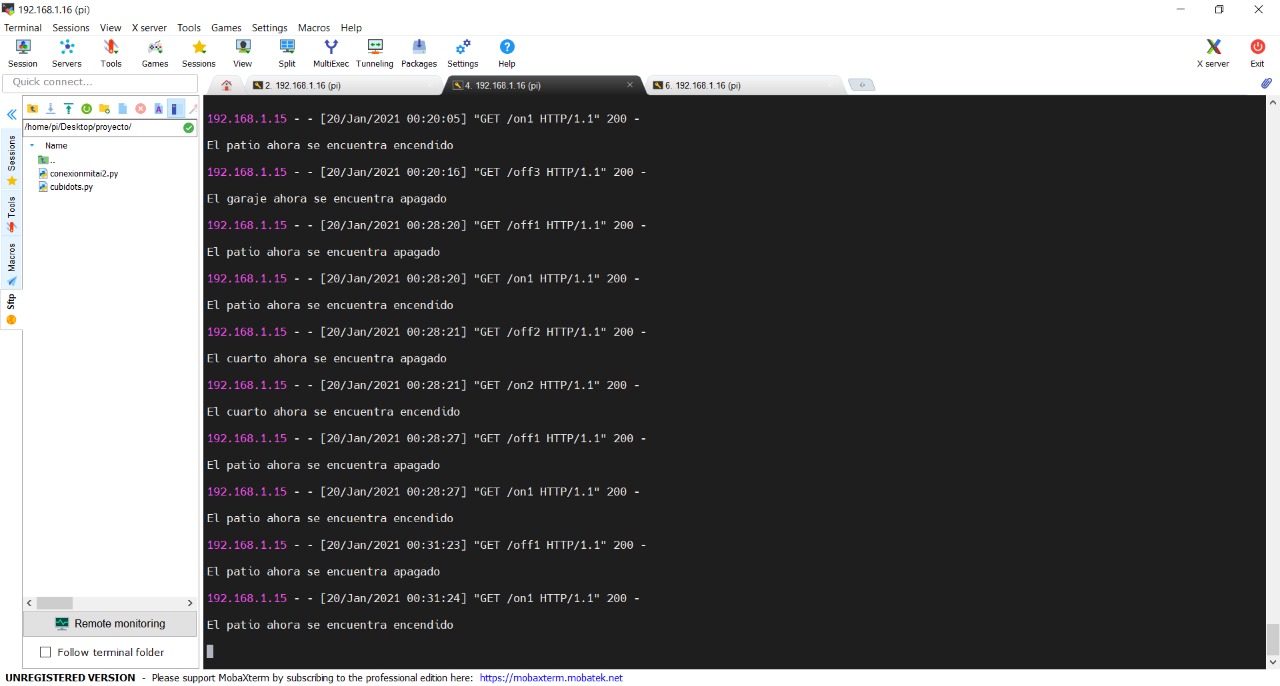
****

Ilustración 3. Conexión del aplicativo mediante MIT

**Conexión con Ubidots:**

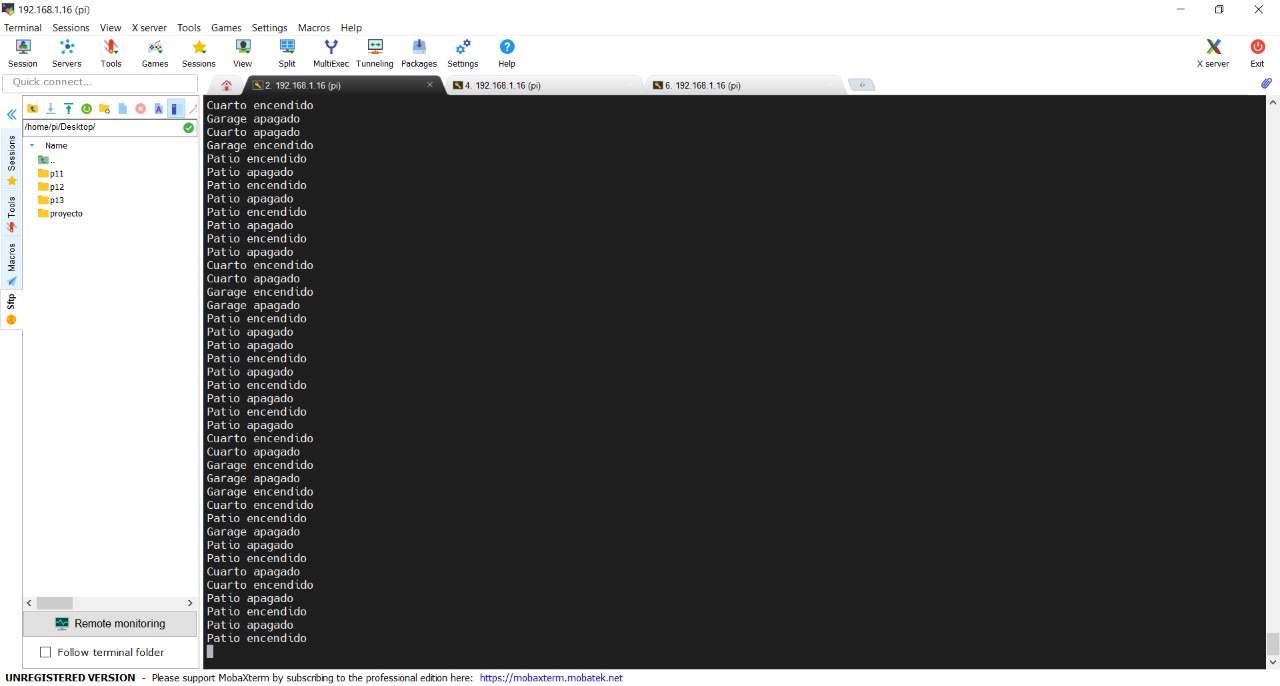
****

Ilustración 4. Conexión del aplicativo mediante el teléfono

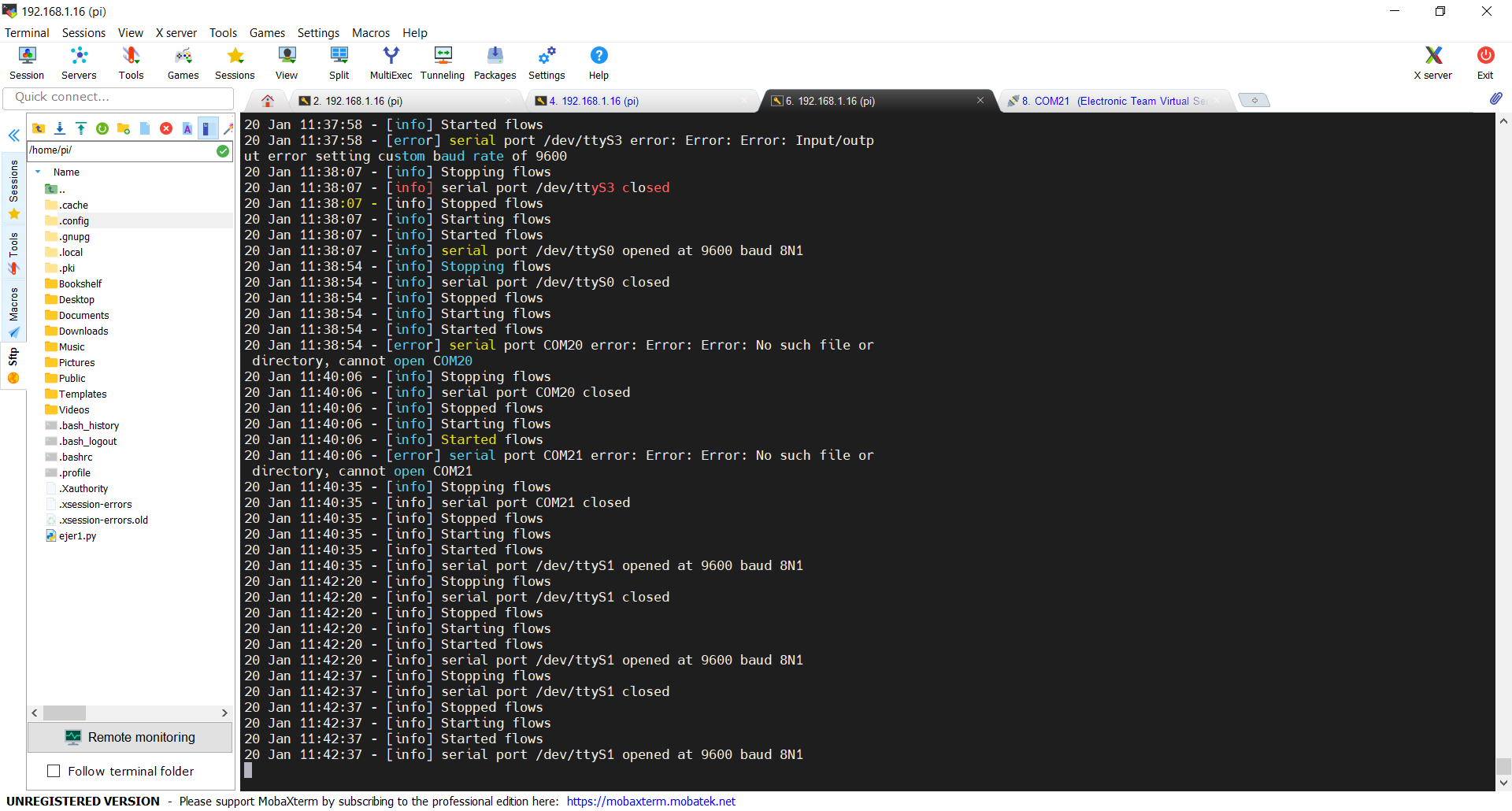


Ilustración 4. Conexión del aplicativo a node-red

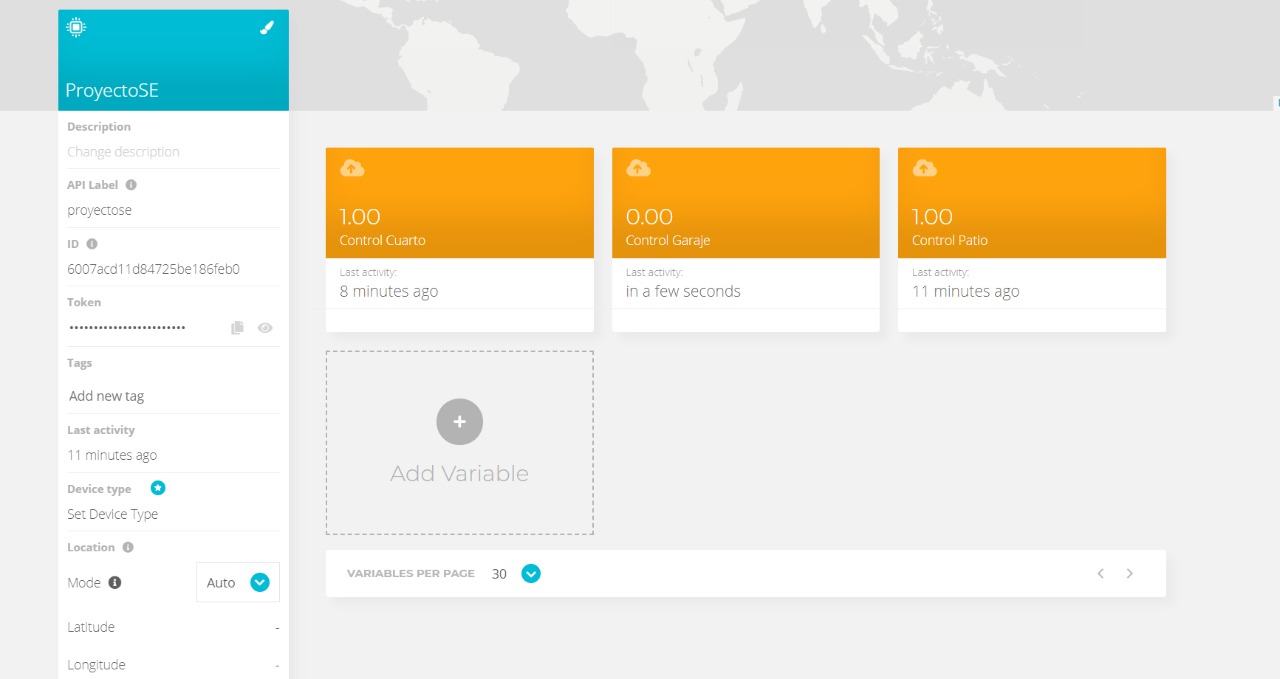


Ilustración 6. Variables estipuladas para el aplicativo de Ubidots

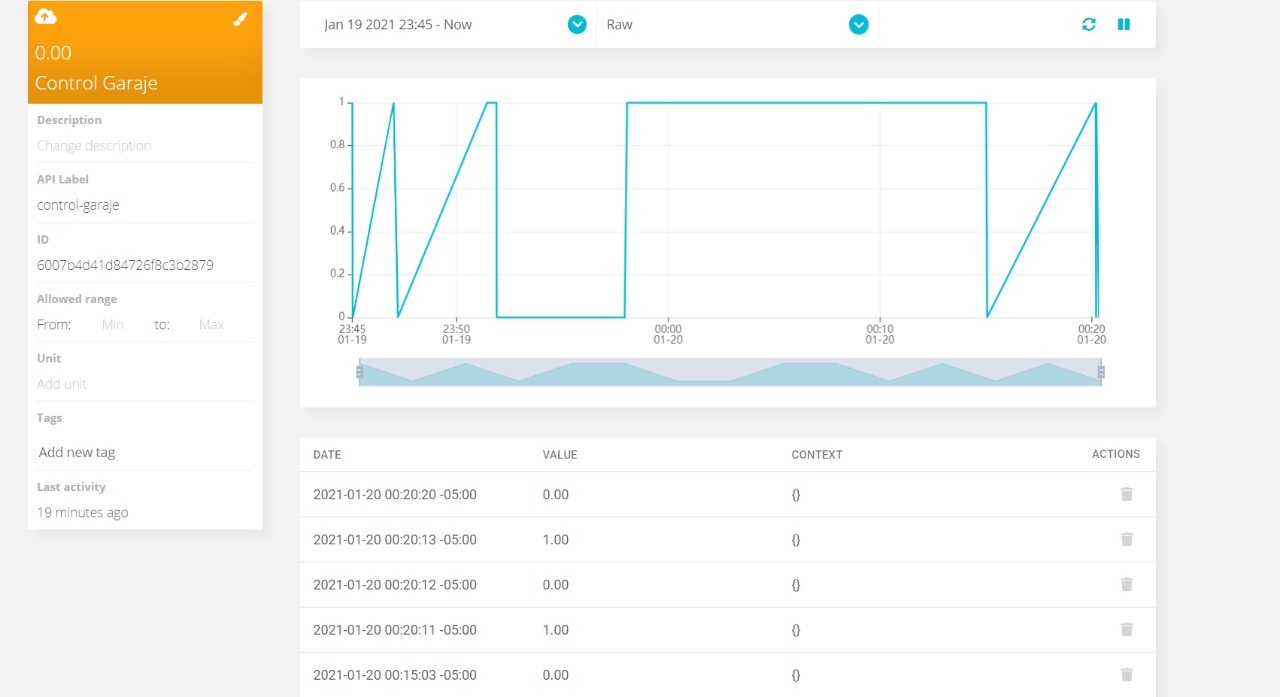


Ilustración 7. Variación de señales debido a la entrada de datos de las variables usadas

**Interfaz gráfica del aplicativo:**

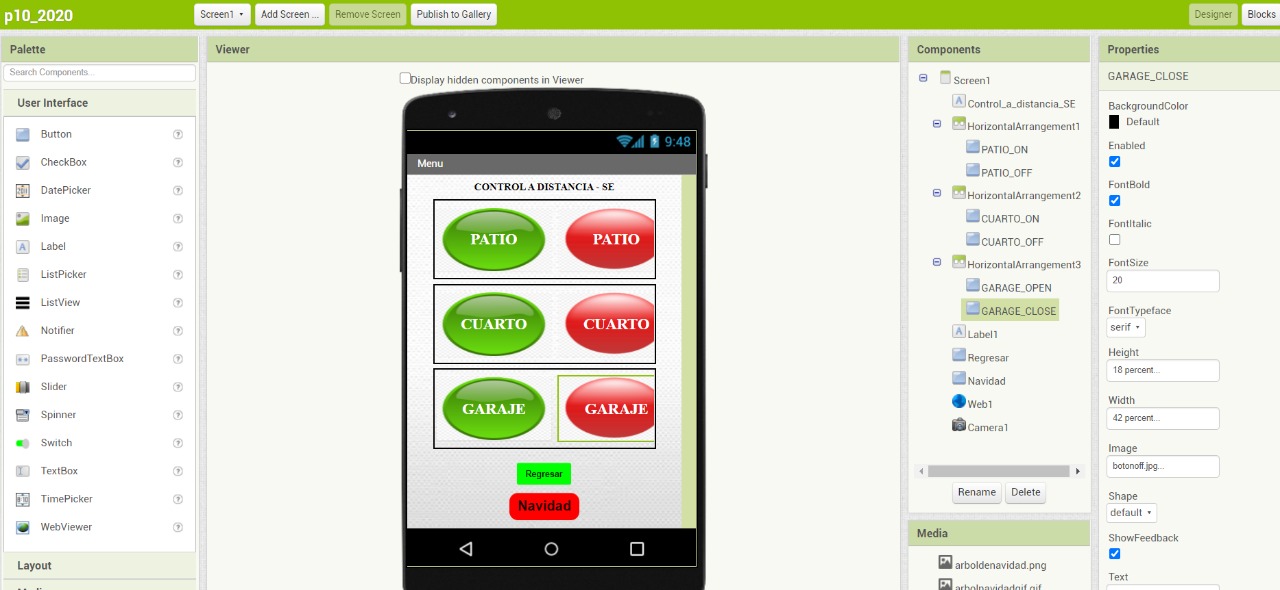
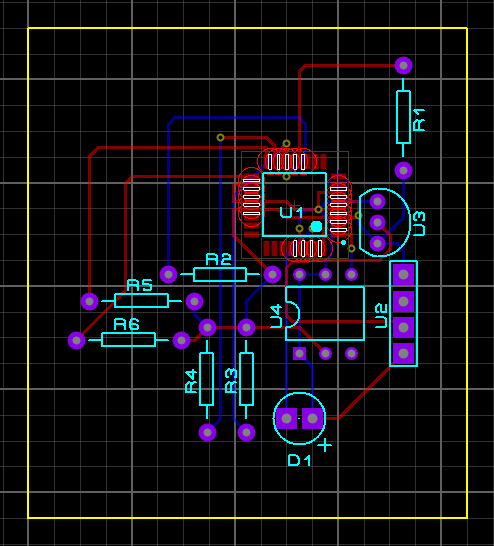
****

Ilustración 8. Interfaz del aplicativo

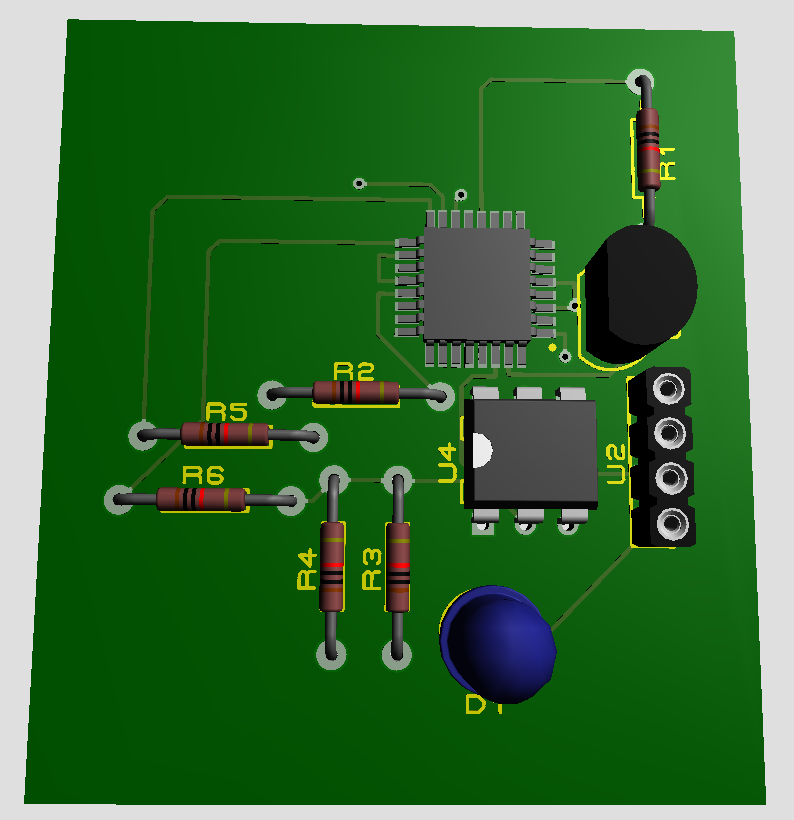


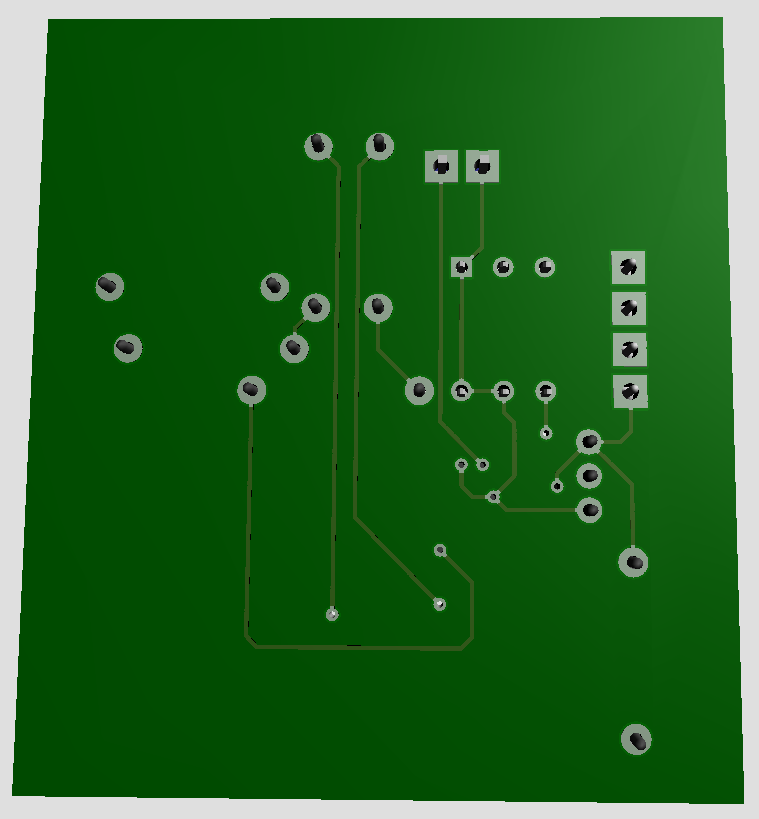
Ilustración 9. Interfaz real del aplicativo en un Oneplus 7 pro

**Diseño PCB:**

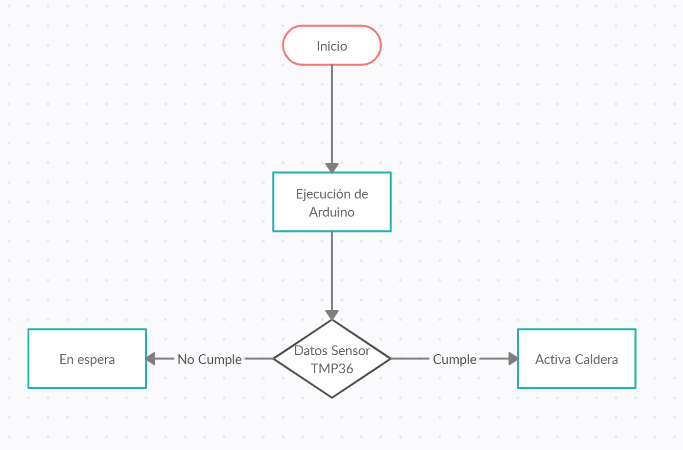
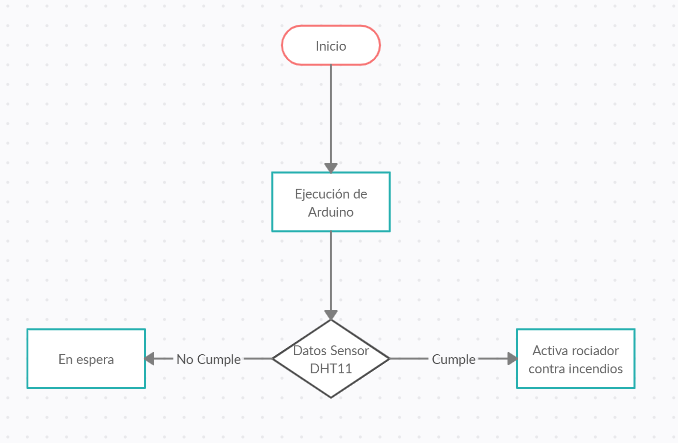


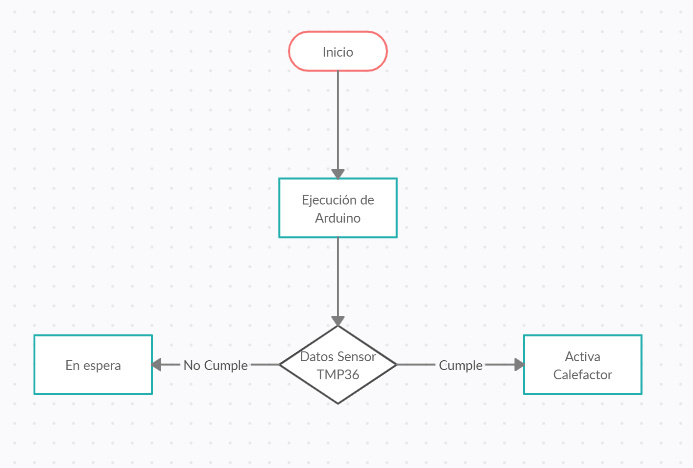
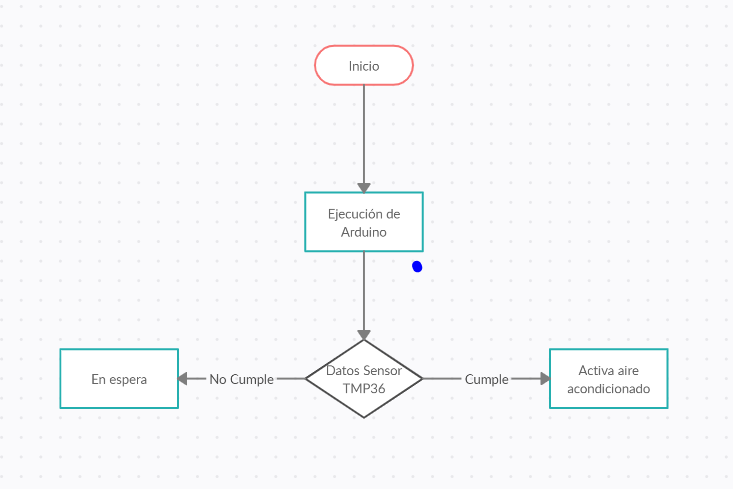
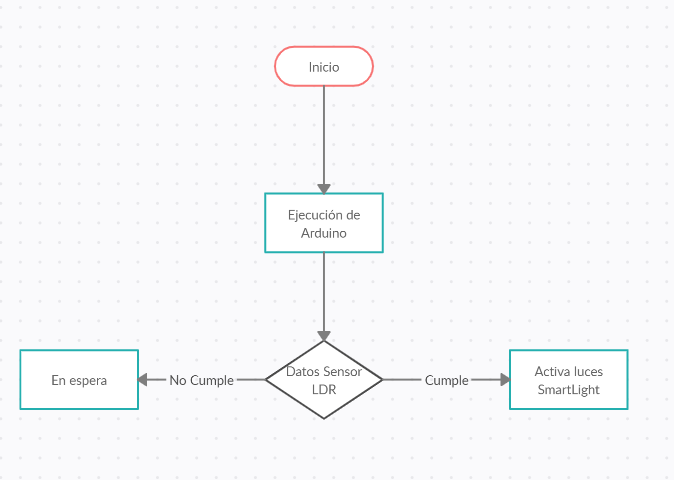
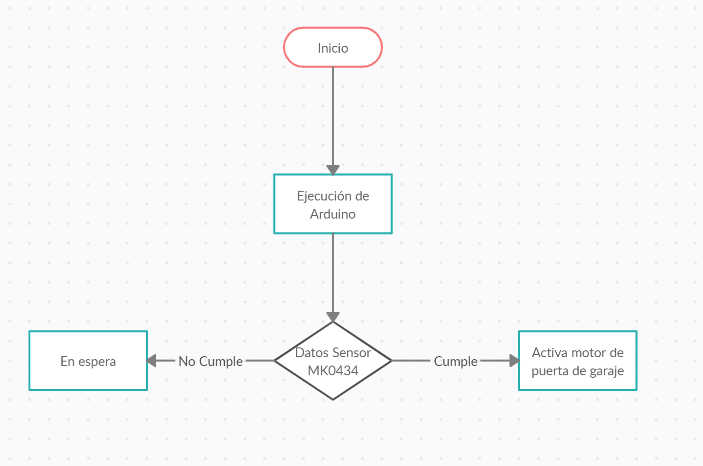
**Vista en 3D:**





**Diagramas de flujo:**



Comandos para la instalacion de Node-Red en Raspberry

**sudo apt-get update**

**sudo apt-get upgrade**

**sudo apt-get install git**

**node -v**

**npm -v**

**sudo apt-get install npm**

**sudo apt-get install curl**

**curl -s https://deb.nodesource.com/setup\_6.x**

**| sudo -E bash**

**sudo apt-get install -y nodejs build-essential**

**sudo npm install -g --unsafe -perm node-red**

**npm i node-red-node-serialport**

**node-red**

Codigo para el Arduino

const int sensorpinagua = A0; //pin sensor

float bomba;

const int tvprincipal= 2;

const int tvdormitorio= 3;

const int acsala= 4;

const int acdormitorio= 5;

const int alarmas= 6;

int estadotvprincipal= 0;

int estadotvdormitorio = 0;

int estadoacsala = 0;

int estadoacdormitorio = 0;

int estadoalarmas = 0;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

pinMode(tvprincipal, INPUT);

pinMode(tvdormitorio, INPUT);

pinMode(acsala, INPUT);

pinMode(acdormitorio, INPUT);

pinMode(alarmas, INPUT);

}

void loop() {

// put your main code here, to run repeatedly:

bomba= analogRead(A0);

bomba= map(bomba, 0, 1023, 0, 100);

//TV

estadotvprincipal= digitalRead(tvprincipal);

if( estadotvprincipal == HIGH){

Serial.println("TV PRINCIPAL ENCENDIDO");

}

else{

Serial.println("TV PRINCIPAL APAGADO");

}

estadotvdormitorio= digitalRead(tvdormitorio);

delay (1000);

if( estadotvdormitorio == HIGH){

Serial.println("TV DORMITORIO ENCENDIDO");

}

else{

Serial.println("TV DORMITORIO APAGADO");

}

// A/C

estadoacsala= digitalRead(acsala);

if( estadoacsala == HIGH ){

Serial.println("AIRE SALA ENCENDIDO");

}

else{

Serial.println("AIRE SALA APAGADO");

}

estadoacdormitorio= digitalRead(acdormitorio);

if( estadoacdormitorio== HIGH){

Serial.println("AIRE DORMITORIO ENCENDIDO");

}

else{

Serial.println("AIRE DORMITORIO APAGADO");

}

//ALARMAS

estadoalarmas = digitalRead(alarmas);

if( estadoalarmas == HIGH){

Serial.println("ALARMA ENCENDIDA");

}

else{

Serial.println("ALARMA APAGADA");

}

//IMPRIMIR POR SERIAL

Serial.print("CANTIDAD DE AGUA");

Serial.println(bomba);

delay(2000);

}

# Referencias

Bochman, M. A. (1 de abril de 2017). *Jstor*. Obtenido de IoT, Automation, Autonomy and Megacities in 2025: https://www.jstor.org/stable/resrep23272

Carlos Cortés, J. D. (2014). *El internet de las cosas, más internet que otra cosa.* Toluca.

Comunicación, G. (viernes 10 de enero de 2020). *Gelileo*. Obtenido de Internet de las cosas: https://www.galileo.edu/trends-innovation/internet-de-las-cosas-75-mil-millones-de-objetos-conectados-en-2025/

Dr Syeda Gauhar Fatima, S. K. (marzo de 2019). *AcademiaEU*. Obtenido de Home Automation System WSN IOT: https://www.academia.edu/38966518/HOME\_AUTOMATION\_SYSTEM\_WITH\_WSN\_AND\_IOT

Gandhi, M. (jueves 27 de febrero de 2020). *Autycom*. Obtenido de Sistemas Ciberfisicos: https://www.autycom.com/sistemas-ciberfisicos-y-la-industria/

Huidobro, J. M. (jueves 22 de agosto de 2019). *Zona Movilidad*. Obtenido de Avances tecnológicos: https://www.zonamovilidad.es/avances-tecnologicos-y-su-impacto-en-la-sociedad.html

Karen Rose, S. e. (octubre de 2015). *Internet Society*. Obtenido de Reporte IoT: https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf

Ponce, A. (miercoles 18 de noviembre de 2020). *Tecnosinergia*. Obtenido de Domotica vs control manual: https://tecnosinergia.zendesk.com/hc/es/articles/115000197652--Qu%C3%A9-es-la-automatizaci%C3%B3n-del-hogar-y-que-beneficios-tiene-Dom%C3%B3tica-

Satish Palaniappan, A. D. (abril de 2015). *Research Gate*. Obtenido de Home Automation Systems: https://www.researchgate.net/publication/275338025\_Home\_Automation\_Systems\_-\_A\_Study

https://scielo.conicyt.cl/pdf/ingeniare/v27n3/0718-3305-ingeniare-27-03-00443.pdf

https://www.ticportal.es/expert/cps-cyber-physical-systems-implicaciones-desafios-futuros

https://www.redalyc.org/jatsRepo/944/94454631006/html/index.html

https://polipapers.upv.es/index.php/RIAI/article/view/8823/9688

https://polipapers.upv.es/index.php/RIAI/article/view/8823/9688

http://oa.upm.es/48625/1/Informe\_AM4G\_OA\_UPM\_2016.pdf

https://revistas.eia.edu.co/index.php/mem/article/download/826/744