Tlamati Sabiduría

Arquitectura cliente-servidor para el desarrollo de aplicaciones IoT

Christian Camilo Laguna-Garzón[1]
Mariluz Romero-García[1]

[1]*Universidad Distrital Francisco José de Caldas Cl.13 #31-75, Localidad de Chapinero, Bogotá, Cundinamarca, Colombia*

**Autor de correspondencia*
cclagunag@correo.udistrital.edu.co

Resumen

Este documento describe los artefactos importantes de un proyecto IoT. El lector debe estar familiarizado con sistemas de software, hardware y protocolos de comunicación, además de conocimientos básicos de redes y equipos de 'networking' como 'router', switches´, topologías de red, etc. En el transcurso del texto no se puede hacer énfasis sobre todos los sensores y actuadores que existen, el uso de tecnologías de programación, por ser temas extensos que un artículo no tiene la suficiente profundidad para mostrarlos, en resumen se pretende dar una generalización teórica de un esquema en capas para mostrar los dispositivos básicos de un proyecto IoT y sobre éste diseñar un algoritmo para comprender, desarrollar e implementar este tipo de artilugios IoT que están de auge en cualquier parte del mundo.

Palabras clave: Hardware, Software, Modelo en capas, IoT, Algoritmo, Protocolos, Comunicación.

Abstract

This document describes the important artifacts of an IoT project. The reader must be familiar with software systems, hardware and communication protocols as well as basic knowledge of networks and network equipment such as routers, switches, network topologies, etc. In the course of the text, it is not possible to

Información del Artículo

Cómo citar el artículo:
Laguna-Garzón, *C.C.*, Romero-García, M. (2023). Arquitectura cliente-servidor para el desarrollo de aplicaciones IoT. *Tlamati Sabiduría*, 17, 17-24.

Editores Invitados: José Efrén Marmolejo-Valle, Manuel de Jesús Matuz-Cruz, María Palmira González-Villegas, Rubén Suárez-Escalona, Adalberto Iriarte-Solís, Samuel Hernández-Calzada, José Antonio Jerónimo-Montes et al.

Recibido en la versión aceptada por los editores invitados: Noviembre de 2023

© 2023 Universidad Autónoma de Guerrero

emphasize, on all the sensors and actuators that exist, especially the use of programming technologies are dense topics that an article does not have enough depth to show, in summary it is intended to give a theoretical generalization from a layered scheme to show the basic devices of an IoT project, on this to design an algorithm to understand and develop this type of IoT gadgets that are being implemented anywhere in the world.

Keywords: Hardware, Software, Layer model, IoT, Algorithmic, Protocols, Communication.

Introducción

El internet de las cosas (IoT por sus siglas en inglés) es una tecnología imprescindible en el mundo actual. Hoy en día es común que una persona pueda monitorear las cámaras de su casa desde su celular o que el gobierno monitoree en tiempo real un bosque mediante detectores de incendios si cuenta con una conexión de internet. Ahora se puede introducir el tema principal de un texto y mostrarle al lector el algoritmo con el que está escrito y que sirva como base para desarrollar un prototipo IoT o proyecto completo, se evidencia la construcción de un modelo en capas llamado USSIoT análogo al modelo OSI, siendo esta la primera versión del documento, en este se pretende generalizar los artefactos que debe tener todo proyecto IoT sin importar su tamaño. El motivo de realizar estos pasos es facilitar la comprensión de este tipo de proyectos. No solo servirá como base para el diseño de estos sino para enseñar a todos los que quieran aprender a implementar este tipo de artefactos. Recuerde que este documento no pretende desarrollar temas de seguridad, pero son importantes en cualquier proyecto de la actualidad. No se pretende tampoco dar énfasis en los distintos protocolos de comunicación que se pretendan usar, porque existen muchos, y eso complicaría la comprensión del documento. Comprender la importancia de crear aplicaciones en tiempo real para monitorear sensores son temas densos, pero de importancia.

Objetivo

Conocer los principales elementos que componen un sistema IoT a nivel de software, hardware y transmisión de datos. Así mismo, se intenta ilustrar el modelo de desarrollo de proyectos IoT (denominado USSIoT), presentando las diferentes capas para la construcción. Finalmente, se presenta un algoritmo para la integración de las capas del modelo USSIoT.

Sistemas IoT

Los sistemas IoT son aquellos que interconectan recursos API de internet con objetos del mundo físico. La pregunta que surge es ¿Por qué es importante aprender sobre ellos? Para responder esta incógnita se va a escribir sobre las ciudades 4.0, las cuales se definen como lugares completamente digitalizados (MinTIC, 2021). Esto quiere decir, por ejemplo, que desde su dispositivo móvil se pueda abrir una puerta o desde una página web sea posible monitorear en tiempo real un área extensa con cualquier tipo de sensor. Las ciudades 4.0 buscan automatizar y desde la distancia dirigir procesos, sean industriales, medio ambientales y de cualquier tipo.

Los beneficios que ha traído la tecnología IoT han sido optimizar procesos desde cualquier parte del mundo que cuente con señal de internet. Sin este servicio, puede convertirse más bien en una desventaja. Las ciudades inteligentes demuestran que la creatividad y la tecnología pueden transformar las ciudades en lugares sostenibles y eficientes como se señala (MinTIC, 2021).

Artefactos imprescindibles en un proyecto IoT

El desarrollo de proyectos IoT requiere un algoritmo lógico de pasos ordenados, donde se busca dilucidar una solución óptima, antes que el lector empiece a esbozar el artículo. En la Figura 1, se muestran los componentes electrónicos necesarios con sus respectivos símbolos que se

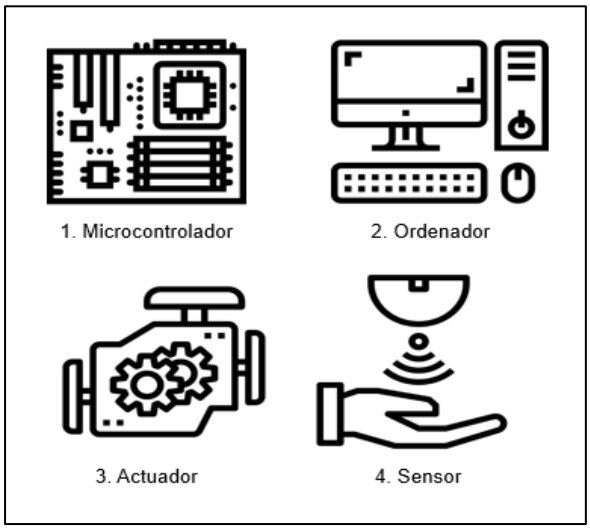

Figura 1. Componentes Electrónicos.

utilizarán para ilustrar más amenamente la solución de acuerdo con los objetivos propuestos.

El microcontrolador es una placa física que tiene procesador, memoria y puertos I/O (Entrada y salida). Estos periféricos permiten la conexión de sensores y actuadores. La diferencia entre un computador de escritorio y un microcontrolador como Arduino es su capacidad. ¡Solo basta comparar un procesador INTEL CORE 7! de novena generación con un procesador ATmega 338 incorporado en el Arduino (Monk, 2019).

La memoria del Arduino es de 32 KB mientras que la de computador de mediano almacenamiento es de 1 TB (Monk, 2019). Además, posee un sistema operativo que permite manejar el hardware, mientras que el Arduino no tiene sistema operativo. No todo es negativo en el microcontrolador; el computador no tiene entradas en sus placas para la implementación sistemática de componentes diferentes a la pantalla, mouse, teclado y sonido que son los que se colocan habitualmente.

Los sensores son dispositivos de entrada que proveen una salida manipulable de la variable física medida como la temperatura y la humedad (Corona-Ramírez *et al.*, 2014). El control de la temperatura es importante ya que una simple chispa puede provocar un incendio en un bosque, de ahí su importancia. Finalmente, los actuadores son dispositivos con la capacidad de generar fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico a partir de la transformación de energía (Monk, 2019). En resumen, los actuadores se distinguen de los sensores en el sentido que no recolectan datos, sino que realizan una acción mecánica. Siguiendo con el ejemplo de la chispa del bosque, los sensores son los detectores de incendios, pero el actuador en este caso es la alarma. Si el sensor transmite un TRUE significa que el incendio existe, entonces la alarma recibe una instrucción de otro componente para que se active. Otra diferencia es que el sensor envía datos y el actuador recibe instrucciones. Existen proyectos grandes, pero siempre se usan actuadores y sensores, es decir, que en proyectos IoT no existirían sin los componentes de la Figura 1.

Se han mencionado las diferencias entre el microcontrolador y el computador, pero se requiere explicar también el funcionamiento del microcontrolador en el proyecto IoT. Puesto que Arduino es también un componente electrónico, para evitar confusiones, en lo sucesivo los ejemplos mencionados serán relacionados con el microcontrolador Arduino.

Se han señalado las partes de un microcontrolador, sea procesador, memoria o puertos de entrada I/O. EN adelante se hará énfasis será en los puertos I/O.

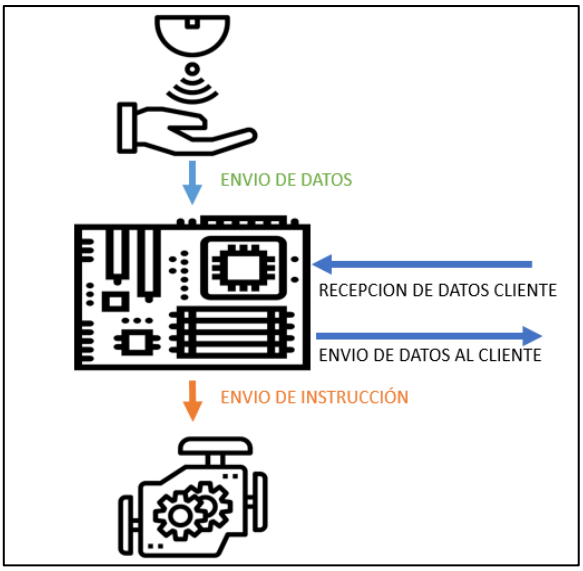

Figura 2. Funcionamiento del microcontrolador

Los puertos I/O permiten la recolección de datos de los sensores y la transmisión de instrucciones a los actuadores, finalizando estos a su vez con una tarea mecánica (Fig. 2) Si se estriba en un ejemplo de Arduino en el desarrollo de un carro electrónico (e.g. Hoffman, 2018), los servomotores hacen referencia en la vida cotidiana al motor del carro, pero todo motor necesita una instrucción para moverse. Las personas envían las instrucciones con el acelerador, los frenos y el manubrio. Arduino en esta anécdota es la persona, quién mediante instrucciones en lenguaje C envía instrucciones en voltios no directamente a los servomotores, sino a un puente H (se observa que este hace referencia al manubrio, freno y acelerador). El puente H se encarga de direccionar los servomotores, sea adelante o atrás, en resumen, para condensar en una simple explicación mediante lenguaje C el microcontrolador envía (actuadores) y recibe (sensores) datos de los sensores. La ventaja es que Arduino no solo se limita a estos dos puntos, sino que cualquier problema computable se puede adaptar en el Arduino, sean algoritmos de redes neuronales o algoritmos genéticos, pero en este no es el enfoque del artículo.

Es menester ahora hacer posible la comunicación entre el microcontrolador y el cliente para la transmisión y recepción de la data, como se ilustra en la Figura 2. El cliente puede ser internet o una aplicación móvil. Ahora, es momento de introducir los componentes de red de la Figura 3.

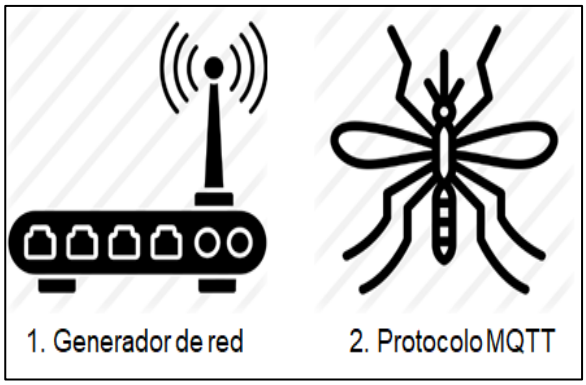

Figura 3. Componentes de Red y protocolo. Universidad Distrital Francisco José de Caldas de comunicación.

El generador de red, provee el medio físico para la comunicación entre el microcontrolador y el servidor. Ahora se ha introducido una nueva palabra: “servidor” La internet no sería posible si no existiesen los servidores que proveen la información que el usuario final necesita, sea Google o Facebook. Por ejemplo, si se quiere crear una aplicación que muestre al usuario final cuánta temperatura hay en la casa que él habita, necesita desarrollar un sistema de información web que revele estos datos a su vista. El servidor, en este caso, es el computador del desarrollador. El generador de red introduce el medio físico que permite que la computadora y el microcontrolador se comuniquen.

Ahora, la imagen derecha de la Figura 3 se trata de un símbolo peculiar: “el mosquito”, representando el protocolo MQTT fundamental para consolidar la transmisión y recepción de datos entre el microcontrolador y el servidor. Recuerde que la diferencia entre el generador de red y el protocolo será un *leitmotiv* de ahora en adelante. El generador (router casero o generador WiFi) se diferencia del protocolo MQTT en que se centra en generar el medio físico inalámbrico para la transmisión de datos. En cambio, el protocolo MQTT es un software o programa que usa el medio WiFi (estándar IEEE 802.11) para enviar mensajes entre el servidor y el microcontrolador (en MQTT será Cliente), mediante reglas definidas en el protocolo se genera una comunicación bidireccional mostrada en la Figura 4. No obstante, en la comunicación no es suficiente implementar el protocolo de comunicación si la aplicación o nuestro proyecto reacciona a cambios constantes que se presenten en la toma de datos, por causa y efecto se necesita instalar websockets (Socket.IO, 2022) que permite que la aplicación cambie en tiempo real constantemente cada vez que el servidor detecte cambios en la toma de datos, por ejemplo, en la toma de datos de temperatura (Santos y Santos, 2022a) o también hacer proyectos de reconocimiento de imágenes (Santos y Santos, 2022b) o cambiar el color de un bombillo (Santos y Santos, 2022c) e incluir una pequeña base de datos para guardar información obtenida por los sensores (Santos y Santos, 2022d).

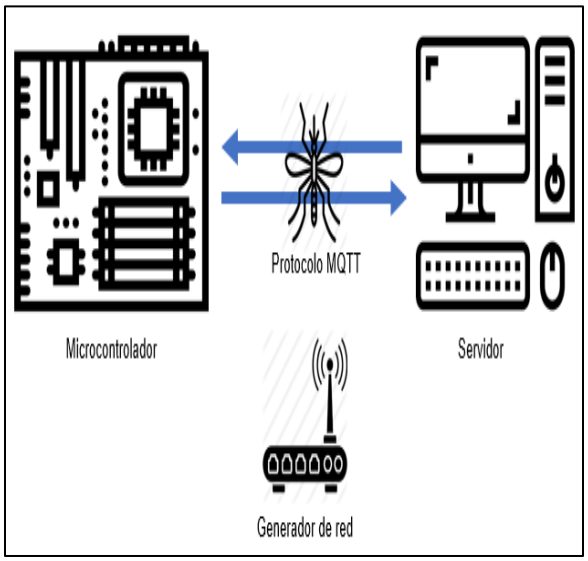

Figura 4. Comunicación entre servidor y microcontrolador.
Universidad Distrital Francisco José de Caldas

Recuerde que MQTT y el generador de red son imprescindibles en un proyecto (MQTT está hecho específicamente para proyectos IoT), por su bajo uso de ancho de banda, mínima latencia y es open source. Es equivalente a decir que todos tienen acceso a su código para implementarlo en sus proyectos para más información sobre el protocolo, sea su instalación uso y securización, consultar (Hillar, 2017).

Continuando con las partes finales de un proyecto IoT, es decir, las partes interactivas con el usuario final, se encuentra el software web y el cliente. El usuario final es importante en el proyecto ya que es quien va a recibir los datos de los sensores, o el que envíe instrucciones a los actuadores a través del software web (Fig. 5).

En resumen, el microcontrolador (puede ser Arduino o cualquier otro) contiene los sensores y los actuadores, recuerde que el microcontrolador y el software son clientes del protocolo MQTT y donde el servidor provee el servicio MQTT para que funcione la comunicación entre hardware y software (Fig. 6). Tanto el microcontrolador y el software web no se comunican directamente, viendo desde otra cima, el Arduino mediante el protocolo MQTT no puede comunicarse

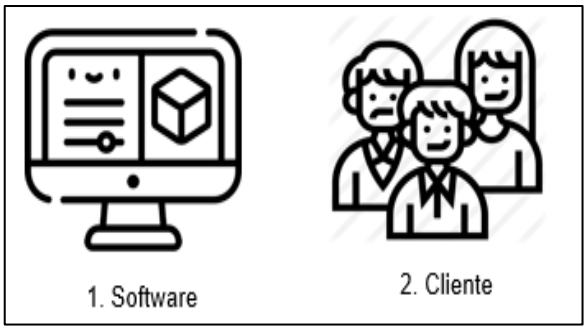

Figura 5. Iconos de Software y usuarios.

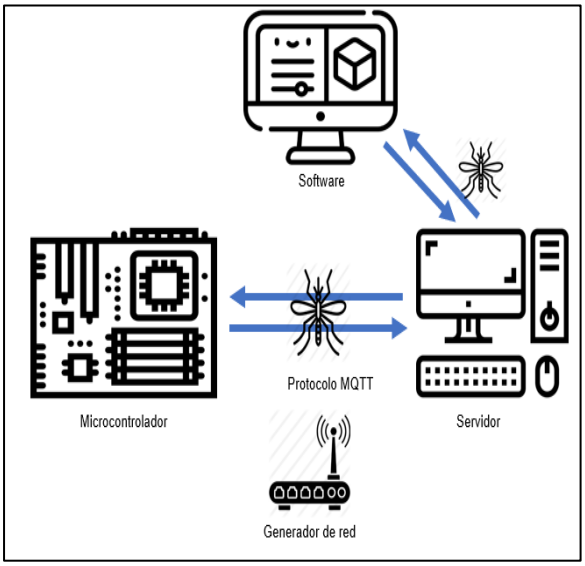

Figura 6. Principales componentes del proyecto IoT.

directamente con el software web, sino que requiere de un servidor, protocolo y un medio inalámbrico como WiFi que conecte permanentemente todos los dispositivos.

Como se ilustra en la Figura 7, la relación entre el cliente y el prototipo IoT nace desde el software. El usuario utiliza el proyecto mediante una aplicación (no confunda cliente con cliente MQTT). Recuerde que tanto el software como el microcontrolador son clientes MQTT. La diferencia radica en que el primero interactúa con el proyecto mediante el software y el segundo con el servidor para transmitir órdenes (actuadores) o información desde los sensores a la aplicación web.

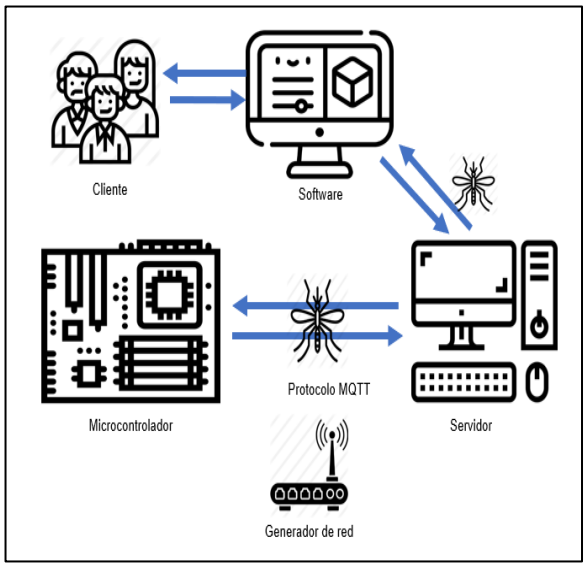

Figura 7. Interacción entre el cliente y el proyecto.

Modelo USSIoT para proyectos IoT

Anteriormente se explicaron los objetos imprescindibles en un proyecto IoT, sea grande o singular, de aquí en adelante se va a dilucidar un algoritmo o procedimiento de acuerdo con el modelo USSIoT mostrado en la Figura 8, éste se lee de abajo a arriba desde la capa de sensores y actuadores hasta usuarios, el lector se cuestionará por qué de esta forma y no de arriba hacia abajo, una buena práctica es empezar programando el hardware, los sensores y la comunicación, y otras dos cuestiones son: principalmente artefactos necesarios y por ende conseguirlos es una prioridad, y la segunda razón es cuando se están analizando las prioridades del cliente, entonces seleccionar los sensores, actuadores y microcontroladores requiere de una planeación y consulta previas.

Algoritmo USSIoT para proyectos IoT

De acuerdo con la Figura 8, el algoritmo para desarrollar proyectos IoT se define de la siguiente manera:

- Definir los requerimientos de acuerdo con las necesidades del cliente (recuerde que este punto es, *a priori*, el más importante de todos, si no se entiende este ítem el proyecto entero quedará mal).
- Según el punto anterior, seleccionar el microcontrolador, los sensores y actuadores y, por último, efectuar la compra.
- Mediante un firmware configurar el funcionamiento del microcontrolador con chip WiFi y configurar el protocolo MQTT (Arduino tiene las librerías), terminando este punto se finaliza la configuración de las capas 1 y 2.
- La capa de comunicación se encuentra en el protocolo MQTT y el generador de protocolos (puede ser un router de baja calidad) recuerde que después de comprar el generador de red asegúrese que el microcontrolador se conecte mediante WiFi al generador de red.
- Instalar el protocolo MQTT en el servidor de capa 4 y conectarlo a la red WiFi del generador. Mediante consola de comandos en el servidor probar la comunicación por MQTT entre el servidor y el microcontrolador (Si no funciona comprobar la configuración del protocolo y que estén conectados en la misma red tanto el servidor como el microcontrolador).

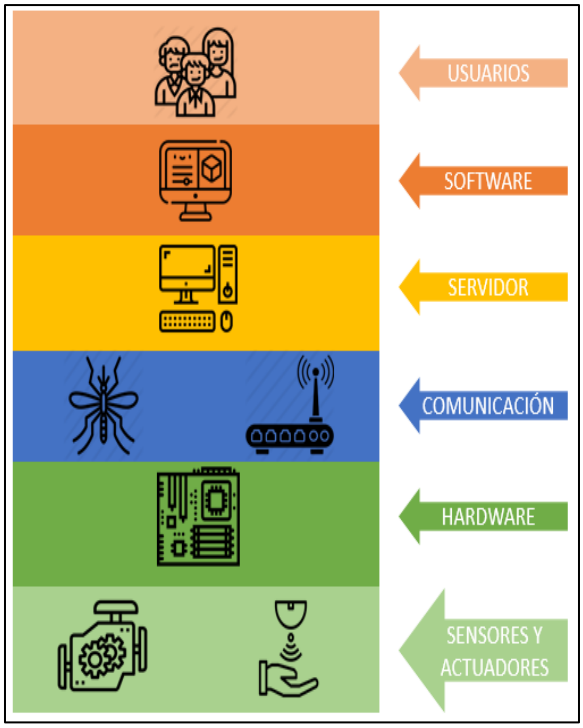

Figura 8. Modelo USSIoT

- Instalar los componentes de software como por ejemplo node js para más información (Nodejs Developers, 2022) y angular para crear el componente visual (Google Developers, 2022) e instalar 'socket-io'.
- Este paso hace parte de la capa de comunicación, el cual involucra una técnica de cifrado para mantener los datos ocultos a los ojos de los intrusos, este cifrado afecta la capa de software y hardware porque en los dos hay que configurar los algoritmos tanto de cifrado como el desencriptado para más información (Klima y Sigmon, 2018).
- Por último, la capa de usuarios involucra a los clientes los cuales son los encargados de evaluar si los requerimientos o el primer punto de este algoritmo se cumple, si llegase el caso de no cumplirse, se debe volver al primer paso y hacer el recorrido secuencial, para obtener más ideas sobre proyectos IoT consultar (Santos y Santos, 2021).

Este algoritmo empieza desde el análisis de requerimientos, capa de sensores y actuadores, punto a punto hasta la capa de usuarios como se muestra en la Figura 9.

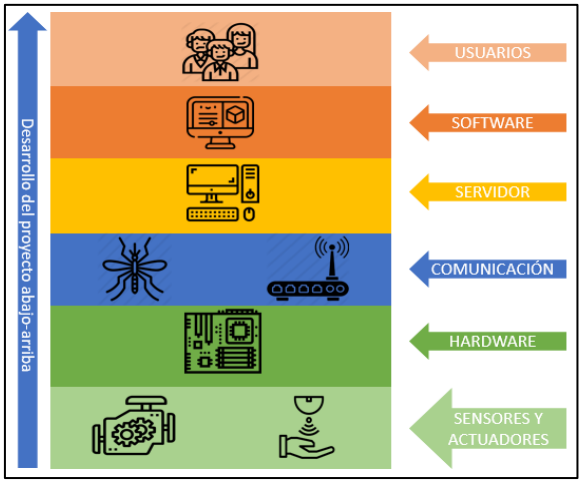

Figura 9. Proceso de desarrollo de proyecto IoT según el modelo USSIoT.

Resultados

De acuerdo con el modelo teórico mostrado en la Figura 9, en el documento se ilustró un algoritmo de 9 pasos, con los cuales involucran las capas definidas del modelo USSIoT, siendo este una analogía al modelo OSI, consta de 6 capas, empezando por análisis de requerimientos hasta presentar el proyecto completo a los clientes, finalizando el proyecto IoT consta de 8 elementos importantes: sensores, actuadores, microcontrolador, generador de red sea WiFi, protocolo MQTT, Servidor, Software y los clientes, cada uno es fundamental para el funcionamiento de un proyecto IoT sea simple o complejo.

Conclusiones

Se ha construido en este documento un modelo de 6 capas denominado USSIoT, en su primera versión, el desarrollo del proyecto involucra este modelo en 9 pasos que empiezan por el análisis de requerimientos, después desde la capa de sensores y actuadores hasta los usuarios finales, es decir se va a hacer de forma ascendente, no se menciona la seguridad del proyecto, porque eso le corresponde al desarrollador de hardware y de software implementarlo, aunque el protocolo MQTT tiene esas opciones, en siguientes versiones del documento se va a enfocar en la parte de seguridad.

Agradecimientos

Agradecimientos de Christian Camilo Laguna Garzón: Agradezco a mi familia por el apoyo incondicional que han brindado siempre, de todas las personas en que han estado en mi vida, nunca me han dejado solo, este trabajo se los dedico.

Referencias

MinTIC (2021). Conozca en Colombia 4.0 cómo las grandes ciudades se están transformando en lugares inteligentes. Ministerio de Tecnologías de la Información y las Comunicaciones. https://mintic.gov.co/portal/inicio/Sala-de-Prensa/Noticias/103706:Conozca-en-Colombia-4-0-como-las-grandes-ciudades-se-estan-transformando-en-lugares-inteligentes.

Monk, S. (2019). Programming Arduino Next Steps Going Further with Sketches. New York: McGraw Hill Education, 264p.

Corona-Ramírez, L.G., Abarca-Jiménez, G.S., Mares-Carreño, J. (2014). Sensores y actuadores aplicaciones con arduino. Ciudad de México, México: Patria, 320p.

Hoffman, J. (2018). Mastering Arduino: A project-based approach to electronics, circuits, and programming. Mumbai, India: Packt Publishing Ltd, 372p.

Socket.IO (2022). What Socket.IO is. Introduction. https://socket.io/docs/v4/

Santos, R., Santos, S. (2022a). Random Nerd Tutorials. https://randomnerdtutorials.com/esp8266-multisensor-shield-with-node-red/

Santos, R., Santos, S. (2022b). Random Nerd Tutorials. https://randomnerdtutorials.com/car-plate-recognition-system-with-raspberry-pi-and-node-red/

Santos, R., Santos, S. (2022c). Random Nerd Tutorials. https://randomnerdtutorials.com/node-red-with-xiaomi-yeelight-rgbw-smart-bulb/

Santos, R., Santos, S. (2022d). Random Nerd Tutorials. https://randomnerdtutorials.com/sqlite-with-node-red-and-raspberry-pi/

Hillar, G.C. (2017). MQTT Essentials-A Lightweight IoT Protocol. Birmingham, Inglaterra: Packt Publishing Ltd, 280p.Node.js, Developers (2022). Documentación. https://nodejs.org/es/docs/

Google, Developers (2022). Introduction to the angular docs. https://angular.io/docs

Klima, R.E., Sigmon, N. (2018). Cryptology Classical and Modern. New York, London, CRC Press, 496p.

Santos, R., Santos, S. (2021). Random Nerd Tutorials. https://randomnerdtutorials.com/