UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE

"Diseño e implementación de una red de estacionamientos inteligentes para bicicletas operados mediante celular"

Matías Ignacio Barrios Núñez

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRÓNICO

PROFESOR GUIA: **Marcos David Zúñiga Barraza**

PROFESOR CORREFERENTE: **Gabriel Torres Yarza**

The most profound technologies are those that disappear.
They weave themselves into the fabric of everyday life
until they are indistinguishable from it.
Mark Weiser, *The Computer for the 21st Century*[53]

Agradecimientos

Quisiera agradecer a todos aquellos que me han apoyado en este camino.

Mi familia, que siempre han estado presentes para darme las energías que necesité para terminar este extenso trabajo de memoria.

A mis papás, por su presencia constante en cada paso que he dado. Jamás podré terminar de agradecerles.

A mis hermanos, mis primeros aliados al momento de requerir ayuda.

A Rosalis Pulido, mi novia, por su paciencia y contención en los momentos más difíciles. Muchas gracias por todo el amor que me has dado.

Al profesor Marcos Zúñiga, director del programa de Memorias Multidisciplinarias, ya que ha estado apoyando este proyecto desde el día 1. Gracias al programa que el dirige he podido desarrollar habilidades que son imprescindibles en el camino de la innovación.

Al profesor Gabriel Torres, por sus consejos relativos a arquitectura de sistema y seguridad informática. Es una gran alegría saber que el Oficial de la Seguridad de Información de mi universidad es una persona tan apasionada por su trabajo.

Al profesor Ahmed Elmesiry, quien me hizo las preguntas precisas que me ayudaron a desarrollar los mecanismos criptográficos que he implementado en este trabajo. Gracias por introducirme en el fascinante mundo de la seguridad informática.

A los profesores Werner Creixell, Milan Derpich y Elías Tuma, quienes escucharon mis preguntas relativas a sistemas computacionales, teoría de la información y matemáticas discretas, respectivamente. Sin sus comentarios, habría sido muy difícil desarrollar el generador de números pseudo-aleatorios que he utilizado como herramienta de comunicación entre dispositivos IoT.

A Jaime Reinoso Vadillo, un gran amigo que hoy descansa en paz. Gracias por darme un segundo hogar. Me duele mucho saber que jamás pudimos resolver nuestras diferencias antes de tu partida, pero estoy seguro que estarías tan orgulloso como yo de este logro que significa terminar mi carrera.

Resumen

Las tecnologías de información y comunicaciones tienen el poder de mejorar la vida de las personas, pero su implementación requiere entender bien cual es el problema que se está solucionando. Es por ello que es imprescindible un riguroso proceso de diseño, mediante una investigación de las soluciones actuales al problema, establecimiento de una propuesta de solución y el desarrollo de un prototipo funcional. De esta forma, se pueden poner a prueba los supuestos y establecer de forma empírica los desafíos que se deben afrontar para implementar una solución que genere impacto real en la sociedad.

En el presente documento, se desarrolla dicho proceso en pos de solucionar el problema de la falta de estacionamientos seguros para bicicletas, presentado en el marco del programa de Memorias Multidisciplinarias de la Universidad Técnica Federico Santa María.

En la primera parte, se presentan las soluciones actuales al estacionamiento de bicicletas. Se hace un listado de los sistemas existentes alrededor del mundo y los presentes en Chile, para luego estudiar sus ventajas, desventajas y variables de diseño a considerar. Luego se establecen los requisitos del sistema, y se desarrolla una gestión de riesgos preliminar para entender las restricciones de la solución.

En la segunda parte, se desarrolla la solución preliminar al problema. Se establece la arquitectura del sistema y se diseñan los módulos e interfaces necesarias para su funcionamiento.

En la tercera parte, se establece el modo de implementación del prototipo funcional. Se hace una revisión de los hitos principales del proyecto, una revisión del diseño y la estructura final del prototipo funcional. Se expone además los resultados de la implementación de los módulos fundamentales para el funcionamiento del sistema.

Abstract

Information and communication technologies have the power to improve people's lives, but their implementation requires knowing what is the problem that is being solved. That is why a rigorous design process is essential, through a research of current problem solutions, establishment of a preliminary solution and development of a functional prototype. In this way, the assumptions can be tested and establish the empirical form in which the challenges must be faced to implement a solution that generates real impact in society.

In this document, the process was presented in the position of solving the problem of the lack of safe parking for bicycles, presented in the framework of the program of Multidisciplinary Memories of the Federico Santa María Technical University.

In the first part, the current solutions to bicycle parking are presented. A list is made of the existing systems in the world and those present in Chile, to then study their advantages, disadvantages and design variables to be considered. Then the system requirements are established, and a preliminary risk management is developed to understand the restrictions of the solution.

In the second part, the preliminary solution to the problem is developed. The architecture of the system is established and the modules and interfaces necessary for its operation are designed.

In the third part, the mode of implementation of the functional prototype is established. A review is made of the main milestones of the project, a review of the design and the final structure of the functional prototype.

Glosario

- 2FA: *2 Factor Authentication*. Sistema mediante el cual se pide una segunda contraseña que permite verificar autenticidad en la petición al sistema.
- AM: Aplicación móvil.
- AWS: Amazon Web Services.
- MITM: Esquema de ataque informático, en que el atacante se ubica en la comunicación entre dos puntos, sin que ambas partes detecten su presencia, i.e. los paquetes de comunicación fluye desde y hacia el atacante, quien redirige los mismos a sus destinatarios originales.
- EB: estación base. Es un grupo de plazas de estacionamiento (PDE) ubicada en cierto lugar. Las estaciones base cuentan con un IoTD (*Internet of Things Device*) para abrir o cerrar las PDE.
- ERL: *Electronic Rotary Latch*. Dispositivo electrónico que será usado en las PDE para abrir y cerrar mediante una orden del IoTD de la PDE.
- ETL: *Extract, Transformation and Load*. Proceso mediante el cual se hace uso de la información disponible en la base de datos para la toma de decisiones, además de la modifiación presente en a base de datos.
- IoTD: *Internet of Things* device. Es un dispositivo electrónico con capacidad de conexión a internet. Se utilizan para operar los estacionamientos de forma remota.
- ISP: *Internet Service Provider*. Proveedor de servicios de internet.
- LDI: Llave de instrucción. Llaves que los IoTD tienen configuradas para interpretar los mensajes provenientes del servidor.

- MQTT: *Message Queue Telemtry Transport*. Protocolo comunicación para IoTD, diseñado para disminuir el volumen de tráfico en la comunicación.
- PDE: plaza de estacionamiento. Unidad física donde se aparcan las bicicletas de manera individual. Cuenta con un IoTD para comunicarse mediante I2C con la EB.
- PK: *Private Key*, Llave privada. Se utilizan para generar números en el PRNG.
- PRNG *Pseudo random number generator*. Generador de números pseudo-aleatorios.
- RFID: Radio Frecuency Identification.
- SLA: Service Level Agreement. Contrato de disponibilidad de servicio.
- SVP: Sistema de verificación de presencia.
- TIC: Tecnologías de Información y Comunicaciones.
- Trade-off: situación en que se deben aceptar ciertas pérdidas, a cambio de una ganancia que produce mayor valor.
- VPS: *Virtual Private Server*. Servicio de pago para alojamiento de servicios web.

Índice general

1. INTRODUCCIÓN

El presente documento tiene por objetivo registrar el proceso de diseño de una red de estacionamiento para bicicletas operado mediante celular.

Se establece la definición técnica del problema, el estado del arte de las soluciones, una propuesta de solución pensada para el problema identificado y una metodología de implementación de dicha solución.

Se trabaja bajo la metodología de diseño KISS[4] mediante la cual los esfuerzos se enfocan en obtener una solución simple pero efectiva, evitando cualquier tipo de desarrollo que impida un crecimiento escalable del sistema.

En pos de la sustentabilidad y escalabilidad, se plantean como requisitos de diseño la utilización de bajos niveles de energía para el funcionamiento del estacionamiento, y el uso de tecnología de información para que sea de fácil acceso para los ciclistas.

En este documento no se diseña la estructura física de dicho estacionamiento ni un modelo de negocios que permita su sostenibilidad económica, si no que una arquitectura de sistema que permita que la implementación del servicio sea lo más sencilla posible, sin perder de vista la seguridad de la información.

Además, se diseña el servicio en función de que la operación del estacionamiento sea lo más sencilla posible para el ciclista, y lo más fácil de mantener para el dueño o administrador del estacionamiento.

2. DEFINICIÓN TÉCNICA DEL PROBLEMA

2.1. Análisis del desafío

2.1.1. Problemas específicos a resolver

En la actualidad los ciudadanos de Chile cuentan con tres alternativas de movilización; automóvil, locomoción colectiva o bicicleta.

De estas tres alternativas, la bicicleta es la que sin duda genera la mayor cantidad de beneficios respectivos a sustentabilidad, pues según [1]: " Con el uso de la bicicleta no solo se obtienen beneficios en materia de movilidad y preservación del medio ambiente. Inconscientemente las personas están adoptando hábitos de vida saludable, favoreciendo tanto la salud física como mental, siendo esto un tema vital, puesto que muchos trabajos demandan que las personas se encuentren sentados gran parte de su jornada laboral, incrementando la probabilidad que los colaboradores sufran de sobrepeso, enfermedades cardiovasculares, desórdenes músculo esqueléticos (DME) y demás factores de riesgo, lo que hace que las organizaciones destinen sus recursos para la prevención de estas enfermedades, pero lo que no están viendo las empresas es la solución integral que brinda la bicicleta."

Sin embargo, los ciclistas se enfrentan a múltiples obstáculos a la hora de querer utilizarla, entre ellos el donde estacionar. Entre los principales problemas que se han identificado respectivos a este tópico están:

- Falta de información respectiva a donde encontrar estacionamientos.
- Falta de información respectiva a la disponibilidad de plazas de estacionamiento en los sistemas de estacionamientos actuales.

- Falta de sistemas de seguridad en los estacionamientos actuales, que permitan asegurar los componentes de la bicicleta y los accesorios necesarios para utilizarla (casco, luces, sillín).

- Problemas en el diseño de los estacionamientos actuales, que dificultan el uso de dispositivos de seguridad como candados tipo U.

- Desconocimiento por parte de los usuarios al momento de comprar artículos de seguridad, como candados tipo U o cadenas.

En el marco del programa de Memoria Multidisciplinaria del segundo semestre del año 2016, se propone como desafío resolver estos problemas mediante el diseño de un estacionamiento inteligente para bicicletas operado mediante celular.
En particular, se identifica la seguridad como eje central de desarrollo, puesto que es la raíz de los problemas que enfrentan los ciclistas en la actualidad. Dicho aspecto no solo involucra un diseño estructural sólido, si no que la arquitectura del sistema que permite administrar dicho estacionamiento debe estar preparado para esta responsabilidad. Se necesita hacer un análisis de los sistemas existentes en la actualidad. Cabe destacar que el análisis solo incluye sistema de estacionamientos, y no sistemas de bicicletas públicas. Esto se debe a que los sistemas de bicicletas públicas no solucionan el problema de estacionamiento para los ciclistas.

2.1.2. Sistemas de estacionamientos para bicicletas existentes en el mundo

- Jaarbeursplein , Holanda:

Fig. 2.1: Estacionamiento Jaarbeursplein

Considerado el estacionamiento para bicicletas más grande del mundo. Es capaz de albergar 12.500 bicicletas y está ubicado en la ciudad de Utrecht, en Holanda [54]. Este estacionamiento es de propiedad pública. Debido a sus dimensiones, este estacionamiento es una edificación con múltiples niveles. Los usuarios deben entrar a pie, moviendo su bicicleta a través de una plataforma, y recorrer el estacionamiento en busca de alguna plaza. Al encontrar un lugar vacío, debe estacionar su bicicleta utilizando uno de los marcos disponibles para ellos. Cabe destacar que los usuarios no utilizan sistemas de seguridad, como candados o cadenas, por lo que no requiere mayor tecnología para su operación. Como ventaja se puede mencionar su eficiencia en espacio. Su desventaja es que tiene requisitos particulares para su funcionamiento, como una infraestructura de grandes proporciones, una cantidad de usuarios considerable y particularmente una cultura social que permita utilizar un sistema que prescinda de sistemas de seguridad físicos. De todas formas, utilizan guardias para resguardar la protección de las bicicletas de los usuarios.

- Bikeep, Estados Unidos:

Fig. 2.2: Estación de BiKeep

BiKeep [55] es una empresa dedicada a entregar el servicio de implementación de estacionamientos inteligentes para bicicletas. Mediante el celular, el usuario puede utilizar el estacionamiento para abrir y cerrar, asegurando el marco de la bicicleta como se ve en la figura 2.2. Su gran ventaja es que cada plaza de estacionamiento funciona de manera independiente, por lo que no requiere un espacio específico para su implantación. Se energizan mediante conexión al sistema eléctrico convencional o mediante paneles solares. Como desventaja se puede mencionar que no cuenta con medios que permitan al usuario asegurar los componentes de la bicicleta, ni los accesorios necesarios para su utilización como casco o luces. Cabe mencionar que el servicio entregado por BiKeep no solo es el diseño e implementación de estacionamientos para bicicletas, si no que además el mantenimiento de ellos.

- Aparka, Alemania:

Fig. 2.3: Estación de Aparka

Aparka[56] ofrece los servicios de diseño, implementación y mantenimiento de estacionamientos para bicicletas. Los estacionamientos de Aparka son tipo Locker, es decir, protegen a la bicicleta íntegramente, protegiéndola tanto del alcance de otras personas como de condiciones climáticas. Su utilización, tanto para abrir como para cerrar, es mediante una tarjeta RFID. Se energizan mediante conexión al sistema eléctrico convencional. Como se puede ver en 2.3, estos estacionamientos tienen la ventaja de que protegen tanto la bicicleta como sus componentes. Como desventaja, no existe una identificación del usuario, puesto que solamente utilizar una tarjeta RFID para su funcionamiento. Esto además genera ineficiencias en el sistema, ya que el usuario debe hacer el proceso de registro y esperar que la tarjeta llegue a su domicilio. Otra desventaja es que cada bicicleta utiliza un espacio considerablemente más grande que los dos sistemas mencionados anteriormente.

- Eco-Cycle, Inglaterra:

Fig. 2.4: Estación de Eco-Cycle

Eco-Cycle[57] es una empresa dedicada al diseño, implementación y mantenimientos de estacionamientos para bicicletas. Los estacionamientos de Eco-Cycle cuentan con la particularidad de ser de gran tamaño, contando con locaciones de 200 y hasta 500 bicicletas. Estos estacionamientos pueden ser construidos como edificaciones, como se ve en la figura 2.4, o pueden ser implantados bajo tierra. Cualquiera de las dos modalidades protege a la bicicleta de manera íntegra. Eco-Cycle propone un sistema absolutamente robotizado, donde el usuario debe acercar la bicicleta a la estación, y es aparcada mediante un sistema de control automático. Para estacionar, el usuario debe instalar un sensor en el *shock* de la rueda delantera, de forma que al acercarla a la estación el sistema la reconoce y la registra. Luego, debe apretar un botón para que la bicicleta sea aparcada. Cuando el usuario desea retirarla, debe acercar una tarjeta RFID a la estación y le es entregada.Se alimentan mediante conexión al

sistema eléctrico convencional. Su ventaja es la comodidad y seguridad entregada al usuario, donde no existe un tercero manipulando su bicicleta. Como desventaja, su gran cantidad de componentes mecatrónicos obligan a incurrir en mantenciones con mayor frecuencia que un sistema que no cuenta con su nivel de automatización. El uso de una tarjeta RFID no permite al sistema identificar quien es verdaderamente la persona que retira la bicicleta.

- CycleSafe, Estados Unidos:

Fig. 2.5: Estación de CycleSafe

CycleSafe[58] es una empresa que se dedica al diseño, implementación y mantención de estacionamientos para bicicletas. Para propósitos de análisis, solo se estudiarán los estacionamientos tipo locker, como el de la imagen 2.5. El control de acceso al sistema de CycleSafe cuenta con diversas alternativas, dependiendo de las necesidades del cliente. Se puede contar con llave, tarjeta RFID, o acceso mediante aplicación más Bluetooth. En este último caso, el usuario debe seleccionar la estación y la plaza de estacionamiento en que desea aparcar, y al llegar a la estación se abre la puerta del locker mediante comunicación Bluetooth. Cuando el usuario desea retirar su bicicleta, la estación reconoce la conexión Bluetooth y le pregunta mediante su celular si desea retirarla.Se energizan me-

diante coenxión al sistema eléctrico convencional o mediante paneles solares. Como ventaja, estos estacionamientos son modulares, y se pueden instalar en cualquier cantidad mientras el espacio sea el suficiente. Además, protege la bicicleta de manera íntegra. Como desventaja, se puede destacar el espacio que utiliza cada uno de los módulos de estacionamiento.

- Don Cicleto, España:

Fig. 2.6: Estacionamiento de Don Cicleto

Don Cicleto[59] es una empresa que administra estacionamientos para bicicletas, ubicados en estacionamientos para automóviles repartidos por ciudades de España. Para asegurar la bicicleta del usuario, la empresa aparte un espacio en los estacionamientos de automóviles por medio de una reja. Dentro de estos espacios, hay rack para bicicleta colgados en el muro, como se muestra en la figura 2.6. Mediante una aplicación el usuario reserva su plaza de estacionamiento. Cuenta con planes de pago mensual o pago por uso. Como ventaja, se destaca la sencillez en su implantación y uso para el usuario. Como desventaja, se menciona que el horario de uso de estos estacionamientos está limitado por el horario de los estacionamientos donde se ubican. Además, necesita de personal

humano para su administración, de manera de asegurar que solamente usuarios autorizados entren a las dependencias del sistema, puesto que no existe control a distancia de la entrada a los estacionamientos.

2.1.3. Sistemas de estacionamientos para bicicletas existentes en Chile.

- BiciLock:

Fig. 2.7: Estación de BiciLock

BiciLock[60] es una empresa chilena dedicada a la implementación y mantenimiento de estacionamientos tipo locker para bicicleta, como la que se ve en al figrua 2.7. Estos estacionamientos son ubicados en los estacionamientos para autos de centros comerciales, supermercados o en la via pública. Funcionan mediante una tarjeta RFID, con la cual el usuario puede abrir o cerrar la plaza de estacionamiento si es que está disponible. Estos lockers de estacionamiento pueden ser instalados también para eventos, a modo de publicidad *below the line*. Se energizan mediante la utilización de paneles solares. Como ventaja, estos lockers protegen completamente la bicicleta. Como desventaja, requiere el uso de una tarjeta RFID, por lo que el sistema no puede corroborar la identidad de

quien accede al sistema. Además, el usuario necesita estar inscrito en el sistema y tener de antemano su tarjeta RFID.

- BiciChile:

Fig. 2.8: Estacionamiento BiciChile

BiciChile[61] es una solución implementada por el Banco de Chile para sus trabajadores ciclistas. BiciChile es un estacionamiento ubicado en el centro de Santiago, el cual cuenta con 150 plazas de estacionamiento, camarines para hombres y mujeres, lockers, mecánica para bicicletas, lavandería y personal de seguridad. Desde el punto de vista de sistema, este estacionamiento requiere que los usuarios estén registrados. Un guardia de seguridad verifica en la base de datos que el usuario esté inscrito, y permite el acceso al estacionamiento. Al ser un recinto cerrado, este estacionamiento no requiere de medidas de seguridad, puesto que el personal de seguridad verifica que los usuarios estén retirando una bicicleta de su propiedad. Como ventaja, del punto de vista de su arquitectura de

sistema es muy sencillo, contando con la única necesidad de un portal con información de usuarios para que lo revise la gente de seguridad. Como desventajas, se puede mencionar que esta solución no es fácilmente escalable, y que necesita de personal humano para su utilización. Además, esta solución esta disponible solamente para trabajadores del Banco de Chile.

- Estacionamientos de Metro de Santiago:

Fig. 2.9: Estacionamiento Metro de Santiago

Los estacionamientos del Metro de Santiago se encuentran ubicados en las estaciones Vespucio Norte, Escuela Militar, Salvador, Irarrazaval, La Cisterna, Cristobal Colón, Grecia, Las Mercedes, Gruta de Lourdes, Blanqueado y Pudahuel. [62] Como se ve en la figura 2.9, estos estacionamientos son tipo locker, y para su utilización se necesita registrarse con el encargado de cada estación, entregando datos personales de forma manual. El retiro de la bicicleta solo se puede hacer con la cédula de identidad. Se puede dejar el casco, y este debe ir amarrado a la bicicleta. El uso del estacionamiento no permite dejar la bicicleta durante la noche, por lo que al retirarla se cobrará una multa que tiene tarifa diaria. Como

ventaja, este sistema es muy sencillo de implantar, prescindiendo de TIC (tecnologías de información y comunicaciones). para su utilización. Como desventaja está la necesidad de uso de personal para la administración del sistema, además de la necesidad de espacio suficiente para construir los estacionamientos.

- Bicipunto SPA:

Fig. 2.10: Estacionamientos de Bicipunto

Bicipunto[63] es una empresa dedicada a la venta y arriendo de estacionamientos de corta estadía, como los que se ven en la figura 2.11. Estos estacionamientos requieren que los usuarios porten sus candados o equipos de seguridad para amarrar sus bicicletas. Los estacionamientos de la figura 2.11 son conocidos como U invertido. Estos estacionamientos prescinden de cualquier TIC para su funcionamiento. Como ventaja se menciona que no necesitan energía y su instalación es sencilla. Como desventaja, se menciona la necesidad de personal tanto para instalarlos, transportarlos y vigilarlos.

- Nación Pedal:

Fig. 2.11: Estacionamientos de Nación Pedal

Nación Pedal[64] es una tienda para bicicletas ubicada en la estación Manuel Montt del Metro de Santiago. Cuenta con cupo para estacionar 24 bicicletas, los cuales deben ser reservados mediante el pago de una cuota mensual. Este servicio necesita que el usuario cuente con un dispositivo de seguridad como cadenas o candados tipo U. Como ventaja, está la seguridad y confianza que entrega en los usuarios. Como desventaja está su baja escalabilidad y horario limitado de uso. Esta solución prescinde del uso de cualquier TIC.

- Bicimapa.cl:

  Bicimapa[65] es una página web que entrega información acerca de servicios relacionados con bicicleta existentes en Santiago. En dicha página, hay un mapa que indica la ubicación de estacionamientos, talleres, ciclovias, tiendas y lugares

bike friendly. Su principal desventaja es que no otorga información en tiempo real acerca de la disponibilidad de plazas en los puntos de estacionamiento.

2.1.4. Resumen de características

Como se puede observar de en la caracterización de los sistemas de estacionamientos expuestos en las secciones 2.1.2 y 2.1.3, existen notorias diferencias entre los sistemas presentes en el mundo y los sistemas presentes en Chile. Dichas diferencias radican en factores como demanda de uso de la bicicleta, modelo de negocios de las compañías e incluso cultura ciclista del país en cuestión.

Se puede destacar el caso de Holanda, donde existen 1,11 bicicletas por cada habitante [2] y el 27 % de los desplazamientos realizados en el país se hacen en bicicleta [3]. Estos factores son importantes a considerar al momento de diseñar un servicio, puesto que determinan cuales son las tecnologías apropiadas a utilizar.

En el caso de Jaarbeursplein en Holanda se justifica la implementación de un estacionamiento de esas dimensiones, debido a la frecuencia de uso que tiene la bicicleta en ese país.

A diferencia de este sistema, existen servicios diseñados para ser flexibles a la necesidad del cliente; Bikeep, Aparka, Eco-Cycle y CycleSafe ofrecen distintas modalidades de operación para distintos casos de uso. Estas 4 empresas utilizan TIC para operación sin personal.

En el caso de Don Cicleto, también se hace uso de TIC, pero apoyado por el uso de personas para la operación del sistema, aprovechando el personal de seguridad presente en los estacionamientos de automóviles en que se encuentran las estaciones.

De los sistemas que se encuentran en Chile, se observa que el uso de TIC es poco frecuente, i.e. solamente BiciLock presenta una solución escalable y con uso de TIC.

De todas formas el uso de tarjetas RFID es un inconveniente, debido a que se debe contar con un sistema de entrega y reposición de tarjetas. El uso de tarjetas RFID también dificulta la identificación efectiva del usuario por parte del sistema de control de acceso.

Del resto de sistemas presentes en Chile, solo BiciChile utiliza TIC para su funcionamiento; mediante un registro de usuarios se puede llevar un registro de ingresos y egresos al edificio, único rol de la tecnología. El aparcamiento en este sistema se hace de forma manual, y como el estacionamiento cuenta con personal de seguridad, no se necesitan dispositivos de seguridad para amarrar la bicicleta. Al salir,los guardias verifican que el usuario esté retirando la misma bicicleta con la que llegó. Como se menciona en 2.1.3, este sistema no es escalable, debido a la necesidad de utilización de una edificación y de personal de seguridad.

En los casos de BiciLock y BiciChile, se destaca el hecho que ambos sistemas protegen a la bicicleta en su totalidad, dejándolas fuera del alcance de personas ajenas. Además, cuentan con la posibilidad de proteger los accesorios necesarios para el uso de la bicicleta, como casco y luces.

Los estacionamientos restantes no cuentan con TIC ni sistemas de seguridad para resguardar la bicicleta del usuario, y tampoco protegen la bicicleta de manera integra ni sus componentes y accesorios.

2.2. Requisitos del sistema

2.2.1. Requerimientos no Funcionales

Como se describe en la sección 2.1.1, el eje central del desarrollo de la propuesta de solución es la seguridad. Para efectos de arquitectura de sistema y diseño de servicio del estacionamiento para bicicletas, el concepto de seguridad puede tomar distintos significados:

- Seguridad de la Información, que son todas las prácticas necesarias para mantener la confidencialidad, integridad y disponibilidad de la información en toda la arquitectura del sistema de estacionamiento para bicicleta.
- Protección de la bicicleta del usuario contra robos.

La principal preocupación que recae en el sistema de control de acceso es la verificación de que el usuario y solo el usuario pueda retirar su bicicleta. Por lo tanto, es necesario que el método de acceso al sistema permita minimizar el riesgo de suplantación de identidad.
El diseño de una red de estacionamientos para bicicletas requiere que la solución propuesta sea escalable, de forma que el crecimiento del sistema no signifique una pérdida de calidad ni una ineficiencia en costos, tanto monetarios como operacionales. Requisitos no funcionales:

- Se debe prescindir de la utilización de guardias de seguridad para resguardar la bicicleta.
- Se debe disminuir el riesgo de suplantación de identidad.
- La información esta disponible en la totalidad del tiempo.
- Se debe resguardar la integridad y la confidencialidad de la información del usuario.
- El sistema debe ser confiable para el usuario; no solo debe ser seguro estructuralmente, si no que debe ser capaz de otorgar la sensación de robustez.
- Se deben mitigar los riesgos por falla de sistema que deje vulnerable la bicicleta ante robos.

2.2.2. Requerimientos Funcionales

Las TIC son una herramienta fundamental para poder generar soluciones con potencial de escalabilidad, por lo que se define su uso como uno de los pilares en el desarrollo de la propuesta de solución. Con el uso de TIC se puede entregar un servicio integral, de forma que el usuario tenga la información suficiente para tomar decisiones en base a ella. La utilización de TIC conlleva ciertos requisitos, como la

necesidad de energizar las estaciones, sistemas de comunicación para poder conectarlas y consideraciones de seguridad de la información para poder operar el sistema de forma correcta. El valor agregado que se busca al implementar este sistema, liberar de responsabilidades tanto al usuario como al cliente que solicite la instalación de un estacionamiento, puesto que ciclista y cliente son dos entes distintos.

- Energía:
  - El uso de TIC requiere energía para su utilización.
  - La energía puede provenir de la red eléctrica convencional o de un panel solar.
  - Se debe tener métricas de uso de energía.
  - Se debe contar con un plan de mitigación en caso de falla de suministro de energía.
- Comunicación:
  - Se debe contar con conexión a internet.
  - La conexión a internet debe ser confiable.
  - Se debe mitigar los problemas referentes a fallas en el sistema, por lo que se debe pensar en un plan de contingencia en caso de que alguna estación pierda conexión.
  - Se necesita un servidor para administrar el sistema y para almacenar datos.
  - Se debe estimar el volumen de datos transmitidos.
  - El servidor debe ser capaz de enviar órdenes a cada estacionamiento, y que estas redirijan dichas órdenes a las plazas de estacionamiento, de forma de tener un único punto de llegada de información.
- Seguridad:
  - Se debe proteger la información de los usuarios.

- El sistema debe ser capaz de identificar que es el usuario quien está utilizando una plaza de estacionamiento, i.e. debe evitar la suplantación de identidad al momento de estacionar.
- La lógica de acción de los estacionamientos debe desarrollarse en pos de la seguridad de la información.
- Se debe evitar el manejo del estacionamiento a distancia, por lo que se debe verificar que el usuario esté en las cercanías del estacionamiento.
- El sistema debe tener un punto único de acceso; se deben minimizar los riesgos de ataque limitando el flujo de datos para poder ser controlados.
- La información debe ser albergada en el servidor.
- La lógica de control debe ejecutarse en el servidor.
- La comunicación entre el servidor y los estacionamientos debe estar cifrada.

- Información:
  - Se debe cumplir con los principios de seguridad de información, que son disponibilidad, confidencialidad e integridad.
  - La actualización de información debe ser en tiempo real, por lo que se debe evitar asimetría de información entre los estacionamientos y el servidor.
  - Se debe contar con canales de comunicación eficaces con los clientes.

2.2.3. Requerimientos de arquitectura

La operación del sistema se hace mediante un único punto de control. Esto quiere decir que toda la información y la lógica del sistema se ejecuta desde una sola localización con fines de poder dar seguridad.

El punto único de control es un servidor remoto, capaz de administrar el acceso a las plazas de estacionamiento (PDE). Para ello se necesita que el servidor cumpla ciertas funciones:

1. Almacenamiento de datos.

2. Lógica de control que proteja las bicicletas.

3. Comunicación remota.

2.2.4. Requerimientos de Interfaces

El uso principal del sistema de estacionamientos es poder estacionar la bicicleta en un lugar seguro. Siguiendo el principio KISS (Keep It Simple, Stupid) [4], se debe tener en cuenta que el usuario tiene en mente solo ese uso para este sistema. Por tanto, la interfaz gráfica debe ser simple, reduciendo al máximo la complejidad del servicio.

- El usuario necesita saber si el estacionamiento en su destino tiene plazas disponibles.

- El usuario debe identificarse con el sistema para tener acceso a una plaza de estacionamiento, o para retirar su bicicleta.

- Los administradores de los estacionamientos deben contar con una interfaz gráfica que les permita autorizar a ciclistas al uso de un estacionamiento.

2.3. Gestión de riesgos

2.3.1. Supuestos

- La conexión a internet tiene una alta disponibilidad.

- Se tendrá la energía necesaria para alimentar cada estación la mayoría del tiempo.

- Sin el uso de herramientas de alta potencia es muy difícil que una persona vulnere los estacionamientos para bicicleta.

- La estructura del estacionamiento es capaz de proteger los dispositivos electrónicos de las plazas de estacionamiento, tanto del clima como del vandalismo.
- El flujo de datos por unidad de tiempo en cada estación es bajo.
- El tamaño de los paquetes de datos intercambiados entre el servidor y las estaciones es pequeño.
- Los dispositivos electrónicos en las estaciones son de baja potencia, ergo, de bajo consumo eléctrico.
- Se detectarán y arreglarán en poco tiempo los desperfectos de las estaciones.

2.3.2. Dependencias

Las dependencias del sistema son las siguientes, en orden de mayor a menor importancia:

1. Energía disponible para alimentar las estaciones.
2. Conexión a internet.
3. Comunicación entre el servidor y las estaciones.
4. Funcionamiento de las estaciones, en términos de sistemas mecánicos y electrónicos.

2.3.3. Restricciones

- No se puede utilizar el sistema sin estar registrado.
- No se puede operar los estacionamientos a distancia.
- El administrador del estacionamiento es quien decide quien puede o no usar una plaza de estacionamiento.

2.3.4. Riesgos

- Cortes de energía dejan sin funcionamiento los estacionamientos para bicicleta:
  Prevenir: Se debe tener en cuenta este tipo de situaciones, de forma que un corte de energía no abra los seguros de los estacionamientos, permitiendo el acceso a terceros a la bicicleta.

- Un desperfecto en el servicio de internet puede dejar incomunicadas las estaciones:
  Prevenir: En caso que se detecte una pérdida en la comunicación, se debe informar al usuario de dicha situación, y pedirle que espere a que se restablezca la comunicación.

- Un desperfecto en el servicio de internet deja incomunicada una estación, y el usuario necesita retirar su bicicleta de manera urgente:
  Aceptar: Se asume que el servicio estará disponible la mayoría del tiempo. Una solución a este tipo de problemas acarrea problemas de seguridad, ya que se necesita almacenar información de manera local en la estación. Se deben tener planes de contingencia para dicha situación, que se asume improbable.

- El vandalismo genera desperfectos en las estaciones:
  Aceptar: A pesar de que una estructura robusta debe ser capaz de contener este tipo de fenómenos sociales, se asume que es posible que aun así se generen desperfectos. Este tipo de costos debe ser incluido en el modelo de negocios.

- Puede ocurrir un robo de información de la base de datos:
  Prevenir: Se deben tomar todas las medidas de seguridad necesarias para que este tipo de situaciones no ocurran.

- El usuario puede ver que existe una plaza disponible en un estacionamiento, y al llegar ha sido ocupada:
  Mitigar: Se debe generar una lógica que permita el usuario tomar decisiones

respectivas a la situación. Se deben generar políticas de uso, de forma que en lo posible no hayan más usuarios que estacionamientos en cierta locación. Esta tarea es parte de las opciones del cliente,y es el administrador del estacionamiento quien toma decisiones.

- El estacionamiento podría estar ubicado en un subterráneo, lo que eventualmente debilitaría o imposibilitaría el uso de internet.
  Prevenir: En el modelo de instalación del estacionamiento debe considerar que la conexión es de suma importancia. Se deben tomar medidas de forma que el receptor de internet(e.g. un router de WiFi móvil) se encuentre a nivel de tierra.

- El administrador necesita abrir una plaza de estacionamiento para retirar una bicicleta.
  Mitigar: se debe evitar que cualquier persona menos el ciclista tenga acceso al estacionamiento. Esto es, por razones de seguridad, debido principalmente a la amenaza de robo por parte del administrador. La decisión de otorgar dicha potestad recae en el cliente, quien asume el riesgo de permitir dicha acción. Se debe recomendar no utilizar esta vía, y se debe implementar la opción de forma de dar este poder al administrador. Este tipo de transacciones se deben registrar, y generar las alarmas correspondientes a personal de la compañía que presta el servicio de estacionamiento y al usuario que tenía estacionada su bicicleta.

3. DISEÑO PRELIMINAR DE LA SOLUCIÓN

3.1. Arquitectura del sistema

3.1.1. Esquema general del sistema

El sistema se puede describir a partir tres componentes fundamentales:

- Servidor: encargado de almacenar la información y controlar la lógica de acceso a las plazas de estacionamiento (PDE).
- Estacionamientos: Estaciones base (EB) con múltiples PDE.

La comunicación entre servidor y los estacionamientos es de forma remota.
Se utiliza una arquitectura *master-slave* en dos etapas. Primero el servidor como *master* con múltiples EB como *slaves*, y luego cada EB como *master* con múltiples plazas de estacionamiento como *slaves*.
De forma general, el sistema debe tomar dos decisiones:

- Prestar o no una PDE a un usuario.

  Dicha decisión se basa en una serie de criterios, con la siguiente jerarquía:

  1. Se certifica que quien intenta hacer uso del servicio es el usuario, i.e. no hay suplantación de identidad.
  2. El usuario ingresa correctamente su clave.
  3. El usuario está habilitado para utilizar el servicio. Esta decisión depende de si el usuario tiene su cuota de utilización (CDU) pagada o si está suspendido por las políticas de uso.

4. El usuario está en las cercanías del estacionamiento de bicicletas, i.e. el usuario no esta operando el sistema de forma remota.
5. Hay PDE disponibles para uso. Esta decisión depende de si hay alguna operativa y vacante.

- Liberar o no la bicicleta en una PDE.

  Dicha decisión se basa en una serie de criterios, con la siguiente jerarquía:

  1. Se certifica que quien intenta hacer uso del servicio es el usuario, i.e. no hay suplantación de identidad.
  2. El usuario ingresa correctamente su clave.
  3. El usuario tiene su bicicleta albergada en una PDE.
  4. El usuario está en las cercanías del estacionamiento de bicicletas, i.e. el usuario no esta operando el sistema de forma remota.

Dichas decisiones deben ser tomadas en *backend* a partir de los datos albergados en la base de datos.

3.1.2. Descripción de componentes

C1 Servidor:

El servidor cumple una serie de roles:

- Almacenamiento de datos.
- Interfaz de contacto con el usuario. Mediante el servidor web el usuario abre el estacionamiento para bicicletas. A través de la misma el administrador puede autorizar a algún usuario registrado para hacer uso del estacionamiento.
- Decidir si prestar o no una PDE al usuario. También debe decidir si libera o no una bicicleta en una PDE.

- Comunicar dicha decisión a la EB, para que redirija dicha orden a la PDE.

Por otro lado, para mitigar en gran medida los problemas de seguridad, es que se opta por un servicio VPS. En ese sentido, existen empresas de renombre que otorgan una serie de garantías que permiten desligarse de un desarrollo profundo de mecanismos de seguridad. A pesar de lo anterior se hacer necesario hacer una evaluación del riesgo de la información.

C2 Estacionamientos:

Los estacionamientos están compuestos por dos sub-componentes, y cumplen los siguientes roles:

- Estación base (EB):
  - Consta de un circuito integrado capaz de comunicarse a través de internet con el servidor.
  - Transfiere las órdenes desde el servidor a las PDE.
  - Comunica el estado de ocupación de las PDE al servidor.
- Plaza de estacionamiento (PDE):
  - Consta de un circuito integrado capaz de comunicarse con la EB de manera alámbrica.
  - Cuenta con dispositivos electrónicos capaces de abrir o cerrar la plaza de estacionamiento.

C3 Alimentación eléctrica. Necesaria para el funcionamiento de los estacionamientos.

C4 Conexión a internet. Necesaria para la comunicación entre servidor y EB.

3.1.3. Matriz de Requisitos Funcionales y Componentes

Los principales requisitos funcionales son la disponibilidad de acceso al servicio y la integridad del servicio:

- Disponibilidad de acceso al servicio:

  RF1 Disponibilidad de energía eléctrica.

  RF2 Disponibilidad de conexión a internet.

  RF3 Disponibilidad de comunicación entre el usuario y el sistema.

  RF4 Disponibilidad de la información.

  RF5 Disponibilidad de operación remota.

- Integridad del servicio:

  RF6 Prevención de uso indebido del servicio.

  RF7 Confidencialidad de la información.

  RF8 Integridad de la información.

La matriz de requisitos funcionales y componentes queda de la siguiente forma:

	C1	C2	C3	C4
RF1			X	
RF2			X	X
RF3	X		X	X
RF4	X		X	X
RF5	X	X	X	X
RF6	X	X	X	X
RF7	X		X	X
RF8	X	X	X	X

Tab. 3.1: Matriz de requisitos funcionales y componentes

3.1.4. Diagrama de contexto

Se identifican 5 procesos necesarios para el funcionamiento del servicio:

- Ciclista desea registrarse en el sistema.
- Ciclista desea estacionar su bicicleta.
- Ciclista desea retirar su bicicleta.
- Administrador desea autorizar a un ciclista.
- Administrador desea retirar una bicicleta de una PDE.

Se especifican los diagramas de contexto para cada proceso en particular, luego para el sistema en conjunto.

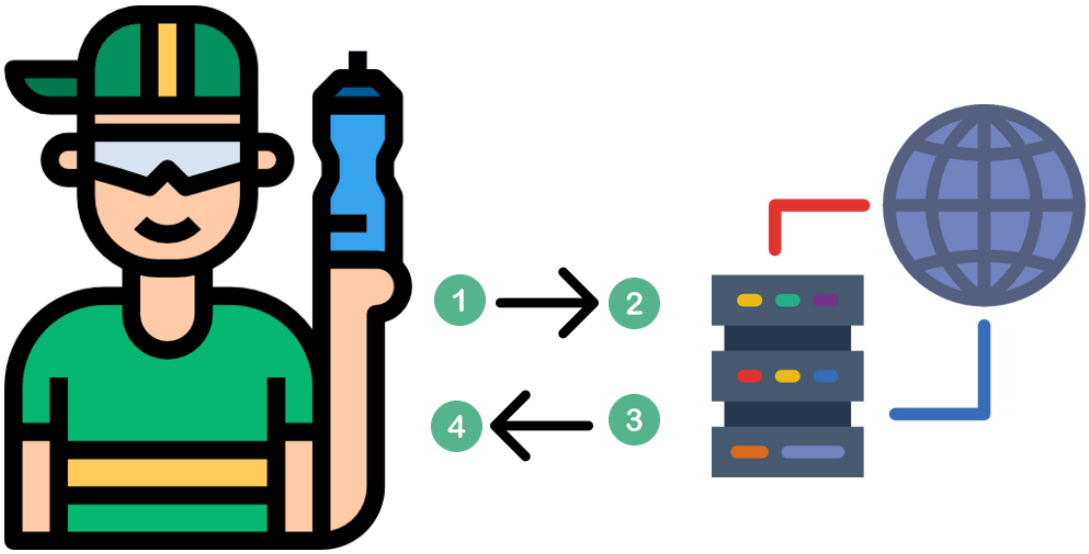

Fig. 3.1: Ciclista desea registrarse en el sistema

En la figura 3.1 se ilustra el proceso de registro en el sistema. Los puntos indicados son:

1. El usuario entra a la AM e ingresa sus datos.
    - RUT (sin puntos ni guión y con dígito verificador).
    - Nombre y Apellido.
    - Nombre de usuario.

- Número de teléfono (56 9 y 8 dígitos).
- Correo electrónico.
- Contraseña (y repetir contraseña) (8 caracteres mínimo).

2. El sistema recibe la petición de registro y verifica:
    - Los datos fueron correctamente ingresados.
    - Las contraseñas coinciden.
    - El rut, correo electrónico y el nombre de usuario no están previamente en uso.
3. El sistema responde la petición del usuario:
    - Si los datos no fueron correctamente ingresados, se le indica su error.
    - Si el correo electrónico o el nombre de usuario ya están en uso, se le indica que ya está en uso.
    - Si los datos ingresados fueron correctos:
        - Se le indica que está correctamente registrado.
        - Se le envía un correo informándole sus datos de registro.
4. El usuario recibe la respuesta de notificación:
    - Si el registro es correcto, ya puede utilizar el servicio.
    - Si no lo es, corrige sus errores y vuelve al paso 1 del proceso.

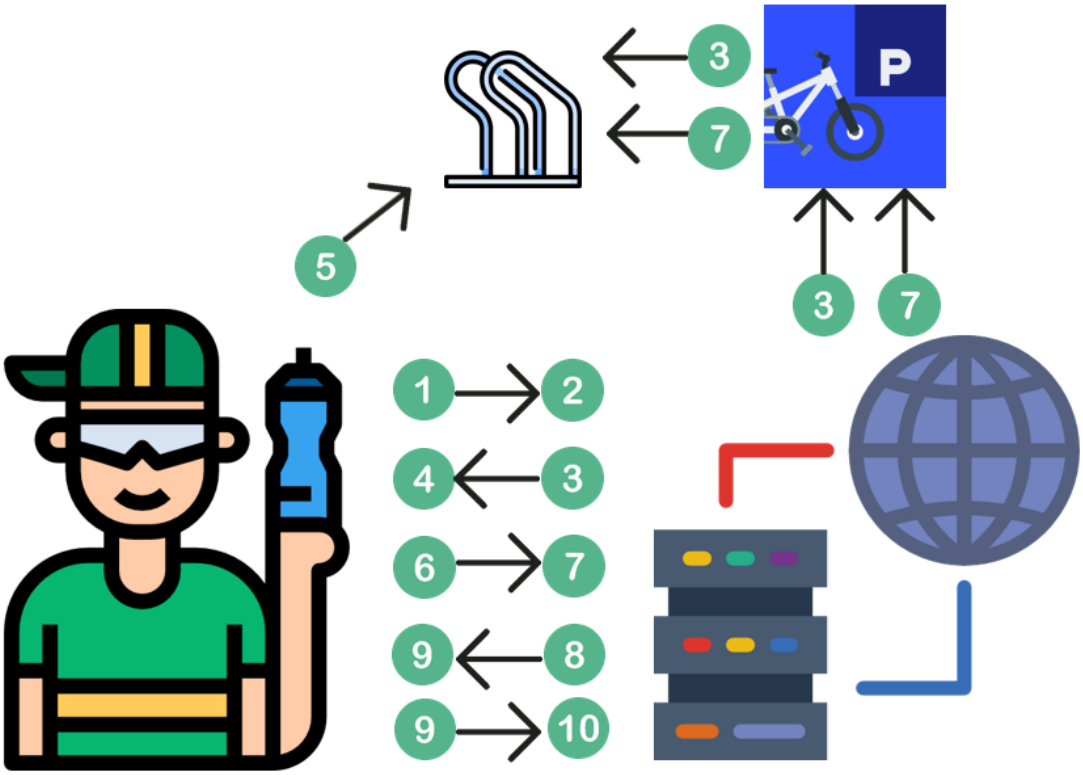

Fig. 3.2: Ciclista desea estacionar su bicicleta

En la figura 3.2 se ilustra el proceso de estacionar una bicicleta. Los puntos indicados son:

1. El ciclista se acerca a la EB:
   - Abre la AM.
   - Escoge la opción *estacionar bicicleta*.
   - Escoge una PDE vacante para estacionar.
   - Ingresa su nombre de usuario y su contraseña.
2. El sistema recibe la petición de acceso:
   - Verifica el ”match” de nombre de usuario y contraseña.
   - Verifica que el ciclista tenga permisos de acceso; puede estar restringido por falta de pago de su CDU o por políticas de uso.

- Verifica que el ciclista no tenga una PDE ya ocupada.
- Verifica que el ciclista esté en la cercanía de la EB.
- Verifica que ingresa el código 2FA correctamente.

3. El sistema responde la petición del ciclista:
   - Si la petición es aceptada, se informa al ciclista y se envía la orden de apertura de una PDE a la EB.
   - Si la petición no es aceptada, se informa al ciclista la razón.
4. El ciclista recibe la respuesta en su AM:
   - Si es aceptada, puede proceder a estacionar su bicicleta.
   - Si no es aceptada, re-ingresa sus datos y vuelve al paso 1.
5. El ciclista estaciona su bicicleta.
6. El ciclista cierra la PDE con su AM.
7. El sistema recibe la solicitud de cierre de PDE:
   - Vincula al ciclista con la PDE.
   - Registra la PDE como ”ocupada”.
   - Envía la orden de cierre de PDE a la EB.
8. El sistema informa al ciclista que la PDE ha sido cerrada:
   - Le transmite un resumen de las políticas de uso y un link con las políticas de uso completas.
   - Pide al ciclista verificar que la bicicleta quedó bien estacionada mediante un botón de ”Aceptar”.
9. El ciclista recibe la ”notificación de aceptación” y acepta el estado de ”estacionado” para su bicicleta a través de la AM.

10. El sistema recibe la aceptación del estado de estacionado.

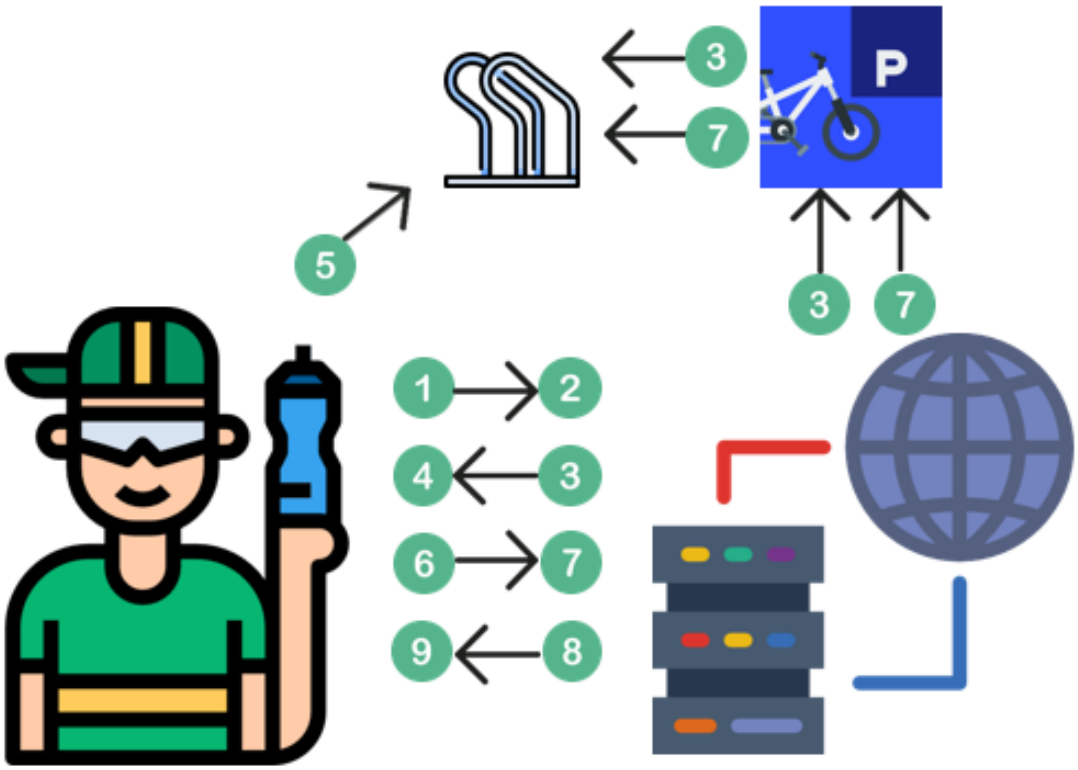

Fig. 3.3: Ciclista desea retirar su bicicleta

En la figura 3.3 se ilustra el proceso de retiro de una bicicleta. Los puntos indicados son:

1. El ciclista se acerca a la EB:
    - Abre la AM.
    - Escoge la opción *retirar bicicleta*.
    - Ingresa su nombre de usuario y su contraseña.
2. El sistema recibe la petición de acceso:
    - Verifica el ”match” de nombre de usuario y contraseña.
    - Verifica que el ciclista tenga una PDE ya ocupada.

- Verifica que el ciclista esté en la cercanía de la EB.

3. El sistema responde la petición del ciclista:
   - Si la petición es aceptada, se informa al ciclista y se envía la orden de apertura a la EB, para que abra la PDE donde está la bicicleta del ciclista.
   - Si la petición no es aceptada, se informa al ciclista la razón.
4. El ciclista recibe la respuesta en su AM:
   - Si es aceptada, puede proceder a retirar su bicicleta.
   - Si no es aceptada, re-ingresa sus datos y vuelve al paso 1.
5. El ciclista retira su bicicleta.
6. El ciclista cierra la PDE con su AM. Esto es opcional para el usuario, pero soluciona el uso indebido de las PDE. Si el usuario no cerrase la PDE con su AM, se cerrará luego de un tiempo determinado.
7. El sistema recibe la solicitud de cierre de PDE:
   - Registra la PDE como ”desocupada”.
   - Envía la orden de cierre de PDE a la EB.
8. El sistema informa al usuario que la PDE ha quedado en estado de ”desocupado”.
9. El usuario ”confirma” que retiró su bicicleta. Este paso también puede ser utilizado como orden para cierre de PDE.

Se describe a continuación el proceso de autorización de un ciclista en el estacionamiento por parte del administrador.Al momento de instalar un estacionamiento para bicicletas, se designa un administrador del mismo.

El proceso de registro de administrador de estacionamiento es iniciado por el operario que instala el estacionamiento. Se asume que el administrador ya cuenta con su cuenta activa.

El proceso de autorización de uso de estacionamiento a un nuevo ciclista es ejecutado por el administrador, y se inicia con la llegada del ciclista y su petición de autorización.

1. El administrador inicia sesión en la página web, escogiendo la opción ”sesión administrador”.

2. En la sección registrar usuario, ingresa el rut del usuario:

   - Si el rut está registrado en el sistema, se envía un correo a dicho usuario indicando que ya ha sido autorizado.
   - Si no está registrado, se informa al administrador dicho suceso.

3. Se indica al administrador la cantidad de plazas de estacionamiento instaladas y la cantidad de usuarios registrados con el fín de que no desborde el estacionamiento de usuarios accidentalmente.

Finalmente, se describe el proceso de retiro de bicicleta por parte de administrador.

1. El administrador inicia sesión en la página web, escogiendo la opción ”sesión administrador”.

2. Accede a la opción ”retirar bicicleta de PDE”, indicando una lista con las PDE ocupadas. Se debe advertir acerca de las implicancias del uso de esta opción.

3. Escoge la PDE que desea abrir, retira la bicicleta y cierra la PDE.

4. El sistema genera una alerta al dueño de la bicicleta, y al operario del sistema que esté a cargo de monitorear el comportamiento del sistema en ese minuto.

Para la sesión de administrador, se sugiere el uso de autenticación de dos factores [10], en especial si el administrador cuenta con permisos de retiro de bicicleta.

3.1.5. *Diagrama de arquitectura*

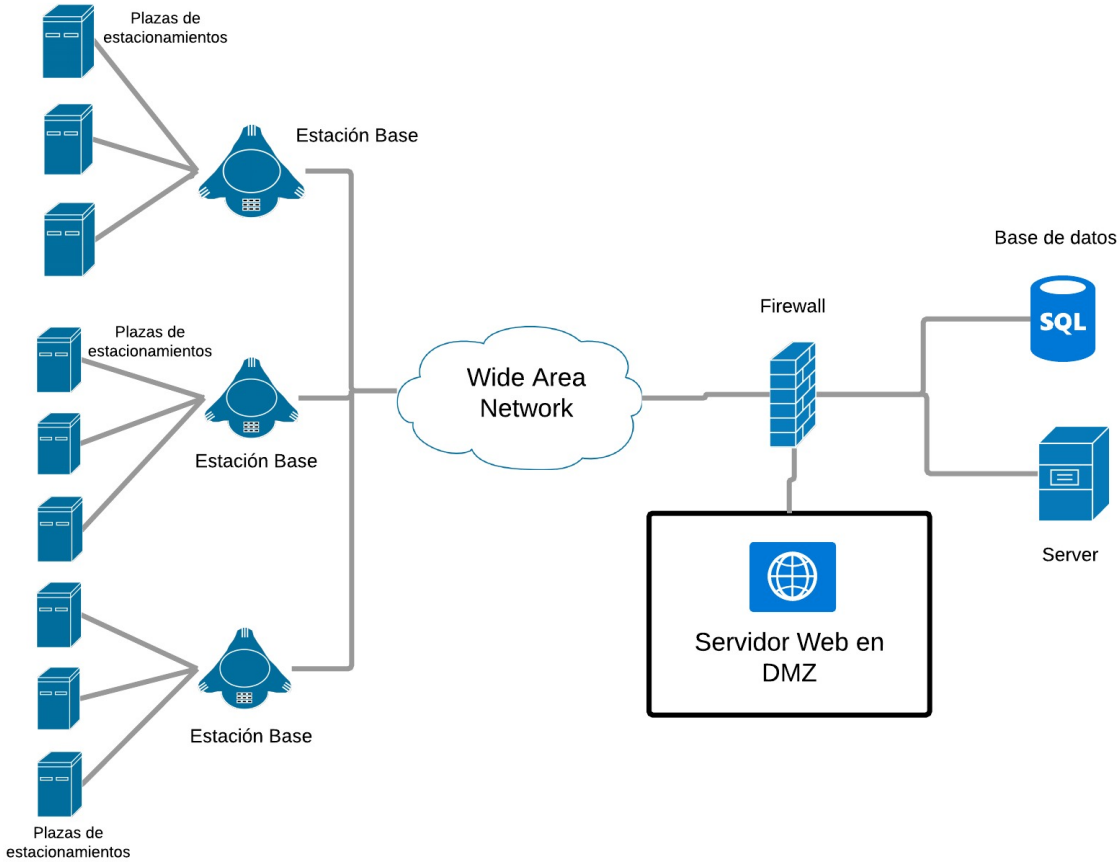

Fig. 3.4: Diagrama de arquitectura del sistema

Como se especifica en la sección 2.2.2, se diseña la arquitectura de forma de tener un punto único de acceso al servicio; el servidor WEB es la interfaz de comunicación entre el sistema y el usuario.

Como se observa en la figura 4.1, el servidor WEB está configurado para trabajar en una zona desmilitarizada o DMZ, y la base de datos y los servicios *backend* del servidor quedan protegidos detrás de un *firewall*,i.e. la arquitectura está orientada a que las órdenes hacia las EB solo provengan del servidor, y estas se pueden comunicar

con el servidor solo si es que este inició la comunicación.

Por otro lado, si algún usuario solicita su registro o acceso al servicio, la petición provendrá desde el servidor WEB, quien está autorizado por el *firewall* a entregar la información a los servicios *backend* para su posterior verificación.

3.1.6. Descripción general de módulos

Los módulos involucrados en el sistema son:

M1 Conexión eléctrica: Necesaria para la utilización del estacionamiento.

M2 Conexión a internet: Necesaria para la utilización del estacionamiento.

M3 Servicios Web: Servicios que permiten la comunicación entre usuario y sistema. Permite además la comunicación entre servidor y EB.

M4 Base de datos: Necesario para el almacenamiento de datos en el servidor.

M5 *Extract, Transformation and Load* ETL: Mecanismo mediante cual se hacen modificaciones en la base de datos en función de los requerimientos del servicio.

M6 Protocolo de comunicación entre EB y servidor: Se usará MQTT para disminuir el volumen de datos del servicio y para establecer una estructura de comunicación que permita la escalabilidad.

M7 Electrónica para sistema de anclaje: Configuraciones electrónicas que transforman las órdenes que llegan desde el servidor en acciones que abran o cierren el estacionamiento.

M8 Lógica de acción de estacionamientos: Establece la relación que existe entre una petición del usuario y la información en la base de datos.

M9 Sistema de verificación de presencia (SVP) de usuario en estacionamiento: Mecanismo mediante el cual se certifica que el usuario está en el estacionamiento

haciendo peticiones de apertura. Pensado para evitar el uso indebido del servicio.

M10 Sistema de respaldo de base de datos: Medida preventiva que establece una o más copias de la base de datos en pos de la disponibilidad e integridad de la información.

3.1.7. Matriz de requisitos funcionales y módulos

Los principales requisitos funcionales son la disponibilidad de acceso al servicio y la integridad del servicio:

- Disponibilidad de acceso al servicio:

  RF1 Disponibilidad de energía eléctrica.

  RF2 Disponibilidad de conexión a internet.

  RF3 Disponibilidad de comunicación entre el usuario y el sistema.

  RF4 Disponibilidad de la información.

  RF5 Disponibilidad de operación remota.

- Integridad del servicio:

  RF6 Prevención de uso indebido del servicio.

  RF7 Confidencialidad de la información.

  RF8 Integridad de la información.

La matriz de requisitos funcionales y módulos queda definida como:

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
RF1	X									
RF2	X	X								
RF3	X	X	X	X						
RF4		X	X	X	X					X
RF5	X	X	X	X	X	X	X	X		
RF6	X	X		X	X	X		X	X	
RF7		X	X	X						
RF8			X	X		X	X	X	X	X

Tab. 3.2: Matriz de requerimientos funcionales y módulos

3.2. Diseño de módulos del sistema

Para el diseño de los módulos, se hacen supuestos en pos de establecer variables de diseño adecuadas para la instalación del estacionamiento.

- Se utilizarán dispositivos ESP8266 [14], tanto para las PDE como para las EB.
- La comunicación entre EB y PDE es mediante I2C [5].
- Se utiliza MQTT [12] para comunicar EB y servidor.
- Se escoge, preliminarmente, trabajar con los servicios de la empresa WOM [8] como ISP.
- Cada cliente pide instalar, como mínimo, 12 estacionamientos para bicicletas.
- El componente para cierre electrónico es un *Electronic Rotary Latch* [15].
- Se esperan horas peak de uso del estacionamiento de bicicleta, en donde se concentrará el mayor consumo de energía eléctrica y paquetes de datos. Dichas horas son en la mañana, a la hora de entrar al trabajo para el usuario, y en la tarde, al momento de volver a casa.

- El edificio donde está ubicado el estacionamiento cuenta con sistema eléctrico de respaldo. De no contar con aquel sistema, se acepta el hecho de que no se podrá operar el estacionamiento. Esto es debido a que, de contar con un sistema de respaldo energético propio para cada estacionamiento, los costos de confección y mantención subirían, y dicha implementación no soluciona un problema crítico, debido a que el estacionamiento se mantendrá cerrado en caso de corte eléctrico.

3.2.1. Diseño de módulo [M1]: Conexión eléctrica

Las EB y las PDE necesitan energía eléctrica para funcionar. Ambos componentes operarán mediante dispositivos IoT, por lo que se necesita diseñar las conexiones eléctricas para alimentarlos, ya que no se pueden conectar directamente a la fuente de 220 VAC. Las PDE contarán con un cierre electrónico operado mediante los dispositivos IoT, por lo que también debe ser considerado dicho módulo.

- En general, se debe procurar instalar los estacionamientos en las cercanías de un enchufe eléctrico.

- Los dispositivos IoT son de baja potencia. Estos controlarán el cierre electrónico, el cual es un módulo aparte que debe ser escogido, y a partir de allí hacer la consideraciones técnicas para su funcionamiento.

- Se debe procurar la facilidad de instalación, sin comprometer la integridad del sistema. Es por ello que no pueden ir conectados mediante un enchufe a la conexión eléctrica convencional, es decir, se debe intervenir dicho enchufe de forma que la conexión quede aislada del manejo de terceros en pos de que no se pueda desconectar fácilmente.

- Al momento de que el cliente solicite la instalación del estacionamiento, se debe medir la distancia desde el lugar donde será instalado el estacionamiento

hasta la conexión eléctrica más cercana, de forma de escoger un largo de cable adecuado.

- La conexión eléctrica a donde debe conectarse el estacionamiento debe ser un enchufe y no, por ejemplo, una ampolleta. Esto es debido a que, al utilizar la conexión de una ampolleta, se puede des energizar el estacionamiento al momento de que se apaga la ampolleta a través de un interruptor.

- Debido a que se intervendrá una conexión eléctrica, y a que posiblemente no se podrán desenergizar dichos enchufes, se debe contar con los implementos de seguridad y las herramientas necesarias para que el operario pueda trabajar sin riesgo de electrocutarse.

- Existe el riesgo de que se corte la luz del estacionamiento, por lo que el cierre electrónico debe mantenerse cerrado en caso de dicha eventualidad.

- Se utiliza un cable de corriente convencional. Por razones de seguridad, se necesita contar con 3 cables por estacionamiento, para incluir la tierra dentro de las conexiones.

- Se utilizan 3 colores de cables para identificar cada componente del enchufe.

- Se utilizarán ESP8266, tanto para las PDE y para las EB. Según el datasheet de dicho dispositivo [18], tiene distintos consumos:

  - 20 [μA] en estado *Deep Sleep*.
  - 0.6 [mA] en estado *Sleep*.
  - Cerca de 1[mA] en estado *Active*.

  Y sus valores típicos:

Hardware	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5V ~ 3.6V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	–40°C ~ 125°C
	Storage Temperature Range	–40°C ~ 125°C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-

Fig. 3.5: Consumo de ESP8266 según fabricante.

Dichos valores pueden ser considerados despreciables para efectos de consumo. Esto es, típicamente el consumo de potencia es de:

$$3{,}6[V] \cdot 80[mA] = 288[mW] \tag{3.1}$$

por cada PDE y EB. Si se consideran 12 PDE, además de un ESP8266 en la EB, operativos las 24 horas del día, en un mes de 30 días:

$$288[mW] \cdot 13 \cdot 24\left[\frac{h}{dia}\right] \cdot 30\left[\frac{dia}{mes}\right] = 2695{,}68\left[\frac{Wh}{mes}\right] = 2{,}69568\left[\frac{kWh}{mes}\right] \tag{3.2}$$

Considerando además un precio promedio de $\$100/kWh$ en la ciudad de Santiago [20], el gasto energético de una plaza de estacionamiento es de:

$$2{,}69568\left[\frac{Wh}{mes}\right] \cdot 100\left[\frac{\$}{kWh}\right] \approx 270\left[\frac{\$}{mes}\right] \tag{3.3}$$

lo que es el gasto energético de los 13 ESP8266 de un estacionamiento por mes.

- El consumo del dispositivo para cierre electrónico según el datasheet del fabricante[16] es:

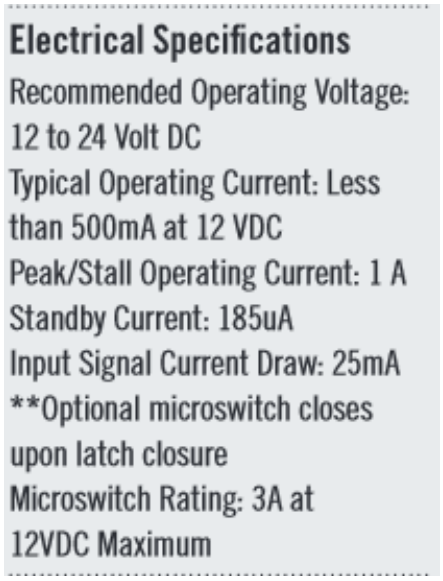

Fig. 3.6: Consumo de dispositivo de cierre electrónico según fabricante.

Se observa un consumo de $185[\mu A]$ en *standby*, y de $500[mA]$ al momento de apertura. Cabe destacar que este último puede parecer un consumo ”grande” comparado con el ESP8266, pero hay que recordar que dicho consumo es ”instantáneo” al momento que el usuario abre su estacionamiento. Por dichas razones, se puede considerar el consumo de este dispositivo como ”despreciable”. En otras palabras, si se tienen 12 PDE operando durante 720 horas mensuales a 24 [VDC], con su consumo de corriente de *standby* de $185[\mu A]$ por PDE, se tiene un consumo total de:

$$13 \cdot 720\left[\frac{h}{mes}\right] \cdot 24[V] \cdot 185[\mu A] = 41{,}5584\left[\frac{Wh}{mes}\right] = 0{,}0415584\left[\frac{kWh}{mes}\right] \tag{3.4}$$

Si se considera un precio de 100 [$/kW], se tiene un gasto en energía de:

$$0{,}0415584\left[\frac{kWh}{mes}\right] \cdot 100\left[\frac{\$}{kWh}\right] \approx 4\left[\frac{\$}{mes}\right] \tag{3.5}$$

Por otro lado, si se considera la energía necesaria para la apertura del estaciona-

miento, siendo 12 dispositivos de cierre electrónico operando durante 1 segundo para la apertura, 2 veces al día durante 30 días en un mes, y utilizando los datos de la tabla 3.6:

$$12[PDE] \cdot 2\left[\frac{s}{dia \cdot PDE}\right] \cdot 30\left[\frac{dia}{mes}\right] \cdot \frac{1}{3600}\left[\frac{h}{s}\right] 12[V] \cdot 500[mA] \tag{3.6}$$

$$= 1200\left[\frac{mWh}{mes}\right] = 0{,}0012\left[\frac{kWh}{mes}\right] \tag{3.7}$$

$$100\left[\frac{\$}{kWh}\right] \cdot 0{,}0012\left[\frac{kWh}{mes}\right] = 0{,}12\left[\frac{\$}{mes}\right] \tag{3.8}$$

- Se debe considerar además el consumo del router instalado en la EB. Al igual que en los casos anteriores, dicho consumo puede ser despreciado. Específicamente, WOM ofrece su servicio con el router modelo *Huawei B310* con las siguientes especificaciones [19]:

Item	Description		
Receiving sensitivity	LTE	Conform to 3GPP Definition	
	UMTS	• 2100 MHz: -106.7 dBm@3.84 MHz • 900MHz: -103.7 dBm@3.84 MHz	
	GSM	-104 dBm	
	WLAN	802.11b	Typ. -92 dBm@1 Mbps Typ. -85 dBm@11 Mbps
		802.11g	Typ. -88 dBm@6 Mbps Typ. -70 dBm@54 Mbps
		802.11n HT20	Typ. -88 dBm@MCS0 Typ. -68 dBm@MCS7
		802.11n HT40	Typ. -84 dBm@MCS0 Typ. -66 dBm@MCS7
Power consumption	< 12 W		
AC/DC power supply	• AC: 100 V - 240 V • DC: 12 V/1 A		
Dimensions (Maximum)	181 mm (Width) x 126 mm (High) x 36 mm (Deep)		
Weight	About 226 g (excluding the power adapter)		
Temperature	• Working temperature: 0°C to 40°C • Storage temperature: -20°C to +70°C		
Humidity	5% - 95%		

Fig. 3.7: Consumo de *Huawei B310* según fabricante.

Si se considera que está operativo durante 720 horas mensuales, su consumo

total es:

$$720\left[\frac{hora}{mes}\right] \cdot 12[W] = 8{,}64\left[\frac{kWh}{mes}\right] \tag{3.9}$$

Utilizando el mismo precio promedio que en el caso de las EB y PDE, se tiene un gasto en energía de :

$$8{,}64\left[\frac{kWh}{mes}\right] \cdot 100\left[\frac{\$}{kWh}\right] = 864\left[\frac{\$}{mes}\right] \tag{3.10}$$

- El consumo de potencia mensual de un estacionamiento de 12 PDE es de:

$$2{,}695689 + 0{,}0415584 + 0{,}0012 + 8{,}64 = 11{,}3784474\left[\frac{kWh}{mes}\right] \tag{3.11}$$

- El gasto en energía eléctrica mensual para un estacionamiento de 12 PDE es de:

$$269{,}5689 + 4{,}15584 + 0{,}12 + 864 = 1137{,}84474\left[\frac{\$}{mes}\right] \tag{3.12}$$

- Dicho aquello, se necesita entonces generar una conexión de 24 [VDC] para el dispositivo de cierre electrónico y de 3.6 [VDC] para los ESP8266. El router viene con transformador, por lo que no necesita una fuente especial.

- La alimentación debe hacerse en pos de la integridad del sistema, por lo que se recomienda que las fuentes de voltaje utilizadas tengan certificación de la Superintendencia de Electricidad y Combustible SEC[17].

3.2.2. Diseño de módulo [M2]: Conexión a internet

Cada EB debe contar con una conexión a internet. Por ello es que se necesita contratar un plan de WiFi móvil. Empresas como WOM cuentan con dicho servicio [8], que incluye el router necesario para la conexión. Se debe entonces verificar la disponibilidad del servicio en donde se vaya a implementar el servicio. En ocasiones se

requiere instalar el estacionamiento en lugares bajo tierra, como un estacionamiento para automóviles subterráneo. En pos de la escalabilidad, se debe instalar el router WiFi de forma que no esté ubicado en el subterráneo, lo que en ocasiones requerirá que se ubique el router en un lugar no subterráneo y extender un cable ethernet hasta la estación base. Como el usuario necesita conectarse a internet para operar el estacionamiento, se debe contar con un segundo router en la EB para que se conecte. Por otro lado, el Ministerio de Vivienda y Urbanismo sugiere en su "Manual Biciestacionamientos"[11] que estos se ubiquen a nivel de suelo, para que los ciclistas puedan acceder a ellos de forma cómoda, por lo que se debe procurar no instalar el estacionamiento en lugares subterráneos.

- Al momento de que el cliente pide la instalación del estacionamiento, se debe medir la distancia necesaria para colocar un router a nivel de suelo, en caso de que el cliente solicite instalarlo en un subterráneo.

- Se debe contar con un router secundario en caso de que el estacionamiento esté instalado en el subterráneo, de forma que el usuario se pueda conectar a internet.

- Las EB utilizan dispositivos IoT para comunicarse con el servidor. Se escoge el protocolo MQTT[12] para dicha comunicación, debido a que es un estándar de la industria y que permite comunicación con paquetes de bajo tamaño.

- El tamaño de los paquetes es una variable importante a considerar. MQTT utiliza hasta 28 bytes de cabeceras [13], y el tamaño del mensaje depende de la aplicación. Para el caso de la comunicación entre EB y servidor, se utilizan mensajes consistentes en *int* de hasta 128 dígitos. Incluyendo una calidad de servicio adecuada, se estiman en 300 bytes por mensaje.

- Se necesita una conexión de buena calidad.La conexión mediante 4G con WiFi móvil ofrecida por WOM [8] otorga facilidad de instalación y un adecuado ancho de banda; velocidad máxima de 20 Mbps y mínima de 32 Kbps, en caso de agotar el plan de datos [8].

- Se requiere entonces hacer los supuestos necesarios para estimar el volumen de datos y ancho de banda a contratar.

  Para un estacionamiento con 12 PDE, se espera entonces:

  - Un consumo de $12 \cdot 300[bytes]$ en la mañana de cada dia, para el proceso de apertura de las PDE.
  - Un consumo de $12 \cdot 300[bytes]$ en la tarde, para el proceso de apertura de las PDE.
  - Contando 30 días por mes, el consumo es de $30 \cdot 2 \cdot 12 \cdot 300[bytes] = 21{,}6[KBytes] = 172{,}8[Kbits]$ mensuales por concepto de operación de los estacionamientos. Es decir, el consumo mensual total del sistema podría ser suplido en 5,4 segundos con una velocidad de tráfico de 32[Kbps], la mínima ofrecida por el contrato de WOM.

    Se concluye entonces que el consumo por concepto de comunicación EB-Servidor es despreciable.
  - Se debe contar con disponibilidad de servicio en el lugar. En general, los estacionamientos estarán ubicados en zonas urbanas, donde existe dicha disponibilidad.
  - El consumo por parte de los usuarios es difícil de estimar, debido que depende de producto final, es decir, del tamaño de paquetes intercambiados entre el servidor y el celular del usuario. Este consumo puede ser despreciado, debido a que se supone que el usuario solo se conectará al estacionamiento para utilizar el estacionamiento. Existen medidas de mitigación para este evento:
    - Se puede restringir el acceso a que sea exclusivo a la página de acceso del estacionamiento.
    - Se puede contratar un plan con mayor cantidad de GB disponibles.

- Se pueden comprar bolsas de minutos en caso de que las que se utilizan se agoten.
- Al momento de que se lance la aplicación móvil, el volumen de datos bajará considerablemente al no tener que cargar una página web.

En general, el riesgo asociado a la disponibilidad de conexión es bajo. Se puede establecer entonces que con un plan de datos de 10 GB es suficiente para las intenciones de este proyecto.

3.2.3. Diseño de módulo [M3]: Servicios Web

Los servicios Web están pensados en darle ubicuidad a la solución. Permiten además hacer un acercamiento escalable a las necesidades de clientes y usuarios. En términos generales, se utilizan 5 servicios web:

3.2.3.1. Landing Page

Es la página mediante la cual se informa de los servicios de la compañía. Sirve como canal de comunicación principal con los clientes y usuarios. Las información más relevante que aparece en esta página web es:

- Se presenta la compañía ante el público. Es esta página la que refleja la personalidad de la empresa, a través del uso de técnicas de marketing como segmentación de mercado, presentación de la propuesta de valor y presentación de la solución para los *stakeholders*.
- Se utiliza además para presentar el plan de precios del servicio.
- Se utiliza para que los usuarios se registren en el servicio, y para informar a los clientes mediante el *call to action* como es que deben actuar en caso de querer instalar un estacionamiento en sus dependencias.

Si bien, el diseño de la *user experience*, *user interface* y diseño *front end* está fuera de los alcances de este documento de memoria, se debe tener en cuenta que es esta

página la primera impresión que tienen los *stakeholders* acerca de la visión, misión y objetivos de la compañía.

3.2.3.2. Página de registro

En esta página se le indica al usuario como registrarse. Se utiliza además para indicar como es que utiliza el estacionamiento, presentando las políticas y condiciones de uso, destacando aquellas que se consideren más relevantes.

3.2.3.3. Página de "pedir estacionamiento"

Si bien el usuario no inicia sesión formalmente, sí utiliza un nombre de usuario, una contraseña y una clave que indica que el usuario está en las cercanías del estacionamiento, con el objetivo de evitar el uso malicioso a distancia.

3.2.3.4. Página de "retirar bicicleta"

El proceso para estacionar y retirar la bicicleta es básicamente el mismo, con la salvedad de que al momento de estacionar la bicicleta, el usuario debe escoger que PDE utilizará, y cuando retire la bicicleta, el sistema identificará automáticamente cual es el estacionamiento que le pertenece.

3.2.3.5. Página de administrador

Es la página donde el administrador inicia sesión, para poder autorizar usuarios a utilizar el estacionamiento o para abrir un PDE en caso que lo requiera.

3.2.3.6. App server

Es el encargado de ejecutar las tareas *back-end*, es decir, todo aquello necesario para validar los datos ingresados por el usuario y contrastarlo con lo presente en la base de datos.

Los servicios Web se implementarán según lo indicado en la figura 4.1. La principal preocupación del sistema es la disponibilidad de los servidores descritos en dicha figura.

En la actualidad existen alternativas que permite operar el servicio con altos estandares de disponibilidad. En particular, *Amazon Web Services, AWS* de Amazon ofrece alternativas interesantes. Al externalizar el servicio de VPS a una compañía dedicada se mitigan muchos riesgos, especialmente asociados a seguridad de información y disponibilidad del servidor. AWS asegura un 99,99 % de disponibilidad de servicio en su *Service Level Agreement*, SLA [9]. Esto quiere decir que como máximo, el servicio estará caído durante 4,5 minutos al mes, suficiente para asegurar disponibilidad óptima.

3.2.4. Diseño de módulo [M4]: Base de datos

Se utiliza una base de datos SQL debido a que se conoce la información que se necesita recolectar. Se utiliza un servidor dedicado a estos fines, de forma de poder usar una base de datos duplicada por razones de seguridad de la información.

Se utiliza un motor de base de datos MySQL, en el cual se almacena información relativa a entidades y transacciones.

3.2.4.1. De las entidades del sistema

- EB:

  Se necesita guardar información relativa a que PDE se le asocian y cual es su estado de ocupación, es decir, si están disponibles para su utilización.

- PDE:

  Se necesita guardar información relativa a su estado de ocupación, la EB a la que está relacionada y el usuario que esté ocupando la PDE.

- Usuario:

Se necesita conocer información personal y de contacto del usuario, como nombre, apellido, y correo electrónico.Además, se necesita conocer si es que el usuario esta utilizando una PDE, y cual es dicha PDE.

- Administrador:

  Al igual que en el caso de usuarios, se necesita información personal y de contacto. A diferencia del caso de los usuarios, el administrador no usará una PDE, pero se debe conocer cual es la EB a la cual el administrador puede agregar usuarios, y si es que puede abrir o no PDE.

- EB-Llaves privadas:

  Se debe tener una tabla para las llaves privadas de inicio de comunicación y autenticación. La utilización de estos parámetros se explica en el diseño del módulo [M8] Lógica de accionamiento.

- EB-Llaves de instrucción (LDI):

  Se debe tener una tabla para las llaves de instrucción para las EB. La utilización de estos parámetros se explica en el diseño del módulo [M8] Lógica de accionamiento.

- PDE-Llave de instrucción (LDI):

  Se debe tener una tabla para las llaves de instrucción para las PDE. La utilización de estos parámetros se explica en el diseño del módulo [M8] Lógica de accionamiento.

3.2.4.2. Tabla para EB

Campo	ID_EB	ID_ADMINISTRADOR	2FA
Tipo	INT(PK)	INT(FK)	INT

Tab. 3.3: Tabla para EB

De la tabla 3.3, se observa que para las EB solo se necesita saber quién es su administrador. La información relativa a qué PDE están asociadas a esta EB está contenida en la tabla PDE. El campo ”2FA” viene del acrónimo *2 factor authentication* [22], el cual es un campo que se utiliza para verificar si el usuario está en las cercanías de la EB. Se explica su uso en el diseño de M9.

3.2.4.3. Tabla para PDE

Campo	ID_PDE	NUMERO_PDE	OCUPACION	ID_EB	ID_USUARIO
Tipo	INT(PK)	INT	BOOLEAN	INT(FK)	INT(FK)

Tab. 3.4: Tabla para PDE

En la tabla 3.4, EB representa la EB a la que esta PDE está asociada. El campo ”OCUPACION” indica si dicha PDE está o no ocupada. La tabla PDE solo contiene su estado de ocupación y la EB a qué está asociada.

3.2.4.4. Tabla para Usuario

Campo	RUT	NOMBRE	APELLIDO	NICKNAME	CORREO	TELEFONO	PASSWORD
Tipo	INT(PK)	CHAR	CHAR	CHAR	CHAR	CHAR	CHAR

Tab. 3.5: Tabla para Usuario

En la tabla 3.5, el rut del usuario es su PK.

3.2.4.5. Tabla para Administrador

Campo	RUT	NOMBRE	APELLIDO	CORREO	NICKNAME	PASSWORD	PDA
Tipo	INT(PK)	CHAR	CHAR	CHAR	CHAR	CHAR	BOOLEAN

Tab. 3.6: Tabla para Administrador

En la tabla 3.6, EB representa la EB que el administrador está habilitado para añadir usuarios, y PDA indica los ”Permisos De Administración” del administrador, es decir, si está habilitado o no para abrir PDE de forma arbitraria.

3.2.4.6. Tabla EB-PK

Campo	ID_EB	MAC	TOPICO_1	B_0	N_0	MU_0	P_0	I_0	J_0	B_1	N_1	MU_1	P_1	I_1	J_1
Tipo	INT	CHAR	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT

Tab. 3.7: Tabla EB-PK

Las PK con subfijo 0 son las PK para inicio de comunicación, mientras que las PK con subfijo 1 son las PK para controlar la EB. El campo TOPICO_1 es el nombre del tópico donde la EB recibe órdenes.

3.2.4.7. Tabla EB-LDI

Campo	ID_EB	LDI_0	LDI_1	LDI_2	LDI_3	LDI_4	LDI_5	LDI_6
Tipo	INT	INT	INT	INT	INT	INT	INT	INT

Tab. 3.8: Tabla EB-LDI

La LDI_0 es la LDI de identidad. El resto son todas de acción.

3.2.4.8. Tabla PDE-LDI

Campo	ID_PDE	LDI_0	LDI_1	LDI_2	LDI_3	LDI_4	LDI_5	LDI_6
Tipo	INT	INT	INT	INT	INT	INT	INT	INT

Tab. 3.9: Tabla PDE-LDI

La LDI_0 es la LDI de identidad. El resto son todas de acción.

3.2.4.9. De las transacciones

- Usuario-sistema:

  Entre las principales transacciones, se cuenta con petición de PDE y retiro de bicicleta.

- Administrador-sistema:

  Entre las principales transacciones, se cuenta el registro de usuario y apertura de PDE.

- Servidor-EB: Las transacciones llevadas a cabo entre estas dos entidades, son el intercambio de información relativo a la ocupación de PDE, y la orden de apertura para que el usuario estacione o saque su bicicleta.

- Registro de apertura de PDE por administrador:

  Se deben registrar estas transacciones para tener futuras referencias. Es importante para tener información clara de cuando se ejecutan estas transacciones.

Cabe mencionar que la base de datos alberga toda la información necesaria para todas las transacciones, por lo que no se necesita conocer el estado de las entidades para tomar decisiones. Por ejemplo, la EB no debe ”leer” el estado de las PDE e informar en la base de datos si es que hay alguna desocupada, puesto que en la base de datos, en la tabla ”PDE” se encuentra disponible información de su estado de ocupación.

Para el diseño de las tablas, se utiliza el principio ACID[21] para mantener la atomicidad, consistencia, integridad y durabilidad de los datos en la base de datos.

Las estructuras de las tablas para cada entidad quedan conformadas de la siguiente forma, donde PK son las *primary key* y FK son *foreign key* [21]:

3.2.4.10. Tabla para transacción Usuario-PDE

Campo	ID_PDE	RUT_USUARIO	NUMERO_PDE	FECHA	OCUPAR_DESOCUPAR
Tipo	INT	INT	INT	TIME_STAMP	BOOLEAN

Tab. 3.10: Tabla para transacción Usuario-PDE

En la tabla 3.10 se observa como es que se hacen los registros de evento en cada PDE. El campo OCUPAR/DESOCUPAR indica si la acción registrada es una ocupación o desocupación de una PDE, de forma que la transacción con fecha más reciente indique si el usuario está o no ocupando una PDE. 0 es que la transacción fue de retiro de bicicleta y 1 indica que la transacción fue de ocupación de una PDE.

3.2.4.11. Tabla para transacción Administrador-EB

Campo	ID_EB	RUT_ADMINISTRADOR	RUT_USUARIO	FECHA
Tipo	INT	INT	INT	TIME_STAMP

Tab. 3.11: Tabla para transacción Administrador-EB

En la tabla 3.11, se indican los campos necesarios para que el administrador pueda registrar a un usuario en una EB. Se utiliza una tabla dedicada debido a que un usuario puede estar asociado a múltiples EB y un administrador puede estar asociado a múltiples EB.

3.2.4.12. Tabla para registro de apertura PDE por Administrador

Campo	ID_EB	ID_PDE	NUMERO_PDE	ID_ADMINISTRADOR	ID_USUARIO	FECHA
Tipo	INT	INT	INT	INT	INT	TIME_STAMP

Tab. 3.12: Tabla para registro de apertura PDE por administrador

La tabla 3.12 muestra la estructura de la tabla que se utiliza para registrar la apertura de una PDE por parte de un administrador.

3.2.4.13. Modelo relacional

El modelo relacional muestra la dependencia de cada tabla con los campos de las demás tablas.

Para el caso de este diseño, sólo se incluyen las tablas para entidades, pues las tablas para transacción no tienen interdependencias con el resto de las tablas.

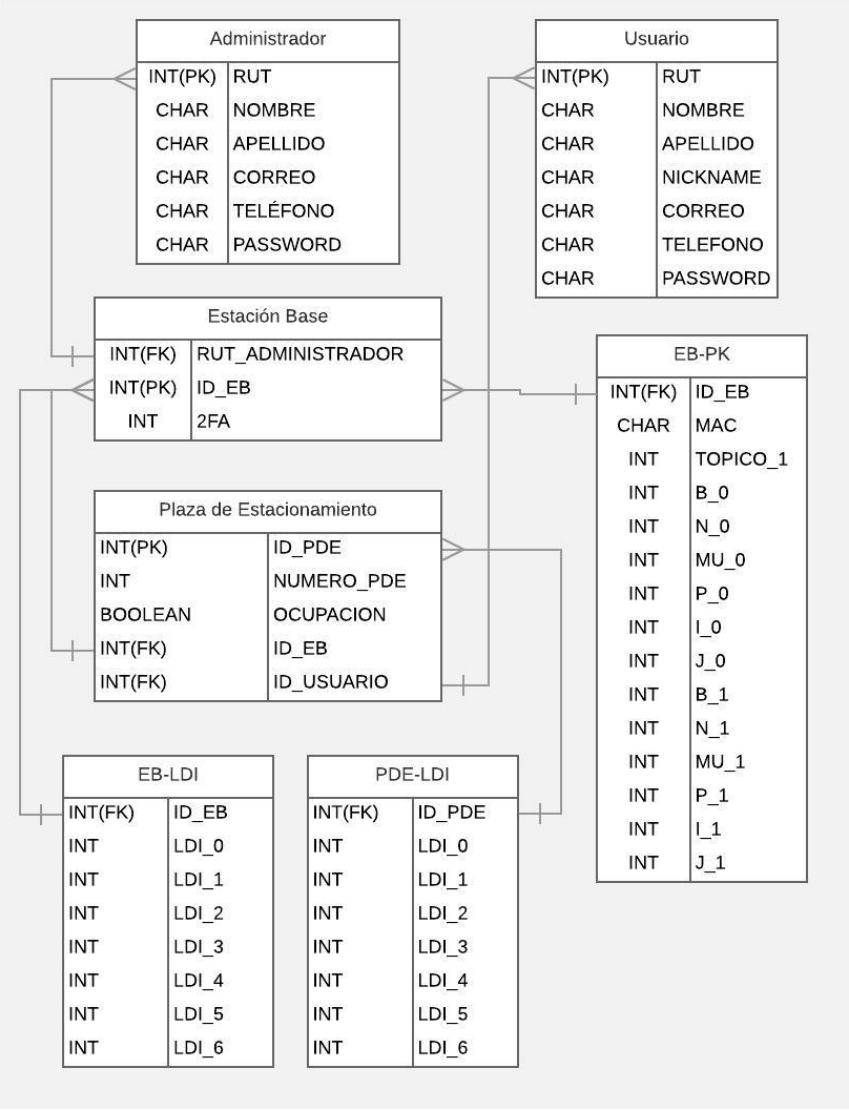

Fig. 3.8: Modelo relacional para entidades del sistema.

3.2.5. Diseño de módulo [M5]: Extract, Transformation and Load ETL

Mecanismo mediante cual se hacen modificaciones en la base de datos en función de los requerimientos del servicio. Como se menciona en el diseño de M3, se utiliza un app server. Este el encargado de albergar los servicios que controlan el sistema. Sus tareas son las descritas en 3.1.4 correspondiente a los diagramas de contexto. Se especifica la ejecución de la búsqueda en la base de datos mediante pseudocódigo para que sea exportable tanto a SQL como a cualquier lenguaje que se escoja *a posteriori* :

- Registrar al usuario en el sistema.
- Permitir al usuario estacionar su bicicleta.
- Permitir al usuario retirar su bicicleta.
- Permitir al administrador registrar al usuario.
- Permitir al administrador retirar bicicleta de una PDE.

Las sub tareas necesarias para cada una de estas tareas son:

3.2.5.1. Registrar al usuario en el sistema.

- Leer información ingresada por el usuario en el servidor web.

```
rut_usuario=$rut_ingresado
nombre_usuario=$nombre_ingresado
apellido_usuario=$apellido_ingreado
nickname_usuario=$nickame_ingresado
telefono_usuario=$telefono_ingresado
correo_usuario=$correo_ingresado
contrasena_usuario=$contrasena_ingresada
rep_contrasena_usuario=$rep_contrasena_ingresada
```

- Compararla con la información presente en la base de datos.

```
IF rut_usuario EXIST IN tabla_usuario
THEN rut_existe=1
IF nickname_usuario EXIST IN tabla_usuario
THEN nickname_existe=1
IF telefono_usuario EXIST IN tabla_usuario
THEN telefono_existe=1
IF correo_usuario EXIST IN tabla_usuario
THEN correo_existe=1
```

- Verificar ingreso correcto de los datos.

```
IF (rut_existe
    OR nickname_existe
    OR telefono_existe
    OR correo_existe)
THEN usuario_existe = 1

registro_correcto = 0
IF (!usuario_existe)
THEN
IF (!is_number(rut_usuario))
    THEN rut_mal_ingresado = 1
ELSE IF (length(rut_usuario)>9 OR length(rut_usuario)<8)
    THEN rut_mal_ingresado = 1
ELSE IF (!is_number(telefono_usuario)
            OR length(telefono_usuario)<11)
    THEN telefono_mal_ingresado = 1
ELSE IF (!is_email(correo_usuario))
    THEN correo_no_cumple_formato = 1
ELSE IF (!is_string(nombre_usuario))
    THEN nombre_no_cumple_formato = 1
ELSE IF (!is_string(apellido_usuario))
    THEN apellido_no_cumple_formato = 1
ELSE IF (length(contrasena_usuario)<8)
    THEN contrasena_muy_corta = 1
ELSE IF (contrasena_usuario!=rep_contrasena_usuario)
    THEN contrasenas_no_coinciden = 1
```

```
ELSE registro_correcto = 1
```

- Ingresar información en base de datos.

```
IF registro_correcto
    THEN INSERT INTO tabla_usuario (RUT,NOMBRE,APELLIDO,NICKNAME,
        CORREO,TELEFONO,PASSWORD) VALUES (rut_usuario,nombre_usuario,
        apellido_usuario, nickname_usuario,correo_usuario,
        telefono_usuario,contrasena_usuario);
```

- Responder la petición del usuario.

```
IF registro_correcto
    THEN PRINT("Registro exitoso")
ELSE IF rut_existe
    THEN PRINT ("RUT ya esta registrado")
ELSE IF nickname_existe
    THEN PRINT ("Nickname ya esta en uso")
ELSE IF telefono_existe
    THEN PRINT ("Telefono ya esta en uso")
ELSE IF correo_existe
    THEN PRINT ("Correo ya esta en uso")
ELSE IF nickanme_existe
    THEN PRINT ("Nickname ya esta en uso")
ELSE IF rut_mal_ingresado
    THEN PRINT ("El rut ingresado es incorrecto")
ELSE IF telefono_mal_ingresado
    THEN PRINT ("El numero de telefono es incorrecto")
ELSE IF correo_no_cumple_formato
    THEN PRINT ("El correo ingresado es incorrecto")
ELSE IF nombre_no_cumple_formato
    THEN PRINT ("El nombre ingresado es incorrecto")
ELSE IF apellido_no_cumple_formato
    THEN PRINT ("El apellido ingresado es incorrecto")
ELSE IF contrasena_muy_corta
    THEN PRINT ("La contrasena es muy corta")
ELSE IF contrasenas_no_coinciden
    THEN PRINT ("Las contrasenas no coinciden")
```

3.2.5.2. *Permitir al usuario pedir un estacionamiento*

Se utiliza el parámetro 2FA para el sistema verificador de presencia del usuario en el estacionamiento (SVP). SVP está detallado en el diseño del módulo M9.

Se supone correcto el ingreso de la PDE seleccionada por el usuario, puesto que en la AM habrá una lista con las PDE disponibles para uso. La interfaz se explica en la sección 3.3.

- Leer información ingresada por el usuario en el servidor web para 2FA y login de usuario.

```
2FA_ingresado=2FA_ingresada_web
nickname_ingresado=nickame_ingresado_web
contrasena_ingresada=contrasena_ingresada_web
```

- Compararla con la información presente en la base de datos para comprobación de 2FA y login de usuario.

```
select_eb = SELECT id_eb FROM tabla_eb WHERE 2fa=2FA_ingresado;

select_rut = SELECT rut FROM tabla_usuario WHERE nickname =
    nickname_ingresado;

select_usuario_autorizado = SELECT 1 FROM tabla_administrador_eb WHERE
     id_usuario = select_rut AND id_eb = select_eb;

select_password = SELECT password FROM tabla_usuario WHERE id_usuario
    = select_rut
```

- Responder petición de login a usuario.

```
IF select_eb == NULL
    THEN PRINT ("2FA incorrecto")
IF select_rut == NULL
    THEN PRINT ("Nickname incorrecto")
IF select_usuario_autorizado == NULL
    THEN PRINT ("No autorizado para usar estacionamiento")
```

```
IF select_password != contrasena_ingresada
    THEN PRINT ("Contrasena incorrecta")
ELSE PRINT ("Proceda a seleccionar una PDE")
```

- Leer información ingresada por el usuario en el servidor web para petición de PDE.

```
pde_seleccionada=pde_ingresada_web
id_pde_seleccionada = SELECT id_pde FROM tabla_pde WHERE numero_pde =
    pde_seleccionada AND id_eb = select_eb
```

- Responder petición de usuario. La lógica de accionamiento se presenta en el diseño del módulo M8.

```
usuario_ya_tiene_pde = SELECT numero_pde FROM tabla_pde WHERE
    id_ususario = select_rut
IF usuario_ya_tiene_pde != NULL
    THEN PRINT ("Ya tiene un estacionamiento pedido")
ELSE
    EXECUTE (abrir_estacionamiento).
    EXECUTE (cambiar_2FA)
    PRINT ("PDE seleccionada lista para su uso.")
```

Donde ”abrir_estacionamiento” y ”cambiar_2FA” son ambos módulos de la lógica de accionamiento, es decir, se ejecutan pasos lógicos en back-end para ejecutar ambas acciones de forma consistente.

- Actualizar la tabla de PDE.

```
UPDATE tabla_pde SET ocupacion = 1, id_usuario = select_rut WHERE
    id_pde=id_pde_seleccionada;
```

- Insertar transacción en tabla Transacción Usuario-PDE.

```
INSERT INTO tabla_usuario_pde (ID_PDE, RUT_USUARIO, NUMERO_PDE, FECHA,
     OCUPAR_DESOCUPAR) VALUES (id_pde_seleccionada, select_rut,
    pde_seleccionada, date.now, 1) ;
```

3.2.5.3. *Permitir al usuario retirar su bicicleta*

- Leer información ingresada por el usuario en el servidor web para 2FA y login de usuario.

```
2FA_ingresado=2FA_ingresada_web
nickname_ingresado=nickame_ingresado_web
contrasena_ingresada=contrasena_ingresada_web
```

- Compararla con la información presente en la base de datos para comprobación de 2FA y login de usuario.

```
select_eb = SELECT id_eb FROM tabla_eb WHERE 2fa=2FA_ingresado;

select_rut = SELECT rut FROM tabla_usuario WHERE nickname =
    nickname_ingresado;

select_usuario_autorizado = SELECT 1 FROM tabla_administrador_eb WHERE
     id_usuario = select_rut AND id_eb = select_eb;

select_password = SELECT password FROM tabla_usuario WHERE id_usuario
    = select_rut

usuario_no_tiene_pde = SELECT 1 FROM tabla_pde WHERE id_usuario=
    select_rut
```

- Responder petición de retiro de bicicleta. La lógica de acción de los estacionamientos se muestra en el diseño de M8.

```
IF usuario_no_tiene_pde == NULL
    THEN PRINT("No tiene una PDE pedida.")
IF select_eb == NULL
    THEN PRINT ("2FA incorrecto")
ELSE IF select_rut == NULL
    THEN PRINT ("Nickname incorrecto")
ELSE IF select_usuario_autorizado == NULL
    THEN PRINT ("No autorizado para usar estacionamiento")
ELSE IF select_password != contrasena_ingresada
```

```
    THEN PRINT ("Contrasena incorrecta")
ELSE
    PRINT ("Retire su bicicleta.")
    EXECUTE(abrir_estacionamiento)
    EXECUTE(cambiar_2FA)
```

Donde ”abrir_estacionamiento” y ”cambiar_2FA” son módulos de la lógica de accionamiento.

- Actualizar la tabla de PDE.

```
UPDATE tabla_pde SET ocupacion = 0, id_usuario = NULL WHERE id_pde=
    id_pde_seleccionada;
```

- Insertar transacción en tabla Transacción Usuario-PDE.

```
INSERT INTO tabla_usuario_pde (ID_PDE, RUT_USUARIO, NUMERO_PDE, FECHA,
     OCUPAR_DESOCUPAR) VALUES (id_pde_seleccionada, select_rut,
    pde_seleccionada, date.now, 0) ;
```

3.2.5.4. Permitir al administrador registrar al usuario

Esto se ejecuta por el operador que estará instalando el estacionamiento mediante una interfaz diseñada para eso.

- Leer información ingresada por el administrador para login.

```
nickname_ingresado=nickame_ingresado_web
contrasena_ingresada=contrasena_ingresada_web
```

- Compararla con la información presente en la base de datos.

```
rut_administrador = SELECT rut FROM tabla_usuario WHERE nickname =
    nickname_ingresado;

select_password = SELECT password FROM tabla_usuario WHERE id_usuario
    = rut_administrador
```

- Responder petición de login de administrador.

```
ELSE IF rut_administrador == NULL
    THEN PRINT ("Nickname incorrecto")
ELSE IF select_password != contrasena_ingresada
    THEN PRINT ("Contrasena incorrecta")
ELSE PRINT ("Login correcto.")
```

- Leer rut ingresado por administrador en servidor web.

```
rut_ingresado = rut_ingresado_web
```

- Verificar que el rut sea correcto, exista en la tabla de usuario y luego insertar el tabla Administrador-EB.

```
rut_usuario = SELECT rut FROM tabla_usuario WHERE rut = rut_ingresado
IF (!is_number(rut_ingresado))
    THEN PRINT ("RUT mal ingresado")
ELSE IF (length(rut_ingresado)>9 OR length(rut_ingresado)<8)
    THEN PRINT ("RUT mal ingresado")
ELSE IF (rut_usuario == NULL)
    THEN PRINT ("RUT no existe")
ELSE {
id_eb = SELECT id_eb FROM tabla_eb WHERE id_administrador =
    rut_administrador
INSERT INTO tabla_administrador_eb (ID_EB, RUT_ADMINISTRADOR,
    RUT_USUARIO, FECHA) VALUES (id_eb, rut_administrador,rut_usuario,
    date.now) ;
}
```

3.2.5.5. Permitir al administrador retirar bicicleta de una PDE

- Leer información ingresada por el administrador para login.

```
nickname_ingresado=nickame_ingresado_web
contrasena_ingresada=contrasena_ingresada_web
```

- Compararla con la información presente en la base de datos.

```
rut_administrador = SELECT rut FROM tabla_usuario WHERE nickname =
    nickname_ingresado;

select_password = SELECT password FROM tabla_usuario WHERE id_usuario
    = rut_administrador;
```

- Responder petición de login de administrador.

```
ELSE IF rut_administrador == NULL
    THEN PRINT ("Nickname incorrecto")
ELSE IF select_password != contrasena_ingresada
    THEN PRINT ("Contrasena incorrecta")
ELSE PRINT ("Login correcto.")
```

- Leer PDE ingresada por administrador en servidor web.

```
pde_ingresada = pde_ingresada_web
```

- Verificar que el administrador tenga permisos de apertura y luego insertar el tabla Administrador-EB. La lógica de accionamiento de las PDE se muestra en el diseño de M8.

```
administrador_tiene_permiso = SELECT 1 FROM tabla_administrador WHERE
    id_administrador = rut_administrador AND pda = 1;

IF (administrador_tiene_permiso == NULL)
    THEN PRINT ("No tiene permiso para apertura de PDE.")
EXECUTE(abrir_estacionamiento)
```

Donde ”abrir_estacionamiento” corresponde a las acciones necesarias para abrir el estacionamiento, i.e. registrar apertura, registrar el usuario que la inicio y enviar la señal electrónica para que se abra el estacionamiento físicamente.

- Registrar apertura de PDE en tabla de registro de apertura de PDE.

```
id_EB = SELECT id_eb FROM tabla_administrador WHERE rut =
    rut_administrador;
```

```
2
3 id_PDE = SELECT id_pde FROM tabla_pde WHERE id_eb = id_EB;
4
5 numero_PDE = pde_ingresada;
6
7 id_administrador = rut_administrador;
8
9 id_usuario = SELECT id_usuario FROM tabla_pde WHERE id_pde = id_PDE
      AND id_eb = id_EB
10
11 INSERT INTO tabla_apertura_pde_administrador (ID_EB, ID_PDE,
      NUMERO_PDE, ID_ADMINISTRADOR, ID_USUARIO, FECHA) VALUES (id_EB,
      id_PDE, numero_PDE, id_administrador, id_usuario, date.now)
```

3.2.6. *Diseño de módulo [M6]: Protocolo de comunicación entre EB y servidor*

Se usará MQTT[13] para disminuir el volumen de datos del servicio y para establecer una estructura de comunicación que permita la escalabilidad.

Su configuración es sencilla. Mediante el uso de ”tópicos”, los dispositivos pueden suscribirse a canales desde donde leen información.

Para la comunicación del sistema, cada EB se suscribe a tópicos según se describe en el diseño del módulo [M8] Lógica de acción de estacionamientos.

Además, se utiliza el *broker* público CloudMQTT[24].

Se utiliza TLS/SSL [25] para agregar seguridad en la capa transporte.

Cuando el sistema se encuentre en producción, se levantará un *broker* MQTT en el servidor, con las mismas configuraciones que el broker público, de forma de privatizar la comunicación.

3.2.7. *Diseño de módulo [M7]: Electrónica para sistema de anclaje*

Configuraciones electrónicas que transforman las órdenes que llegan desde el servidor en acciones que abran o cierren el estacionamiento.

3.2.7.1. Comunicación I2C

Cada EB esta conectada mediante I2C a cada PDE. La ventaja principal de este protocolo es que permite la conexión de múltiples dispositivos en un esquema *master-slave* a traves de una única línea de control.

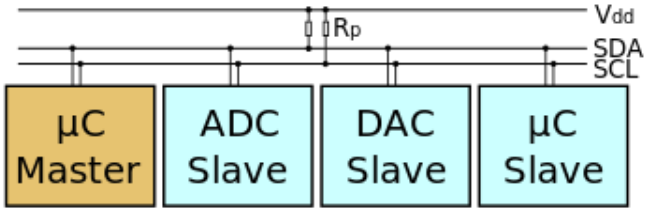

Fig. 3.9: Esquema de conexión I2C

3.2.7.2. Dispositivo para cierre electromecánico

Para el sistema de cierre, se escoge utilizar un *Electronic Rotary Latch* [15].

Fig. 3.10: Electronic Rotary Latch

Dicho dispositivo funciona con una línea de control a modo de *Relay*, de forma que se desbloquea al recibir una señal en dicha línea y se cierra de forma mecánica. Esto quiere decir que el sistema se cierra automáticamente cuando el usuario cierra el estacionamiento, por lo que no necesita una señal de cierre.

3.2.7.3. *IoTD para comunicación EB-Servidor*

Para la comunicación EB-servidor se utilizan NodeMCU [26], el cual es un IoTD basado en ESP8266 [27], debido a su bajo costo y posibilidad de conexión a internet mediante WiFi.

Para las PDE se utilizan ESP1 [28] por su bajo costo y tamaño pequeño.

Con propósitos de prototipo, se utilizan NodeMCU tanto para EB como para PDE, y un Relay de 3[Volts] [29].

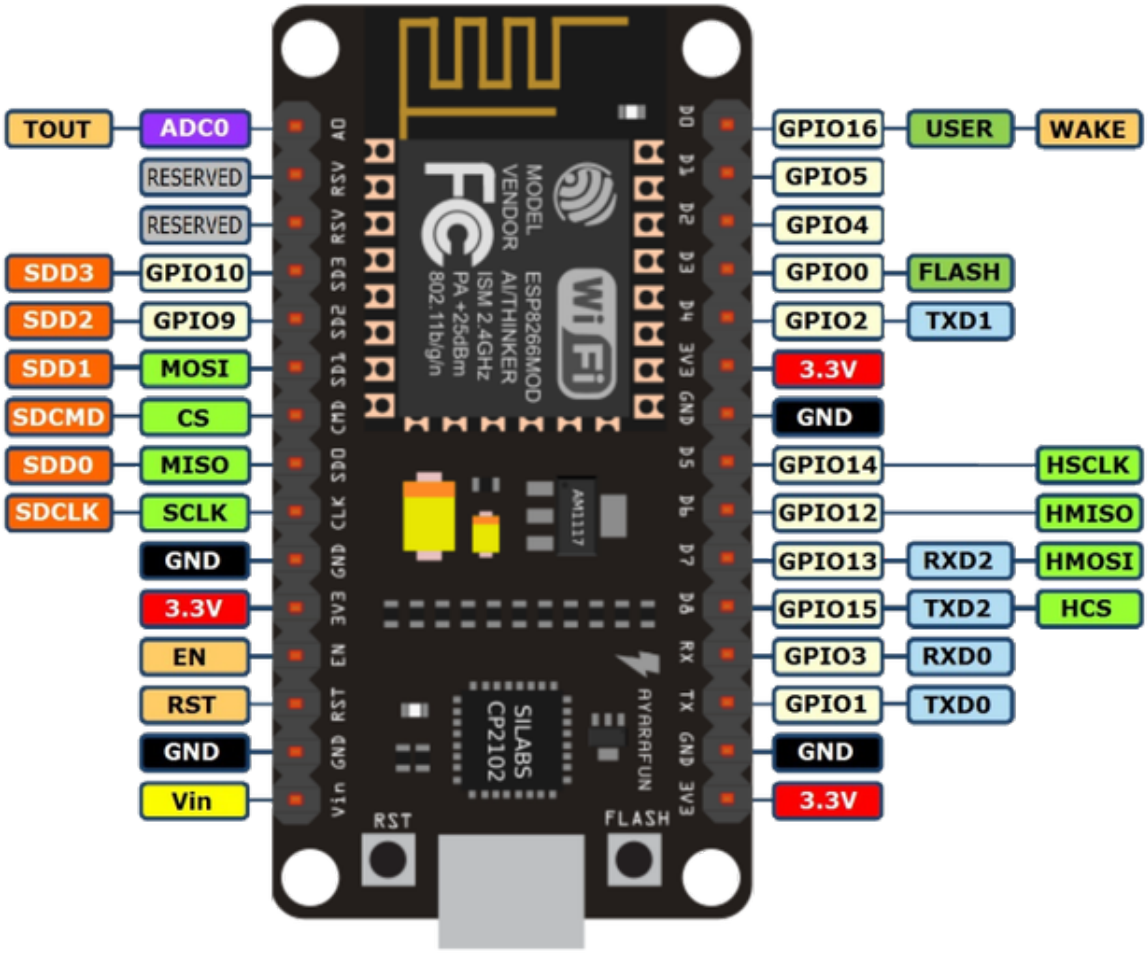

Fig. 3.11: General Purpose Input/Output de NodeMCU

En la figura 3.11 se muestran los GPIO del NodeMCU. Se conecta en D3 y D5 como SDA y SCL respectivamente para comunicar mediante I2C. Se utiliza D7 de

cada PDE para controlar el Relay.

3.2.7.4. *Esquema de conexión master-slave*

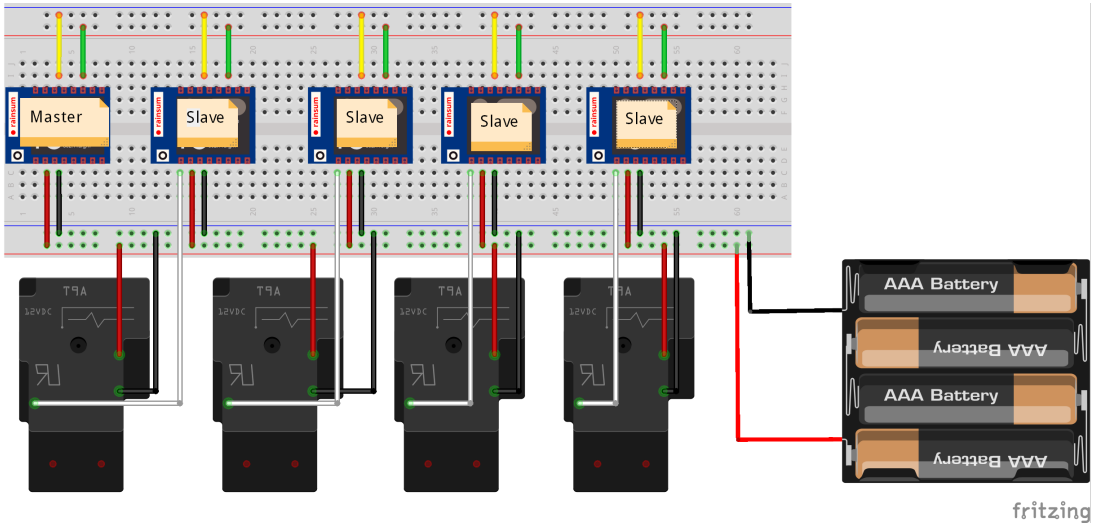

Fig. 3.12: Esquema de conexión master-slave prototipo

En la figura 3.12 se observa el esquema de conexión master-slave para el prototipo:

- El master es un NodeMCU que está conectado al router de la EB.
- Los slave son NodeMCU que reciben instrucciones mediante I2C.
- De las líneass superiores, la amarilla es SDA y la verde es SCL del protocolo I2C.
- De las líneass inferiores. la roja es VCC, la negra es GND y la blanca es el GPIO de control del Relay del *Electronic Rotary Latch*.
- La batería representa la fuente de alimentación.

Los ESP01 de las PDE pueden estar instaladas por cada uno o pueden estar todas concentradas en la EB:

- Si se coloca una ESP01 en cada PDE, se facilita la instalación de las PDE debido a que el solo hecho de desconectar una provoca la reconexión de la siguiente. Por otro lado, esta decisión implica que la estructura física de la PDE debe ser muy robusta, de forma de proteger la electrónica.

- Si se colocan todos los ESP01 de cada PDE en la EB, se puede trabajar de manera más sencilla por el lado de la electrónica, es decir, no se necesita abrir cada PDE en caso de que se necesite un cambio. Permite además hacer el diseño electrónico más sencillo en el sentido de que, como están todos los dispositivos en el mismo lugar, se debe fabricar un circuito integrado que permita enchufar y desenchufar fácilmente cada componente. Por otro lado, esto impide el cambio automático de PDE, puesto que al sacar una PDE también se debe sacar su respectivo ESP01.

3.2.8. Diseño de módulo [M8]: Lógica de acción de estacionamientos

Ejecuta las funciones necesarias para responder a las peticiones de usuario y administrador en función de la información presente en la base de datos. Además genera los cambios necesarios en la base de datos en pos de mantener en sincronía la información entre las EB/PDE y el servidor. Para su funcionamiento, el sistema necesita los siguientes módulos para la lógica de acción:

- Generador de números pseudo-aleatorios (PRNG)[30].

- Sistema de criptografía simétrica[32].

- Lógica de acciónamiento en EB y PDE.

- Rutina de selección de ”llaves de instrucción”(LDI) seguras.

- Rutina de acuerdo de llaves privadas(PK) para instalación de EB nueva.

- Rutina de acuerdo LDI para instalación de PDE nueva.

- Rutina de preconfiguración de EB.
- Rutina de preconfiguracion de PDE.
- Rutina de apertura de estacionamiento.
- Rutina de cambio de PK y LDI.
- Rutina de generación de número 2FA.
- Rutina de cambio de 2FA en EB y servidor.

3.2.8.1. Generador de números pseudo-aleatorios (PRNG)[30]

Es una función que genera números de forma ”aleatoria”. Se conoce como pseudo-aleatoria porque la función es determinista; para los mismos parámetros en *input* produce los mismos números en el *output*. Su característica es que son funciones unidireccionales, i.e. son fáciles de calcular y difíciles de invertir. Además, deben ser altamente entrópicas, por lo que *input* distintos deben generar *output* distintos.

En los módulos de lógica de acción, el PRNG es fundamental, ya que mediante él se puede generar:

- Sistema de encriptación simétrico[32].
- Generación de llaves seguras[31].
- Generación de código 2FA[22].

Para el diseño de los módulos siguientes, se utiliza una PRNG de 6 parámetros, mediante la cual se genera un número que se utiliza para diversos propósitos. Dicha

función corresponde a una serie de funciones matemáticas, descritas a continuación:

$$M(b,n,\mu) = \frac{b}{10^n - \mu} = \sum_{r=0}^{\infty} \frac{b \cdot \mu^r}{10^{n(r+1)}} \tag{3.13}$$

$$SCF(b,n,\mu,p) = 10^p \cdot M(b,n,\mu) \tag{3.14}$$

$$Z(b,n,\mu,p,i) = mod(SCF(b,n,\mu,p), 10^i) \tag{3.15}$$

$$RCF[b,n,\mu,p,i,j] = \frac{Z(b,n,\mu,p,i+j) - Z(b,n,\mu,p,i)}{10^i} \tag{3.16}$$

El número pseudo-aleatorio se genera a través de la función RCF. Este PRNG tiene las siguiente características:

- Es una función unidireccional, altamente entrópica y que se calcula mediante operaciones matemáticas de bajo costo computacional.
- Utiliza los siguientes parámetros:
  - b = parámetro fijo. Puede variar bajo ciertas restricciones, pero para este caso es innecesario.
  - n = parámetro variable.
  - μ = parámetro variable.
  - p = parámetro variable.
  - i = parámetro variable.
  - j = parámetro variable. Corresponde al ”largo” de la llave que encripta el mensaje, es decir, la cantidad de dígitos que tiene el número generado con el PRNG.

  Estos parámetros son conocidos como las ”llaves privadas(PK)”.

Las demostraciones de su alta entropía y su característica de ”función unidireccional.[es]tán fuera del alcance de la investigación presentada en este documento, pero dichas demostraciones han sido ejecutadas apropiadamente.

3.2.8.2. *Sistema de criptografía simétrica[32]*

Sistema de encriptación que utiliza la misma llave tanto para encriptar como para desencriptar el mensaje. Esta llave se conoce como ”llave privada”. El principal inconveniente de un sistema de encriptación simétrico es la distribución de llaves, i.e. como iniciar la comunicación con una contraparte sobre un canal inseguro sin que ambas partes hayan acordado que llave utilizar. De todas formas, existen mecanismos que solventan el problema de la encriptación simétrica como el esquema Diffie-Hellman [32], mediante el cual se puede hacer este cambio de llaves de forma segura sobre un canal inseguro,e.g. internet. El sistema de encriptación simétrica se describe a continuación:

- Se utiliza el PRNG para generar un número muy grande, i.e. con múchos digitos. Dicho número se ”multiplica” con el mensaje . Luego, el receptor debe generar el mismo número grande con el mismo PRNG y las mismas llaves privadas(PK), para luego dividir el mensaje cifrado por dicho número y obtener el mensaje. A partir de que los parámetros del PRNG pueden ser considerados como un ”secreto compartido”, se puede decir que el método acá presentado es un sistema de encriptación simétrico.

- Si denominamos F como la función de PRNG, C como el número generado mediante F que se multiplica por el mensaje M y E es el mensaje encriptado mediante la multiplicación de C y M, se puede representar las operaciones de cifrado y descifrado.

  Para el cifrado, se envía un código E al receptor.

$$F(b, n, \mu, p, i, j) = C \tag{3.17}$$

$$E = M \cdot C \tag{3.18}$$

  Para el descifrado, el receptor descifra E mediante la división de E con C, gene-

rando C con F y las mismas llaves privadas(PK) que utiliza el emisor.

$$E = M \cdot C \tag{3.19}$$

$$C = F(b, n, \mu, p, i, j) \tag{3.20}$$

$$M = \frac{E}{C} \tag{3.21}$$

- Como el PRNG es una función altamente entrópica, cambiar uno de los parámetros de F genera un número C completamente distinto al anterior, por lo que E varía en cada cambio de parámetro.
- Si un atacante quisiera interpretar E, debe realizar su factorización prima[33]. La criptografía actual se basa en el hecho de que no existen algoritmos computacionales que puedan hacer esta tarea en tiempo lineal, i.e. la factorización prima de E requeriría un tiempo considerable, en el cual los parámetros de F ya habrían cambiado muchas veces.
- Si consiguiese hacer dicha factorización prima, se encontraría con el problema de que no podría distinguir cuales factores corresponden a M y cuales factores corresponden a C.
- De esta forma, quien quiera ”adivinar” la salida de dicha función debe generar muestras en $\mathbb{R}^6$, con $b, n, \mu, p, i, j \in \mathbb{N}$.
- Como los mensajes comunicados M corresponden a instrucciones digitales, no se necesita un sistema que genere un mensaje M alfadecimal, basta con establecer que las instrucciones en cada EB correspondan a un número, i.e. se debe determinar parámetros que sean interpretables como instrucciones. Esto es, ON = 127 y OFF =372, por ejemplo.
- Se debe establecer un mecanismo que permita acordar los cambios de llave tanto en servidor como en EB, de forma que no resulte obvia para el atacante saber cuales son las siguientes llaves, representadas por los parámetros de F.

3.2.8.3. Lógica de accionamiento en EB y PDE

Como se explica con anterioridad, el proceso de encriptado y desencriptado es como sigue:

$$F(b, n, \mu, p, i, j) = C \tag{3.22}$$

$$E = M \cdot C \tag{3.23}$$

$$C = F(b, n, \mu, p, i, j) \tag{3.24}$$

$$M = \frac{E}{C} \tag{3.25}$$

El mensaje M corresponde a lo que la EB interpreta como una instrucción. Como dicho mensaje es un número, se deben establecer la ”lógica” que debe cumplir.

Las instrucciones que la EB debe interpretar en el mensaje M del servidor son:

- Acordar las llaves que se utilizan con el servidor en el inicio de la comunicación.
- Abrir una PDE en particular.
- Cambiar el 2FA.
- Cambiar una llave en particular.

De este modo, se debe tener la interpretación de los mensajes M de antemano, i.e. se debe tener establecido el valor numérico para cada instrucción antes de la lectura de un mensaje M.

Para efectos de demostración se puede ilustrar la apertura de una PDE. Se escoge no cifrar M, de forma que las PDE solo deben interpretarla más no descifrarla.

La decisión de apertura de una PDE se genera en el servidor. Este debe enviar un mensaje cifrado E a la EB mediante el protocolo de comunicación diseñado en M6, i.e. debe publicar E en el tópico correspondiente a la EB objetivo. Como la EB no debe decidir si prestar o no una PDE, solo debe encargarse de enviar la instrucción de apertura a la PDE que indica el servidor en el mensaje M cifrado en E por medio C

según lo expuesto en la ecuación 3.25. Mediante I2C, la EB publica el mensaje M ya descifrado a sus *slaves* representados por las PDE. Todas las PDE leerán el mensaje, pero el mensaje M solo está dirigido a la PDE que el servidor decidió prestar al usuario y solo aquella PDE ejecutará la orden de apertura. Para ello, el mensaje M puede o no estar cifrado, de forma que esté compuesto por la orden de apertura, la identidad de la PDE y un valor numérico que encripte el mensaje del mismo modo que C encripta M. Esta acción requiere que las PDE desencripten el mensaje M de la misma forma que la EB descifro E para obtener M. Como la comunicación entre EB y PDE es por cable, no hay riesgo de un ataque MITM, ergo, no es necesario cifrar dicha comunicación más allá de la interpretación de las órdenes numéricas. El cifrado de M solo agrega otra capa de seguridad para quien intercepte la comunicación EB-servidor.

En concreto, siendo C el número de gran cantidad de dígitos generado por F(b,n,μ,p,i,j), E el mensaje que se entrega a la EB, resultado de multiplicar C por M, I es la identidad de la PDE que se quiere abrir y A es el número que representa la orden de apertura, se tiene:

$$C = F(b, n, \mu, p, i, j) \tag{3.26}$$

$$I = 127 \tag{3.27}$$

$$A = 311 \tag{3.28}$$

$$M = I \cdot A = 127 \cdot 311 = 39497 \tag{3.29}$$

$$E = M \cdot C = 39497 \cdot C \tag{3.30}$$

El mensaje E se publica en el tópico correspondiente a la EB donde se encuentra la PDE que se desea abrir.

Visto desde el lado de la EB, al recibir E debe descifrar y enviar el mensaje M a todas las PDE, de forma que solo la PDE que se desea abrir pueda interpretar el mensaje. La EB puede descifrar E ya que sabe los valores de las llaves privadas(PK)

(b,n,μ.p,i,j):

$$E = M \cdot C \tag{3.31}$$

$$C = F(b, n, \mu, p, i, j) \tag{3.32}$$

$$M = \frac{E}{C} \tag{3.33}$$

$$M = I \cdot A = 39497 \tag{3.34}$$

Así, la EB ”publica” M en la línea de control SDA de I2C, de forma que todas las PDE leen el mensaje.

Para ilustrar la interpretación de M, veremos el caso de dos PDE, una con el I distinto al I del mensaje M, y otra con otro I igual al I del mensaje M, es decir, una que es el receptor del mensaje y otra que no.

De la PDE que no es receptora,PDE 1 , y que ya recibió el mensaje M:

```
I=137 //Llave de instruccion , identidad
A=311 //Llave de instruccion , accion de apertura
B=97 // Llave de instruccion, instruccion 2
M = 39497 //Mensaje recibido por I2C

DEF LEER_MENSAJE (M):
    IF (mod(M,I) == 0) THEN
        PRINT ("El mensaje es para esta PDE")
        IF (mod(M/I,A) == 0) THEN
            PRINT ("Abrir estacionamiento")
            PIN.RELAY == ON
        ELSE IF (mod(M/I,B) == 0) THEN
            PRINT ("Ejecutar instruccion 2")
            EXECUTE (instruccion_2)
    ELSE
        PRINT ("El mensaje no es para esta PDE")

LEER_MENSAJE(M)
```

Y su *output*:

```
"El mensaje no es para esta PDE"
```

Esto es debido a que el módulo de M y I es distinto de 0 en la PDE 1, es decir, el mensaje no estaba dirigido a esta PDE.

Si vemos el caso de una PDE que es la receptora, PDE 2:

```
I=127 //Llave de instruccion , identidad
A=311 //Llave de instruccion , accion de apertura
B=97 // Llave de instruccion, instruccion 2
M = 39497 //Mensaje recibido por I2C

DEF LEER_MENSAJE (M):
    IF (mod(M,D) == 0) THEN
        PRINT ("El mensaje es para esta PDE")
        IF (mod(M/D,A) == 0) THEN
            PRINT ("Abrir estacionamiento")
            RELAY.PIN == ON
        ELSE IF (mod(M/D,B) == 0) THEN
            PRINT ("Ejecutar instruccion 2")
            EXECUTE (instruccion_2)
    ELSE
        PRINT ("El mensaje no es para esta PDE")

LEER_MENSAJE(M)
```

Y su *output*:

```
"El mensaje es para esta PDE"
"Abrir estacionamiento"
```

Debido a que el módulo de M e I es cero, y que el módulo de M/I y A también es cero.

Este es un ejemplo de como preconfigurar las instrucciones. Nótese que también existe una llave B, para ejecutar una instrucción 2. De este modo, si M = I*B = 127*97 = 12319, entonces la segunda PDE ejecutaría la instrucción 2.

Estas llaves, A y B, son las ”llaves de instruccion” (LDI) de acción, y la llave I es la ”llave de instrucción” de identidad. Deben ser conocidas por el servidor para ejecutar las rutinas necesarias para enviar instrucciones a los distintos IoTD del siste-

ma. Nótese también que las LDI deben ser coprimas entre sí, i.e. no debe compartir factores primos entre sí. Esto se debe a que para que cada PDE ejecute la función LEER_MENSAJE de forma correcta, solo uno de ellos lo debe leer como propio. Si llaves de instrucción no fueran coprimas entre sí, puede ocurrir que más de una PDE ejecute dicha instrucción. De todas formas, las LDI deben ser coprimas entre sí, pero no entre IoTD,i.e. la LDI de instrucción 1 a n pueden ser las mismas entre IoTD pero no en cada uno de ellos. La única LDI que debe ser absolutamente distinta entre PDE es la LDI de identidad. De nuevo, esta restricción es para cada ”sistema autónomo”, i.e. para cada una de las PDE que pertenecen a una EB las LDI de identidad deben ser distintas, pero se pueden repetir entre PDE de distintas EB. Esto queda ilustrado en las LDI de identidad y acción de los dos casos mencionados; la LDI de identidad de la PDE 1 es distinta que la PDE 2, pero sus LDI de acción son iguales.

3.2.8.4. Rutina de selección de ”llaves de instrucción” (LDI) seguras

Una vez iniciada la comunicación, se procura ir cambiando las llaves de instrucción de los distintos dispositivos, de forma de incrementar el comportamiento aleatorio para los atacantes que logren leer el mensaje. Esto se debe hacer en pos de la seguridad del sistema, pero se debe diseñar de forma que no disminuya el rendimiento en términos de transmisión, procesamiento y almacenamiento de datos.

Se necesita un método que genere llaves seguras con la mayor eficiencia computacional y aleatoriedad posible.

Como se menciona en la ”Lógica de accionamiento en EB y PDE”, las LDI de identidad y acción deben ser coprimas entre sí en cada PDE de manera individual, las LDI de identidad deben ser coprimas entre PDE distintas y las LDI de acción pueden no ser coprimas entre distintas PDE, incluso en este último caso pueden ser iguales.

Se pueden utilizar números primos para las LDI, ya que todos ellos son coprimos entre sí. El *trade-off* es que al utilizar un número primo, solo dos factores de la factorización prima de E corresponderían a LDI de identidad y LDI de acción. Esto es, la dificultad de ”adivinación” de cuales factores corresponden a C y cuales corresponden

a LDI disminuye con respecto al caso de que las LDI sean números compuestos, es decir, sean el resultado de la multiplicación de diversos números primos.

Cabe destacar que en la actualidad dicha factorización es computacionalmente muy costosa, y virtualmente imposible, por lo que no debería ser una mayor preocupación. El problema es que dicha afirmación puede cambiar en la medida que aparezca alguna forma de hacer factorización prima en tiempo líneasl. Un ejemplo de ello es el Algoritmo de Shor [34], que permite hacer la factorización prima en tiempo lineal, pero requiere un computador cuántico para su ejecución.

Por otro lado, la elección de un número C grande permite asegurar que dicho número es no-primo. Esta afirmación se sustenta en la *función enumerativa de números primos* $\pi(n)$[33] conjeturada por Gauss en el siglo XVII:

$$\pi(n) = \frac{n}{ln(n)} \tag{3.35}$$

Esta función indica la cantidad de números primos que existen en el intervalo [1,n]. Por ejemplo, si se tiene $n = 10$, entonces la cantidad de números primos en el intervalo [1,10] es:

$$\pi(10) = \frac{10}{ln(10)} \approx 4{,}34294481903 \tag{3.36}$$

Es decir, hay 4 números primos en el intervalo [1,10]. Dicha afirmación es correcta, puesto que en el intervalo [1,10] existen 4 primos: 2, 3, 5, 7.

Por otro lado, la probabilidad $\phi(n)$ de que un número sea primo en el intervalo [1,n] es:

$$\phi(n) = \frac{\pi(n)}{n} = \frac{1}{ln(n)} \tag{3.37}$$

Por ejemplo, para $n = 10^{10}$:

$$\pi(10^{10}) = \frac{10^{10}}{ln(10^{10})} \approx 434294481{,}903 \tag{3.38}$$

$$\phi(10^{10}) = \frac{1}{ln(10^{10})} \approx 0{,}04342944819 \tag{3.39}$$

Es decir, existen 434294481 primos en el intervalo [1,10^{10}], lo que da una probabilidad de $4{,}34\,\%$ de que el número generado por el PRNG sea primo si es que aquel número tiene por lo menos 10 dígitos.

Nótese que $\phi(n)$ es inversamente proporcional a n, quedando entonces demostrado que es altamente probable que C no sea primo si es que es un número de muchos dígitos.

Por otro lado, existen funciones que generan números primos, llamadas *fórmulas de números primos* [35] que podrían utilizarse para la elección de las LDI. Si se quisiera utilizar LDI que sean números compuestos, se pueden obtener diversos números primos mediante las *fórmulas de números primos*, multiplicarlos entre ellos, y utilizarlos como LDI de identidad y acción, siempre con el cuidado de mantener la condición de coprimalidad ya mencionada.

Para efectos de simplificación de implementación, se utilizan solamente números no compuestos para las LDI, i.e. LDI son números primos. Estos pueden ser extraidos de una tabla o generados con las *fórmula de números primos*. Queda por demostrar entonces que esta elección es suficientemente segura para el caso de que un atacante logre hacer la factorización prima de E. Para efectos de demostración, se escoge un C con 128 dígitos.

$$\pi(10^{128}) = 3{,}392926 \cdot 10^{125} \tag{3.40}$$

$$\phi(10^{128}) = 0{,}003392926 \tag{3.41}$$

Esto quiere decir que existe un $0{,}3392926\,\%$ de que el número C sea primo, lo que es despreciable en la práctica. Dicho eso, el número C tendrá por lo menos dos factores

primos en el peor de los casos, ya que de otra forma sería primo.

Si C se multiplica por el mensaje M, que es la multiplicación de la LDI de identidad y la LDI de acción, entonces se tiene que $E = M \cdot C$ tiene por lo menos 4 factores primos, i.e. 2 factores primos de C, y dos de M debido a que este último es la multiplicación de LDI de identidad y LDI de acción, ambos primos.

Si en alguna ocasión el atacante lograse hacer la descomposición prima de E, entonces obtendría que:

$$E = P_1 \cdot P_2 \cdot P_3 \cdot P_4 \tag{3.42}$$

Siendo P_n el factor primo n de E.

El atacante se encuentra ahora con el problema de identificar cuales factores primos corresponden a C, a LDI de identidad y a LDI de acción. Por consiguiente, se tiene que:

- δ_i es la probabilidad de que el atacante "adivine" cual factor primo de E es la LDI de identidad.
- δ_a es la probabilidad de que el atacante "adivine" cual factor primo de E es la LDI de acción.
- δ_t es la probabilidad de que el atacante "adivine" ambas al mismo tiempo.

. Entonces:

$$\delta_i = 1/4 \tag{3.43}$$

$$\delta_a = 1/4 \tag{3.44}$$

$$\delta_t = \delta_i \cdot \delta_a = 1/16 = 6{,}25\,\% \tag{3.45}$$

Despreciable desde un punto de vista de probabilidades. Nótese que no es necesario incluir la probabilidad de adivinar C, puesto que al adivinar LDI de identidad y LDI de acción, automáticamente los factores restantes corresponden a C.

Si en alguna desafortunada ocasión, PRNG genera un C que es primo, entonces E tendría 3 factores primos, i.e. 1 factor primo de C y 2 de las LDI, por lo que la probabilidad de adivinación quedaría:

$$\delta_i = 1/3 \tag{3.46}$$

$$\delta_a = 1/3 \tag{3.47}$$

$$\delta_t = \delta_i \cdot \delta_a = 1/9 \approx 11,1\,\% \tag{3.48}$$

Dicho caso queda sujeto a la probabilidad de que C sea primo. Teniendo que C tiene 128 dígitos, la probabilidad δ_f de que ocurra C primo y al mismo tiempo el atacante adivine cual factor corresponde a C y cuales a LDI es:

$$\delta_f = 1/9 \cdot 0{,}003392926 \approx 0{,}00037699177 \tag{3.49}$$

Es decir, una probabilidad de $0{,}037699177\,\%$, despreciable desde el punto de vista de probabilidades.

Vale mencionar que existe la función $\Omega(n)$[36] que permite conocer la cantidad de factores primos de un número, más no cuales son aquellos factores. Mediante esta función se puede determinar la cantidad de factores primos de C, pero como se ha trabajado sobre el peor caso posible, no resulta necesario encontrar dicha cantidad. Es decir, si es que la cantidad de factores primos de C es mayor que 2, o si LDI de identidad y LDI de acción son números compuestos por múltiples primos, teniendo los cuidados ya indicados, entonces el porcentaje obtenido en la ecuación 3.45 disminuiría, mejorando la seguridad del método acá propuesto.

En resumen, el atacante debe:

- Hacer la factorización prima de E.
- Identificar cuales factores corresponden a C y cuales corresponden a LDI de identidad y LDI de acción.

Ambos procesos resultan infructuosos si se sabe que la factorización prima es virtualmente imposible sin un computador cuántico y que la probabilidad de adivinar cuales factores de E corresponde a C y LDI es baja. Si a eso se suma que el valor de las llaves privadas(PK) (b,n,μ,p,i,j) pueden cambiar con cada orden ejecutada en las PDE, entonces el incentivo para intentar un ataque disminuye drásticamente.

Otra forma de ataque sería probar con cada C posible, es decir, que el atacante utilice F(b,n,μ,p,i,j) con cada valor de llave privada posible. Como la llave privada j representa la cantidad de dígitos de C, el atacante tendría que probar todos los números en un intervalo [1,10^j], puesto que la función F es altamente entrópica y genera casi tantas llaves como 10^j. Dicho ataque puede ser mitigado con una configuración de sistema tal que imposibilite una cantidad de intentos mayor a k. A modo de ejemplo, si k = 100 y j=10, entonces la probabilidad de acierto δ_p es:

$$\delta_p = \frac{100}{10^{10}} = 0,000001 \tag{3.50}$$

Es decir, una probabilidad de acierto de $0,0001\,\%$, lo que hace que sea prácticamente imposible realizar un ataque para un k relativamente grande y un j relativamente pequeño.

Queda entonces demostrado que la elección de las LDI como números primos es suficientemente segura para el peor caso, i.e. en el caso de que C y las LDI sean números primos.

3.2.8.5. Rutina de acuerdo de llaves privadas(PK) para instalación de EB nueva

En pos de la escabalibilidad del sistema, se deben diseñar las PDE y EB de forma que su configuración de fábrica con sus respectivas llaves privadas(PK) y LDI no signifique un riesgo para el servicio. Esto es, si se necesitara implementar un gran número de EB y PDE, su configuración podría resultar en un proceso tedioso y arriesgado en el sentido de que sería muy fácil equivocarse.

Ante el problema del inicio del acuerdo de llaves en un canal inseguro como inter-

net se tienen dos alternativas:

1. El algoritmo de Diffie-Hellman [33], que soluciona al problema del inicio de comunicación en un esquema simétrico.

2. Configurar las EB desde fábrica con un secreto compartido con el servidor para iniciar el proceso de comunicación y acordar las llaves.

Para efectos de prototipo, se utiliza la segunda alternativa. Cuando el sistema esté en producción, se puede decidir si la primera o la segunda alternativa es la más adecuada en términos de seguridad, i.e. la implementación de la primera alternativa no debe impedir que la segunda sea implementable.

El proceso de acuerdo de llaves se describe a continuación:

1. La EB nueva debe estar configurada para:

    - Publicar en un tópico del broker MQTT designado para iniciar el proceso de acuerdo de llaves con el servidor, llamado tópico 0.
    - Suscribirse a un tópico mediante el cual recibirá las PK desde el servidor, llamado tópico 1.

2. El acuerdo de llaves lo inicia la EB nueva mediante:

    - La publicación de un mensaje en el tópico 0 que la autentique con el servidor y la identifique de forma única.
    - La suscripción al tópico 1.

3. Como cada IoTD de las EB cuenta con conexión WiFi, cada una de ellas cuenta también con una dirección MAC única, la cual será utilizada para identificar la EB con el servidor.

4. La EB debe utilizar las PK iniciales para generar un número con el PRNG, de forma que el servidor haga lo mismo y pueda autenticar a la EB.

5. El mensaje publicado por la EB en el tópico 0 estará compuesto entonces por su dirección MAC y el número generado por el PRNG con las PK.

6. Antes de publicar el mensaje, la EB debe suscribirse al tópico 1:

    - Como el mensaje publicado en el tópico 0 contiene la dirección MAC y esta es única, se utilizará para suscribirse al tópico 1.
    - Esto es debido a que el servidor puede publicar en el tópico 1 con la seguridad que solo estará enviando las PK a la EB correspondiente.
    - Para suscribirse al tópico 1, debe tomar los seis pares hexadecimales de MAC, transformarlos a base decimal y utilizarlos como los parámetros del PRNG.
    - El número resultante de aquella operación es el nombre del tópico 1, al cual la EB se suscribe.

7. El nombre del tópico 0 es el número generado por el PRNG con las PK iniciales.

8. El servidor debe estar esperando un mensaje en el tópico 0, mediante el cual recibe peticiones de acuerdo de llaves.

9. Cuando el servidor lea el mensaje, debe separar el MAC del número generado por el PRNG.

10. Luego debe generar un número con las PK y el PRNG y compararlo con el que viene en el mensaje.

11. Si coinciden, procede al siguiente paso. En caso contrario, registra el incidente en la base de datos y no continúa.

12. El servidor genera las PK y las almacena junto con el MAC y el tópico 1 en la base de datos. Asigna además un ID único a la EB en el servidor, de forma que sirva para futuros cambios en las llaves y búsqueda en la base de datos.

13. El servidor utiliza el MAC para publicar las PK nuevas en el tópico 1 de la misma forma que hizo la EB para suscribirse; descompone el MAC y lo utiliza como parármetros del PRNG.

14. El servidor comienza entonces a publicar las PK en el tópico 1.

15. Al terminar, el servidor queda a la espera de la confirmación de recepción desde la EB. Para ello se suscribe al tópico 1, donde la EB publicará su mensaje de confirmación.

16. La EB lee las PK en la medida que se van publicando en el tópico 1 y las almacena.

17. Al recibir todas las PK la EB debe informar al servidor que el almacenamiento fue exitoso mediante un mensaje de confirmación.

18. Este mensaje es el número generado por PRNG con las PK nuevas.

19. La EB publica este mensaje en el tópico 1.

20. El servidor lee el mensaje y utiliza las PK nuevas para el PRNG. Si su resultado coincide con el mensaje, se actualiza el estado de la EB en la base de datos como ”exitosamente registrado” y se da por terminado el proceso de acuerdo de llaves.

Se debe mantener la suscripción en el tópico 1, a pesar de que las PK cambien con el tiempo. Dicho tópico sirve para tener un canal de comunicación autenticado mediante el uso de MAC, lo que puede servir para monitoreo de las estaciones.

3.2.8.6. Rutina de acuerdo de LDI para instalación de PDE nueva.

Para la distribución de LDI, el servidor debe ser capaz de enviar información a una EB, y esta debe distribuirla de forma que llegue a la PDE indicada. Esto incluye tanto la distribución de las LDI como una orden de apertura.

Las alternativas que existen son:

1. La EB conoce las LDI de identidad de las PDE, de forma que al llegar el mensaje desde el servidor sepa a que PDE corresponde y envíe las LDI mediante I2C.

2. La EB recibe el mensaje desde el servidor, y hace un *broadcast* del mensaje de forma que solo la PDE que tenga la LDI de identidad correcta ejecute la acción.

Cada alternativa tiene requerimientos distintos:

- La alternativa 1 requiere que la EB conozca la dirección I2C de la PDE objetivo. Esto quiere decir que la EB debe estar configurada para asociar una LDI de identidad con una dirección I2C.

- La alternativa 2 requiere que cada PDE conozca su LDI de identidad de antes de ser instalada, de forma de identificar que el mensaje en *broadcast* le pertenece a ella.

Esto quiere decir que ambas alternativas requieren que las PDE estén configuradas de fábrica, i.e. al cargar un programa a los IoTD de EB y de PDE se debe conocer cual pertenece a cada identidad. Como se requiere preconfigurar las EB con las PK, se decide utilizar la primera alternativa, ya que facilita el proceso de implementación.

En el caso de las PDE, la selección de LDI puede ser fácilmente solventado si es que se toma en cuenta la rutina para selección de LDI:

- Las llaves LDI de identidad deben ser coprimas entre distintas PDE.

- Las llaves LDI de acción deben ser coprimas en cada PDE.

- Las llaves LDI de acción pueden ser las mismas entre distintas PDE.

Se demostró además que basta con escoger números primos como LDI para que el sistema sea suficientemente seguro. Es por ello que se escoge el siguiente diseño:

- Las LDI de identidad serán números primos distintos entre cada PDE.

- Las LDI de acción serán números primos distintos para cada acción.

- Las LDI de acción se repetirán entre las distintas PDE del sistema.
- Se utilizan *funciones de números primos* para escoger las LDI.

Dicho eso, el proceso de acuerdo de LDI comienza al momento que el proceso de acuerdo de PK finaliza. Esto es, al recibir la confirmación desde la EB, el servidor puede comenzar a publicar las LDI en el tópico 1. Al terminar de recibirlas, la EB confirma su recibo de la misma forma que hizo con el proceso de acuerdo de PK.

3.2.8.7. Rutina de preconfiguración de EB

Como se dijo, las EB deben estar preconfiguradas con:

- Las PK para inicio de intercambio de llaves.
- Las direcciones I2C de cada PDE.
- Variables inicializadas en 0, de forma que sean reemplazadas por las PK y LDI provenientes desde el servidor.

Las direcciones I2C deben estar asociadas a un número de PDE, i.e. la dirección I2C 1 debe corresponder a la dirección I2C que tiene el IoTD de la PDE número 1. Así, cuando se necesite cambiar la PDE 1, se debe configurar el nuevo IoTD con la dirección I2C que la EB tiene preconfigurada para ”comunicarse” con la PDE 1.

3.2.8.8. Rutina de preconfiguración de PDE

Las PDE deben tener preconfiguradas:

- Sus direcciones I2C.
- Variables inicializadas en 0, de forma que sean reemplazadas por las LDI provenientes desde la EB.

. Cabe recordar que cada dirección I2C esta asociada a un número de PDE en la EB, por lo que al configurar la PDE en la fábrica, su IoTD debe estar etiquetado

con el número de estacionamiento que le corresponde, de forma que se instale en el estacionamiento correspondiente.

3.2.8.9. Rutina de apertura de estacionamiento.

Para la apertura del estacionamiento se deben considerar las siguientes condiciones:

- El usuario tiene una lista con los estacionamientos disponibles en la página de acceso.
- El administrador también tiene una lista con los estacionamientos en su página de acceso.
- Los dos puntos anteriores quieren decir que no se debe verificar que el *input* ambos casos sea correcto.
- Las LDI y las PK ya están acordadas.
- El usuario o administrador ingresaron correctamente los datos.

La rutina de apertura de estacionamiento comienza cuando el usuario o el administrador ingresaron los datos correctamente. La rutina se describe a continuación:

1. El servidor busca la EB correspondiente a la PDE en la base de datos.
2. El servidor extrae las PK con subfijo 1 de la tabla de EB-PK.
3. El servidor extrae las LDI de la tabla de PDE.
4. El servidor extrae el nombre del tópico 1 de la base de datos.
5. El servidor genera un número con el PRNG y las PK.
6. El servidor multiplica dicho código por las LDI.
7. El servidor publica el mensaje en el tópico 1, donde la EB recibe las órdenes.

8. La EB lee el mensaje en el tópico.
9. La EB decodifica el mensaje con las PK y el PRNG.
10. La EB identifica a la PDE destinataria con las LDI.
11. La EB envía el mensaje a la PDE correspondiente mediante I2C.
12. La PDE recibe el mensaje y certifica que sea para ella con las LDI.
13. La PDE ejecuta la acción de apertura de estacionamiento.
14. La PDE responde a la EB el resultado de la acción.
15. La EB calcula la multiplicación del número del PRNG con la respuesta de la PDE.
16. La EB publica el resultado en el tópico para comunicación con el servidor.
17. El servidor recibe el mensaje desde el tópico.
18. El servidor decodifica el mensaje.
19. El servidor interpreta el mensaje.
20. El servidor registra el evento de apertura de estacionamiento.

3.2.8.10. Rutina de cambio de PK y LDI

Los cambios de llaves pueden ser efectuados cuando:

- Se ejecute efectivamente una apertura de estacionamiento.
- Ocurra un evento que gatille el cambio. e.g. cambio de día o de hora.

La rutina del cambio de llave se inicia cuando ocurre uno de los dos eventos antes mencionados, y se hace para cada LDI o PK de cada EB en forma individual:

1. El servidor calcula las nuevas LDI o PK.

2. El servidor extrae las LDI de las tablas de EB.
3. El servidor extrae las PK de las tablas de EB.
4. El servidor calcula el número generado por las PK antiguas y el PRNG.
5. El servidor multiplica las LDI de la EB en cuestión con el número generado con el PRNG.
6. El servidor publica el mensaje en el tópico 1 de la EB en cuestión.
7. La EB lee el mensaje del tópico.
8. La EB decodifica el mensaje.
9. La EB ejecuta el cambio de PK o LDI.
10. La EB calcula el número con las PK nuevas y el PRNG.
11. La EB publica el número obtenido con el PRNG en el tópico 1.
12. El servidor lee el mensaje publicado en el tópico 1.
13. El servidor genera un número con las PK nuevas y el PRNG.
14. El servidor compara el mensaje con el número que generó.
15. Si coinciden, salta al paso siguiente. En caso contrario, vuelve al paso 1.
16. El servidor actualiza las LDI o PK en la base de datos.

Para poder hacer los cambios de llave, ya sea PK como LDI, la EB debe tener:

- Una LDI de acción para indicar cambio de llave.
- Una LDI de identificación, para saber que el cambio de llave es para si misma.
- Una LDI de acción para identificación cada PK que se cambiará.
- Una LDI de acción para identificación de cada LDI que se cambiará.
- Una LDI de identificación de cada PDE que controla.

3.2.8.11. Rutina de generación de número 2FA

Para la generación del 2FA se utiliza el PRNG. Los parámetros utilizados son las PK presentes en la base de datos, con la salvedad que el parámetro j e i deben elegirse para obtener un número de 6 dígitos.

3.2.8.12. Rutina de cambio de 2FA en EB y servidor

La rutina de cambio de 2FA se inicia cuando se abre exitosamente un estacionamiento:

1. El servidor extrae las PK de la base de datos.
2. El servidor extrae la LDI de cambio de 2FA de la base de datos.
3. El servidor decide el valor de i.
4. El servidor genera el 2FA con el PRNG y las PK extraidas, reemplazando el parámetro i con el seleccionado y j con un 6.
5. El servidor verifica que el 2FA generado no se repita con el de alguna otra EB. Si se repite, vuelve al paso 3.
6. El servidor genera el número con el PRNG y las PK extraidas.
7. El servidor multiplica el i con el número generado por el PRNG.
8. El servidor publica el resultado en el tópico 1.
9. La EB lee el mensaje en el tópico 1.
10. La EB decodifica el mensaje.
11. La EB calcula el 2FA en base a sus PK, reemplazando i con el recibido i recibida y j con un 6.
12. La EB guarda el 2FA.

13. La EB calcula el número generado por el PRNG con las PK.
14. La EB multiplica el 2FA con el número generado.
15. La EB publica el mensaje en el tópico 1.
16. El servidor lee el mensaje del topico 1.
17. El servidor decodifica el menasje.
18. El servidor compara su resultado con el del mensaje.
19. El servidor informa del cambio correcto a la EB mediante el tópico 1.
20. La EB recibe la confirmación y actualiza la vista en pantalla.

3.2.9. Diseño de módulo [M9]:Sistema de verificación de presencia (SVP) de usuario en estacionamiento

Mecanismo mediante el cual se certifica que el usuario está en el estacionamiento haciendo peticiones de apertura. Pensado para evitar el uso indebido del servicio.

Para su utilización:

- Cada EB tiene una pantalla indicando un número de 6 dígitos.
- El usuario debe ingresar dicho número junto con su nombre de usuario y contraseña si quiere estacionar o retirar su bicicleta.
- Este número cambia cada vez que un usuario pide un estacionamiento o retira su bicicleta de manera exitosa.
- Este número es generado por el IoTD de la EB, pero queda almacenado en la base de datos para compararlo con la información ingresada por el usuario en los formularios web.
- Se utiliza un PRNG para generar el número, de forma que su comportamiento sea impredecible.

El motivo de utilizar este mecanismo, es el de verificar que el usuario esté presente en el estacionamiento, puesto que de otra forma no sabría cual es el número que debe ingresar. Si bien podría haber alguien que le informe de forma remota cual es dicho número, hacer este ejercicio pierde sentido, ya que requiere un esfuerzo extra para ejecutar una acción que no genera mayor riesgo para el estacionamiento, i.e. lo único que puede pasar es que el usuario abra o cierre un estacionamiento, pero solo podría hacerlo con uno, dado que el sistema no permite la utilización de más de un estacionamiento.

Otro motivo para utilizarlo es la identificación del estacionamiento al que el usuario quiere acceder. Esto es, cuando el usuario quiere acceder al sistema, debe informar ”donde” es que quiere estacionar. Esto se puede lograr mediante la presentación de una lista de estacionamientos, pero por temas de interfaz y experiencia de usuario no es la mejor alternativa, i.e. se debe poner ”nombre” a cada estacionamiento, de forma tal que el usuario sea capaz de identificar cual es el nombre del estacionamiento que desea ocupar. Esto se soluciona mediante una búsqueda inversa; al ingresar el 2FA mediante la página web, se busca en la base de datos si dicho 2FA está asociado a algún estacionamiento, y si es que ese usuario tiene permisos de acceso a aquel estacionamiento, entonces se le muestra la lista de PDE disponibles. De esta forma también se evita que el usuario opere un estacionamiento al que no tiene acceso, lo que además facilita que las transacciones en la base de datos se mantengan atómicas, consistentes, aisladas y durables (principio ACID de diseño de base de datos, [21]).

3.2.10. Diseño de módulo [M10]:Sistema de respaldo de base de datos

Medida preventiva que establece una o más copias de la base de datos en pos de la disponibilidad e integridad de la información. Este servicio es suministrado por *Amazon Web Services* mediante su sistema RDS5 [23].Este se encarga de duplicar la instancia de base de datos, monitorear que la instancia principal esté funcionando, y en caso que esta falle, hacer inmediatamente los cambios necesarios para que el sistema siga funcionando con la instancia secundaria como reemplazo de la principal.

3.3. Diseño de interfaces

A grandes rasgos, el interfaz es una página web con las siguientes secciones:

- Registro de usuario.
- Pedir estacionamiento.

  Para pedir estacionamiento, se necesitan 2 páginas, una para ingresar los datos de inicio se sesión, y una segunda página que, una vez aceptado el login, indique al usuario cuales PDE están disponibles.
- Retirar bicicleta.
- Página de administrador.

  La página de administrador contiene:

  - Registrar usuario nuevo.
  - Abrir PDE.

Para el diseño de interfaces, se utilizan *mockups* que permita graficar como es que se verá la página web.

3.3.1. *Página de registro de usuario*

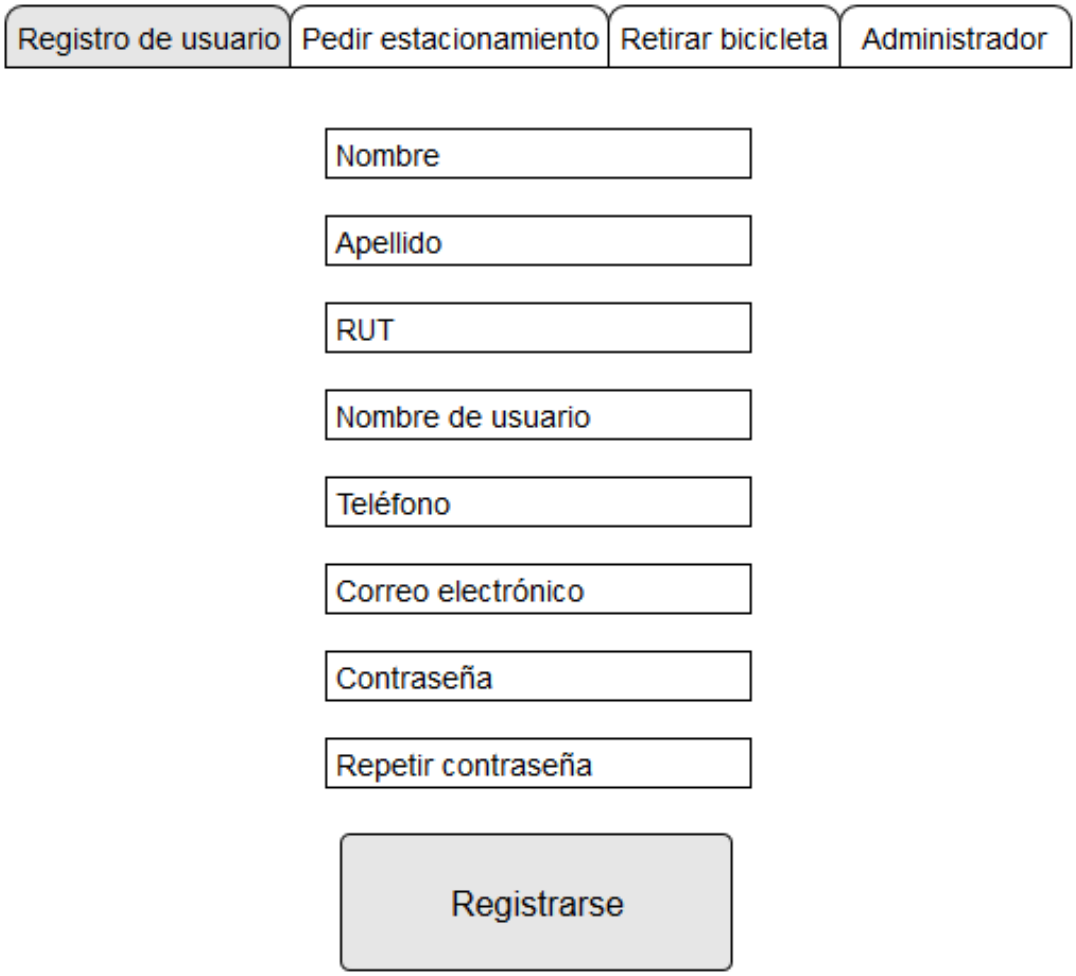

Fig. 3.13: Página de registro de usuario

3.3.2. Página de pedir estacionamiento

3.3.2.1. Identificación de usuario e ingreso de 2FA

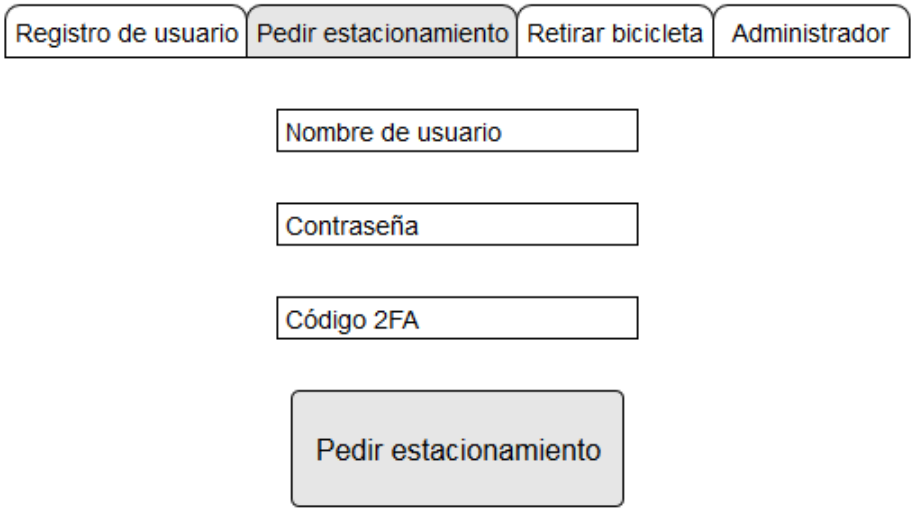

Fig. 3.14: Identificación de usuario e ingreso de 2FA

3.3.2.2. Elegir estacionamiento

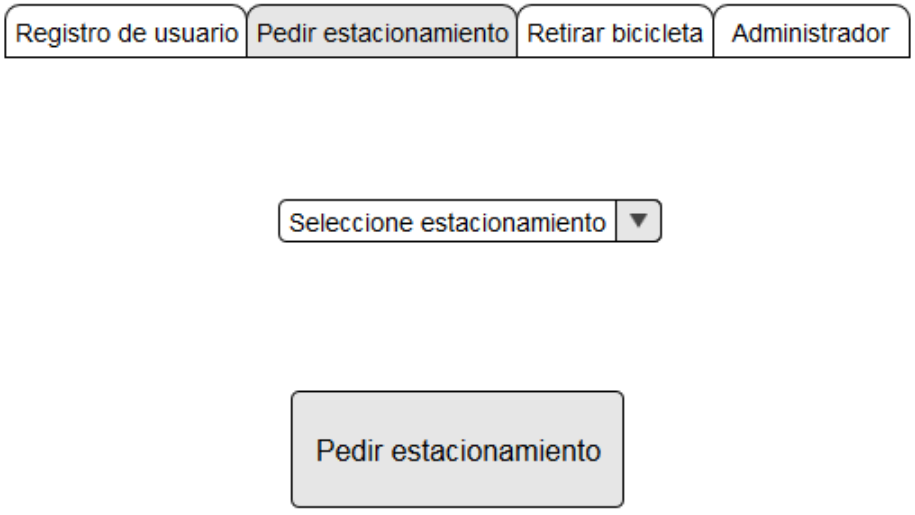

Fig. 3.15: Elegir estacionamiento

3.3.3. *Página de retirar bicicleta*

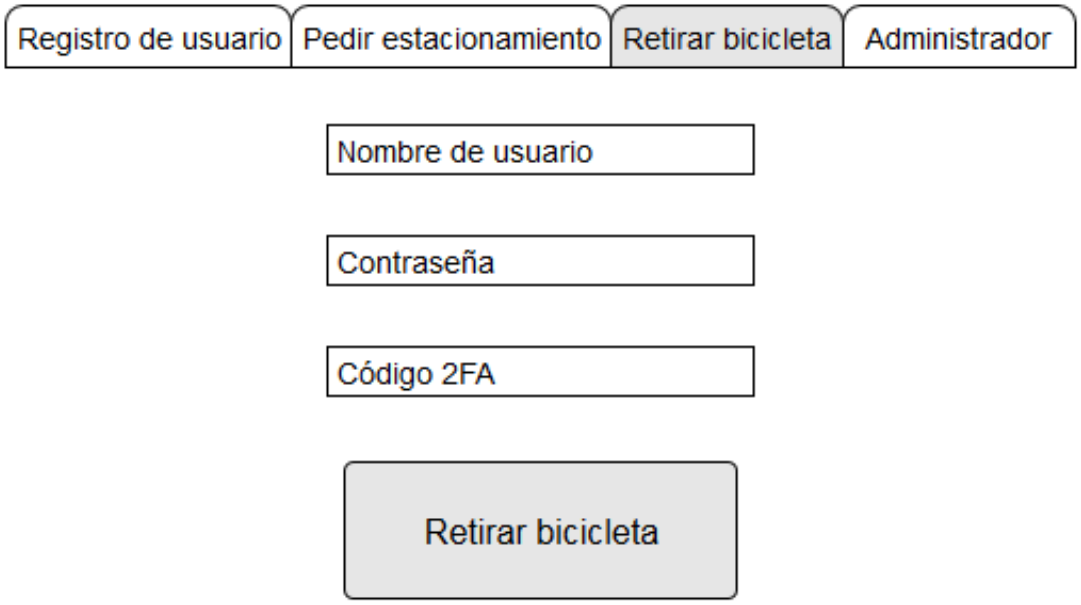

Fig. 3.16: Página de retirar bicicleta

3.3.4. *Página de inicio de sesión de administrador*

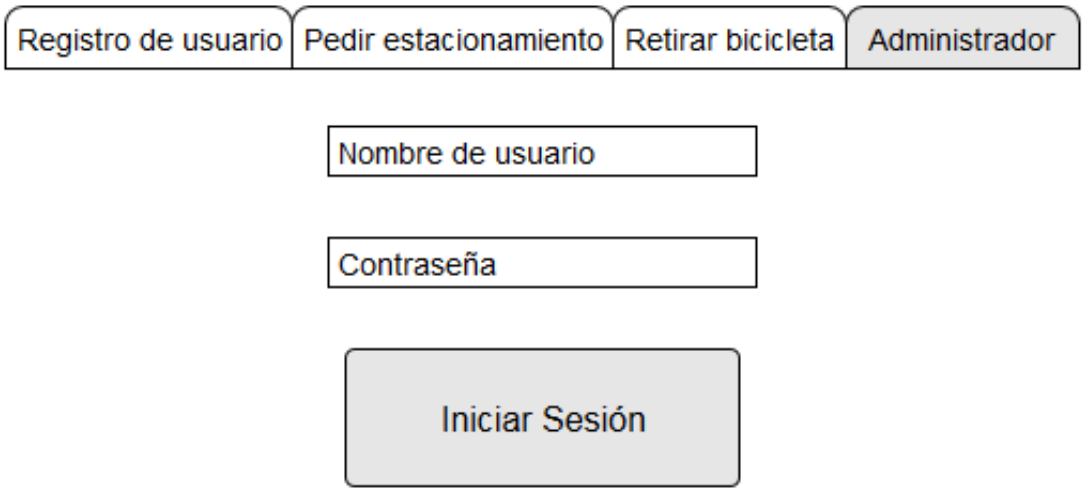

Fig. 3.17: Página de inicio de sesión de administrador

3.3.4.1. *Página de administrador, Autorizar usuario*

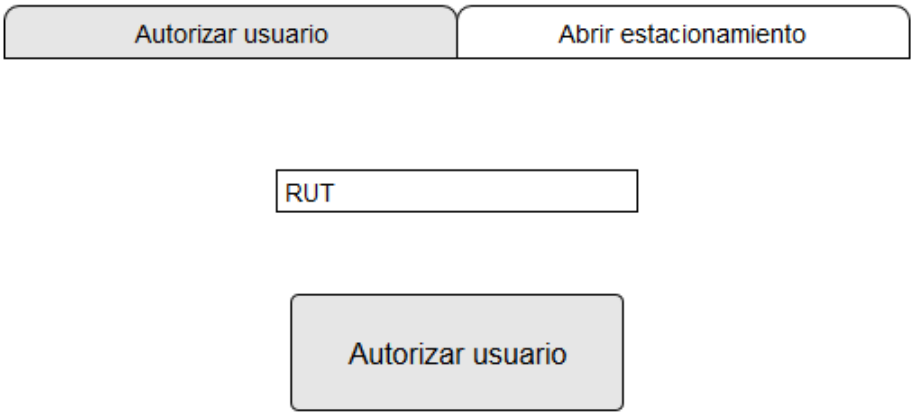

Fig. 3.18: Autorizar usuario

3.3.4.2. *Página de administrador, Abrir estacionamiento*

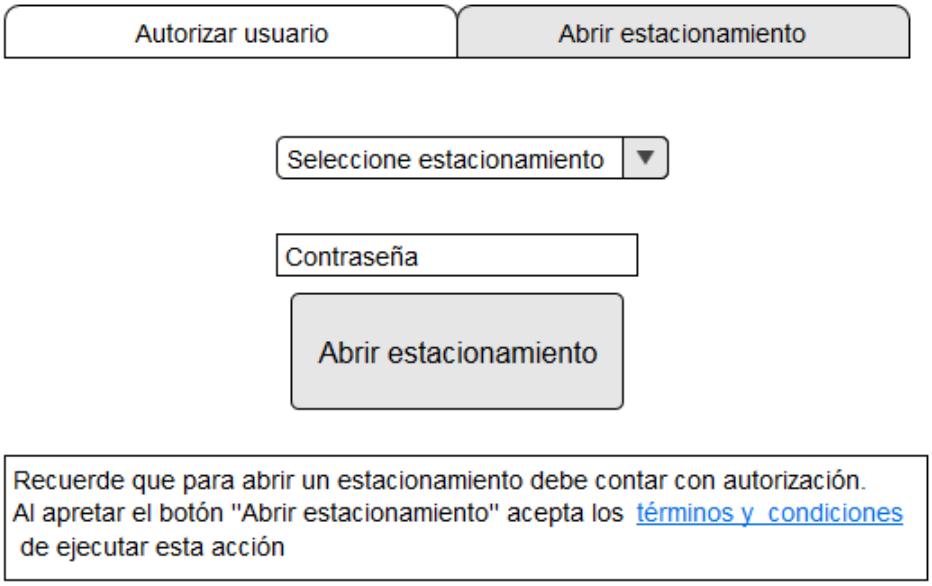

Fig. 3.19: Página de administrador, abrir estacionamiento

Para esta última figura, la lista debe mostrar la PDE a abrir, y si es que hay algún usuario utilizando dicha PDE, se muestra tambíen el RUT del usuario.

4. IMPLEMENTACIÓN DE PROTOTIPO

4.1. Marco General

4.1.1. Hitos principales del proyecto

Milestone: **Definición técnica del problema.**

Actividad 1: Identificación de los problemas específicos a resolver.

Tarea 1: Descripción de los sistemas de estacionamientos para bicicletas en el mundo.

Tarea 2: Descripción de los sistemas de estacionamientos para bicicletas en el mundo.

Tarea 3: Identificar sus fortalezas y sus debilidades.

Actividad 2: Definición de requisitos del sistema.

Tarea 1: Definición de los requerimientos no funcionales.

Tarea 2: Definición de los requerimientos funcionales.

Tarea 3: Definición de los requerimientos de arquitectura.

Tarea 4: Definición de los requerimientos de interfaces.

Actividad 3: Gestión de riesgos.

Tarea 1: Definición de los supuestos del proyecto.

Tarea 2: Definición de las dependencias del proyecto.

Tarea 3: Definición de las restricciones del proyecto.

Tarea 4: Definición de los riesgos del proyecto.

Milestone: **Diseño preliminar de la solución.**

Actividad 1: Arquitectura del sistema.

Tarea 1: Diseño del esquema general del sistema.

Tarea 2: Descripción de los componentes necesarios

Tarea 3: Definición de la matriz de requisitos funcionlaes y componentes.

Tarea 4: Establecimiento de los diagramas de contexto.

Tarea 5: Diseño del diagrama de arquitectura del sistema.

Tarea 6: Descripción general de los módulos del sistema.

Milestone: **Diseño de módulos del sistema.**

Actividad 1: Diseño de módulo [M1]: Conexión eléctrica.

Tarea 1: Establecimiento de restricciones de instalación.

Tarea 2: Establecimiento de medidas de seguridad para instalación.

Tarea 3: Cálculo de de consumo energético de ESP8266.

Tarea 4: Cálculo de gasto energético de ESP8266.

Tarea 5: Cálculo de de consumo energético de *Electronic Rotary Latch*.

Tarea 6: Cálculo de gasto energético de *Electronic Rotary Latch*.

Tarea 7: Cálculo de de consumo energético de router de EB.

Tarea 8: Cálculo de gasto energético de router de EB.

Actividad 2: Diseño de módulo [M2]: Conexión a internet.

Tarea 1: Establecimiento de restricciones de instalación.

Tarea 2: Establecimiento de tamaño mínimo de paquetes transferidos por MQTT.

Tarea 3: Establecimiento de tamaño de paquetes transferidos por la EB.

Tarea 4: Cálculo de volumen de datos transferidos por EB.

Tarea 5: Cálculo de volumen de datos mensual transferidos por EB.

Actividad 3: Diseño de módulo [M3]: Servicios Web.

Tarea 1: Descripción de *Landing Page*.

Tarea 2: Descripción de página de registro.

Tarea 3: Descripción de página de ”pedir estacionamiento”.

Tarea 4: Descripción de página de administrador.

Tarea 5: Descripción de *App server*.

Actividad 4: Diseño de módulo [M4]: Base de datos.

Tarea 1: Establecimiento de entidades del sistema.

Tarea 2: Establecimiento de transacciones del sistema.

Tarea 3: Diseño de campos de tabla para EB.

Tarea 4: Diseño de campos de tabla para PDE.

Tarea 5: Diseño de campos de tabla para usuario.

Tarea 6: Diseño de campos de tabla para administrador.

Tarea 7: Diseño de tabla para transacciones usuario-PDE.

Tarea 8: Diseño de tabla para transacciones de administrador-EB.

Actividad 5: Diseño de módulo [M5]: ETL.

Tarea 1: Establecimiento de los *job* que el ETL debe ejecutar.

Tarea 2: Diseño de *job*: Registrar al usuario en el sistema.

Tarea 3: Diseño de *job*: Permitir al usuario pedir un estacionamiento.

Tarea 4: Diseño de *job*: Permitir al usuario retirar su bicicleta.

Tarea 5: Diseño de *job*: Permitir al administrador registrar al usuario.

Tarea 6: Diseño de *job*: Permitir al administrador retirar bicicleta de una PDE.

Actividad 6: Diseño de módulo [M6]: Protocolo de comunicación entre EB y servidor.

Tarea 1: Establecimiento de protocolo de comunicación.

Tarea 2: Establecimiento de broker público a utilizar.

Tarea 3: Establecimiento de medida de seguridad de capa transporte.

Tarea 4: Establecimiento de tópicos a utilizar.

Actividad 7: Diseño de módulo [M7]: Electrónica para sistema de anclaje.

Tarea 1: Establecimiento de protocolo de comunicación EB-PDE.

Tarea 2: Establecimiento de dispositivo de cierre electrónico, *Electronic Rotary Latch*.

Tarea 3: Establecimiento de IoTD para comunicación EB-Servidor.

Tarea 4: Establecimiento de IoTD para comunicación EB-PDE.

Actividad 8: Diseño de módulo [M8]: Lógica de acción de estacionamientos.

Tarea 1: Establecimiento de los módulos necesarios para accionamiento de estacionamientos.

Tarea 2: Establecimiento de PRNG y su funcionamiento.

Tarea 3: Diseño de sistema criptográfico simétrico.

Tarea 4: Diseño de lógica de acción en EB y PDE.

Tarea 5: Diseño de rutina de selección de ”llaves de instrucción” (LDI) seguras.

Tarea 6: Diseño de rutina de acuerdo de llaves privadas (PK) para instalación de EB nueva.

Tarea 7: Diseño de rutina de acuerdo de LDI para instalación de PDE nueva.

Tarea 8: Diseño de rutina de preconfiguración de EB.

Tarea 9: Diseño de rutina de preconfiguración de PDE.

Tarea 10: Diseño de rutina de apertura de estacionamiento.

Tarea 11: Diseño de rutina de cambio de PK y LDI.

Tarea 12: Diseño de rutina de generación de número 2FA.

Tarea 13: Diseño de rutina de cambio de 2FA en EB y servidor.

Actividad 9: Diseño de módulo [M9]: Sistema de verificación de presencia (SVP) de usuario en estacionamiento.

Tarea 1: Establecimiento de hardware necesario para SVP.

Tarea 2: Establecimiento de funcionamiento de SVP.

Tarea 3: Establecimiento de forma de uso de SVP.

Tarea 4: Establecimiento de caso de uso 1: verificación de presencia de usuario en estacionamiento.

Tarea 5: Establecimiento de caso de uso 2: identificación de estacionamiento que el usuario pretende utilizar.

Actividad 10: Diseño de módulo [M10]: Sistema de respaldo de base de datos.

Tarea 1: Determinación de factibilidad de aplicación mediante servicio RDS5 de *Amazon Web Services*.

Milestone: **Diseño de interfaces.**

Actividad 1: Diseño de interfaces para usuario.

Tarea 1: Diseño de *mockup* para registro de usuario.

Tarea 2: Diseño de *mockup* para pedir estacionamiento, inicio de sesión.

Tarea 3: Diseño de *mockup* para pedir estacionamiento, selecciónar estacionamiento.

Tarea 4: Diseño de *mockup* para retirar bicicleta.

Actividad 2: Diseño de interfaces para administrador.

Tarea 1: Diseño de *mockup* para inicio de sesión de administrador.

Tarea 2: Diseño de *mockup* para autorizar usuario.

Tarea 3: Diseño de *mockup* para abrir PDE.

4.1.2. Revisión del diseño

4.1.2.1. Supuestos generales de diseño del sistema

En general, el diseño del sistema esta basado en los siguientes supuestos:

- El modelo de negocio está basado en la venta de un servicio de estacionamiento para bicicletas.
- El servicio incluye el diseño, la construcción, la operación y el mantenimiento de los estacionamientos para bicicletas.
- El cliente es quien paga por el servicio, siendo el ciclista el usuario del sistema, i.e. no se requiere que el ciclista pague por el servicio.
- El cliente cuenta con un espacio para instalar el estacionamiento, con requisitos mínimos como un terreno pavimentado y acceso a electricidad.
- El servicio de WiFi móvil de algún ISP está disponible en el lugar de instalación.
- Si el cliente requiere instalar en un subterráneo se deben hacer dos cosas:
    1. Sugerirle que no lo haga, debido a la incomodidad que le significa al ciclista subir y bajar la bicicleta de nivel.
    2. Si aún así necesita hacerlo, se debe establecer el largo del cable Ethernet RJ45 necesario para conectar el router que estará en el exterior con el router de la EB.
- Se utiliza el *Electronic Rotary Latch* para el cierre del estacionamiento.
- Cada estacionamiento cuenta con un administrador. Esto quiere decir que habrá una sola cuenta de administrador por estacionamiento, pero la cuenta la pueden utilizar diversas personas.
- Cada administrador cuenta con acceso a internet ya sea desde su celular o desde un computador.

- Las transacciones registradas en la base de datos solo indican el nombre de cuenta de administrador, i.e. si varias personas son administradores de estacionamiento, las transacciones solo se registran con la cuenta de usuario de administrador.
- Tanto las EB como la PDE están correctamente configuradas desde fábrica.

4.1.2.2. Requisitos mínimos para el prototipo funcional

En la sección "Diseño preliminar de la solución" se establecen los parámetros necesarios para la puesta en marcha del servicio. Muchos de ellos son necesarios para un sistema productivo, pero no resultan necesarios para propósitos de prototipo. Se necesita entonces detallar los requisitos mínimos para implementar el prototipo funcional del estacionamiento, basado en los módulos diseñados en la sección 3.2:

M1: Conexión eléctrica:

No es necesario intervenir un enchufe. Dicha tarea es, a priori, sencilla del punto de vista de ejecución y no es requerida en el prototipo.

Para propósitos de prototipo:

Se utiliza la conexión eléctrica convencional. Para alimentar el NodeMCU de la EB se utiliza un cargador de celular típico con un cable micro usb clase B. Se Alimentan los NodeMCU de las PDE con los pines VCC y GND del NodeMCU de la EB.

Para simular el *Electronic Rotary Latch*, se utiliza un Relay que encienda o apague una ampolleta.

M2: Conexión a internet:

No es necesario contratar un servicio de WiFi móvil para propósitos de prototipo. Si bien la potencia de señal que reciben los router puede cambiar en cada lugar de instalación, se puede verificar a priori cual es dicha potencia comparándola con la del celular.

Para propósitos de prototipo:

Se utiliza una conexión WiFi disponible en el lugar donde esté el prototipo. El protocolo MQTT está diseñado para funcionar en ambientes de ancho de banda limitado.

M3: Servicios Web:

No es necesario implementar el *Landing page*.

Puede servir como un medio de difusión, pero primero se necesita tener la estructura física del estacionamiento, porque de otra forma no queda clara la propuesta del servicio.

Para propósitos de prototipo:

Se utilizan las páginas de:

- Registro.
- Pedir bicicleta.
- Retirar bicicleta.
- Administrador: *login*, autorizar usuario y abrir PDE.

Para el *App Server*, se utiliza un VPS gratuito, ya que permiten albergar páginas web y bases de datos de bajo ancho de banda.

M4: Base de datos:

Se necesita implementar todas las tablas.

Para propósitos de prototipo:

Las pruebas de concepto del prototipo requieren utilizar la relación entre distintas tablas, y el tiempo que requeriría evaluar cuales son los campos mínimos que se necesitan de cada tabla resulta contraproducente comparado con la demora de implementar la base de datos completa.

M5: ETL:

Se necesita implementar todos los *job*.

Para propósitos de prototipo:

Las pruebas de concepto necesitan de todos los *job* para funcionar. De nuevo, estimar el las transformaciones de información mínimas resulta contraproducente con el tiempo que toma implementar el prototipo.

M6: Protocolo de comunicación entre EB y servidor:

Se necesita utilizar un broker estático, es decir, que esté en la misma ”dirección web” siempre

Para propósitos de prototipo:

Se utiliza un broker público, debido a que existen algunos que permiten implementar TSL/SSL y acceso mediante nombre de usuario y contraseña, que son los requerimientos de seguridad de este protocolo. Permiten además conectar hasta 10 dispositivos, con lo que se podría simular el funcionamiento con múltiples clientes a la vez, e.g. se podrían implementar de manera ficticia 10 EB.

M7: Electrónica para sistema de anclaje:

Las pruebas de concepto que se necesitan desarrollar son relativas a testear la lógica de acción de los estacionamientos, i.e. cuando se ejecuta la orden de abrir en la página web, se prende o se apaga el relay de prototipo. No es necesario tener entonces el *Electronic Rotary Latch* para concepto de prototipo, porque puede ser reemplazado con una ampolleta.

Para propósitos de prototipo:

Se necesitan los siguientes componentes:

- NodeMCU: Se necesita uno para cada EB y para cada PDE.
- Relay: Se necesita uno por cada PDE.

- Ampolleta: Se necesita una por cada PDE.
- Cable: Se necesita un par por cada PDE, i.e. uno para linea y otro para neutro.
- Soquete macho y hembra: Se necesita uno de cada uno para cada PDE.
- Cargador de celular: Se necesita uno por cada EB.
- Jumpers.
- Protoboard.

M8: Lógica de acción de estacionamientos:

Este es la principal innovación presentada en este trabajo de memoria. Durante el desarrollo de este trabajo se diseñó un PRNG pensado para cumplir con los requisitos de bajo costo computacional y facilidad de implementación. Por razones de protección intelectual no se expresa la fórmula explícitamente.

Para propósitos de prototipo:

Se necesita implementar todo lo referente a este módulo, de forma de comprobar en la práctica su funcionamiento.

M9: Sistema de verificación de presencia (SVP) de usuario en estacionamiento:

Se necesita implementar este módulo de forma íntegra para comprobar el funcionamiento del módulo [M8].

Para propósitos de prototipo:

Se utiliza una pantalla OLED SSD1306 [37] para simular la pantalla presente en la EB para el SVP.

M10: Sistema de respaldo de base de datos:

No es necesario para la implementación del prototipo.

Para propósitos de prototipo:

No se utiliza este módulo.

4.2. Revisión del prototipo

4.2.1. Contexto general del prototipo

El proyecto presentado en esta memoria responde a la necesidad tanto de clientes como de usuarios de contar con un estacionamiento para bicicletas.

La falta de estacionamientos seguros han generado una ola de robos de bicicletas en Santiago, llegando a la razón de 1 robo cada 5 minutos[41]. Lo paradójico es que en Chile todo edificio debe contar con 1 estacionamiento para bicicletas por cada 2 estacionamientos para autos[42] , pero menos de un tercio de los edificios están equipados con biciestacionamientos, y de aquellos, menos del 5 % cumplen con la norma[41]. A pesar de ello, quienes optan por la bicicleta como medio de transporte han estado sumando 510.569 viajes diarios por la ciudad de Santiago, posicionándola en segundo lugar a nivel latinoamericano después de Bogotá[43]. El ciclista continúa utilizando los pocos estacionamientos actuales, pero estos están pensados para corta estadía y no brindan la seguridad emocional que el ciclista requiere; mi bicicleta estará donde la dejé hasta que vuelva por ella.

Por el lado de los clientes, existen diversas entidades que muestran interés en fomentar el uso de la bicicleta. La iniciativa CoolPlaceToBike[39] es una competencia interempresas en que los ciclistas de cada institución suman puntos por kilómetro recorrido en bicicleta. La entidad con mayor puntaje se gana la instalación de un biciestacionamiento.

Inmobiliarias como Ralei Development Group[40] buscan la sustentabilidad mediante la instalación de estacionamiento para bicicletas o ascensores con generación de energía, por lo que podemos decir que existe una necesidad respectiva al uso de la bicicleta en el mercado, tanto de usuarios como de clientes.

Respecto a los estacionamientos en sí, como se explica en la sección 2.1, en Chile no existen estacionamientos como el presentado en este documento de memoria. El mejor acercamiento es el presentando por BiciLock, quienes entregan el servicio a em-

presas para que tengan estacionamientos para sus clientes. Las principales diferencias son:

- En BiciLock se necesita una tarjeta RFID. Esto quiere decir que, si el usuario pierde su tarjeta, también pierde acceso al estacionamiento. Debiese entonces tratar de recuperar dicha tarjeta, y para eso BiciLock debe tener diversos puntos distribuidos de forma que los usuarios puedan acceder al sistema de nuevo.

- El sistema de BiciLock protege de forma íntegra la bicicleta. No se ha optado por dicha capacidad debido a que el proyecto está pensado para ser implementado con clientes que cuentan con espacios para poner los estacionamientos en zonas seguras, e.g. empresas que cuentan con estacionamiento para autos y que necesitan un estacionamiento para bicicletas. Dichos entornos entregan las condiciones de seguridad que complementarán la función del estacionamiento; que la bicicleta esté en el lugar que la dejó el ciclista.

La diferencia es que BiciLock está enfocado a un uso comercial, mediante el arriendo de bici estacionamientos con fines publicitarios, lo que deja inmediatamente fuera los estacionamientos para los hogares de los ciclistas.

El servicio que se diseña en este documento de memoria, busca generar una solución escalable en términos de disponibilidad de servicio y eficiencia de espacios. Además, contempla todo lo que implica la implementación de un estacionamiento de forma apropiada en términos de diseño, construcción, operación y mantenimiento. Esto es lo que lo diferencia de los servicios actuales que solo buscan la venta del estacionamiento que necesitan candados.

En conclusión, se puede decir que mediante la implementación de bici estacionamientos en los lugares de trabajo y los hogares de los ciclistas, se puede conectar la ciudad a través de un medio de movilización sustentable como la bicicleta. De implementar la red de estacionamientos, se generarían beneficios tanto a los ciclistas como a los clientes, demostrando que el desarrollo sustentable es posible mediante la generación de valor para todos.

4.2.2. *Esquema general de componentes del prototipo*

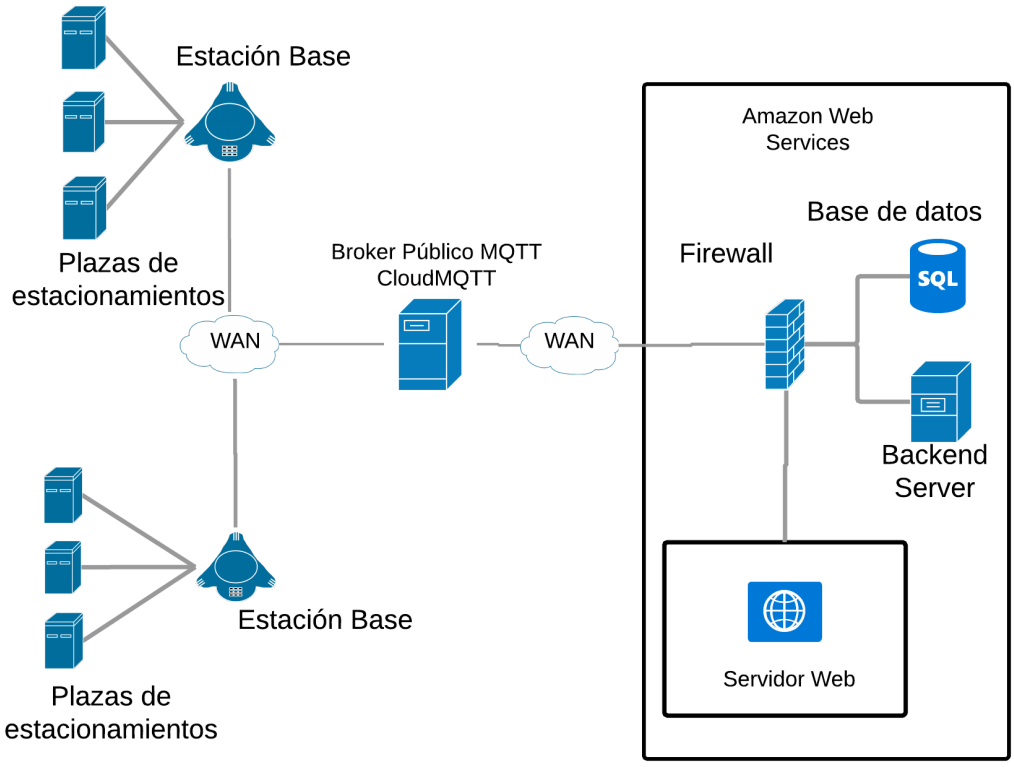

Fig. 4.1: Diagrama de arquitectura del sistema

4.2.3. *Descripción de componentes*

Para que el trabajo de prototipo funcional sea exportable a un prototipo comercial, se necesita utilizar:

1. Motor de base de datos: MySQL Community Edition[45].

   Se escoge este motor de base de datos debido a su rapidez y robustez comparado con otros motores, e.g. PostgreSQL o MariaDB. Permite además utilizar Python como lenguaje para ETL, mediante bibliotecas.

2. Lenguaje *Backend*: Python 3.6.5[44]:

La última versión de Python. Se escoge utilizar este lenguaje debido a:

- La facilidad de lectura de su sintaxis.
- La gran cantidad de bibliotecas que tiene disponible. Se utilizan bibliotecas que permiten extraer información desde MQTT y hacer los procesos de ETL a través de bibliotecas dedicadas a MySQL.
- La modularidad que necesita cada unos de los módulos diseñados en la sección 3.2 indica que se necesita un lenguaje que permita exportar e importar funciones entre distintos script.
- La versión 2.7 ya no tendrá soporte a partir de 2019. Las empresas que utilizan Python, como Facebook o Google, están utilizando en sus nuevos desarrollos Pyhton 3.6.5, lo que indica una buena práctica a seguir

3. *Web Development Framework*: Django 2.0.6 for Python[38].

   Como se utiliza Python como lenguaje *backend*, es necesario que el desarrollo WEB sea compatible. La utilización de Django permite:

   - Hacer un diseño web en base al formato *Model-View-Controller*, lo que permite un diseño modular de los servicios web, separando el modelo, la vista y el controlador de cada página, de forma que el cambio en uno no afecte a los demás.
   - Trabajar en entornos virtuales, por lo que si algún módulo del diseño web necesita más bibliotecas que otro, no se generaran problemas debido a que están separados.
   - Exportar fácilmente lo desarrollado a otro entorno, e.g. servidor nuevo, mediante el uso de paquetes instalables mediante linea de comandos.
   - Modificar las tablas de la base de datos MySQL mediante la modificación de los Modelos del MVC. Al modificar los modelos del MVC, y mediante la linea de comando, Django modifica las tablas correspondientes a los modelos de forma automatica.

- Prototipar de forma rápida con un *WebServer* propio de Django, evitando el uso del servidor en producción o el uso de *sandbox*.

4. *Virtual Private Server* gratuito: *Amazon Web Services*:

   Se escoge utilizar AWS por el respaldo que entrega trabajar con una compañía de su categoría; lider mundial en ventas online y servicios de virtualización web. Como se explica en el diseño del módulo [M3], su SLA es del 99.99 %, suficiente para certificar la disponibilidad del servicio.

   La distribución de sus datacenter[46] permite un acceso rápido desde muchos puntos del planeta, y su infraestructura asegura que el servicio está protegido de riesgos.

   Además, permite hacer uso de ”computación elástica”; en la medida que se necesiten más recursos, Amazon informa automáticamente al administrador de sistema para que pueda optar a aumentar la capacidad del servicio, tanto en ancho de banda como en almacenamiento.

   Permite además usar una cuenta gratuita bajo ciertas restricciones, lo que permite utilizar sus servicios con propósito de prototipar.

5. *Integrated Development Environment* (IDE): Arduino IDE 1.8.5 [47]:

   Necesario para programar los NodeMCU de las EB y de las PDE. Se utiliza porque permite integrar bibliotecas fácilmente, además de su vasta compatibilidad con diversas placas de desarrollo que se pudiesen utiliza en un futuro.

6. Broker MQTT público: CloudMQTT.

   Se escoge por ser un servicio gratuito, que ofrece la posibilidad de conectar hasta 10 dispositivos mediante TSL/SSL, con nombre de usuario y contraseña, lo que es ideal para el prototipo.

7. Dispositivos IoT para EB y PDE: NodeMCU.

8. biblioteca especial para ”números grandes”: BigNumber by Nick Gammon [48]:

   Como se menciona en el diseño del módulo [M8], el PRNG genera un número de muchos dígitos. De los tipos de datos que existen, el que es capaz de albergar el mayor número es un *string*, con hasta 2 millones de caracteres, ergo, 2 millones de dígitos. El problema es que no permiten utilizar su información como un número, i.e. no se pueden hacer operaciones matemáticas con ellos. El que si tiene esa capacidad es el *int*, con capacidad de aproximadamente 15 decimales, lo que sigue siendo muy poco para los propósitos del PRNG. Es por ello que se utiliza una biblioteca que permita obtener números de una precisión arbitraria, tanto en decimales como en dígitos enteros. Se utiliza la biblioteca *BigNumber* de Nick Gammon [48] para dicho propósito.

4.2.4. Interfaces implementadas en el prototipo

Las interfaces implementadas en el prototipo son las que se presentan en la sección 3.3. Todas estas interfaces son necesarias para las pruebas de concepto.

El diseño *frontend*[49] en términos de *user interface* y *user experience* [50] no se desarrolla en el prototipo, debido a que requiere tiempo y habilidades. Dicho trabajo puede ser delegado a un funcionario con las capacidades necesarias, o puede ser comprado a un desarrollador web profesional.

4.3. Verificación y validación

Para la verificación y validación, se puede descomponer los requerimientos en seguridad y disponibilidad:

4.3.1. Verificación de la seguridad del estacionamiento

La seguridad incluye la seguridad de la bicicleta y la seguridad del sistema. En el primer caso, la seguridad viene dada por el diseño que se escoja para el estacionamiento. Hay que comprender entonces cuales son la cosas que el ciclista desea proteger de

su bicicleta:

- Marco:

  Es lo más caro de una bicicleta. Proteger el marco significa que la bicicleta debe quedar protegida de magulladuras o golpes.

- Ruedas:

  En las bicicletas actuales, desmontar las ruedas es relativamente sencillo, pues la mayoría incluye sistemas que permiten hacerlo de forma fácil. En las ruedas estan incluidos los piñones para cambios y en ocasiones los discos de los frenos.

- Componentes:

  Incluye los sistemas de freno, luces, manubrio, sistema de cambios y sillín, entre otros componentes. Muchos de ellos son fáciles de desmontar, por lo que los ciclistas prefieren, por ejemplo, sacar el sillín para evitar robos.

- Casco:

  Por norma, los ciclistas deben utilizar casco para circular en la vía pública. Si bien este componente se puede portar, el ideal es que el estacionamiento cuente con un lugar donde albergar los cascos.

4.3.2. Validación de la seguridad del estacionamiento

De todos los componentes mencionados, el sistema acá desarrollado permite proteger el marco de la bicicleta de forma que sea muy difícil abrir los estacionamientos mediante fuerza o ataques informáticos. No se trabaja con la seguridad del resto de los componentes porque la solución planteada está orientada a implementarse en lugares que entregan ciertos factores de seguridad, e.g. el estacionamiento de un edificio, donde existe control de acceso y ciertos niveles de vigilancia por parte de personal contratado para ello. Es decir, el estacionamiento acá diseñado está pensado para uso

privado. Cuando el servicio se ofrezca para instalarlo en la vía pública, el diseño estructural debe estar preparado para proteger la bicicleta de forma íntegra, es decir, cubriendola por completo, como lo hace BiciLock.

4.3.3. Verificación de la seguridad del sistema

De la seguridad del sistema, se puede traducir en cuatro factores:

- Confidencialidad.

  Se refiere a que la información que maneja el sistema debe estar lejos del alcance de terceros no autorizados. Mediante la implementación de diversas medidas de seguridad se puede fortalecer la confidencialidad del sistema, como control de acceso a la base de datos mediante listas de permisos, firewalls y servicios con seguridad reforzada como lo entregado por *Amazon Web Services*.

- Integridad.

  Se refiere a mantener la información libre de modificaciones no autorizadas que podrían perjudicar el servicio. La integridad es de máxima necesidad sobre todo para la comunicación entre EB y servidor, debido a que está encriptada. Si se llegase a perder la coordinación entre EB y servidor, se debería aplicar soluciones manuales, lo que significa ir directamente al estacionamiento a arreglar los IoTD.

- Disponibilidad.

  Se refiere a que la información debe estar disponible en todo momento. La importancia de este aspecto radica en que para el uso del sistema se necesita que la información esté disponible para abrir un estacionamiento, o retirar una bicicleta o para que el administrador de estacionamiento pueda autorizar a algún ciclista. Existen momentos críticos, como las horas con mucho tráfico, en que se necesita que el servicio esté disponible.

- Autenticación.

  Se refiere a la posibilidad de identificar a quien emite cierta información. Es necesaria ya que el registro de los usuarios debe ser auténtico, para evitar usos maliciosos del sistema. Además, la EB debe autenticar las órdenes provenientes desde el servidor, y el servidor debe autenticar la órdenes provenientes desde la EB, de forma que no existan ataques MITM en que el atacante intente sacar provecho de su posición para operar los estacionamientos.

4.3.4. Validación de la seguridad del sistema

De los cuatro puntos anteriores, se puede decir que se cumple con todos. La confidencialidad está dada por proteger el acceso a la base de datos, para lo cual ya existen soluciones respectivas. Para el caso de este proyecto, se busca lograr confidencialidad e integridad mediante el uso de un firewall y una lista de acceso suficientemente restringida a solo lo necesario. Existen otras alternativas como IPS o IDS, que buscan examinar los paquetes entrantes al servidor, pero su implementación se escapa de los propósitos de esta memoria. Cuando el sistema entre en producción, se deben tener en cuenta esos requisitos. De la disponibilidad, se puede mencionar que la elección del servicio EC2 de *Amazon Web Services* entrega las herramientas necesarias para su aseguramiento en términos de disponibilidad de servicios web. El diseño del sistema permite certificar además dicha disponibilidad mediante el uso de un protocolo de alta fidelidad como MQTT, el cual está pensado para trabajar en condiciones poco favorables en términos de ancho de banda.

Por otro lado, la autenticación viene dada por el uso de diversos mecanismos. Si bien cualquier usuario se puede registrar en el servicio, es el administrador del estacionamiento el que debe certificar que dicho usuario está habilitado para su uso. El diseño está pensado para dejar en manos de esta persona que el usuario es quien dice ser, pues como debe ingresar el rut del usuario, le es fácil verificar que quien pide la autorización está correctamente registrado si es que, por ejemplo, le pide su carnet de

identidad. De esta forma, cada cliente fija sus políticas de uso del estacionamiento, por lo que puede decidir quien puede o no usar el estacionamiento. La autenticación también se logra mediante el sistema de criptografía simétrica. Las rutinas implementadas en el diseño del módulo [M8] están pensadas para que cuando se genera un cambio en la información disponible tanto en EB como servidor se necesite el uso de ”secretos compartidos” entre ambas partes, de forma que si un atacante intenta algo, sea muy improbable que pase la fase de autenticación.

4.3.5. Verificación de disponibilidad del estacionamiento y del sistema

La disponibilidad se refiere a la capacidad que presenta el servicio para estar listo para su uso cuando se necesite. Entre los puntos importantes a mencionar:

- Disponibilidad de la mecánica del estacionamiento:

  Los estacionamientos estarán sometidos a un uso constante, por lo que se debe contemplar su mantenimiento mecánico.

- Disponibilidad de la electrónica de los estacionamientos:

  La electrónica debe estar disponible pues es a partir de ella que se operan los estacionamientos. Los dispositivos que se usen deben estar preparados para recibir órdenes en todo momento.

- Disponibilidad de conexión a internet:

  Necesaria para la comunicación EB-servidor. Se deben contratar planes que entreguen un ancho de banda adecuado el volumen de datos necesarios.

- Disponibilidad de conexión eléctrica:

  Esencial para el funcionamiento, pues sin electricidad no hay comunicación entre PDE, EB y servidor.

- Disponibilidad de los servicios web:

Viene fundamentalmente dado por la calidad del servicio que entrega el proveedor del VPS. Los servicios web son parte del núcleo importante de necesidades del sistema.

4.3.6. Validación de la disponibilidad del estacionamiento y del sistema

La disponibilidad viene entonces dada por la entrega de un servicio integral. Es por eso que se debe contemplar un todo, que incluya el diseño, la construcción, la operación y el mantenimiento del servicio.

Para el caso de la disponibilidad mecánica se debe diseñar un plan de mantenimiento acorde a las necesidades. El diseño del estacionamiento debe estar pensado en incluir la menor cantidad de partes mecánicas, y que las partes mecánicas sean suficientemente robustas para aguantar el uso y los embates de usuarios mal intencionados.

La disponibilidad de los dispositivos electrónicos viene dada por la elección de valores adecuados para su operación, como las fuentes de voltaje adecuadas para cada parte. El desgaste de los dispositivos no es un riesgo, pero sí deben estar protegidos mediante un diseño estructural adecuado, que lo aísle de factores ambientales y sociales.

La disponibilidad de la conexión a internet está dada por la elección de un ISP con estándares adecuados de servicio. En general, los ISP de Chile presentan buenas características, ya que el Ministerio de Transporte y Telecomunicaciones vela constantemente porque los servicios sean los adecuados. La mayoría de los planes de datos disponibles en el mercado son suficientes para la comunicación, como queda consignado en el diseño del módulo [M2]. La preocupación viene dada entonces por la disponibilidad de conexión en el lugar donde se implementa el estacionamiento, puesto que se requiere internet para la comunicación EB-servidor y para el usuario a la hora de iniciar sesión. Se debe procurar contar con una buena conexión en el lugar de instalación.

La disponibilidad de conexión eléctrica viene dada por la elección del lugar de

instalación. Se debe tener especial cuidado con el lugar desde donde se toma la corriente, puesto que debe ser un flujo continuo de energía. Se debe informar al cliente que se debe intervenir un enchufe para la conexión, puesto que la desconexión del estacionamiento debe estar imposibilitado. Muchos edificios cuentan con sistemas de respaldo, puesto que no pueden quedarse sin servicios como ascensores o luz en los pasillos. De todas formas, si es que llegase a faltar energía eléctrica, no se podrían utilizar los estacionamientos, pero tampoco se abrirían con la falta de ella debido a la elección del *Electronic Rotary Latch*.

La disponibilidad de los servicios web viene dada por la elección de AWS como proveedor de VPS. El servicio entregado por AWS permite disponer de mejor calidad de servicio en cuanto a ancho de banda y capacidad de almacenamiento se refiere, y en la medida que estos requerimientos cambian se puede pedir mayor capacidad sin interrumpir el funcionamiento del sistema.

4.4. Plan de testing

El plan de testing está pensado para poner al sistema en situaciones que puedan comprometer el servicio, y observar si el diseño de los módulos es el adecuado. Para ejecutar el plan de testing, se deben probar los módulos de forma aislada, y luego como un todo. Además, se debe probar el sistema en contra de ataques, pensando que el atacante conoce como se implementó el sistema, pero sin tener las credenciales adecuadas para operar los estacionamientos.

4.4.1. Prueba de concepto 1

La EB sincroniza de forma automática las PK y LDI al conectarse a la red:

4.4.1.1. Prerrequisitos de prueba de concepto 1

- PK iniciales, PK nuevas y LDI escogidas y guardadas en la base de datos.
- EB configurada:

- Introducir el SSID y password del WiFi.
- Configurar las PK iniciales.
- Inicializar 3 variables en 0 para las PK nuevas.
- Establecer la LDI de acción, actualización de llave.
- Establecer la LDI de identidad de la EB.
- Establecer las LDI de identidad de las PDE.
- Establecer las LDI de acción de las PDE. Se deben inicializar en 0.
- Establecer la LDI de acción, cambio de PK b.
- Establecer la LDI de acción, cambio de PK n.
- Establecer la LDI de acción, cambio de PK μ.
- Establecer la LDI de acción, cambio de PK p.
- Establecer la LDI de acción, cambio de PK i.
- Establecer la LDI de acción, cambio de PK j.
- Configurar para publicar en tópico 0, mediante el uso de PRNG y las PK iniciales.
- Configurar para suscribirse a tópico 1, mediante el uso de PRNG y su MAC.
- Direcciones I2C de las PDE.

- Conocer el MAC de la EB objetivo.
- Tablas de base de datos implementadas y vacias.
- Suscribirse manualmente al tópico 0.
- Suscribirse manualmente al tópico 1.

4.4.1.2. Pasos para prueba de concepto 1

1. Conectar la EB a la energía eléctrica .
2. Enviar un mensaje de forma manual al tópico 1, que contenga la LDI de identidad de una PDE, la LDI de acción y el número generado por el PRNG y las PK_1 de la tabla EB-PK.
3. Si el IoTD que representa la PDE ejerce la función ”abrir estacionamiento’, la prueba ha sido exitosa.
4. Si no ha sido una prueba exitosa, se debe revisar lo publicado en el tópico 1 y 0 para revisar los errores. Se debe decodificar los mensajes y seguir los pasos presentes en el diseño de la rutina de intercambio de llaves.

4.4.2. Prueba de concepto 2

El servidor intercambia llaves de manera correcta con la EB.

4.4.2.1. Prerrequisitos de prueba de concepto 2

- Prueba de concepto 1 exitosa.
- Suscribirse a los tópicos 0 y 1.
- Pantalla conectada para visualizar las PK y LDI.

4.4.2.2. Pasos para prueba de concepto 2

1. Se deben extraer manualmente las PK_1 de la tabla EB-PK.
2. Se debe generar el número con las PK_1 y el PRNG.
3. Se debe escoger la LDI de cambio de llave de la EB.
4. Se debe escoger la LDI de identidad de una PDE.

5. Se debe escoger una nueva LDI de acción para esa PDE.

6. Se debe publicar manualmente en el tópico 1 el mensaje ”cambio de llave” de PDE mediante el uso de lo obtenido en el paso 2, 3 y 4.

7. Se debe publicar manualmente en el tópico 1 el mensaje ”apertura de estacionamiento” mediante el uso de lo obtenido en el paso 2, 4 y 6.

8. Si la PDE no ejerce la función ”abrir estacionamiento”, se debe revisar el programa de la EB para verificar que haya cambio de llave.

9. Si la PDE ejerce la función, se continua con el siguiente paso.

10. Se debe procurar hacer todas las combinaciones de cambio posibles, como cambiar LDI de acción de EB o LDI de identidad de PDE.

11. Una vez concretado lo anterior, la prueba de concepto 2 habrá sido exitosa.

4.4.3. Prueba de concepto 3

El sistema de verificación de presencia (SVP) de usuario en estacionamiento funciona correctamente.

4.4.3.1. Prerrequisitos de prueba de concepto 3

- Prueba de concepto 1 exitosa.
- Prueba de concepto 2 exitosa.
- Interfaz ”pedir estacionamiento”.
- Pantalla para visualizar la PK y el valor de i.
- Tener una cuenta de usuario registrada y autorizada para utilizar la EB.

4.4.3.2. Pasos para prueba de concepto 3

1. Se inicia sesión con la cuenta de usuario autorizada.
2. Se ingresa el nombre de usuario, contraseña y código 2FA.
3. Si existe un error respectivo de nombre de usuario, contraseña o permisos, se debe revisar si se ingresaron las credenciales correctamente.
4. Si el error está relacionado con el 2FA, se debe ver en la base de datos si es que el 2FA, el i y las PK coinciden con las de la EB.
5. Si no existe error, se escoge un estacionamiento.
6. Si la PDE responde adecuadamente, se continúa con los siguientes pasos de la prueba de concepto.
7. Si el código 2FA no cambia, se detiene la prueba. El problema puede estar en la generación del 2FA por parte del IoTD de la EB.
8. Si el código 2FA cambia, se empieza desde el paso 1 y se vuelve a comprobar.
9. Si existe error, se debe revisar que la base de datos esté actualizando el 2FA de la EB.
10. En caso contrario, la prueba de concepto 3 ha sido exitosa.

4.4.4. Prueba de concepto 4

El usuario se puede registrar correctamente.

4.4.4.1. Prerrequisitos de prueba de concepto 4

- Prueba de concepto 1 exitosa.
- Prueba de concepto 2 exitosa.

- Prueba de concepto 3 exitosa.
- Interfaz registro de usuario.
- Tabla de usuario vacía.

4.4.4.2. Pasos para prueba de concepto 4

1. Se debe registrar un usuario nuevo.
2. Si existe error de ”usuario ya registrado” se debe revisar el *job* de registro de usuario y las conexiones a la base de datos.
3. Si no existe error, se debe revisar que en la tabla de usuario aparezca el usuario nuevo, si no aparece, se debe revisar el *job* de registro de usuario.
4. Si el usuario aparece, se procede al siguiente paso.
5. Se debe registrar un usuario con el mismo nombre de usuario que el primero.
6. Se debe registrar un usuario con el mismo rut que el primero.
7. Se debe registrar un usuario con el mismo correo que el primero.
8. Si no existe error en alguno de los tres pasos anteriores, se debe revisar el *job* de registro de usuario.
9. Si existe error en alguno, se debe registrar un usuario con nombre de usuario, rut y correo distintos a los del primero.
10. Si se logra registrar, se procede al siguiente paso.
11. Se debe intentar pedir una PDE de la EB.
12. Si se logra pedir, se debe revisar la tabla Administrador-EB para ver que no haya errores.
13. Si no logra pedir una PDE, la prueba de concepto 4 ha sido exitosa.

4.4.5. Prueba de concepto 5

El administrador puede autorizar un usuario nuevo de forma correcta.

4.4.5.1. Prerrequisitos de prueba de concepto 5

- Prueba de concepto 1 exitosa.
- Prueba de concepto 2 exitosa.
- Prueba de concepto 3 exitosa.
- Prueba de concepto 4 exitosa.
- Tener un administrador registrado con la EB que se utilizará de prueba.
- Interfaz de administrador.
- Interfaz de usuario.
- Rut de un usuario exitosamente registrado.

4.4.5.2. Pasos para prueba de concepto 5

1. Se inicia sesión con la cuenta de administrador.
2. Se intenta la autorización de un rut no registrado.
3. Si no existe error, se debe revisar el *job* de ”permitir al administrador registrar usuario”.
4. Si existe error, se procede al siguiente paso.
5. Se intenta autorizar un rut ya registrado.
6. Si existe error, se debe revisar el *job* de ”permitir al administrador registrar usuario”.

7. Si no existe error, se procede el siguiente paso.

8. Se debe intentar pedir una PDE con el rut registrado y autorizado.

9. Si existe error, se debe revisar la tabla para transacción Administrador-EB y el *job* de ”permitir al administrador registrar usuario”.

10. Si no existe error y se abre la PDE, la prueba de concepto 5 ha sido exitosa.

4.4.6. Prueba de concepto 6

El administrador puede abrir una PDE de forma correcta.

4.4.6.1. Prerrequisitos de prueba de concepto 6

- Prueba de concepto 1 exitosa.
- Prueba de concepto 2 exitosa.
- Prueba de concepto 3 exitosa.
- Prueba de concepto 4 exitosa.
- Prueba de concepto 5 exitosa.
- Tener un administrador registrado con la EB que se utilizará de prueba y sin permiso de abrir las PDE, es decir, el campo PDA de su tabla debe ser 0.
- Interfaz de administrador.
- Interfaz de ”abrir PDE” de administrador.

4.4.6.2. Pasos para prueba de concepto 6

1. Se debe iniciar sesión como administrador y navegar hasta la sección abrir estacionamiento.

2. Se debe intentar abrir una PDE.

3. Si se logra, se debe revisar el *job* de ”permitir al administrador retirar bicicleta de una PDE”.

4. Si no se logra, se sigue con el paso siguiente.

5. Se debe modificar el campo PDA del administrador a 1.

6. Se debe intentar abrir una PDE.

7. Si no se logra, se debe revisar el *job* de ”permitir al administrador retirar bicicleta de una PDE”.

8. Si se logra, la prueba de concepto 6 ha sido exitosa.

4.4.7. Prueba de concepto 7

Se puede cambiar una PDE de forma exitosa.

4.4.7.1. Prerrequisitos de prueba de concepto 7

- Prueba de concepto 1 exitosa.
- Prueba de concepto 2 exitosa.
- Prueba de concepto 3 exitosa.
- Prueba de concepto 4 exitosa.
- Prueba de concepto 5 exitosa.
- Prueba de concepto 6 exitosa.
- Se debe tener una EB conectada a una PDE que se vaya a cambiar.
- La PDE nueva debe estar configurada con la misma LDI de identidad y de acción de la que se está cambiando.
- Usuario registrado y autorizado para utilizar el estacionamiento.

- Administrador registrado en el estacionamiento y autorizado para abrir PDE.

4.4.7.2. Pasos para prueba de concepto 7

1. Se debe reemplazar la PDE antigua por la nueva.
2. Se debe iniciar sesión con usuario registrado.
3. Se debe pedir la PDE reemplazada.
4. Si no se abre, se debe revisar el programa de la PDE reemplazada, verificando que las LDI coincidan.
5. Si se abre, se procede al siguiente paso.
6. Se debe iniciar sesión con cuenta de administrador.
7. Se debe acceder a la sección de ”abrir PDE”.
8. Se debe intentar abrir la PDE reemplazada.
9. Si no se abre, se debe revisar el programa de la PDE reemplazada.
10. Si se abre, la prueba de concepto 7 ha sido exitosa.

4.4.8. Prueba de concepto 8

Se puede cambiar una EB de forma exitosa.

4.4.8.1. Prerrequisitos de prueba de concepto 8

- Prueba de concepto 1 exitosa.
- Prueba de concepto 2 exitosa.
- Prueba de concepto 3 exitosa.
- Prueba de concepto 4 exitosa.

- Prueba de concepto 5 exitosa.
- Prueba de concepto 6 exitosa.
- Prueba de concepto 7 exitosa.
- Se debe configurar una nueva EB, con la configuración de la EB que se está reemplazando.
- Se debe cambiar el permiso de administrador para administrar la EB nueva.
- Se deben agregar a la tabla administrador-eb los usuarios antiguos con la ID de EB nueva.

4.4.8.2. Pasos para prueba de concepto 8

1. Se reemplaza la EB.
2. Se inicia sesión como administrador.
3. Se intenta abrir una PDE.
4. Si hay error, se debe revisar al configuración de la EB.
5. Si no hay error, se procede al siguiente paso.
6. Se inicia sesión con cuenta de usuario.
7. Se intenta abrir una PDE.
8. Si hay error, se debe revisar al configuración de la EB.
9. Si no hay error, la prueba de concepto 8 ha sido exitosa.

4.4.9. Prueba de concepto 9

La conexión a internet entrega un buen ancho de banda en el lugar de instalación.

4.4.9.1. Prerrequisitos de prueba de concepto 9

- Se debe estar en el lugar donde se instalará el estacionamiento para bicicletas.
- Se debe tener un teléfono celular.
- Se debe tener una tarjeta SIM de la misma compañía que entrega el servicio de internet.
- La tarjeta SIM debe tener saldo para navegar en internet en la misma red del WiFi móvil.
- Se debe descargar la aplicación *SpeedTest*.
- Hacer esta prueba de concepto en las horas de mayor tráfico.

4.4.9.2. Pasos para prueba de concepto 9

- Abrir la aplicación *SpeedTest*.
- Medir la velocidad de descarga y carga.
- Si alguna de las dos está por debajo de 1 MB/s, se debe cambiar la locación del estacionamiento a un lugar que cuente con dicha velocidad.

El objetivo de esta prueba de concepto es conocer el estado de la red in-situ. En general, las compañías entregan un servicio que fácilmente supera la velocidad que se describe en los pasos para la prueba de concepto.

4.5. Implementación de módulos del sistema

Se debe identificar cuales módulos son los más críticos a implementar. Por ejemplo, si bien el sistema de registro e inicio de sesión son fundamentales, su desarrollo es largo pero es un proceso estandarizado. Es por ello que se identifican cuales son los módulos que significan un profundo impacto en términos de innovación. Estos son:

- Generador de números pseudo-aleatorios.
- Método criptográfico simétrico.
- Autenticación de dos pasos 2FA.
- Message Authentication Code (MAC).

4.5.1. Generador de números pseudo-aleatorios

El objetivo del desarrollo experimental de este módulo es comprobar la factibilidad de implementación en los IoTD. Esto es debido a que el método criptográfico requiere utilizar números de gran cantidad de dígitos, y los dispositivos utilizan los siguientes tipos de datos:

Tipo	Tamaño	Rango de valores
char	8 bits	Desde -128 hasta 127, o 0 hasta 255
unsigned int	16 bits	Desde 0 hasta 65535
unsigned long	32 bits	Desde 0 hasta 4,294,967,295

Tab. 4.1: Valores máximos para distintos tipos de datos [66].

En el mejor de los casos, al utilizar un *unsigned long* se puede obtener como máximo 4294967295, lo que se contrapone con la idea de utilizar un número con gran cantidad de dígitos, como se expone en la sección 3.2.8.2. Para superar este contratiempo es que se deben utilizar bibliotecas *BigNumber*[67], que permitan especificar una precisión arbitraria tanto en decimales como en dígitos enteros.

Dicho aquello, se debe proceder a implementar la biblioteca en un IoTD para verificar su funcionamiento, mediante la comparación con un resultado esperado, i.e. utilizar otra fuente desde la cual obtener un número a comparar por el generado por el IoTD.

Para ello se utiliza un script en algún lenguaje de programación que soporte *BigNumbers*, e.g. Python permite utilizar *BigNumbers* de forma nativa. Se utilizará el

script presente en el anexo A.

Dicho script incluye todas las variables que permiten utilizar el generador de números pseudoaleatorios PRNG.

Se debe implementar el PRNG en el IoTD mediante la biblioteca *BigNumber*, utilizar los mismos parámetros que en el script y comparar los resultados. Se utiliza un NodeMCU[26] como IoTD y el Sketch del anexo B.

```
1  def Mbn(b, n, mu, i, j,p):
2      M = Decimal(b)/Decimal(10**n-mu)
3      SCF = int((10**p)*M)
4      RCF = (SCF%10**(i+j)-SCF%10**i)/10 ** i
5      return RCF
6
7  from decimal import *
8
9  b = 1
10 n = 8
11 mu =  7
12 p = 200
13 i = 10
14 j = 90
15 getcontext().prec = p
16
17 print Mbn(b, n, mu, i, j,p)
18
```

```
PROBLEMS 34   OUTPUT   DEBUG CONSOLE   TERMINAL

MacBook-Air-de-Matias:python matiasbarrios$ python mbn.py
81698901718923120324618422723289590630271344118994088329586183071032814972297048060793364 2
```

Fig. 4.2: Resultado script Python

```
WiFi connected
Server started
192.168.1.17
privKey = 81698901718923120324618422723289590630271344118994088329586183071032814972297048060793364 2
privKey = 81698901718923120324618422723289590630271344118994088329586183071032814972297048060793364 2
privKey = 81698901718923120324618422723289590630271344118994088329586183071032814972297048060793364 2
privKey = 81698901718923120324618422723289590630271344118994088329586183071032814972297048060793364 2
```

Fig. 4.3: Resultado Sketch NodeMCU

Se aprecia de las figuras que ambos resultados coinciden, por lo que se puede considerar que la implementación fue exitosa.

4.5.2. *Método criptográfico simétrico*

La implementación del método criptográfico simétrico requiere que el PRNG esté funcionando correctamente tanto en el computador como en el IoTD. En los resultados expuestos en la sección 4.5.1 se comprueba que ambos requisitos se cumplen.

Ya con ambos programas funcionando, se procede a implementar una prueba de concepto que permita demostrar el método criptográfico.

Para dicha instancia, se establece un mensaje que puede ser interpretado como éxito. A través de MQTT, el NodeMCU recibirá un mensaje encriptado, lo desencriptará y ejecutará la función dependiendo del mensaje:

- Recibir un 52 como mensaje, imprimirá en puerto serial del NodeMCU un ”Mensaje 1 recibido”.
- Recibir un 53 como mensaje, imprimirá en puerto serial del NodeMCU un ”Mensaje 2 recibido”.
- Recibir un mensaje distinto a los dos casos anteriores, imprimirá en puerto serial del NodeMCU un ”Mensaje no reconocido”.

El mensaje será enviado desde el computador al broker MQTT público. Se utilizará Mosquitto[68] como mensajero. Se utiliza la biblioteca PubSubClient[69] para controlar el NodeMCU mediante MQTT. Se debe modificar el archivo *header* para aceptar mensajes mayores a 107 caracteres[70]. Se utilizarán los siguientes valores para los parámetros:

$$b = 1, \quad n = 8, \quad mu = 7, \quad p = 200, \quad i = 10, \quad j = 128 \tag{4.1}$$

Por lo que la función RCF descrita en la ecuación 3.16 de la sección 3.2.8.1 genera el

numero C:

$$C = RCF(1, 8, 7, 200, 10, 128) \tag{4.2}$$

$$C = 430576480140353609824752687732 68... \tag{4.3}$$

$$8141288169890171892312032461842 2... \tag{4.4}$$

$$7232895906302713441189940883295 8... \tag{4.5}$$

$$6183071032814972297048060793364 2 \tag{4.6}$$

Y para los mensajes $M_1 = 52$ y $M_2 = 53$, los mensajes encriptados E_1 y E_2 son:

$$E_1 = M_1 \cdot C = 52 \cdot C \tag{4.7}$$

$$E_1 = 22389976967298387710887139762 09... \tag{4.8}$$

$$9783346984834288938400225688015... \tag{4.9}$$

$$7981611058712774109894187692593... \tag{4.10}$$

$$1384815196937063785594464991612... \tag{4.11}$$

$$549384 \tag{4.12}$$

$$E_2 = M_2 \cdot C = 53 \cdot C \tag{4.13}$$

$$E_2 = 22820553447438741320711892449 83... \tag{4.14}$$

$$2471488273004179110292537720477... \tag{4.15}$$

$$6404334348303404381238306686681... \tag{4.16}$$

$$4680677027647391935317435472220... \tag{4.17}$$

$$483026 \tag{4.18}$$

Se utiliza *mosquitto_sub* para suscribirse al tópico *mensaje_encriptado*, y *mosquitto_pub* para publicar mensajes en dicho tópico. Se prepara el NodeMCU para suscribirse a dicho tópico, y esperar un mensaje para ser desencriptado. Se utilizan los scripts mostrados en los anexos C, C.1 y D.

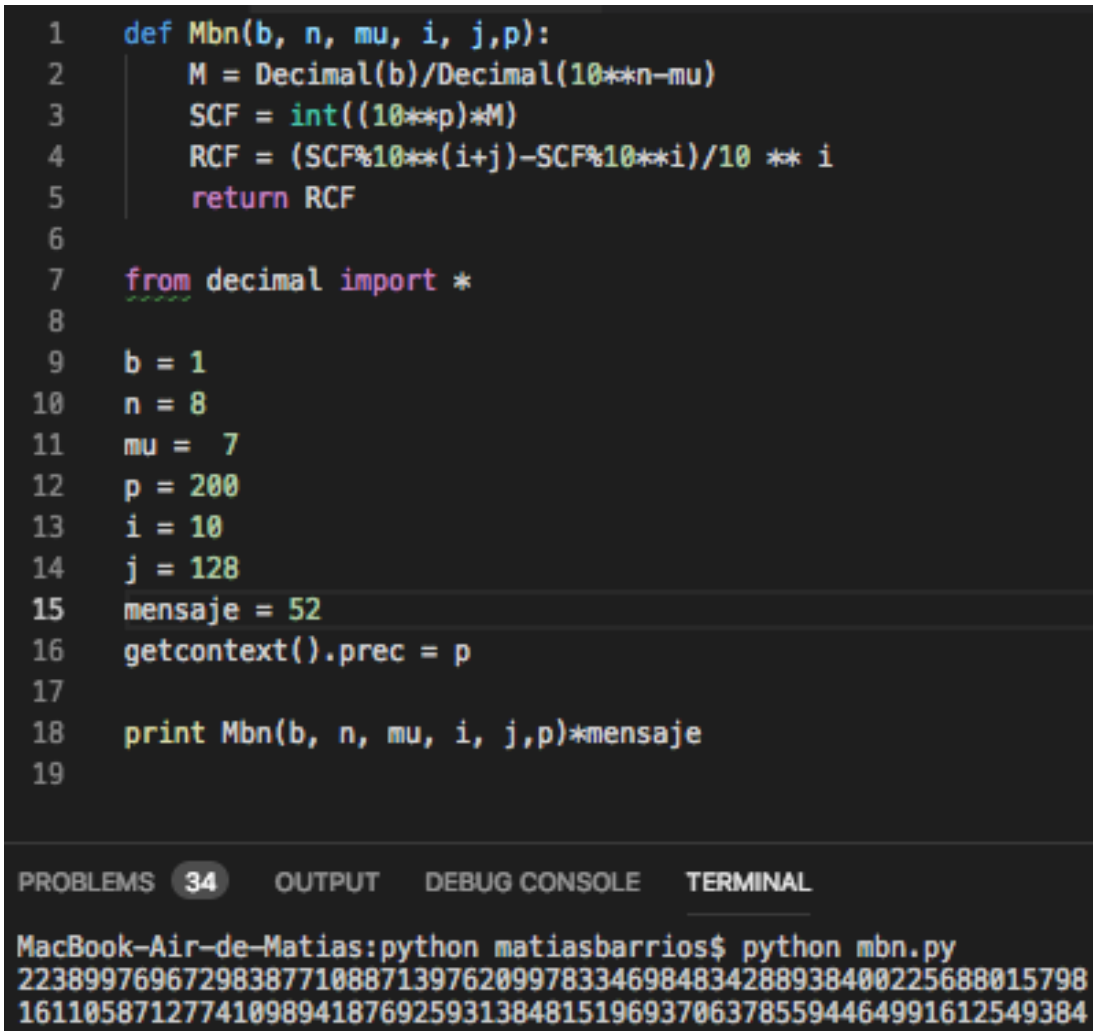

Fig. 4.4: Script Python y resultado para M=52

```python
def Mbn(b, n, mu, i, j,p):
    M = Decimal(b)/Decimal(10**n-mu)
    SCF = int((10**p)*M)
    RCF = (SCF%10**(i+j)-SCF%10**i)/10 ** i
    return RCF

from decimal import *

b = 1
n = 8
mu =  7
p = 200
i = 10
j = 128
mensaje = 53
getcontext().prec = p

print Mbn(b, n, mu, i, j,p)*mensaje
```

```
PROBLEMS 34   OUTPUT   DEBUG CONSOLE   TERMINAL

MacBook-Air-de-Matias:python matiasbarrios$ python mbn.py
22820553447438741320711892449832471488273004179110292537720477640
43343483034043812383066866814680677027647391935317435472220483026
```

Fig. 4.5: Script Python y resultado para M=53

```
z@U@@Connecting to WiFi..
Connecting to WiFi..
Connected to the WiFi network
Connecting to MQTT...
Connected to the MQTT Broker
Mensaje encriptado:22820553447438741320711892449832471488273004179110292537720477640433434830340438123830668668146806770276473919353174354722204830 26
Mensaje desencriptado 53
Mensaje 2 recibido
------------------------
Mensaje encriptado:22389976967298387710887139762099783346984834288938400225688015798161105871277410989418769259313848151969370637855944649916125493844
Mensaje desencriptado 52
Mensaje 1 recibido
------------------------
Mensaje encriptado:21959400487158034101062387074367095205696664398766507913655553955888776912214383855006869850480889533662267356358714945110046157424
Mensaje desencriptado 51
Mensaje no reconocido
------------------------
```

Fig. 4.6: Resultado de mensajes recibidos en NodeMCU

Al comparar los resultados de las imágenes 4.4, 4.5 y 4.6, se puede establecer que la implementación fue correcta y los resultados fueron los esperados.

Cabe destacar que los NodeMCU imprimen un mensaje al desencriptar un mensaje. Esta acción puede ser reemplazada y configurada para controlar un GPIO y poder así abrir y cerrar un estacionamiento para bicicleta.

4.5.3. Autenticación de dos pasos 2FA y Message authentication code (MAC)

Estos dos módulos necesitan de la lógica back-end funcionando, puesto que en el caso de 2FA se busca que el código 2FA presente en la base de datos coincida con el que ingresa el usuario mediante la interfaz web, y en el caso de Message authentication code (MAC) se requiere una comunicación de varios pasos entre EB y servidor.

Para ambos casos, sin embargo, es necesaria principalmente la implementación del PRNG, tanto en IoTD como en servidor, por lo que los resultados acá presentados son suficientes para determinar que una vez los servicios web estén implementados, tanto 2FA como MAC podrán ser utilizados.

4.6. Análisis de resultados de las implementaciones

Lamentablemente, el trabajo de una correcta implementación de los servicios web es una tarea larga y que necesita de un exhaustivo plan de testing, por lo que dicho trabajo está fuera del alcance de la investigación presentada en este documento.

La estructura física del estacionamiento tampoco está terminada, por lo que las pruebas de concepto no podrán ser completadas hasta que dicha tarea sea llevada a cabo. Esta tarea requiere de un profesional de aquel rubro, puesto que no pertenece al área de mis competencias.

Sin embargo, las pruebas presentadas en el plan de testing de la sección 4.4 y el diseño detallado en la sección 3.2 son suficientes para comprobar la funcionalidad de los módulos, una vez que las condiciones sean las propicias.

En lo fundamental, el PRNG y el método criptográfico simétrico son los compo-

nentes esenciales de la seguridad de este sistema. La implementación de ambos deja campo abierto a la espera de la terminación de los otros módulos, lo que permitirá implementar los restantes elementos diferenciadores de este desarrollo, i.e. 2FA y MAC.

4.7. Análisis critico y evaluación de riesgos

Para el proceso de evaluación de riesgos, es necesario entender el contexto de la implementación del servicio. En general, para que el servicio funcione se necesita lo siguiente.

4.7.1. Evaluación de riesgo respectivos a la disponibilidad de energía eléctrica

Se asume en primer lugar que el lugar en donde se instala el estacionamiento cuenta con una fuente de energía eléctrica,i.e. hay un enchufe en las cercanías del estacionamiento. Se asume además que dicho enchufe no es controlado mediante un interruptor, pues eso dejaría al estacionamiento fuera de servicio. Por la misma razón, la instalación no se debe hacer en un arranque para una luz con interruptor, debido a la facilidad con que se cortaría la energía eléctrica. Existen alternativas que podrían mitigar dicho riesgo, como la utilización de baterías de alimentación en caso de corte. El problema con esa alternativa es que incrementa los costos del estacionamiento, en términos de materiales y diseño de electrónica. Se debe estudiar el caso en que sea necesario instalar dicho sistema, informando al cliente el aumento en los costos de implementación.

Existen edificios que cuentan con sistema de respaldo, pero no se encontró información acerca de la regulación de la misma situación, por lo que no se puede asumir que existen.

Es recomendable entonces que al momento del trato con el cliente del servicio se pregunte por la disponibilidad de un sistema de respaldo eléctrico.

De todas formas, la elección del *Electronic Rotary Latch* mitiga el riesgo, en el sentido que dicho dispositivo no se abre ante la falta de energía eléctrica, por lo que las

bicicletas ya estacionadas no se podrán abrir hasta que haya energía y que el usuario se haya autenticado.

Desde el punto de vista de los dispositivos electrónicos, estos pueden perder sincronía si es que al reiniciar el sistema pierden el estado de sus PK y LDI. Es por eso que deben ser guardadas en EEPROM para evitar su pérdida.

4.7.2. Evaluación de riesgo respectivos a la disponibilidad de conexión a internet

La falta de internet también dejaría al sistema fuera de servicio. Se debe contratar servicios que provean de un SLA que permita asegurar la disponibilidad. En general, los ISP de Chile entregan una buena calidad de servicio, por lo que basta hacer una prueba in-situ de la calidad de la señal. Esto se logra mediante soluciones como SpeedTest[51] que entregan el ancho de banda según donde se esté. El volumen de datos utilizado se muestra en el diseño del módulo [M2], en que se demuestra que se necesita cerca de 200[Kbits] mensuales solo para conceptos de información intercambiada entre EB y servidor. De un plan de 10 [GB] es cerca del 20 % del total ofrecido, considerable pero aun bajo.

4.7.3. Evaluación de riesgo respectivos a la disponibilidad de la mecánica y electrónica del estacionamiento

Los estacionamientos deben estar implementados en lugares que entreguen cierta seguridad para el estacionamiento mismo. El diseño de este servicio no está pensado para la vía pública, aunque podría llegar a serlo si es que se toman en cuenta lo detallado en la sección de verificación y validación. De todas formas, esta necesidad apunta a que no ocurrirá nada a las bicicletas de los usuarios, de modo que alguien intente forzar o maltratar los estacionamientos. El diseño estructural debe apuntar a la durabilidad del estacionamiento, utilizando materiales preparados para condiciones similares a las que se podría enfrentar si es que el estacionamiento se encontrara en la vía pública. Dicho eso, basta una mantención mensual adecuada para que el servicio

esté funcional. Los dispositivos electrónicos deben estar protegidos de los factores sociales y ambientales, como manipulación de los IoTD por parte de terceros o corrosión debido a la exposición al clima.

4.7.4. Evaluación de riesgo respectivos a la seguridad del sistema

Toda la seguridad del sistema está basada en sistemas de secreto compartido. Dicho secreto se encuentra tanto en servidor como EB. En el caso de las EB, es más difícil extraer información. Sus dispositivos están preparados para recibir información y verificar si viene desde un ente reconocido. El riesgo es que alguien pueda robar un dispositivos de EB desde un estacionamiento. En dicho caso se podría tener información valiosa como las PK y las LDI. Es por ello que se debe considerar este factor y generar un sistema que entregue PK y LDI para cada EB en particular. Además, la estructura del estacionamiento debe proteger antes estos posibles robos.

Por el lado del servidor es un poco más complejo. Existen diversas formas en que los atacantes pueden entrar en el sistema y robar la información que es vital para el funcionamiento. Como la lógica del sistema se ejecuta en el servidor, este debe contar con toda la información que le permita administrar los estacionamientos. Un robo de dicha información resultaría catastrófico. Al momento de detectar dicha falla, se debe dejar el servicio inhabilitado para evitar un desastre mayor. Se debe además cambiar las credenciales referentes a PK, LDI, nombre de usuario y contraseña de base de datos y de broker MQTT. Para mitigar dicho riesgo, se debe tener personal capacitado para el monitoreo del sistema, en especial de proteger el acceso a la base de datos. Es recomendable además contratar los servicios de empresas de seguridad informática, de modo que ejecuten pruebas de penetración y mejorar la seguridad del sistema. Si bien el giro de la empresa no es referente a la seguridad informática, si es un factor preponderante para la disponibilidad del servicio. Como la intención del desarrollo de este proyecto es la conformación de una empresa, es necesario establecer los vectores de desarrollo, y la seguridad informática es uno de ellos. Es por ello que es altamente probable que se integre dicha disciplina en el giro principal de la empresa. Existen

fuentes de información como el *OWASP Top Ten*[52], que indica las 10 principales fallas de seguridad que se encuentran en una aplicación web. Mediante el estudio de ella, se puede mitigar aún más el riesgo de usurpación de información.

5. CONCLUSIONES

La implementación de un estacionamiento de bicicletas no es una tarea sencilla. Se requiere tener una serie de consideraciones técnicas antes de implantarlo, como el lugar de instalación, el material que se utiliza para su fabricación o el modelo que se debe escoger. Es por ello que los estacionamientos actuales han fallado en su utilización, porque además la instalación en la vía pública conlleva a riesgos difíciles de mitigar.

Las soluciones actuales no satisfacen a ciclistas ni clientes.

Cuando un ciclista deja su bicicleta estacionada en un estacionamiento de los actuales, puede como mucho asegurar que no le robarán su marco, y en ocasiones las ruedas de la bicicleta. El ciclista además encuentra problemas a la hora de asegurar otras partes como el sillín, los frenos, el sistema de cambios o el manubrio, lo que genera una inquietud constante y molesta, que lo mantiene preocupado porque su bicicleta puede ser dañada. Esto quiere decir que las soluciones actuales cumplen con su nivel funcional, pero no completan las necesidades del ciclista y su inseguridad emocional de dejar la bicicleta estacionada en la vía pública; ¿estará la bicicleta en las mismas condiciones en que la dejé?.

Los clientes también encuentran dificultades al elegir un estacionamiento apropiado, debido a desconocimiento o al poco interés en gastar recursos en una investigación apropiada. Al instalar los estacionamientos tipo U invertida, cometen errores al elegir el lugar donde instalarlo. Esto es debido a que quedan fuera de la vista de las personas que circulan y que podrían incrementar la seguridad de dicho estacionamiento, contrario a la creencia de que ”se deben ocultar las cosas para que estén seguras” que se ha adoptado al instalar estos estacionamientos.

Los servicios actuales en Chile ofrecen opciones con limitaciones que se analizan en la primera parte de esta memoria, y que en general están pensados en su uso en la vía pública sin el debido diseño de experiencia de usuario que un sistema de esta categoría requiere. Esto es debido a que asumen que una estructura robusta es suficiente para que se usen los estacionamientos, sin tener en cuenta que el ciclista no solo necesita proteger un objeto de valor monetario si no que muchas veces sentimental. La falta de facilidad de uso de sus soluciones alejan a posibles ciclistas de utilizarlos. Aunque existen casos como el de BiciChile de Banco de Chile, que presentan una excelente solución al problema, son casos muy puntuales y soluciones con poco potencial de escalabilidad.

La utilización de tecnologías de información y comunicaciones permite solventar gran parte de estos problemas, pues transfieren la mayor parte de la responsabilidad de los usuarios y administradores al sistema. Además, facilitan el proceso de uso de un estacionamiento, pues el ciclista no necesita portar candado, y el administrador solo debe registrar a los usuarios una vez y olvidarse del estacionamiento, debido a su mantención mensual.

El principal *trade-off* del uso de las TIC es que, como se dijo, se transfiere la mayor parte de las responsabilidades al sistema. Esto quiere decir que se debe diseñar el servicio para que cumpla con el mínimo de los requisitos; la bicicleta estará estacionada en un lugar seguro, y nadie mas que las personas autorizadas podrán sacarla de allí.

Esta implementación requirió un estudio profundo de los mecanismos que podrían potenciar la seguridad del sistema sin perder capacidad de escalabilidad. Después de todo, quien instala los estacionamientos no es una persona con profundos conocimientos técnicos, o por lo menos no los necesita para instalarlo.

La automatización en los procesos de ”secretos compartidos” entre EB y servidor requirió un estudio del proceso de instalación en detalle. El diseño del prototipo expuesto en este documento de memoria tiene como uno de sus pilares la facilidad de instalación, lo que requiere establecer el proceso completo de configuración de las EB

y PDE para que puedan ser instalados y cambiados con facilidad.

La seguridad es un punto a destacar. La seguridad se comprende como seguridad de la bicicleta y del sistema. Desde el punto de vista de la bicicleta, basta con examinar el lugar donde se instalará para establecer su factibilidad de implementación. La seguridad del sistema es mas complicada. Se requiere un estudio mas profundo del área, además de asesoría profesional antes de poner el sistema en etapa de producción.

La principal innovación tecnológica de lo propuesto en este documento de memoria es lo relacionado al diseño del módulo [M8]: Lógica de acción de estacionamientos. Mediante el uso de una poderosa herramienta como un PRNG se pueden personalizar tres soluciones relacionadas con la seguridad: la criptografía simétrica, la autenticación de 2 pasos (2FA) y los *Message Authentication Code* (MAC) que se producen en las etapas de confirmación de intercambio exitoso de información. Esta solución es, a priori, universal, pues puede ser implementada en cualquier dispositivo IoT debido a su bajo costo computacional y facilidad de implementación.

Por último, la principal innovación social es demostrar la capacidad que tienen las tecnologías de mejorar la calidad de vida de las personas. El uso de la bicicleta trae beneficios a la salud, disminución del tráfico, ahorro en bencina, reducción de la huella de carbono, reducción de la polución del aire y reducción de la contaminación acústica. Estos no solo ayudan al ciclista, si no que toda la sociedad se ve impactada por los beneficios debido a que disminuye el uso del automóvil. La implementación de una red de estacionamientos para bicicletas evidenciará la gran cantidad de personas que la utilizan, y dará pruebas fidedignas de que la sustentabilidad es un camino que nos beneficia a todos de manera directa e inmediata.

Bibliografía

[1] VILLALOBOS Julian David. LA BICICLETA COMO MEDIO PARA LA DISMINUCIÓN DE CONTAMINACIÓN AMBIENTAL, ACCIDENTES DE TRABAJO Y ENFERMEDADES LABORALES EN LAS ORGANIZACIONES. Bogotá, Colombia. 2016. http://unimilitar-dspace.metabiblioteca.org/handle/10654/15511 [Consulta el 18 de abril de 2018]

[2] Ministerie van verkeer en waterstaat, página 8, punto 1.3, Holanda, 2009. http://www.fietsberaad.nl/library/repository/bestanden/Labicicletaenpaisesbajos2009.pdf [Consulta el 18 de abril de 2018]

[3] Ministerie van verkeer en waterstaat, página 7, punto 1.2, Holanda, 2009. http://www.fietsberaad.nl/library/repository/bestanden/Labicicletaenpaisesbajos2009.pdf [Consulta el 18 de abril de 2018]

[4] Principio de diseño KISS (Keep it simple, stupid). `https://www.interaction-design.org/literature/article/kiss-keep-it-simple-stupid-a-design-principle` [Consulta el 18 de abril de 2018]

[5] Documentación oficial I2C. `https://www.nxp.com/docs/en/user-guide/UM10204.pdf` [Consulta el 21 de abril de 2018]

[6] API de Transbank. `http://www.transbankdevelopers.cl/api/webpay` [Consulta el 21 de abril de 2018]

[7] API de KHIPU. https://khipu.com/page/api [Consulta el 21 de abril de 2018]

[8] Opciones de WiFi móvil de WOM. https://www.wom.cl/internet/?utm_source=Google&utm_medium=Search&utm_content=Search&utm_campaign=wominternet2018 [Consulta el 21 de abril de 2018]

[9] SLA de Amazon Web Services https://aws.amazon.com/es/compute/sla/ [Consulta el 21 de abril de 2018]

[10] Autenticación de dos factores. https://support.microsoft.com/en-us/help/12408/microsoft-account-about-two-step-verification [Consulta el 21 de abril de 2018]

[11] Movilidad Urbana, biciestacionamientos en el espacio público, MINVU, 2013, página 20. http://www.minvu.cl/opensite_20131009101013.aspx [Consulta el 30 de abril de 2018]

[12] Message Queue Telemetry Transport, MQTT. https://www.ibm.com/support/knowledgecenter/es/SSFKSJ_7.5.0/com.ibm.mq.pro.doc/q002870_.htm [Consulta el 30 de abril de 2018]

[13] Message Queue Telemetry Transport, MQTT. https://www.ibm.com/support/knowledgecenter/es/SSFKSJ_7.5.0/com.ibm.mq.pro.doc/q002870_.htm [Consulta el 02 de mayo de 2018]

[14] ESP8266, página del fabricante. https://bbs.espressif.com/viewtopic.php?f=67&t=225/ [Consulta el 02 de mayo de 2018]

[15] Electronic Rotary Latch R4-EM de Southco. https://www.southco.com/en-us/r4-em [Consulta el 02 de mayo de 2018]

[16] Datasheet de Electronic Rotary Latch R4-EM de Southco. https://www.southco.com/static/Literature/R4-EM.en.pdf [Consulta el 15 de mayo de 2018]

[17] Superintendencia de Electricidad y Combustible de Chile,SEC. http://www.sec.cl/portal/page?_pageid=33,1&_dad=portal&_schema=PORTAL [Consulta el 15 de mayo de 2018]

[18] Datasheet de ESP8266, página del fabricante. https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf [Consulta el 15 de mayo de 2018]

[19] Datasheet de Huawei B310. https://www.4gltemall.com/downloads/dl/file/id/308/product/614/huawei_b310s_22_lte_cpe_specs_datasheet.pdf [Consulta el 15 de mayo de 2018]

[20] Precios promedio kWh según ENEL Chile. https://www.eneldistribucion.cl/tarifas [Consulta el 15 de mayo de 2018]

[21] HARRINGTON, Jan. Relational Database Design and Implementation, Fourth Edition. [Consulta el 15 de mayo de 2018]

[22] Time-Base On-Time Password Authentication, TOTP, RFC 6238, https://tools.ietf.org/html/rfc6238. [Consulta el 21 de mayo de 2018]

[23] Relational Database Services, Amazon Web Services https://aws.amazon.com/es/rds/. [Consulta el 21 de mayo de 2018]

[24] Broker público CloudMQTT, https://www.cloudmqtt.com/. [Consulta el 28 de mayo de 2018]

[25] TLS/SSL, RFC 526, https://tools.ietf.org/html/rfc5246. [Consulta el 28 de mayo de 2018]

[26] NodeMCU http://nodemcu.com/index_en.html. [Consulta el 28 de mayo de 2018]

[27] ESP8266 https://www.espressif.com/en/products/hardware/esp8266ex/overview [Consulta el 28 de mayo de 2018]

[28] ESP01 http://www.microchip.ua/wireless/esp01.pdf [Consulta el 28 de mayo de 2018]

[29] Relay Groove 3v, SeeedsStudio https://www.seeedstudio.com/Grove-Relay-p-769.html [Consulta el 29 de mayo de 2018]

[30] Practical Random Number Generation in Software, John Viega, Virginia Tech, 2003 https://www.acsac.org/2003/papers/79.pdf [Consulta el 29 de mayo de 2018]

[31] GNU Privacy Guard https://www.gnupg.org/gph/es/manual.html [Consulta el 02 de junio de 2018]

[32] J. Katz, Y. Lindell. Introduction to Modern Criptography, segunda edición. [Consulta el 02 de junio de 2018]

[33] Schroeder, Manfred Robert. Number theory in science and communication: with applications in cryptography, physics, digital information, computing, and self-similarity, Editorial Springer-Verlag, primera edición, 1986. [Consulta el 02 de junio de 2018]

[34] Shor, Peter. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. [Consulta el 02 de junio de 2018]

[35] James P. Jones, Daihachiro Sato, Hideo Wada, Douglas Wiens: Diophantine representation of the set of prime numbers (1976), American Mathematical Monthly 83: 449–464. [Consulta el 03 de junio de 2018]

[36] *Omega(n), number of prime factors of n (with multiplicity)*, http://oeis.org/wiki/Omega(n),_number_of_prime_factors_of_n_(with_multiplicity). [Consulta el 03 de junio de 2018]

[37] Pantalla OLED SSD1306 para SVP, https://www.amgkits.com/home/64-display-oled-096-128x64-i2c.html?gclid=EAIaIQobChMIwuLVuubi2wIVQgeRCh0jmwXgEAYYASABEgLZEvD_BwE. [Consulta el 20 de junio de 2018]

[38] Django, Python Web Development Framework, https://www.djangoproject.com/. [Consulta el 20 de junio de 2018]

[39] Competencia *Cool Place to Bike*, KAPPO.bike, www.coolplacetobike.com. [Consulta el 20 de junio de 2018]

[40] Inmobiliaria Ralei Development Group, http://www.ralei.cl/. [Consulta el 20 de junio de 2018]

[41] Biciestacionar en edificios de uso público, ADC Bicicultura, https://www.bicicultura.cl/2016/07/01/4658/. [Consulta el 20 de junio de 2018]

[42] DDU 288 2015, Ministerio de Vivienda y Urbanismo, http://www.minvu.cl/incjs/download.aspx?glb_cod_nodo=20070621120807&hdd_nom_archivo=DDU%20288.pdf. [Consulta el 20 de junio de 2018]

[43] Ciclo-inclusión en América Latina y el Caribe: Guía para impulsar el uso de la bicicleta, Banco Interamericano para el Desarrollo, https://publications.iadb.org/handle/11319/6808?scope=123456789/1&thumbnail=true&rpp=5&page=1&group_by=none&etal=0. [Consulta el 20 de junio de 2018]

[44] Lenguaje de programación Python, https://www.python.org/. [Consulta el 20 de junio de 2018]

[45] MySQL Community Server 8.0.11, https://dev.mysql.com/downloads/mysql/. [Consulta el 20 de junio de 2018]

[46] Datacenters de *Amazon Web Services*, https://aws.amazon.com/es/compliance/data-center/data-centers/. [Consulta el 20 de junio de 2018]

[47] Arduino IDE 1.8.5, https://www.arduino.cc/en/main/software. [Consulta el 20 de junio de 2018]

[48] Biblioteca *Big Number*, Nick Gammon, https://github.com/nickgammon/BigNumber. [Consulta el 20 de junio de 2018]

[49] Diseño *frontend* y *backend*, https://es.wikipedia.org/wiki/Front-end_y_back-end. [Consulta el 20 de junio de 2018]

[50] Que es UX y UI, http://blog.acantu.com/que-es-ux-y-ui/. [Consulta el 20 de junio de 2018]

[51] SpeedTest, servicio de prueba de ancho de banda, http://www.speedtest.net/. [Consulta el 21 de junio de 2018]

[52] OWASP Top Ten, 10 fallas críticas de seguridad de una aplicación web, https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. [Consulta el 21 de junio de 2018]

[53] Mark Weiser, *The Computer for the 21st Century*, https://www.ics.uci.edu/~corps/phaseii/Weiser-Computer21stCentury-SciAm.pdf. [Consulta el 21 de junio de 2018]

[54] Expansión México, *Holanda construye el estacionamiento de bicicletas más grande del mundo*, https://expansion.mx/tendencias/2017/08/21/

holanda-construye-el-estacionamiento-de-bicicletas-mas-grande-del
[Consulta el 13 de septiembre de 2018]

[55] Homepage Bikeep, https://bikeep.com/. [Consulta el 13 de septiembre de 2018]

[56] Homepage Aparka, https://bizintek.es/proyecto/aparka. [Consulta el 13 de septiembre de 2018]

[57] Homepage Ecocycle, http://www.ecocycle.co.uk/. [Consulta el 13 de septiembre de 2018]

[58] Homepage Cyclesafe, https://cyclesafe.com/. [Consulta el 13 de septiembre de 2018]

[59] Homepage Don Cicleto, https://www.web.doncicleto.com/. [Consulta el 13 de septiembre de 2018]

[60] Homepage Bicilock, http://www.bicilock.cl/. [Consulta el 13 de septiembre de 2018]

[61] Información Bicichile, http://trabajaenelchile.cl/post/bicichile-un-pasatiempo-camino-al-trabajo/. [Consulta el 13 de septiembre de 2018]

[62] Localización de Estacionamientos de Bicicletas en Santiago, http://www.subtrans.cl/upload/estudios/LocalizacionBicicletas-IF.pdf. [Consulta el 13 de septiembre de 2018]

[63] Homepage Bicipunto, http://bicipuntospa.cl/. [Consulta el 13 de septiembre de 2018]

[64] Información Nación Pedal, http://www.nacionpedal.com/custodia/. [Consulta el 13 de septiembre de 2018]

[65] Bicimapa de Bicicultura, https://www.bicicultura.cl/bicimapa/. [Consulta el 13 de septiembre de 2018]

[66] Data Types in Arduino , https://learn.sparkfun.com/tutorials/data-types-in-arduino. [Consulta el 21 de septiembre de 2018]

[67] Biblioteca *BigNumber*, por Nick Gammon, https://github.com/nickgammon/BigNumber. [Consulta el 21 de septiembre de 2018]

[68] *Mosquitto*, MQTT Broker http://wiki.eclipse.org/Mosquitto. [Consulta el 21 de septiembre de 2018]

[69] Biblioteca PubSubClient https://github.com/knolleary/pubsubclient. [Consulta el 22 de septiembre de 2018]

[70] Modificación de archivo header para aceptar mensajes de mas de 107 caracteres https://www.esp8266.com/viewtopic.php?f=160&t=15872. [Consulta el 22 de septiembre de 2018]

Apéndice A

SCRIPT PYTHON DEL PRNG

```
def Mbn(b, n, mu, i, j,p):
    M = Decimal(b)/Decimal(10**n-mu)
    SCF = int((10**p)*M)
    RCF = (SCF%10**(i+j)-SCF%10**i)/10 ** i
    return RCF

from decimal import *

b = 1
n = 8
mu =  7
p = 200
i = 10
j = 128
getcontext().prec = p

print Mbn(b, n, mu, i, j,p)
```

Apéndice B

SKETCH ARDUINO PARA IMPLEMENTACIÓN DE PRNG EN NODEMCU

```
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <math.h>
#include "BigNumber.h"
#include <string.h>
const char* ssid = "maco.net";
const char* password = "JeBn*20471052-K#";

//Led
int ledPin = BUILTIN_LED;

//Private Keys
int b = 1;
int n = 8;
int mu = 7;
int p = 200;
int i = 10;
int j = 90;
BigNumber ten = 10;

IPAddress ip(192, 168, 1, 17);
IPAddress gateway(192,168,1,1);
IPAddress subnet (255,255,255,0);
WiFiServer server(301);

void setup() {
```

```
  BigNumber::begin ();
  Serial.begin(115200);
  delay(10);
  WiFi.config(ip,gateway,subnet);
  Serial.println(WiFi.localIP());

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  server.begin();
  Serial.println("Server started");
  Serial.println(WiFi.localIP());
}

void loop() {
  //Generar n mero
  BigNumber numbAmount = ten.pow(p);
  BigNumber::setScale(p);
  BigNumber M = BigNumber(b)/BigNumber(pow(10,n)-mu);
  BigNumber::setScale(0);
  BigNumber SCF = M *numbAmount;
  BigNumber RCF = (SCF%ten.pow(i+j)-SCF%ten.pow(i))/ten.pow(i);
  //Print
  Serial.print("privKey = ");
  Serial.println(RCF);
  delay(5000);
}
```

Apéndice C

SCRIPT PARA ENVIAR MENSAJES ENCRIPTADOS CON PYTHON

```
def Mbn(b, n, mu, i, j,p):
    M = Decimal(b)/Decimal(10**n-mu)
    SCF = int((10**p)*M)
    RCF = (SCF%10**(i+j)-SCF%10**i)/10 ** i
    return RCF

from decimal import *
b = 1
n = 8
mu =  7
p = 200
i = 10
j = 128
mensaje = 53
getcontext().prec = p
print Mbn(b, n, mu, i, j,p)*mensaje
```

C.1. Comandos para usar mosquitto en linea de comando

```
mosquitto_pub -h m13.cloudmqtt.com -p 16342 -u "jdlolrja" --pw "
    AG5uhSft5ylk" -t mensaje_encriptado -m "mensaje encriptado"
mosquitto_sub -h m13.cloudmqtt.com -p 16342 -u "jdlolrja" --pw "
    AG5uhSft5ylk" -t mensaje_encriptado
```

Apéndice D

SKETCH PARA RECIBIR MENSAJES ENCRIPTADOS MEDIANTE MQTT

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <math.h>
#include "BigNumber.h"
#include <string.h>

/************************ WiFi Access Point
    *********************************/

#define WLAN_SSID       "maco.net"
#define WLAN_PASS       "JeBn*20471052-K#"

/************************ Adafruit.io Setup
    *********************************/

#define MQTT_SERVER      "m13.cloudmqtt.com"
#define MQTT_PORT  16342                    // use 8883 for SSL
#define MQTT_USER    "jdlolrja"
#define MQTT_PASS         "AG5uhSft5ylk"

WiFiClient espClient;
PubSubClient client(espClient);

//Private Keys
int b = 1;
```

```
int n = 8;
int mu = 7;
int p = 200;
int i = 10;
int j = 128;
BigNumber ten = 10;
BigNumber RCF = 0;

void setup() {
  BigNumber::begin ();
  Serial.begin(115200);
  delay(10);
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);

  WiFi.begin(WLAN_SSID, WLAN_PASS);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");

  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);

  while (!client.connected()) {
    Serial.println("Connecting to MQTT...");

    if (client.connect("ESP8266Client", MQTT_USER, MQTT_PASS )) {

      Serial.println("Connected to the MQTT Broker");

    } else {

```

```
      Serial.print("failed with state ");
      Serial.print(client.state());
      delay(2000);

    }
  }
  client.subscribe("mensaje_encriptado");
}

void callback(char* topic, byte* payload, int length) {
  char inputChar;
  BigNumber encryptedMessage;
  Serial.print("Mensaje encriptado:");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
    inputChar = (char)payload[i];
    encryptedMessage+=((BigNumber)inputChar-(BigNumber)48)*ten.pow(length-i
        -1);
  }
  BigNumber decryptedMessage = encryptedMessage/RCF;

  Serial.println();
  Serial.print("Mensaje desencriptado ");
  Serial.println(decryptedMessage);
  if(decryptedMessage%BigNumber(52)==0){
    Serial.println("Mensaje 1 recibido");
  }else if(decryptedMessage%BigNumber(53)==0){
    Serial.println("Mensaje 2 recibido");
  }else {
    Serial.println("Mensaje no reconocido");
  }

  Serial.println("----------------------");

}

void loop() {
  //Decryptor
```

```
100   BigNumber numbAmount = ten.pow(p);
101   BigNumber::setScale(p);
102   BigNumber M = BigNumber(b)/BigNumber(pow(10,n)-mu);
103   BigNumber::setScale(0);
104   BigNumber SCF = M *numbAmount;
105   RCF = (SCF%ten.pow(i+j)-SCF%ten.pow(i))/ten.pow(i);
106   client.loop();
107 }
```