1.Introduccion

Se puede pensar a Internet de la Cosas como un escenario en el cual, los objetos, animales o personas están provistos de una única identificación y la habilidad de transferir datos automáticamente sobre la red sin requerimientos de intervención humano a humano o humano a máquina. Esto permite automatizar “cosas”, las cuales en función de configuraciones personales y un procesamiento sofisticado basado en la “nube”, hacen que eventos sucedan sin nuestra intervención.

Las aplicaciones de Internet de las Cosas, ya sean las asociadas a salud, energía o ciudades inteligentes están básicamente constituidas por: uno o varios dispositivos finales o “cosas”, una unidad concentradora (Gateway/Border Router), la infraestructura de comunicaciones (antenas, cable, servidores, routers, etc.) y los servicios y aplicaciones. Cada uno de estos elementos encierra en sí mismo un alto nivel de complejidad, así como una gran variedad de alternativas para implementarlos tanto a nivel netamente hardware como a niveles de protocolos de comunicación, sistemas operativos y software de desarrollo.

Típicamente cuando hablamos de los dispositivos finales hablamos de “cosas inteligentes” o “Smart things”, el término “inteligente” está relacionado con la finalización de una tarea de forma más consistente y confiable, por ejemplo: tostador mecánico vs tostador electrónico, un sistema de iluminación manual vs sistema de iluminación con sensores. Esta “inteligencia” se logra con la integración de un procesamiento embebido (típicamente un microcontrolador), lo que además permite una comunicación en forma electrónica con el usuario usando pantallas, touchs, pulsadores, etc. (interfaz Hombre Maquina – HMI). Estos dispositivos además de la inteligencia deben incluir sensores que les permitan interactuar con el medio, métodos de identificación, integridad y seguridad de los datos y una comunicación remota que lo permitirá la transferencia de datos en forma univoca a servidores donde se realizara el procesamiento de los mismos. Dependiendo la aplicación nos podemos encontrar con parámetros a medir que utilizan sensor cuyo desarrollo han logrado una gran madurez, como ser la medición de temperatura, o parámetros que requieren el desarrollo de un sensor a medida o una medición indirecta. En lo asociado a la identificación, integridad y seguridad de los datos podemos encontrar dispositivos que simplemente se preocupan de integridad de los datos con un simple CheckSum hasta sistemas asociados con transferencia de dinero con sofisticados algoritmos de encriptación. En lo referente a la comunicación remota existen un sin número de tecnologías, tanto cableadas como inalámbricas, que permiten realizar el enlace, algunas sin direccionamiento IP (RS232, Zigbee, Bluetooth, LORA), lo que implica el uso de un concentrador obligado, y otras con (Ethernet, Wifi, GSM/GPRS).

Los concentradores son dispositivos que deben poseer una capacidad de procesamiento superior a los dispositivos finales, ya que deben tener la capacidad de dialogar con múltiples dispositivos finales, realizar una conversión de protocolo (desde un protocolo no orientado a IP a uno que sí, dependiendo el tipo de tecnología utilizada), enrutar los datos y ofrecer una interfaz HMI (Interfaz hombre-máquina) para la administración de la red. Esto implica el uso de microcontroladores de alta capacidad de procesamiento o inclusive microprocesadores con la capacidad de correr algún sistema operativo (Linux, Android, Windows).

La infraestructura suele estar en manos de prestadoras de servicio de Internet, compañías de telefonía fija, telefonía celular, proveedores de servicios de internet y proveedores de televisión por cable, y nosotros solo estar al tanto de las limitaciones de cada una de ellas. Sin embargo, en las aplicaciones de ciudades inteligentes, donde el área de cobertura de una red de Internet de las cosas pueden ser varios kilómetros y dependiendo la tecnología empleada para el enlace comunicación, parte de la infraestructura debe ser contemplada. Un ejemplo de esto puede ser un sistema cuyo enlace sea a través de tecnología LORA (Long

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

Range), lo cual implica un despliegue de antenas para tomar los datos de los sensores remotos.

En lo que respecta a servicios y aplicaciones, las mismas pueden ser una simple base de datos a través de la cual se puede acceder por SQL y presentar la información en la pantalla de una computadora personal o un teléfono inteligente para nuestro análisis, pueden incluir la posibilidad de que actuemos sobre los parámetros que estamos monitoreando y pueden llegar a ser sistemas totalmente autónomos que a través de técnicas como aprendizaje profundo (Deep learnig),minería de datos(data mining) y otras que realizan tareas sin nuestra intervención. A nivel mundial existen gran cantidad de proveedores de estos tipos de soluciones, están quienes ofrecen el servicio de almacenamiento en la nube, quienes ofrecen servicios Middleware, servicios de presentación de datos y servicios análisis de datos, ejemplos de estos sistemas pueden ser Amazon, IBM- Watson, Microsoft – Azure – Microsoft Cognitive, Telit –Device Cloud, etc.

2. Computación en la nube (Cloud Computing)

La computación en la nube [Fig 1] viene a cubrir las necesidades planteadas en los últimos dos párrafos de la sección anterior. La misma, la cual es también conocida como servicios en la nube, informática en la nube, nube de cómputo, nube de conceptos o simplemente "la nube", es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet.

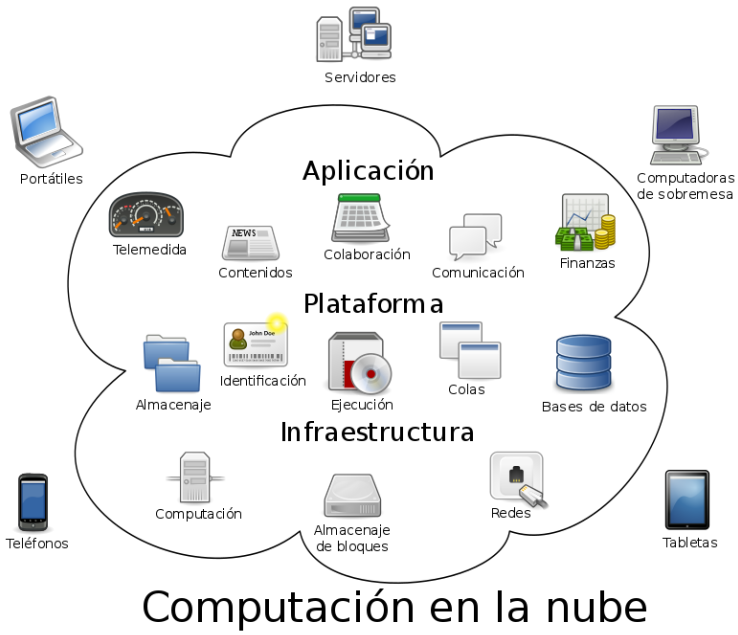

Figura 1. Cloud Computing

La computación en la nube son servidores en Internet encargados de atender las peticiones en cualquier momento. Se puede tener acceso a su información o servicio, mediante una conexión a internet desde cualquier dispositivo móvil o fijo ubicado en cualquier lugar. Esta medida reduce los costos, garantiza un mejor tiempo de actividad y la invulnerabilidad de los sitios web a los delincuentes informáticos, a los gobiernos locales y a sus redadas policiales pertenecientes.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

El concepto de “nube informática” es muy amplio, y abarca casi todos los posibles tipo de servicio en línea, pero cuando las empresas predican ofrecer un utilitario alojado en la nube, por lo general se refieren a alguna de estas tres modalidades: el software como servicio(por sus siglas en inglés SaaS -Software as a Service-), Plataforma como Servicio (PaaS) e Infraestructura como Servicio (IaaS) [Fig 2].

El software como servicio (SaaS) es un modelo de distribución de software en el que las aplicaciones están alojadas por una compañía o proveedor de servicio y puestas a disposición de los usuarios a través de una red, generalmente la Internet. Plataforma como servicio (PaaS) es un conjunto de utilitarios para abastecer al usuario de sistemas operativos y servicios asociados a través de Internet sin necesidad de descargas o instalación alguna. Infraestructura como Servicio (IaaS) se refiere a la tercerización de los equipos utilizados para apoyar las operaciones, incluido el almacenamiento, hardware, servidores y componentes de red.
.
El concepto de la computación en la nube empezó en proveedores de servicio de Internet a gran escala, como Google (Google Cloud Services), Amazon (2006), Microsoft (Microsoft Azure), Alibaba y otros que construyeron su propia infraestructura. De entre todos ellos emergió una arquitectura: un sistema de recursos distribuidos horizontalmente, introducidos como servicios virtuales de TI (Tecnología Informática) escalados masivamente y manejados como recursos configurados y mancomunados de manera continua. Este modelo de arquitectura fue inmortalizado por George Gilder en su artículo de octubre de 2006 en la revista Wired titulado «Las fábricas de información». Las granjas de servidores, sobre las que escribió Gilder, eran similares en su arquitectura al procesamiento “grid” (red, rejilla), pero mientras que las redes se utilizan para aplicaciones de procesamiento técnico débilmente acoplados (loosely coupled), un sistema compuesto de subsistemas con cierta autonomía de acción, que mantienen una interrelación continua entre ellos, este nuevo modelo de nube se estaba aplicando a los servicios de Internet [ARCHIVE 2010].

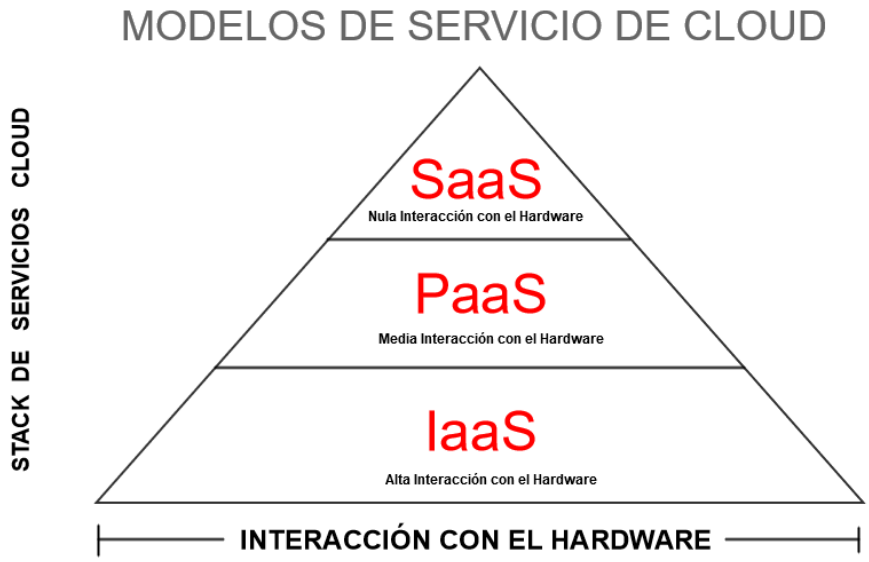

Figura 2. Pila de Servicios

En la figura N°2, se puede ver los tres distintos servicios que puede proporcionar una empresa de Computación en la nube, donde la capa inferior puede contener (o no) los servicios de las capas superiores.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

2.1 Ventajas

Las principales ventajas de la computación en la nube son:

- Integración probada de servicios Red. Por su naturaleza, la tecnología de Computación en la nube se puede integrar con mucha mayor facilidad y rapidez con el resto de las aplicaciones empresariales (tanto software tradicional como Computación en la nube basado en infraestructuras), ya sean desarrolladas de manera interna o externa.

- Prestación de servicios a nivel mundial. Las infraestructuras de Computación en la nube proporcionan mayor capacidad de adaptación, recuperación completa de pérdida de datos (con copias de seguridad) y reducción al mínimo de los tiempos de inactividad.

- Una infraestructura 100% de Computación en la nube permite también al proveedor de contenidos o servicios en la nube prescindir de instalar cualquier tipo de software, ya que este es provisto por el proveedor de la infraestructura o la plataforma en la nube. Un gran beneficio de la Computación en la nube es la simplicidad y el hecho de que requiera mucha menor inversión para empezar a trabajar.

- Implementación más rápida y con menos riesgos, ya que se comienza a trabajar más rápido y no es necesaria una gran inversión. Las aplicaciones de la computación en la nube suelen estar disponibles en cuestión de días u horas en lugar de semanas o meses, incluso con un nivel considerable de personalización o integración.

- Actualizaciones automáticas que no afectan negativamente a los recursos de TI. Al actualizar a la última versión de las aplicaciones, el usuario se ve obligado a dedicar tiempo y recursos para volver a personalizar e integrar la aplicación. Con la Computación en la nube no hay que decidir entre actualizar y conservar el trabajo, dado que esas personalizaciones e integraciones se conservan automáticamente durante la actualización.

- Contribuye al uso eficiente de la energía. En este caso, a la energía requerida para el funcionamiento de la infraestructura. En los datacenters tradicionales, los servidores consumen mucha más energía de la requerida realmente. En cambio, en las nubes, la energía consumida es solo la necesaria, reduciendo notablemente el desperdicio.

2.2 **Desventajas**

- La centralización de las aplicaciones y el almacenamiento de los datos origina una interdependencia de los proveedores de servicios.

- La disponibilidad de las aplicaciones está sujeta a la disponibilidad de acceso a Internet (conjunto descentralizado de Redes).

- La confiabilidad de los servicios depende de la "salud" tecnológica y financiera de los proveedores de servicios en nube [Richard Stallm 2008]. Empresas emergentes o alianzas entre empresas podrían crear un ambiente propicio para el monopolio y el crecimiento exagerado en los servicios.

- La disponibilidad de servicios altamente especializados podría tardar meses o incluso años para que sean factibles de ser desplegados en la red.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

- La madurez funcional de las aplicaciones hace que continuamente estén modificando sus interfaces, por lo cual la curva de aprendizaje en empresas de orientación no tecnológica tenga unas pendientes significativas, así como su consumo automático por aplicaciones.

- Seguridad. La información de la empresa debe recorrer diferentes modos para llegar a su destino, cada uno de ellos (y sus canales) son un foco de inseguridad. Si se utilizan protocolos seguros, HTTPS (Protocolo seguro de transferencia de hipertexto) por ejemplo, la velocidad total disminuye debido a la sobrecarga que estos requieren.

- Escalabilidad a largo plazo. A medida que más usuarios empiecen a compartir la infraestructura de la nube, la sobrecarga en los servidores de los proveedores aumentará, si la empresa no posee un esquema de crecimiento óptimo puede llevar a degradaciones en el servicio o altos niveles de jitter(variabilidad temporal durante el envío de señales digitales).

3.Servicios Disponibles

Ahora bien, visto las ventajas y desventajas de poseer una computación en la nube. Centraremos en explicar los distintos servicios ofrecidos. Si bien anteriormente se nombraron, en este apartado profundizaremos que hace cada servicio, como funciona y a que capa pertenece.

3.1 Software como servicio

El software como servicio (en inglés software as a Service, SaaS) se encuentra en la capa más alta y caracteriza una aplicación completa ofrecida como un servicio, por demanda, vía multitenencia — que significa una sola instancia del software que corre en la infraestructura del proveedor y sirve a múltiples organizaciones de clientes —. Las aplicaciones que suministran este modelo de servicio son accesibles a través de un navegador web — o de cualquier aplicación diseñada para tal efecto — y el usuario no tiene control sobre ellas, aunque en algunos casos se le permite realizar algunas configuraciones. Esto le elimina la necesidad al cliente de instalar la aplicación en sus propios computadores, evitando asumir los costos de soporte y el mantenimiento de hardware y software.

Podemos entender ejemplos de Software como servicios, cosas cotidianas que podemos utilizar día a día como una casilla de correo (Gmail, Yahoo!, Hotmail) o de mensajería (Skype, WhatsApp web).

3.2 Plataforma como servicio

La capa del medio, que es la plataforma como servicio (en inglés Platform as a Service, PaaS), es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal (persistencia de datos, autenticación, mensajería, etc.). De esta forma, un arquetipo de plataforma como servicio podría consistir en un entorno conteniendo una pila básica de sistemas, componentes o APIs (siglas de 'Application Programming Interface') preconfiguradas y listas para integrarse sobre una tecnología concreta de desarrollo (por ejemplo, un sistema Linux, un servidor web, y un ambiente de programación como Perl o Ruby). Las ofertas de PaaS pueden dar servicio a todas las fases del ciclo de desarrollo y pruebas del software, o pueden estar especializadas en cualquier área en particular, tal como la administración del contenido.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

Ejemplos comerciales son: Google App Engine (servicio de alojamiento web que presta Google); Microsoft Azure, entre otras que sirven de una plataforma en la nube la cual, permite el desarrollo y ejecución de aplicaciones codificadas en varios lenguajes y tecnologías, entre otros servicios. Servicios PaaS como estos permiten gran flexibilidad, pero puede ser restringida por las capacidades disponibles a través del proveedor.

En este modelo de servicio al usuario se le ofrece la plataforma de desarrollo y las herramientas de programación por lo que puede desarrollar aplicaciones propias y controlar la aplicación, pero no controla la infraestructura.

3.3 Infraestructura como servicio

La infraestructura como servicio (infrastructure as a Service, IaaS) —también llamada en algunos casos hardware as a Service, HaaS [IEEE 2008] — se encuentra en la capa inferior y es un medio de entregar almacenamiento básico y capacidades de cómputo como servicios estandarizados en la red.

Servidores, sistemas de almacenamiento, conexiones, enrutadores, y otros sistemas se concentran (por ejemplo, a través de la tecnología de virtualización) para manejar tipos específicos de cargas de trabajo —desde procesamiento en lotes (“batch”) hasta aumento de servidor/almacenamiento durante las cargas pico—. El ejemplo comercial mejor conocido es Amazon Web Services, cuyos servicios EC2 y S3 ofrecen cómputo y servicios de almacenamiento esenciales (respectivamente).

También puede existir la posibilidad de que estén todos los servicios involucrados en conjunto. Muchas empresas ofrecen la posibilidad de adquirir los 3 servicios en un pack o combo.

Además, hay que remarcar que este nuevo paradigma reduce el tiempo para llevar un producto al mercado, consiguiendo así que nuevos servicios puedan estar disponibles en muy poco tiempo de producción. Y todo va tan rápido que las empresas requieren sacar los productos al mercado lo antes posible.

La elección de alguno de estos servicios consiste en evaluar que recursos son necesarios para llevar adelante la aplicación encarada, o replanteado desde un enfoque técnico – comercial, lo mínimo que se precisa para desarrollar la aplicación al menor costo. Si a la hora de elegir, se está en la disyuntiva entre varios servicios, los análisis concluyen que, si se elige un servicio más caro, pero que reduciría los recursos que yo preciso para hacerlo funcionar, este va a ser la decisión óptima, ya que el tiempo ahorrado es mayor al costo de recursos.

Para la solución de la problemática mencionada en la introducción se desarrollará un **Sistema de Estacionamiento**, como sistema de ejemplo. Luego de realizar un estudio de los recursos necesarios para dicha aplicación, se analizarán las prestadoras de servicios de Computación en la nube, a fin de poder llegar a una conclusión sobre el prestador a emplear en la aplicación.

4.Escenarios a analizar

4.1. Ingreso

La figura N°3 y N°4 presentan una representación del sistema de ingreso al estacionamiento. A continuación, se detallan los puntos relevantes:

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

1. Cada vehículo/usuario cuenta con una identificación, la misma puede ser realizada a través de un tag de RFID/NFC o a través de una aplicación celular.
2. Un dispositivo lector tomara la identificación del vehículo/usuario.
3. Dicha información se compara con los distintos ID de los vehículos/usuarios almacenados en una base de datos, la cual puede estar en la nube.
4. Si el ID es encontrado la puerta se abre.

Fig 3 . Ingreso

En el punto 3 la identificación obtenida debe ser manipulada por una base de datos, preferentemente en la nube para poder acceder a ella desde cualquier ubicación.

Fig 4. conectividad

Se deberá poder acceder a la información cuando sea necesario para poder hacer estadísticas gráficos y consultas de los ingresos al establecimiento, horarios más concurridos entre otros. Para el cual, se necesita una infraestructura en la nube (IaaS) capaz de almacenar los datos históricos, y otro capaz de poder consultarlos cuando sea necesario (SaaS). Vale aclarar que la base de datos deberá tener la seguridad correspondiente para que los datos no sean violados ni alterados.

4.2. Disponibilidad.

Las figuras N°5 y N°6 presentan una representación de la parte del sistema asociado a la presentación de la disponibilidad de plazas de estacionamiento. A continuación, se detallan los puntos relevantes:

1. Una vez que el vehículo/usuario ingrese al estacionamiento, el conductor desde el dispositivo móvil visualiza los lugares de estacionamiento disponibles.
2. Éste selecciona un lugar para estacionar su vehiculo, pudiendo indicar el tiempo que estará allí, y estacionará.
3. Cuando el vehículo deje de utilizar la plaza. Pasará su estado de "**Ocupado**" a "**Disponible**" en la aplicación para que otro usuario pueda usarlo.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

Fig 5. Aplicación de Estacionamiento

Fig 6. Estacionamientos disponibles u ocupados

Para este segundo escenario el sistema deberá operar con bases de datos en tiempo real (donde los datos serán consultados con mayor frecuencia. Para ver el estado de la plaza). Esto implica que el sistema trabaje con servicios online para poder establecer los tiempos de cada plaza y su estado (ocupado o disponible) en el instante. Ya que, dependiendo del tamaño del estacionamiento cambiará constantemente. Entonces, es necesario el uso de un servicio Computación en la nube de base de datos para poder tener un control preciso del estado de la plaza.

4.3. Pago

A continuación, se detallan los puntos relevantes asociados al pago del estacionamiento, el cual se realiza al momento de retirarse del estacionamiento:

1. El conductor podrá contener dentro de la aplicación una pestaña de pago “billetera digital” que podrá cargar tanto en kioscos como con tarjetas de débito y crédito. El sistema podrá contemplar otros métodos de pago.
2. El cobro dependerá del tiempo y el vehículo en cuestión.
3. Al ingresar al establecimiento conductor recibirá un comprobante, pudiendo ser el mismo virtual a través de la aplicación móvil. Al momento de retirase se chequera el tiempo que estuvo por medio del comprobante y el conductor podrá realizar el pago a través de la aplicación o a través de otro medio.

En este escenario se deberá trabajar con sistemas de tiempo real de cobro, que permita mantener informado al usuario de su estado de cuenta y al estacionamiento del monto que posee el mismo para efectuar el pago.

4.4. Beneficios

El conductor podrá recibir beneficios. Si es discapacitado podrá tener lugares privilegiados, o si es un cliente habitual podrá recibir promociones o descuento.

4.5. Administrador del estacionamiento

El administrador del estacionamiento debe conocimiento de estado y control de cada plaza del estacionamiento, logrando tener estadísticas de días con mayor concurrencia, tiempo de estadía y horas por vehículo. Es de suma importancia, que el administrador pueda tener un

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

correcto acceso a la base de datos de su estacionamiento y corroborar los ingresos que le genere. Por otro lado, podrá acceder a estadísticas semanales mensuales y anuales, para ello es necesario que el administrador tenga acceso a la interfaz de administración, donde el sistema guardará los reportes semanales, mensuales y anuales, y le permita cambiar las configuraciones del sistema, con gráficos y estadísticas para que pueda visualizar la productividad del estacionamiento.

4.6. Conclusión

Como resumen de los escenarios vistos, se llega a la conclusión que el sistema, independiente al hardware asociado, deberá poseer un sistema de base de datos, e infraestructura de almacenamiento para el historial de datos del estacionamiento, y así poder realizar la administración del ingreso, la disponibilidad de plazas, el cobro, la generación de beneficios y la realización de estadísticas que permitan un uso más eficiente del estacionamiento.

En este contexto, si se pretende realizar el sistema a través de servicios de Computación en la nube (el cual puede o no incluir el diseño de la aplicación de gestión), se debe pensar en contratar servicios de IaaS y PaaS.

5. Prestadoras de servicios

A continuación, se detallará varias prestadoras de servicios y se analizaran los servicios que ofrecen a fin de poder elegir entre ellas la que se adapte a la necesidad planteada en el punto de nuestra aplicación.

Son muchas empresas hoy en día que se dedican a prestar servicios de cloud computing tales como Microsoft, Amazon, IBM, Google, Alibaba, entre otras.

A continuación, se detallará algunas soluciones de servicios, que contienen Servicios en todas las etapas de Cloud (SaaS, PaaS y IaaS), tales como, **Microsoft Azure, Amazon Web Services** e **IBM Cloud**.

5.1 Microsoft Azure

Microsoft Azure es una nube pública de pago por uso de servicios, que permite almacenar datos y usar aplicaciones de manera online, a través de una red global de centros de datos de Microsoft. La empresa estadounidense garantiza la disponibilidad de la nube para cualquier usuario en la Red, ya que tiene repartidos centros de datos en todo el mundo. En la actualidad, Azure está presente en 54 regiones del planeta y otras 6 que vienen de camino, incluyendo ya a 140 países en todo el mundo, dentro de ellas Argentina.

5.2 Amazon Web Services (Amazon EFS)

Amazon Web Services (AWS) es una plataforma de servicios de nube que proporciona una variedad de servicios de infraestructura tales como almacenamiento, redes, bases de datos, servicios de aplicaciones, potencia de cómputo, mensajería, inteligencia artificial, servicios móviles, seguridad, identidad y conformidad, entre otros, los cuales permiten el crecimiento de las empresas. Esta plataforma de servicios tiene presencia en 44 zonas de disponibilidad dentro de 16 regiones geográficas en el mundo, y se crearán próximamente 14 zonas y 5 regiones más en países como China, Hong Kong, Francia y otra región AWS GovCloud en Estados Unidos.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

5.3 IBM Cloud

IBM cloud computing es un conjunto de servicios de cloud computing para empresas que ofrece la empresa de tecnología de la información IBM . La nube de IBM incluye infraestructura como servicio(IaaS), software como servicio (SaaS) y plataforma como servicio (PaaS) ofrecidos a través de modelos de entrega de nube pública, privada e híbrida , además de los componentes que forman esas nubes. Esta red tiene más de 55 centros, que están repartidos por 19 países diferentes.

5.4. Comparativas

A continuación, se realizará la comparación de los servicios ofrecidos por los proveedores antes mencionados siguiendo dos posibles caminos de implantación: el primero de ellos analizando los servicios específicos ofrecidos para cada uno de nuestras necesidades, y el segundo analizando los servicios de plataformas para aplicaciones de IoT.

5.4.1. Servicios específicos

A continuación, se compararán las prestaciones de cada servicio, y a su vez llegar a una conclusión de elección. Dentro del punto 5.4.1.1 se compararán ***Bases de Datos*** *y* en el punto 5.4.1.2 ***Servicio de almacenamiento.***

5.4.1.1. Base de datos

De acuerdo con la necesidad de poder manejar información online, es indispensable una base de datos para la solución de nuestro sistema de estacionamiento. En breve se distinguirán ciertos aspectos de utilidad que presta cada una de las prestadoras de servicios.

5.4.1.1.1. Núcleos y almacenamiento

Si bien estamos hablando de un servicio en la nube, este servicio debe estar soportado por algún tipo de hardware. Por lo tanto, se debe considerar el tipo de hardware que poseen las instalaciones de las prestadoras de servicios, a fin de poder determinar el óptimo para nuestra aplicación.

5.4.1.1.1.1. Microsoft Azure SQL Server

Microsoft Azure SQL Server está separada en 2 grupos, Gen 4 y Gen 5.

5.4.1.1.1.1.1. Gen 4

Las CPU Gen 4 se basan en procesadores Intel de 2x 2,4 GHz, (en Gen 4, 1 núcleo virtual= 1 CPU física) variando en sus Capacidades pudiendo elegir entre 56GB, 112GB y 157 GB de memoria.

5.4.1.1.1.2. Gen 5

Mientras que las CPU lógicas Gen 5 se basan en procesadores Intel de 4x 2,3 GHz, en Gen 5, 1 núcleo virtual = 1 hiperproceso (permite que un solo núcleo de procesador físico se comporte como dos procesadores lógicos) sus capacidades de almacenamiento varían en 40.8GB, 81.6GB, 122.4GB, 163.2GB, 204GB, 326.4GB, 396GB de memoria [Azure Prestaciones 2019].

5.4.1.1.2. IBM

IBM presenta 3 tipos de servicios, Estándar Pequeño, Mediano y Grande.

5.4.1.1.2.1. Estándar Pequeño

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

Posee Núcleos privados de 2 x 2.0 GHz, 8GB de RAM, 1x100GB (SAN: redes de área de almacenamiento), con 1x500GB de almacenamiento

5.4.1.1.1.2.2. Estándar Mediano

Posee Núcleos privados de 4 x 2.0 GHz,16GB de RAM,1x100GB (SAN), de almacenamiento.

5.4.1.1.1.2.3. Estándar Grande

Posee núcleos privados de 8 x 2.0 GHz, 32GB de RAM, 1x100GB (SAN), 1x2TB de almacenamiento.

5.4.1.1.1.3. Amazon Web Services Aurora

En cuanto en Amazon Web Services Aurora, está dividida en 2 grupos, Económico y Premium, y en estos la capacidad de la base de datos se mide en unidades de capacidad de Aurora (ACU).

5.4.1.1.1.3.1. Económico

El económico posee un núcleo de 1 ACU con 2 x 2.0 GHz, 8 Gb de RAM con una capacidad de 150GB de almacenamiento

5.4.1.1.1.3.2. Premium

Mientras que la Premium posee un núcleo de 4 ACU con 8 x 2.0 GHz, 32 Gb de RAM con 500GB de almacenamiento [AWS-SQL 2019].

5.4.1.1.2. Seguridad

Otro aspecto a tener en cuenta al momento de optar por una base de datos es la seguridad, si una base de datos es segura así lo será para los usuarios de la aplicación de estacionamiento. A continuación, en la tabla N°1, se puede apreciar el tipo de seguridad de cada una de las bases de datos

Microsoft Azure SQL Server	Amazon Web Services Aurora	IBM DB2
SOC	SOC 1/ISAE 3402, SOC 2 y SOC 3,	SOC 2
GDPR	DIACAP	
FedRAMP/FISMA,	FedRAMP/ FISMA	
PCI DSS	PCI DSS	PCI DSS
ISO/IEC 27001/27002	ISO 9001, ISO 27001 e ISO 27018	
HIPAA		HIPA

Tabla N° 1. Comparativas de Estándares

Podemos observar en la tabla que son compartidas por las tres:

- SOC: Sistema de Organización de Servicios
- PCI DSS: Estándar de Seguridad de Datos de la Industria de Tarjetas de Pago.

Tanto Azure como Aurora comparten:

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

- FedRAMP: Programa Federal de Gestión de Riesgos y Autorizaciones
- FISMA: Ley Federal de Gestión de la Seguridad de la Información
- ISO 27001: Estándar para la seguridad de la información aprobado y publicado como estándar internacional.

Se puede remarcar de Azure SQL server: ***GDPR*** (General Data Protection Regulation) (En español: Reglamento general de protección de datos.) que se encarga que todo consumidor y ciudadano tiene derecho a saber cómo se utilizan sus datos personales.

Mientras que IBM con Azure poseen **HIPAA** (Ley de Responsabilidad y Portabilidad de la Información de Salud). HIPAA fue diseñada por comités gubernamentales que intentan proteger los datos de los ciudadanos

A su vez, AWS Aurora contiene certificación en ***DIACAP***: Proceso de certificación y acreditación de aseguramiento de la información del Departamento de Defensa, garantizar que las empresas y organizaciones apliquen la gestión de riesgos a los sistemas de información.

Para nuestra aplicación, sistema de estacionamiento, los estándares de seguridad de mayor relevancia serian: PCI DSS, ya que la aplicación incorpora el pago de las plazas de estacionamiento y SOC, que se responsabiliza de que el servicio sea seguro. Por lo tanto, en este aspecto las tres prestadoras de servicios cumplen con los requisitos de la aplicación de estacionamiento.

A su vez, los tres servicios poseen:

- ***Data scheme*** (Esquema de datos) organización de datos como un plano de cómo se construye la base de datos.
- ***XML support*** (Soporte de XML) es un lenguaje de marcado que define un conjunto de reglas para codificar documentos en un formato que es legible para los humanos y para la máquina
- ***Secondary Indexes*** (Índice Secundario) es una estructura de datos que contiene un subconjunto de atributos de una tabla, además de una clave alternativa para admitir las operaciones Query.

5.4.1.1.3. Conclusión Base de Datos

A nivel núcleos y almacenamiento Noto se observa que poseen características semejantes, en IBM al presentar tres grupos, me permite más amplitud al escoger, y a su vez un almacenamiento superior en cuanto a su paquete "estándar grande". En cuanto a seguridad, las tres prestadoras cumplen los requisitos de la aplicación, de manera que la aspirante a elegir sea el servicio de IBM.

5.4.1.2. Almacenamiento

Para la aplicación de Estacionamiento se necesita de un servicio de almacenamiento para poder guardar datos históricos y con esto poder hacer estadísticas y tener un control de comportamientos en distintas zonas geográficas que serán mostrados en tablero o comúnmente llamado dashboard.

5.4.1.2.1 Amazon EFS (Elastic File System)

Amazon Elastic File System (Amazon EFS) proporciona almacenamiento de archivos de forma sencillo y escalable para su uso con instancias de Amazon EC2 en la nube de AWS

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

(servicio Web de Amazon). Amazon EFS, posee una capacidad de almacenamiento es elástica con límite de 10TB, es compatible con la versión 4 (NFSv4.1 y NFSv4.0) del protocolo Network File System. Las instancias Amazon EC2 pueden tener acceso a un sistema de archivos Amazon EFS simultáneamente [Amazon EFS 2019].

5.4.1.2.2. Microsoft Azure File Storage

Este servicio ofrece archivos compartidos totalmente administrados que utilizan el protocolo estándar SMB 3.0 (*Server Message Block***)**. Azure Storage ofrece un almacén de objetos masivamente escalable para objetos de datos, un servicio de sistema de archivos para la nube, un almacén de mensajería para mensajería confiable y un almacén NoSQL. Por otro lado, sus prestaciones de almacenamiento se separan en 128GB, 512GB, 1TB, 2TB, 4TB y 8TB, muy amplia a la hora de elegir la que más convenga con la aplicación de estacionamiento [Azure Almacenamiento 2019].

5.4.1.2.3 IBM File Storage

Almacenamiento de archivos respaldado por Flash basado en NFS con IOPS *(*Input/Output Operations Per Second, operaciones de entrada/salida por segundo*, es una medida del rendimiento de referencia común para los dispositivos informáticos)*. Suministre almacenamiento de hasta 12 TB con un máximo de 48000 IOPS, cifrado para datos inactivos, instantáneas y replicación, duplicación de volumen, volúmenes expansibles y IOPS ajustables están disponibles actualmente en las regiones de EE. UU., UE, Australia, Canadá, Latinoamérica y Asia Pacífico. [File Storage- IBM 2018]

Los volúmenes de File Storage se pueden suministrar con dos opciones:

- Suministro de niveles de **Resistencia** que presentan niveles de rendimiento predefinidos y otras características como instantáneas y réplica.
- Crear un entorno de **Rendimiento** de alta potencia con operaciones de entrada/salida asignadas por segundo (IOPS).

5.4.1.2.4. Conclusión

Para concluir, los prestadores de servicios facturan de acuerdo con lo utilizado. Es de fundamental importancia analizar que Azure proporciona máximamente 8TB, mientras que IBM provee 12TB, y Amazon EFS proporciona un servicio "Elástico" con un límite de 10 TB, asimismo podemos agregar que IBM File Storage posee dos opcionalidades, niveles de resistencia y entorno de rendimiento mencionados con anterioridad en 5.4.2.3 lo que la hace candidata a elegirla.

5.4.2. Plataforma IoT

Estas plataformas IoT que a continuación abordaremos son servicios PaaS.

5.4.2.1. Aspectos de una plataforma IoT

Al momento de analizarlas un conjunto de cuestiones deben ser tenidas en cuenta, en los siguientes puntos nombraremos cada una de ellas.

5.4.2.1.1. Robustez

Una buena plataforma debe ofrecer un máximo de funcionalidad, la plataforma debe permitir la comunicación entre los dispositivos y los protocolos de conexión de maquina a máquina necesarios.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

5.4.2.1.2. Compatibilidad del proveedor

La plataforma debe integrarse fácilmente a su sistema de backend (administrador del sitio con sus respectivos sistemas) y, además de la interoperabilidad entre dispositivos, debe haber una fuerte comunicación entre aparatos y sistemas externos. También debe permitir la utilización de APIs (Interfaces de Programa de Aplicación) para interactuar y compartir datos con otras empresas.

5.4.2.1.3. Seguridad

La seguridad es una preocupación principal al elegir una plataforma cada componente de la plataforma debe tener sus propias opciones de seguridad, tales como SSL (Secure socket Layer).

5.4.2.1.4. Empaquetado

Otros de los factores con una consideración importante es el empaquetado, es decir cuales son los protocolos de IoT que la plataforma soporta.

5.4.2.2. Comparativas de plataformas

Realizada la enumeración de los principales factores a tener en cuenta al momento de seleccionar una plataforma de IoT, continuaremos con la comparación entre las distintas soluciones ofrecidas.

5.4.2.2.1. **IBM Watson IoT Platform** [IoT Watson – IBM 2018]

A través de IBM Watson se puede conectar y controlar sensores, aparatos, viviendas e industrias IoT. Proporciona un conjunto de herramientas integradas y de complementos, en pocas palabras, es una herramienta que facilita la conexión de dispositivos con la Nube. Se pueden hacer uso de herramientas integradas y externas para la limpieza y transformación de datos, el resumen y el aprendizaje automático, entre otros. Watson IoT Platform Analytics (complemento) puede ser utilizado para permitir a los usuarios de negocio interactuar fácilmente con los datos de métrica en bruto procedentes de entidades IoT utilizando funciones de análisis configurables incorporadas. A su vez a lo que se refiere al empaquetado los dispositivos, las pasarelas y las aplicaciones se pueden conectar a Watson IoT Platform utilizando el protocolo MQTT o HTTP. Las conexiones pueden ser seguras o no seguras, lo que implica la utilización de protocolos como TLS, el a tabla N°2 se presenta un resumen de lo expuesto.

Tipo de conexión	Protocolo	Número de puerto
No segura*	MQTT y HTTP	1883 u 80
Secura (TLS)	MQTT y HTTPS	8883 o 443

Tabla 2. Tipos de conexiones, empaquetamiento y puertos.

Si bien es una de las mejores prestadoras de plataformas IoT, pero tiene una gran desventaja, que este producto no está disponible para Argentina, lo que hace imposible

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

elegirla como opción, pero es importante tenerla en cuenta, ya que en un futuro seguramente sea aprobada en el país en cuestión, o puede ser tomada en cuenta para otros países.

5.4.2.2.2. Azure IoT Hub [Azure hub – Microsoft 2018]

IoT Hub es un servicio administrado, hospedado en la nube, que actúa como centro de mensajes para comunicaciones bidireccionales entre la aplicación de IoT y los dispositivos que administra.

Proteger las soluciones IoT requiere garantizar el aprovisionamiento seguro de los dispositivos, proteger la conectividad entre estos dispositivos y la nube y garantizar la protección de los datos en la nube durante su procesamiento y almacenamiento. Azure IoT Suite protege los dispositivos proporcionando una clave de identidad única a cada uno, para establecer una comunicación única, con esto se gana que los dispositivos no acepten conexiones de red no solicitadas. Los dispositivos solo se conectan a servicios conocidos con los que están emparejados, o con los que tienen rutas establecidas, como un centro de IoT de Azure. La eficacia es un factor importante para garantizar la conservación de recursos y el funcionamiento en un entorno con recursos limitados. HTTPS (HTTP seguro), la versión segura estándar del sector del conocido protocolo HTTP, se admite en Azure IoT Hub, lo que permite una comunicación eficaz. Advanced Message Queuing Protocol (AMQP) y Message Queuing Telemetry Transport (MQTT), admitidos en Azure IoT Hub, están diseñados no solo para la eficacia en términos del uso de recursos, sino también para la entrega confiable de mensajes.

5.4.2.2.3. AWS IoT [AWS IoT – AWS 2019]

AWS IoT proporciona comunicación bidireccional entre dispositivos conectados a Internet, como sensores, actuadores, microcontroladores integrados o dispositivos inteligentes y la nube de AWS. Proporciona un mecanismo de publicación y suscripción de mensajes, que están manejados con el protocolo MQTT, a su vez proporciona integración con otros servicios de AWS. Se puede apreciar que AWS IoT dispone de funciones de seguridad para el envío de mensajes, este debe estar cifrado sobre el protocolo Transport Layer Security (TLS), que garantiza la confidencialidad de los protocoles de aplicación (MQTT, HTTP), además los dispositivos deben estar conectados utilizando certificado X.509 sobre una conexión segura para autenticar y autorizar las acciones de la cuenta. En simples palabras, se debe cumplir reglas que impone AWS donde el centro de la atención es la seguridad y confidencialidad de los datos y dispositivos.

AWS IoT proporciona tres formas de aprovisionar dispositivos:

- Aprovisionamiento de un solo objeto con una plantilla de aprovisionamiento. Esta es una buena opción si solo necesita aprovisionar los dispositivos de uno en uno.
- Aprovisionamiento "just-in-time" (JITP) con una plantilla que registra y aprovisiona un dispositivo cuando se conecta por primera vez a AWS IoT. Esta es una buena opción si necesita registrar un gran número de dispositivos, pero no dispone de información sobre ellos que se pueda incluir en una lista de aprovisionamiento por lotes.
- Aprovisionamiento por lotes. Esta opción le permite especificar una lista de valores de aprovisionamiento de un solo objeto que se almacenan en un archivo en un bucket de S3(Servicio de almacenamiento de AWS). Este enfoque funciona bien si tiene un gran número de dispositivos conocidos cuyas características puede incluir en una lista.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

5.4.2.2.4. Selección de Plataforma

Para establecer una selección de producto. Hacemos una tabla comparativa con pesos

	Peso	IBM Watson IoT		Azure IoT Hub		AWS IoT	
Características		Nota	Puntaje	Nota	Puntaje	Nota	Puntaje
Robustez	30%	8	2.4	9	2.7	7	2.1
Compatibilidad del proveedor	20%	3*	0.6	7	1.4	8	1.6
Seguridad	40%	9	3.6	7	2.8	7	2.8
Empaquetado	10%	7	0.7	8	0.8	7	0.7
TOTAL	100%	27	7.3	31	7.7	29	7.2

Tabla 3. comparativa ponderada

Con la tabla ponderada, estableciendo las características marcadas en el comienzo del punto 5.4.2. se puede llegar a la conclusión del resultado obtenido que obtiene como ganadora a

Tanto IBM, Azure y AWS IoT publican mensajes de eventos cuando se producen determinados eventos. Por ejemplo, el registro genera eventos cuando se añaden, actualizan o eliminan objetos. Cada evento provoca que se envíe un único mensaje de evento. Los mensajes de evento se publican a través de MQTT con una carga JSON. El contenido de la carga depende del tipo de evento.

Para recibir mensajes de eventos, el dispositivo debe usar una política adecuada que le permita conectarse a la Gateway de dispositivos de IoT y suscribirse a los temas de eventos de MQTT. También debe suscribirse a los filtros de temas adecuados.

5.5. Conclusión Final

Como se ha mostrado se llega a la conclusión que los tres servicios poseen similitudes en sus prestaciones. La determinación de optar por una u otra va a depender siempre de la necesidad y la adaptación a la aplicación de estacionamiento. Una base de datos veloz, compacta y confiable, Azure en lo que compete a base de datos, tiene mayor prestigio y un servicio afable a los usuarios de su sistema operativo (Windows). Con lo dicho anteriormente, la opcionalidad que dispone IBM es aún más amplia ya que, en su paquete estándar grande, ofrece mayor tamaño hasta 2TB. En cuanto al almacenamiento Amazon, es más flexible con sus servicios de cobro por uso, en cambio IBM al brindar mayor capacidad de almacenamiento, sustrae ventaja a sus competidores. Sin embargo, son muy similares y con dificultades para su elección. En cuanto a la selección de la plataforma IoT, IBM queda descartada ya que no está disponible en Argentina, y entre Azure y AWS. Para este caso en particular, para la aplicación de estacionamiento, la facultad al tener establecido un convenio con la empresa Microsoft, direcciona la elección hacia elegir servicios de Azure.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

6. Características de Hardware

Una vez elegida el prestador de servicio, se deberá conocer cuáles son los requisitos tanto del hardware como los requisitos de la prestadora de servicio. A continuación, se detallará algunas cuestiones de hardware.

6.1. Características de red

Para conectarse a la base de datos de Un servidor de Azure SQL Database escucha en el puerto 1433. Se necesita detalles de base de datos (como el nombre de host), primero se debe tener instalado y corriendo SQL Server Management Studio (SSMS). Dentro de un ordenador para hacer las configuraciones. Y luego poder acceder por el puerto 1433, en simples palabras se debe fabricar la base de datos para luego poder accederla y trabajarla como una ABM (alta, baja, modificación). Con lo que respecta a lo planteado para el proyecto de estacionamiento. El mismo será accedido desde el dispositivo directamente sin un ordenador de por medio, esto conlleva a utilizar protocolos de comunicación, tales como HTML (Lenguaje de Marcas de Hipertexto) o MQTT (un protocolo de mensajería basado en ISO estándar de publicación-suscripción) que explicaremos con profundidad en el punto 6.3. Haciendo hincapié en el proyecto de estacionamiento, recordar en pocas palabras que este dispositivo cuenta con varios sensores trabajando todos en conjunto. Por ende, necesita una conexión liviana y precisa, quiere decir que solo envíe y reciba paquetes de datos cuando sea necesario y en un formato plano y practico como lo es JSON (JavaScript Object Notation) que es un formato de texto sencillo para el intercambio de datos.

6.2. Seguridad [Proteger IoT- RedesZone 2017]

Es de muy importancia tener en cuenta la seguridad cuando se conectan los dispositivos de IoT a internet. Ya que, si el dispositivo no es asegurado, puede ser vulnerable a accesos desde cualquier lugar sin permisos apropiados, produciendo robos de información o modificaciones en las configuraciones. Por ejemplo, dentro de dispositivos centrales del estacionamiento, si no están bien asegurados con contraseñas encriptadas o contraseñas no obvias (por ejemplo: 1234, admin o contraseña) se generaría una primera capa de seguridad luego es importante que los mensajes que son enviados o recibidos desde el dispositivo de estacionamiento y la nube estén encriptados de extremo a extremo, para que no se produzca robo o alteración de la información. Por otra parte, prevenir que el dispositivo no se convierta en un Botnet (se explicara dentro de este capítulo).

6.2.1.1 Prevenir la violación del sistema o que se ponga en peligro. [Seguridad IoT –IBM 2017]

Cada nivel de la aplicación IoT debe implementar medidas preventivas efectivas para mantener alejados a los hackers. Por ejemplo, se necesita *endurecer* el dispositivo para asegurar que la comunicación entre el dispositivo y la nube sea segura. Se puede lograr endurecer el dispositivo, produciendo encriptación de extremo a extremo, o haciendo uso de contraseñas dinámicas (van cambiando cada cierto tiempo, determinado por el programador), que lograría un sistema seguro y difícil de penetrar.

6.2.1.2. Soportar una supervisión continua.

Incluso los sistemas más seguros todavía tienen muchas vulnerabilidades. Además, la mejor solución segura actual (tanto de hardware como de software) puede no ser suficientemente buena para prevenir ataques en el futuro. Por lo tanto, se debe complementar las medidas de seguridad con una supervisión continua y la constante actualización del sistema para protegerlo contra las más recientes formas de ataques. Realizar, cada determinado tiempo

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

(puede ser días, semanas o meses) una actualización, para el dispositivo de Estacionamiento como para la aplicación móvil, con realizar actualizaciones semanales o mensuales estaría bien.

6.2.1.3 Ser resiliente.

Finalmente, si ocurriese una violación, se debe minimizar el daño y el sistema se debe recuperar tan rápido como sea posible. El riesgo siempre está latente y puede ocurrir en cualquier momento, por eso siempre se tiene que contar con un plan sea de mitigación (hacer algo todo el tiempo para que no ocurra, por ejemplo, el punto 6.2.1.2 supervisión continua) o un plan de contingencia (tener considerado un plan "B" por si la violación al dispositivo ya ha ocurrido, por ejemplo, bloquear el dispositivo, con activación manual por personal de soporte), es importante, tener un plan para actuar luego de haber ocurrido el riesgo.

6.2.2 aplicaciones del IoT seguras

La mayor parte de las soluciones IoT están formadas por tres niveles principales. Los componentes de la solución IoT que son ejecutados en cada nivel tienen que incorporar medidas de seguridad específicas para protegerse contra múltiples vulnerabilidades.

6.2.2.1. Nivel de Dispositivos/Gateways

Proteger contra un servidor "falso" que envía comandos maliciosos, o proteger contra un hacker que intenta escuchar los datos de sensores privados que están siendo enviados desde los dispositivos. El dispositivo central de

6.2.2.2. Nivel de Red/Transporte

Proteger contra un dispositivo "falso" que envía mediciones falsas que pueden corromper los datos que persisten en la aplicación. Las consideraciones de seguridad para este nivel serán discutidas en la Parte 2.

6.2.2.3. Nivel de aplicaciones

Proteger contra el uso inválido de datos, o proteger contra la manipulación de los procesos analíticos que se ejecutan en el nivel de la aplicación.

- Autenticación del ID/Contraseña de Usuario
- Autenticación de Una sola contraseña (OTP)
- Autenticación de ID único de servidor
- Autenticación de carga de mensaje

6.2.3. Botnet

El término "*botnet"* nace de dos palabras en inglés: "*bot"* por "robot", mientras que "*net"* proviene de *network* o red en español. En resumen, se trata de un **robot de red.** Es lo que ocurre si el dispositivo de IoT es accedido de manera ilícita, con fines o no nocivos, es posible que, de no poseer seguridad apropiada en el dispositivo de estacionamiento, genere un punto libre para la creación de un botnet, sin la necesidad de que sea autorizado o el conocimiento del propietario del dispositivo. Es importante, tener bien definidos, los tamaños de paquete de datos, quien es el que publica y quienes son los suscriptores de mensaje (estructura MQTT explicada en el punto 6.3.1)

6.2.4 Autorización de aplicaciones con OAuth 2.0

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

Para el proyecto en cuestión se puede utilizar un mecanismo de autorización centralizado para los dispositivos MQTT(será explicado a continuación en el punto 6.3.1 protocolos de conexión), podrán utilizar una infraestructura basada en OAuth. OAuth 2.0 habilita la separación del servidor de autorización del servidor de recursos, como un servidor MQTT. Cuando se utiliza OAuth 2.0, el cliente presenta sus credenciales al servidor de autorización, quién entonces realiza la verificación de autenticación y devuelve un token de acceso que concede el permiso para acceder a un recurso.

El token de acceso es utilizado para conectar al servidor MQTT. El servidor MQTT valida el token de acceso, usualmente mediante la comunicación con el servidor de autorización, después otorga acceso al recurso. El flujo está representado en el siguiente diagrama:

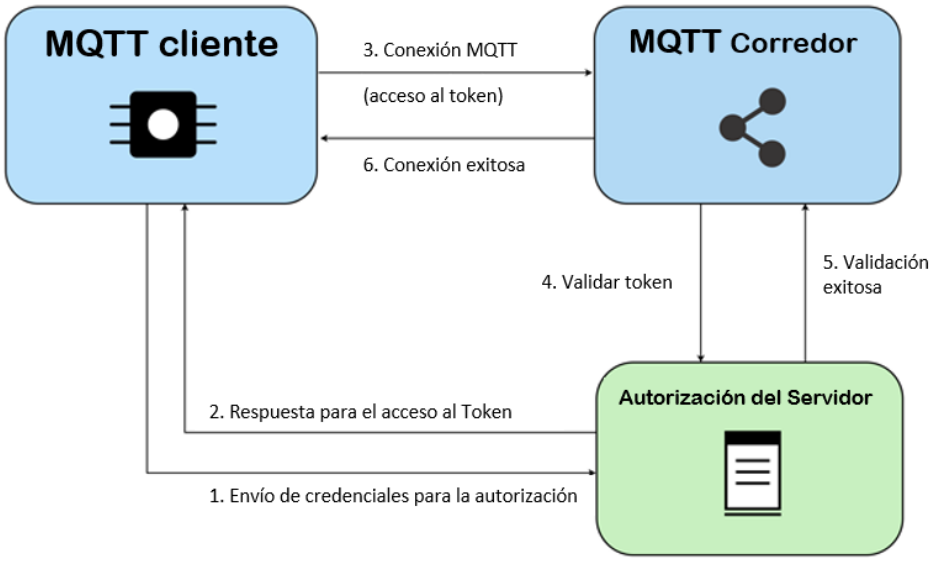

Fig 7. Validación de token de acceso

6.2.5. Conclusión

La seguridad es un factor importante a tener en cuenta desde el comienzo del proyecto, para el proyecto de estacionamiento, se debe considerar un dispositivo solido con seguridad de extremo a extremo y validaciones de token (explicado en la fig 7), por otra parte, si el riesgo se activó y se produjo la vulnerabilidad. Como se mencionó en el punto 6.2.1.3 lo que se refiere a enfrentar el riesgo ya ocurrido como mitigación y contingencia. En definitiva, la seguridad juega un papel importante en la calidad del producto, por eso utilizaremos OAuth 2.0 para el envío de datos, y contraseñas encriptadas.

6.3 Conexión

Para poder conectar el modulo central del sistema de estacionamiento con los servicios en la nube de Azure, se deberá poseer conectividad a internet. Por ello, se necesita un dispositivo tal, que pueda conectar la unidad de procesamiento con la nube inalámbricamente. Seguidamente se verá los siguientes ATSAMW25 y ATWINC1500, primero describiremos que es el wifi (Wireless Fidelity -Fidelidad inalámbrica).

6.3.1 Requerimientos de servicio de Azure

Para sincronizar la información de estado entre un dispositivo y un centro de IoT, se utilizará *dispositivos gemelos*. Un dispositivo gemelo es un documento JSON, asociado con un dispositivo específico y almacenado por IoT Hub en la nube en donde

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

puede consultarlo. Un dispositivo gemelo contiene *propiedades deseadas*, *propiedades notificadas* y *etiquetas*.

Una propiedad deseada la establece una aplicación back-end y la lee un dispositivo. Una propiedad notificada la establece un dispositivo y la lee una aplicación back-end. Una etiqueta la establece una aplicación back-end y nunca se envía a un dispositivo. Las etiquetas se usan para organizar los dispositivos.

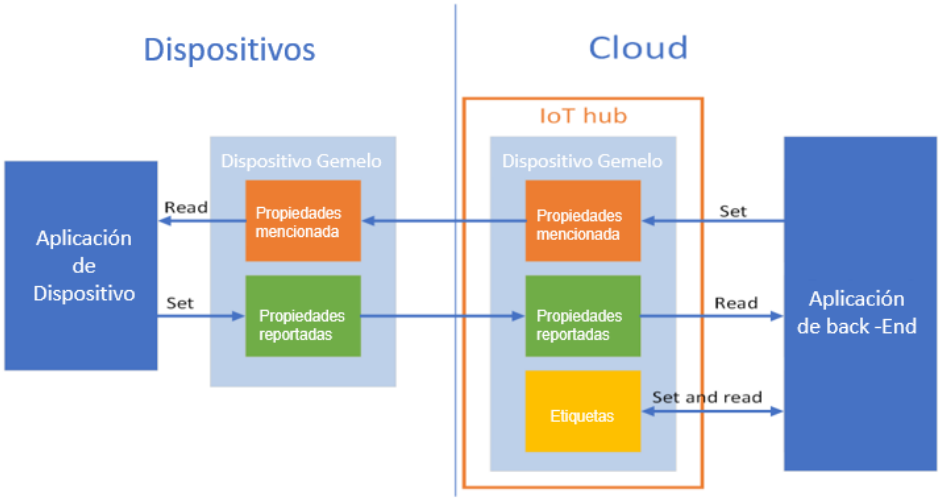

Fig 7. Diagrama de conexión de dispositivo con IoT Hub

para poder efectuar la conexión se debe poseer una suscripción de Azure, también tiene que contener un centro de IoT con un dispositivo agregado en el registro de identidad del dispositivo. La entrada en el registro de identidad del dispositivo permite que el dispositivo se conecte con el centro.

Para enviar la información de estado de una aplicación back-end a un dispositivo se usan las propiedades deseadas.

6.3.1 Protocolo de conexión

6.3.1.1 MQTT

MQTT es el protocolo de mensajería más popular para los dispositivos y aplicaciones IoT, y es soportado por muchos de los principales actores en el campo del IoT. MQTT proporciona un protocolo de comunicaciones ligero y fácil de usar para las soluciones IoT.

El propio MQTT especifica algunos mecanismos de seguridad, pero todas las implementaciones comunes soportan los estándares de seguridad de última generación, tales como SSL/TLS para la seguridad en el transporte. Para sus aplicaciones, el MQTT no impone la utilización de un enfoque de seguridad en particular.

La mayor parte de los despliegues del MQTT utilizan la seguridad en la capa de transporte (TLS), así que los datos son cifrados y se valida su seguridad. De igual manera, para controlar el acceso, la mayor parte de las implementaciones del MQTT (incluyendo la que se encuentra en IBM Watson IoT Platform) también utilizan las funciones de autorización del servidor MQTT.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

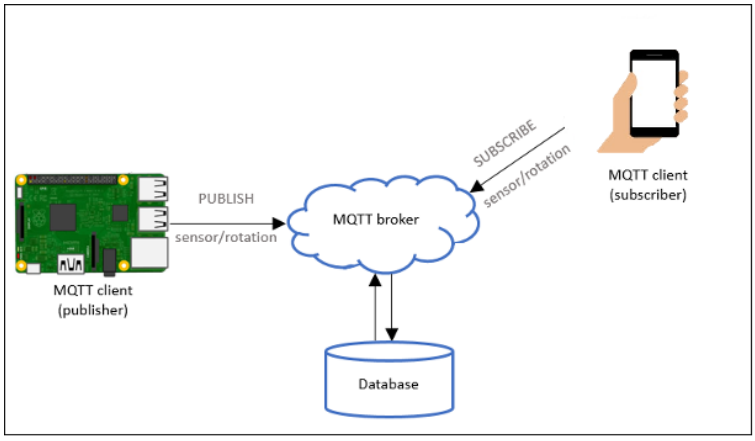

Fig 8. MQTT

La forma en que serán conectados el modulo y la nube de Azure será con el protocolo MQTT para ello, necesitamos la librería de **Pubsubclient.h**

6.3.2 **Pubsubclient** [Librería MQTT – knolleary 2018]

6.3.2.1 Hardware Compatible

La biblioteca utiliza la API de Arduino Ethernet Client para interactuar con el hardware de red subyacente. Esto significa que solo funciona con un número creciente de tablas y escudos, que incluyen:

- Arduino Ethernet
- Arduino Ethernet Shield
- Arduino YUN - use el incluido YunClienten lugar de EthernetClient, y asegúrese de hacer una Bridge.begin() primera
- Arduino WiFi Shield: si desea enviar paquetes> 90 bytes con este escudo, habilite la MQTT_MAX_TRANSFER_SIZEdefinición en PubSubClient.h.
- Sparkfun WiFly Shield - biblioteca
- TI CC3000 WiFi - biblioteca
- Intel Galileo / Edison
- ESP8266
- ESP32

Usando las librerías de arduino

- conexión PubSubClient>mqtt_auth,
- PubSubClient>mqtt_basic,
- PubSubClient>mqtt_publish_in_callback

6.3.2.2 Arduino Client para MQTT

Esta biblioteca proporciona un cliente para realizar mensajes simples de publicación / suscripción con un servidor que admite MQTT.

Para la comunicación entre las partes, se necesitará un lenguaje que hablaran ambos extremos a continuación se explicara

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

6.4. JSON

JSON soporta dos tipos de estructuras, una de ellas son objetos que contienen una colección de pares llave-valor y el otro tipo se trata de arrays de valores. Esto proporciona una gran sencillez en las estructuras.

1. JSON no tiene espacios de nombres, cada objeto es un conjunto de claves independientes de cualquier otro objeto.
2. JSON no necesita ser extensible por que es flexible por sí solo. Puede representar cualquier estructura de datos pudiendo añadir nuevos campos con total facilidad.
3. JSON es mucho mas simple que XML, el cual proporciona pesadas tecnologías que le avalan (Scheme, XSLT, XPath).
4. JSON es optimista y no requiere de este tipo de tecnologías, confía en el desarrollador.

Para que la conexión sea establecida se precisará de un medio de comunicación por ello la que se adapta más a la del proyecto del Estacionamiento es la tecnología de conexión inalámbrica Wifi, que se explicará a continuación.

6.5 WIFI

El wifi es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con wifi como Arduino, pueden conectarse a internet a través de un punto de acceso de red inalámbrica. [Wifi – arduino 2016]

Wi-Fi es una marca de la Alianza Wi-Fi, la organización comercial que adopta, prueba y certifica que los equipos cumplen con los estándares 802.11 relacionados a redes inalámbricas de área local.

- Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n disfrutan de una aceptación internacional debido a que la banda de 2,4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente.
- En la actualidad ya se maneja también el estándar IEEE 802.11ac, conocido como WIFI 5, que opera en la banda de 5 GHz y que disfruta de una operatividad con canales relativamente limpios.

Existen varias alternativas para garantizar la seguridad de estas redes. Las más comunes son la utilización de protocolos de cifrado de datos para los estándares wifi como el WEP, el WPA, o el WPA2 que se encargan de codificar la información transmitida para proteger su confidencialidad, proporcionados por los propios dispositivos inalámbricos. La mayoría de las formas son las siguientes:

WEP, cifra los datos en su red de forma que sólo el destinatario deseado pueda acceder a ellos. Los cifrados de 64 y 128 bits son dos niveles de seguridad WEP. WEP codifica los datos mediante una “clave” de cifrado antes de enviarlo al aire. Este tipo de cifrado no está recomendado debido a las grandes vulnerabilidades que presenta ya que cualquier cracker puede conseguir sacar la clave, incluso aunque esté bien configurado y la clave utilizada sea compleja.

WPA: presenta mejoras como generación dinámica de la clave de acceso. Las claves se insertan como dígitos alfanuméricos.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

WPA2 (estándar 802.11i): que es una mejora relativa a WPA. En principio es el protocolo de seguridad más seguro para Wi-Fi en este momento. Sin embargo, requieren hardware y software compatibles, ya que los antiguos no lo son. Utiliza el algoritmo de cifrado AES (Advanced Encryption Standard).

IPSEC (túneles IP) en el caso de las VPN y el conjunto de estándares IEEE 802.1X, que permite la autenticación y autorización de usuarios.

Filtrado de MAC, de manera que solo se permite acceso a la red a aquellos dispositivos autorizados. Es lo más recomendable si solo se va a usar con los mismos equipos, y si son pocos.

Ocultación del punto de acceso: se puede ocultar el punto de acceso (router) de manera que sea invisible a otros usuarios.

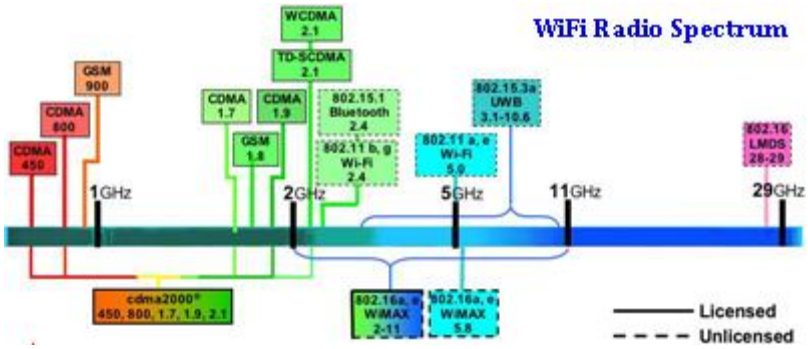

Fig 10 Espectro radio WIFI

6.6. Módulos

6.6.1. ATSAMW25 [Librerías Arduino-Arduino 2016]

El ATSAMW25 está compuesto por tres principales bloques:

- SAMD21 Cortex-M0+ 32bit low power ARM MCU
- WINC1500 low power 2.4GHz IEEE® 802.11 b/g/n Wi-Fi (mismo que el wifi 101 shield)
- ECC508 CryptoAuthentication

6.6.1.1 Características:

- Unibanda de 2.4 GHz b/g/n
- Aceptación mundial: FCC (USA), CE (Europa) y TELEC
- Espacio compacto: 33.8 mm x 14.9 mm
- Voltaje de funcionamiento de 2.7 V a 3.6 V
- Servicios de red: DHCP, DNS, TCP/IP (IPv4), UDP, HTTP, HTTPS
- Cumple con la directiva RoHS.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

- Protocolos de seguridad admitidos: WPA / WPA2 personal, TLS, SSL
- Interfaz de host serial SPI o UART
- Consumo extremadamente bajo en los órdenes de los microAmperes

6.6.2 ATWINC1500

6.6.2.1 Características

- IEEE® 802.11 b / g / n 20MHz.
- Funciona en la banda ISM de 2,4 GHz
- Interruptor Integrado PA
- Antena PCB Integrado
- Sensibilidad superior y rango a través de procesamiento de señales PHY avanzada
- Nivelación de avanzada y el Canal de Estimación
- Portador de avanzada y sincronización de tiempo
- Wi-Fi Direct y apoyo Soft-AP
- Soporta IEEE 802.11 WEP, WPA, WPA2 Seguridad
- Compatible con la seguridad de China WAPI
- rendimiento Superior MAC a través de hardware acelerado de dos niveles
- A-MSDU / A-MPDU agregación de tramas y el bloque de reconocimiento
- Motor de gestión de memoria on-chip para reducir la carga de acogida
- Soporta las interfaces de comunicación SPI, UART, y I2c
- 2 o 3 cables de interfaz Bluetooth® convivencia
- Temperatura de funcionamiento de -40 ° C a + 85 ° C
- E / S de tensión de funcionamiento de 2.7V a 3.6V
- Memoria flash integrada de software de sistema
- Ahorro de energía Modos

Para poder conectar el modulo con internet se necesita las librerías correspondientes. Con el Arduino WiFi Shield, esta biblioteca permite que una placa Arduino se conecte a Internet. Puede servir como un servidor que acepta conexiones entrantes o un cliente que realiza conexiones salientes. La biblioteca admite el cifrado personal WEP y WPA2, pero no WPA2 Enterprise. También tenga en cuenta que, si el SSID no se transmite, el escudo no se puede conectar.

6.7 Conclusion

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

Para que exista la conexión con el servicio de Azure sea IoT Hub y la base de datos, se deberá conectar de manera MQTT con el tipo de conectividad TCP/IP, necesitando el host o url de nuestro servidor de base de datos, además, también un nombre de usuario con permisos en la base de datos y la contraseña del mismo, los datos enviados deben estar en formato JSON por medio del puerto 1433. Además, se deberá tener en cuenta la seguridad mencionadas en el punto 6.2 para poder tener una aplicación de estacionamiento rígida y no vulnerable.

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

Bibliografía

- [Richard Stallm 2008]«Cloud computing is a trap, warns GNU founder Richard Stallman» en TheGuardian https://www.theguardian.com/technology/2008/sep/29/cloud.computing.richard.stallman
- [IEEE 2008]The Cloud Is The Computer https://spectrum.ieee.org/computing/hardware/the-cloud-is-the-computer

  [AWS Aurora 2017] Documentación Amazon web Services https://d1.awsstatic.com/whitepapers/es_ES/aws-overview.pdf

- [ARCHIVE 2010]«¿Cómo empezó el Cómputo Cloud?» https://web.archive.org/web/20100115083643/http://www.itnews.ec/news/000396.aspx

- [Comparativas 2019] AWS Aurora vs Azure https://db-engines.com/en/system/Amazon+Aurora%3BMicrosoft+Azure+SQL+Database

- [Amazon EFS 2019] ¿Qué es Amazon EFS (Elastic File System)? https://www.itcentralstation.com/products/amazon-efs-elastic-file-system-reviews

- [Máquinas Virtuales 2016] Azure VM vs Amazon EC2 vs Google CE https://www.cloudberrylab.com/resources/blog/azure-vm-vs-amazon-ec2-vs-google-ce-cloud-computing-comparison/

- [IBM Servidores virtuales 2018] IBM Servidores virtuales https://cloud.ibm.com/docs/vsi?topic=virtual-servers-provisioning-selections&locale=es#provisioning-selections

- [Azure Prestaciones 2019] rendimiento y los precios https://azure.microsoft.com/es-es/pricing/details/sql-database/managed/

- [File Storage- IBM 2018] Prestaciones de File Storage https://cloud.ibm.com/docs/infrastructure/FileStorage?topic=FileStorage-about#getting-started-with-file-storage

- [AWS-SQL 2019] Base de Datos https://aws.amazon.com/es/rds/mysql/pricing/

- [Azure Almacenamiento 2019] page blobs https://azure.microsoft.com/en-us/pricing/details/storage/page-blobs/

- [DB2Connect-IBM 2014] ftp://ftp.software.ibm.com/ps/products/db2/info/vr105/pdf/es_ES/DB2Connect-db2c0z1051.pdf

- [Librerías Arduino-a arduino 2016] librerías Arduino https://aprendiendoarduino.wordpress.com/category/librerias/

*La calificación establecida es 3 ya que este producto no está disponible en Argentina

- [Wifi – arduino 2016] Arduino con wifi https://aprendiendoarduino.wordpress.com/2016/11/12/wifi-en-arduino/

- [Librería MQTT – knolleary 2018] Librería de conexión suscriptor y publicador MQTT https://github.com/knolleary/pubsubclient

- [IoT Watson – IBM 2018] IBM Watson IoT Platform https://www.ibm.com/support/knowledgecenter/es/SSQP8H/iot/kc_welcome.html

- [Azure hub – Microsoft 2018] Azure IoT Hub https://docs.microsoft.com/es-es/azure/iot-hub/about-iot-hub

- [AWS IoT – AWS 2017] Amazon web services IoT https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html

- [Proteger IoT-redesZone 2017] proteger los dispositivos IoT de una forma adecuada

  https://www.redeszone.net/2017/01/02/proteger-los-dispositivos-iot-una-forma-adecuada/

- [Conexión a SQL database – Azure 2018] conexión con SSMS https://docs.microsoft.com/es-es/azure/sql-database/sql-database-connect-query-ssms
- [Seguridad IoT –IBM 2017] Protegiendo dispositivos y Gateway IoT

  https://www.ibm.com/developerworks/ssa/library/iot-trs-secure-iot-solutions1/index.html

*La calificación establecida es 3 ya que este producto no está disponible en Argentina