FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DISEÑO DE UN MECANISMO DE AUTORIZACION EN UN ECOSISTEMA IOT.

AUTOR

Francisco Xavier Jiménez Bautista

AÑO

2020

FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DISEÑO DE UN MECANISMO DE AUTORIZACIÓN EN UN ECOSISTEMA IOT.

Trabajo de Titulación presentado en conformidad con los requisitos establecidos para optar por el título de Ingeniero en Sistemas de Computación e Información.

Profesor Guía

MSc. Eddy Mauricio Armas

Autor

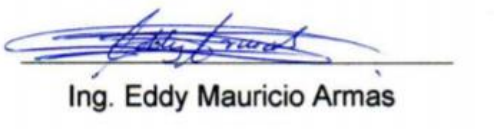
Francisco Xavier Jiménez Bautista

Año

2020

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido el trabajo, Diseño de un mecanismo de autentificación en un ecosistema IoT, a través de reuniones periódicas con el estudiante Francisco Xavier Jiménez Bautista, en el semestre 2020-2 orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.



Ing. Eddy Mauricio Armas

MAGISTER EN GERENCIA DE SISTEMAS Y TECNOLOGÍAS DE INFORMACIÓN

C.I.: 1711715803

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, de Francisco Xavier Jiménez Bautista, en el semestre 2020-2, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Bernarda Cecibel Sandoval Romo

Mestra en Ciencia da Computacao

C.I.: 1709974453

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”.

Francisco Xavier Jiménez Bautista

C.I.: 1717005266

AGRADECIMIENTOS

A Dios y mi familia, en especial a mi hermana y padres quienes me han apoyado en todo momento. También a los docentes que han aportado en mi formación académica.

DEDICATORIA

El trabajo lo dedico a Dios por estar en cada paso en el transcurso de mi vida, a toda mi familia y sobre todo a mi hermana quien estuvo siempre como amiga y modelo a seguir gracias por todo, a mi mami quien nos inculco valores y con el apoyo incondicional para seguir adelante y a mi nueva familia que es un motivo más para ser mejor y luchar por mis metas.

RESUMEN

El uso del Internet of Things (IoT) ha incrementado considerablemente, logrando introducir ecosistemas IoT en diferentes ambientes como salud, educación, industrias, hogares entre otros. Al ser una tecnología nueva hay que tener en cuenta que puede tener fallos de seguridad por parte de los diferentes dispositivos o compañías que se dedican a la fabricación o prestación de servicios IoT, presentando falta de control respecto a la seguridad y problemas con los clientes o usuarios que la usan. Esta tecnología es vulnerable al robo de información que se da en el proceso de autorización y autenticación del usuario a diferentes aplicaciones o servicios. Ocurre también, cuando se proporciona credenciales de una forma no segura, lo que facilita el robo o suplantación dentro del internet. Una vasta revisión bibliográfica ha permitido identificar, analizar estudios y procesos vinculados a la autorización y autenticación en un ecosistema IoT, de donde nace esta propuesta del modelo de autorización que tiene como objetivo primordial la seguridad del usuario, garantizando el acceso de forma segura y sin riesgo de robo de información. El protocolo OAuth es base de la propuesta y partiendo de su estructura se diseña un modelo que mitigue las vulnerabilidades en la seguridad. Con ayuda de las funciones que plantea Proof of Possesion (POP) podemos garantizar un proceso de autorización y autenticación de manera segura y confiable en un ecosistema IoT.

Palabras Claves: IoT, autorización, autenticación, OAuth 2.0.

ABSTRACT

The use of the Internet of Things has increased considerably, being able to introduce Iot ecosystems in different environments such as health, education, industries, homes, among others. As it is a new technology it has to be taken into account that it may have security failures on the part of the different devices or companies that are dedicated to the manufacture or provision of Iot services, presenting lack of control regarding security and problems with customers or users who use it. This technology is vulnerable to the theft of information that occurs in the authorization and authentication process of the user to different applications or services. It also occurs when credentials are provided in an unsafe manner, which facilitates theft or impersonation within the internet. A vast bibliographical review has made it possible to identify, analyze studies and processes related to authorization and authentication in an Iot ecosystem, from where this proposal of the authorization model was born, whose primary objective is user safety, ensuring access in a safe manner and without risk of theft of information. The Oauth protocol is the basis of the proposal and, based on its structure, a model is designed to mitigate security vulnerabilities. With the help of Proof of Possesion (POP) functions, we can guarantee a secure and reliable authorization and authentication process in an Iot ecosystem.

Key Words: IoT, authorization, authentication, Oauth 2.0.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Glosario de acrónimos

IoT = Internet Of Things

MQTT = Message Queing Telemetry Transport

AWS STS = AWS Security Token Service

AWS IAM = AWS Identity and Access Management

SSL = Secure Socket Layer

TLS = Transport Layer Security

TCP = Transmission Control Protocol

HTTPS = Hyper Text Transport Protocol Secure

HTTP = Hyper Text Transport Protocol,

JWS = Json Web Signature

JWK = Json Web Key

JWT = Json Web Tokens

CAPÍTULO I. INTRODUCCIÓN

A continuación, veremos algunos argumentos del por qué este tema es de interés para los usuarios de la tecnología, de qué se trata esta propuesta y cuál es su fin con el estudio de la autorización en un ecosistema IoT.

1.1. Antecedentes

Cuando se habla de seguridad de la información es importante conocer el termino CIA (Confidencialidad, Integridad, Disponibilidad), tres conceptos que representan los principios básicos. Una correcta gestión de la seguridad de la información prevé estos tres elementos, sin ellos no existe nada seguro, la falta o falla de uno de estos elementos es causal de peligro para nuestra seguridad. Tenemos que recordar que ningún sistema de seguridad es completamente fiable, siempre se debe tener claro que un sistema es mucho más vulnerable de lo que se piensa. Una vez que tenemos esto claro podemos tomar medidas necesarias para mitigar riesgos. (ISO 27001:2013, 2017).

El paso de los años, el avance tecnológico y desde la llegada del Cloud Computing ha permitido que el desarrollo de dispositivos IoT sea una realidad y ha comenzado a formar parte de la sociedad. En la actualidad puede ser utilizado en la vida cotidiana, hoy lo encontramos en hogares, educación, salud, vehículos y en importantes industrias (Rivas, 2018).

A medida que las implementaciones de IoT aumentan constantemente, incrementa la necesidad de una mejor experiencia de usuario para manejar las tareas de autenticación y autorización en entornos restringidos. Si bien ya se han desarrollado varias tecnologías que permiten el acceso a recursos protegidos, la naturaleza de las implementaciones de IoT requiere atención con los recursos limitados disponibles en muchos de estos dispositivos (Tschofenig, 2016).

La falta de seguridad en IoT es algo que está presente con frecuencia, lo que produce ataques o fallas que las empresas tecnológicas intentan solucionar

rápidamente con una actualización o parche. Al crear dispositivos de fácil uso, conectables y con nuevas funciones, las compañías descuidan la seguridad en los dispositivos que es crucial por atención a la competencia, inmediatez y agresividad del mercado. (Albors, 2018).

La mayoría de las aplicaciones utilizadas por los dispositivos IoT, solicitan al usuario como paso inicial él envió de sus credenciales y su autenticación para acceder al sistema, convirtiéndose este paso en el primer problema de inseguridad por las diversas amenazas que existen en la Internet. En caso de no contar con las debidas seguridades será inevitable que la información sea robada. Una de las trascendentales ventanas de robo de información es la autenticación, al ser el paso principal para el acceso hacia las diferentes aplicaciones, los usuarios proporcionan las credenciales en la Internet y al no tener una seguridad adecuada los dispositivos IoT se pueden convertir en recursos vulnerables. (Srivastava & Sivasankar, 2017)

1.2. Objetivos

1.2.1. Objetivo General

Diseñar un mecanismo que permita gestionar eventos de seguridad en la autorización en un ecosistema IoT.

1.2.2. Objetivos específicos

- Analizar soluciones existentes de autorización y modelos de seguridad para dispositivos IoT.
- Diseñar un modelo de seguridad para la autorización en un ecosistema IoT.
- Articular las funciones de seguridad del modelo Proof of Possession para la autorización de Ecosistemas IoT.

1.3. Alcance

El desarrollo de los objetivos planteados en el apartado anterior con llevan una identificación clara de acciones concretas que permitan ejecutar un mejor análisis de los diferentes mecanismos existentes.

Para el primer objetivo, *análisis las posibles soluciones de autorización existentes y modelos de seguridad para dispositivos IoT mediante una revisión de bibliografía preliminar*, se prevé los siguientes puntos:

- Análisis de protocolos de seguridad para dispositivos IoT.
- Diseño de un modelo de seguridad que contenga características de mitigación de ataques necesarias para seguridad de autorización de dispositivos IoT.
- Caracterización de las técnicas y métodos de autorización para asegurar un ecosistema IoT.

El segundo objetivo, *diseño de un modelo de seguridad para la autorización en un ecosistema IoT*, implica los siguientes pasos:

- Determinación del protocolo a utilizar en el mecanismo.
- Implementación del modelo y protocolo de seguridad diseñado.

Y el tercer objetivo, articulación de las funciones de seguridad del modelo Proof of Possession para la autorización de Ecosistemas IoT, se alcanzará mediante:

- Adaptación del modelo PoP en la propuesta de seguridad.
- Identificación de los beneficios que brinda el modelo PoP a la propuesta.

1.4. Justificación

Algunos referentes numéricos citados por iT Reseller (2017) hacen referencia a la publicación de la firma de investigación Gartner, la cual menciona que se evidencia un aumento del 31% de dispositivos IoT de 2016 a 2017, alcanzando

8.400 millones de “cosas” conectadas en el 2017 y a la vez, proyecta que el número aumentará a 20.400 millones para 2020. En la figura 1 se observa una gráfica de un constante incremento de ecosistemas IoT lo que hace que la seguridad de los dispositivos conectados se muestre vulnerable. En este caso la demanda de usuarios provoca que las empresas atiendan la comercialización de sus productos, en primera instancia, antes que la seguridad que requieren éstos. Un ejemplo claro es el caso Uber, esta empresa fue afectada con el robo de datos de identidad de sus clientes al punto de tener que pagar a piratas informáticos $100.000 para recuperar 57 millones de datos de conductores y usuarios (York, 2017).

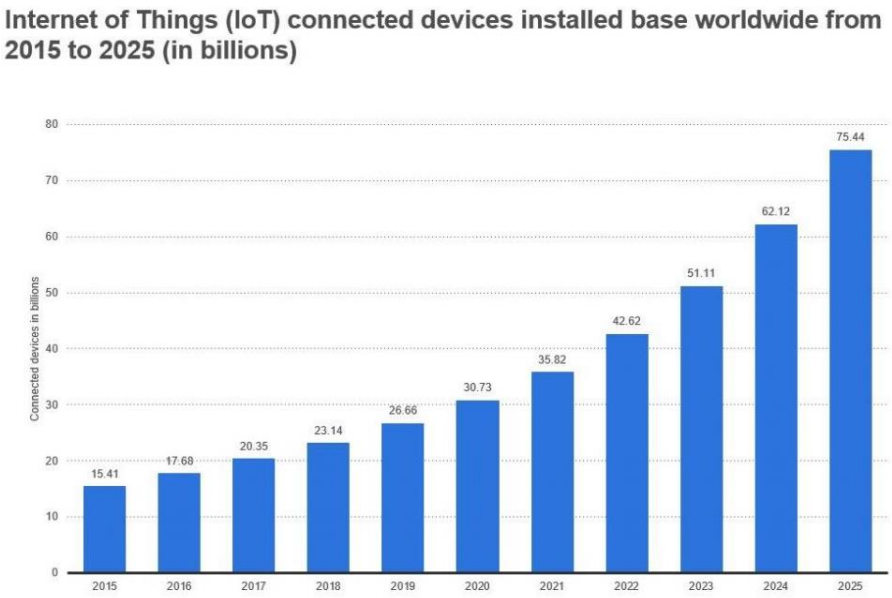


Figura 1. Dispositivos IoT conectados para el año 2025

Tomado de de (IoT World Online, 2018).

Solo en EE. UU. las brechas de seguridad cuestan a las empresas más de $ 445 mil millones de dólares anuales. A medida que IoT crece este número solo aumentará. (Sandoval, 2017). En este sentido y para prevenir el robo de información y suplantación de identidad en empresas o usuarios, se justifica las acciones preventivas más que correctivas. Es así como, todas las propuestas de seguridad digital que se desarrollen enfocan un plan de mitigación de riesgos ante cualquier amenaza que haga vulnerable al usuario.

Este diseño del mecanismo de autorización se alinea al plan de seguridad y permite garantizar la comunicación directa entre servidores convirtiéndolo en una ruta con mayor eficacia para el manejo de credenciales.

CAPÍTULO II. MARCO TEÓRICO

En este capítulo describiremos algunos conceptos necesarios para poder analizar y entender de mejor manera el tema a desarrollar. Además, tendremos una perspectiva más amplia de los componentes que intervienen en un proceso de autorización.

En la figura 2 podemos observar una arquitectura IoT de Amazon desde el cliente hasta el acceso a recursos y podemos observar que existe un módulo de autorización y autenticación.

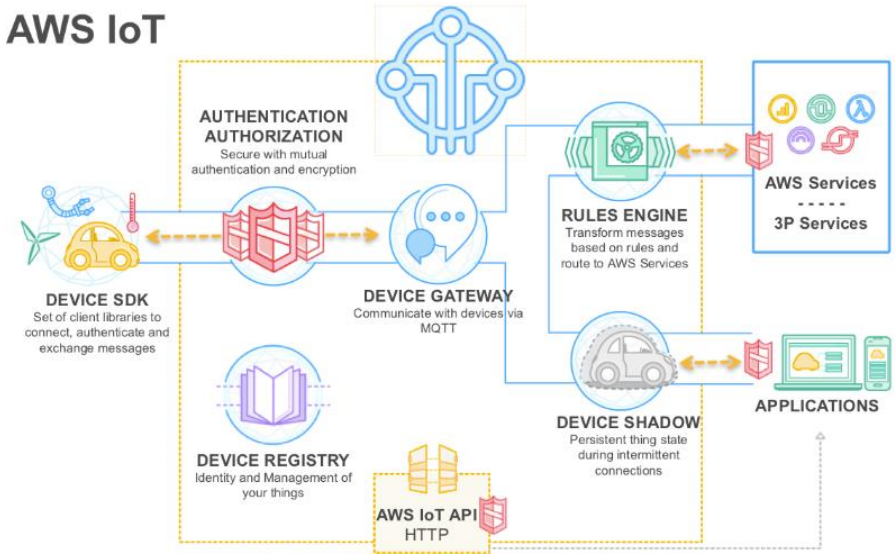


Figura 2. AWS IoT

Tomado de (Amazon, 2018).

2.1. Internet of Things

Con la evolución tecnológica y el uso del Internet como necesidad, se ha dado lugar al denominado IoT que representa la interconexión de objetos cotidianos como teléfonos, lavadora, y otros, que se conectan entre sí a través de una Red, convirtiéndose así en objetos inteligentes que permiten enviar y recibir datos en tiempo real, uniendo tanto el mundo digital con el físico. Para la interconexión de objetos entre sí a través de una red como se observa en la figura 3 se debe tomar en cuenta tres factores, los dispositivos, la red y el sistema de control (Valois, 2020).

Figura 3.Internet of Things

Tomado de (Peña, 2020)

En la actualidad es posible que casi cualquier cosa forme parte del IoT; sin embargo, los beneficios no llegan sin riesgos; en gran cantidad los dispositivos IoT no tienen instalado un software que les brinde seguridad y el hecho de estar conectados a una red la que se proporciona todo tipo de información personal, como por ejemplo contraseñas, vuelve vulnerables a los usuarios de ataques existentes en el Red (Peña, 2020).

2.2. Arquitectura de referencia

Existen varias arquitecturas que brindan servicios para el manejo de dispositivos IoT, una de ellas es la Amazon IoT, la cual permite conectar las cosas de forma segura, analizar datos y entregar los resultados a compañías en plataformas inteligentes de IoT. Permite a los clientes automatizar el proceso de conexión de dispositivos a través de una red global con altas medidas de seguridad (Cisco, 2016).

En la figura 4 se puede observar una arquitectura en la que existe un módulo de autorización y autenticación el cual cumple la función de verificar credenciales, de esta forma podemos tener un acceso seguro y confiable.

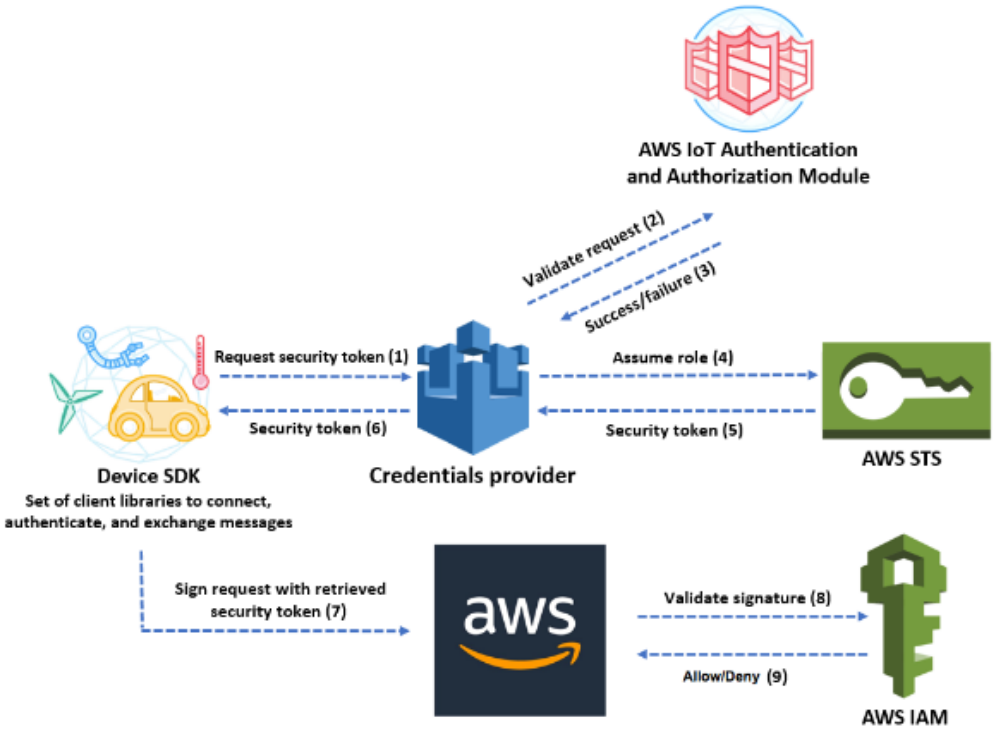


Figura 4. Autorización AWS IoT

Adaptado de (Amazon, 2018).

2.3. Plataformas

Dentro del mundo de IoT existen dos terminologías importantes como son plataformas y dispositivos. La plataforma IoT es la base para tener un ecosistema IoT con dispositivos interconectados, este representa el software que une puntos de acceso, redes de datos y hardware, en palabras más simples, la aplicación que utiliza el usuario (Cárdenas, 2016). Los dispositivos son objetos o el medio que permite al usuario conectarse a una Red como por ejemplo Internet, con el objetivo de recoger e intercambiar información. Existen un sin número de dispositivos inteligentes que permiten la conexión a través de la red como, por ejemplo, los que cita (Apiumhub, 2018):

- Sensores que permiten tener funciones de alerta anticipada de terremoto o tsunami.
- Objetos que usan energía, como interruptores, bombillas, televisores.
- El termostato Nest, facilita a los usuarios mediante el uso de teléfonos inteligentes o tablet controlar la temperatura de su hogar.
- Philips Hue, facilita controlar las luces desde cualquier lugar de su casa.

2.3.1. Plataformas como Hardware

Existen una gran cantidad de plataformas como hardware que nos ayudan a conectar dispositivos en un ecosistema IoT, en la tabla 1 se presentan los más usados.

Tabla 1.

Plataformas hardware.

Hardware	Descripción
Arduino	Es una plataforma libre con su propio software para desarrollar proyectos con facilidad, está basado en una placa compuesta por microcontroladores.

Waspmote	Es una plataforma de código libre de elaboración de redes de sensores inalámbricos.
Intel Galileo	Es una placa Arduino basada en la arquitectura de Intel.
Raspberry PI	Es un ordenador reducido, por ser de software abierto es utilizado para estimular la enseñanza.

2.3.2. Plataformas como Software

En un ecosistema IoT además de los hardware las plataformas como software son importantes al ser el motor o corazón del hardware, en la tabla 2 se describen los softwares más importantes que se usa en esta tecnología.

Tabla 2.

Plataformas software.

Software	Descripción
Amazon Web Services IoT	Amazon tiene diferentes servicios para dispositivos, de control y datos, además se tiene un SDK (software development kit) para dispositivos con AWS (Amazon Web Services) IoT que conecta el hardware del dispositivo con AWS IoT Core. “Además permite la comunicación bidireccional y segura entre elementos conectados a Internet (sensores, accionadores, dispositivos integrados o inteligentes) y la nube de AWS a través de protocolos MQTT (Message Queue Telemetry Transport) y HTTP (Hypertext Transfer Protocol)” (Amazon,2020).

Azure IoT	Azure IoT Hub se conectan los objetos de manera segura al Internet, este servicio tiene certificado digital X.509 para procesos de actividades de autenticación de los dispositivos que estén conectados por los protocolos HTTP, MQTT o AMQP (Advanced Message Queuing Protocl). (Microsoft, 2020)

2.4. Comunicaciones

Las tecnologías existentes en el momento que inició IoT no eran las óptimas para la comunicación entre dispositivos; sin embargo, con el paso de los años el mercado se ha innovado mejorando de esta manera el tema de la comunicación (Cendón, 2017).

A continuación, se describe las diferentes tecnologías utilizadas para la comunicación entre dispositivos IoT.

- **Tecnologías de acceso**

Existen tecnologías tradicionales de conectividad inalámbrica como Wifi, conectividad celular (2G y 4G), también existen tecnologías de corto alcance como ZigBee, Z-Wave, 6LowPAN, Bluethooth, NFC (Near field communication).

- **Tecnologías de core**

Existen tecnologías nuevas para una comunicación en IoT, de largo alcance, con un bajo costo y consumo de dispositivos.

- **Sigfox**, un operador que gestiona su red basada en el uso de su tecnología propia, en la figura 5 se observa el proceso desde el dispositivo hasta acceder a la información.

- **LoRa**, tiene como función extender redes privadas, o ser utilizadas por operadores para sus propias redes IoT, es una tecnología alternativa a Sigfox.

Figura 5. Arquitectura de red Sigfox

Tomado de (Sigfox, 2019)

2.5. Seguridad

En la actualidad la mayoría de los sectores económicos, y por no decir todos, tienen o están migrando sus sistemas a un ambiente online, por ejemplo, comercialización de artículos, servicios de aprendizaje continuo, transacciones bancarias o reserva de libros (Gupta, Vashisht, & Singh, 2016).

Bajo esta realidad se presenta un nuevo escenario para la humanidad, donde el Internet se encuentra en la vida diaria de las personas las cuales por diferente razón comparten su información pública y privada como nombre, direcciones, número de celular, cuentas bancarias, etc. En base a este contexto cada organización debe hacer lo necesario para salvaguardar la información de sus usuarios frente a diferentes ataques cibernéticos (Gupta et al., 2016).

La seguridad de la información en la actualidad es un requisito fundamental de las organizaciones, donde los datos que protegen se han convertido el activo digital más importante (Mir, Wani, & Ibrahim, 2013).

Este aspecto se relaciona directamente con la privacidad de las personas. Las organizaciones y sus usuarios deben estar totalmente consientes que es una corresponsabilidad tomar en cuenta las directrices para evitar el robo de la

información. Las empresas por su parte pueden proponer políticas de seguridad en la creación de contraseñas hasta la correcta protección de los aspectos de seguridad de los servidores. Los usuarios también deben poner atención en una correcta gestión de sus credenciales. Las organizaciones y sus usuarios deben trabajar de manera conjunta para poder tener una seguridad informática robusta.

2.6. Mecanismos de autentificación

La autenticación es un proceso primordial para verificar que la persona que intenta acceder a un sistema posee el permiso respectivo. El sistema mediante reglas valida que la persona es la que dice ser, este proceso busca cumplir dos de los objetivos de seguridad, que es mantener la confidencialidad e integridad de la información del usuario del sistema, el componente central de la seguridad informática es la autenticación (Almuairfi, Veeraraghavan, & Chilamkurti, 2011).

Al pasar de los años se han propuesto un número significativo de mecanismos de autenticación, los cuales se clasifican en tres categorías que responden a tres preguntas ¿Algo que yo sé? ¿Algo que yo poseo? Y ¿Algo que yo soy? (Lin, Weng, & Huang, 2008) (Almuairfi et al., 2011) (Gao, Liu, Li, & Qiu, 2014) :

Autenticación basada en conocimiento: En esta categoría se encuentran las contraseñas que responden a la pregunta “¿Algo que yo sé?”, es decir, que tengo el conocimiento necesario para poder acceder a un sistema, las contraseñas basadas en texto y las contraseñas gráficas se encuentran en esta clasificación (Gao et al., 2014).

Autenticación basada en token: En esta categoría se encuentran las contraseñas que responden a la pregunta “¿Algo que yo poseo?”, es decir, en donde el usuario del sistema tiene un objeto o “token”, el cual le va a permitir autenticarse en el sistema, como puede ser una tarjeta inteligente, un tag de proximidad, una tarjeta bancaria, etc (Rajarajan, Prabhu, Palanivel, & Karthikeyan, 2014) (Lin et al., 2008).

Autenticación basada en factores biométricos: En esta categoría se encuentran las contraseñas que responden a la pregunta “¿Algo que yo soy?”,

las características biométricas se pueden dividir en dos tipos, los mecanismos de autenticación que se centran en los aspectos físicos, como las huellas digitales, el iris o la retina (Haque et al., 2016). Y los mecanismos de autenticación que se enfocan en los aspectos conductuales, como la forma de firmar, caminar o de escribir en el teclado (Rajarajan et al., 2014).

2.7. Mecanismos de autorización.

La autorización es el proceso en el cual se determina el qué, cómo y cuándo un usuario autenticado puede acceder a los recursos de un servicio. Para el uso de un recurso la autorización puede hacerse de diferentes formas; además, estas autorizaciones siempre deben quedar registradas para controles posteriores.

2.8. Sustracción de credenciales

Las credenciales se presentan como el filtro principal para evitar el ingreso no permitido a los sistemas. Últimamente la sustracción de credenciales es un problema social, cualquier sector económico está propenso a ser vulnerado (Symantec, 2018).

En el 2017 la empresa Watchguard obtuvo las siguientes estadísticas en el ámbito de robo de credenciales (Watchguard, 2017):

- 200 millones de Yahoo, 159 millones de Hotmail y 90 millones de Gmail.
- Se filtraron 375 mil direcciones y contraseñas gubernamentales y más de medio millón de militares.
- Su firewall bloqueo 30.3224.010 diferentes variantes de malware y 6.907.718 ataques a la red.

Por otro lado, Symantec también nos ofrece estadísticas del 2017 en este ámbito (Symantec, 2018):

- Bloquearon un total de 223.066.372 ataques a sitios web, donde en promedio se obtiene que fueron 611.141 ataques diarios
- Dedujeron que 7.710 organizaciones son afectadas por estafas para comprometer a la empresa.

- Incremento el ataque de Phishing comparado al 2016, en donde 1 de cada 53 usuarios recibió un ataque de este tipo.

En el año 2019 se publicó un informe donde se expone que existió un robo masivo de credenciales y se detalla un total de 2.692.818.23 correos y contraseñas (Hunt, 2019).

Como podemos observar esto se ha convertido en una constante lucha, nadie está a salvo de perder su información o que ésta sea utilizada de manera fraudulenta, mientras incrementan los mecanismos de autenticación crecen de manera paralela las técnicas de vulneración.

Es necesario que las organizaciones y usuarios pongan énfasis en este aspecto, ya que es importante que exista una cultura de protección de la información, algo que en la actualidad es un elemento importante.

2.9. Técnicas para sustracción de credenciales.

Si tenemos que poner un punto de partida referente a la sustracción de credenciales podríamos mencionar al mejor hacker del mundo Kevin Mitnick; el cual implemento la primera y más antigua de las técnicas, la ingeniería social, con la que logro engañar a diferentes usuarios autorizados para poder obtener información sensible que le permitió encontrar huecos de seguridad en un sistema (López Grande & Edgardo, 2015), entonces no podemos dar cuenta que el usuario es vulnerable en cualquier espacio o actividad que desarrolla.

En la actualidad existe un número alto de técnicas de sustracción de credenciales, cada una presenta características diferentes, entre estas técnicas tenemos:

- **Ingeniería social:** El atacante interactúa con la víctima, la cual es un usuario autorizado del sistema, en la búsqueda de ganar su confianza, persuadirlo o engañarlo para obtener la información necesaria para adivinar las credenciales de acceso al sistema o encontrar una brecha

de seguridad que le permita acceder de manera fraudulenta (Lashkari, Manaf, & Masrom, 2011).

- **Baiting**: Consiste en que el atacante deja una memoria externa, la cual contiene algún tipo de virus, cerca del computador de la víctima, para que cuando la víctima lo encuentre, lo conecte en su computador y automáticamente el virus se inyecte en el sistema y le de algún tipo de control al atacante (López Grande & Edgardo, 2015).
- **Phising:** Conocido como suplantación de identidad, ya que, el atacante mediante un medio de comunicación como correo electrónico o llamada telefónica se hace pasar por una empresa legítima, como un banco o entidad del gobierno, y engaña a la víctima, para que le proporcione las credenciales o la información para acceder al sistema, generalmente suelen usar sitios fraudulentos que tienen un parecido bastante alto al original (Gao et al., 2014).
- **Shoulder-surfing:** El atacante se ubica en una posición que le permita tener una visión directa del computador de la víctima, puede también poseer un dispositivo de grabación o visión lejana, con el objetivo de que cuando el usuario vaya a autenticarse en un sistema el atacante pueda ver sus credenciales de acceso y de esa manera obtenerlas (Goutham, Kim, & Yoo, 2014).
- **Keylogger:** Es un tipo de malware que se enfoca en capturar todas las pulsaciones de teclado, mediante el uso de un tipo de software o hardware (Kolekar & Vaidya, 2016).
- **Brute Foce Attack – Dictionary Attack:** Son ataques de búsqueda exhaustiva, consiste en que prueban diferentes contraseñas o palabras de una base de datos, de manera sistemática, hasta encontrar la que les permita acceder al sistema (Rajarajan et al., 2014).
- **Spyware Attack:** El ataque consiste en instalar una aplicación en la computadora de la víctima con el objetivo de grabar todo lo que está haciendo como el movimiento del ratón o las pulsaciones de las teclas, esta aplicación tiene la capacidad de almacenar esa información y enviarla al atacante (Lashkari et al., 2011).

- **Man-in-middle attack:** El atacante intercepta la sesión o el canal de comunicación entre el usuario y el servidor, para poder extraer la información necesaria que le permita acceder de manera no autorizada a un sistema (Almuairfi et al., 2011).

- **Replay Attack:** El atacante graba la secuencia de mensajes entre el usuario y el servidor, para posteriormente reproducirlas en el servidor para obtener acceso al sistema (Goutham et al., 2014).

2.10. Protocolos

2.10.1. MQTT

MQTT es un protocolo encargado de la mensajería, tiene tres conceptos fundamentales en la seguridad que son identidad, autenticación y autorización. La identidad consiste en dar nombre al cliente que se va a autorizar y dar autorización. La autenticación consiste en probar la identidad del cliente y la autorización consiste en gestionar los derechos que se otorgan al cliente (IBM,2020).

La autorización no forma parte de Protocolo MQTT. La proporcionan los servidores MQTT. Lo que se autoriza depende de lo que hace el servidor. Los servidores MQTT son intermediarios de publicación/suscripción, y unas útiles reglas de autorización de MQTT controlan los clientes que se pueden conectar al servidor y los temas en los que un cliente puede publicar o suscribirse. Si un cliente MQTT puede administrar el servidor, hay otras reglas de autorización que controlan los clientes que pueden administrar los distintos aspectos del servidor (IBM,2020).

2.10.2. OAuth 2.0

OAuth 2.0 es un framework de autorización que permite al usuario compartir la información de un punto A (proveedor de servicio) a un punto B (consumidor), el usuario otorga acceso limitado a sus recursos sin exponer sus credenciales, se trata de interactuar con datos protegidos sin compartir datos de contraseñas,

utiliza tokens de autorización para probar una identidad entre los consumidores y proveedores de servicios (Sobers, 2020).

Suministra flujos de autorización para aplicaciones web de dispositivos móviles o de escritorio; es un mecanismo utilizado por grandes empresas para permitir a los usuarios compartir información sobre sus cuentas con aplicaciones de terceros o sitios web (auth0, 2018).

En la figura 6 se presenta el proceso de comunicación entre los roles y flujo que intervienen en el framework OAuth 2.0. para poder acceder a los recursos solicitados.

Figura 6. Roles y Flujo de OAuth 2.0

Tomado de (Jenkov, 2014)

2.10.3. OAuth Proof of Possesion(PoP)

Tiene como objetivo demostrar que el presentador de JWT (Json Web Token) tiene en su posesión la clave de Proof of Possesion (PoP) y que el destinatario puede confirmar criptográficamente la prueba de posesión de la clave emitida por el presentador. En la figura 7 se muestra el proceso de Proof of Possesion, Jones (2016) explica que intervienen los siguientes actores:

- Editor (Issuer): Es quien crea el JWT y enlaza la clave de prueba de posesión.

- Presentador (Presenter): Es quien debe demostrar la posesión de una clave privada o secreta con una firma asimétrica o simétrica, respectivamente, a un destinatario.
- Receptor (Recipient): Es el destinatario que recibe el JWT y quien contiene la clave del comprobante de posesión del presentador.

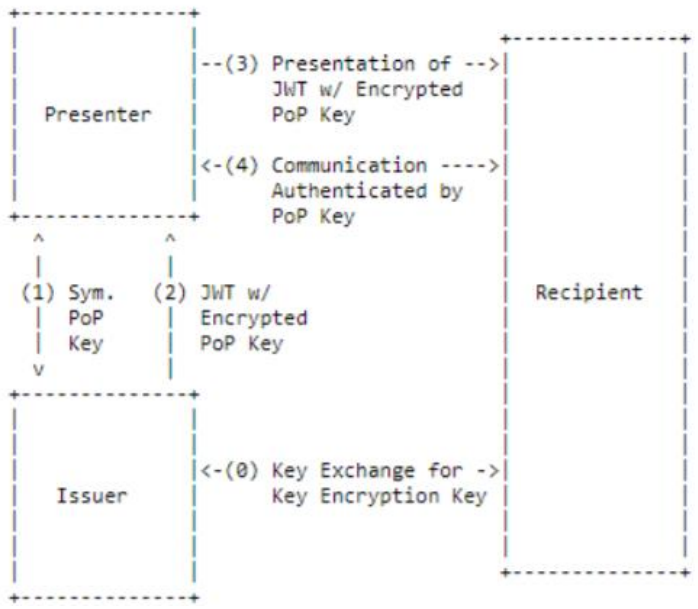


Figura 7. Proof of Possession

Tomado de (Jones, 2016)

2.10.4. Json Web Tokens (JWT)

Es un estándar basado en Json que sirve para crear tokens de acceso entre dos aplicaciones (Cliente y Servidor). Los JWT son firmados digitalmente con un algoritmo de cifrado pero no está encriptado por lo tanto, el uso de HTTPS es obligatorio al almacenar datos de usuario, el objetivo de un JWT no es cifrar los datos para que no puedan ser leídos sino para que la parte receptora pueda confiar que los datos recibidos no se modificaron por ningún intermediario durante el transporte. Un JWT se compone de tres partes Cabecera, Payload y Firma separadas por un punto (Jones M. , 2015).

- **Header:** Se define el algoritmo utilizado y el tipo de JWT

- **Payload**: Se compone por Claims que representan los atributos del token como, por ejemplo: fecha de creación del token, fecha de expiración, emisor, asunto del token, entre otros.
- **Signature**: Es la más importante del token, sirve para detectar la manipulación no autorizada de un token, es la parte que hace que sea seguro, se genera en base a la combinación del header y payload cifrados con una clave secreta.

2.10.5. JWK (Json Web Key)

Una clave web JWK es una estructura de datos de JavaScript (JSON) que representa una clave criptográfica. La representación del conjunto de JWKs se denomina JWK Set JSON. La construcción de una clave JWK se realiza mediante la definición de un conjunto de parámetros que son declarados acorde la necesidad del usuario. A continuación, se presenta algunas de las propiedades que se pueden establecer en un JWK (Jones, 2015).

- **“kty” (Key Type) Parameter:** Identifica el algoritmo criptográfico utilizado, existen 2 tipos RSA (Rivest, Shamir y Adleman).
- **"use" (Public Key Use) Parameter:** Indica si la clave pública es utiliza para cifrar datos o verificar la firma de los datos. Existe dos tipos: "sig" (firma) o "enc" (cifrado).
- **“key_ops” (Key Operations) Parameter:** Identifica la operación para la cúal va a ser utilizada la clave.
- **“alg” (Algorithm) Parameter:** Es el algoritmo de cadena ASCII que se define para usar con la llave.
- **"kid" (Key ID) Parameter:** Representa el ID de la clave y se utiliza para coincidir con una clave específica. Este parámetro es utilizado en Json Web Sign (JWS.)

2.10.6. JWS (Json Web Signature)

La firma web JSON es la parte más importante de un JWT ya que representa la seguridad y contenido protegido de firmas digitales o códigos de autenticación

de mensajes, contiene la prueba que la información no haya sido alterada desde la firma del emisor, proporcionan protección de integridad. JWS representa contenido firmado utilizando estructuras de datos JSON y codificación base64url. Un Json Web Signature se compone de tres partes (Jones, 2012).

- **Encabezado JWS:** Describe el método de firma y los parámetros empleados.
- **Carga útil JWS:** Contenido del mensaje que debe protegerse.
- **Firma JWS:** Garantiza la integridad tanto del encabezado y carga útil JWS.

La siguiente figura presenta un ejemplo de un JWS, donde, typ: representa el tipo de token, alg: representa el tipo de algoritmo empleado y finalmente se muestra la firma digital.

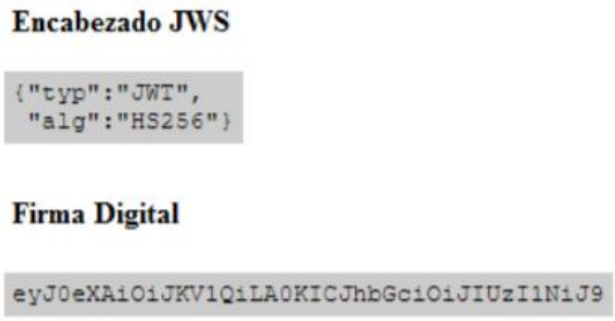


Figura 8. JWS

Tomado de (Jones, 2012)

2.10.7. HTTP vs HTTPS

Para diferenciar los términos HTTP y HTTPS es importante primero conocer el significado de las siglas HTTP y el objetivo de este, del inglés Hyper Text Transport Protocol, es un proceso con la función de petición-respuesta y está diseñado para asegurarse que los datos transportados entre distintos equipos (Cliente y Servidor) que conforman una red, lleguen de manera correcta durante el proceso de petición–respuesta. En este proceso el Cliente es quien emite la petición y el Servidor la respuesta. Una vez identificado el termino HTTP, se puede continuar con las siglas HTTPS del inglés, Hyper Text Transport Protocol

Secure, este se encuentra basado en los protocolos HTTPS y SSL (Secure Socket Layer) / TLS (Transport Layer Security), cuyo objetivo es mantener segura y cifrada la información que se transporta entre cliente y servidor, de esta manera si un tercer actor intentara descifrar la información enviada no le será posible como se visualiza en la figura 9.

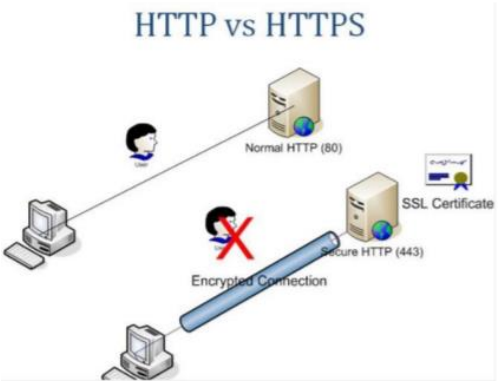


Figura 9. HTTP vs HTTPS

Tomado de (Losada Perez, 2015)

Para que en HTTPS la información permanezca segura es necesario activar el esquema de “Certificado” que la autoridad firma (Losada Perez, 2015). A continuación, se presenta un cuadro de las diferencias, sabiendo que HTTPS se caracteriza por brindar seguridad mediante un certificado, en la tabla 3 se muestra las diferencias más detalladas.

Tabla 3.

HTTP vs HTTPS

http	**HTTPS**
URL comienza con http://	URL comienza con https://
Sin Garantía	Asegurado

Funciona a nivel de aplicación	Funciona a nivel de transporte
Sin cifrado	Con certificado
No hay certificados requeridos	Certificado prescrito

2.10.8. TCP (Transmission Control Protocol)

Este protocolo ésta orientado a la conexión que permite la transmisión de datos entre dos máquinas y que cada una de ellas controle el estado de la transmisión.

Cuando se usa este protocolo las aplicaciones pueden establecer conexión de forma segura, la máquina emisora (cliente) es la que solicita la conexión y la máquina receptora (servidor) es quien recibe la petición. Por eso se dice que es en un entorno Cliente-Servidor. Una de las funciones que permite el protocolo es la multiplexación que es la transferencia de datos a diversas aplicaciones por la misma línea. En la figura 10 y 11 se presentan ejemplos de la función (Villagómez, 2017).

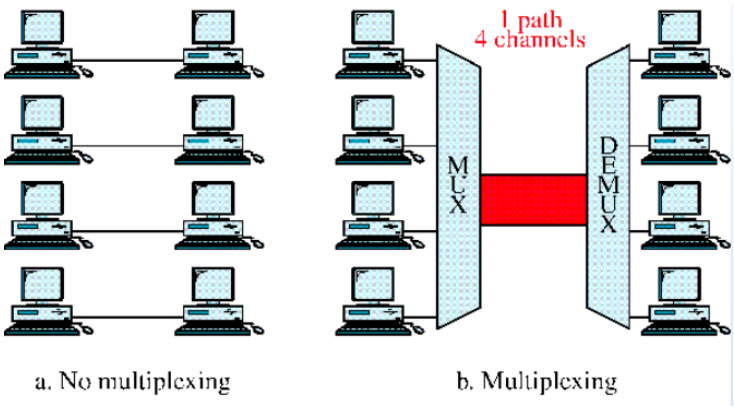


Figura 10. No Multiplexación vs Multiplexación

Tomado de (Villagómez, 2017)

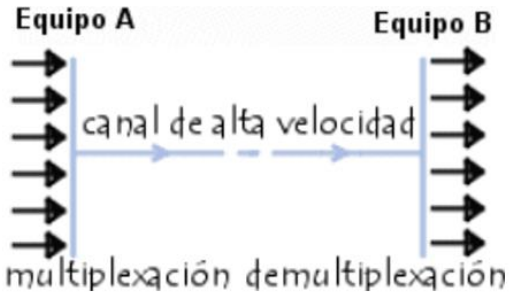


Figura 11. Función de multiplexación

Tomado de (Villagómez, 2017)

2.10.9. TLS (Transport Layer Security)

Es un protocolo que permite la comunicación segura entre dos aplicaciones Cliente-Servidor a través de Internet, impide que durante la comunicación los datos sean manipulados, falsificados o que atacantes vean los datos. Para que un canal de comunicación sea seguro debe presentar las siguientes propiedades y componentes principales (Rescorla, 2018).

Propiedades

- **Autenticación**: El Servidor debe estar autenticado, el Cliente es opcional.
- **Confidencialidad**: Los datos enviados son solo visibles para los puntos finales.
- **Integridad**: Los datos no pueden ser manipulados por atacantes.

Componentes

- **Protocolo de Enlace:** Sirve para autenticar, negociar, comunicar y establecer parámetros criptográficos y claves compartidas entre las dos partes Cliente-Servidor, en caso de un ataque no se puede forzar la comunicación entre las partes.
- **Protocolo de Registro:** Usa parámetros para proteger el tráfico entre ambas partes; divide la información en varias partes durante el tráfico y cada registro se encuentra protegido de manera independiente utilizando claves.

2.10.10. SSL (Secure Socket Layer)

SSL es un protocolo de seguridad que garantiza que el transporte de datos entre un cliente y servidor web se realice de forma segura e integra y lleguen al servidor correcto. Los datos enviados son cifrados o encriptados de manera que se asegura no puedan ser leídos por un tercer actor. En la ejecución de SSL se crean caminos de cifrado para las sesiones privadas y la clave pública puede ser compartida con cualquier actor, por esta razón se utilizan dos claves: la clave pública para encriptar la información y la privada para desencriptar la información. Es importante conocer que para utilizar un certificado SSL en una página web, el servidor de Internet debe soportar SSL (certsuperior, 2016).

CAPÍTULO III. Análisis

En este capítulo se presenta algunos modelos de ecosistemas IoT de algunas empresas que brindan este servicio. Se cita estos casos con la intención de conocer con más detalle los modelos de autorización manejados por estas empresas.

3.1. Azure

Azure IoT es una plataforma que brinda diferentes recursos tecnológicos a las empresas, basadas en años de experiencia empresarial y su diseño lo hace accesible para que todas las organizaciones. (Microsoft,2020)

3.1.1. Modelo IoT

Microsoft presenta una arquitectura divida en 3 grupos: presentación, conectividad y procesamiento de datos y análisis. Los datos recolectados se envían a la puerta de enlace donde están disponibles para un tratamiento según la necesidad a través de otros servicios llamados Back-end donde entran los datos a diferentes aplicaciones por medio de un dispositivo de presentación, como se observa en la figura 12: (Microsoft,2020).

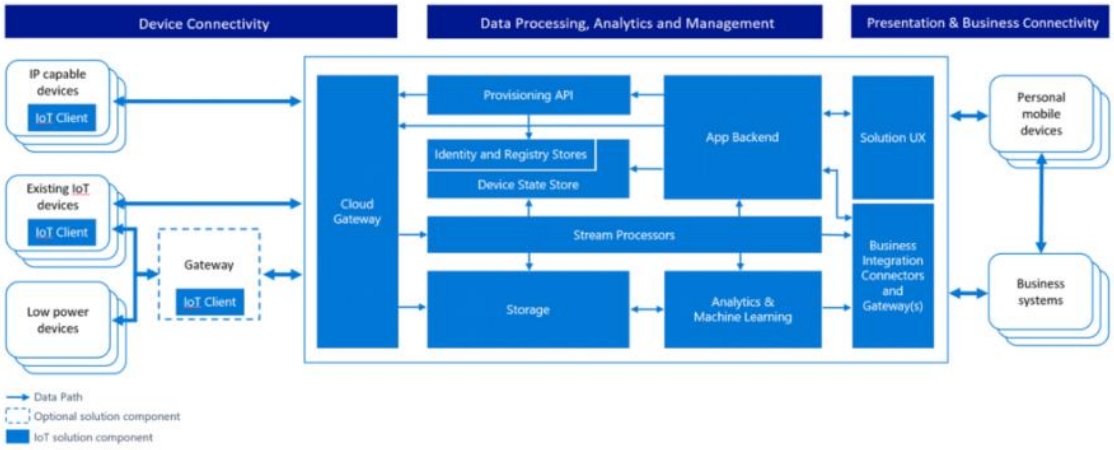


Figura 12. Servicio de Microsoft Azure

Adaptado de (Microsoft Azure, 2020)

3.1.2. Modelo de Seguridad

Microsoft por medio de Azure, soporta Aplicación IoT, el servicio de Microsoft se conoce como Azure IoT Hub que admite conectar a los objetos de forma segura a Internet. Admite certificado digital X.509 para acciones de autenticación de los dispositivos por medio de los protocolos HTTP, MQTT, IoT Hub mediante certificados y con un identificador son asociados a Internet con la clave privada, estos certificados son validados y generados por una certificadora con el fin de que estos dispositivos se puedan autenticar en IoT Hub (Arias, 2019).

3.1.3. Proceso de autenticación y autorización

El proceso de autenticación en Azure IoT Hub está basada en tokens de seguridad para autenticar dispositivos y servicios, de esta forma no envía claves de acceso en la conexión, sino que son enviados por los protocolos mencionados en la sección anterior, los cuales tienen diferentes formas de transportarlos y tienen un periodo de validez.

Para tener más claro los pasos de autenticación en Azur IoT Hub usaremos como ejemplo el acceso desde un celular a una cámara de vigilancia inteligente, a continuación, veremos los pasos en la figura 13 para que un dispositivo trabaje con el modelo de servicio de tokens:

1. El celular para acceder a la cámara de vigilancia solicita un token firmado al servicio de tokens.
2. El servicio de token devuelve un token un id o identificador como dispositivo autenticado, con una validez previamente programada.
3. El celular usa el token directamente con el centro de IoT (servicio solicitado)

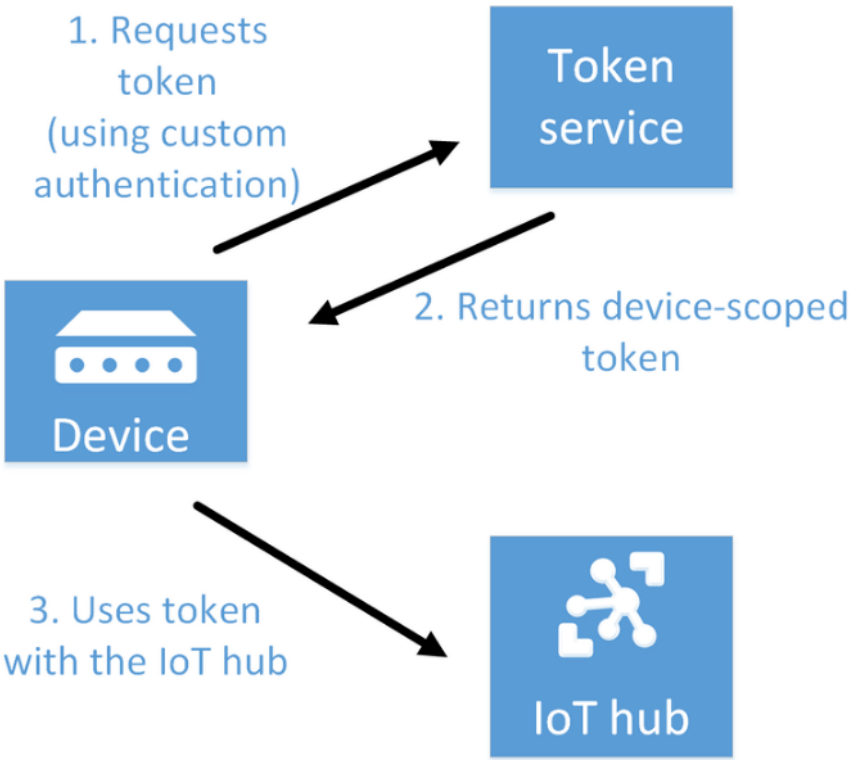


Figura 13. Proceso de autenticación.

Adaptado de (Microsoft Azure, 2020)

3.2. Amazon AWS

Amazon IoT tiene servicios sólidos y amplios, desde el dispositivo hasta la nube. Este es un proveedor que combina el análisis y administración de datos fáciles de usar en datos IoT ruidosos. (Aws,2020)

3.2.1. Modelo IoT

AWS IoT se basa en el protocolo MQTT. De esta forma las “cosas” notifican su estado a través de mensajes a través de un puente con referencia a un tema específico, y quien tiene el permiso para recibirlos los mensajes están suscritos al tema. (AWS, 2020).

La estructura AWS IoT, posee un SDK para dispositivos con AWS IoT. Un paquete de desarrollo que admite la conexión, autenticación de mensajes de dispositivos o aplicaciones de acuerdo con el protocolo. La comunicación entre los dispositivos y el Cloud se da gracias a la puerta de enlace; los datos entrantes de los dispositivos son procesados por el motor de reglas y sus estados son guardados en una Device Shadow, para ser manipulados por alguna aplicación, como se observa en la figura 14. (AWS, 2020).

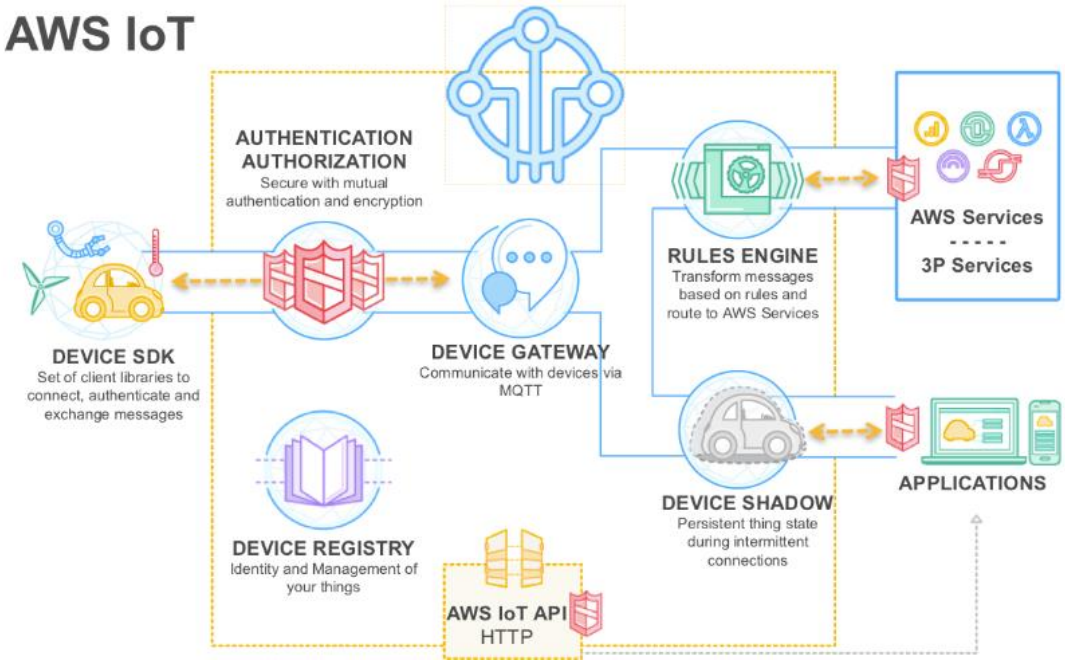


Figura 14. AWS IOT

Adaptado de (Amazon, 2018).

3.2.2. Modelo de Seguridad

Amazon AWS, por medio de sus servicios, efectúa métodos de autenticación y seguridad para sistemas IoT, usa TLS para cifrar todo el tráfico que pasa por su infraestructura, de esta forma los dispositivos conectados son identificados por medio de certificados X509, sabiendo que es un método más seguro que sistemas de autenticación por credenciales o tokens; este método de certificación digital usa en un sistema de cifrado asimétricos que le ofrece grabar las claves privadas en un almacenamiento seguro del dispositivo.

3.2.3. Proceso de autenticación y autorización

El proceso de autenticación en AWS IoT puede utilizar certificados X.509 y el uso de tokens para acceder a Amazon IoT mediante los protocolos de autenticación. (Amazon,2020)

En la figura 15 podremos ver los pasos que un dispositivo realiza para la autenticación en AWS IoT.

1. Los dispositivos hacen una solicitud HTTPS al proveedor para obtener un token de seguridad, este proceso tiene un certificado X.509.
2. El servidor de credenciales devuelve una solicitud al módulo de autorización y autenticación, para verificar que el dispositivo tiene permiso.
3. Si el certificado en un dispositivo es válido y tiene permiso para obtener un token de seguridad, el módulo de autorización indica que el proceso se realizó correctamente.
4. Después de validar el certificado, el módulo de STS (Security Token Service) para asumir el rol de IAM (Identity and Access Management).
5. AWS STS devuelve un token temporal con acceso limitado al proveedor de credenciales.
6. El proveedor de credenciales envía un token al dispositivo.
7. Los dispositivos tienen un token de seguridad para verificar la solicitud de AWS con ayuda de AWS Signature Version 4.

8. Para validar la firma el servicio se llama a IAM y permite el acceso de la solicitud asociado al rol de IAM que creo el proveedor de credenciales.
9. Si el servicio IAM verifica que la firma es correcta y autoriza la solicitud.

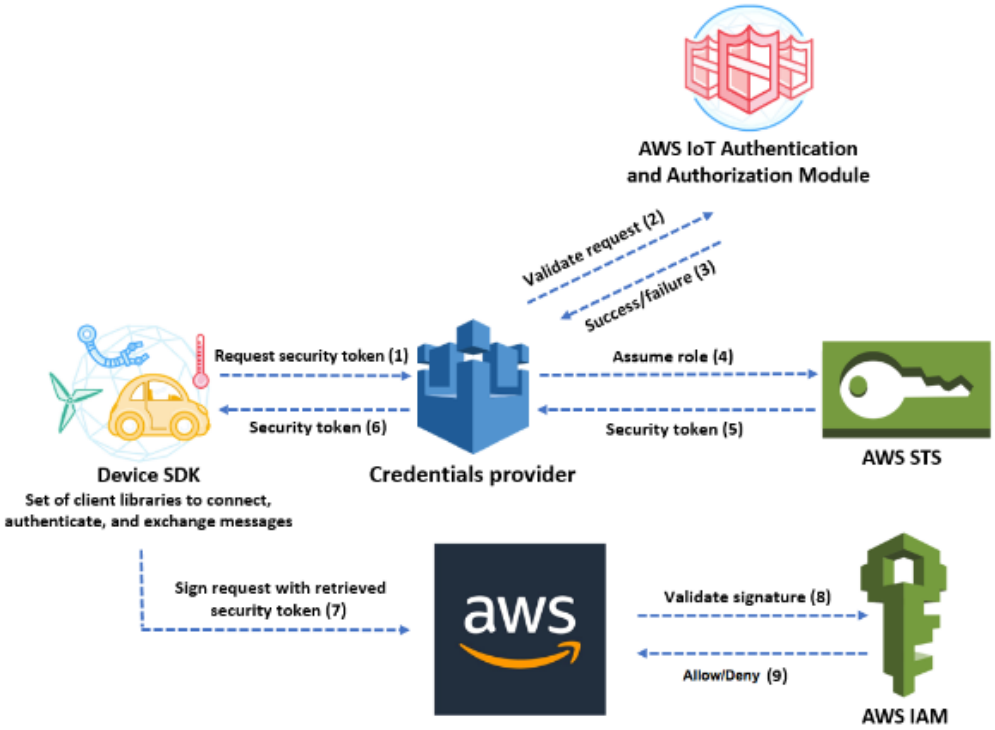


Figura 15. Autenticación AWS

Tomado de (Amazon, 2018).

3.3. Google IoT

Google Cloud IoT es un grupo de herramientas para analizar datos, procesar, almacenar y conectar en la nube. Cloud IoT consta de servicios administrados y escalables en la nube, con un grupo de software integrado para cumplir tareas de procesamiento con una capacidad autónoma de aprendizaje para cumplir las necesidades de IoT. (Cloud,2020).

3.3.1. Modelo IoT

Google IoT tiene un servicio administrado para la manipulación de dispositivos, eso incluye el registro, la autorización y autenticación en los recursos de Cloud.

Cuando un proyecto de IoT funciona, los dispositivos que estén conectados producirán una cantidad inmensa de datos, permitiendo que el procesamiento y manejo de información sea eficaz, sin demoras y en concurrencia permanente. La escalabilidad es una característica fundamental en la manipulación de datos producidos ya que permite potenciar el espacio de almacenamiento de datos y evita la saturación o perdida de estos.

Cuando se trata de analizar, procesar y almacenar macrodatos la nube es la mejor opción por las características mencionadas y porque no tiene competidor. (Cloud,2020) En la figura 16, se observa las etapas de administración en Cloud desde la generación de datos por sensores o diferentes dispositivos hasta llegar a las diferentes necesidades de procesamiento:

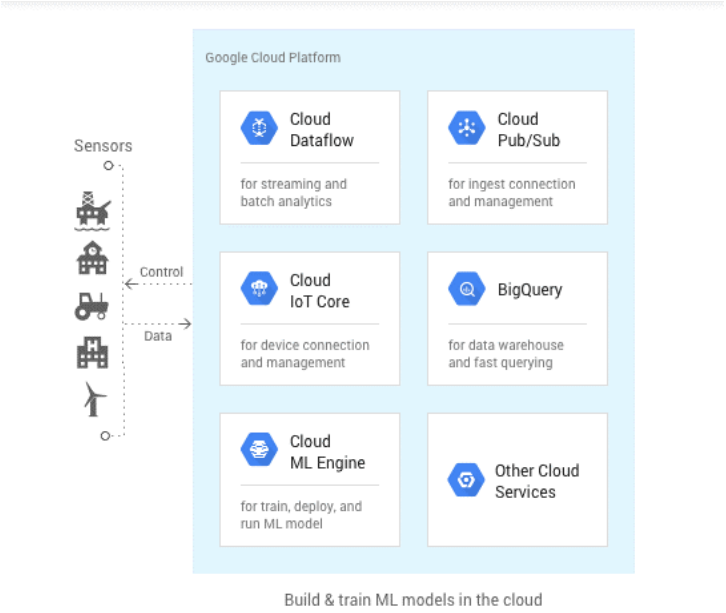


Figura 16. Google IoT

Tomado de (Cloud, 2020).

3.3.2. Modelo de Seguridad

Google IoT (Cloud IoT Core) utiliza MQTT y HTTPS como protocolos certificados para la conexión y comunicación de dispositivos, generando un puente que permite el acceso a los servicios de manera segura. En la figura 17 podemos

observar el proceso de conexión usando el puente MQTT, desde el dispositivo hasta el Cloud IoT para su uso.

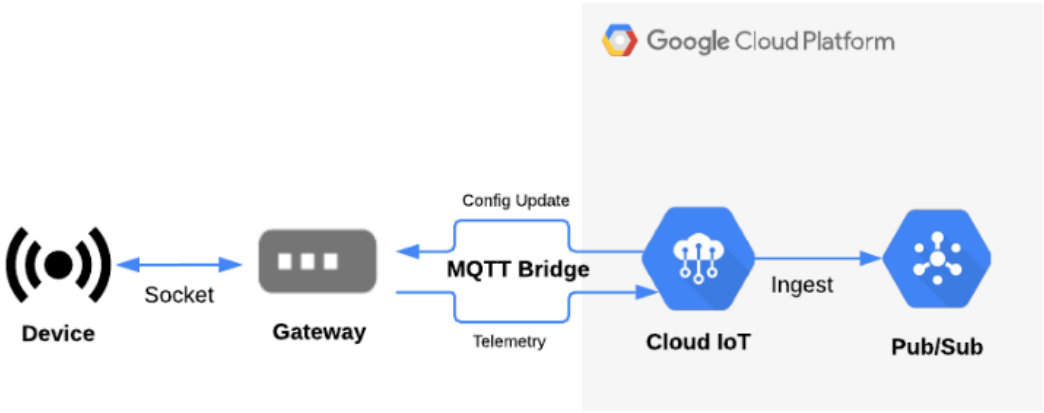


Figura 17. Puente MQTT

Tomado de (Cloud, 2020).

3.3.3. Proceso de autenticación y autorización

Cloud IoT Core utiliza la clave pública o asimétrica.

- El dispositivo utiliza una clave privada para la firma un token web JSON, de esta forma el token pasa a Cloud IoT como prueba de la identificación del dispositivo.
- El servicio utiliza la clave pública del dispositivo que es cargada previamente para verificar su identidad.

Además de usar tokens esta plataforma ocupa el protocolo OAuth 2.0 para autorizar cuentas de usuarios y de servicio.

Si se desea compilar una aplicación con las API (Application Programming Interface) de Google Cloud existen unos pasos a seguir como:

- Seleccionar las bibliotecas cliente de cloud proporcionadas.
- Determinar el flujo adecuado de autenticación para la aplicación.
- Crear o buscar las credenciales adecuadas para la aplicación.

- Pasar las credenciales de la aplicación a las bibliotecas cliente durante su ejecución.

Para el uso del OAuth se tiene unas recomendaciones como:

- Requisito: Acceder a los datos privados en nombre de un usuario final
- Recomendación: Cliente OAuth2.0
- Comentario: Un cliente de OAuth 2.0 identifica la aplicación y permite que los usuarios finales autentiquen la aplicación con Google. Permite que la aplicación acceda a las API de Google Cloud en nombre del usuario final.

3.4. Amazon AWS IoT vs Microsoft Azure IoT vs Google IoT

A continuación, en la tabla 4 se presenta un resumen de lo analizado en las secciones anteriores, este contiene diferentes características entre las cuales están; protocolos, patrones de comunicación, plataformas, SDK/lenguaje, métodos de seguridad, métodos de autenticación. Los puntos analizados nos dan una breve idea de cómo se diferencian estas plataformas. El uso del protocolo OAuth2.0 por parte de Google IoT dentro de su proceso de autorización en un ecosistema IoT es la característica que más llama la atención en este análisis.

Tabla 4.

AWS IoT vs Azure IoT vs Google IoT

	Amazon IoT	**Azure IoT**	**Google IoT (Cloud IoT Core)**
Protocolos que usan.	• HTTPS • WebSockets • MQTT	• HTTP • AMQP • MQTT • Protocolos personalizados	• HTTP • OAuth2.0 • MQTT

Patrones de Comunicación	Mando y telemetría	Mando y Telemetría	Mando y telemetría
Plataformas	Microchip, Intel, Raspberry Pi, Intel entre otras	Intel, Raspberry Pi, Freescale, Texas Instruments, entre otras	Raspberry Pi
SDK/Lenguaje	• C • JavaScript • Arduino	• Java • C • .Net • Nodejs	Java entre otros
Métodos de Seguridad	Protocolo TLS	Protocolo TLS	Protocolo TLS
Métodos de Autenticación	AWS IAM o AWS, Certificado X.509, autenticación basado en conexiones MQTT	Diferentes métodos entre ellos: token de SAS, etc.	Diferentes métodos entre ellos: Tokens, etc.

3.5. Análisis de protocolos actuales de autorización

En esta sección se hará un análisis previo de protocolos de autorización actuales, los cuales darán una idea amplia de cómo se maneja en la actualidad el proceso de autorización.

3.5.1. OAuth 2.0

En la figura 18 se explica el flujo del protocolo OAuth que tiene para acceder a los recursos paso a paso desde la solicitud de autorización hasta el acceso de los recursos.

Pasos del protocolo:

1. El cliente solicita autorización al propietario del recurso.
2. El usuario otorga la autorización.
3. El cliente pide un token al servidor de autorización, presentando sus credenciales otorgadas.
4. El servidor de autorización envía un token de acceso si las credenciales son válidas.
5. El cliente envía el token pidiendo acceso al servidor de recursos.
6. Si el token es válido el servidor de recursos permite el acceso y manipulación a los recursos.

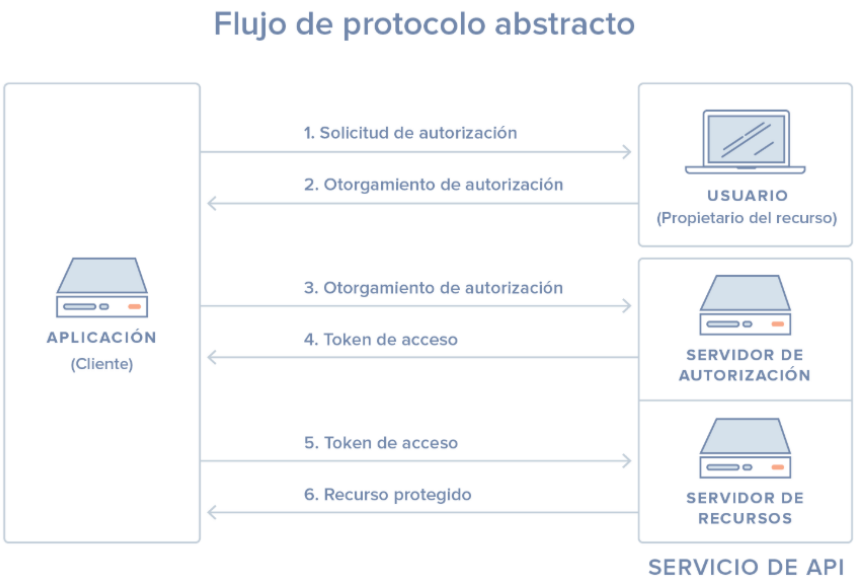


Figura 18. Flujo general de OAuth 2.0

Tomado de (Anicas, 2018)

3.5.2. MQTT

En la figura 19 observamos el flujo del protocolo MQTT utilizado para la autorización por algunas empresas.

Pasos:

1. El cliente se comunica con el mediador. Se puede registrar en cualquier tema como servicios, dispositivos. La conexión puede ser TCP/IP simple o conexión TLS.
2. El cliente publica el mensaje, sobre un tema inscrito, enviando el mensaje y el tema al mediador.
3. El mediador redirecciona el mensaje a todos los clientes que se encuentren en la lista del tema

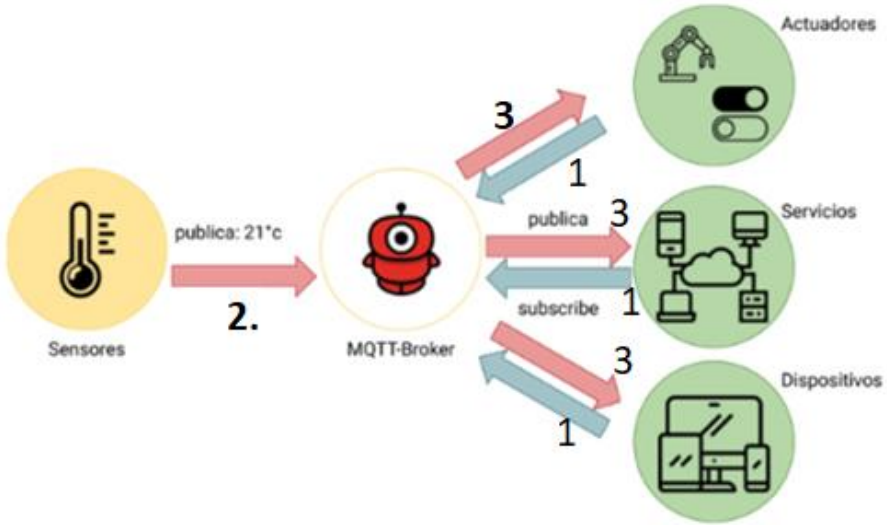


Figura 19.MQTT

Adaptado de (murkyrobot,2020)

3.5.3. AMQP

Este protocolo consiste en generar tres acciones claves en el proceso se autorización: conexión y sesión, enlaces y traslados. Es un protocolo abierto en la capa de aplicación del modelo OSI y se caracteriza por que está orientado a la mensajería.

Conexiones y sesiones

AMQP como protocolo permite el uso de la multiplexación cuando llama a los contenedores de comunicación, es decir, el cliente puede recibir y enviar colas a través de la misma conexión como se muestra en la figura 20 que se crea un canal de comunicación para la sesión.

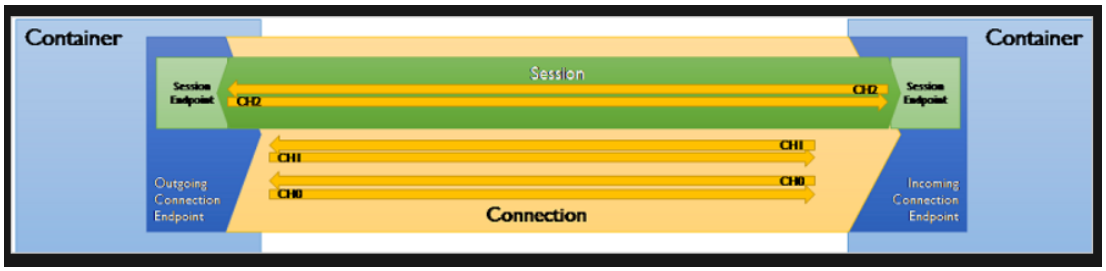


Figura 20. Conexiones y sesiones

Tomado de (Microsoft,2020)

Enlaces

AMQP trasfiere sus mensajes a través de un enlace; es una ruta de comunicación, como se observa en la figura 21, donde se crea con cada sesión un enlace utilizado para trasferir mensajes en una dirección manteniendo segura la información del usuario.

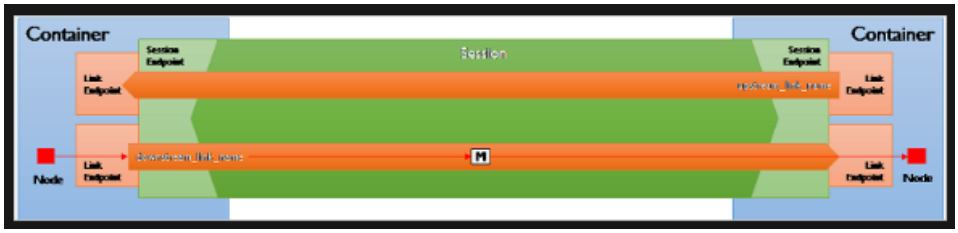


Figura 21. Enlaces

Tomado de (Microsoft,2020)

Traslados

El protocolo AMQP usa una trasferencia performativa que mueve un mensaje del emisor al receptor a través del enlace. La transferencia se termina o liquida, lo que significa que las dos partes han establecido un entendimiento del resultado de esa trasferencia. En la figura 22 se muestra este proceso.

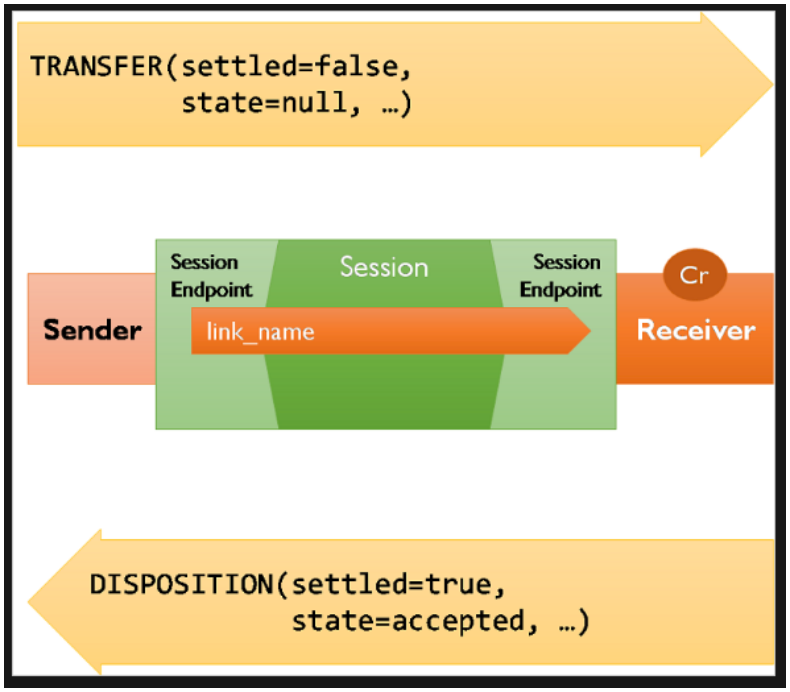


Figura 22. Traslados

Tomado de (Microsoft,2020)

3.5.4. Cuadros Resumen

En la tabla 5 se observa un cuadro de resumen de los dos protocolos más importantes en la etapa de autorización, a continuación, se detalla cada protocolo.

Tabla 5.

Protocolos

Protocolo	Descripción

Oauth 2.0	Su objetivo principal es la autorización para el acceso a un recurso, este protocolo tiene la facilidad de ser usado de diferentes formas para poder cumplir su propósito
Message Queue Telemetry Transport (MQTT)	Es un protocolo de maquina a máquina, también se lo conoce como puente y debido al bajo el consumo de ancho de banda se puede utilizar en dispositivos IoT.

En la tabla 6 se muestran las características de los diferentes protocolos como capa de modelo OSI, objetivo principal y secundario. De la tabla se puede observar que el protocolo OAuth 2.0 es más adecuado porque su objetivo principal y secundario se enfocan en la autorización y especializado en el proceso de autorización en un ecosistema IoT.

Tabla 6.

Características Protocolos

Característica	**Oauth 2.0**	**MQTT**	**AMQP**
Capa del modelo OSI	En la capa 7	Capas superiores 5	En la capa 7
Objetivo principal	Autorización	Transporte	Transporte
Objetivo secundario	Autorización	Autorización	Autorización

Mediante el análisis realizado anteriormente se pudo deducir que el mejor protocolo para autorización es OAuth 2.0, que es usado por la plataforma Google

IoT, esta usa este protocolo para el acceso a sus diferentes servicios, en la figura 24 se observa el uso de OAuth en Google a continuación, se detalla el proceso.

- Se obtiene las credenciales en la consola API de Google
- Solicita un token de acceso
- Se obtiene un token de acceso al servidor de autorización de Google
- Examina los parámetros de acceso otorgados por el usuario.
- Envía el token de acceso a un API.
- Actualiza el token de acceso, si es necesario.

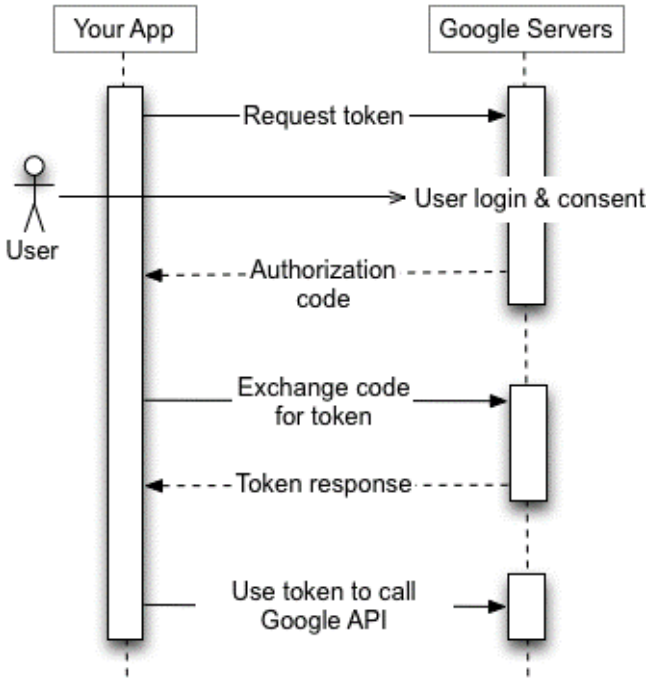


Figura 23. Oauth en google.

Tomado de (Developers Google,2020)

CAPÍTULO IV. Diseño del modelo de autorización

En este capítulo se presenta una propuesta que se basa en un análisis de los requerimientos y diseño del modelo con la información obtenida en los capítulos

anteriores, para esta propuesta se realizará los casos de usos entre otras herramientas que ayudarán a entender el desarrollo de este modelo.

4.1. Propuesta

El análisis realizado en los capítulos anteriores permitió tener un amplio panorama para proponer un modelo de autorización basado en el protocolo OAuth 2.0. Esta propuesta está enfocada para las empresas que quieran mejorar este módulo o dar un nuevo uso a este protocolo, así como para personas que desarrollen aplicaciones en un ecosistema IoT mediante la posibilidad de que este modelo se use como librería en diferentes frameworks.

En la actualidad el protocolo OAuth 2.0 usa cuatro actores principales que son: usuario, aplicación, servidor de autorización y servidor de recursos, los cuales tienen una comunicación entre ellos. En el modelo propuesto se define tres actores que son:

- Cliente: se considera en este modelo al cliente como la unión de dos actores, usuario y aplicación, dado que no hay incidencia en el usuario sino a la aplicación.
- Servidor de autorización: es el encargado de generar tokens para el acceso a los recursos.
- Servidor de recursos: provee al cliente los servicios por los que accede a su información.

La propuesta se basa en una comunicación directa entre los dos servidores para que de esta forma no exista robo de información, sea más segura y confiable la validación del token y su actualización. En la figura 23 se representa una visión general de la propuesta, en donde se comunican los servidores para verificar las credenciales de manera segura entre ellos y no exista la posibilidad de que sea extraviada o robada la clave generada para ingresar a los recursos.

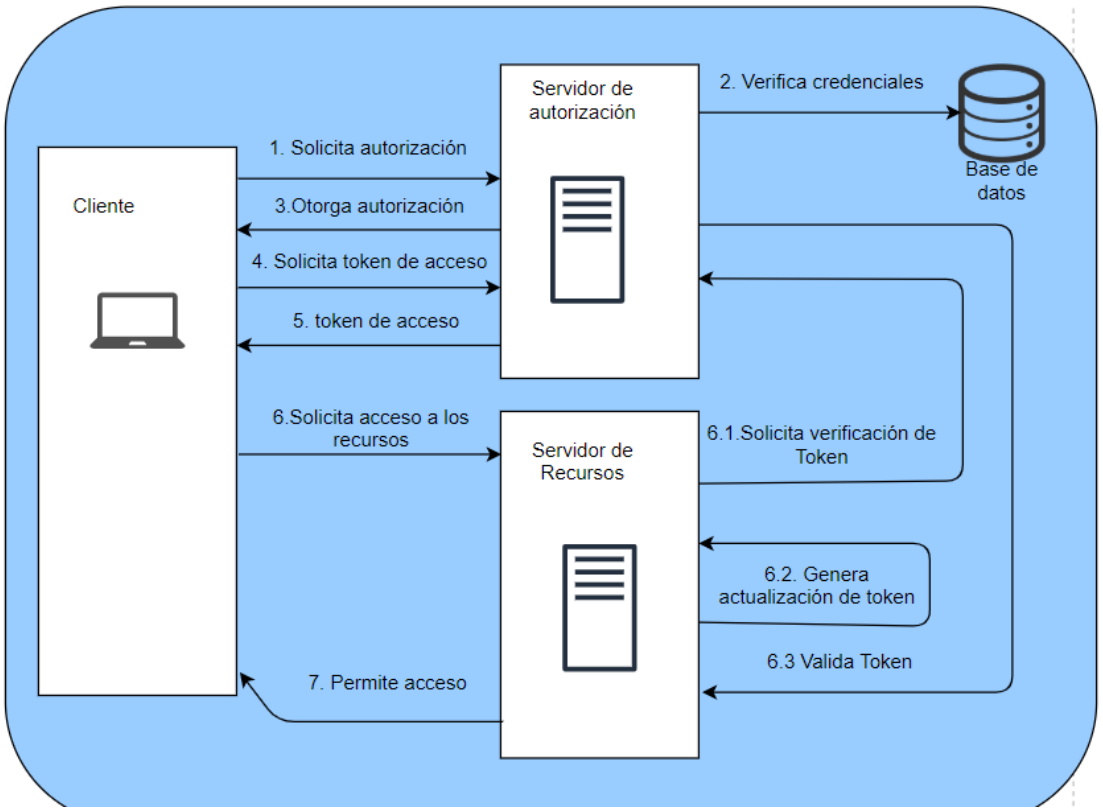


Figura 24. Visión general del modelo de autorización.

4.1.1. Requisitos identificados

Con el objetivo de tener una perspectiva precisa y completa, se ha detectado algunos requisitos importantes para que un dispositivo trabaje en un ecosistema IoT de una marea correcta como:

- Se requiere de conexión a Internet para poder realizar el proceso de autenticación
- Garantizar la confiabilidad y el buen desempeño en el manejo de la concurrencia para todos los usuarios del modelo

4.1.2. Diagramas de Casos de Uso

En el diagrama presentado en la figura 25 nos ayuda a entender el proceso del modelo de autorización propuesto de una manera más exacta, también se conoce los actores del proceso y cómo influyen en cada caso de uso.

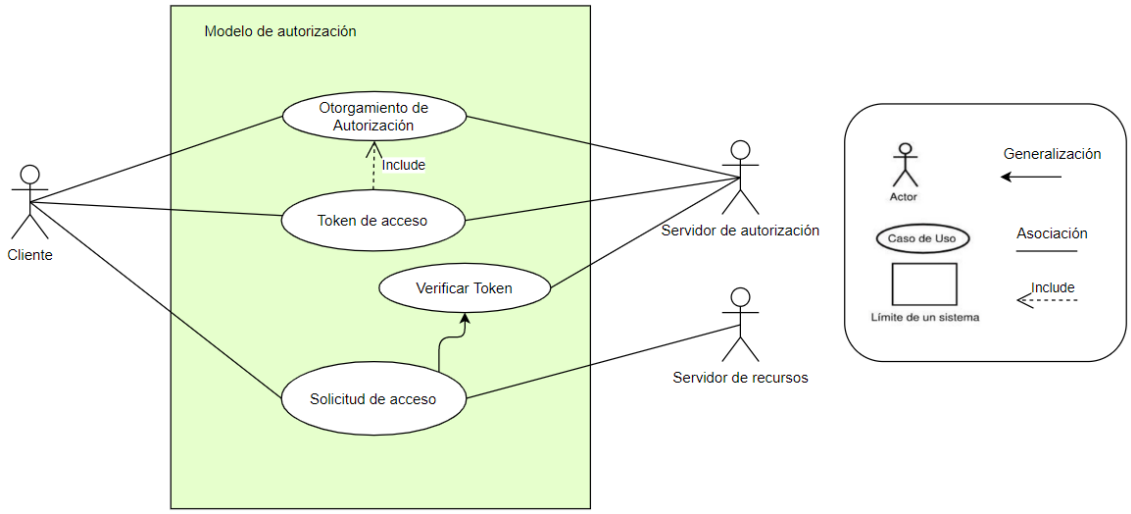


Figura 25. Diagramas Casos de Uso de la propuesta del modelo de autorización.

En el diagrama expuesto se evidencia el siguiente proceso: se considera al otorgamiento de autorización como punto de arranque en el cual interviene el servidor de autorización de forma obligatoria para realizar el proceso de verificación de credenciales, aquí el cliente actúa de forma indirecta al entregar las credenciales; el token de acceso es complementado por el otorgamiento de autorización que sirve para completar el proceso, el servidor de autorización es el encargado de generar el token como actor principal el cual envía al cliente que actúa solamente como receptor; en la solicitud de acceso el cliente envía obligatoriamente el token al servidor de recursos, quien a su vez envía y verifica con el servidor de autorización y permite el acceso al cliente.

4.1.3. Descripción de Casos de Uso

En la tabla 8 hasta la 11 se detalla el proceso de cada caso de uso, de esta forma conocemos como interviene cada actor en cada caso de uso y su proceso interno.

Tabla 7.

RF-001

Número de requisito	RF-001
Nombre de requisito	Otorgamiento de autorización

Actores	Cliente, Servidor de Autorización
Prioridad del requisito	
Descripción	Proceso de validación de la existencia del usuario para poder generar un código de autorización.
Precondición	Registrar credenciales
Flujo	1.Ingresar credenciales de usuarios. 2. Enviar credenciales del cliente al Servidor de Autorización. 3.Validación de credenciales. 4.Generación de código de autorización.
Postcondición	Enviar el código de autorización o la notificación de datos incorrectos al usuario.
Excepciones	Si los datos son correctos genera el código, caso contrario notificar datos incorrectos con mensaje.
Comentarios	

Tabla 8.

RF-002

Número de requisito	RF-002
Nombre de requisito	Token de acceso
Actores	Cliente, Servidor de Autorización
Prioridad del requisito	
Descripción	Validación de la existencia de un código de autorización para la generación de un token de acceso.
Precondición	Obtener el código de autorización
Secuencia normal	1.El cliente envía el código de autorización al Servidor de Autorización. 2.Verificar el código de autorización sea correcta 3.Generar el token de acceso y el token de actualización.
Postcondición	Enviar el token de acceso o la notificación de datos incorrectos al usuario.
Excepciones	Si el código de autorización es correcto genera los tokens, caso contrario notificar invalidez del código de autorización
Comentarios	

Tabla 9.

RF-003

Número de requisito	RF-003
Nombre de requisito	Solicitud de acceso
Actores	Cliente, Servidor de Recursos
Prioridad del requisito	
Descripción	Una vez el token este generado, se muestran los dispositivos IoT a los cuales el cliente tiene acceso
Precondición	Verificación de token
Flujo	1.El cliente envía el token de acceso al Servidor de Recursos. 2.El Servidor de Autorización valida el token de acceso.

	3.Se obtiene la información de los dispositivos del cliente si el token es correcto.
Postcondición	Enviar al cliente los dispositivos IoT que se tiene acceso o las notificaciones de token invalido o expirado.
Excepciones	Si el token ha expirado se enviará un mensaje para que el cliente acceda al token de actualización, caso contrario se notifica la invalidez del token
Comentarios	La generación de tokens de acceso será un ciclo repetitivo cada vez que el token expire (esto dependerá del administrador del software), creando uno nuevo.

Tabla 10.

RF-004

Número de requisito	RF-004
Nombre de requisito	Verificar token
Actores	Servidor de Autorización, Servidor de Recursos
Prioridad del requisito	
Descripción	El token de acceso que fue enviado por el cliente al servidor de recursos será verificado en el Servidor de Autorización si es válido o no
Precondición	Obtener token de acceso
Flujo	1.Enviar el token de acceso al Servidor de Autorización desde el Servidor de Recursos 2.Verificacion del token de acceso 3.Si el token de acceso es correcto, se devolverá la información de los dispositivos.
Postcondición	Enviar al Servidor de Recursos los datos obtenidos
Excepciones	Si los datos son erróneos se envía mensaje de advertencia.
Comentarios	

4.1.4. Diagrama de secuencia

En la figura 26 se puede observar un diagrama de secuencia que indica el modelo propuesto de una forma más clara, en la cual se puede ver como se propone usar el protocolo OAuth 2.0. donde los Servidores de Recursos y Autorización tienen una comunicación directa con el cliente. No obstante, en la propuesta también presentada en esta figura maneja una comunicación directa entre los servidores para la verificación del token sin la intervención del cliente, de esta manera cuidamos el robo del token o suplantación de identidad. Este

proceso se evidencia desde el punto 6.1 al 6.3 de la figura, los cuales son los pasos encargados de la comunicación directa entre servidores.

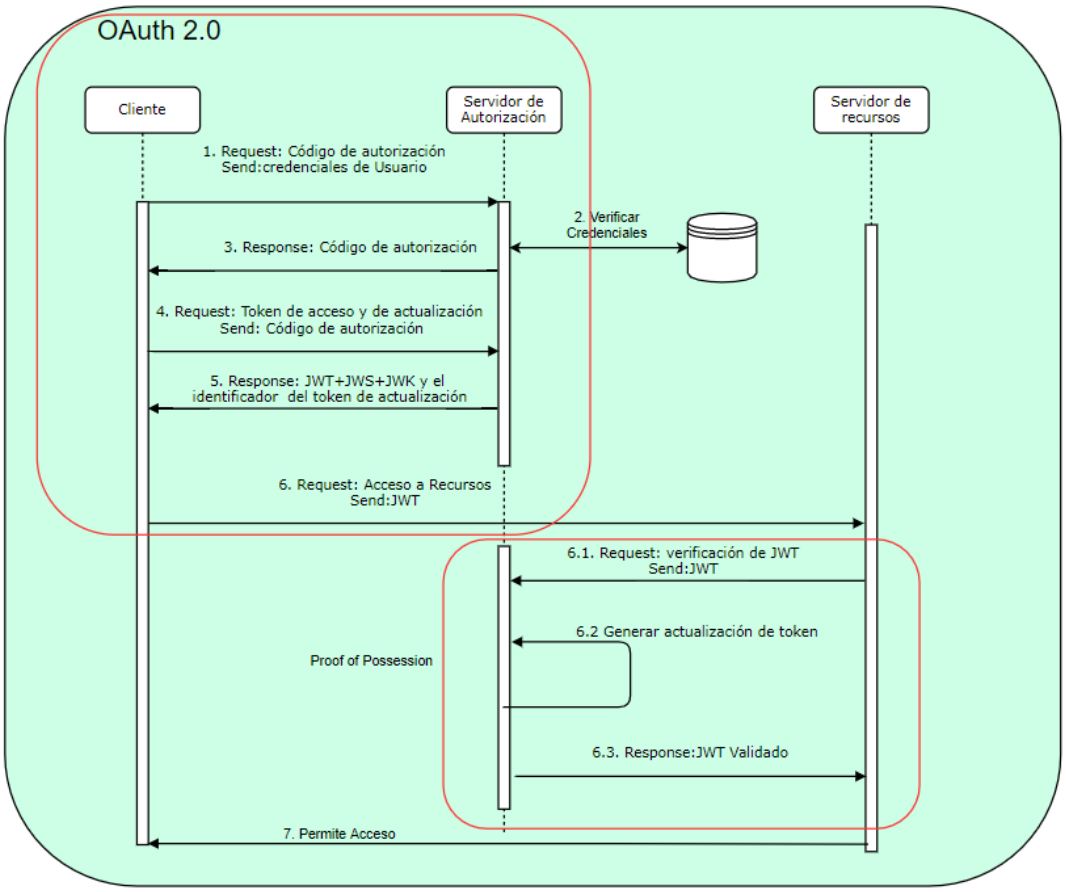


Figura 26. Diagrama de secuencia del modelo propuesto

4.1.5. Descripción detallada del modelo

Para comprender el proceso del modelo propuesto se detalla los pasos que se reutilizan del protocolo OAuth 2.0:

1. El cliente envía una petición para obtener el código de autorización enviando sus credenciales al Servidor de Autorización.
2. El Servidor de Autorización procede a la verificación en la base de datos que las credenciales sean correctas.
3. El Servidor de Autorización devuelve el código de autorización al cliente.
4. El cliente envía el código de autorización al Servidor de Autorización.

5. El Servidor de Autorización genera y envía un token de acceso de tipo JSON web Token (JWT), el cual tiene una clave JSON Web Key (JWK) es firmado con JSON Web Signature (JWS).

Desde el siguiente paso son adicionados por la propuesta:

6. Este paso es para el acceso a los recursos, esto se da cuando el Servidor de Autorización verifica el token, este sea correcto o no haya expirado. El proceso para este paso se da de la siguiente forma:
   a) El cliente envía el token al Servidor de Recursos.
   b) El Servidor de Recursos reenvía el token al Servidor de Autorización para la verificación de dicho código.
   c) El Servidor de Autorización valida el token mediante la verificación de la firma (JWS) con la que fue creada y comprueba el tiempo de vida del token.
      i. Si el token es correcto y no ha expirado se devuelve la información del usuario con los recursos a los cuales el cliente puede acceder.
      ii. Si el token es correcto y ha expirado se hace una actualización para generar un nuevo token (JWT) y comunicar al cliente hasta que se cumpla el paso anterior.
      iii. Si el token es erróneo se devuelve un mensaje de error
7. El Servidor de Recursos permite el acceso, si son correctos los datos del usuario los recursos son devueltos con permiso para usarlos.

4.1.6. Beneficios del modelo.

El modelo propuesto tiene como objetivo principal la seguridad de autorización y autenticación en un ecosistema IoT, para lo cual ocupamos mecanismos para cumplir dicho objetivo como el protocolo OAuth 2.0, JSON Web Token y Proof of Possession (PoP), siendo este último parte fundamental del modelo para brindar seguridad a los clientes.

El OAuth 2.0 es un protocolo ya existente en el mercado y usados por empresas que trabajan en un ecosistema IoT no obstante la propuesta enriquece al protocolo al considerar la comunicación directa entre servidores que benefician al cliente de la siguiente manera:

- El Servidor de Autorización permite la generación de un JWT si se verifica el código de autorización, propuesto por OAuth 2.0 de esta se restringe el acceso a usuarios que no tengan dicho código.
- Al juntar el protocolo OAuth 2.0 con JWT se puede reducir el tiempo de vida de los tokens, de esta forma se protege la identidad del usuario.
- Los tokens (JWT) al ser firmados con JWK y JWS, verifican que la información no haya sido cambiada de esta forma se valida la identidad.
- Al utilizar el PoP da la facultad al Servidor de Autorización para que reconozca la firma (JWS) del token (JWT) de esta forma evita el envío de la clave (JWK) esto permite impedir el acceso a recursos si esta clave es robada.

El uso de PoP nos permite el acceso seguro a recursos, este modelo de seguridad tiene diferentes formas de uso como lo especifica el estándar RFC4210 de IETF, en la propuesta lo usamos en los siguientes puntos para un control seguro:

- Tokens de seguridad con una única firma (JWS) que se encuentra solo en el Servidor de Autorización.
- La verificación entre la entidad final y el token.
- El PoP es actor en el proceso de validación de la clave simétrica del token enviado.

CONCLUSIONES Y RECOMENDACIONES

4.2. Conclusiones

Después de haber hecho el proceso de revisión de literatura se define que el protocolo OAuth en su versión 2.0 es el más usado en el proceso de autorización en aplicaciones y ecosistemas IoT y se considera que es uno de los más seguros en autorización de acceso a un recurso.

El diseño propuesto en comparación a diferentes modelos existentes con semejantes características permite al usuario autenticarse de una manera segura con los parámetros del CIA (Confidencialidad, Integridad, Disponibilidad).

Al usar Proof of Possesion (PoP) en el modelo propuesto se brinda una seguridad necesaria al usuario, mediante la directa comunicación entre los Servidores de Recursos y Autorización, esto nos ayuda a evitar el robo o suplantación de datos de un usuario.

Mediante la comparación y análisis de los diferentes modelos actuales se pudo observar que cada uno tiene diferente tecnología en el proceso de autorización, Google ocupa el protocolo OAuth 2.0 en su diseño original y en la propuesta se potenció este protocolo para mejorar las garantías de seguridad.

A lo largo del proyecto se ha podido evidenciar que existen varias formas para el proceso de autorización basadas en diferentes protocolos, lo que le hace más seguro; así como se debe tener en cuenta los protocolos HTTPs, TLS los cuales son un complemento a un proceso de seguridad, con ayuda de estos la propuesta podrá tener un proceso seguro de autorización.

4.3. Recomendaciones

Es recomendable el uso de Proof of Possesions (PoP) ya que el token es enviado sin la firma al cliente lo cual le hace más seguro, garantizando seguridad e integridad en la información enviada en una red evitando un ataque.

Es necesario que los mecanismos de autorización vayan de la mano con una correcta cultura de seguridad informática, ya que los sistemas pueden presentar una robustez alta pero el usuario, al ser el eslabón más débil del sistema, puede ocasionar una brecha de seguridad.

Con el fin de brindar confidencialidad e integridad en la información de un usurario, se recomienda utilizar herramientas que estén actualizadas y brinden esas características.

Se recomienda concientizar a los usuarios sobre el manejo de sus credenciales y datos personales, ya que el aumento de la tecnología abre varias oportunidades para el robo de la información.

Se recomienda realizar investigación y validación con esta propuesta y otros protocolos que puedan ayudar a tener un proceso de autorización seguro y ver su funcionamiento o cómo mejorarlo.

Se recomienda el estudio e investigación del transporte del token en el proceso de autorización para que sea confiable y no exista vulnerabilidad en este.

REFERENCIAS

Abomhara, M., & Koien, G. (2015). Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. Recuperado el 12 de febrero del 2020 de https://www.riverpublishers.com/journal_read_html_article.php?j=JCSM/4/1/4

Albors, J. (2018). Seguridad en dispositivos IoT: ¿Aún a tiempo de ganar la batalla? Obtenido de https://www.welivesecurity.com/la-es/2018/07/25/seguridad-iot-a-tiempo-ganar-batalla/

Almuairfi, S., Veeraraghavan, P., & Chilamkurti, N. (2011). IPAS: Implicit password authentication system. Proceedings - 25th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2011, 430–435. http://doi.org/10.1109/WAINA.2011.36

Almuairfi, S., Veeraraghavan, P., & Chilamkurti, N. (2011). IPAS: Implicit password authentication system. Proceedings - 25th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2011, 430–435. http://doi.org/10.1109/WAINA.2011.36

Amazon AW. (2020) “Documentación de AWS IoT Core”. Recuperado el 20 de abril del 2020 https://docs.aws.amazon.com/es_es/iot/?id=docs_gateway

Amazon Web Services. (2020) Cómo funciona la plataforma AWS IoT. Recuperado el 3 .de mayo del 2020 https://aws.amazon.com/es/iot/how-it-works/

Amazon Web Services. (2020) Preguntas frecuentes acerca de AWS IoT. Recuperado el 3 de mayo del 2020 https://aws.amazon.com/es/iot/faqs/

Anicas, M. (2018). Una introducción a OAuth 2. Recuperada el 10 de marzo del 2020 de https://www.digitalocean.com/community/tutorials/una-introduccion-a-oauth-2-es

Arias, S. E., Vargas, L., Gionantonio, D., Alejandra, M., Serrano, D. J., Cucchi, A., ... & Arch, D. F. (2019, June). Análisis de plataformas de cloud

computing. Caso Microsoft Azure y Amazon Web Services, haciendo uso de versiones privadas de prueba en entornos educativos. In XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan). Recuperado el 20 de abril del 2020 http://sedici.unlp.edu.ar/handle/10915/77256

auth0. (2020). OAuth 2.0 Authorization Framework. Recuperada el 5 de marzo del 2020 de https://auth0.com/docs/protocols/oauth2

AWS. (2020) “Características de AWS IoT Core”. Recuperado el 27 de abril del 2020 https://aws.amazon.com/es/iot-core/features/

AWS. (2020) “Guía del desarrollador”. Recuperado el 27 de abril del 2020 https://docs.aws.amazon.com/es_es/iot/latest/developerguide/iot-dg.pdf

Developers Google. (2020) Uso de OAuth 2.0 para acceder a las apis de Google Recuperado el 4 de junio del 2020 https://developers.google.com/identity/protocols/oauth2?hl=es

Gao, H., Liu, N., Li, K., & Qiu, J. (2014). Usability and security of the recall-based graphical password schemes. Proceedings - 2013 IEEE International Conference on High Performance Computing and Communications, HPCC 2013 and 2013 IEEE International Conference on Embedded and Ubiquitous Computing, EUC 2013, (60903198), 2237–2244. http://doi.org/10.1109/HPCC.and.EUC.2013.321

Google. (2020) Introducción a Cloud IoT Core . Recuperado el 4 de junio del 2020 https://cloud.google.com/iot/docs/how-tos/getting-started?hl=es

Goutham, R. A., Kim, D.-S., & Yoo, K.-Y. (2014). Implicit Graphical Password Mutual Authentication Using Mirror-image Encryption. Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems, 218–223. http://doi.org/10.1145/2663761.2664194

Gupta, S., Vashisht, S., & Singh, D. (2016). A CANVASS on cyber security attacks and countermeasures. 2016 1st International Conference on Innovation and Challenges in Cyber Security, ICICCS 2016, (Icicc), 31–35. http://doi.org/10.1109/ICICCS.2016.7542335

Haque, M. A., Khan, N. Z., & Khatoon, G. (2016). Authentication through keystrokes: What you type and how you type. Proceedings of 2015 IEEE International Conference on Research in Computational

Intelligence and Communication Networks, ICRCICN 2015, 257–261. http://doi.org/10.1109/ICRCICN.2015.7434246

Hidalgo Proaño, EA (2019). Análisis comparativo de tráfico entre la red WiFi y la red IOT-WiFi en el campus sur de la Universidad Politécnica Salesiana (UPS) (Tesis de licenciatura). Recuperado el 14 de abril del 2020 de https://dspace.ups.edu.ec/handle/123456789/16811

IBM. (2020) Seguridad de MQTT. Recuperado el 10 de junio del 2020 https://www.ibm.com/support/knowledgecenter/es/SSFKSJ_7.5.0/com.ibm.mm.tc.doc/tc00150_.htm

IdentityServer3. (2020) Proof of posesión – overview. Recuperada el 30 de junio del 2020 de https://identityserver.github.io/Documentation/docsv2/pop/overview.html

IETF. Infraestructura de clave Publica de Internet x.509 protocolo de gestión de certificados. Recuperado el 29 de abril del 2020 de https://tools.ietf.org/html/rfc4210#section-2

IETF. OAuth 2.0 Proof-of-Possession:Authorization server to client key doistribution. Recuperado el 29 de abril del 2020 de https://tools.ietf.org/id/draft-ietf-oauth-pop-key-distribution-04.html

Intel. (2020) “Introducción a las Placas Intel® Galileo”. Recuperado el 20 de abril del 2020 https://www.intel.la/content/www/xl/es/support/articles/000005912/boards-andkits/intel-galileo-boards.html

IoT World Online. (2020). Las grandes estadísticas del Internet de las Cosas (IoT). Recuperada el 15 de febrero del 2020 de https://www.iotworldonline.es/las-grandes-estadisticas-del-internet-de-las-cosas-iot/

ISO 27001:2013. (2017). ¿Qué es el CIA (Confidencialidad, Integridad, Disponibilidad) en la seguridad de la información? Obtenido de https://www.pmg-ssi.com/2017/07/cia-confidencialidadintegridad-disponibilidad-seguridad-de-la-informacion/

iT Reseller. (2017). El auge de IoT: en 2017 habrá 8.400 millones de dispositivos conectados. Recuperada el 12 de febrero del 2020 de https://www.itreseller.es/en-cifras/2017/02/el-auge-de-iot-en-2017-habra-8400-millones-de-dispositivos-conectados

Jenkov, J. (2014). OAuth 2.0 Roles. Recuperada el 10 de marzo del 2020 de http://tutorials.jenkov.com/oauth2/roles.html

Jiménez Casas, J. (2019). Sistema IoT para el control y monitorización de un terrario con Eclipse Kura, Amazon AWS y Angular. Recuperado el 20 de abril del 2020 dehttps://idus.us.es/handle/11441/91377;jsessionid=591E4F818394EE691D3AB7F6A5489D11?

Kolekar, V. K., & Vaidya, M. B. (2016). Click and session based - Captcha as graphical password authentication schemes for smart phone and web. Proceedings - IEEE International Conference on Information Processing, ICIP 2015, 669–674. http://doi.org/10.1109/INFOP.2015.7489467

Lashkari, A. H., Manaf, A. A., & Masrom, M. (2011). A secure recognition based Graphical Password by watermarking. Proceedings - 11th IEEE International Conference on Computer and Information Technology, CIT 2011, 07(1), 164–170. http://doi.org/10.1109/CIT.2011.29

Lin, P. L., Weng, L. T., & Huang, P. W. (2008). Graphical passwords using images with random tracks of geometric shapes. Proceedings - 1st International Congress on Image and Signal Processing, CISP 2008, 3, 27–31. http://doi.org/10.1109/CISP.2008.603

López Grande, C. E., & Edgardo, C. (2015). Ingeniería social: el ataque silencioso, (1). Retrieved from http://www.redicces.org.sv/jspui/handle/10972/2910

Microsof. (2020) “Arquitectura de seguridad de Internet de las cosas (IoT)”. Recuperado el 27 de abril del 2020 https://docs.microsoft.com/es-es/azure/iot-fundamentals/iot-security-architecture

Microsoft. (2020) Guia de protocolo deAzureService Bus. Recuperado el 4 de junio del 2020 https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-amqp-protocol-guide

Mir, M. S., Wani, S., & Ibrahim, J. (2013). Critical information security challenges: An appraisal. 2013 5th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2013. http://doi.org/10.1109/ICT4M.2013.6518890

Mondragón, M. V. P., & Guillén, E. P. (2019). Servicios de autenticación y autorización orientados a internet de las cosas. Revista

Telemática, 17(2), 42-51. Recuperado el 14 de abril del 2020 de https://revistas.udistrital.edu.co/index.php/vinculos/article/view/15758/15392

Morales-Suárez, A. C., Díaz-Ávila, S. S., & Leguizamón-Páez, M. Á. (2019). Mecanismos de seguridad en el internet de las cosas/Security mechanisms on the internet of things. Revista vinculos, 16(2), NA-NA. Recuperado el 14 de abril del 2020 de https://go.gale.com/ps/anonymous?id=GALE%7CA611933161&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=1794211X&p=IFME&sw=w

Padilla Cañadas, K. (2015). Estudio de un protocolo de identificación para el internet de las cosas (Bachelor's thesis). Recuperado el 14 de abril del 2020 de https://e-archivo.uc3m.es/handle/10016/23700

Peña, M. (2020). Qué es el Internet de las Cosas y cómo afecta tu vida diaria. Recuperada el 15 de febrero del 2020 de https://es.digitaltrends.com/tendencias/que-es-el-internet-de-las-cosas/

Perez, N. B., Bustos, M. A., Berón, M., & Rangel Henriques, P. (2018). Análisis sistemático de la seguridad en internet of things. In XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste). Recuperado el 14 de abril del 2020 de http://sedici.unlp.edu.ar/handle/10915/68387

Rajarajan, S., Prabhu, M., Palanivel, S., & Karthikeyan, M. P. (2014). Gramap: Three stage graphical password authentication scheme. Journal of Theoretical and Applied Information Technology, 61(2), 262–269.

Rivas, G. (2018). El Internet de las Cosas: 5 medidas para garantizar la seguridad del IoT. Obtenido de https://www.gb-advisors.com/es/iot-seguridad-el-internet-de-las-cosas/

Sandoval, K. (2017). OAuth 2.0 – Why It’s Vital to IoT Security. Obtenido de https://nordicapis.com/whyoauth-2-0-is-vital-to-iot-security/

Sobers, R. (2020). ¿Qué es OAuth? Definición y cómo funciona. Recuperada el 26 de febrero del 2020 de https://www.varonis.com/blog/what-is-oauth/

Srivastava, S., & Sivasankar, M. (2017). On the generation of alphanumeric one time passwords. Proceedings of the International Conference on

Inventive Computation Technologies, ICICT 2016, 1(i), 1–3. http://doi.org/10.1109/INVENTIVE.2016.7823287

Symantec. (2018). Informes sobre las Amenazas para la Seguridad en Internet. Physical Review B, 72(10), 1–13.

Tschofenig, H. (2016). Fixing User Authentication for the Internet of Things (IoT). Obtenido de https://link.springer.com/article/10.1007/s11623-016-0582-1

Valois, M. A. (2020). Qué es internet de las cosas y cómo funciona. Recuperada el 17 de febrero del 2020 de https://www.hostgator.mx/blog/internet-de-las-cosas/

Watchguard. (2017). Internet Security Report - Q4 2017. Recuperada el 17 de febrero del 2020 de Retrieved from https://www.watchguard.com/wgrd-resource-center/security-report-q4-2017

York, P. (2017). Uber fue hackeado, datos de 57 millones de usuarios comprometidos. Recuperada el 15 de febrero del 2020 de https://www.fayerwayer.com/2017/11/uber-fue-hackeado-datos-de-57-millones-de-usuarioscomprometidos/