



**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN
DEPARTAMENTO DE TELECOMUNICACIONES**

**Tesis Monográfica para optar al Título de
Ingeniero en Telecomunicaciones**

Título

“Smart lighting System controlado por Asistente virtual para el confort visual basado en IoT”.

Autores:

- Br. Carlos Abrahán Gutiérrez Pérez 2015-0054I
- Br. Scayli América González Palma 2015-0414I

Tutor:

MSc. Ing. Ernesto José Lira Rocha

Managua, enero 2022

DEDICATORIA

Dedicó esta tesis a Dios, quien me ha guiado y dado fortaleza de seguir adelante, A mis padres, por ser parte esencial en mi vida; motores de mis proyectos, guías, y ayuda presente en el momento de los problemas que se me presentaron, A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que nos abrieron puertas y compartieron sus conocimientos.

Scayli América González Palma

Le dedico en primer lugar a Dios por haberme permitido llegar hasta este punto; por haberme dado salud, ser el manantial de vida y darme lo necesario para seguir adelante día a día para lograr mis objetivos. Además, agradezco infinitamente a mis padres por darme las bases necesarias para culminar con éxito este gran proyecto monográfico.

Carlos Abraham Gutiérrez Pérez

AGRADECIMIENTO

Agradecemos a nuestro tutor **MSc.Ing. Ernesto Lira Rocha** por creer en nosotros, por escuchar y corregirnos constructivamente en el desarrollo de este trabajo monográfico.

Resumen

En esta tesis se describe la extensión de una plataforma que facilita el desarrollo de aplicaciones de Ambient Assisted Living AAL que permitan transformar un ambiente convencional en un ambiente inteligente utilizando tecnología de bajo costo.

En nuestro país la automatización del hogar para la mayoría de la gente es un tema de exclusividad y sobre todo de altos costos. Es desconocido para muchos, contrario a lo que se piensa, que hoy en día es posible tener un hogar muy de vanguardia, por lo que hemos visto factible la construcción de un prototipo llamado “**Smart lighting System**” apoyado por sistemas de hardware y software de control de bajo costo debido a que no todas las personas están en el poder adquisitivo de implementarlo en su hogar u oficina.

En el mercado actual se cuenta con un gran número de aplicaciones de muy alto nivel sobre el tema, las cuales en su mayoría tienen costos elevados como el **Hue Bridge Philips**, pero más que todo un sistema de código abierto. El sistema **Smart lighting System** está llamado a cumplir una labor de extensión a la comunidad, y brindar un importante aporte en la solución de problemas aparentemente simples, pero de especial significación para aquellos que no tienen acceso a elaboradas tecnologías o dispositivos importados a un alto costo, lo que en últimas representa un proyecto de innovación por el uso de recursos al alcance de la economía local.

Un sistema inteligente proporciona un sinnúmero de beneficios y ventajas inalcanzables mediante una instalación tradicional. Si tuviésemos que resumir las principales razones serían sin duda: comodidad, seguridad, confort, información, y ahorro energético, razones mencionadas que se reducen a una sola: aumento de la calidad de vida.

El objetivo principal de este trabajo de tesis es demostrar que es posible implementar el **Smart lighting System**, que ayudará a controlar la iluminación por medio de un asistente virtual (Amazon Echo), que va a interactuar con una placa Arduino programable, mediante conexión vía wifi, permitiendo que estos recursos autómatas simplifiquen el proceso de cambiar los perfiles de iluminación.

Indicé de Figuras

| | |
|--|----|
| Figura 1: Dominio de un sistema de AAL. (Kleinberger et. al, 2007). | 8 |
| Figura 2: Puente Hue de Philips | 10 |
| Figura 3: Sistema de iluminación de Philips adaptado a ritmos circadianos | 11 |
| Figura 4: Amazon Echo. | 16 |
| Figura 5: Interface de Smartnest y configuración de cuenta | 20 |
| Figura 6: Vinculación exitosa de Smartnest y Alexa vinculando dispositivos | 21 |
| Figura 7: Gestión proactiva de Alexa | 21 |
| Figura 8: Directivas de MQTT | 24 |
| Figura 9: Reportes enviados después de recibir una directiva. | 24 |
| Figura 10: Mensajes y descripción de solicitud de los dispositivos | 25 |
| Figura 11: Conexiones del módulo de conmutación | 27 |
| Figura 12: Esquema del Módulo MOSFET de alta potencia | 27 |
| Figura 13: Placa de Arduino | 29 |
| Figura 14: IDE de Arduino | 29 |
| Figura 15: Arduino con Batería | 30 |
| Figura 16: Arduino MKR WIFI 1010 | 32 |
| Figura 17: Interfaz del software de Arduino | 33 |
| Figura 18: Ventana del monitor serie | 34 |
| Figura 19: Tira led RGB | 35 |
| Figura 20: Led de una RGB | 35 |
| Figura 21: Infinitas combinaciones de color con el RGB | 36 |
| Figura 22: Característica de un circuito eléctrico de una tira RGB | 36 |
| Figura 23: Punto positivo de alimentación en común. | 37 |
| Figura 24: control de una tira led a través de un microcontrolador | 37 |
| Figura 25: Esquema General de la red LAN | 39 |
| Figura 26: Modem HG630 VDSL V2 | 41 |

| | |
|--|----|
| Figura 27: Esquema general para proyectos IoT con Arduino | 42 |
| Figura 28: Interacción de Amazon Echo con el servicio Alexa. | 45 |
| Figura 29: Elementos en el proceso de control de la Smart light system | 47 |
| Figura 30: Dispositivo creado en Smartnest Fuente propia | 48 |
| Figura 31: ID creado en Smartnest | 48 |
| Fuente 32: Fuente propia usuario creado en el servidor Smartnest | 49 |
| Fuente 33: Dispositivos controlados desde Amazon Alexa | 49 |
| Figura 34: rutinas creadas para los diferentes perfiles de iluminación | 50 |
| Figura 35: Algoritmo implementado en MRK WIFI 1010 | 52 |
| Figura 36: Funciones loop y checkMqtt | 53 |
| Figura 37: Función comunicación bróker-micro usando protocolo MQTT | 54 |
| Figura 38: Función Callback, envío y recepción mensajes con protocolo MQTT | 55 |
| Figura 39: Armado y pruebas de funcionamiento | 56 |
| Figura 40: Perfil de iluminación por comando de voz me siento alegre. | 57 |
| Figura 41: Perfil de iluminación por comando de voz me siento relajado. | 57 |
| Figura 42: Perfil de iluminación por comando de voz me siento cansada. | 58 |
| Figura 43: Perfil de iluminación por comando de voz, Alexa voy a dormir. | 58 |
| Figura 44: Presupuesto del diseño para la implementación | 61 |

INDICE DE CONTENIDO

| | |
|--|----|
| I. Introducción | 1 |
| II. Objetivos del Estudio..... | 3 |
| III. Justificación | 4 |
| IV. Marco Teórico | 5 |
| 4.1 Sistemas Iluminación Inteligente | 5 |
| 4.1.1 Aplicaciones de la domótica..... | 6 |
| 4.2 Inteligencia Ambiental (Aml) y sus aplicaciones más relevantes..... | 7 |
| 4.3 Ambient Assisted Living | 8 |
| 4.4 Centro de automatización doméstica inteligente Hue Bridge Philips..... | 10 |
| 4.5 Amazon Echo..... | 16 |
| 4.6 Smartnest | 17 |
| 4.6.1 Integración con Alexa..... | 20 |
| 4.7 MQTT..... | 22 |
| 4.7.1 N-MOS | 26 |
| 4.8 Generalidades Arduino..... | 28 |
| 4.9 MKR WIFI 1010..... | 31 |
| 4.10 Software | 33 |
| 4.11 Anatomía de una LEDS STRING RGB..... | 35 |
| 4.12 Red LAN | 39 |
| 4.13 Servidores Web | 40 |
| 4.14 Router y Modem | 40 |
| 4.15 Wifi | 41 |
| 4.16 Internet de las cosas..... | 42 |
| V. ANALISIS Y PRESENTACION DEL RESULTADO | 43 |
| VI. Conclusión | 62 |
| VII. Recomendaciones | 63 |
| VIII. Bibliografía | 64 |
| IX. ANEXOS | 66 |

I. Introducción

El presente trabajo de tesis se basa en la propuesta de un control de iluminación inteligente que permite funciones de automatización de un hogar inteligente tales como rutinas de programación para el confort visual, basándonos en el objetivo principal de AAL (Ambient Assisted Living) que aplica tecnología de inteligencia ambiental para permitir a las personas vivir con una mayor independencia y confort en sus hogares.

Basándose en la creación de la tecnología (Arduino), protocolos de comunicación & energía eléctrica permitirá el desarrollo de nuestra tesis, la cual se centra en la implementación de un sistema de iluminación inteligente que regule su intensidad por medio de diversos patrones identificados por la Inteligencia Artificial (Perfiles), estos perfiles se encargan de realizar muestras de la iluminación ambiente y verificar si proviene alguna orden del Asistente virtual.

Para desarrollar el prototipo se requiere un microcontrolador que sincronice el flujo de entrada de datos enviados por la I.A (Inteligencia Artificial) desde el Asistente Virtual hasta el circuito, con los requerimientos de iluminación solicitado del entorno. Dichas circunstancias involucran el estado de ánimo de la persona e interacciones posibles con la I.A.

El proyecto complementa la base de su estructura en el uso moderado de un Lenguaje de programación de un sistema embebido; Arduino. El entorno de Arduino es un **conjunto** de Processing y Wiring. De este último coge el lenguaje simplificado de C/C++ y su compilador y de Processing se queda con el entorno de desarrollo.

Permitiendo dicho lenguaje y protocolos de comunicación, desarrollar un prototipo a pequeña escala de un Sistema inteligente, Valiéndonos de la utilidad de la tarjeta MKRWIFI 1010, Como tarjeta de Recepción & ejecución lógica, a través de servidores estables de comunicación inalámbrica, dando paso a ejecución de datos por comandos de voz de una inteligencia Artificial.

En este documento se describe el desarrollo del trabajo monográfico, la introducción a dicho trabajo sobre el tema a desarrollar, los antecedentes de los trabajos relacionados a dicho tema, los objetivos planteados que se desean alcanzar , el marco teórico que sustenta el trabajo a desarrollar y el diseño metodológico que incluyen las herramientas teóricas que se necesitaron para la ejecución del trabajo de tesis que nos llevaron a los resultados para el proceso de desarrollo del trabajo en cuestión.

II. Objetivos del Estudio

4.1. Objetivo General

- Implementar un prototipo sobre Smart lighting System controlado por Asistente virtual para el confort visual basado en protocolos IoT.

4.2 Objetivo Especifico

- Implementar un algoritmo haciendo uso del lenguaje de programación Processing y Wiring para crear un cliente a fin de establecer una comunicación lógica entre Protocolos Web & Hardware, basado en desarrollo del IDE de ARDUINO con el fin de controlar el Smart lighting System.
- Integrar diferentes tecnologías para establecer una comunicación del reconocimiento del lenguaje hablado permitiendo de esta manera que el usuario puede usar un comando de voz para controlar las luces de la habitación.
- Crear diferentes perfiles de iluminación en un ambiente inteligente basado en el confort visual.
- Realizar el montaje del circuito electrónico que permita llevar a cabo el control del estado operativo de las luminarias.
- Elaborar un presupuesto que nos permita proyectar el costo del prototipo a implementar.

III. Justificación

Se propone extender y mejorar las capacidades de un sistema inteligente, integrando nuevos dispositivos comerciales para mejorar la captura de datos de los usuarios, con el objetivo de ampliar el alcance del mismo, permitiendo incorporar nuevos gestos para controlar dispositivos y mejorar el reconocimiento de comandos de voz utilizando una tecnología más precisa.

El sistema propuesto está basado en los principios de la tecnología asistida, centrado en la persona y no centrado en la tecnología. Esto quiere decir que no hay una única solución posible para asistir a una persona, sino que hay una solución para cada individuo, según sus habilidades, necesidades y contexto.

La motivación de este trabajo radica en aplicar conocimientos de ingeniería de software e internet de las cosas para extender una plataforma de vida cotidiana asistida por computadora que pueda ayudar a personas, brindándoles un mayor confort visual de iluminación dentro de sus viviendas y áreas de trabajo.

Este sistema de automatización es novedoso y tiene gran aceptación debido al uso de nuevas tecnologías y al gran aporte que este genera para la manipulación de control de luminosidad sin necesidad de un control manual. Este proyecto es accesible debido a la compatibilidad de los dispositivos que se van a implementar y a su fácil programación, tiene un impacto positivo, ya que tiene acceso al control de las luminarias a través remota por medio de una interfaz de línea de comandos y usuario de voz.

La Implementación de esta Tesis pretende, abrir camino en nuestra sociedad para la evolución de un campo que aún tiene mucho espacio para desarrollarse, abre una solución a actividades monótonas del día a día, aplicándose con eficiencia en campos como el debido uso de metodologías ingenieriles, resolución de actividades del día a día, adquiriéndose como un proyecto domótico tiene muchas utilidades con diversos elementos eléctricos, pretendemos desarrollar este proyecto con ardua dedicación para su adquisición en diversas áreas en donde puede resaltar su gran descubrimiento y uso.

IV. Marco Teórico

4.1 Sistemas Iluminación Inteligente

Ambient Assisted Living (AAL) es un área de investigación y desarrollo en la cual la accesibilidad, usabilidad y el aprendizaje juegan un papel de gran relevancia. El objetivo principal de AAL es aplicar tecnología de inteligencia ambiental para permitir a las personas una mayor independencia y confort en sus hogares. [3]

Así, entendemos por Iluminación Inteligente la incorporación de equipamiento en nuestras viviendas con tecnología que permita gestionar de forma energéticamente eficiente, segura y confortable para el usuario, los distintos aparatos e instalaciones domésticas que conforman la vivienda.

Es aquí donde toman protagonismo los controladores programables. Hoy en día, la complejidad de los sistemas a gobernar hace que la supervisión, control y regulación sean tareas difíciles de llevar a cabo por el ser humano.

La domótica también se suele asociar actualmente, sobre todo en ámbitos de telecomunicaciones, al denominado hogar digital u hogar conectado. El gran progreso tecnológico sufrido por los sistemas de telecomunicaciones y el desarrollo y proliferación de Internet, han incrementado exponencialmente nuestra capacidad para crear información, almacenarla, transmitirla, recibirla, y procesarla. El mayor acceso a la información, ha venido además asociado a una mayor facilidad para comunicarnos, para establecer nuevas vías de diálogo con el resto del mundo, en cualquier momento y desde cualquier lugar. [1]

Algunas de las ventajas de vivir en un hogar digital y conectado, son:

- Climatización y consumo energético.
- Entretenimiento y confort.
- Seguridad.
- Servicios comunitarios.

Los sistemas domóticos pueden ser clasificados:

- Centralizados
- Descentralizados

4.1.1 Aplicaciones de la domótica

Entendemos por aplicación al uso de sistemas técnicos que implementen ciertas funciones con el fin de satisfacer una determinada necesidad del usuario final. Desde el primer momento está presente la interacción entre personas y tecnología, entre sistemas técnicos y sistemas sociales.

Cualquier aplicación es susceptible de ser ofrecida como un servicio por parte de un proveedor de servicios: un actor externo especializado que permite el acceso, proporciona el mantenimiento o gestiona las funciones a las que nos referíamos con anterioridad. En este contexto, la provisión de un servicio por lo general lleva asociada una contraprestación, no necesariamente con pago, por parte del usuario doméstico a la empresa prestataria.

Existen abundantes aplicaciones y servicios domóticos: [1]

- Programación automática de luces.
- Control del aire acondicionado.
- Tele vigilancia profesional.
- Sistemas de diagnóstico médico remoto.
- Sistemas de difusión de audio en el hogar.
- Control de energía.
- Amenidades.

4.2 Inteligencia Ambiental (Aml) y sus aplicaciones más relevantes

La inteligencia ambiental, o Ambient Intelligence (Aml) es una disciplina en pleno crecimiento desarrollada a fines de los años 90. Aml describe un entorno en el que las personas están rodeadas y asistidas por interfaces inteligentes e intuitivas, embebidas en objetos cotidianos que se comunican entre sí. [3]

Estos conforman un medioambiente electrónico que reconoce y responde a la presencia de los individuos inmersos en él. Los avances tecnológicos, como el desarrollo de sensores, redes de sensores, inteligencia artificial y computación ubicua fortalecen la investigación en esta disciplina.

Estas tecnologías prometen revolucionar la vida cotidiana creando entornos flexibles y adaptables para las personas. Una de las áreas de aplicación más relevantes es colaborar con la vida independiente de las personas ancianas o discapacitadas, principalmente, con aquellas que padecen impedimentos físicos o cognitivos que vivan en sus casas, aunque, deban ser monitoreadas por el bien de su seguridad y bienestar. Vida cotidiana asistida por computadora o, en inglés Ambient Assisted Living (AAL) es un concepto derivado de Aml que no trata de automatizar tareas, sino que busca asistir a los usuarios en tiempo real dentro de sus viviendas.

4.3 Ambient Assisted Living

Ambient Assisted Living (AAL) es un área de investigación y desarrollo en la cual la accesibilidad, usabilidad y el aprendizaje juegan un papel de gran relevancia. El objetivo principal de AAL es aplicar tecnología de inteligencia ambiental para permitir a las personas vivir con una mayor independencia y confort en sus hogares. [3]

Algunas de las características más importantes de un sistema de AAL son:

- Ahorro de energía
- Comunicación inalámbrica para evitar cableados adicionales en las viviendas ya construidas.
- Capacidad para percibir la información del ambiente
- Adaptabilidad en el contexto.
- Proveer interfaces fáciles de utilizar por personas ancianas o discapacitadas.
- Interfaces de interacción con el sistema no tradicionales.

Los sistemas de AAL permiten a los usuarios interactuar con dispositivos electrónicos, tales como electrodomésticos y equipos de entretenimiento, luces y sistemas de alarmas, entre otros. Algunos sistemas brindan asistencia en salud y otros son capaces de adelantarse a siniestros al encontrar patrones en la conducta y movimientos del usuario que indiquen riesgo para su vida. Como se muestra en la siguiente figura, las aplicaciones de los sistemas de AAL pueden estar dirigidas a la asistencia a usuarios ante casos de emergencia, vida diaria o confort y combinaciones de las mismas:

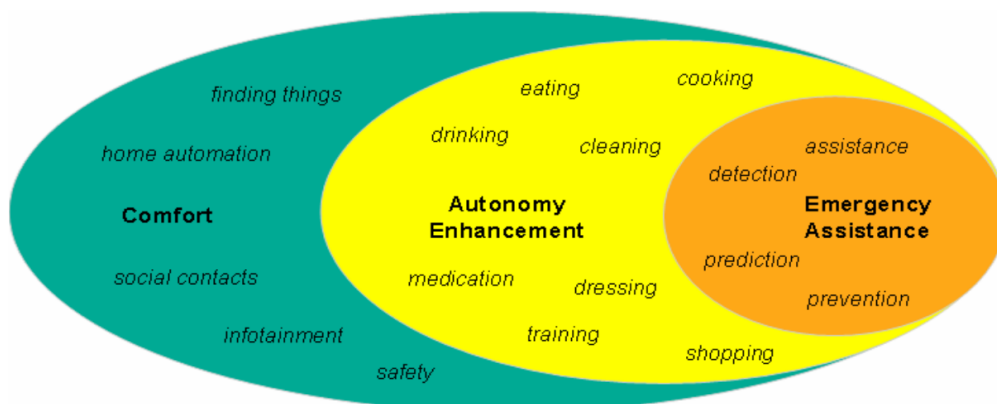


Figura 1 - Dominio de un sistema de AAL.

Una de las formas más importantes de interacción en ambientes asistidos se realiza a través de comandos de voz. Tal y como ocurre en la comunicación verbal, las personas tienden a realizar gestos al reproducir unidades lingüísticas que contienen información espacial y, generalmente, tienden a intercambiar palabras por gestos. Esto se ha considerado un factor fundamental para interpretar los requerimientos de los usuarios ya que, de esta forma, pueden comunicarse con el sistema de forma más natural. [2]

Uno de los desarrollos más importantes en AAL se encuentra en Bremen, Alemania. El laboratorio de vida cotidiana asistida por computadora cuenta con un departamento de pruebas, que recrea una casa habitable con capacidad para dos personas. El lugar posee electrodomésticos inteligentes adaptables y muebles para compensar las limitaciones especiales que se acercan al usuario cuando este los necesita a su alcance.

Adicionalmente, el laboratorio cuenta con una silla de ruedas inteligente que es capaz de transportar a los usuarios desde un ambiente a otro de forma autónoma. Esta réplica de una casa a escala real, indica el compromiso con la usabilidad y accesibilidad de este tipo de estudios.

En los sistemas de AAL, los usuarios son observados por medio de cámaras o sensores. Cada luz, alarma o electrodoméstico es controlado por medio de dispositivos actuadores que, generalmente, implican el uso de un circuito electrónico. Para ello, primeramente, se procesa en tiempo real la interacción entre el usuario y los dispositivos a través de gestos y comandos de voz, permitiendo al usuario ser su propio control remoto. Este conjunto de componentes se describe brevemente a continuación. [2]

4.4 Centro de automatización doméstica inteligente Hue Bridge Philips

El Hue Bridge es un componente esencial de un sistema de iluminación inteligente personal de Philips Hue. Es el cerebro del sistema que se comunica con las lámparas inteligentes y la app Hue para garantizar que todo funcione conjuntamente. También permite funciones de automatización de hogar inteligente tales como temporizadores y rutinas de programación. [4]



Figura 2: Puente Hue de Philips

Que es Hue Bridge

El Hue Bridge, el cerebro del sistema de iluminación inteligente Philips Hue, te permite conectar y controlar hasta 50 luces y accesorios. Basta enchufarlo y utilizar la app Hue para configurar rutinas, temporizadores, escenas de luz personalizadas, etc.

Cómo la iluminación puede ayudarnos a concentrarnos

Ya sea en un entorno laboral o en nuestro hogar, la búsqueda de un ambiente que invite a la relajación o la concentración suele estar relacionada con el sonido. Un vecino molesto o el ruido del tráfico pueden parecer nuestros peores enemigos, pero hay otro ingrediente que debemos tener muy en cuenta y suele pasar desapercibido: hablamos de la iluminación. [4]

La luz tiene un impacto muy importante en nuestro estado de ánimo, también en nuestra productividad. Cuando la iluminación de una estancia no está bien configurada, podemos encontrarnos con problemas poco evidentes, pero muy molestos, como la fatiga, el cansancio o los dolores de cabeza.

Una buena iluminación es, además, un factor importante para el desempeño de los estudiantes, logrando mejores resultados académicos aquellos que organizan sus estudios y clases en ambientes con luz ideal. Parece obvio, **la falta de luz reduce la capacidad del cerebro para procesar** y registrar la información.

Si queremos que la iluminación trabaje a favor de nuestro bienestar, tenemos que rodearnos de **sistemas que se puedan adaptar a nuestros ritmos circadianos**. A grandes rasgos, se trata de los cambios físicos, mentales y conductuales que siguen un ciclo diario y que responden, principalmente, a la luz y la oscuridad en el ambiente de un organismo. [4]

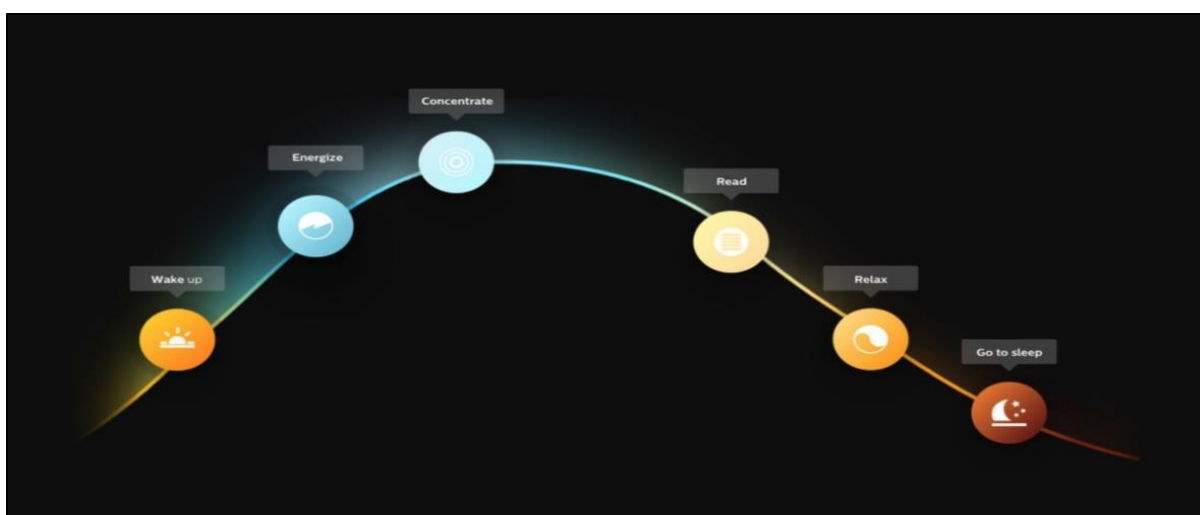


Figura 3: Sistema de iluminación de Philips adaptado a ritmos circadianos

Necesitamos luces inteligentes que se puedan ajustar de forma gradual en intensidad y temperatura a lo largo del día. También tenemos que entender los tipos de luz que podemos utilizar, su orientación y distancia, la mezcla ideal con la luz natural, así como algunos conceptos como temperatura de color o flujo luminoso.

Por ejemplo Philips Hue nos permite crear un sistema de iluminación LED completo en el que no hay que cambiar los elementos para poder cubrir todas las necesidades y puede ser controlado desde nuestro teléfono móvil, sin cables. [4]

La importancia de la temperatura de color

Hay quien se apoya en la **psicología de los colores** para determinar el diseño de una habitación y su iluminación. Parece una planificación acertada, aunque en el caso de la concentración, el blanco no tiene rival, por lo que en este artículo no abordaremos las opciones de iluminación en color disponibles. Con todo, a lo que tenemos que atender es a lo que se denomina como **temperatura de color**. Con ella, podemos crear la atmósfera adecuada.

Cualquier sistema de iluminación tiene que tener esta especificación bien visible. La temperatura **se mide en grados Kelvin** y, según la cifra, nos estará indicando si la luz es fría (más de 5.000K), cálida (de 2.500 a 3.300K) o neutra (entre 3.300 y 5.000K). En el primer caso, el color se asemeja más al azul, mientras que en el segundo el tono es más amarillo-anaranjado.

Elegir la temperatura de tus luces es **algo bastante subjetivo**, pero al mismo tiempo produce cambios en el ambiente y, por lo tanto, en las personas que van a vivir en él. En líneas generales, podemos establecer una serie de usos que están relacionados con la temperatura: [4]

- ✓ Si elegimos un **blanco cálido** —cercano a los 2.700K— obtendremos un ambiente suave, relajante y acogedor. Un dormitorio, salón o sala de estar podrían estar iluminados de esta forma.
- ✓ Para una iluminación **uniforme y neutra**, iremos a por un blanco de 3.500-4.000K. Es la forma de conseguir un ambiente equilibrado. La podemos utilizar en recibidores, pasillos o tiendas que no tengan necesidades especiales.
- ✓ El siguiente escalón lo podemos denominar como un **blanco más frío** (4.100K). Nos ofrecerá una iluminación limpia y radiante. Es la ideal para realizar tareas en entornos de estudio o trabajo, así como en cocinas y baños.
- ✓ El último nivel es reconocido como **luz del día**. Por encima de los 5.000K, nos ofrece una imitación de la luz solar y favorece un ambiente estimulante. Trabajos específicos como el dibujo técnico o la arquitectura podrían optar por esta temperatura.

Todos estos son ejemplos que pueden variar según nuestras necesidades, pues no todos usamos las estancias del mismo modo y los requerimientos de iluminación pueden cambiar incluso en una misma habitación. Por ejemplo, una luz en el dormitorio necesita **características diferentes en la mañana y en la noche**. Para dar una inyección de energía en la mañana, mejor una luz azul; sin embargo, en la noche la luz cálida nos ayudará a preparar el sueño.

El flujo luminoso

Hemos vivido una historia muy larga relacionada con bombillas halógenas e incandescentes, algo que nos ha hecho pensar en que la cantidad de luz emitida depende de los vatios consumidos, aunque en realidad está supeditada a **los lúmenes** producidos por el cuerpo.

¿Qué son los lúmenes? Es **la unidad utilizada para indicarnos la cantidad de luz emitida por una fuente**, es lo que podemos definir como el flujo luminoso o potencia lumínica. En este concepto no se tienen en cuenta factores condicionantes como la habitación o el ángulo de apertura del haz de luz.

La llegada de la tecnología LED ha revolucionado la iluminación en términos energéticos, también en la calidad de la misma

Las actuales **lámparas con tecnología LED son mucho más eficientes** que los sistemas de iluminación anteriores. Por ejemplo, para generar los 4.000 lúmenes necesarios para iluminar un salón, necesitaríamos una bombilla LED de 45 W, mientras que en el caso de una bombilla incandescente nos iríamos a un rango de 150 a 300 W.

Introduciendo este flujo luminoso en nuestra ecuación para conocer cómo relajarnos o concentrarnos, diremos que para el primer caso estaría bien elegir bombillas en torno a los 2.700K de temperatura con **unos 800 lúmenes** de potencia. Sería lo ideal para dormitorios o una iluminación suave de salón. [4]

Si vamos a necesitar activarnos, ya estamos hablando de luces que suben de los 4.000K de temperatura, pero no hay razón para pasar de los 800 lúmenes. Esta combinación es válida para una oficina o lugares donde necesitemos energía para trabajar.

Más allá de subir o bajar intensidad

Para tener esa capacidad de adaptar la iluminación a los requerimientos del momento, apostaremos por luces inteligentes que tengan la posibilidad de cambiar no solo su intensidad, sino la temperatura de color, como sucede con las Hue White ambiance de Philips, capaces de moverse **entre los 2.200 y 6.500K**.

Aunque todos los elementos se comercializan por separado, no es mala idea partir del kit de inicio, que reúne tres bombillas, un interruptor regulador y el puente Hue o Hue Bridge que hace de cerebro del sistema. Este componente controla hasta 50 luces y sensores en diferentes estancias y **las vincula con nuestro teléfono móvil**, que se convertirá en el controlador ideal de las operaciones. El Hue Bridge contempla además funciones de automatización del hogar inteligente, como temporizadores y rutinas de programación. Si no queremos usar nuestro teléfono, un método alternativo de control es el interruptor inalámbrico de pared, que a su vez también funciona como mando a distancia, ya que se puede soltar de su anclaje magnético. [4]

Como sucede con otras soluciones, quienes quieran algo fácil y rápido pueden echar mano de los cuatro modos de escenas de luz predefinidos: **Energía, Concentración, Lectura o Relax**. Están programados en el sistema, pero también pueden ser personalizados con la aplicación Hue.

¿Necesitas relajarte o concentrarte?

En el contexto de creación de ambientes que nos ocupa, es importante que haya diferentes puntos de luz para poder gestionar de forma separada la iluminación general de la de zonas más específicas. Unir luz natural con artificial debe llevarnos a la mejor iluminación posible. Por motivos horarios, no siempre es factible aprovechar la luz del sol, pero ten en cuenta que está ahí y puede entrar por tu ventana.

Lo ideal durante el día es tener una buena luz natural y que las fuentes artificiales sean las que rellenen lo poco que te falta para conseguir el efecto deseado sin perder de vista el equilibrio energético. Pero para alcanzar nuestro objetivo, no es suficiente con correr las cortinas o encender una luz por muy inteligente que sea.

Busco un ambiente de concentración

Si quieres llegar a concentrarte de verdad, no te queda otra que **eliminar las distracciones más importantes**, como pueden ser nuestro teléfono móvil, una consola de videojuegos, comida u otras personas. Hay que maximizar la atención en la tarea que tenemos entre manos.

Mantén limpio tu espacio de trabajo y toma descansos cada media hora

Organiza los elementos de la estancia de forma que estés cómodo, tengas todo a mano y no te encuentres agobiado. Lo más importante es una buena silla, ropa cómoda y una temperatura ideal. No descartes música de ambiente que nos ayude a restar ruido exterior, incluso podemos pensar en usar unos auriculares. En cuanto a la iluminación, ya lo hemos comentado, luz natural mezclada con artificial. Conforme avance el día, necesitaremos una luz ambiente o general para poder ver con garantías cosas de la habitación, además de una lámpara que se encargue de centrar el haz de luz en nuestra tarea. Bombillas con un índice superior a los 5.000K nos mantendrán más activos; más cercanas a los 3.000K, nos ayudarán en procesos más creativos. [4]

Es el momento de desconectar

Todo el mundo necesita **un instante de reposo** y relajación, y encontrarlo en tu hogar es lo ideal. Hay cosas básicas que hay que evitar, como el desorden, la suciedad o los olores. A la hora de construir tu santuario, necesitas medir el impacto del color de las paredes, muebles o accesorios como la típica mantita de sofá.

Aunque en próximos artículos nos detendremos en profundidad en el impacto del color en nuestro estado de ánimo y descubriremos las opciones de iluminación en este sentido, valga ahora decir que los tonos elegidos en cualquier formato, si son suaves y **recrean coloraciones del mar o el bosque**, tendrán un efecto casi balsámico. [4]

4.5 Amazon Echo

Amazon Echo es un dispositivo de reconocimiento de voz de Amazon.com, que se caracteriza por responder preguntas, reproducir música, y controlar dispositivos inteligentes. El mismo, incluye un parlante cilíndrico largo de 23 cm, con un micrófono de 7 piezas. Echo responde al nombre “Alexa”, aunque la palabra de activación puede ser modificada por el usuario. [5]

Alexa es el asistente virtual controlado por voz creado por Amazon, y lanzado en noviembre de 2014 junto a su línea de altavoces inteligentes Echo. Su nombre fue elegido por tener una consonante fuerte al principio e incluir una x, algo que haría que el asistente reconociese más fácil su nombre, y también en honor a la Biblioteca de Alejandría.

El funcionamiento de Alexa es muy parecido al de otros asistentes como Google Assistant, Siri y Cortana. Empiezas invocándolo diciendo su nombre, momento en el que el altavoz o dispositivo en el que esté integrado se pondrá a escuchar. Entonces debes decirle un comando con tu voz, y el asistente reconocerá lo que le preguntas y te dirá una respuesta. [5]



Figura 4 – Amazon Echo.

4.6 Smartnest

Smartnest es un servicio web que permite a entusiastas de la automatización del hogar con conocimiento en electrónica conectar sus propios dispositivos con asistentes de voz como Google Assistant, Amazon Alexa, Siri, IFTTT y más. El servicio puede ser utilizado con diferentes tarjetas de Arduino y también con otros dispositivos con los microcontroladores STM32. la comunicación se realiza por medio del protocolo MQTT, por lo que solo necesitas una Librería MQTT y tendrás acceso a todos los servicios. [6]

¿Por qué usar Smartnest?

Digamos que acabas de construir un ingenioso sistema que abre y cierra tus cortinas, sería interesante controlarlo usando tu voz, o desde su teléfono, o desde un navegador web, o simplemente usando cualquier dispositivo a tu alcance de la manera que sea más conveniente para ti en cualquier momento, pero esto requeriría que programes tu dispositivo para conectarse a cada servicio, configures tu propio servidor web, configurarlo de manera segura, administrar solicitudes web, etc., o simplemente puedes registrarte y usar Smartnest gratis, puedes iniciar y en menos de 10 minutos tendrás tu dispositivo conectado a cientos de servicios web. [6]

¿Por qué se llama Smartnest?

Todos están hablando hoy sobre Smart Homes o casas inteligentes, pero nos gusta pensar en ellos como Smart nests o nidos inteligentes ya que eres tú, quien se toma el tiempo para salir y obtener componente por componente para construir algo propio y configurarlo para ti y tu familia o amigos. [6]

¿Qué integraciones hay disponibles?

Smartnest está ahora en fase Beta

Aquí hay una lista de los métodos disponibles para controlar tus dispositivos y su estado:

- Desde tu computadora: Usando la aplicación web [Visitar Web App](#)
- Desde tu teléfono: Puedes utilizar la aplicación para [iOS](#) y [Android](#), o si prefieres puedes agregar el ícono de la aplicación web a tu pantalla de inicio para iOS, Android y Windows phone. [Visitar Web App](#)
- Desde tu tableta: Usando la aplicación para [iOS](#) y [Android](#), o la aplicación web que se adaptarán a cualquier tamaño de pantalla. [Visitar Web Ap](#)
- Skill de Alexa: la skill está disponible en la tienda de skills de Amazon. [Visita la Skill](#)
- Siri: descarga el [atajo de Siri](#).
- Google Home: Activa la acción de Google Home. Sigue la [guía de integración con Google](#)
- IFTTT: Activa el servicio de Smartnest en tu aplicación de [IFTTT](#)
- API: se dará acceso completo a la API a los desarrolladores de aplicaciones (Es necesario contar con un plan tipo business) [6]

Desarrollos futuros:

- Controlador para cortinas
- Controlador para luces de colores

La conexión se realiza utilizando MQTT, hay muchas librerías MQTT en la web como pubsubclient de knolleary, se recomienda elegir la que mejor se adapte a tu proyecto y a tu tarjeta.

Además para seleccionar la librería que más se adecua, se debe tomar en cuenta el modelo de tarjeta con el que estás trabajando, así como los periféricos, para más detalles puedes ver la librería pubsubclient de knolleary que admite los siguientes dispositivos: [6]

- Arduino Ethernet
- Arduino Ethernet Shield
- Arduino YUN
- Arduino WiFi Shield
- Sparkfun WiFly Shield

-
- TI CC3000 WiFi
 - Intel Galileo/Edison
 - ESP8266
 - ESP32

Primeros pasos con Smartnest

Bienvenido al futuro de la automatización del hogar, estos son los pasos para que puedas comenzar rápidamente:

Antes de comenzar necesitarás:

- Algún dispositivo programable con acceso a internet (Arduino, RaspBerry, ESP)
- Una [librería MQTT](#) que puedas utilizar con tu dispositivo.

Pasos:

1. Crea tu cuenta en [el sitio web de Smartnest](#).
2. Haz clic en Agregar dispositivo y dale un nombre
3. Haz clic en el botón con los tres puntos y copia el ID del dispositivo
4. Configura tu cliente MQTT con las siguientes opciones:
 1. Dirección del Broker: 3.122.209.170
 2. Puerto del Broker: 1883.
 3. ID del Cliente: (ID del dispositivo copiado en el paso 3)
 4. Nombre de usuario: (Tu usuario de Smartnest)
 5. Contraseña: (Tu contraseña de Smartnest)
5. Haz clic en el widget del dispositivo en la aplicación web y revisa si has recibido la instrucción en tu dispositivo.
6. Agrega Integraciones, sigue los pasos de las siguientes guías para conectar tu cuenta de Smartnest con Amazon Alexa, el Asistente de Google, Siri y más. [6]

4.6.1 Integración con Alexa

Para configurar la integración con Alexa, debes ir a tu aplicación de Alexa y abrir el menú tocando el icono en la esquina superior izquierda, luego seleccionar la opción "Skills y juegos" y luego tocar el botón de búsqueda en la esquina superior derecha, escribir Smartnest y luego tocar buscar. [6]

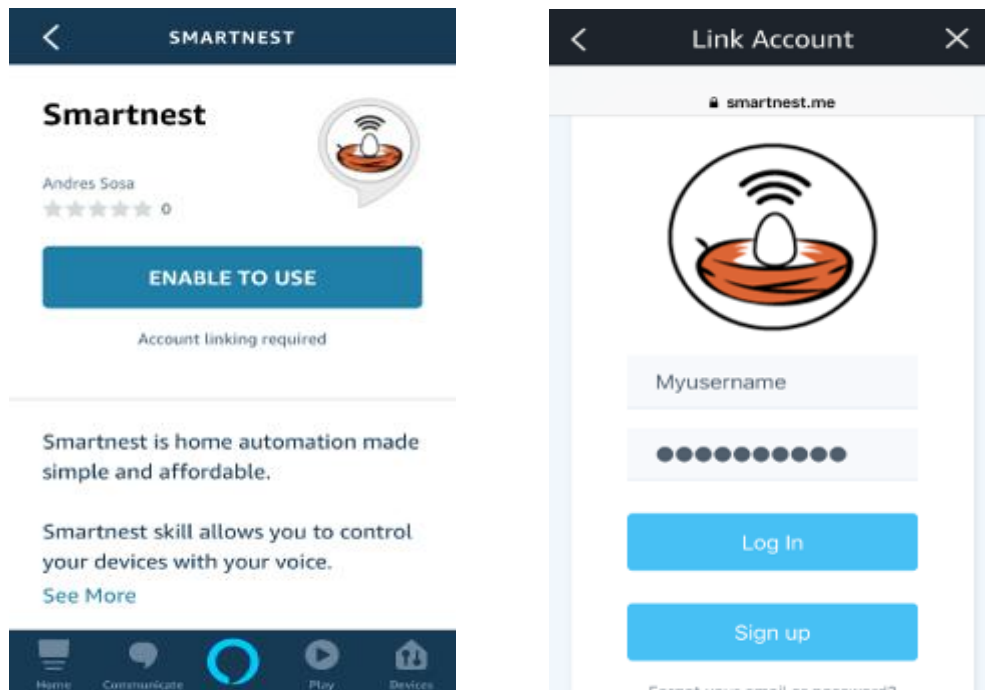


Figura 5: Interface de Smartnest y configuración de cuenta

Verás la skill de Smartnest lista para ser vinculada a tu cuenta, toca el botón Habilitar para usar y serás redirigido a una pantalla de inicio de sesión. Después de proporcionar tu nombre de usuario y contraseña, verás la página de confirmación que dice que la skill se ha vinculado correctamente. [6]

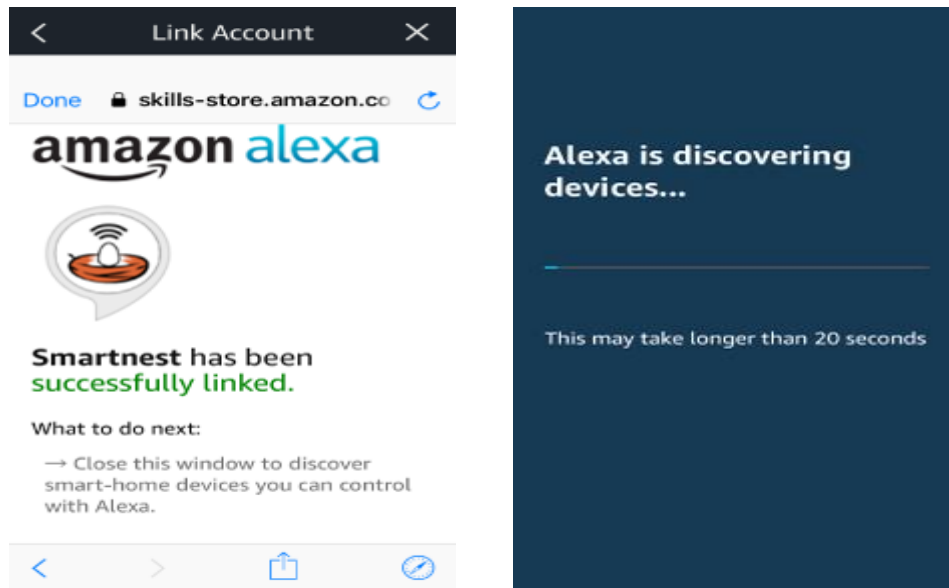


Figura 6: Vinculación exitosa de Smartnest y Alexa vinculando dispositivos

Después del primer enlace, Alexa comenzará a descubrir dispositivos y obtendrá todos los dispositivos que se hayan creado en tu cuenta. Alexa mostrará una confirmación de cuántos dispositivos encontró y luego podrás configurarlos y agregarlos a sus habitaciones o rutinas. [6]

La skill de Alexa tiene una gestión proactiva de los dispositivos, lo que significa que cada vez que crees o elimines un dispositivo en la aplicación de Smartnest Alexa hará lo mismo y también te mostrará una notificación de los nuevos dispositivos. [6]

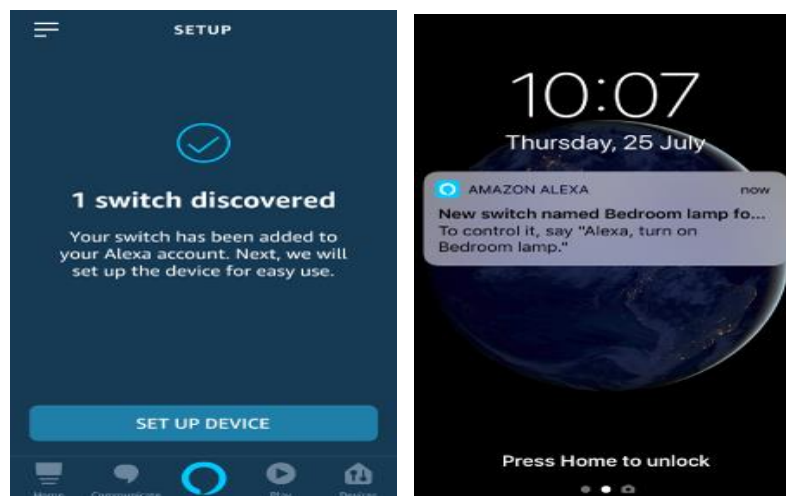


Figura 7: Gestión proactiva de Alexa

4.7 MQTT

Qué es MQTT

MQTT es un protocolo de mensajería simple y liviano, diseñado para dispositivos con poca memoria y con bajo ancho de banda. Esto significa que es la solución perfecta para las aplicaciones de Internet de las cosas y automatización del hogar y, por lo general, es más flexible, estable y confiable que otros protocolos como WebSockets. MQTT te permite enviar comandos para controlar salidas digitales, leer y escribir datos en sus dispositivos y más. [9]

Conceptos básicos de MQTT

En MQTT hay 3 conceptos básicos:

Temas (Topics):

Estos funcionan como la dirección de una casa, en este caso son las direcciones de un mensaje y determinan a donde se debe entregar, así es como se vería un tema con su respectivo mensaje:

Topic: MiCasa/Sala/Lampara

Mensaje: Encender

Aquí el tema es MiCasa/Sala/Lámpara y el mensaje es Encender, en este caso la lámpara recibirá el mensaje y se encenderá. Esta es la estructura básica de todos los mensajes que se envían a través de MQTT.

Sistema de Publicación y Suscripción

Cada dispositivo puede suscribirse a temas, y recibirá todos los mensajes enviados con los temas a los que están suscritos. pueden suscribirse a múltiples temas y también a subtemas.

Además, al publicar un mensaje, los dispositivos deben proporcionar el tema para que el mensaje se pueda entregar correctamente. [9]

Broker:

Es el dispositivo central que permite y restringe las conexiones y también recibe, filtra, redirige y publica mensajes a todos los dispositivos conectados. En este caso, el Broker es el servidor de Smartnest.

MQTT no es solo una herramienta para hablar con el servidor ,también puedes interconectar tus dispositivos usando este protocolo, por ejemplo, si tienes un sensor de movimiento y un sistema de iluminación, y están demasiado lejos el uno del otro, no necesitas instalar cables en la habitación para permitirles comunicarse, puedes hacer que la luz se suscriba al tema Casa/Dormitorio/Luz y puede hacer que el sensor publique los datos en ese tema, de tal forma estarán conectados entre sí y al mismo tiempo que se conectarán a Smartnet y todas las integraciones. [9]

Smartnest y MQTT

En Smartnest, cada dispositivo tiene una ID único, este ID debe usarse como la primera parte de un tema. Si se está enviando un mensaje al broker, la primera parte del tema debe ser el ID del dispositivo que está enviando el mensaje. Si se está enviando el mensaje a otro de sus dispositivos, la primera parte del tema debería ser el ID del dispositivo al que envía el mensaje, el usuario puede personalizar las siguientes partes del tema. [9]

Por ejemplo, este sería un tema válido desde dispositivo para Smartnest Broker:

Tema: jasds3421FDC1CDSCDSfdsas/Habitación/Temperatura

mensaje:25

Mientras que este sería un mensaje inválido

Tema: Bedroom/Temperature **mensaje:**25

El sistema solo permitirá la publicación de un mensaje si el ID es válido y pertenece al usuario que lo envía.

Comandos MQTT de Smartnest

Hay algunos subtemas reservados para fines especiales. Aquí está la lista de todos los temas reservados:

Directivas:

Las directivas son enviadas desde el servidor al dispositivo cada vez que hay un nuevo comando de alguna de las integraciones.

| Topic | Mensaje | Descripción | Soportado por |
|--------------------------------|--------------|------------------------------------|---------------|
| DeviceID/directive/powerState | ON - OFF | Encendido o apagado | Todos |
| DeviceID/directive/percentage | 0-100 | Porcentaje | Luz |
| DeviceID/directive/serpoint | -100 - 200 | Punto deseado de temperatura | Termostato |
| DeviceID/directive/lockedState | true - false | true significa cerrar la cerradura | Cerradura |

Figura 8: Directivas de MQTT

Reportes

Los reportes son enviados desde el dispositivo al servidor para informar de algún cambio de estado. Estos reportes son enviados siempre después de recibir una directiva y también si se produce algún cambio debido a una interacción física como presionar un botón. [9]

| Topic | Message | Description | Supported types |
|--------------------------------|--------------|-----------------------------------|---------------------------------------|
| DeviceID/report/powerState | ON - OFF | Nuevo estado de poder | Todos |
| DeviceID/report/percentage | 0-100 | Nuevo porcentaje | Luz |
| DeviceID/report/serpoint | -100 - 200 | Nuevo punto deseado | Termostato |
| DeviceID/report/temperature | -100 - 200 | Nueva temperatura | Termostato, Sensor de Temperatura |
| DeviceID/report/scale | C - F | Seleccionar la escala de T | Termostato, Sensor de Temperatura |
| DeviceID/report/lockedState | true - false | Disp. abierto o cerrado | Cerradura |
| DeviceID/report/online | true - false | El disp. está conectado | Todos |
| DeviceID/report/info | (Any) | Para mostrar en la aplicación Web | Todos |
| DeviceID/report/detectionState | true - false | Nuevo valor lógico | Sensor de Presencia , Sensor Contacto |
| DeviceID/report/notification | (Any) | Envía una notificación push | Todos |

Figura 9: Reportes enviados después de recibir una directiva.

Solicitudes y respuestas

Las solicitudes son enviadas desde los dispositivos al servidor para solicitar información especial como la hora, el servidor responderá con un topic response y la información requerida en el mensaje.

| Topic | Message | Description |
|------------------------|----------|---|
| DeviceID/request/time | GMT+XX | Obtener el tiempo real, el mensaje puede tener opcionalmente la zona horaria "GMT+XX", si la zona horaria no está presente en el mensaje se enviará la hora de GMT+00 |
| DeviceID/response/time | HH:MM:SS | Responde la solicitud de hora. |

Figura 10: Mensajes y descripción de solicitud de los dispositivos

Reglas en MQTT en Smartnest

En Smartnest somos una comunidad de automatización del hogar y todos estamos utilizando los mismos recursos web, de tal modo que deben existir reglas para garantizar un servicio continuo, seguro y confiable. Estas son las reglas que todos deben seguir al usar comandos MQTT: [9]

- La conexión al broker solo se permitirá cuando se proporcione un nombre de usuario y contraseña válidos
- La conexión se permitirá cuando la ID del dispositivo utilizado pertenezca al nombre de usuario utilizado
- La primera parte del topic MQTT debe ser un ID válido de uno de los dispositivos del usuario utilizado
- Se debe evitar enviar mensajes a los temas reservados con demasiada frecuencia y se debe enviar sólo cuando el estado del dispositivo ha cambiado.

(los dispositivos que no sigan estas reglas se desconectan y bloquean automáticamente por el sistema, el primer bloqueo es por 1 minuto, el segundo es por 2 minutos el tercero por 4 y así sucesivamente.) [9]

4.7.1 N-MOS

La placa es de módulo del controlador del interruptor del disparador del MOS del poder más elevado y tablero de control con el tubo del efecto del campo para ajustar el interruptor electrónico. Adopta la salida activa paralela del MOS doble para mostrar una resistencia interna más baja, una mayor corriente eléctrica y potencia. Además, funciona a 15A, 400W bajo temperatura común que satisface la mayoría de los dispositivos de uso y realiza el control de dispositivos de gran potencia de una manera relajada.

Lo que, es más, se puede aplicar a dispositivos potentes que podrían ser controlados por terminal de salida como motores eléctricos y bombillas para controlar la velocidad del motor y la luz de bulbs. Ciertamente, usted puede aplicarlo a los dispositivos que podrían entrar PWM para el control como correa de luz LED, motor de CC, bomba micro y válvulas solenoide.[10]

Especificaciones técnicas:

Voltaje de funcionamiento: DC 5V - 36V

La fuente del disparador: digital alto-bajo (DC3.3V - 20V), se puede conectar el puerto de E/S del microcontrolador, las interfaces del PLC, la alimentación de CC, usted puede acceder a la señal PWM, la frecuencia de señal 0--20KHZ soporte perfecto
Capacidad de salida: DC 5V - 36V, a temperatura ambiente, corriente continua 15A, potencia 400W, mejora las condiciones de enfriamiento, la corriente máxima de hasta 30A

Aplicaciones: Puede controlar la salida de equipos de potencia, motores, bombillas, luces LED, motores de CC, microbombas, electroválvulas, etc., puede entrar PWM, controlar la velocidad del motor, brillo de la lámpara y así sucesivamente

Temperatura de funcionamiento: -40.0-185.0 °F

Dimensiones: 1.339 in x 0.669 in x 0.472 in (largo x ancho x alto). [10]



Figura 11: Conexiones del módulo de conmutación

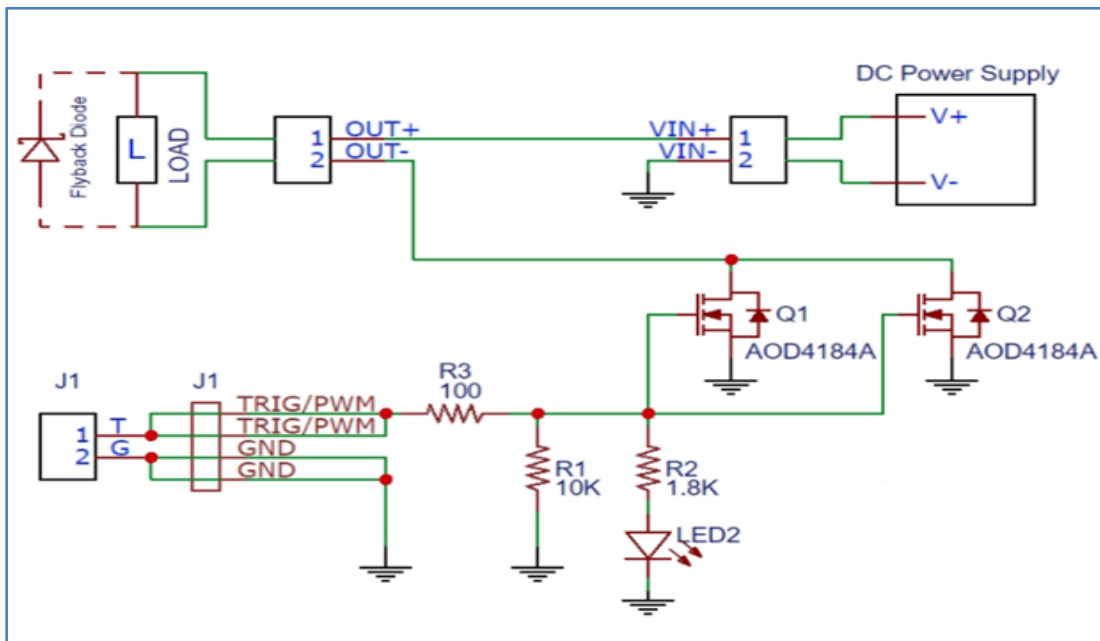


Figura 12: Esquema del Módulo MOSFET de alta potencia.[10]

4.8 Generalidades Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware libre, flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquier interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada, para esto toda una gama de sensores puede ser usada y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing).

Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectarlo a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP). [11]

Las placas pueden ser hechas a mano o comprarse montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, así pues, eres libre de adaptarlos a tus necesidades.

Qué es Arduino

Las placas Arduino son pequeños ordenadores con los que puedes leer información de diferentes sensores, así como controlar luces, motores y muchas otras cosas. La gran mayoría de los sistemas que nos rodean son ordenadores de diferentes tamaños. Los ordenadores no necesitan tener teclado ni pantalla. Hay ordenadores en el microondas de la cocina, dentro de los ascensores para detectar qué botón pulsas y, en los coches. Hay más de 70 ordenadores similares a Arduino... hay ordenadores por todas partes. [11]

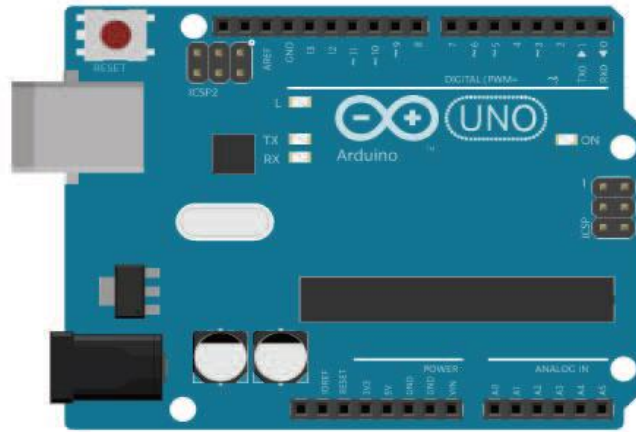


Figura 13: Placa de Arduino

Puesto que Arduino, a diferencia del ordenador que usas normalmente, no tiene pantalla ni teclado, se necesita un programa externo ejecutado en otro ordenador para poder escribir programas para la placa Arduino.

Este software es lo que llamamos Arduino IDE. IDE significa “Integrated Development Environment” (Entorno de Desarrollo Integrado), y es un término común para llamar a este tipo de desarrollo de software. Escribes tu programa en el IDE, lo cargas en el Arduino, y el programa se ejecutará en la placa.

El IDE de Arduino es muy sencillo y parecido a Processing. Hay una sencilla barra de herramientas que puedes utilizar para:

| | | |
|--|---|--|
| | Verificar si tu programa va a funcionar. | |
| | Cargar el programa a la placa de Arduino. | |
| | Crear un programa nuevo. | |
| | Abrir un programa. | |
| | Guardar el programa en el disco duro del ordenador. | |
| | (En la parte derecha de la barra de herramientas se encuentra el Monitor Serial) abre una ventana de comunicación con la placa Arduino. | |

Figura 14: IDE de Arduino

Las placas Arduino se conectan a tu ordenador utilizando un cable USB, al igual que cualquier otro periférico, como la impresora, el teclado o incluso, un mando de videojuegos. Arduino necesita estar conectado al ordenador a través del cable USB para cargar un programa. El cable USB sirve también para suministrar energía a la placa, pero también puedes alimentarla usando una fuente de energía externa, como una batería o un transformador apropiado. [11]

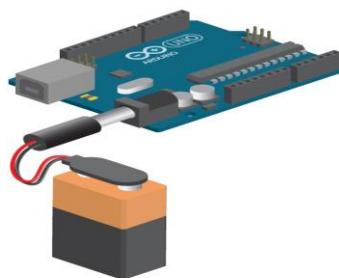


Figura 15: Arduino con Batería

Cuando programes tu Arduino, debes asegurarte de que el IDE está configurado correctamente para la placa que estás utilizando. Compruébalo en el menú 'Herramientas → Puerto Serial' para ver que el puerto esté configurado correctamente, y en 'Herramientas → Tarjeta' para comprobar que esté asignado 'Arduino Uno'. [11]

Nota: para saber a qué Puerto Serial está conectado tu Arduino, sigue estos pasos:

1. Desconecta Arduino del ordenador.
2. Comprueba 'Herramientas → Puerto Serial' para ver si está disponible.
3. Conecta Arduino al ordenador.
4. Entra en 'Herramientas → Puerto Serial' de nuevo y verás un puerto nuevo, ése es tu Arduino.
5. Selecciona ese puerto.

4.9 MKR WIFI 1010

El Arduino MKR 1010 está equipado con un módulo ESP32 hecho por U-BLOX. Esta tarjeta está diseñada para acelerar y simplificar el prototipado de aplicaciones IoT basadas en el WiFi gracias a la flexibilidad del ESP32 y su bajo consumo de energía.

La placa está compuesta por tres bloques principales:

- SAMD21 Cortex-M0+ de 32 bits y de bajo consumo
- Wi-Fi IEEE® 802.11 b / g / n U-BLOX serie NINA-W10 de baja potencia de 2,4 GHz
- ECC508 Crypto Authentication.

El MKR WIFI 1010 cuenta con una arquitectura computacional de 32 bits, un conjunto amplio de interfaces I/O y Wi-Fi de bajo consumo con un Cryptochip que utiliza encriptación SHA-256 para una comunicación segura. Además, ofrece una gran facilidad de uso, desarrollo y programación gracias al Software de Arduino (IDE). Todas estas características hacen a esta placa la preferida para proyectos IoT energizados mediante baterías. [12]

La placa es compatible con dos formas de alimentación; una batería de Li-Po y USB de 5V. Debido a que la tarjeta cuenta con un circuito de carga es posible suministrar 5V mediante USB mientras la batería es recargada. El cambio de fuente de energía es realizado automáticamente por el Arduino MKR WIFI 1010. Advertencia: El voltaje de operación del MKR WIFI 1010, a diferencia de la mayoría de tarjetas Arduino, es de 3.3V y es este el nivel máximo tolerado por los pines I/O. Rebasar este límite de tensión en cualquiera de las terminales puede causar daños a la placa. Aunque la salida digital hacia dispositivos de 5V es posible, la comunicación bidireccional con estos requiere de un cambio de niveles de voltaje adecuado. [12]

Especificaciones

- Microcontrolador SAMD21 Cortex-M0 + 32 bit MCU de bajo consumo de energía
- Fuente de alimentación de la placa (USB / VIN) 5V
- Batería compatible (*) de celda única Li-Po, 3.7 V, mínimo 700mAh
- Voltaje de funcionamiento del circuito 3.3V
- Pines de E/S digitales: 8

-
- Pines PWM: 12 (0, 1, 2, 3, 4, 5, 6, 7, 8, 10, A3 - o 18 -, A4 -o 19)
 - UART: 1
 - SPI: 1
 - I2C: 1
 - I2S: 1
 - Conectividad: wifi
 - Entradas analógicas: 7 (ADC 8/10/12 bit)
 - Salidas analógicas: 1 (DAC 10 bit)
 - Interrupciones externas: 8 (0, 1, 4, 5, 6, 7, 8, A1 -o 16-, A2- o 17)
 - Corriente DC por pin: 7 mA
 - Memoria Flash: 256 KB
 - SRAM: 32 KB
 - EEPROM: No
 - Velocidad del reloj: 32.768 kHz (RTC), 48 MHz
 - Dispositivo USB de velocidad completa y host integrado: incluido
 - LED_BUILTIN: 6
 - Longitud: 61.5 mm
 - Ancho: 25 mm

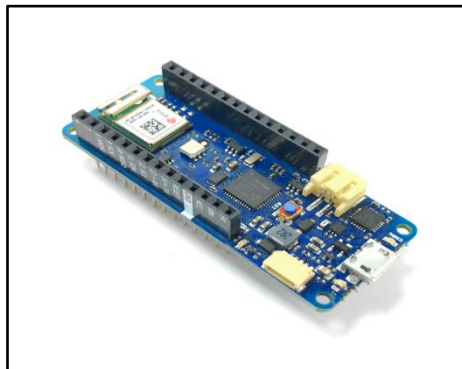
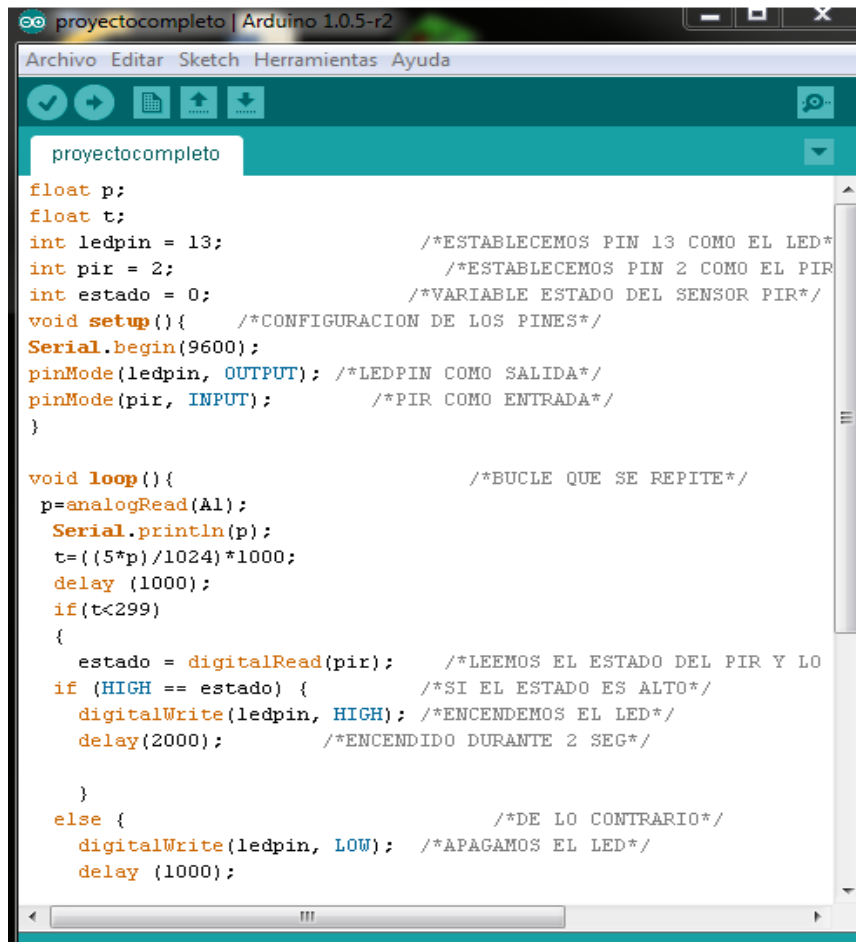


Figura 16: Arduino MKR WIFI 1010 [12]

4.10 Software

El software o plataforma Arduino tiene un lenguaje propio que está basado en C/C++ y por ello soporta las funciones del estándar C y algunas de C++. No obstante, es posible la utilización de otros lenguajes de programación y aplicaciones populares en Arduino como Java, Processing, Python, Matemática, Matlab, Perl, Visual Basic, etc. Esto se puede realizar debido a que Arduino tiene comunicación mediante la transmisión de datos en forma serie que es una característica que los lenguajes anteriormente citados soportan. Para los que no soportan el formato serie pueden hacer uso de software intermediario que traduzca la información de ambas partes para que haya una comunicación fluida.[8]

La Figura indica la interfaz IDE en la que se puede trabajar con Arduino de una manera directa.

The image shows a screenshot of the Arduino IDE software interface. The window title is "proyectocompleto | Arduino 1.0.5-r2". The menu bar includes "Archivo", "Editar", "Sketch", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for saving, undo, redo, and other functions. The main area displays a C++ sketch for a project named "proyectocompleto". The code defines variables for a float 'p', a float 't', an integer 'ledpin' (13), an integer 'pir' (2), and an integer 'estado' (0). The 'setup' function configures the pins and starts the serial port at 9600 baud. The 'loop' function reads an analog value 'p', prints it, and delays for 1000ms. It then checks if 't' is less than 299. If true, it reads the PIR sensor state. If the state is HIGH, it turns the LED on and delays for 2000ms. Otherwise, it turns the LED off and delays for 1000ms.

```
float p;
float t;
int ledpin = 13;           /*ESTABLECEMOS PIN 13 COMO EL LED*/
int pir = 2;              /*ESTABLECEMOS PIN 2 COMO EL PIR*/
int estado = 0;          /*VARIABLE ESTADO DEL SENSOR PIR*/
void setup(){            /*CONFIGURACION DE LOS PINES*/
  Serial.begin(9600);
  pinMode(ledpin, OUTPUT); /*LEDPIN COMO SALIDA*/
  pinMode(pir, INPUT);    /*PIR COMO ENTRADA*/
}

void loop(){             /*BUCLE QUE SE REPITE*/
  p=analogRead(A1);
  Serial.println(p);
  t=(5*p)/1024*1000;
  delay(1000);
  if(t<299)
  {
    estado = digitalRead(pir); /*LEEMOS EL ESTADO DEL PIR Y LO
  if (HIGH == estado) {      /*SI EL ESTADO ES ALTO*/
    digitalWrite(ledpin, HIGH); /*ENCENDEMOS EL LED*/
    delay(2000);           /*ENCENDIDO DURANTE 2 SEG*/
  }
  else {                     /*DE LO CONTRARIO*/
    digitalWrite(ledpin, LOW); /*APAGAMOS EL LED*/
    delay(1000);
  }
}
```

Figura 17: Interfaz del software de Arduino

A continuación, se describe las opciones más relevantes del programa.

Menú

La parte más importante está en la pestaña de Herramientas, desde ahí podremos configurar el programa para permitir la comunicación con la placa Arduino.

Al pasar el mouse sobre la Tarjeta aparecerá una lista con los tipos de placa Arduino que el programa comprende, aquí seleccionaremos el Arduino que se esté utilizando. Dependiendo del tipo de sistema que utilice la PC se asignará un puerto exclusivo para el Arduino. [8]

Botones comunes

Estos botones son accesos rápidos a ciertas acciones que también están disponibles mediante el menú. Los botones son los siguientes:

- Verificar: comprueba y compila el código.
- Cargar: además de compilar el código lo inyecta en la placa.
- Nuevo: crea un nuevo sketch.
- Abrir: abre un sketch previamente guardado.
- Guardar: almacena en disco los cambios realizados en el sketch.
- Monitor Serial: abre una nueva ventana desde la que podremos comunicarnos bidireccional vía serie con la placa, es decir, podremos leer la información que nos envía o proporcionar



Figura 18: Ventana del monitor serie [8]

4.11 Anatomía de una LEDES STRING RGB

Las tiras RGB están compuestas por especiales LED llamados RGB (red, green, blue) porque en el mismo substrato se encuentran tres LEDs, cada uno de un color básico. En la mezcla aditiva, es decir, en el campo de la iluminación las luces básicas son la roja, la verde y la azul. [7]



Figura 19: Tira led RGB

En las fotografías podemos ver una tira RGB y un detalle de led donde se pueden observar los distintos LEDs. Se puede observar a la izquierda el conector de conexión de 4 contactos. Tres de ellos sirven para los distintos canales (R, G y B) mientras que el último es común a todos y se debe conectar a positivo. [7]



Figura 20: Led de una RGB

La ventaja de estas tiras es que, controlando la intensidad de los tres colores básicos, se puede obtener cualquier color de luz que se desee. Por ejemplo, si encendemos los tres canales al máximo obtendremos el color blanco, si encendemos el azul y el rojo obtendremos el púrpura mientras que si encendemos el verde y el rojo obtendremos el color amarillo. Naturalmente, usando sistemas que controlan la intensidad de cada color, se pueden lograr infinitas combinaciones de color. [7]

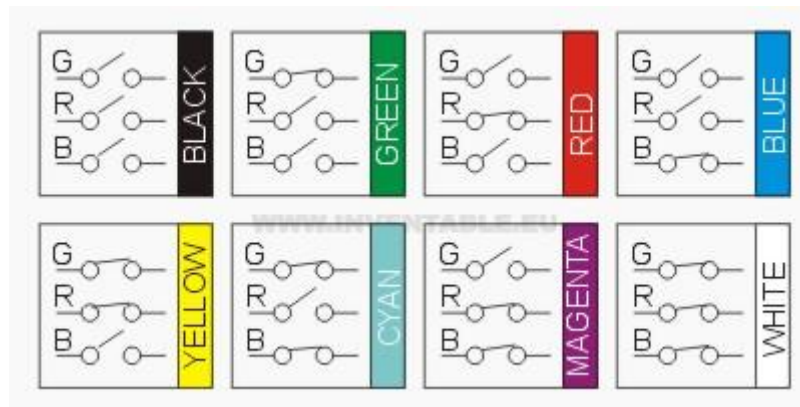


Figura 21: Infinitas combinaciones de color con el RGB

Características principales de una tira RGB genérica:

- Tensión de alimentación: 12V DC
- Largo de la tira: 5 metros
- Potencia: 14,4 Watt por metro
- Cantidad de LEDs: 300
- Angulo de luz: 120 grados
- Característica de uso: tiras flexibles, autoadhesivas y cortables cada 3 LEDs

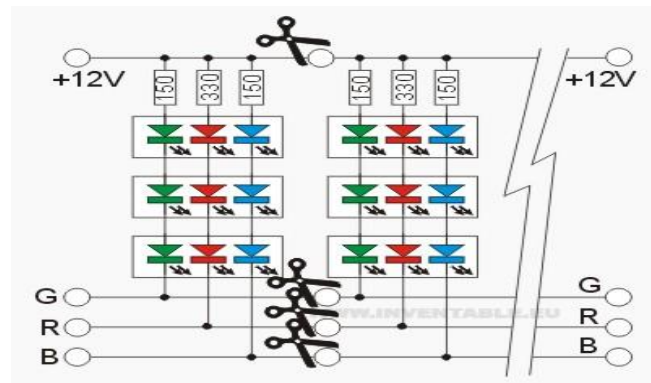


Figura 22: Característica de un circuito eléctrico de una tira RGB

En la figura podemos ver el circuito eléctrico de una tira RGB donde se observa que los canales de los distintos colores están conectados en grupos de 3 LEDs en serie con 3 resistencias (una para cada canal) y estos grupos, en paralelo con las líneas de alimentación. Esto nos permite de poder cortar las tiras en cualquier punto, pero siempre cada tres led. El canal rojo tiene una resistencia más grande porque la tensión característica de la juntura roja es menor respecto a la de los otros colores. [7]

Observen también que los grupos de LEDs tienen como común el positivo de alimentación por lo tanto un eventual circuito de control enciende los leds llevando a masa las líneas R/G/B como se observa en la figura siguiente:

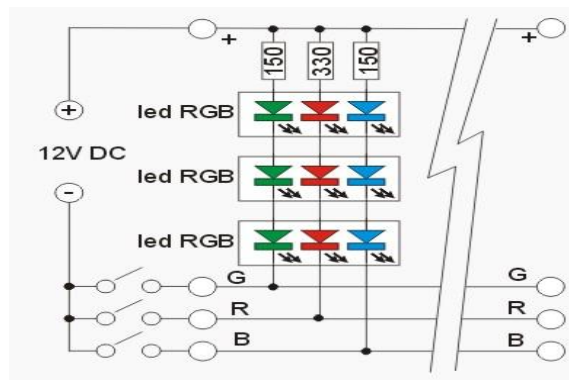


Figura 23: Punto positivo de alimentación en común.

En el ejemplo anterior, para encender los distintos canales se han usado simplemente interruptores. Los interruptores nos permiten 8 combinaciones de colores como visto anteriormente. Para obtener un control más sofisticado de las tiras RGB es necesario usar dispositivos electrónicos. El ejemplo que presento permite de controlar una tira de LEDs a través de un microcontrolador (como por ejemplo el 12F629). [7]

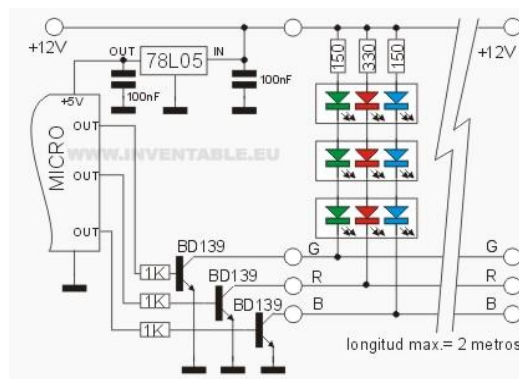


Figura 24: control de una tira led a través de un microcontrolador

Este circuito, usando la técnica de la modulación por ancho de impulso (PWM), nos permite de regular la intensidad de cada canal y, por lo tanto, de obtener una gran variedad de colores y transiciones. El desarrollo del programa necesario para el microcontrolador va más allá del alcance de este artículo, probablemente escribiré otro más adelante sobre este argumento. [7]

Consumos y tipos de fuentes de alimentación

En el elenco siguiente podemos ver los consumos y las fuentes de alimentación necesarias para tiras de distintos tamaños.

- Largo de la tira: 0,5 m
- Cantidad de led: 30 (compuestos por tres leds cada uno)
- Corriente: 600mA
- potencia: 7,2 Watt
- Fuente de alimentación: 12V 1 Amp DC
- Largo de la tira: 1 m
- Cantidad de led: 60 (compuestos por tres leds cada uno)
- Corriente: 1,2 Amp
- Potencia: 14,4 Watt
- Fuente de alimentación: 12V 1,5 Amp DC
- largo de la tira: 2 m
- cantidad de led: 120 (compuestos por tres leds cada uno)
- corriente: 2,4 Amp
- potencia: 28,8 Watt
- fuente de alimentación: 12V 3 Amp DC

4.12 Red LAN

Es una red de área local comúnmente pequeña que conecta a diferentes ordenadores en un área pequeña, la red WIFI está clasificada como una red LAN.

Los elementos que componen una red LAN son: AP, medio de transmisión, servidor, cliente. Hoy en día los teléfonos celulares inteligentes integran la función llamada Hostpot, el teléfono celular Samsun Galaxy J7 integra esta función, lo que le permite comportarse como hotspot,

Un servidor como ARDUINO MKR WIFI1010 puede conectarse a la red WIFI proveída por el celular, al mismo tiempo, lo puede hacer otro celular, todo de forma inalámbrica, así se podría fácilmente conectarse al servidor WEB desarrollado en ARDUINO MKR WIFI1010 mediante otro celular o mediante cualquier computadora portátil, de escritorio o cualquier Tablet. El esquema de la red LAN se puede ver en la Figura 25.

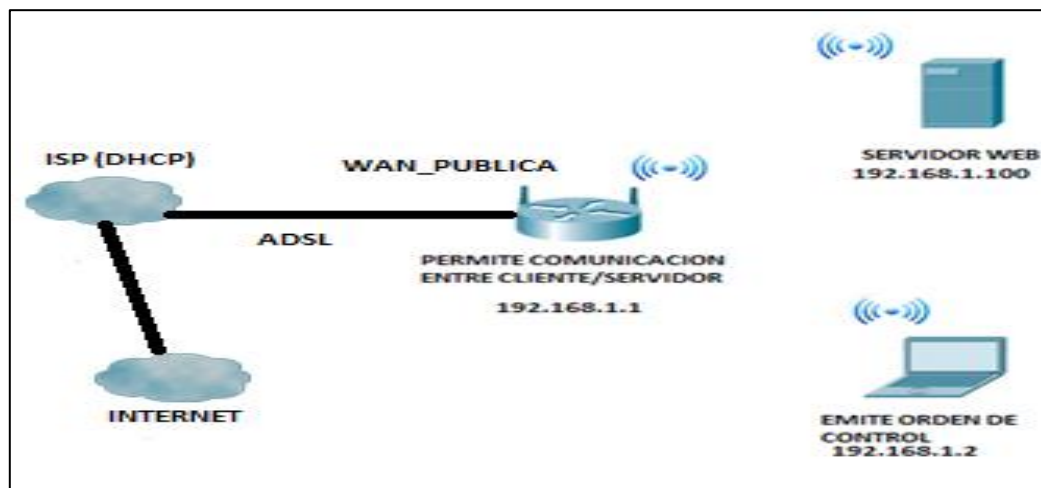


Figura 25: Esquema General de la red LAN Fuente: Propia.

4.13 Servidores Web

Un servidor web es una combinación de hardware y software que permiten el acceso remoto a herramientas o información que generalmente residen en una red de dispositivos. Hablando de un servidor como un equipo, es una computadora que se mantiene a la espera de peticiones por parte de un cliente (un navegador web) y a estas peticiones mediante una página web se exhibirá en el navegador o mostrando su respectivo mensaje si se detectó algún error.

Hablando en términos de software, es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El programa implementa el protocolo HTTP (HyperText Transfer Protocol) que pertenece a la capa de aplicación del modelo OSI.

4.14 Router y Modem

El router y el modem son dispositivos de hardware usados para que los ordenadores puedan tener acceso a internet. Un router se encarga de distribuir la conexión a distintos ordenadores dentro de la misma red local, y enrutar ese tráfico a host que pertenecen a otros segmentos de red, actúa de intermediario entre un conjunto de dispositivos en internet, como un puente entre el ordenador y la red de redes.

El modem es el encargado de transformar las señales analógicas en señales digitales, el medio utilizado para transmitir puede ser el aire o un medio guiado. Por lo tanto, ambos equipos operan en diferentes capas del modelo OSI. El router opera en la capa tres del modelo OSI.

Switch

Es el dispositivo encargado de operar en la capa dos del modelo OSI, básicamente se encarga de distribuir el tráfico usando las direcciones MAC del host de la red LAN, MAN o WAN.

El modem A7600A1 que se ve en la figura, que es proveído por claro, integra en un mismo hardware todas las funciones mencionadas, integra una aplicación para hacer uso del servicio DDNS con el proveedor DYNU, hace NAT, funciona como switch, router y AP.



Figura 26: Modem HG630 VDSL V2

4.15 Wifi

Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica, los dispositivos habilitados con Wifi como ARDUINO, pueden conectarse a internet a través de un punto de acceso de red inalámbrica. Wifi es una marca de Alianza-Wifi, que es una organización que adopta, prueba y certifica los equipos que cumplen con los estándares 802.11 relacionados a redes inalámbricas de área local.

Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n disfrutaron de una aceptación internacional debido a que la banda de 2,4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente .

4.16 Internet de las cosas

La plataforma IOT permite monitorizar los datos de un módulo ARDUINO, de tal manera que se puede almacenar información como también intercambiar información. Cada uno de estos sistemas se trata por separado y se comunican entre ellos a través de protocolos de comunicación. Estos protocolos deberían ser un estándar para que, independientemente de la plataforma, se puedan comunicar. [13]

➤ Dispositivos IOT

MRK WiFi1010



Figura 27: Esquema general para proyectos IoT con Arduino

➤ Plataforma del IOT

Los protocolos de comunicación más utilizados son HTTP, MQTT y CoAP. Además, para comunicarnos, existen diferentes redes como LoRa o SigFox. Son redes WAN para el IoT y una alternativa a los sistemas tradicionales de comunicación. Smartnest es una plataforma IOT que permite entablar MKR módulo ARDUINO MKRWIFI 1010 con el servidor de AMAZON ALEXA.

➤ API de acceso a los datos

Tener una **API** (Application Programming Interface) permitirá consultar, modificar y borrar la información desde otros dispositivos.

Al final tenemos que entender que es **una** capa de comunicación estándar para conectarnos a los datos. Existen diferentes protocolos y estándares. El más utilizado sería a través de [servicios web RESTful](#). El acceso a dicha API dependerá del software desde donde se realice la conexión. Si se trata de una aplicación web, frameworks como jQuery, AngularJS o React, facilitan esta tarea enormemente. [13]

V. ANALISIS Y PRESENTACION DEL RESULTADO

El presente apartado comienza con una **descripción del proceso** de desarrollo que nos llevó al resultado final del prototipo de Smart Lighting System propuesto.

Luego se realiza una descripción de cada elemento utilizado para el desarrollo entre dispositivos hardware, servidores y aplicaciones, incluso el mismo código desarrollado que permitió la conexión de todo el sistema.

Además, se exponen los resultados del prototipo desarrollado con sus respectivas pruebas realizadas, las cuales fueron documentadas.

Finalmente se realiza un análisis de resultados, porque se eligió trabajar con la tecnología Arduino

a. Descripción del proceso

Primeramente, se realizó un estudio necesario con el fin de determinar la tecnología y componentes específicos para cumplir con los requerimientos de dicho sistema (dispositivos -hardware), incluso se inició trabajando con Raspberry Pi.

Seguido se elaboró un programa a fin de establecer una comunicación lógica entre Protocolos red (UDP/TCP-IP) & Hardware. La base que fundamenta este proyecto es la relación que existe entre conexiones inalámbricas, principalmente la fase más importante fue cuando se integró el sistema de comandos a una terminal de LINUX en la tarjeta Arduino, desde ahí es donde se creó las líneas enteras de códigos y comandos que permitió establecer una conexión entre servidores y cliente, Que de una forma brindan una lógica de ON/OFF mediante el uso de PWM en el código del proyecto desarrollado.

Luego se configuro el hardware Amazon Echo para su mejora y adaptación y se valió de varios algoritmos, como sigue:

Estado; Activo, accionamiento por voz, recepción por servidor, recepción y distribución de señales lógicas, atravez del perfil gestionado del perfil de iluminación.

Seguidamente, una vez desarrollado el programa (Código) y hardware, se elaboró el prototipo, que cumplió con los requisitos para el servicio que brindará la Smart lighting System controlado por Asistente virtual para el confort visual basado en IoT.

Finalmente se realizaron pruebas de ensayo con el prototipo propuesto para asegurarnos que cumplía con los requisitos que esperaban.

b. Descripción de los elementos involucrados en el sistema

-Integración del dispositivo Amazon Echo

Alexa, es el servicio de voz que maneja el Amazon Echo, provee skills que permiten que el usuario interactúe con dispositivos de una manera más intuitiva utilizando la voz. Las habilidades incluidas oficialmente son la reproducción de música, respuesta de preguntas generales, seteo de alarmas y temporizadores entre otros.

Alexa Skills Kit es una colección de APIs (Interfaces de programación de aplicaciones), herramientas, documentación, y ejemplos, que facilitan la creación de nuevas habilidades a Alexa. Todas las habilidades de Alexa, corren en la nube. Esto permite que cualquier usuario tenga acceso a estas nuevas capacidades, solo interactuando con Alexa, haciéndole preguntas, o realizando pedidos.

Cuando un usuario le realiza una pregunta, o le indica a Alexa que haga algo, ese pedido es enviado a un servicio en la nube. Este determina que está queriendo hacer el usuario, y vincula el pedido con el servicio correspondiente que provee la lógica, y espera la respuesta. Para que Alexa pueda interactuar con las habilidades creadas, las mismas deben estar desplegadas como un servicio web.

Ejemplo de cómo interactúan los usuarios con Alexa (Algoritmo)

Dado que actualmente el servicio provisto por Alexa soporta solo el lenguaje inglés, se introducirán los ejemplos de una manera simple en este idioma. Los usuarios interactúan con las habilidades desarrolladas para comunicar Alexa con Intellihome realizando una pregunta o indicando un comando en lenguaje natural. El usuario indica la frase soportada por ejemplo "Ask", el nombre de invocación definido para nuestra habilidad, y el pedido o pregunta a realizar. [Usuario: "Alexa, ask Intellihome to turn on the kitchen light."](#)

El ejemplo que refleja lo desarrollado, se descompone de la siguiente forma

- “Alexa” es la palabra que inicia el reconocimiento en el dispositivo Amazon Echo, la cual permite iniciar una conversación con el servicio.
- “Ask...to” es una de las frases aceptadas para solicitar la realización de una habilidad en particular.
- “Intellihome” es el nombre que identifica la habilidad que el usuario quiere invocar. El nombre es utilizado por Alexa para definir a qué servicio se debe derivar la petición realizada por el usuario
- “turn on the kitchen light” es el pedido específico, la pregunta o el comando que se asocia a un intent particular para este servicio
- En este contexto un intent representan una acción de alto nivel que cumple con el pedido del usuario, Estos pueden tener argumentos como en este caso lo son “Kitchen y “Light”
- Definir las frases que el usuario utilizara para interactuar y asociarlos a los *intent* es parte de la interfaz de voz de una habilidad de Alexa.

El diagrama siguiente demuestra a grandes rasgos cómo es la interacción con el servicio Alexa. Este modelo de conversación se repite hasta que el usuario obtiene lo que desea, independientemente de cómo se comienza la interacción con el servicio, la conversación continúa hasta concretar un objetivo.

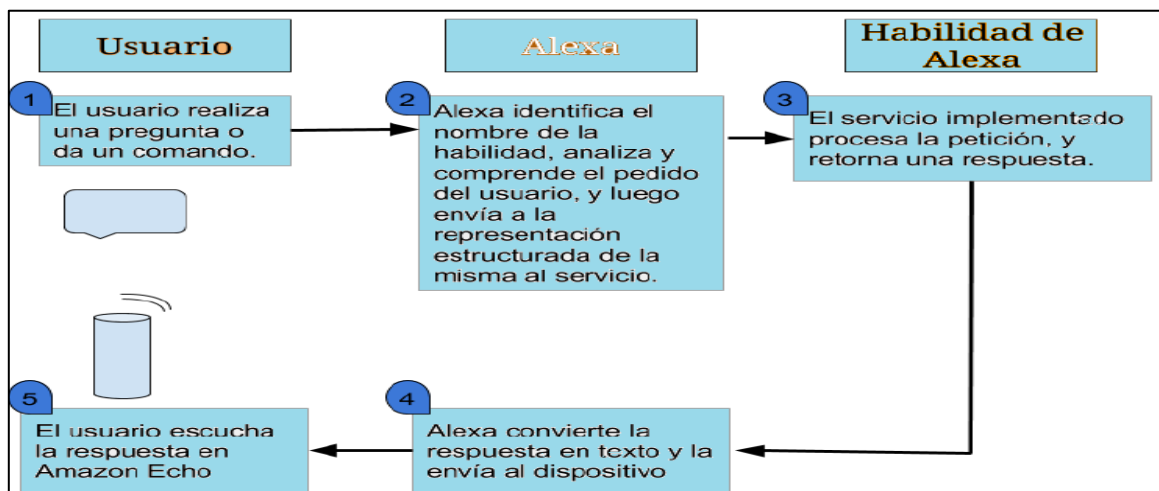


Figura 28: Interacción de Amazon Echo con el servicio Alexa.

Cada vez que se envía una respuesta, se puede incluir el estado de la conversación, o si debe terminar o no la misma. Si el servicio de Intellihome necesita más información del usuario para ejecutar una acción, establece que debe continuar.

Si el usuario no responde, o responde con una sentencia que no está asociada a un intent definido en la interfaz de voz, Alexa pide nuevamente el comando.

Para que Alexa pueda proveer todas las interacciones mencionadas, las mismas fueron definidas en el servicio desarrollado.

Siguiendo con el ejemplo inicial, una conversación corta sería:

Usuario: "Alexa, ask Intellihome to turn on the kitchen light."

Alexa: "The kitchen light has been turned on."

<fin>

Una conversación más larga sería:

Usuario: "Alexa, ask Intellihome"

Alexa: "Which rooms do you want me to help you with?"

Usuario: "kitchen"

Alexa: "What do you want to do in the kitchen"

Usuario: "turn the lights on"

Alexa: "The kitchen light has been turned on."

<fin>

Como se puede observar al diseñar la interacción con el usuario, se pueden establecer diferentes conversaciones que lleven a un mismo resultado, y así poder detectar mayor cantidad de expresiones del mismo.

c. Diseño de la interfaz de voz.

Para definir una interfaz de voz, se debe especificar una asociación entre los comandos del usuario y los intents que el servicio va a manejar. Para definir esa asociación, se deben incluir dos recursos principales:

1. Un esquema de Intents: Es una estructura JSON que declara el set de intents que el servicio va a aceptar y procesar.

2. Las frases de entrada:

- Enunciados: Un archivo de texto estructurado que conecta los intents con las frases que utilizara el usuario, conteniendo la mayor cantidad de frases representativas posibles.
- Valores personalizados: Una lista representativa de valores para ítems específicos usados por la habilidad, y referenciados por los intents.

ESQUEMA GENERAL DEL PROTOTIPO

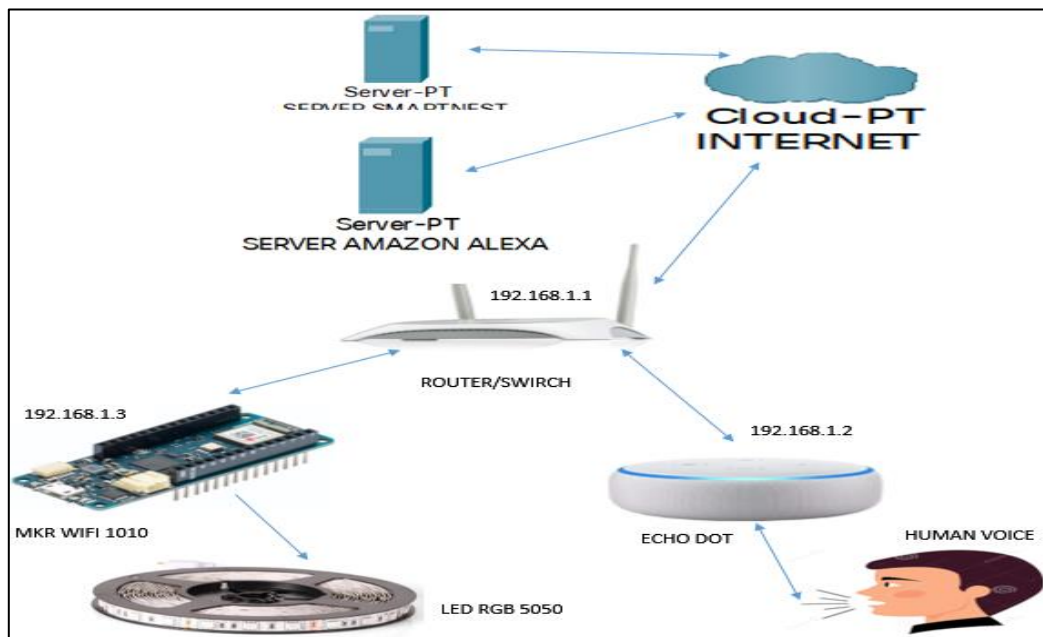


Figura 29: Elementos involucrados en el proceso de control de la Smart light system

Fuente: Propia

d. Configuración de Smartnest

En Smartnest, se creó un dispositivo con un ID llamado RGB, este ID debe usarse como la primera parte del tema.

Ese ID lo usara el microcontrolador para poder comunicarse con el bróker y es la primera parte del mensaje enviando [11].

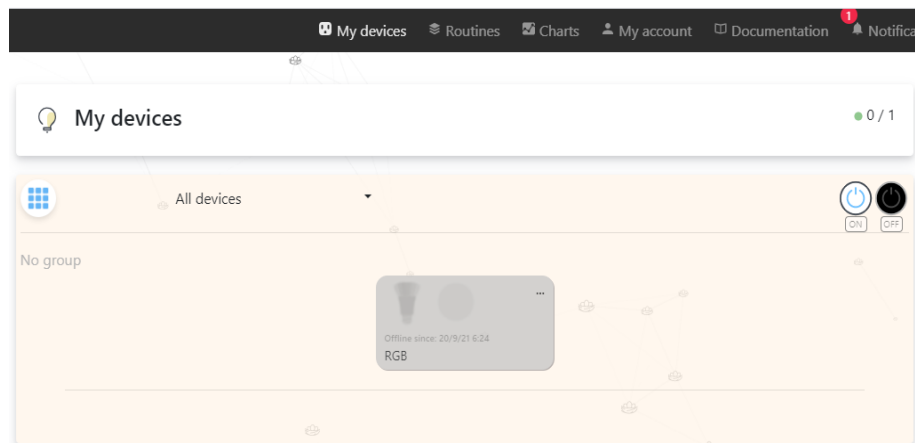


Figura 30: Dispositivo creado en Smartnest Fuente propia

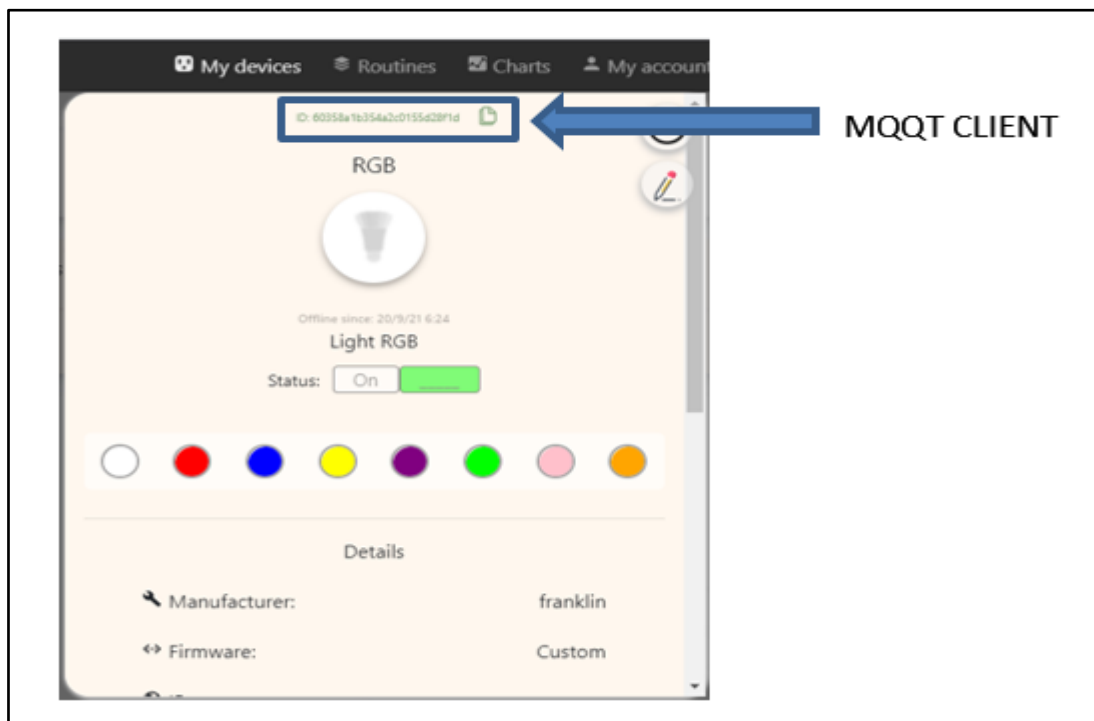
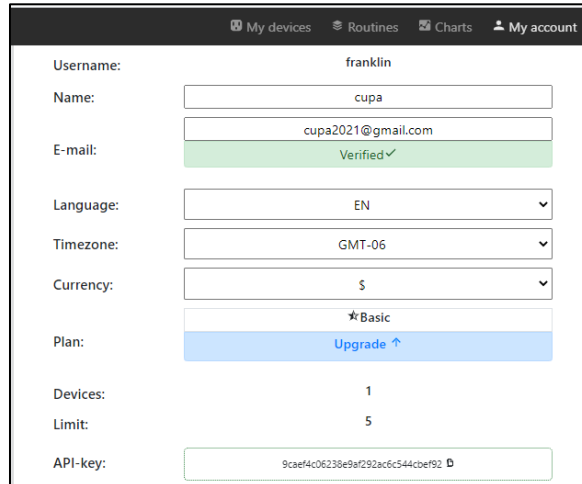


Figura 31: ID creado en Smartnest Fuente propia



Fuente 32: Fuente propia usuario creado en el servidor Smartnest

e. Configuración de Amazon eco

Dispositivos vinculados con Amazon Alexa (Parlante Echo Dot 4ta generación, luces RGB)



Fuente 33: Dispositivos controlados desde Amazon Alexa

f. Perfiles de iluminación

Descripción grafica de las rutinas creadas con Amazon echo para representar los diferentes perfiles de iluminación

| | |
|--------------------|--------------------|
| MIS RUTINAS | |
| Me siento alegre | Azul cielo |
| | |
| Me siento relajado | Anaranjado |
| | |
| Me siento cansada | Rojo |
| | |
| Apaga las luces | Alexa voy a dormir |

Figura 34: rutinas creadas para los diferentes perfiles de iluminación

Fuente: propia

g. Creación del código ejecutado en MRKWIFI 1010

En la primera parte del código se declaró las librerías (PubSubClient.h) a utilizar, luego se procedió a declarar las variables globales como el nombre de la red wifi, la contraseña de la red wifi, el nombre de dominio del brokers, el puerto usado por el brokers, el usuario Y contraseña utilizado para establecer la comunicación mediante el protocolo MQTT. Luego se procedió a la asignación de variables (tipo entero) para cada uno de los tres puertos de salida. Además, se procedió a declarar las variables encargadas de la configuración de los colores de las luces led RGB.

A continuación, se declararon el menú de funciones a utilizarse para la realización de los diferentes procesos (Conexión del microcontrolador a una red wifi , la conexión del microcontrolador con el servidor smartnest , la función que permite almacenar la cadena de caracteres para el control on/off y la implementación del PWM , la función que realiza el llamado a la transmisión y recepción de los caracteres de control , la función que permite configurar los diferentes perfiles de iluminación , las funciones de encendido y apagado de las luminarias).

A partir de acá se procedió a la configuración de los procesos que se ejecutan una vez (configuración de los puertos de salida, configuración del puerto serial, el llamado a la conexión a la red wifi y el llamado a la conexión entre el microcontrolador y el servidor smartnest).

Se prosigue con la ejecución de los procesos cíclicos (revisión de la conexión entre el microcontrolador y el brokers.)

Código en lenguaje de programación basado en processing y C++

Lo encontramos en el Anexo A

A continuación, se realizó la argumentación de las funciones que se declararon previamente.

Secuencia de los Diagramas de flujos

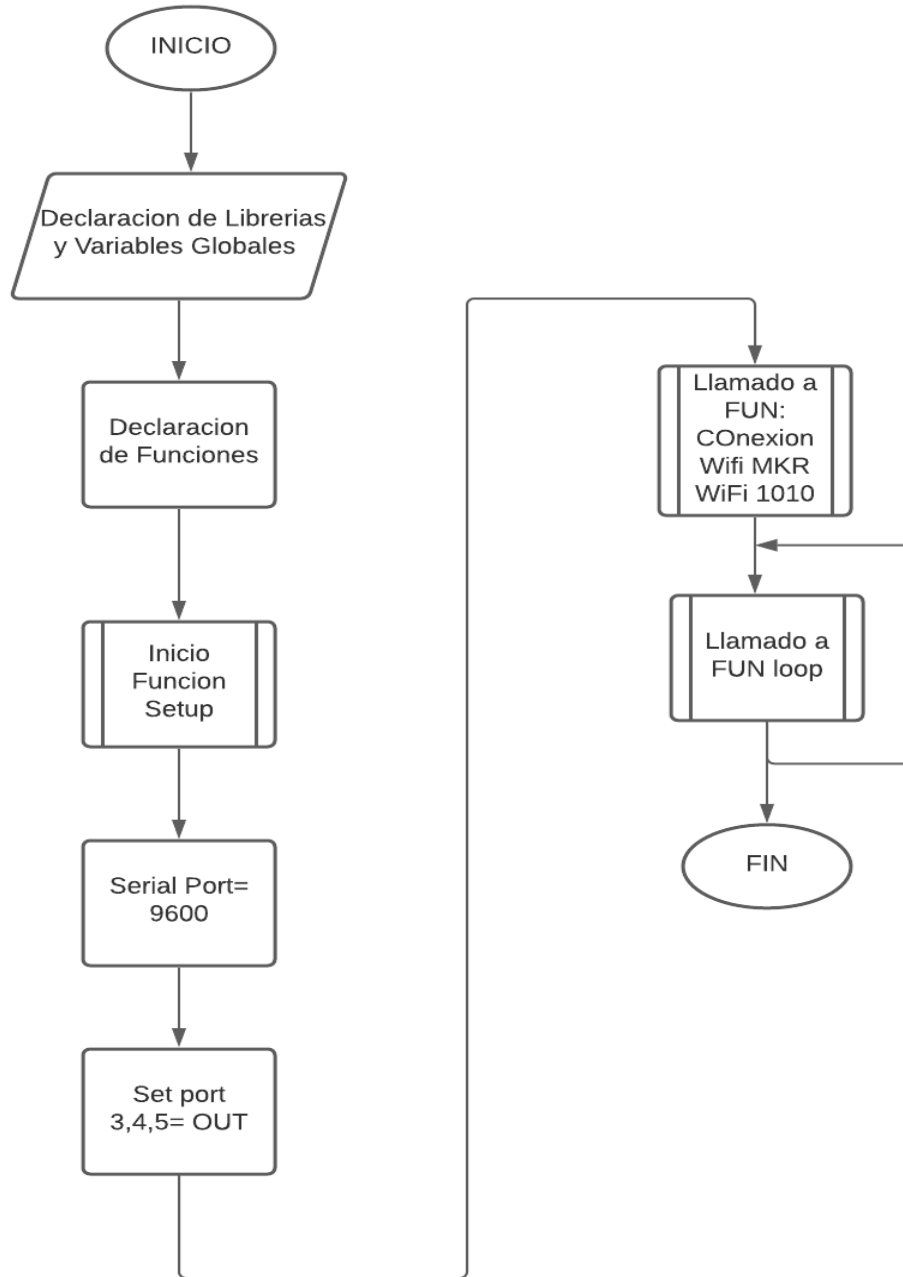


Figura 35: Algoritmo implementado en MRK WIFI 1010

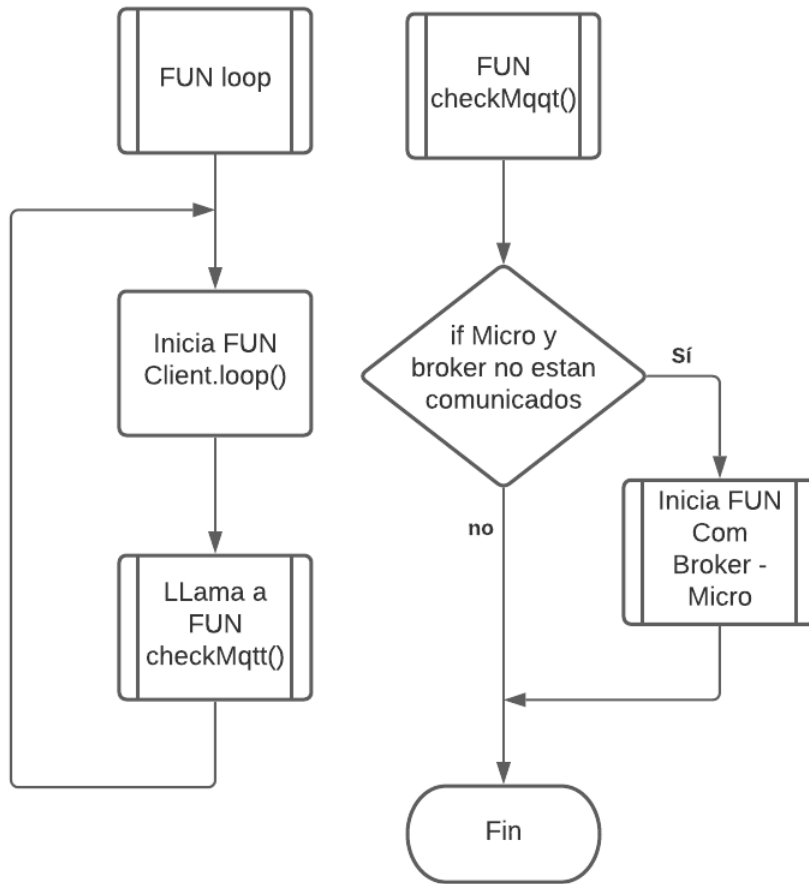


Figura 36: Funciones loop y checkMqtt()

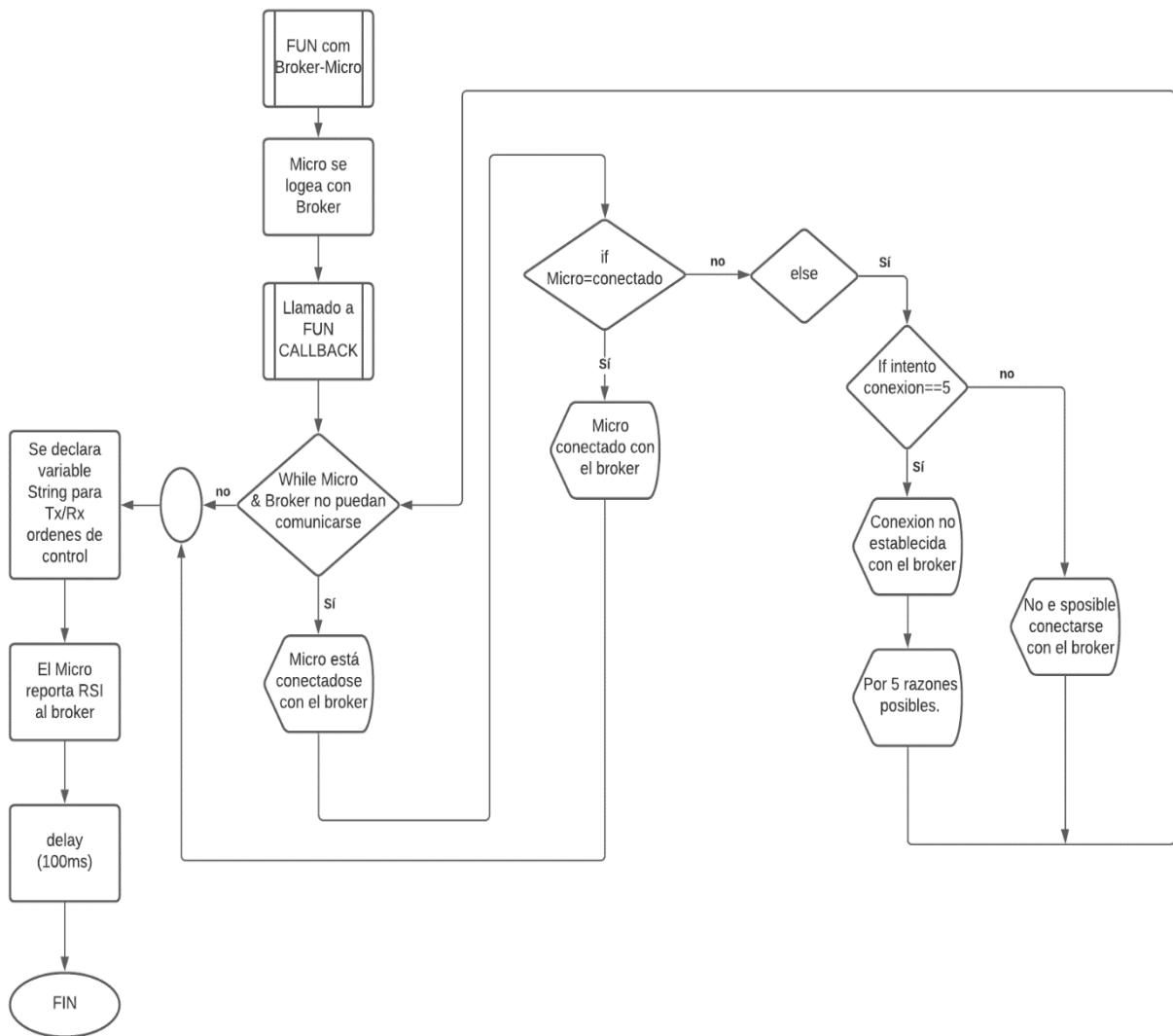


Figura 37: Función comunicación bróker-micro usando protocolo MQTT.

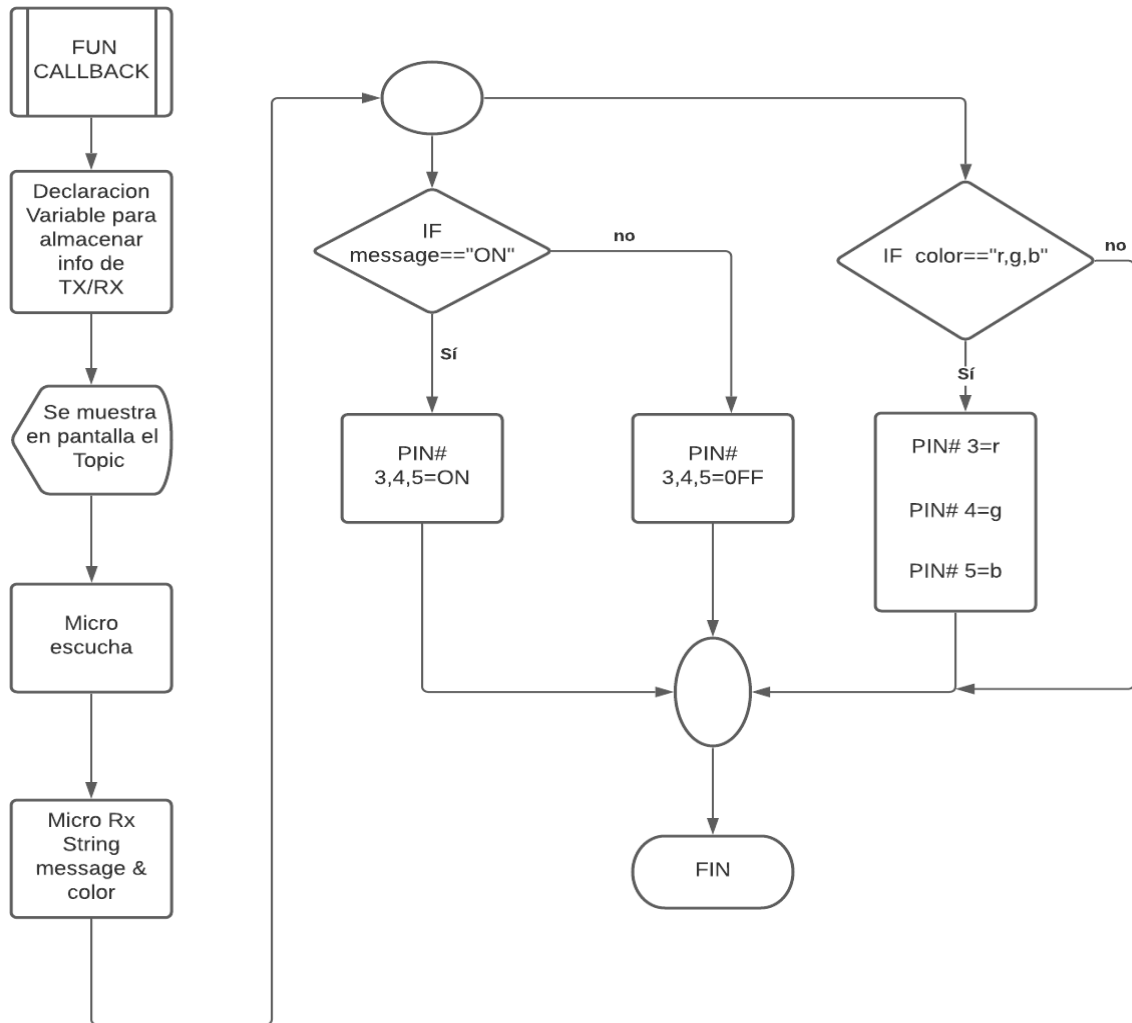


Figura 38: Función Callback, usada para el envío y recepción de mensajes usando protocolo MQTT

h. Realización montaje y pruebas de funcionamiento

Primera realización de pruebas.



Figura 39: Montaje y pruebas del prototipo

Se realizó el montaje de todos los circuitos con las luminarias led en el edificio Rigoberto López Pérez de la UNI, en el ALA B-I-2. Dichas pruebas se realizaron en el mes de agosto del 2021 con resultados satisfactorios como se muestra en la imagen. En este proceso logramos adaptar los componentes electrónicos en el case (PVC) sin tener ningún problema o interferencia y así de esta manera obtener una mejor presentación del prototipo.

Segunda realización de pruebas.

Pruebas en pasillo de usos múltiples de la FEC

El día 7 de diciembre del 2021 logramos pruebas exitosas de funcionamiento de las 4 rutinas declarada en los perfiles de iluminación los cuales representan el estado de ánimo de la persona o cliente. Estas rutinas y sus colores de iluminación fueron seleccionadas en base a la psicología de colores del fabricante Philips, incluso dicho fabricante ya comercializa un dispositivo inteligente para el control de iluminación llamada Hue Bridge Philips.

A continuación, vamos a describir las pruebas de cada perfil de iluminación declaradas anteriormente. Son 4 rutinas con sus respectivos perfiles de iluminación, donde se indicará la rutina y el color seleccionado, mismo que podrán constatar en las siguientes figuras.

Primer perfil: Alexa, Me siento alegre, al momento de nosotros ejercer dicho comando de voz tendríamos como resultado una tonalidad de color azul cielo reflejada en nuestra RGB



Figura 40: Perfil de iluminación por comando de voz me siento alegre.

Segundo perfil: Alexa, Me siento relajado, al momento de nosotros ejercer dicho comando de voz tendríamos como resultado una tonalidad de color naranja reflejada en nuestra RGB, esta sería nuestra primera rutina.



Figura 41: Perfil de iluminación por comando de voz me siento relajado.

Tercer perfil: Alexa, Me siento cansada, al momento de nosotros ejercer dicho comando de voz tendríamos como resultado una tonalidad de color rojo reflejada en nuestra RGB.



Figura 42: Perfil de iluminación por comando de voz me siento cansada.

Cuarto perfil: Alexa, Voy a dormir, al momento que nosotros ejerzamos dicho comando de voz nuestra RGB quedaran en estado Down (apagadas).



Figura 43: Perfil de iluminación por comando de voz, Alexa voy a dormir.

i. ARDUINO VS RASPBERRY PI

Ambas placas son bastante similares en funcionalidades y composición, ambas tienen memoria, las dos presentan entradas y salidas, con las dos se pueden sensorizar o actuar sobre elementos...pero la manera en la que hacen estas tareas es donde radica su mayor diferencia.

Las dos placas han proporcionado plataformas **open software** y **open hardware**, sobre las que están proliferando multitud de ideas y **proyectos IoT** que permiten tomar datos del entorno y actuar en consecuencia.

Pero, cuando se trata de elegir una en particular, puede existir dudas, ya que es difícil decidir **cuál es la mejor** opción para un proyecto en particular.

Algunos Ejemplo aparte de la programación:

- Precios
- Disponibilidad en tu mercado local
- Complejidad del proyecto

Resumen y análisis de las principales características a resaltar desde la parte de la programación:

Las dos placas son versátiles y son buenas para algunas cosas específicas. Veamos las diferencias entre Arduino y Raspberry Pi.

- Un **Arduino** es un **microcontrolador**. Un microcontrolador es un ordenador simple que puede ejecutar una sola tarea de forma repetitiva. Es muy fácil de usar.
- Ejecutar pequeñas aplicaciones desarrolladas en **programación C++**
- Excelente para **controlar pequeños dispositivos** como motores, luces, sensores, etc.
- Indicada para '**prototipos**' electrónicos y cómodo, conveniente para hablar con otra máquina y perfecto para ejecutar tareas sencillas y repetitivas

-
- Una **Raspberry Pi** es un **mini PC o mini ordenador completo** que necesita un sistema operativo como Linux y que puede llevar a cabo múltiples tareas, Es un poco más complejo de usar.
 - Ejecuta un **sistema operativo** a escala completa para tareas múltiples.
 - **Se pueden implementar programas** para varios proyectos.
 - Excelente para actuar como un **servidor** y para comunicarse con otros dispositivos informáticos.
 - **Es** mejor cuando se necesita un **ordenador completo**: conducir un robot más complicado, realizar múltiples tareas, realizar cálculos intensos (como con Bitcoin, aunque esto pondrá la placa a su máximo rendimiento).


Con todo lo anterior, queda claro que la elección de una u otra placa se deberá más al deseo de cada uno, si necesitas potencia de cálculo, lo mejor sería Raspberry Pi, pero si necesitas actuar como interfaz con otros sistemas y no quieres preocuparte por sistemas operativos, entonces la mejor opción es **Arduino**.

En cuanto al soporte y disponibilidad en el mercado

En este aspecto, ambas placas son muy parecidas, la comunidad de aficionados y profesionales que da soporte a estas placas es enorme, quizá un poco más para Arduino, pero posiblemente se debe a que apareció mucho antes en el mercado que la Raspberry Pi.

En Nicaragua es más común la comercialización de Arduino y es más fácil encontrar en el mercado local, durante el proceso de nuestro proyecto pasamos esa experiencia y más aun con la pandemia de COVID se nos hizo muy difícil conseguir la placa Raspberry Pi y fue una de las razones por que **elegimos Arduino**.

j. Costo de la implementación del prototipo Smart lighting System



UNIVERSIDAD NACIONAL DE INGENIERÍA

Tabla de costo proyecto Smart lighting System basado en IoT

| ITEM | CANTIDAD | PRODUCTO | PRECIO USD | PRECIO TOTAL USD |
|------|----------|--------------------------|------------|------------------|
| 1 | 1 | MKR WIFI 1010 | 45.00 | 45 |
| 2 | 1 | LUCES LED RGB 5050 | 11.50 | 11.50 |
| 3 | 1 | CAJA REGISTRO 4x4 | 10 | 10 |
| 4 | 1 | CABLES (MACHO/HEMBRA) | 10 | 10 |
| 5 | 1 | TARJETA PREPERFORADA | 3 | 3 |
| 6 | 1 | OTROS | 20 | 20 |
| 7 | 1 | AMAZON ECHO | 80 | 80 |
| 8 | 1 | N-MOS | 6 | 6 |
| 9 | - | MANO DE OBRA | 100 | 100 |
| | | | TOTAL | 285.5 |

Figura 44: Presupuesto del diseño para la implementación

VI. Conclusión

El presente proyecto tesis ha logrado cumplir con los alcances propuestos en el inicio del mismo. Debido a las pruebas realizadas, ha sido posible comprobar y verificar el correcto funcionamiento del sistema, basado en desarrollo del IDE de ARDUINO con el fin de controlar el Smart lighting System.

Tanto la tecnología Amazon Echo, como Arduino, permiten la implementación de un sistema domótico , con el fin de mejorar su calidad de vida.

La alta funcionalidad de los dispositivos de la tecnología Arduino nos ha permitido emular el sistema domótico de forma completa y satisfactoria. Las diferentes interfaces de control y monitoreo implementadas cubren las exigencias del usuario brindando seguridad y confort.

El prototipo del sistema domótico implementado con la tecnología arduino ha posibilitado la autonomía del usuario, ofreciendo confort, accesibilidad, movilidad, seguridad y comunicación con el sistema. Estos sistemas domóticos permiten controlar una vivienda a través de diferentes interfaces como por ejemplo comando de voz para controlar las luces de la habitación.

Se ha logrado Implementar el algoritmo haciendo uso del lenguaje de programación para crear un cliente a fin de establecer una comunicación lógica entre Protocolos Web & Hardware, basado en desarrollo del IDE de ARDUINO con el fin de controlar el Smart lighting System, como resultado un algoritmo funcional durante las pruebas.

Se permitio Integrar tecnologías para el reconocimiento del lenguaje hablado permitiendo de esta manera que el usuario puede usar un comando de voz para controlar las luces de la habitación.

Se Crearon diferentes perfiles de iluminación en un ambiente inteligente basado en el confort.

Se Realizo el montaje del circuito electrónico que permitio llevar a cabo el control del estado operativo de las luminarias , como resultado el prototipo.

VII. Recomendaciones

- En lugar de crear dispositivos en la plataforma smartnest, sería opcional crearlo directamente en el IDE de Arduino. Con el fin, por un lado, se incrementa la complejidad del código, a la vez que se reducen una plataforma ya que los dispositivos podrían ser creados en el IDE de Arduino y visualizados directamente en la plataforma de Amazon Echo.

VIII. Bibliografía

1. J. Ortega & F. Rojas & E. Vargas. Prototipo de control a través de una app para una casa inteligente. Tesis.UNAM, Cd México, 2016.
2. K. Gisele. Integración de dispositivos en una plataforma de vida cotidiana asistida por computadora. Tesis. Universidad Nacional del Centro de la Provincia de Buenos Aires. Junio 2016.
3. T. Kleinberger y M. Beker y E, Ras. Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces. Beijing, China, July.2007
4. Webedia Brand Services (2013). [Internet]. Disponible:[https://philipshue.xatakahome.com/como-iluminacion-puede ayudarnos-a-concentrarnos-todo-contrario](https://philipshue.xatakahome.com/como-iluminacion-puede-ayudarnos-a-concentrarnos-todo-contrario).
5. Y. Fernandez. (2021, enero 21). Alexa-skills-kit [Online]. Disponible: <https://www.xataka.com/basics/que-alexa-que-puedes-hacer-que-dispositivos-compatibles#:~:text=Su%20nombre%20fue%20elegido%20por,Google%20Assistant%2C%20Siri%20y%20Cortana>.
6. Integración con Alexa.(s.f).[Internet]. Disponible:<https://documentacion.smartnest.cz/integracion-con-alexa>.
7. Creative Commons BY_NC_ND.(2011). [Internet]. Disponible <https://www.inventable.eu/2011/04/27/anatomia-de-una-leds-string-rgb/>
8. E. Sánchez, Diseño de un sistema de control domótico basado en la plataforma Arduino Tesis (para obtener el título de Ingeniería Técnica en Informática de Sistemas). Universitat Politècnica de València. Diciembre 2012.
9. Guia MQTT.(s.f).[Internet]. Disponible:<https://documentacion.smartnest.cz/guia-mqtt>,
10. High-Power Dual MOSFET Switch Module. (s.f).[Internet]. Disponible:<https://protosupplies.com/product/high-power-dual-mosfet-switch-module/>

-
11. Castilla la Mancha. Clase de tecnologías creativas. Arduino Verklad (2014) [Internet]. Disponible:(<https://pdfcoffee.com/manual-arduino-muy-completo-4-pdf-free.html>).
12. Arduino MKR WiFi 1010. (s.f).[Internet].Disponible: <https://www.330ohms.com/collections/arduino/products/arduino-mkr-wifi-1010>
13. Proyectos IoT con Arduino. (s. f). [Internet]. Disponible: https://programarfacil.com/podcast/proyectos-iot-con-arduino/#Esquema_general_para_proyectos_IoT_con_Arduino
14. *TCP, Efecto del color de la luz en los usuarios [tabla], generación propia con información del artículo “The Psychological Impact of Light and Color” (El Impacto Psicológico de la Luz y el Color).*2020 Disponible en: <https://31ie0amlw353ita8n1t3t6v1-wpengine.netdna-ssl.com/wp-content/uploads/2017/12/Psychological-Impact-of-Light-and-Color.pdf>

IX. ANEXOS

i. Anexo A: Código en lenguaje de programación basado en processing y C++

```
#include <SPI.h>//LIBRERIA QUE SE USA EN CONJUNTO CON WiFININA
#include <WiFININA.h>//LIBRERIA QUE SE USA EN CONJUNTO CON SPI
#include <PubSubClient.h>// Librería usada para configurar el protocolo MQTT
#include <WiFiClient.h>//Librería para configurar el WIFI

#define SSID_NAME "CLARO_4DF6F8"           // Nombre de la red WIFI
#define SSID_PASSWORD "Claro.1111"        // Contraseña de la red WIFI
#define MQTT_BROKER "smartnest.cz"        // Dominio usado para conectarse al
broker server
#define MQTT_PORT 1883                    // puerto del servidor broker
#define MQTT_USERNAME "franklin"          // Usuario creado en el server
#define MQTT_PASSWORD "9caef4c06238e9af292ac6c544cbef92" //
Contraseña del usuario configurado en SERVER Smartnest (or API key)
#define MQTT_CLIENT "60358a1b354a2c0155d28f1d" // Dispositivo RGB
creado previamente en el server Smartnest
#define FIRMWARE_VERSION "Example-lightRgb" //

WiFiClient espClient;
PubSubClient client(espClient);
int lightPin3 = 3;//Se asocia la variable de tipo entero lightPin3 al pin 3 del
microcontrolador.
int lightPin4 = 4;//Se asocia la variable de tipo entero lightPin4 al pin 4 del
microcontrolador.
int lightPin5 = 5;//Se asocia la variable de tipo entero lightPin4 al pin 4 del
microcontrolador.

int brightness = 0;//
int red = 1;// se declara la variable red en 0, que permite controlar el ancho de pulso de
la salida del pin#3.
int green = 1;// se declara la variable green en 0, que permite controlar el ancho de pulso
de la salida del pin#4.
int blue = 1;// se declara la variable green en 0, que permite controlar el ancho de pulso
de la salida del pin#5.
```

```

void startWifi();//Se declara la función a usarse para que el microcontrolador se conecte
a una red WIFI.
void startMqtt();//Se declara la función a usarse para que el microcontrolador se conecte
al broker smartnest.
void checkMqtt();//Se declara la función que
int splitTopic(char* topic, char* tokens[], int tokensNumber);//
void callback(char* topic, byte* payload, unsigned int length);// se declara la función que
void sendToBroker(char* topic, char* message);// funcion que permite enviar al broker
el titulo del topic al que quiere suscribirse el microcontrolador
void setBrightness(int value);//
int splitColor(char* color, char* tokensC[]);
void setColor(int r, int g, int b);//
void turnOff();//Se declara la funcion que apaga las luminaria RGB.
void turnOn();//Se declara la funcion que enciende las luminaria RGB.

void setup() {
  pinMode(lightPin3, OUTPUT);//Se declara el pin#3 como salida
  pinMode(lightPin4, OUTPUT);//Se declara el pin#4 como salida
  pinMode(lightPin5, OUTPUT);//Se declara el pin#5 como salida
  Serial.begin(115200);//inicializa el puerto serial del microcontrolador en 115200
  baudios
  startWifi();//se manda a ejecutar la función startWifi(), con el fin de que el micro se
  conecte a la red WIFI
  startMqtt();//Se inicializa la conexión del micro con el broker smartnest
}
void loop() {
  client.loop();
  checkMqtt();//Se revisa el estado de la conexión del micro con el broker smartnest
}
void callback(char* topic, byte* payload, unsigned int length) { //Un nuevo mensaje ha
  sido recibido
  //void callback(const char[] topic, byte* payload, unsigned int length)
  Serial.print("Topic:");//Imprime en pantalla la la palabra Topic.
  Serial.println(topic);//Se imprime s e presenta en pantalla el tema de conversación.
  int tokensNumber = 10; // se declara la variable local tokensNumber, es del tipo entero
  y se iguala a 10.
  char* tokens[tokensNumber];//Se declara la constante local tokens, para contener una
  cadena de caracteres, la longitud será
  // el valor de la variable tokensNumber

```

char message[length + 1]; //Se declara la variable message para contener una cadena de caracteres, cuya longitud será igual: Al valor que contenga //la variable length y se le suma 1 ya que al final del mensaje que envía en server se le agrega un 0.

splitTopic(topic, tokens, tokensNumber); //Se hace uso de un constructor llamado splitTopic, se hace uso de las variables de la clase callback.

sprintf(message, "%c", (char)payload[0]); //Se captura el mensaje quitándole el valor nulo que se le agrega al final.

for (int i = 1; i < length; i++) { // se hace una captura de la cadena aleatoria que esta enviando el server y se almacena en la variable message.

 sprintf(message, "%s%c", message, (char)payload[i]);

}

Serial.print("Message:"); //Imprime en pantalla la palabra Message:

Serial.println(message); //Se captura el contenido del mensaje.

//-----ACTIONS HERE-----

if (strcmp(tokens[1], "directive") == 0 && strcmp(tokens[2], "powerState") == 0) {

 if (strcmp(message, "ON") == 0) {

 turnOn();

 }

 else if (strcmp(message, "OFF") == 0) {

 turnOff();

 }

}

else if (strcmp(tokens[1], "directive") == 0 && strcmp(tokens[2], "percentage") == 0) {

 int val = atoi(message);

 //Serial.println(val);

 if (val >= 0 && val <= 100) {

 setBrightness(val);

 }

}

else if (strcmp(tokens[1], "directive") == 0 && strcmp(tokens[2], "color") == 0) {

 char* tokensC[3];

 splitColor(message, tokensC);

 int r = atoi(tokensC[0] + 4);

 int g = atoi(tokensC[1]);

 int b = atoi(tokensC[2]);

 setColor(r, g, b); // se manda a la función los valores de r,g,b provenientes de

smartnest

}

```

}

void startWifi() { // el micro se conecta a una red WIFI
  // WiFi.mode(WIFI_STA);
  WiFi.begin(SSID_NAME, SSID_PASSWORD);
  Serial.println("Connecting ...");
  int attempts = 0;
  while (WiFi.status() != WL_CONNECTED && attempts < 10) {
    attempts++;
    delay(500);
    Serial.print(".");
  }
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println('\n');
    Serial.print("Connected to ");
    Serial.println(WiFi.SSID());
    Serial.print("IP address:\t");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println('\n');
    Serial.println("I could not connect to the wifi network after 10 attempts
\n");//Después de 10 intentos no se ha podido conectar a internet.
  }
  delay(500);
}

void startMqtt() { //El micro establece conexión con el server smartnest.
  client.setServer(MQTT_BROKER, MQTT_PORT); // se envía dominio y puerto para
conectarse al broker.
  client.setCallback(callback); //
  while (!client.connected()) { //mientras cliente no esté conectado
    Serial.println("Connecting to MQTT..."); //Imprima: conectandose a MQTT
    if (client.connect(MQTT_CLIENT, MQTT_USERNAME, MQTT_PASSWORD)) { //SI el
micro ya se conectó
      Serial.println("connected"); //imprime la palabra conectado //Imprima en pantalla:
COnectado.
    } else {
      if (client.state() == 5) { //Si el cliente a intentado conectarse por mas de 5 veces.
"client.state() es una función de la biblioteca xxxxxx."
      Serial.println("Connection not allowed by broker, possible reasons:"); //Se informa
que la conexión no es permitida por el broker por las siguientes posibles razones:

```

```

    Serial.println("- Device is already online. Wait some seconds until it appears offline
for the broker");//El dispositivo ya esta en linea, espere unos minutos par que el
dispositivo
    // aparezca fuera de linea.
    Serial.println("- Wrong Username or password. Check credentials");//Error de
usurario oocntaseña.
    Serial.println("- Client Id does not belong to this username, verify ClientId");//El ID
del cliente no es usuario, verifique cliente ID.
    } else {
        Serial.println("Not possible to connect to Broker Error code:");//Imprime en pantalla
que no es posible conectarse con el broker por el codigo de error.
        // coenctarse con el broker.
        Serial.print(client.state());// Aqui explica la razon por la cual no ha podido
conectarse.
    }
    delay(0x7530);// se espera aproximadamente 0x750 milisegundos
}
}
char subscribeTopic[100];//Se declara la variable del tipo String llamada subscribeTopic
de longitud igual a 100 indices.
sprintf(subscribeTopic, "%s/#", MQTT_CLIENT);// pasa de int a String
client.subscribe(subscribeTopic); //Subscribes to all messages send to the device
/*sendToBroker("report/online", "true"); // Reports that the device is online
delay(100);
sendToBroker("report/firmware", FIRMWARE_VERSION); // Reports the firmware
version
delay(100);
sendToBroker("report/ip", (char*)WiFi.localIP().toString().c_str()); // Reports the ip
delay(100);
sendToBroker("report/network", (char*)WiFi.SSID().c_str()); // Reports the network
name
delay(100);*/
char signal[5];// se declara la variable signal del tipo char
sprintf(signal, "%d", WiFi.RSSI());//
sendToBroker("report/signal", signal); // Reporta la intensidad d ela señal WIFI
delay(100);// se interrumpe el flujo d einstrucciones por 100 milisegundos
}
int splitTopic(char* topic, char* tokens[], int tokensNumber) {//
const char s[2] = "/";
int pos = 0;

```

```

tokens[0] = strtok(topic, s);
while (pos < tokensNumber - 1 && tokens[pos] != NULL) {
    pos++;
    tokens[pos] = strtok(NULL, s);
}
return pos;
}

int splitColor(char* color, char* tokensC[]) {
    const char s[2] = ",";
    int pos = 0;
    tokensC[0] = strtok(color, s);
    while (pos < 3 - 1 && tokensC[pos] != NULL) {
        pos++;
        tokensC[pos] = strtok(NULL, s);
    }
    return pos;
}

void checkMqtt() {
    if (!client.connected()) { //Si el micro está conectado con el broker
        startMqtt(); //Se inicia la comunicación entre el micro y el server smartnest.
    }
}

void sendToBroker(char* topic, char* message) {
    if (client.connected()) {
        char topicArr[100];
        sprintf(topicArr, "%s/%s", MQTT_CLIENT, topic);
        client.publish(topicArr, message);
    }
}

void turnOff() {
    printf("Turning off...\n");
    analogWrite(lightPin3, 0);
    analogWrite(lightPin4, 0);
    analogWrite(lightPin5, 0);
    delay(100);
    // Serial.println(lightPin);
    sendToBroker("report/powerState", "OFF");
}

void turnOn() {

```

```
Serial.println("Turning on...\n");
//digitalWrite(lightPin, HIGH);
//Serial.println(lightPin);
analogWrite(lightPin3, red);
analogWrite(lightPin4, green);
analogWrite(lightPin5, blue);
delay(100);
sendToBroker("report/powerState", "ON");
}
void setColor(int r, int g, int b) {
  red = r;
  green = g;
  blue = b;
  printf( "Color set to %d,%d,%d\n", red, green, blue);
  //Serial.println( "Hola Mundo");
  analogWrite(lightPin3, red);
  analogWrite(lightPin4, green);
  analogWrite(lightPin5, blue);
  char msg[18];
  sprintf(msg, "rgb(%d,%d,%d)", red, green, blue);
  sendToBroker("report/color", msg);
}
```