UTN SANTA FE

Proyecto Final de Carrera
Ingeniería en Sistemas de Información

Sistema de cerraduras comandadas por NFC con servidor centralizado

Alumnos
Eichhorn, Lucas Alberto
Etcheverrigaray, Gonzalo Damián
Mazzei, Julián Ezequiel

Director
Ing. Bracalenti, Tomás Alfredo

Codirector
Ing. Filippa, Gabriel

2019

Índice

1. Introducción

Debido al gran avance de la tecnología en los últimos años, se ha logrado aumentar el confort de las personas con la creación de edificios inteligentes. Para lograr esto la tecnología utiliza un concepto nuevo de dispositivos que nació en el Instituto de Tecnología de Massachusetts (MIT) denominado Internet de las cosas (IoT, por sus siglas en inglés).

Se trata una revolución en las relaciones entre los objetos y las personas, incluso entre los objetos directamente, que se conectarán entre ellos y con la red y ofrecerán datos en tiempo real. Mediante la interconexión de elementos domésticos a través de redes de información y dispositivos que administran de forma lógica su funcionamiento, se automatizan las tareas que debe realizar una persona en cuestiones de mantenimiento y utilización cotidiana de cualquier tipo de vivienda u oficina, este concepto se denomina domótica.

La domótica permite dar respuesta a los requerimientos que plantean estos cambios sociales y las nuevas tendencias de nuestra forma de vida, facilitando el diseño de casas y hogares más humanos, más personales, funcionales y flexibles, que sin lugar a dudas nos puede proporcionar un mejor y cómodo estilo de vida, y si bien algunos sistemas más complejos de este tipo aún se encuentran en desarrollo, lo cierto es que ya existen en el mundo edificios inteligentes que utilizan la domótica para ello.

Adentrándonos en este proyecto analicemos ahora otro aspecto importante que cubre la tecnología, y es la seguridad física del edificio. La tecnología permite aumentar la seguridad de un edificio reemplazando su obsoleto sistema de cerradura por un sistema de cerradura electrónicas autorizadas. Mediante políticas de control, se puede restringir el acceso a los distintos recintos solamente a personas autorizadas; para esto es necesario poseer dispositivos diseñados para tal fin.

Con este proyecto se buscó hacer un aporte a la industria de la domótica, con un producto innovador y competitivo con soluciones actuales, permitiendo no solamente la simplificación en la apertura de una puerta, sino también mantener un control más específico. Por ejemplo, registrando eventos de cuándo y por quienes se realizó la apertura de una puerta.

Se planteó crear un sistema domótico enfocado en mejorar los aspectos de confort y seguridad en una edificación, mejorando los sistemas de cerraduras físicas y magnéticas comandadas por dispositivos pasivos por lo que se estableció un conjunto de objetivos que guiaron el desarrollo del presente proyecto.

a. Descripción

La idea principal es un producto que permita la apertura y cierre de puertas utilizando como “llave” de acceso dispositivos móviles, que, con el simple hecho de acercar dicho aparato a la cerradura de la puerta, ésta se accione.

El sistema propuesto en este proyecto es una solución con servidor centralizado con el cual se permite la administración tanto de los dispositivos habilitados para la apertura, los cuales deben ser registrados de forma única, como de las cerraduras, que podrán ser monitorizadas y reguladas de forma meticulosa, permitiendo a un administrador de un edificio controlar los accesos a cada ambiente de forma rigurosa.

El producto planteado permite ser ofrecido a distintos perfiles de cliente, por lo que se presentan tres grandes grupos:

- Sistema enfocado para hogares, permitiendo a una persona la administración de su hogar con ciertas funcionalidades adecuadas para tal fin.
- Sistema enfocado para empresas, donde los administradores tendrán no solamente la posibilidad de administrar dispositivos, sino que también generar distintos perfiles de usuario, obtener reportes, entre otras funciones.
- Sistema enfocado para hoteles, en donde se podrá administrar a gran nivel, determinando fechas, horarios, usuarios frecuentes, y todos los requerimientos que se necesiten para este rubro.

Los dispositivos presentes en el mercado que cumplen con funciones relacionadas con el proyecto no cuentan con la tecnología suficiente para lo que representa la seguridad en la actualidad. Hoy en día la mayoría tiende a estar conectados a las redes, pero el paradigma de cerraduras vigente no cuenta con esta funcionalidad. Si bien existen actualmente algunas comandadas mediante dispositivos pasivos (conocidos como tags NFC), al no estar conectados a una red no permiten la interoperabilidad entre ellas, o con un servidor, haciendo su funcionamiento más aislado y más difícil de administrar y monitorizar.

Con la posibilidad de tener un dispositivo conectado a la red, viene un gran problema en cuanto a la seguridad, debido a que éste se encontrará expuesto a posibles ataques; por lo tanto, este proyecto plantea un gran foco en cuanto a la seguridad informática de cada uno de los agentes involucrados. En este aspecto se tienen varias etapas de control tanto internas como externas, como se requiere en un desarrollo de sistema de esta índole.

b. Objetivos generales

Desarrollar el módulo de un sistema domótico, el cual se encargará de administrar y comandar las cerraduras de un inmueble.

c. Objetivos específicos

· Evaluar las tecnologías necesarias para el proyecto.
· Evaluar los módulos del sistema que se diseñarán e implementarán.
· Seleccionar un nivel de seguridad adecuado.
· Diseñar el modelo de cerradura que se va a utilizar.
· Desarrollar el módulo servidor de comando de las cerraduras.
· Desarrollar la interfaz administrativa.
· Desarrollar la aplicación cliente.
· Definir la forma de interacción entre el servidor y los dispositivos comandados.
· Validar el sistema mediante la creación de un prototipo.

d. Alcance

Se definió que, debido a la magnitud del proyecto, era conveniente dividir éste en distintas etapas. Por lo tanto, el alcance de la primera etapa comprendió todas las tareas necesarias para construir la versión inicial del sistema. Ésta estuvo enfocada al desarrollo del sistema orientado a un ámbito hogareño, sin tener en cuenta las funcionalidades para empresas u hoteles.

i. Etapa inicial

Con respecto al trabajo que se desarrolló, se dividió en dos sub-etapas, las cuales serán explicadas de forma detallada en el presente documento.
En la primera etapa, de investigación, se realizaron estudios pertinentes al dominio de la problemática para familiarizarse con las soluciones posibles a implementar, pudiendo con esto seleccionar las tecnologías necesarias para el desarrollo del proyecto.
Por otra parte, se llevó a cabo el diseño y el desarrollo del software necesario para el funcionamiento tanto de la cerradura como del servidor. Esto, sumado a la aplicación móvil ya mencionada, contempla el funcionamiento básico del sistema.

ii. Siguientes etapas

Está proyectado realizar las etapas de desarrollo del producto para ámbitos hoteleros y empresariales, desarrollando en ellas el producto con las funcionalidades necesarias para cubrir las necesidades de éstos. Se determinará la ampliación de funcionalidades

necesarias, como pueden ser la eliminación automática de dispositivos de apertura, para el servicio de hotelería, o el servicio de notificaciones en tiempo real a los administradores.

Además, se deberá realizar en una siguiente etapa el desarrollo de la aplicación móvil para dispositivos iOS que, debido a inconvenientes que serán descritos en el desarrollo del informe, no se llevará a cabo en ésta.

Por último, se llevará a cabo una etapa que hará énfasis a la seguridad, ya que por la complejidad de dicho aspecto se determinó que se necesitará la profundización especializada en el tema.

2. Introducción al proyecto

a. ¿Cómo se compone el sistema?

Para el desarrollo de este proyecto, se plantearon primero los tres grandes elementos que lo componen. Estos son: las cerraduras, el panel de control central de administración de las cerraduras, y la aplicación del dispositivo móvil que interactúa con ellas.

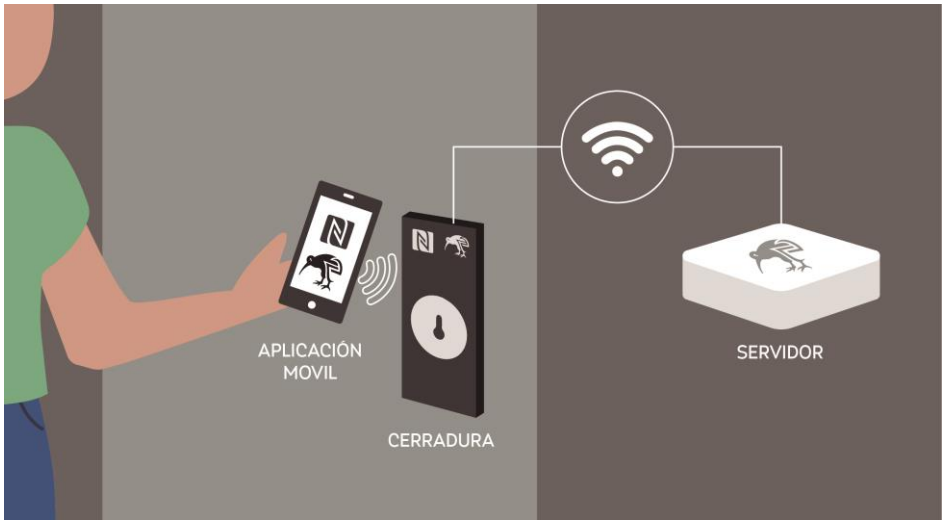


Figura 1) Ilustración de los 3 subsistemas

La cerradura fue diseñada para que, una vez configurada y dada de alta en el panel central, pudiera realizar su tarea de manera autónoma. Es decir, la cerradura tendrá información almacenada que le permitirá reconocer los dispositivos habilitados para accionar (abrir o cerrar la misma) y registrar el ingreso o egreso de los usuarios para posteriormente notificar dichos eventos al panel central.

Operar de forma autónoma permite que la interacción del usuario con la cerradura sea lo más rápido posible, debido a que la alternativa hubiera requerido una comunicación constante con el panel central, resultando en un mayor tiempo de respuesta.

Las cerraduras se encuentran conectadas a la red local para la comunicación con el panel central. Por otro lado, la aplicación móvil presenta una solución que simplifica al usuario la configuración inicial de la cerradura.

El panel central se diseñó con tecnologías web y, si bien estará conectado a la red local, no tendrá acceso desde Internet. Esta última característica se planteó como una restricción fundamental en el desarrollo por cuestiones de seguridad e integridad del sistema, lo cual además no representa una pérdida de funcionalidades para el usuario. El servidor que contiene el panel central de administración tal vez represente la parte que más configuración requiera por parte del usuario, quizás con algunos pasos específicos de configuración de redes, ya que su simplificación requería un mayor esfuerzo sobre el

desarrollo del sistema, el cual no se plantea abordar en el alcance de esta etapa del proyecto.

La aplicación del dispositivo móvil es instalada por el usuario para su interacción con la cerradura. El dispositivo móvil será el responsable de la configuración inicial de las cerraduras para el alta de las mismas en la red. Esto se decidió, en principio, para hacer más amigable la instalación para el usuario porque será más frecuente el agregar nuevas cerraduras que paneles centrales, que sólo son configurados una única vez.

Además, el dispositivo mismo deberá ser dado de alta en el sistema mediante el panel central, para luego poder otorgarle permisos de apertura de las cerraduras. Esto se realizará a través de la cerradura por medio de NFC, efectivamente realizando la primera comunicación NFC entre dispositivo y cerradura, en la cual el dispositivo tendrá asignado un código de identificación que almacenará en su memoria interna. Este código de identificación es único para el sistema de cerraduras local y será la que se transmita en las subsiguientes comunicaciones NFC con la cerradura.

El rol central de la cerradura es la comunicación con el dispositivo móvil a través de NFC. Esta diferirá de las interacciones NFC con las que el usuario probablemente pueda estar familiarizado, como la lectura de un código de una tarjeta o tag NFC, ya que en nuestro caso ocurre un intercambio de información entre cerradura y móvil en vez de una simple lectura. Si bien la decisión de hacerlo de esta manera significa para el usuario que deberá mantener el dispositivo en contacto con la cerradura por un instante de tiempo más prolongado, esto otorga el beneficio al usuario de poder usar con un mismo dispositivo móvil distintos sistemas de cerraduras en distintos edificios.

b. Arquitectura del sistema

Se contempló para la arquitectura que el servidor central contenga el módulo de administración general del sistema, una interfaz encargada de la comunicación con los dispositivos y que se almacene en una base de datos toda la información pertinente a la identificación de cerraduras y dispositivos, así como también alojar los registros de acceso.

El módulo de administración general se divide en tres subsistemas con funcionalidades bien diferenciadas entre sí. En primer lugar, tenemos el servidor web con la interfaz de usuario del panel administrativo que facilita la gestión de cerraduras y dispositivos por parte del usuario desde un navegador. Este se comunica con otro subsistema presente en el servidor central, una API REST, la cual es la encargada del procesamiento, distribución y almacenamiento de la información generada tanto por el usuario como por las cerraduras y dispositivos. En otro subsistema encontramos un servidor de *websockets*, dicho de forma

sencilla y breve este componente es el encargado de enviar mensajes al navegador en tiempo real al ocurrir eventos de distinto tipo en el sistema para dar aviso al usuario.

Este módulo central tiene conocimiento de las cerraduras del sistema y establecerá una comunicación con ellas a través del segundo componente del servidor que cumple la función de interfaz, tanto para enviar información nueva sobre permisos como para recibir registros de acceso y almacenarlos. De esta manera, si bien la cerradura puede responder a los usuarios de manera independiente, siempre deberá estar comunicándose con el panel central para mantener la información actualizada de los eventos ocurridos.

Por último, el servidor contiene una base de datos en donde se aloja toda la información requerida por el sistema, la cual es accedida por los dos componentes descritos anteriormente.

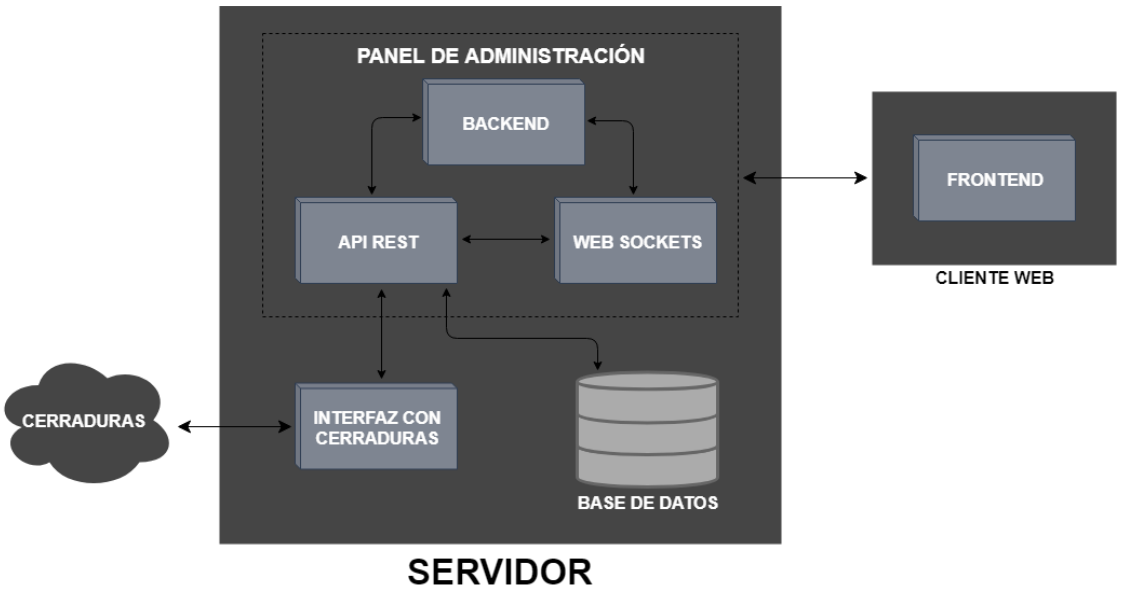


Figura 2) *Interacción de componentes del servidor*

La aplicación móvil que permite identificar a los usuarios es otro componente de la arquitectura planteada del sistema de cerraduras. La misma se diseñó para que el usuario pueda interactuar con las cerraduras de la forma más sencillamente posible, de manera que no sea necesario que la aplicación esté en ejecución en primer plano para su interacción con una cerradura.

En el caso de la aplicación móvil, ésta no posee conocimiento de cada una de las cerraduras del sistema, como lo tiene el módulo central de administración, sino que guarda una credencial del sistema de cerraduras general al que pertenece, es decir, que poseen un identificador del sistema global. De esta manera, el dispositivo podrá ser registrado en más de un sistema de cerraduras correspondiente a distintos edificios, pero gestionados mediante la misma aplicación.

Por último, pero no menos importante, están las cerraduras, las cuales están compuestas por un software, el cual centraliza toda la lógica de ellas, y posee los módulos de interfaz con los dispositivos móviles (por NFC) y con el servidor (por WLAN). Además, posee una base de datos para almacenar la información de qué dispositivos están habilitados para interactuar con ella.

La utilización y descripción detalladas de las tecnologías antes mencionadas serán debidamente explicadas a medida que corresponda en las siguientes secciones a lo largo del presente informe.

3. Organización del proyecto

Durante la etapa de planeación del proyecto, se determinó que el grupo utilizaría técnicas de metodologías ágiles, tomando las herramientas convenientes para la coordinación del grupo de trabajo y la optimización de los recursos.

Previo al comienzo del desarrollo, se realizó una etapa de investigación, obteniendo una lista de herramientas y plataformas a utilizar. Se homogeneizaron los métodos de utilización de estas para mantener la coherencia dentro del proyecto.

Para diseñar una óptima configuración organizativa a fin de cumplir los tiempos planeados, se evaluó la arquitectura de diseño del sistema en general, y se evaluaron las distintas capacidades de los integrantes. Se optó por dividir el trabajo en tres grandes grupos:

- Desarrollo de sistema web
- Desarrollo de aplicación móvil
- Desarrollo de software de apoyo e interfaces de sistemas.

a. Metodología de desarrollo

En primer lugar, es necesario resaltar que el equipo optó por respetar el *Manifiesto ágil* [1], con el fin de desarrollar el producto mediante una metodología acorde.

Se estudió la factibilidad de la implementación pura de la metodología ágil “XP” (Programación Extrema), y se decidió que no podrían ser utilizadas todas las herramientas para el ciclo de vida actual del proyecto debido a distintas circunstancias como la cantidad de miembros en el grupo, la falta de un cliente específico, y el riesgo del proyecto debido a la falta de conocimiento de la tecnología NFC. Por lo tanto, se diseñó una adaptación ad hoc acorde:

- Se optó por adoptar todos los **valores** de XP [2]: Comunicación, Respeto, Coraje y Retroalimentación. El equipo consideró que éstos eran necesarios para poder seguir adelante cualquier proyecto. La **comunicación** constante entre el equipo de trabajo, pese a que el desarrollo (en su mayoría) fue individual, fomentó el avance constante y la corrección de los sistemas, el **respeto** es fundamental en cualquier grupo de trabajo principalmente si los integrantes tienen el mismo poder al momento de tomar decisiones, sin éste, es imposible el éxito del proyecto. Para la **retroalimentación** se optó por solicitar la ayuda de terceros, principalmente para obtener un *feedback* del punto de vista de un potencial usuario del producto final, consultándole a otras personas si el producto final les parecía acorde o si tenían alguna inquietud. Se decidió no realizar una retroalimentación en cuanto al código o los algoritmos utilizados. Por último, al ser un proyecto basado en una tecnología muy poco explotada (El protocolo NFC) al momento de inicio y con necesidad de

conocimientos de electrónica; se necesitó un gran **coraje** para poder llevar a cabo el mismo de una forma exitosa.

- En cuanto a los **roles**, por cuestiones de cantidad de participantes, no se pudo respetar la metodología, por lo que los integrantes del grupo optaron por cumplir casi todos los roles, siendo de esta forma tanto desarrolladores como clientes y *testers*. El único rol no cubierto por los miembros del equipo del proyecto fue el de consultor, el cual fue adoptado (no formalmente) por distintos ingenieros en sistemas y en electrónica o especialistas en ciertas tecnologías, como desarrollo de aplicaciones móviles, infraestructura, telecomunicaciones, entre otras.
- Se utilizó el mismo ciclo de vida de programación extrema:

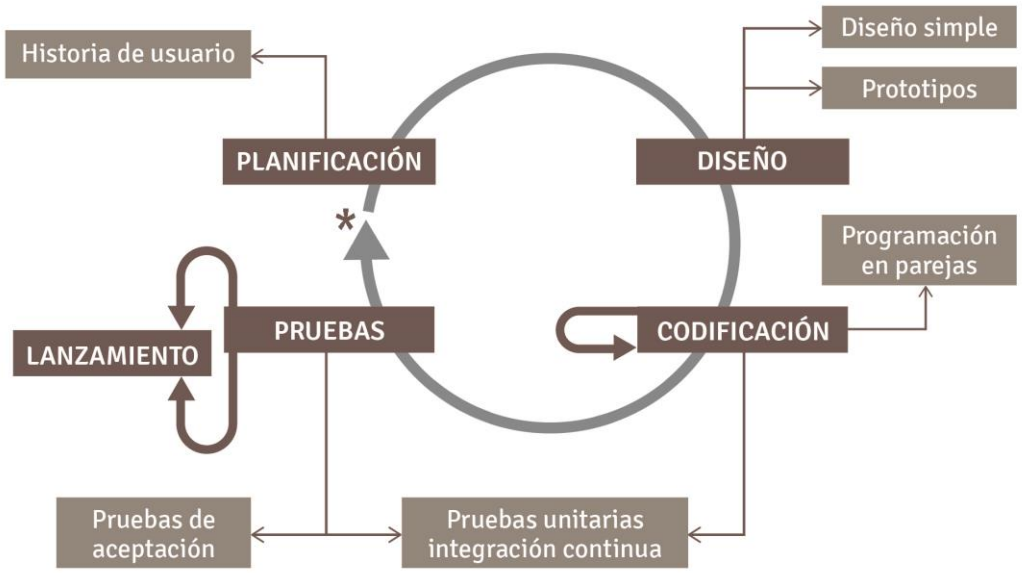


Figura 3) Ciclo de vida de la metodología XP

La etapa de **planificación** se llevó a cabo con reuniones del grupo de trabajo. En éstas, uno de los integrantes tomaba el rol de cliente, y los otros tomaban el rol de equipo de desarrollo, se recolectaban los requerimientos en formato “historia de usuario” y luego se procedía a cambiar de roles. Este procedimiento se implementó hasta recolectar todos los requerimientos necesarios.

En la etapa de **diseño**, así como la metodología XP sugiere, se encararon diseños de los componentes simples y sencillos, procurando hacer todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable en la etapa siguiente. De esta forma se consiguió que a la larga cueste menos tiempo y esfuerzo desarrollar todos los componentes e interconectarlos.

El diseño de prototipos fue un proceso reiterativo que nos acompañó en el proceso de investigación de la tecnología, para así conocerlas, entenderlas e ir dando forma a lo que luego de varias iteraciones serían los componentes finales.

Luego del dominio de las tecnologías seleccionadas durante la construcción de prototipos se procedió a la etapa de **codificación** de cada uno de los componentes, explotando las funcionalidades requeridas y principalmente buscando lograr la interconexión entre cada uno de los subsistemas.
En esta etapa el cliente es una parte más del equipo de desarrollo y en este caso como ya se mencionó anteriormente, se debieron intercambiar los roles de cliente y desarrolladores para lograr cumplir los requerimientos del sistema. Las áreas de dominio de tecnologías de cada uno de los miembros del equipo eran distintas y es así como se rotaba el rol del cliente en algún momento para establecer las condiciones requeridas por cada uno en la construcción de los demás componentes del sistema dado que debían funcionar como un todo.
XP opta por la programación en pareja ya que permite un código más eficiente y con una gran calidad, esta técnica fue adoptada por los integrantes ya que permitió a la vez ir teniendo un seguimiento de la construcción de cada uno los componentes de manera que luego sean fácilmente integrables ya que las interfaces se desarrollaban por distintos miembros. Además de esta forma se logró obtener un *feedback* continuo sobre la aceptación las funcionalidades desarrolladas entre los distintos miembros del equipo.

En la etapa de **pruebas** a medida que se fueron dando forma a los distintos componentes que forman parte del sistema se emplearon pruebas de usuario para evaluar las funcionalidades implementadas en cada iteración. Estas fueron llevadas a cabo por un miembro distinto del grupo cada vez ya que no se recomienda que el mismo desarrollador pruebe su propia implementación.
Esta práctica de usabilidad resulta irreemplazable, ya que entrega información directa de cómo los usuarios reales utilizan el sistema.
Así como el hecho de que las partes del sistema no tengan errores es indispensable, la interacción sin errores entre estas también resulta indispensable para el correcto funcionamiento del sistema. Para lograr este objetivo luego de que los componentes fueron desarrollados se realizaron pruebas de integración tanto en el *frontend* como en el *backend*.

Consideraciones sobre prácticas básicas de XP utilizadas

- En cuanto a la **planificación**, se respetó la metodología, utilizando las historias de usuario y su refinación, utilizando todos los pasos del juego de planificación de la

metodología. Además, se procedió revisando continuamente la planificación. No se utilizaron bocetos preliminares, también conocidos como *spikes* en XP, debido a que no se consideraron necesarias ya que no se contaba con un cliente, por lo tanto, la interfaz gráfica quedaba siempre a criterio del grupo de desarrollo.

- Se respetó el desarrollo iterativo, logrando así ir completando el producto de forma modular, generando incrementos continuos. En la etapa de planificación se diagramó el alcance de cada una de las iteraciones.
- Se optó por realizar pruebas de usuario, integración y de unidad en sus correspondientes tiempos.
- A lo largo del desarrollo, se realizaron las refactorizaciones de código necesarias para tener un producto de mayor legibilidad y calidad.
- Si bien el código se escribió de forma libre y compartido entre todos los miembros, no se optó por no realizar un control sobre el código ajeno, al menos que así lo solicite el creador de este. En dicho caso, el procedimiento fue solicitar al miembro del grupo que posea el mismo nivel de conocimientos sobre la tecnología utilizada.

c. Estimación del proyecto

En las etapas previas al proyecto se realizaron las siguientes actividades que darían inicio al mismo:

- Relevamiento de requerimiento y análisis de las funcionalidades.
- Definir la metodología a utilizar.
- Crear historias de usuarios.
- Elaborar un plan de contingencia.
- Redacción del plan de proyecto.

Con el comienzo del desarrollo del proyecto, se realizaron las actividades que determinaron la base para la implementación en etapas posteriores:

- Definir la arquitectura de desarrollo.
- Investigación y definición de las tecnologías utilizadas en el desarrollo del proyecto.
- Identificar y especificar la estructura de datos, entidades y atributos más importantes.
- Configurar el ambiente de desarrollo a utilizar

Luego de generadas las estructuras bases del sistema, en cada iteración se realizaron las siguientes actividades:

- Analizar y planificar las tareas de implementación de historias de usuario.
- Implementar las nuevas funcionalidades.
- Diseñar e implementar las interfaces de usuario.
- Testear las nuevas funcionalidades mediante pruebas de usuario y de integración.
- Revisión del avance del proyecto.
- Planificar la realización de cambios luego de la retroalimentación entre directores del proyecto y el equipo de desarrollo.

Antes de llevar adelante todas estas tareas realizamos la estimación de la implementación de las historias de usuario del proyecto que habíamos planteado.

Historia	Nombre	*Story Points*	Prioridad
A001	ABM Dispositivo Autorizado	3	Alta
A002	Sincronización de un dispositivo	4	Alta
A003	Sincronización de claves con las cerraduras	3	Alta
A004	ABM Cerraduras	6	Alta
A005	Generación de reportes	3	Baja
A006	Monitorización de las cerraduras	2	Media
A007	Apertura de cerraduras vía web	3	Baja
A008	Modificación de usuario	1	Baja
A009	Nombre del sistema	1	Baja
A010	Bloqueo de cerradura	2	Baja
A011	Suspensión de dispositivo	2	Baja
M001	Estado del servicio	10	Alta
M002	Almacenamiento de claves	2	Alta
M003	Mensaje de error	3	Alta
M004	Sincronización del dispositivo	4	Alta
M005	Conexión de una cerradura a la red	5	Alta
C001	Establecimiento de la conexión	12	Alta
C002	Restablecimiento de configuración	3	Alta
C003	Reconexión automática	2	Alta
C004	Servicio de acceso	5	Alta
C005	Cierre de puerta	3	Alta
C006	Alarma de puerta abierta	3	Media
C007	Sincronización de datos de dispositivos	4	Alta
C008	Registro de actividad	2	Baja
C009	Bloqueo de cerradura	2	Baja

Tabla 1) Estimación de tiempos por historia

Se estableció una velocidad de 15 *Story Points* por iteración (3 semanas), donde un *story point* contempló un día de trabajo, y una semana laboral de lunes a viernes. De acuerdo a la velocidad elegida y la sumatoria de 90 Story Points, se planificaron 6 iteraciones que contemplaron el desarrollo del sistema.
El primer mes del proyecto se destinó a la investigación y aprendizaje de las tecnologías a utilizar, además de la adquisición de las herramientas necesarias y familiarización con ellas.
En este periodo inicial también se llevó a cabo la creación de la arquitectura base para el comienzo del desarrollo del proyecto en las etapas subsiguientes.
En la última etapa se realizó el desarrollo del prototipo de la cerradura que se extendió hasta finalizar el período planteado junto con la elaboración del informe final.
Para llevar a cabo el presente proyecto, se ejecutaron las distintas actividades mencionadas en el periodo de 6 meses, originalmente a partir del mes de octubre de 2017, pero luego de iniciada la etapa de análisis y llegadas las fechas de exámenes académicos y por cuestiones laborales decimos posponer el comienzo del mismo.

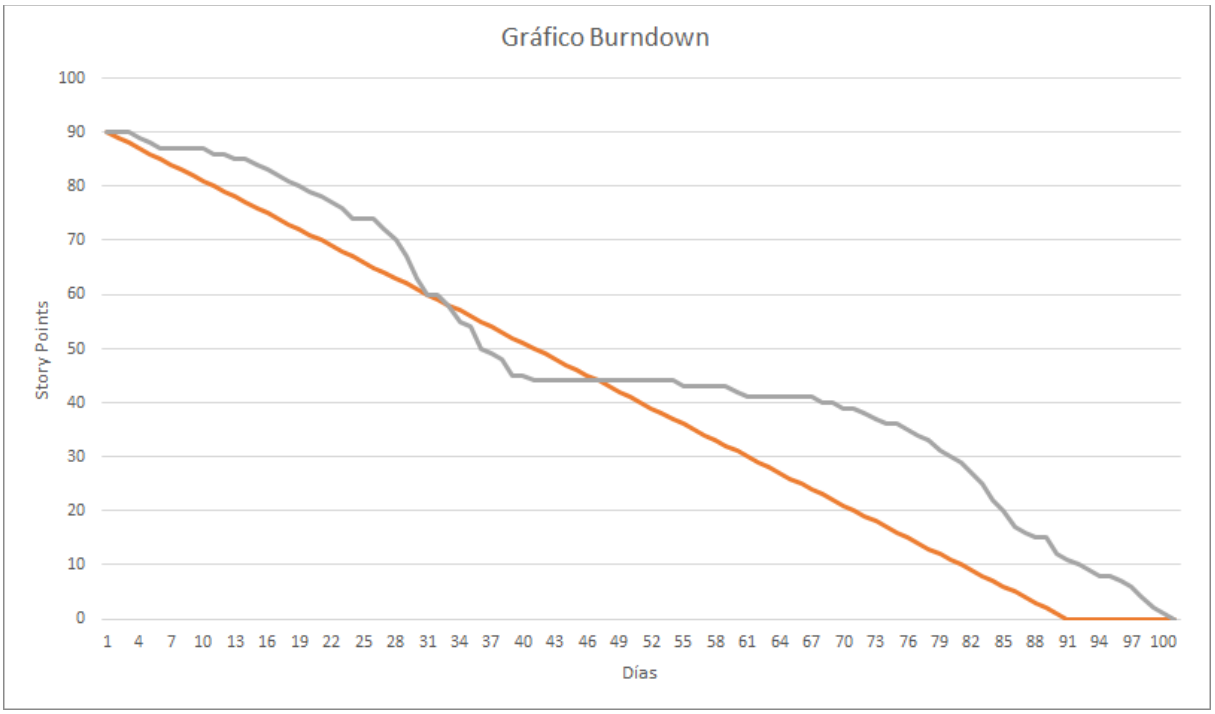


Figura 4) *Burn down chart del proyecto*

El gráfico anterior es una representación de la medición de trabajo pendiente a lo largo del proyecto que muestra la velocidad a la que se había contemplado cumplir los requerimientos planteados y su realización real.
El *burn down chart* ayudó en muchas situaciones y es utilizado por muchos equipos. Fue útil en nuestro caso por ser parte esencial para proyectos ágiles ya que permite mostrar al equipo de forma clara que está pasando con el desarrollo y cómo avanza cada iteración.
Tal como se puede observar, al comienzo del proyecto nos resultó más difícil de lo planteado llevar adelante la realización de las historias de usuario por desconocer las tecnologías y tener que tomar un ritmo de desarrollo constante. Una vez que el proyecto

estuvo encaminado y las tecnologías ya eran dominadas, pudimos comenzar a reutilizar componentes de código, lo que nos permitió ponernos al dia y ganar tiempo al proyecto.
Una vez desarrollados todos los componentes de los subsistemas, llegó la parte de la integración. Este fue otro de los momentos en el proyecto donde subestimamos el tiempo requerido como se puede observar. Luego de realizar las adaptaciones necesarias a los subsistemas se pudo lograr que estos funcionaran de forma interconectada y así avanzar con las tareas restantes tratando siempre mejorar la performance del equipo para recuperar el tiempo.
Finalmente se decidió que no era de alta exigencia el cumplimiento de la fecha de entrega pautada, al tratarse de un desarrollo propio y no para un cliente externo, por lo que nos enfocamos en la prioridad de tener un producto funcional, aunque el proyecto se extendiera un par de semanas.

d. Herramientas administrativas

Se utilizaron las siguientes herramientas para la coordinación del grupo en el proyecto:

Trello: trello.com
Es una herramienta para la organización de tareas dentro de un proyecto. Ideal para la coordinación de equipos de trabajo. Permite la distribución de tareas de mediante tarjetas y etiquetas.

Bitbucket: bitbucket.org
Es un software web Utilizado para control de versiones que permite el almacenaje utilizando el sistema Git. Facilita la distribución y la creación de código con comentarios en línea y solicitudes de incorporación de cambios

Google Drive: drive.google.com
Permite acceder a los archivos del proyecto desde cualquier lugar gracias al almacenamiento seguro en la nube y a la creación de copias de seguridad. Se utilizó para almacenaje de archivos extra (imágenes, textos, información de apoyo, etc.).

Google Docs: docs.google.com
Es un procesador de textos online que te permite crear documentos y darles formato, así como trabajar con otros usuarios de forma remota al mismo tiempo.

Microsoft Visio: www.microsoft.com
Software para dibujo vectorial utilizado para digitalizar los diagramas de flujo, gráficos de organización del proyecto entre otros.

e. Herramientas de desarrollo

Los siguientes programas se utilizaron para la programación de los distintos componentes del sistema.

Android Studio: developer.android.com
Es el entorno de desarrollo integrado oficial para la plataforma Android de aplicaciones móviles.

Visual Studio Code: code.visualstudio.com
Editor de código fuente. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código

FileZilla: filezilla-project.org
Es un gestor de archivos vía FTP para la administración de archivos en un servidor.

mRemoteNG: mremoteng.org
mRemoteNG es una bifurcación de mRemote: un administrador de conexiones remotas de código abierto, con pestañas y multiprotocolo. mRemoteNG agrega correcciones de errores y nuevas características a mRemote.

4. Desarrollo

a. Sistema

En esta etapa se comenzó diseñando la arquitectura del sistema, seleccionando las tecnologías que se iban a trabajar, y los medios y métodos de comunicación que se utilizarían entre dispositivos.

i. Tecnologías Analizadas

En primera instancia se determinó cuáles eran las necesidades que se deberían cubrir y las distintas alternativas disponibles que existían:

Hardware

Computadoras

Se requería seleccionar un dispositivo para el servidor central capaz de proveer servicio web, base de datos y comunicación con los dispositivos, además de ser económico y pequeño. Para esto se eligió utilizar un ordenador de placa reducida o SBC (por su nombre en inglés *Single Board Computer*), por lo tanto, se estudiaron las siguientes opciones:

Arduino [3]**:** el hardware consiste de un microcontrolador Atmel AVR, conectado bajo la configuración de "sistema mínimo" sobre una placa de circuito impreso a la que se le pueden conectar placas de expansión (*shields*) a través de la disposición de los puertos de entrada y salida presentes en la placa seleccionada.

- Ventajas:
    - Es una placa de muy bajo costo, con capacidad de procesamiento adecuada y de programación simple.
    - Tiene un bajo consumo eléctrico y proporciona un rendimiento de procesador integrado con periféricos.
    - El entorno de desarrollo ajustado al hardware proporciona una interfaz de programación con varias bibliotecas que facilita en gran medida el trabajo del desarrollador.
- Desventajas:
    - En relación a este proyecto, poseen una capacidad de procesamiento muy ajustada para prestar servicios como los que se requieren para la administración.
    - Se considera que son muy limitadas en cuestiones de seguridad, un aspecto muy importante para la viabilidad del proyecto.

Raspberry Pi [4]**:** es un ordenador de placa reducida de bajo coste desarrollado en el Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de informática en las escuelas. Posee un conjunto de pines de entrada y salida de información que le otorgan la capacidad de comandar dispositivos electrónicamente y comunicarse con otras computadoras.

- Ventajas:
  - El alcance de este dispositivo es muy elevado en cuestiones informáticas.
  - Posee la compatibilidad con cualquier software que funcione con una arquitectura ARM debido a su procesador.
  - Posee mayor capacidad computacional con respecto a la competencia, debido a que cuenta con mayores recursos.
- Desventajas:
  - El costo es levemente más elevado
  - El consumo energético es mayor.

Comunicación

Otra elección por tomar fue con qué tecnología de comunicación se va a trabajar para comunicar los dispositivos con las cerraduras. Esto incluye cuestiones de compatibilidad y comodidad no sólo para la comunicación física, sino también para el protocolo necesario para el funcionamiento de la tecnología. Así, se llegó a dos alternativas: Bluetooth y NFC.

Bluetooth [5]**:** Tecnología de comunicación inalámbrica por radiofrecuencia que utiliza la banda ISM de los 2.4 GHz, que permite un alcance de por lo general 10 metros.

- Ventajas:
  - Es compatible con una mayor cantidad de dispositivos, y las personas que los utilizan ya tienen conocimiento de cómo funciona una comunicación Bluetooth.
  - La incorporación de bluetooth en la plaqueta a elegir para la cerradura es muy sencilla, incorporando simplemente un adaptador USB si es que la plaqueta no incluye el soporte Bluetooth de fábrica.
- Desventajas:
  - El tiempo de establecimiento de comunicación puede ser un problema, debido a que se deben emparejar los dispositivos previo al envío de mensajes. El emparejamiento es el punto de fallo más común para comunicaciones Bluetooth y podría significar dejar encerrado a un usuario.

- Si bien tiene un alcance de varios metros, no es el suficiente para que el usuario pueda accionar la cerradura desde todo el edificio, por lo que se tendría que acercar de igual manera a la puerta. Esto hace que el mayor alcance no sea significativo.

NFC [6]**:** Corresponde a su nombre en inglés *Near Field Connection* o comunicación de campo cercano, es una tecnología de comunicación inalámbrica de corto alcance (los dispositivos tienen que estar prácticamente en contacto) que trabaja con frecuencias de 13 MHz.

- Ventajas:
  - El corto alcance agrega al sistema la seguridad de que el usuario que está accionando la puerta debe estar lo suficientemente cerca para estar en contacto con la cerradura.
  - El establecimiento de la conexión consta en simplemente identificar el tipo de dispositivo emisor con el que se está interactuando para iniciar un procedimiento dentro del dispositivo receptor.
- Desventajas:
  - Como una comunicación NFC no es algo comúnmente usado, no existen plaquetas que ya vengan con soporte de fábrica y se debe incorporar a partir de un módulo aparte.
  - No todos los dispositivos disponibles en el mercado son compatibles con esta tecnología o al menos de la manera necesaria para la utilización de este tipo de sistemas.

Software

Se necesitó desarrollar la lógica de las aplicaciones del módulo de administración central en el servidor, el software embebido de la cerradura y una aplicación móvil, por lo que se estudiaron las distintas tecnologías vigentes, haciendo una comparación tanto de tecnologías ya conocidas y trabajadas por los integrantes, así como aquellas nuevas que nos interesaba aprender y utilizar.

Módulo de administración central

PHP: Lenguaje de desarrollo web de contenido dinámico.

- Ventajas:
  - Ya se tenía conocimiento en esta tecnología.
  - Es lo suficientemente robusta para la aplicación que necesitamos.
- Desventajas:
  - Está perdiendo mercado en el ámbito laboral, por lo que se prefería utilizar una tecnología más actual y comercial.
  - Su implementación para intercambio de mensajes en tiempo real con el navegador es más compleja que con otras tecnologías analizadas.

Python: Lenguaje interpretado con énfasis en eficiencia y código legible. Para su utilización en una plataforma web, se utilizan librerías como Django y Flask.

- Ventajas:
  - Se tenía cierto conocimiento de esta tecnología, aunque no particularmente de las librerías necesarias para su utilización en esta plataforma.
  - Amplia cantidad de información, soporte y tutoriales de utilización de esta tecnología disponibles en Internet.
- Desventajas:
  - El uso de librerías como Django obliga a una estructuración específica para el proyecto, que para un desarrollo de nuestras dimensiones (muchos desarrollos pequeños en lugar de uno grande) complejizaba demasiado su utilización.
  - El hecho de que sea un lenguaje dinámico e interpretado dificulta el desarrollo de pruebas y *testing*.

Java: plataforma para la ejecución de aplicaciones sobre una máquina virtual que hace que su implementación sea indistinta a cualquier sistema operativo que se utilice. Para el desarrollo web existen librerías como Spring.

- Ventajas:
  - Ya se tenía conocimiento en esta tecnología.
  - Muy utilizada en el mercado y con amplia cantidad de soporte.
- Desventajas:
  - Al tener tantas capas de abstracción, se vuelve complejo el programa a ejecutar en la solución que se elija para soporte físico, produciendo demoras que serían inaceptables para el tiempo de respuesta requerido para el proyecto.

- La excesiva verbosidad del lenguaje hace contemplar en su lugar soluciones como *Kotlin*.

JavaScript: lenguaje dinámico de compilación *just-in-time*. Si bien es conocido por ser el lenguaje de scripting para la web, también se puede utilizar para otras aplicaciones de *backend* fuera del navegador, como Node.js.

- Ventajas:
  - Su utilización en conjunto con Node.js significaba que sólo se necesitaba utilizar un lenguaje para tanto la web como el servidor, de manera de tener una base de código unificada.
  - Es muy utilizada actualmente en el ambiente laboral y tiene altos estándares en medida de soporte y herramientas ofrecidas por la comunidad.
- Desventajas:
  - El hecho de que sea dinámico obliga a utilizar soluciones como TypeScript para poder desarrollar patrones de diseño y programación basados en un lenguaje estático.

Servidores web

El servidor Web es el programa que utiliza el protocolo de transferencia de hipertexto, HTTP [7] (*Hypertext Transfer Protocol*), para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes.

En cuestiones menos técnicas, el servidor web no es más que un software con procesos en espera de que algún usuario conectado a él solicite el contenido que almacenan. Al llegar una solicitud, el servidor web, genera los procesos propios para entregar el contenido por medio de protocolo HTTP al solicitante según los permisos que éste tenga sobre los archivos, por lo cual si el sitio que se solicite solo está disponible para usuarios que se identifiquen ante el sistema quien no esté en la lista de esos usuarios no podrán alcanzar el contenido solicitado.

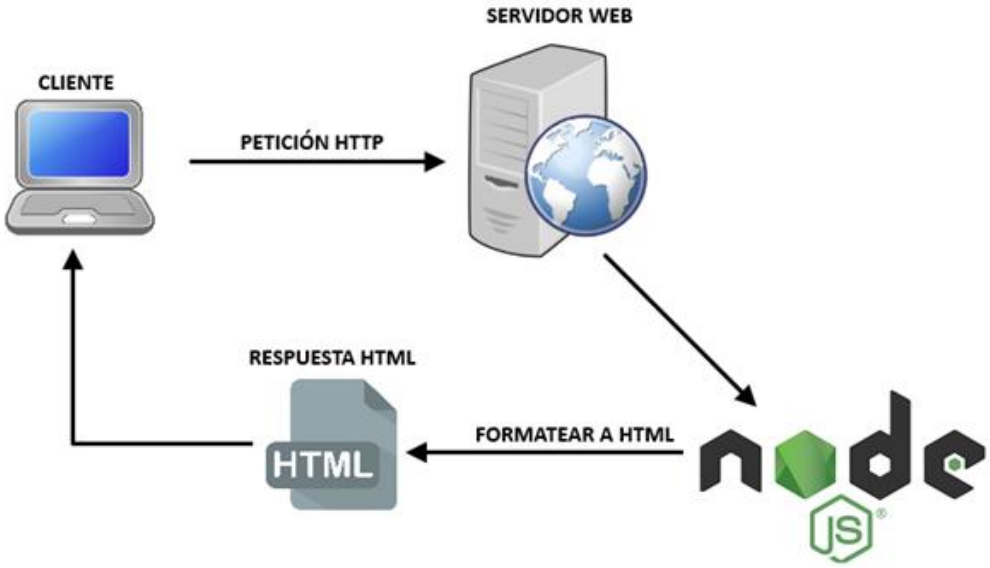


Figura 5) *Esquema de comunicación cliente-servidor mediante nodeJS*

Las consideraciones al elegir un servidor Web incluyen cuán bien funciona con el sistema operativo y otros servidores, su capacidad para manejar la programación del servidor, las características de seguridad y las herramientas particulares de publicación, motor de búsqueda y creación de sitios que vienen con él.
La aplicación web debía correr en un entorno que soporte la tecnología de desarrollo seleccionada previamente, por lo que se estudiaron las siguientes opciones:

Apache: es un servidor web HTTP de código abierto multiplataforma, actualmente es el más utilizado en la comunidad.

- Ventajas:
  - Es de código abierto, es además software gratuito.
  - Es multiplataforma (Windows, Linux y Unix).
  - El *stack* LAMP (Linux, Apache, MySQL y PHP) lo popularizó muchísimo durante el auge de las aplicaciones hechas en PHP desde el año 2000 en adelante.
- Desventajas:
  - Bajo rendimiento cuando se reciben miles de *requests* simultáneos en procesamiento de pedidos de contenido dinámico o archivos estáticos.
  - Ha quedado rezagado por su arcaica arquitectura versus nuevas y mejores opciones como Nginx o NodeJS.

NginX: pronunciado como "*engine-ex*", es un servidor web de código abierto que, desde su éxito inicial como servidor web, ahora también es usado como proxy inverso, caché de HTTP, y balanceador de carga.

- Ventajas:
  - Configuración simple para integrarse nativamente con casi cualquier tecnología y lenguaje de programación moderno.
  - Es multiplataforma (Windows, Linux y Unix).
  - Es ideal para despachar archivos estáticos y también dinámicos.
  - Se destaca por consumir muy pocos recursos bajo entornos de muchas visitas simultáneas
- Desventajas:
  - No soporta los archivos ".htaccess" del clásico Apache, aunque incluye su propio lenguaje de *rewrites*.

Microsoft IIS: *Internet Information Services*, es un tipo de servidor web creado por Microsoft específicamente para su plataforma de sistemas operativos Windows. Tuvo su origen en el viejo «*Option Pack*» que corría en Windows NT, pero luego dada su creciente popularidad se integraría con Windows Server 2003, Windows Server 2008 y en posteriores ediciones.

- Ventajas:
  - Permite el procesamiento de páginas desarrolladas en tecnología ASP / ASP.NET, y permite interpretar páginas programadas en Perl o PHP.
  - No es sólo un servidor web, sino también una suite de servicios para la web, ya que ofrece también servicios de SMTP y FTP.
  - Se integra naturalmente con Microsoft Azure.
- Desventaja:
  - Es un servidor web propietario exclusivo de Windows, y carece de integración para tantas tecnologías y lenguajes como otros servidores.

Node.js: también conocido simplemente como "Node", es un entorno de desarrollo *open source* para Javascript basado en eventos que corre desde el lado del servidor. Utilizando como base el motor V8 de Javascript desarrollado por Google en Chrome [8]. El entorno de Node.JS ha desarrollado un conjunto amplio de librerías comparables a las de otras plataformas además de eliminar funcionalidades que en el entorno de servidor no tenían sentido como por ejemplo el uso de *Document Object Model*.

- Ventajas:
  - Se caracteriza por ser ágil y rápido para crear aplicaciones que demandan sobre todo rápida interacción con el usuario.

    - Es una buena opción para servidor real-time con sockets.
- Desventajas:
    - Muchos opinan que no es un web server en sí, por sus capacidades para despachar contenido directo por el puerto 80.
    - No es bueno para servir archivos estáticos.

Bases de Datos

MySQL/MariaDB: "[…] ha sido desde hace décadas el líder indiscutido de bases de datos utilizadas en desarrollo web y de aplicaciones que utilizan lenguajes populares como PHP, Ruby o Python." (Borges en Tutoriales de Hosting) [9]

Es un servidor de bases de datos de tipo relacional, es considerada por muchos como la base de datos más popular del mundo.

- Ventajas:
    - Se caracteriza por la ejecución de tareas en simultáneo tanto lectura como escritura.
    - Es software libre licenciado bajo GNU/GPL, ofrece gran velocidad de acceso a los datos y soporta múltiples motores de almacenamiento como MyISAM e INNODB.
    - Permite uso de índices, múltiples transacciones, balanceo de carga, *clustering*, *backups* en caliente, etc.
- Desventajas:
    - Difícil de escalar
    - No es un desarrollo impulsado por la comunidad de desarrolladores.

PostgreSQL: es un servidor de bases de datos de tipo relacional, *open source* y orientado a objetos licenciado bajo la licencia PostgreSQL.

- Ventajas:
    - Gran seguridad en el almacenamiento de los datos por lo que es muy elegida en entornos empresariales y gubernamentales.
    - Se caracteriza por ofrecer una gran estabilidad, robustez y velocidad a la hora de administrar los datos.
    - Excelente forma de manejar grandes volúmenes de datos y alta simultaneidad de usuarios.
- Desventajas:
    - Es relativamente lento en inserciones y actualizaciones en bases de datos pequeñas, PostgreSQL está diseñado para ambientes de alto volumen.

- No cuenta con un soporte oficial en línea o telefónico. PostgreSQL cuenta con foros oficiales donde los usuarios pueden exponer sus dudas que responden otros usuarios de la comunidad.
- La sintaxis de algunos de sus comandos o sentencias puede llegar a no ser intuitiva si no tienes un nivel medio de conocimientos en lenguaje SQL.

Microsoft SQL Server: es el servidor de base de datos SQL relacional de Microsoft. Es muy popular entre usuarios de la plataforma Windows Server, debido a que ofrece una compatibilidad nativa con el lenguaje de programación ASP/ASP.NET, así como con toda la suite de desarrollo de aplicaciones de sistemas operativos Windows.

- Ventajas:
  - Ofrece soporte de procedimientos almacenados y transacciones.
  - Ofrece una estrecha integración con el marco .NET
  - Posee administración mediante una interfaz gráfica (GUI) que Permite el uso de comandos DDL y DML gráficamente.
- Desventajas:
  - La principal desventaja de Microsoft SQL SERVER es la gran cantidad de memoria RAM que utiliza para la instalación y utilización del software.
  - Las opciones de licencia son caras y posee una relación calidad-precio baja para este tipo de aplicación.
  - Sólo está diseñado para ejecutarse en servidores basados en Windows.

MongoDB: es a diferencia del resto de las opciones, el motor de base de datos NoSQL orientado a documentos más popular del mundo.

- Ventajas:
  - Es software libre
  - No guarda datos en tablas, sino en estructuras BSON (muy parecidas a JSON) dinámicas, algo que hace que su acceso sea rápido y fácil.
  - Se caracteriza por ser multiplataforma, corriendo sin problemas en Windows, Linux, MacOS y Solaris.
  - Otras funciones que ofrece son indexación, replicación de datos, balanceo inteligente de carga, almacenamiento de archivos, agregación de datos (similar al GROUP BY de SQL), configuración de privilegios de usuarios y encriptación por SSL/TLS.
- Desventajas:
  - La salida de datos será variable, por lo que si se necesitara una salida más estructurada no es conveniente el uso de bases de datos NoSQL.

- No es una solución adecuada para aplicaciones con transacciones complejas
- No tiene un reemplazo para las soluciones de herencia
- Aún es una tecnología joven

Lenguajes de programación

C: Lenguaje que permite control a muy bajo nivel, apreciado por la eficiencia de código que produce.

- Ventajas:
  - Permite un acceso directo a las funcionalidades de la plaqueta elegida para soporte físico.
  - Soporte, ejemplos y librerías de utilización de los módulos requeridos están mayormente en este lenguaje.
- Desventajas:
  - Al utilizar manejo de tan bajo nivel, se complejiza el código incluso para tareas sencillas.
  - Se dificulta la creación de pruebas automáticas.

JavaScript: A través de herramientas como Node.js, también se puede utilizar JavaScript para manejo interno del servidor y la cerradura.

- Ventajas:
  - Resultaría en la utilización de JavaScript para todo lo que sea relacionado con el servidor, facilitando interfaces y comunicaciones.
  - Librerías estándares de Node.js ya tienen incorporado el manejo de consultas web, facilitando su utilización.
- Desventajas:
  - Se dificulta la utilización de los módulos de la plaqueta a programar.
  - Se depende de la compatibilidad que se tenga para la plaqueta elegida con Node.js.

Python: Lenguaje de alto nivel, con soluciones incorporadas para la resolución de la mayoría de las tareas necesarias, incluso desarrollo de pruebas.

- Ventajas:
  - Se cuenta con una librería para el manejo de ciertas plaquetas, de manera que se simplifica mucho la programación de estas.
  - Cuenta con la posibilidad de ser expandido a través de módulos en código C, de manera que la funcionalidad que no sea provista por el lenguaje puede ser añadida cuando es necesaria.

- Desventajas:
    - El agregado de los módulos C es dificultoso y rompe con la forma de utilización del lenguaje.

Lenguaje para aplicaciones móviles

Flutter: Nueva solución de Google para la creación de aplicaciones multiplataforma tiene componentes ya hechos y desarrollados con tecnologías web que hacen que la creación de una aplicación sea tarea sencilla.

- Ventajas:
    - Tiene soporte directo de Google, por lo que se tiene certeza de que se va a poder seguir manteniendo.
    - Tiene un manejo más sencillo del ciclo de vida de la aplicación.
- Desventajas:
    - Si bien está orientado a un desarrollo web, se debe utilizar un lenguaje desarrollado por Google llamado Dart, que difiere de JavaScript.
    - El soporte para servicios de comunicación (NFC, Bluetooth) no es muy bueno.

React Native: Librería desarrollada por Facebook para la creación de aplicaciones multiplataforma a partir de componentes ya hechos en JavaScript y React, que luego se compilan a una aplicación nativa.

- Ventajas:
    - Las tecnologías con las que se trabaja serían las mismas con las que se desarrolla el sistema web: JavaScript, HTML y CSS.
    - Permite mayor comodidad para desarrollar diseños únicos y agnósticos de la plataforma final donde se utilice la aplicación.
- Desventajas:
    - La realización de tareas por fuera de lo que respecta a interfaces se complejiza con la utilización de esta herramienta: utilización de servicios, almacenamiento de datos, etc.
    - No tiene soporte para servicios de comunicación (NFC, Bluetooth), por lo que para utilizarlos se requiere anexar código nativo de todas las plataformas requeridas.

Android: Desarrollo de la aplicación solamente para Android con las herramientas que provee Google. Cabe destacar que con un nuevo lanzamiento de unas herramientas

denominadas *Jetpack*, el desarrollo de los módulos más complicados de Android se ha simplificado bastante.

- Ventajas:
  - Al desarrollar directamente en la plataforma se obtiene mayor performance.
  - Se puede utilizar los servicios de comunicación (NFC, Bluetooth) sin inconvenientes.
- Desventajas:
  - El diseño de interfaces es más complejo que las otras alternativas.
  - Se debe tener mayor cuidado en la utilización de funcionalidades para que sean compatibles con la mayor cantidad de dispositivos posible.

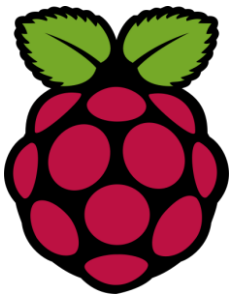
ii. Tecnologías seleccionadas

Luego del análisis de posibles infraestructuras a utilizar para el desarrollo de este proyecto, se planteó una solución basada en la utilización de tecnologías conocidas que permitan cumplir con los tiempos y objetivos del proyecto. A continuación, brindamos una breve descripción al lector de las tecnologías seleccionadas, las cuales se abordarán con más detalles en las siguientes secciones.

Hardware

Raspberry Pi

La utilización de esta tecnología está en auge ya que funciona con un sistema operativo Linux, entre otras soluciones, con el cual se facilita el rápido despliegue de las aplicaciones requeridas lo que nos llevó a seleccionar esta tecnología que a continuación describiremos más a fondo.



Raspberry Pi es un ordenador de placa única desarrollado por la Fundación Raspberry Pi en el Reino Unido con el objetivo inicial de estimular la enseñanza de informática en las escuelas.

Estas computadoras de bajo costo vienen en distintas versiones. Las versiones 3A+, 3B, 3B+, pertenecen a la última generación. Además, existen las versiones más pequeñas denominadas Zero y Zero W.

Con un tamaño de tarjeta de crédito (o más pequeño en su versión Zero), este dispositivo nos permite fabricar un dispositivo simple, manipulable y acorde al mercado de dispositivos IoT, además, el bajo costo facilita la incorporación del producto a un mercado de competencia.

Figura 6) Computadora Raspberry Pi 3B+

Como se explicó anteriormente la arquitectura planteada del sistema requiere del montaje de un servidor centralizado. Con Raspberry Pi se montará un servidor web y una base de datos, además de un servidor central para la comunicación e interacción con las cerraduras. La distintas librerías y lenguajes de programación a utilizar para el desarrollo de las distintas aplicaciones que corren sobre Raspberry Pi son públicas y gratuitas, además de presentar extensa documentación.

Por otro lado, Raspberry Pi también será utilizada en el otro componente principal de la arquitectura de la cerradura como la computadora encargada de comunicarse con el servidor centralizado y los dispositivos móviles que interactúan en ella mediante el envío de mensajes.

La cerradura tiene como requerimiento el control de dispositivos externos a la placa, como una interfaz NFC, un motor para el control del pestillo, luces indicadoras, botón, entre otros. La solución Raspberry posee un conjunto de conectores denominados GPIO (por su sigla en inglés *General Purpose Input Output* o entradas y salidas de propósito general) en los cuales se encuentran destinados algunos para entrada/salida de información, alimentación eléctrica y un grupo destinado para las interfaces UART, I2C y SPI (Estándares de comunicaciones para comunicación de dispositivos por dichas interfaces). Estos pines nos permiten conectar y controlar todos los dispositivos necesarios.

Hardware del servidor central de gestión

Se requería hardware que contará con una interfaz WiFi, Ethernet y satisfacer las necesidades básicas del software que correría sobre él (el cual se detallará en las siguientes secciones del documento) y además sea económico. Por estos motivos se seleccionó la solución Raspberry Pi 3 B+, que cuenta con las siguientes características técnicas: [10]

- Procesador: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- RAM: 1GB LPDDR2 SDRAM
- Redes WiFi:2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN
- Interfaz Ethernet: Gigabit Ethernet over USB 2.0
- Otras interfaces: Extended 40-pin GPIO header
- Almacenamiento: Micro SD
- Alimentación: 5V/2.5A DC power input

Nota: *La Raspberry Pi 3 B+, cuenta además con otras interfaces, como 4 puertos USB, salida HDMI, puerto para cámara, los cuales serían removidos físicamente de la placa para cuestiones de construcción del dispositivo.*

Se seleccionó además una tarjeta SD de 8 GB de espacio de almacenamiento para almacenar el sistema operativo y subsistemas.
Se decidió utilizar un sistema operativo llamado Raspbian, el cual utiliza un núcleo GNU/Linux, basado en la solución de la comunidad Debian.
Se definió para cada comunicación entre servidor-cerradura el protocolo de comunicación que se utilizaría (entre TCP o UDP), siempre comunicando entre scripts desarrollados en el lenguaje Python.

Hardware de la cerradura

De la misma forma que el servidor central de gestión, se utilizó la computadora Raspberry, pero debido a los requisitos de este dispositivo (inferiores al servidor), se utiliza una Raspberry Pi Zero W, con las siguientes características técnicas: [11]

- Procesador: 1GHz, single-core CPU
- RAM: 512MB
- Alimentación: Micro USB power
- Interfaz Ethernet: 802.11 b/g/n wireless LAN
- Otras interfaces: HAT-compatible 40-pin header

Además de otras características como Bluetooth 4.1, Bluetooth Low Energy, USB On-The-Go, las cuales no se utilizan.

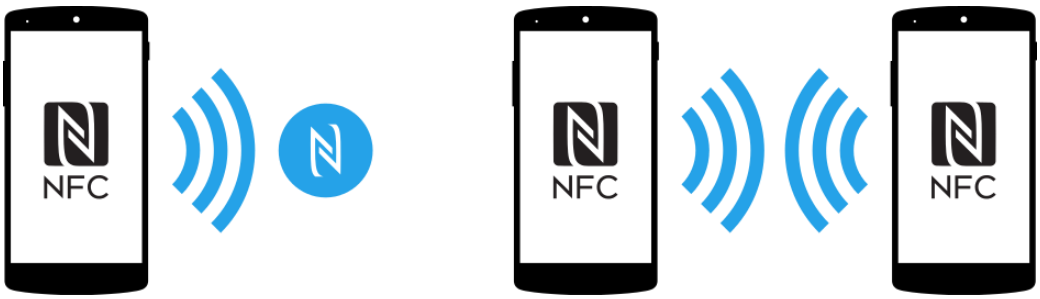
- Se comunica con el servidor central por LAN utilizando los protocolos convenientes para cada comunicación (UDP o TCP).
- Se utilizan módulos acordes para cumplir las funciones mecánicas de la cerradura. Motor, sensor de proximidad, buzzer, entre otros. El funcionamiento de la cerradura será indicado en el punto **4.b.iii** del presente informe
- Se utiliza como software de base GNU/Linux al igual que en el servidor de administración central.

Comunicación

En cuanto tecnología de comunicación que se seleccionó para vincular los dispositivos móviles con las cerraduras, se optó por NFC [12] (por *Near Field Communication* o comunicación de campo cercano) ya que provee conectividad inalámbrica de corto alcance, desarrollada a partir de tecnologías de identificación e interconexión sin contacto al estilo de las etiquetas RFID o identificación por radiofrecuencia (del inglés *Radio Frequency Identification*). Este último tal vez resulte ser más conocido como los dispositivos pequeños, similares a una pegatina, que suelen ser adheridas o incorporadas a un producto en supermercados o tiendas, y permiten el almacenamiento y recuperación de datos de manera remota.

El concepto de NFC es poder proveer un método simple de conectividad entre dispositivos que se encuentran a pocos centímetros de distancia, sin necesidad de complejos mecanismos de seguridad o establecimiento de la conexión presentes en otras tecnologías.

La comunicación es realizada a través de radiofrecuencia y opera en la banda de 13.56 MHz, que está globalmente disponible y no requiere licencia para ser utilizada.



Las transmisiones NFC son half duplex, dado que el mismo canal es utilizado tanto para transmitir como para recibir, por lo que cada dispositivo debe primero escuchar y comprobar que no haya otro dispositivo transmitiendo antes de comenzar la transmisión.

Debido a las cortas distancias, el protocolo utilizado por NFC no necesita ser exhaustivo en cuestiones de seguridad como es habitual en un protocolo de transmisión inalámbrica.

Cuando dos dispositivos NFC se aproximan, hasta una distancia máxima de 20 centímetros, ya queda establecida la conexión. Las interfaces y capas subyacentes de comunicación son estipuladas por estándares ISO.

Al tratarse de transmisiones de corto alcance, las transacciones realizadas mediante NFC son inherentemente más seguras que las realizadas por otros medios inalámbricos. Esta característica de la tecnología empleada nos aporta el concepto de seguridad requerido para el desarrollo de un sistema de cerraduras.

Tarjetas NFC

Se investigó con respecto a la tecnología NFC y las distintas soluciones electrónicas compatibles con los distintos dispositivos utilizados, y se llegó a la conclusión que la única opción era la utilización de un módulo con tecnología PN532, el cual cuenta con compatibilidad tanto con las computadoras Raspberry Pi como con los dispositivos móviles.

Por cuestiones de costos y disponibilidad se seleccionó el módulo PN532 del fabricante IteadStudio.

Software

Módulo de administración central

Se decidió utilizar JavaScript ya que su utilización en conjunto con Node.js significó que sólo se necesitaba utilizar un lenguaje tanto para la web como para las aplicaciones del servidor, de manera de tener una base de código unificada. Esto nos permitió manejar en los distintos componentes del módulo de administración como el panel de administración web, el servidor web, la API REST y el servidor de *websockets* para la comunicación en tiempo real un mismo lenguaje. Por otro lado, el tener conocimiento previo del lenguaje y sus características debido al gran crecimiento y auge de soluciones con esta tecnología en la actualidad nos llevó a inclinarnos sin duda por JavaScript.

Base de datos

En general, motores como MySQL o MariaDB son soluciones estandarizadas para la mayoría de las bases de datos pequeñas o medianas. MariaDB es el sistema de gestión de bases de datos derivado de MySQL con licencia GPL.

Nos inclinamos por esta tecnología ya que MariaDB ofrece un rendimiento bueno, es flexible y fácil de implementar. Usando buenas prácticas en las consultas y formas de almacenar la información llegamos a tener un gran rendimiento.

Cerraduras e interfaces

Para el desarrollo del sistema embebido de las cerraduras y las interfaces de comunicación con el servidor central se debió utilizar una combinación de lenguajes, debido a la variedad de componentes, servicios e interfaces que debían ser utilizadas para su funcionamiento. Se optó por utilizar Python, BASH y C, ya que cada uno nos aportaba soluciones existentes para la mayoría de las tareas necesarias. Además de que cada uno cuenta con librerías ya desarrollada para el manejo de las plaquetas utilizadas, de manera que simplificó la programación de estas. La aplicación de los distintos lenguajes se explicará con más detalle en la sección específica de la construcción de la cerradura.

Aplicación móvil

La elección de Android a nivel nativo fue indiscutible, debido a que el diseño de la interfaz tenía mucho menos prioridad con respecto al funcionamiento del servicio de comunicación.

iii. Funcionamiento del sistema

Como ya se explicó previamente, existen tres entidades principales en el sistema, las cuales utilizan los distintos subsistemas que serán explicados con profundidad en la sección **4.b**.

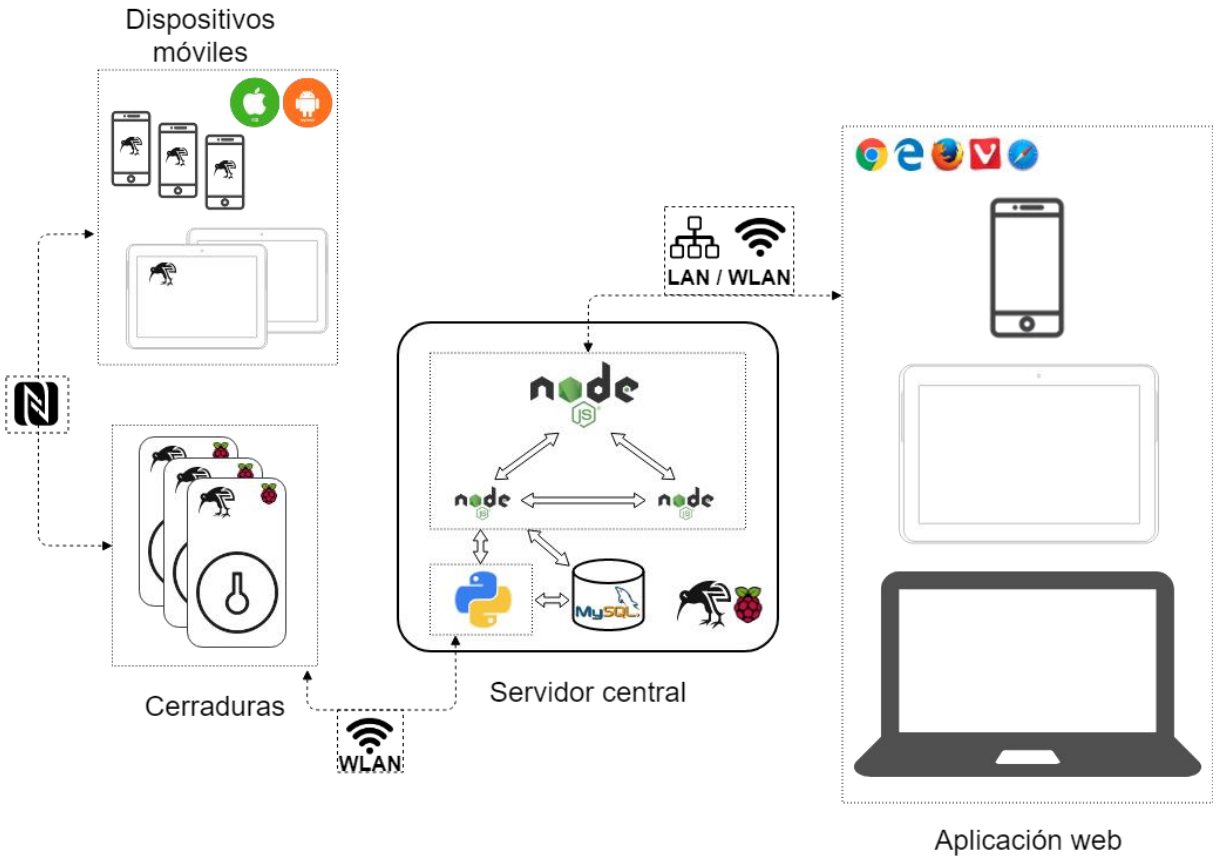


Figura 7) *Diagrama del sistema completo*

En el servidor corren los subsistemas pertinentes a la interfaz con el usuario vía web y la interfaz con las cerraduras. A su vez, estos componentes están comunicados entre ellos utilizando protocolos TCP y UDP, más precisamente utilizando librerías de cada lenguaje para el envío de información basado en estos protocolos, y HTTP para el envío de información al servidor API REST. Más detalles del servidor pueden encontrarse en la sección **4.b.i**.

Luego, el servidor central se comunica con las cerraduras utilizando la red local (LAN) como medio de comunicación, también mediante protocolos TCP y UDP, esta vez encriptando la información mediante una metodología de clave asimétrica. Se eligió este método de encriptación frente a otros, ya que se realizaría la creación de las claves públicas y privadas al momento de la fabricación de cada dispositivo, utilizando una autoridad certificante interna de la empresa fabricante, y esto daría como resultado el método más seguro para la suplantación de identidad de cualquier dispositivo integrado en el sistema. Además, se decidió no utilizar clave simétrica debido al riesgo de robo de claves que se corre al

momento del intercambio de estas, ya que se debería realizar en cada sincronización de nueva cerradura.

En las cerraduras, están constantemente corriendo servicios, teniendo un hilo principal atendiendo en un puerto TCP, dónde le llegan los mensajes del servidor. Cuando llega un mensaje del servidor, este servicio deriva al servicio correspondiente la acción, y continúa atento a cualquier otro requerimiento.

En las cerraduras, están implementados los servicios requeridos para cada uno de los requerimientos de esta, que son administrados por un hilo principal de ejecución. Éste será el encargado de interpretar los mensajes provenientes de TCP, UDP y NFC para ejecutar el servicio correspondiente. Cada uno de estos servicios será explicado en la sección **4.b.iii**.

Por último, la cerradura se comunica con la aplicación móvil utilizando el protocolo NFC por medio de la interfaz correspondiente en la cerradura, para esto, la aplicación móvil detecta que el celular posea la capacidad de utilización del protocolo. Esta aplicación, una vez instalada, permanece en ejecución en segundo plano y al momento que detecta un intento de comunicación por NFC, identifica si es correspondiente al sistema *KeyWi* y, si lo es, establece la comunicación con la cerradura. Este proceso y aplicación serán explicados con mayor detalle en el punto **4.b.ii**.

Procesos del sistema

- **Incorporación de una cerradura en una red LAN**

Antes de la incorporación de la cerradura en el sistema, se debe tener acceso a ella a través de la red local. Para este proceso, se implementó una solución que involucra a la aplicación móvil:

1. En primer lugar, se debe accionar un botón físico presente en la cerradura, el cual iniciará un script dentro de la misma que levantará un Access Point.
2. El usuario debe conectarse con su dispositivo móvil a la red transmitida por la cerradura.
3. El usuario debe ingresar en la aplicación móvil y seleccionar la interfaz correspondiente a la conexión de red, iniciando la conexión con la cerradura para su configuración.
4. El usuario debe seleccionar la red WiFi de las que se le mostrarán, visibles para la cerradura, desde la lista correspondiente.
5. El usuario debe ingresar las credenciales correspondientes a la red seleccionada.
6. La cerradura deja de transmitir su Access Point para intentar conectarse a la red especificada.

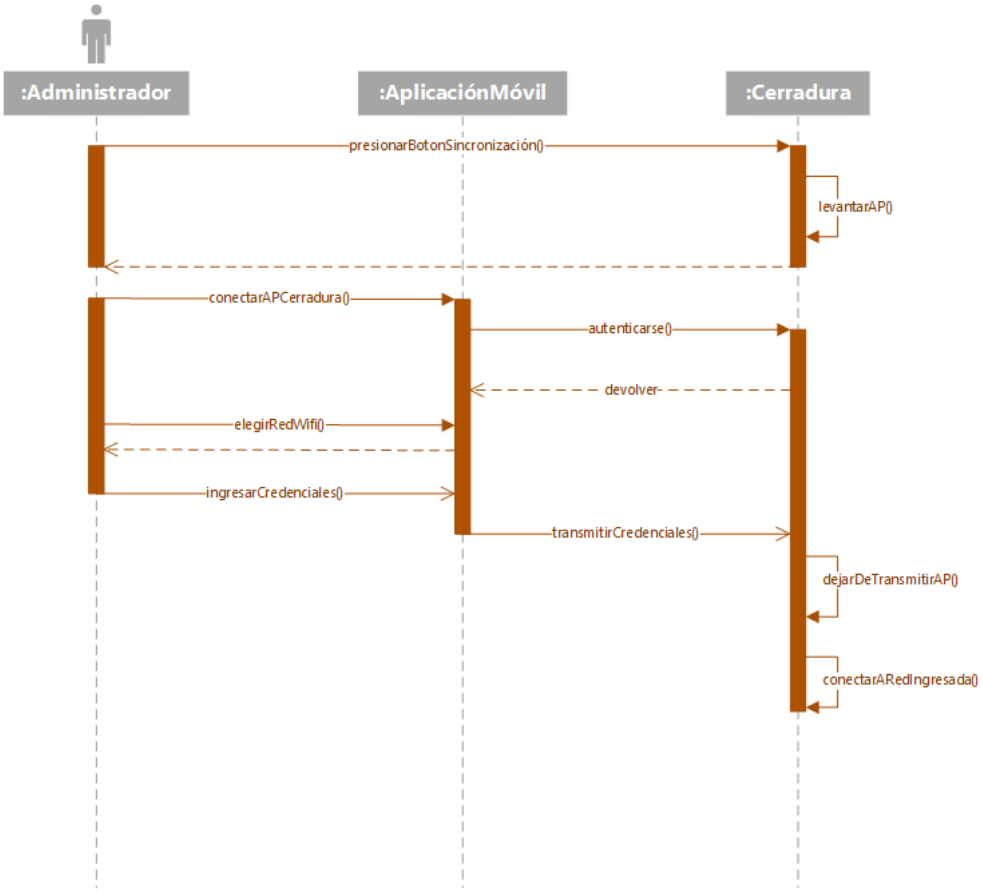


Figura 8) Incorporación de una cerradura en una red LAN

- **Alta de una cerradura al sistema**

Previo a este proceso, se debe tener incorporada en la red LAN la cerradura que se desee incorporar al sistema.

1. Para comenzar, el usuario debe seleccionar en el panel web de administración la opción de incorporación de cerraduras y el servicio web le envía al servidor el mensaje de sincronización de dispositivo.
2. El servidor comienza el proceso de escucha de cerraduras. *****
3. Una vez obtenida la lista de cerraduras existentes en la red, el servidor le envía al panel de administración la información y este la lista en pantalla.
4. El usuario selecciona uno de los dispositivos listados (éstos se identifican por código de serie) y el servicio web le envía la opción seleccionada al servidor.
5. El servidor le envía la información del sistema a la cerradura y una vez recibida la confirmación por la misma, éste la incorpora a la base de datos y le envía al usuario el mensaje de éxito a través de la interfaz web.

Al finalizar este proceso se tiene la cerradura incorporada en el sistema, ésto se podrá visualizar en el panel administrativo en la sección de cerraduras, dónde se podrá operar con ésta.

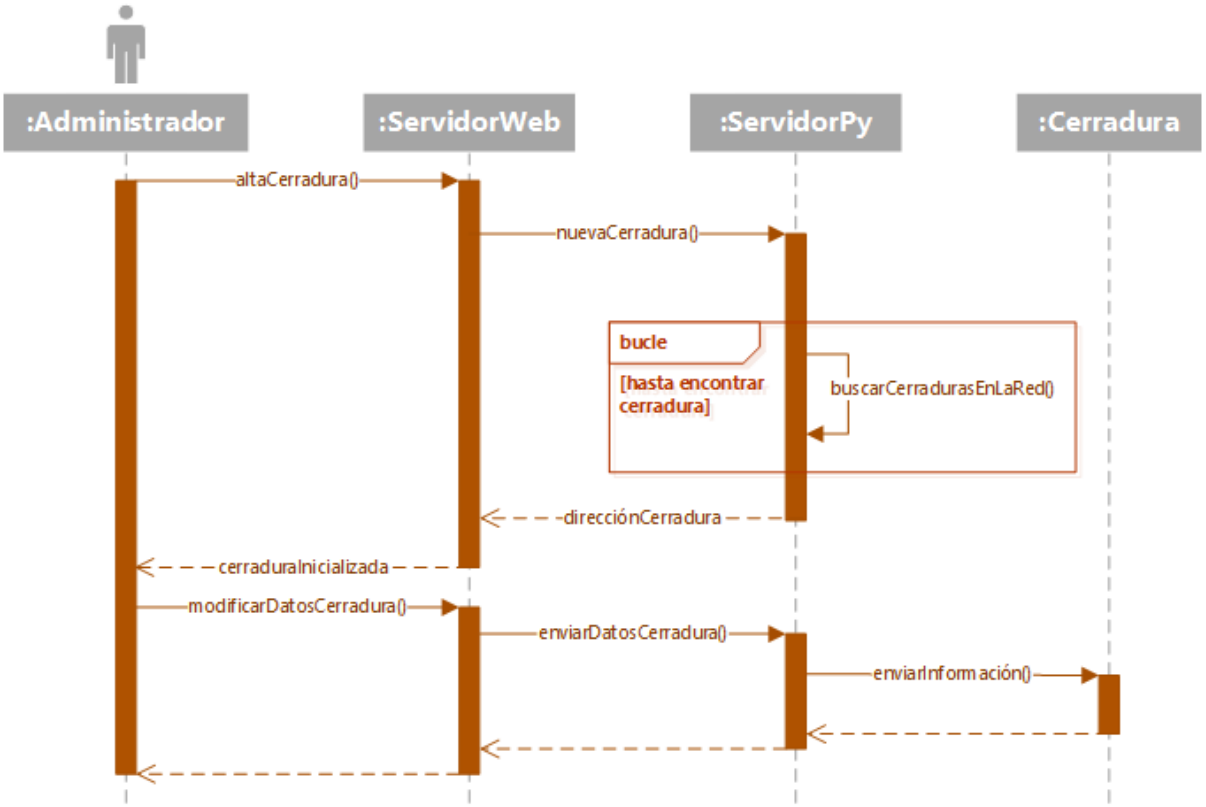


Figura 9) *Alta de cerradura al sistema*

- **Alta de un dispositivo en el sistema**

Para este proceso se involucra a la cerradura, previamente cargada al sistema, como intermediaria de comunicación del dispositivo con el servidor.

1. El usuario administrador debe seleccionar dentro del panel web de administración la opción para agregar un nuevo dispositivo.
2. Se debe seleccionar la cerradura con la que se realizará el proceso de sincronización.
3. El servidor web le envía a través del servidor Python a la cerradura un mensaje para cambiar su servicio de NFC para iniciar la sincronización.
4. La cerradura debe detener el servicio de NFC que cumple la función de la apertura.
5. La cerradura le envía al dispositivo un mensaje para establecer la conexión NFC.
6. La aplicación móvil debe, si es que no lo ha hecho antes, pedir permisos al usuario para acceder a los datos del dispositivo.
7. La aplicación recupera los datos del dispositivo y se los envía a la cerradura a través de la comunicación NFC establecida.
8. La cerradura, a través del servidor Python, retorna los datos del dispositivo al servidor web.
9. El servidor web genera una clave única para el dispositivo que son almacenados con los datos del dispositivo y enviados al mismo a través de la cerradura por la misma conexión NFC ya establecida.
10. El usuario debe ingresar un nombre para el dispositivo.
11. El servidor web actualiza el registro con el nombre ingresado.

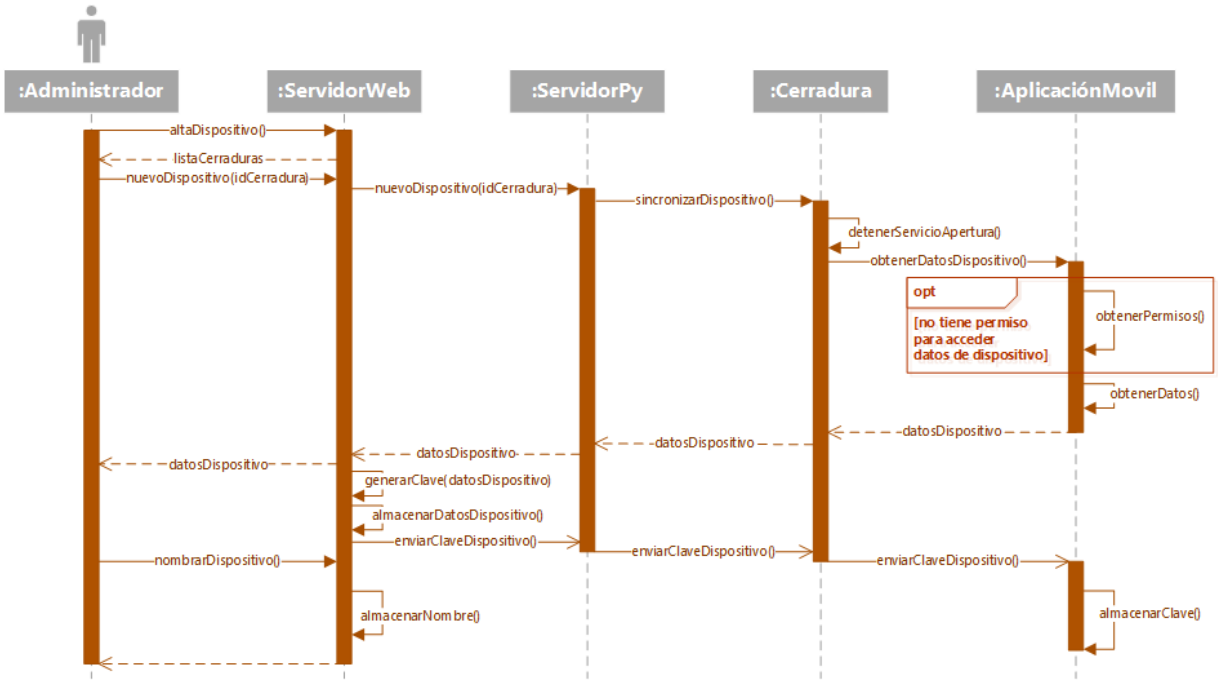


Figura 10) Alta de un dispositivo en el sistema

Cabe destacar que el envío de la clave al dispositivo, además de ser un mensaje asíncrono, no tiene una validación por parte del servidor web. Esto se diseñó de esta manera debido a que la validación se realiza al momento de almacenar los datos del dispositivo, reemplazando los datos en caso de detectar que el dispositivo ya se encuentra registrado. De esta forma, se contempla el caso tanto de que el dispositivo no se encuentre correctamente sincronizado por un error en la transmisión, ya sea porque se cayó la conexión con la cerradura (conexión TCP) o con el dispositivo (conexión NFC), así como también el caso en que el usuario borre, desinstale o modifique la aplicación en su dispositivo, perdiendo así su sincronización.

- **Reporte de estado de cerraduras**

Este proceso inicia una vez que la cerradura se encuentra incorporada en un sistema. Se repite indefinidamente con un intervalo de 3 segundos.

1. La cerradura realiza un análisis de su estado, identificando su dirección IP y sus condiciones físicas (abierta o cerrada) y le envía esta información al servidor.
2. El servidor analiza el mensaje y su contenido. En caso de encontrar una discrepancia con la información existente en la base de datos, procederá actualizar el contenido.
3. Por último, el servidor Python envía la información al Servidor web mediante Sockets para que se disponga del estado actualizado de la cerradura en el panel administrativo.

Nota: el servidor registra todos los reportes de las cerraduras en un log.

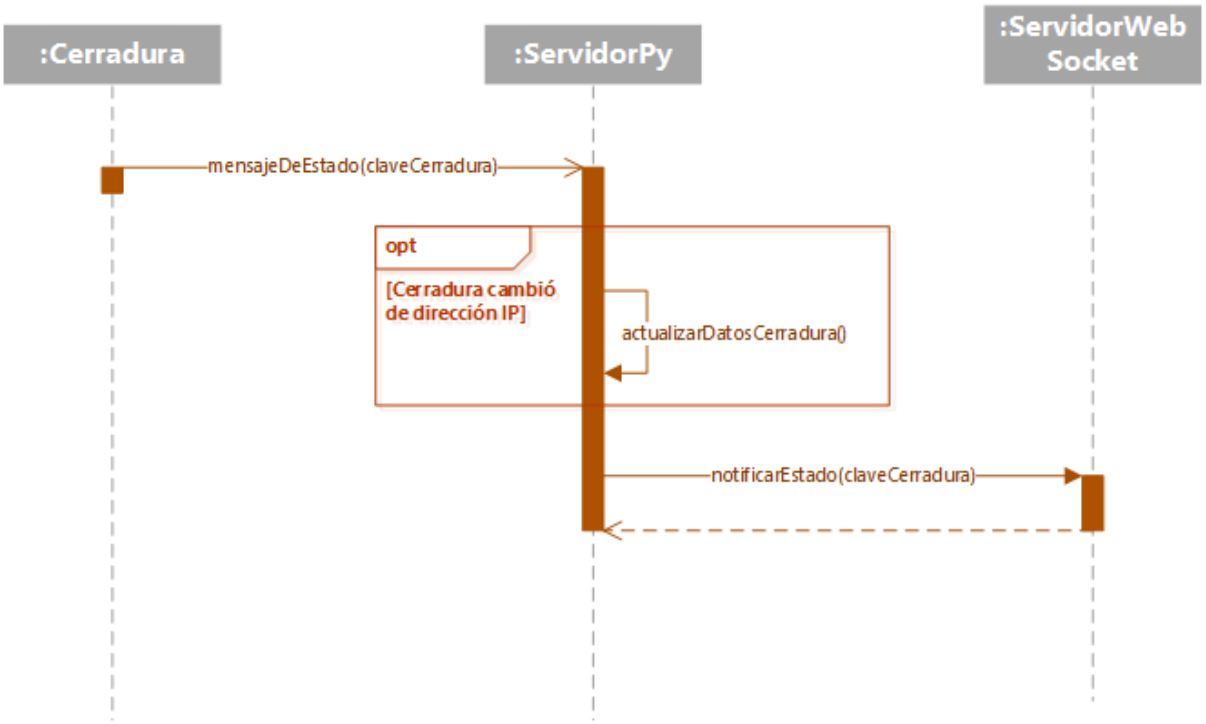


Figura 11) Reporte de estado de cerraduras

Este proceso permitirá incorporar información del estado de las cerraduras al reporte en etapas futuras del proyecto, tales como el estado de la batería, condiciones de salud del hardware, versión del software, entre otros.

- **Apertura de cerradura**

Para este proceso, es necesario que el dispositivo y la cerradura se encuentren incorporados en el sistema, y que el primero esté registrado como autorizado para la apertura de la cerradura.

Se requiere que el usuario aproxime el dispositivo móvil a la cerradura. Una vez que éste se aproxima, la cerradura detecta el campo NFC del mismo y comienza el proceso.

1. La cerradura comienza el proceso de conexión. *
2. Una vez reconocido el sistema por el celular, se establece la conexión y la cerradura envía la información del sistema.
3. La aplicación móvil procesa esta información y una vez identificado el sistema con el cual se está comunicando envía la clave que identifica el dispositivo en el mismo.
4. Por último, la cerradura valida la información recibida, y si el dispositivo se encuentra autorizado a realizar la tarea de apertura, ésta activa el mecanismo, suena el *buzzer* de la cerradura para indicar el éxito y el motor desliza el pestillo, permitiendo la apertura de la puerta.

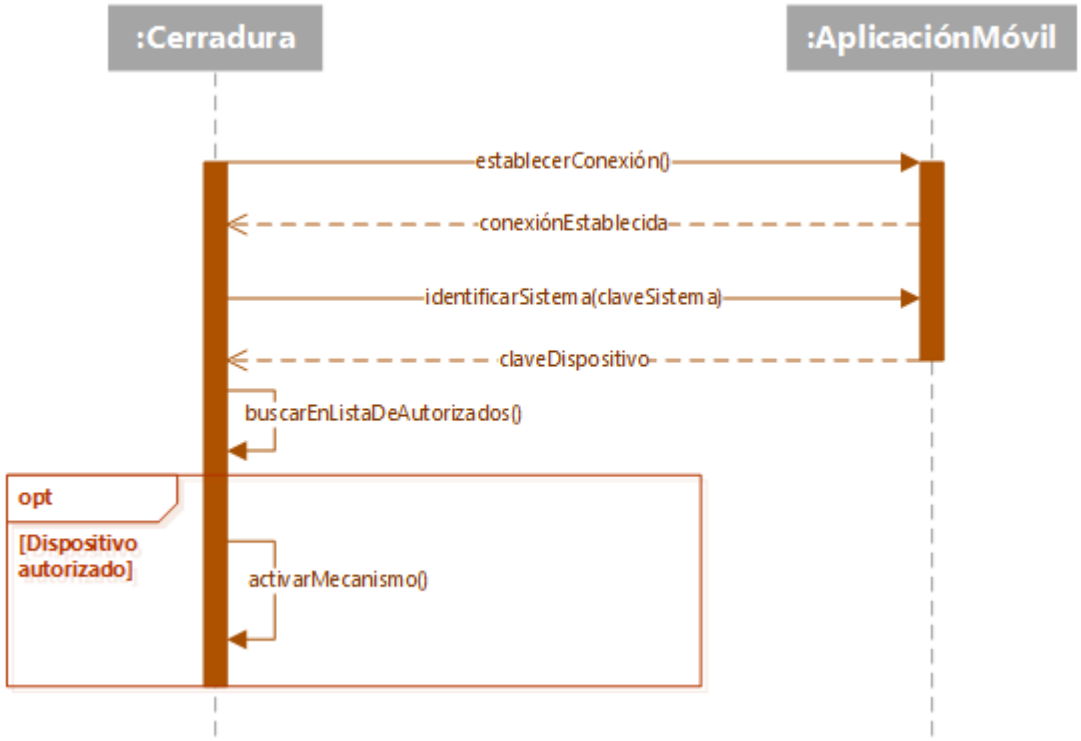


Figura 12) Apertura de cerraduras

Cabe destacar que, una vez realizado el proceso de apertura de la cerradura, la cerradura permanecerá abierta un tiempo programado por el usuario desde el panel administrativo, una vez cumplido el tiempo, la cerradura se cerrará automáticamente.

iv. Seguridad

Si bien en esta etapa del proyecto no está planificado aplicar la seguridad total requerida para este sistema, se llevaron adelante los desarrollos teniendo en cuenta diseños y tareas que permitan aplicar las condiciones necesarias básicas de seguridad, principalmente definiendo algunos aspectos en las capas de aplicación y de transporte de comunicación.

Seguridad en capa de transporte

Firewall

Para control del firewall de kernel proporcionado por el sistema operativo, se decidió aplicar la utilidad **iptables** tanto para las cerraduras como para el servidor.
Como política por defecto, los dispositivos están configurados para bloquear cualquier tipo de conexión entrante que no sea necesaria.
Por parte del servidor, en primera instancia solamente se aceptan conexiones TCP en el puerto correspondiente al panel administrativo, además aceptan mensajes UDP en el puerto destinado a detectar las conexiones de las cerraduras. Luego, cada vez que una cerradura sea incorporada en el sistema, el servidor genera las entradas correspondientes para habilitar las conexiones a los puertos pertinentes a los servicios que éste les ofrece y de los paquetes ICMP utilizados para el monitoreo de dispositivos. Además, todas las conexiones TCP estarán limitadas en cantidad por dispositivo, para poder disminuir los riesgos de ataques de denegación de servicio.
Los dispositivos cerradura tienen dos estados distintos que proteger, cuando se encuentran conectados en una red **sin** estar incorporados en un sistema, y cuando sí lo están. En el primero, la cerradura bloquea toda conexión que intente acceder por un puerto que no es el de incorporación de cerradura. Luego, cuando es adoptada por un servidor, se incorporan las reglas necesarias que aceptan conexiones con dirección de origen de éste y puertos de destino en los que se le prestan servicios.

Seguridad en la capa de aplicación

HTTPs

En primera instancia para obtener una comunicación segura a nivel de aplicación resultó necesario crear un canal cifrado entre el servidor remoto y el navegador utilizado por el cliente utilizando el protocolo HTTPS (Protocolo seguro de transferencia de hipertexto o en inglés: *Hypertext Transfer Protocol Secure* o HTTPS) para el tráfico de información. [12]

La utilización de una conexión cifrada permitió que la información sensible, como usuarios y claves, no puedan ser usadas por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que de esta forma lo único que obtendría es un flujo de datos cifrados que le resultará imposible de interpretar.

HTTPS es una extensión segura de HTTP. Los sitios web que instalan y configuran un certificado SSL/TLS pueden usar el protocolo HTTPS para establecer una conexión segura con el servidor.

Los detalles del certificado, por ejemplo, la entidad emisora y el nombre corporativo del propietario del sitio web, se pueden ver haciendo clic en el símbolo de candado de la barra del navegador.

En nuestro caso hemos optado por utilizar una herramienta para gestionar de forma automática certificados TLS/SSL y automáticamente configurar el cifrado HTTPS en nuestro servidor web.

Esta herramienta se llama Certbot[1], y permite obtener los certificados de *Let's Encrypt*[2]. Esta es actualmente una de las Autoridades Certificantes más grandes a nivel mundial, y nos permite obtener un certificado TLS/SSL para incorporar a la aplicación web el protocolo HTTPS de forma fácil y gratuita.

Encriptación de la comunicación

Se planteó como requisito de sistema que toda comunicación debe ser segura para evitar posibles ataques de hombre en el medio y/o suplantación de identidad. Como se mencionó previamente en este punto, se definió que lo más correcto es la utilización de un sistema de encriptación asimétrica.

En el momento de la creación de cada dispositivo (cerradura o servidor), se le genera a éste una clave pública y una clave privada, firmadas por una autoridad certificante interna de la empresa. Luego, una vez que se incorporan en un sistema, estos dispositivos intercambian sus claves públicas. Por último, a la hora de intercambiar mensajes, éste primero es cifrado con la clave pública del destinatario, para garantizar la confidencialidad del mensaje, debido a que sólo podrá lo desencriptar el destinatario, y luego se firma con la clave privada del emisor, esto garantiza la autenticidad del mensaje, evitando que intrusos le envíen mensajes a cualquier dispositivo. Una vez el mensaje recibido, se procede a realizar el proceso inverso, primero se comprueba con la clave pública del remitente su identidad, si éste es aceptado por la autoridad certificante, se procede a desencriptar el mensaje con la clave privada del receptor.

[1] https://certbot.eff.org/
[2] https://letsencrypt

El siguiente gráfico ilustra el envío de un mensaje desde una cerradura al servidor:

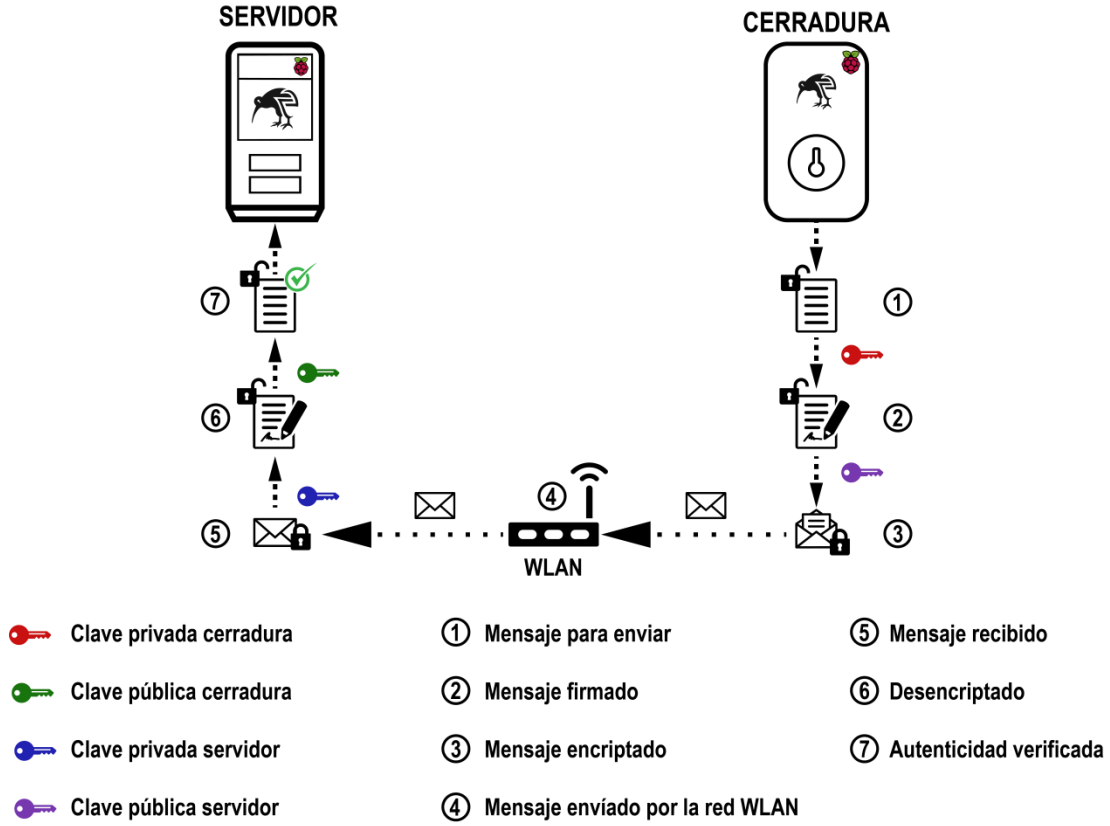


Figura 13) Esquema de encriptación de la comunicación

Este proceso es primordial en cuanto a la seguridad de los mensajes en este proyecto, ya que es la principal defensa ante los posibles ataques *man in the middle* (hombre en el medio) al sistema.

En la comunicación entre dispositivo móvil y cerradura, no se considera necesaria la utilización de este sistema debido a que existe una restricción física en el protocolo NFC que requiere que los dispositivos estén a una distancia cercana.

b. Subsistemas

Un sistema puede estar constituido por múltiples componentes y subsistemas. En este caso nuestro sistema de cerraduras es el conjunto constituido por los elementos físicos y lógicos necesarios para captar información, almacenarla y procesarla.

En la teoría de sistemas, los niveles de organización (o jerarquías) se refieren al orden en distintos niveles de los sistemas más simples a los más complejos; por ejemplo, la identificación de un subsistema, dentro de un sistema, dentro de un suprasistema.

Sin fronteras entre los requerimientos funcionales que presenta el sistema de cerradura, difícilmente se podrían establecer los subsistemas, sistemas o suprasistemas. Por lo que para esta distinción resultó fundamental establecer los límites o fronteras precisas de los sistemas de cada nivel con las funcionalidades que cada uno debería cumplir.

Nos apoyamos en la ingeniería de software basada en componentes que enfatiza la separación de funcionalidades disponibles a través de un sistema de software dado.

> En esencia, un componente es una pieza de código preelaborado que encapsula alguna funcionalidad expuesta a través de interfaces. Los componentes se unen y combinan para llevar a cabo una tarea.
>
> El paradigma de ensamblar componentes y escribir código para hacer que estos funcionen se conoce como Desarrollo de Software Basado en Componentes. El uso de este paradigma posee algunas ventajas:
>
> 1. **Reutilización del software.** Nos lleva a alcanzar un mayor nivel de reutilización de software.
> 2. **Simplifica las pruebas.** Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
> 3. **Simplifica el mantenimiento del sistema.** Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
> 4. **Mayor calidad.** Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.
>
> (Julio Casal Terreros - Desarrollo de Software basado en Componentes [2008] – Microsoft Docs) [13]

Con el paradigma de componentes no solo buscamos focalizarnos en el principio de reúso sino también atacar principalmente la mantenibilidad. El reúso es un objetivo admirable pero no es sencillo de obtener. Bajo el enfoque de componentes se buscó construir para el cambio. Los sistemas actuales cambian sus requerimientos incluso cuando el sistema ya

está en producción. El principal objetivo buscado en los componentes del sistema de cerraduras desarrollados y que exponemos a continuación no es el reúso, sino es que sean fácilmente reemplazables a futuro por una versión mejorada que responda eficientemente a las demandas tanto en funcionalidad como en seguridad.

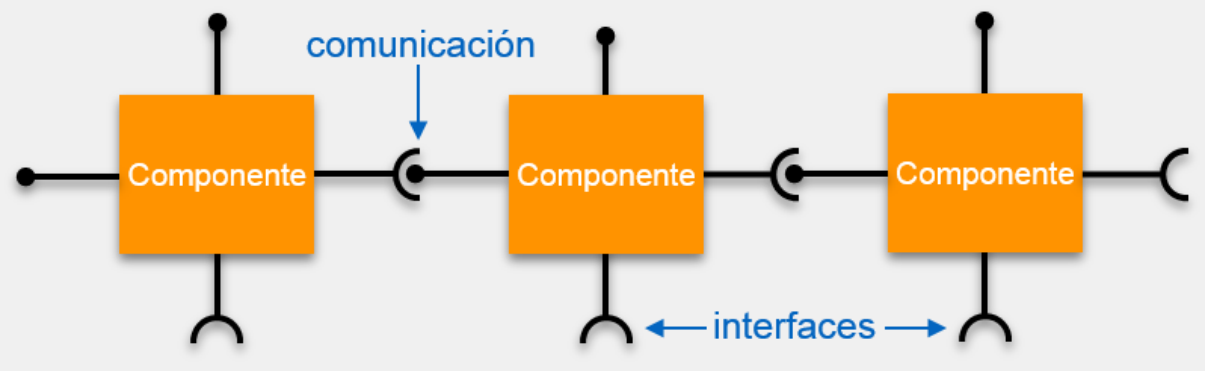


Figura 14) Esquema de componentes del sistema

i. Servidor central de gestión

Hardware

Como se mencionó anteriormente en las tecnologías seleccionadas, se requería un hardware más potente que el de las cerraduras para desplegar las aplicaciones que corren en el servidor central de gestión.
En este componente del sistema es donde se optó por utilizar una Raspberry Pi 3 B+ con un sistema operativo Raspbian que utiliza un núcleo GNU/Linux.
El hardware cuenta con una interfaz WiFi, puerto Ethernet, 4 puertos USB, salida HDMI y finalmente se seleccionó una memoria SD de 8 GB de espacio de almacenamiento para almacenar el sistema operativo y subsistemas.

Software

En este componente del sistema tiene lugar el procesamiento y almacenamiento de información propia al control y gestión de las cerraduras y dispositivos. Aquí se encuentran distintas componentes de software con roles bien diferenciados por lo que para un mejor desempeño se utilizaron distintas tecnologías y lenguajes de programación en cada uno de ellos con fin de lograr una mejor interoperabilidad.

Panel de Administración web

La aplicación del Panel de Administración, también conocida como *Frontend*, o Panel de Control, es la parte de un sitio web que interactúa con los usuarios ejecutándose en el navegador de este, por lo que decimos que está del lado del cliente.
En el panel administrativo encontramos todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.
"HTML, CSS y JavaScript son los lenguajes principales del *Frontend*, de los que se desprenden una cantidad de *frameworks* y librerías que expanden sus capacidades para crear cualquier tipo de interfaces de usuarios. Algunos ejemplos de ellos son Bootstrap, React, Redux, Angular, Foundation, LESS, Sass, Stylus y PostCSS." [14]

HTML: *HyperText Markup Language*, es un lenguaje de marcado, que funciona a base de etiquetas para la estructuración y organización del contenido de la web.

CSS: *Cascading Style Sheets*, son hojas de estilos en cascada, encargada de dar formato al contenido.

Javascript: Lenguaje de programación muy potente, orientado a objetos y desde hace muchos años usado para el desarrollo de aplicaciones web. Hoy día es de los lenguajes que más se están extendiendo y evolucionando en el ámbito del desarrollo web y de desarrollo móvil con la cualidad de poder ser interpretado en equipo cliente y en cualquier navegador web e interactuar fácilmente con HTML y CSS entre otros.

En este caso, al encontrarnos familiarizados ampliamente con el *framework* y por ser uno de los más utilizados actualmente, se decidió optar por Bootstrap. Este es un *framework* desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien.

Figura 15) Estructura de una aplicación web.

En la otra cara de la moneda, encontramos el *Backend* o Servidor Web, la cual es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El *Backend* es una aplicación especializada que entiende la forma como el navegador solicita cosas. La implementación del servidor se abordará con más detalle en el correspondiente apartado.

¿Cómo interactúan el Frontend y el Backend?

El panel administrativo web está compuesto de un montón de documentos que están conectados entre sí, a través de enlaces. El usuario escribe la dirección del panel administrativo en la barra del navegador, eso quiere decir que se le está solicitando al servidor que muestre la página web.

Finalmente, el *Frontend* recibe la información que le pasó el *Backend* y la acomoda en la interfaz del sitio, que en este caso es el panel de administración del sistema de cerraduras que se muestra en el navegador. [14]

Funcionalidades desde el Panel Administrativo

Esta interfaz es en la que el usuario responsable, con los privilegios apropiados, puede manipular y visualizar los datos inherentes al sistema.

Aquí se puede configurar nuevas cerraduras, dispositivos o bien establecer las relaciones o permisos de acceso entre ellos, así como gestionar sus estados o disponibilidad.

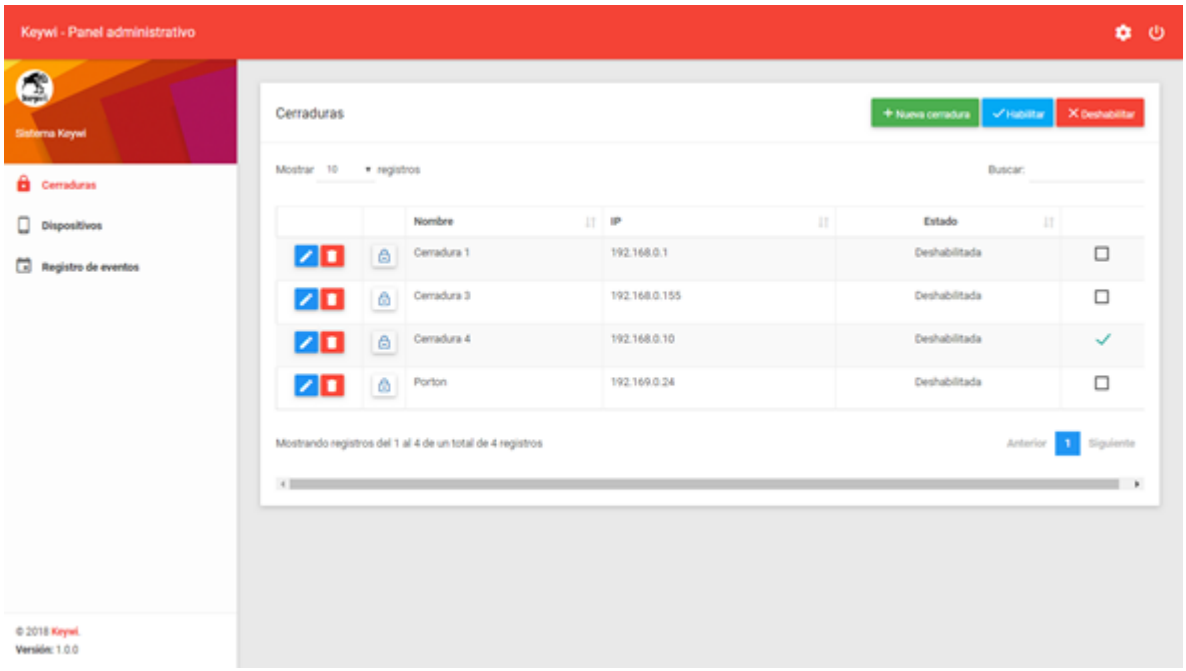


Figura 16) Interfaz gráfica del panel administrativo – Administración de cerraduras

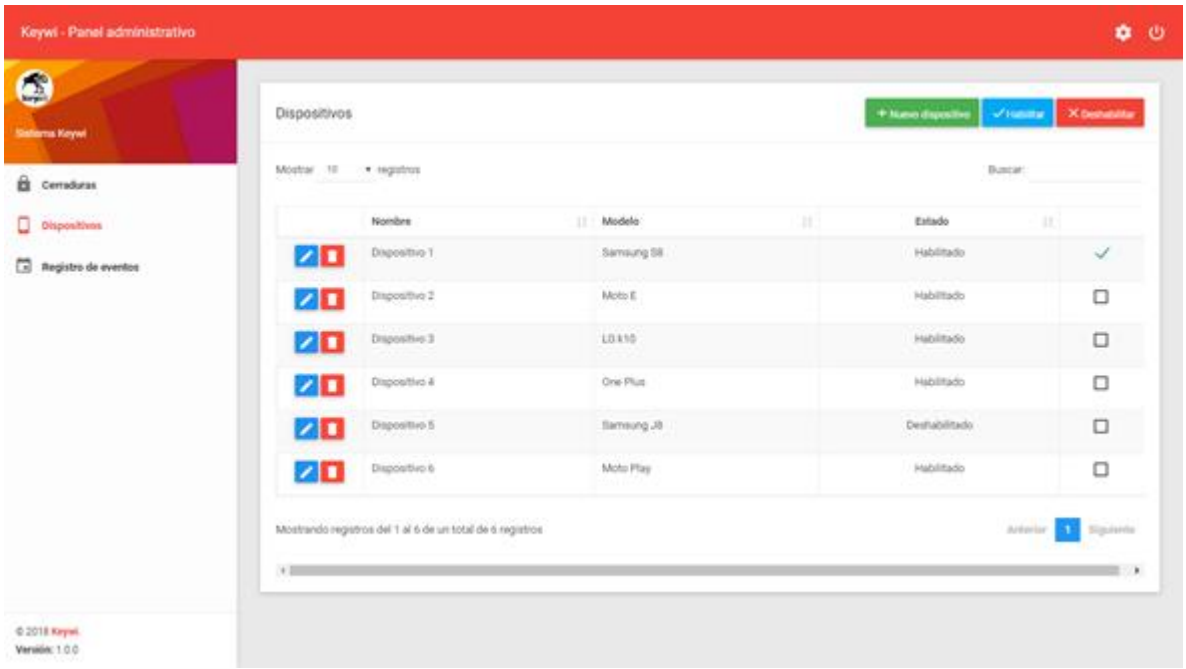


Figura 17) Interfaz gráfica del panel administrativo – Administración de dispositivos

A continuación, se presenta al lector en una breve muestra del funcionamiento y usabilidad del panel administrativo para la gestión del sistema. Para un mayor nivel de detalle sugerimos ver el refinamiento de las historias de usuario ubicadas en el Anexo.

Cerraduras

En esta primera sección, dentro de todas las funcionalidades permitidas en el CRUD de cerraduras (acrónimo de "Crear, Leer, Actualizar y Borrar" del original en inglés: *Create*, *Read, Update and Delete*), se desea destacar principalmente la funcionalidad con la que el usuario puede dar de alta las cerraduras que ya se han incorporado a la red doméstica.

Esta tiene lugar cuando el usuario una vez autenticado en el sistema presiona el botón de “Nueva Cerradura”. En ese momento se envía la petición al servidor de cerraduras para que mediante la ejecución de scripts en el servidor Pyhton y la utilización de la aplicación móvil como interfaz de conexión, se reporten las cerraduras que se encuentren en estado pasivo en la red, es decir aquellas sin haber sido registradas en el sistema aún.

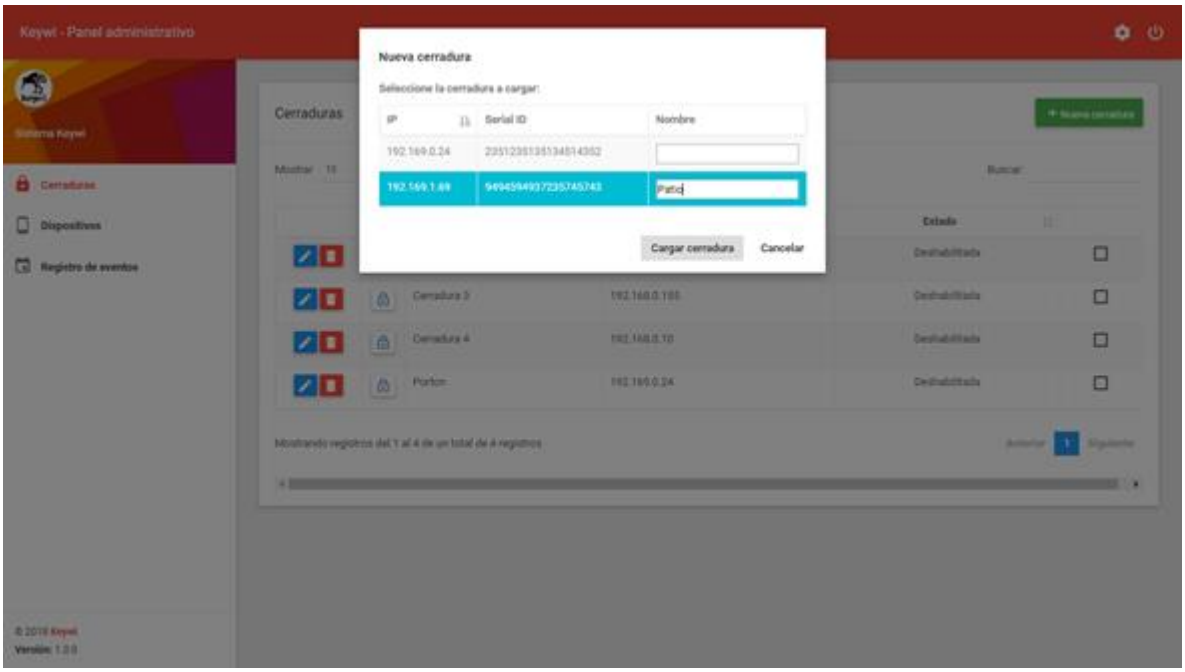


Figura 18) Interfaz gráfica panel de administración – Incorporación de nueva cerradura

Una vez recabada esta información, es procesada por el servidor y presentada al usuario en un listado en el cual podrá registrar las cerraduras encontradas en la red y que aún no han sido dadas de alta para así ingresarlas al sistema eligiendo un nombre para identificarlas.

La emisión de mensajes desde el servidor que se propagan hasta el panel administrativo cuando haya nuevas cerraduras se lleva a cabo mediante la tecnología de *websockets* que se explicará con mayor detalle en las siguientes secciones.

Dispositivos

Así como por un lado el sistema está compuesto por cerraduras encargadas de controlar el acceso a habitaciones, por el otro encontramos los dispositivos capaces de actuar como llave para accionar dichas cerraduras mediante una comunicación con el protocolo NFC.

Para poder gestionar los dispositivos del sistema que accionan las cerraduras, el usuario cuenta con un apartado propio en el panel administrativo con todas las funcionalidades CRUD como se menciona anteriormente para las cerraduras.

En este caso para realizar esta tarea, luego de seleccionar el botón “Nuevo dispositivo” el usuario deberá seleccionar una cerradura, la cual ha sido dada de alta previamente en el sistema. La cerradura seleccionada actúa como una interfaz de comunicación entre el servidor central y los dispositivos. Mediante ella se lee la información del dispositivo que es enviada por la aplicación móvil a través de mensajes NFC, permitiendo así registrar un nuevo dispositivo en el sistema.

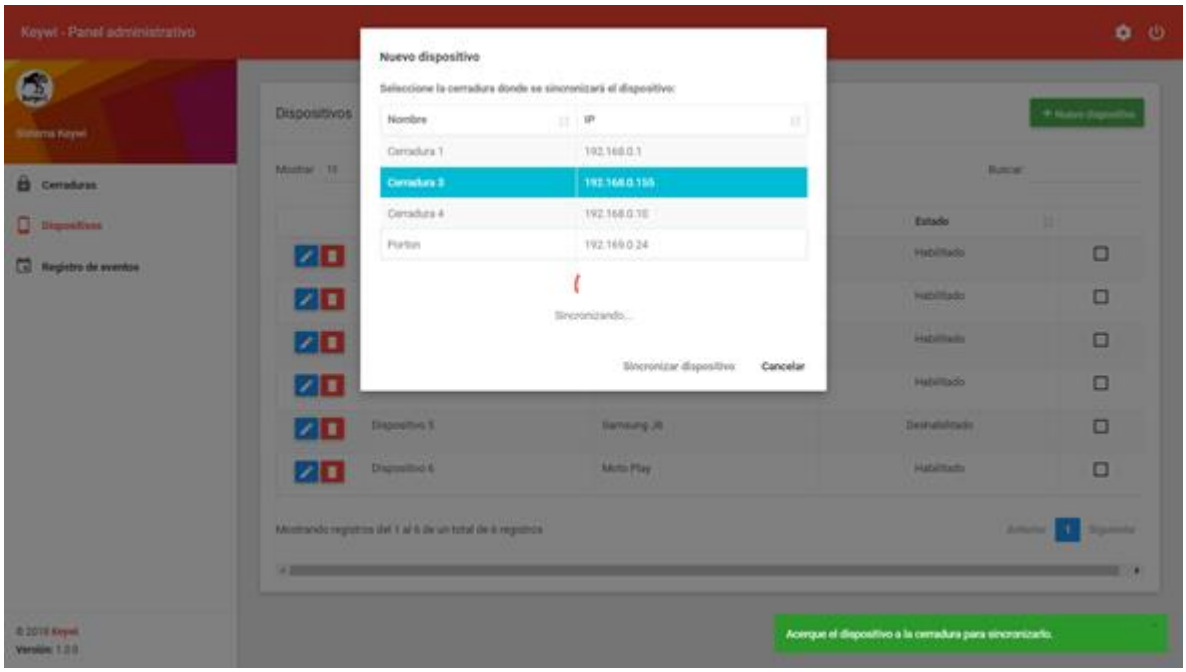


Figura 19) Interfaz gráfica panel de administración – Incorporación de dispositivo

Una vez que ha acercado el dispositivo a la cerradura seleccionada, la cual posee un led indicador parpadeante, se recaba la información (modelo e IMEI) del mismo y luego de ser procesada es presentada al usuario en el panel administrativo.

Finalmente, para incorporarlo al sistema, de la misma manera que se realiza con las cerraduras, el usuario ingresa un nombre al dispositivo a fin de poder identificarlo de manera práctica en el sistema.

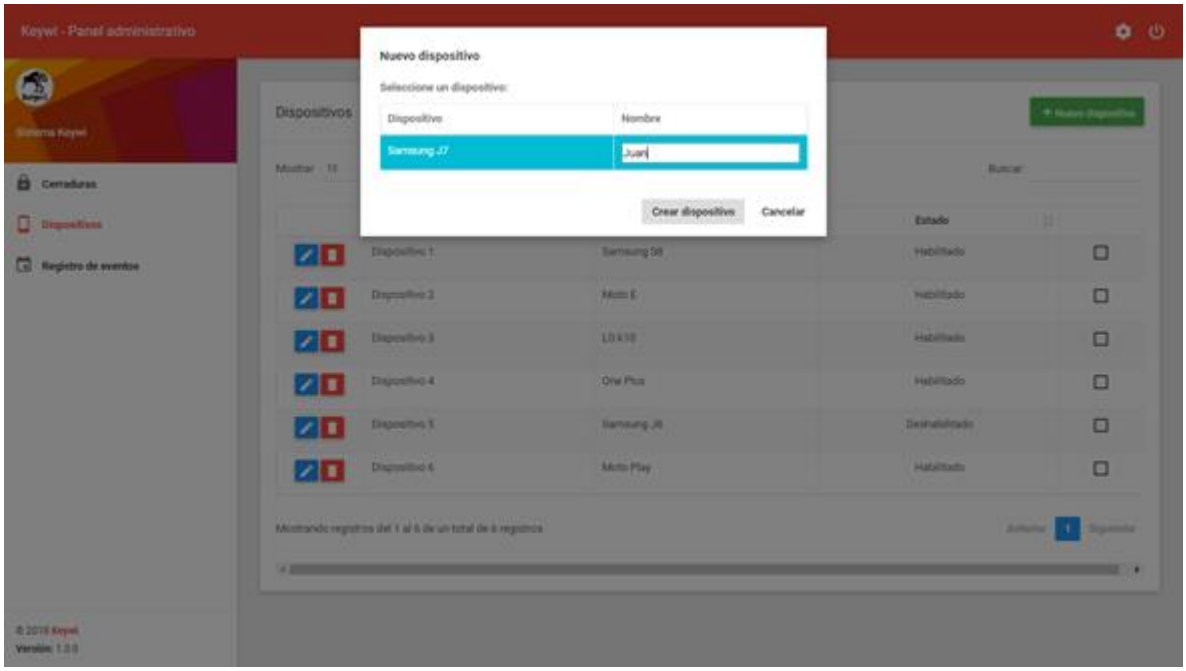


Figura 20) *Interfaz gráfica panel administrativo – Incorporación de dispositivo*

Gestión de permisos

En cuanto a los permisos de accesos, el administrador puede gestionar que cerraduras acciona un dispositivo dirigiéndose al panel de edición del dispositivo. Aquí se le permite al usuario seleccionar de un listado las cerraduras deseadas y luego se sincroniza esta información con las mismas para que puedan ser accionadas de forma autónoma sin tener que consultar al servidor en cada acción de apertura o cierre.

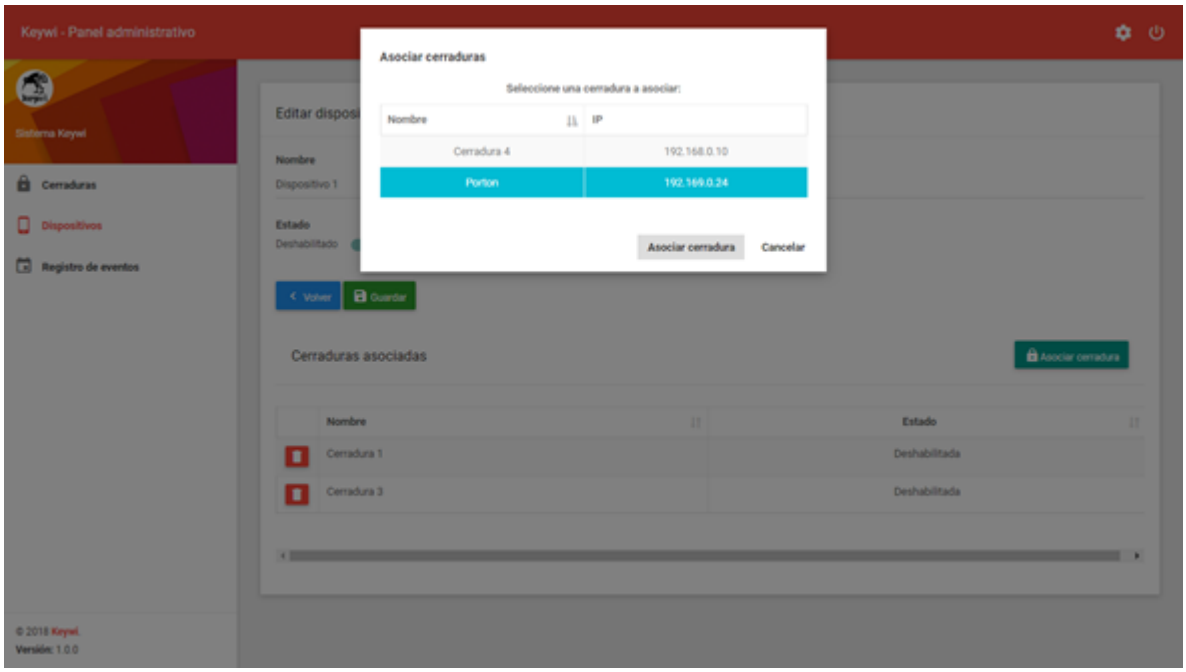


Figura 21) *Interfaz gráfica panel administrativo – agregar cerradura*

Esta misma tarea puede ser realizada desde el aspecto de la cerradura. En el panel de edición de cerraduras se le permite al usuario gestionar que dispositivos serán capaz de accionarlas siguiendo los pasos anteriormente mencionados.

Registro de eventos

La grabación secuencial en la base de datos de todos los acontecimientos (eventos o acciones) que afectan a un proceso particular constituye una evidencia del comportamiento de sistema.

La visualización de eventos del sistema también cumple un rol de interés para el usuario en el panel de administración y se provee un apartado para el seguimiento de los acontecimientos con la posibilidad de ser filtrados y exportados para un posterior análisis.

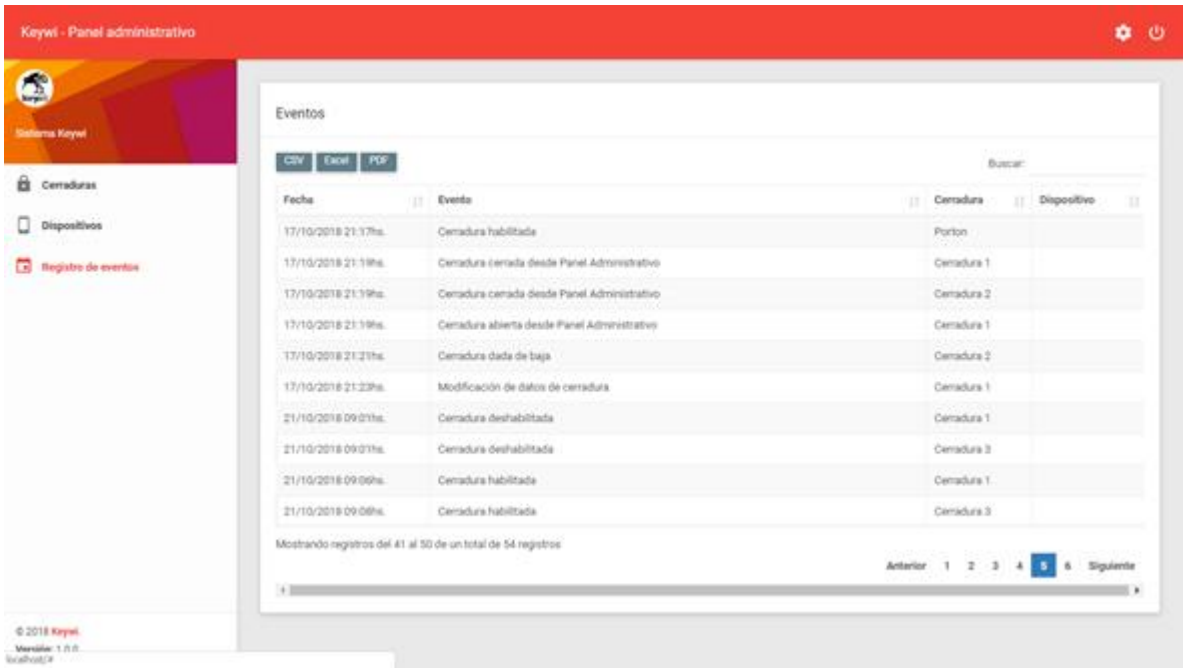


Figura 22) Interfaz gráfica panel administrativo – registro de eventos

Configuración del panel administrativo

Finalmente tenemos las configuraciones de perfil y usuario, infaltables en el panel administrativo, le permiten al usuario controlar su información personal de acceso al sistema.

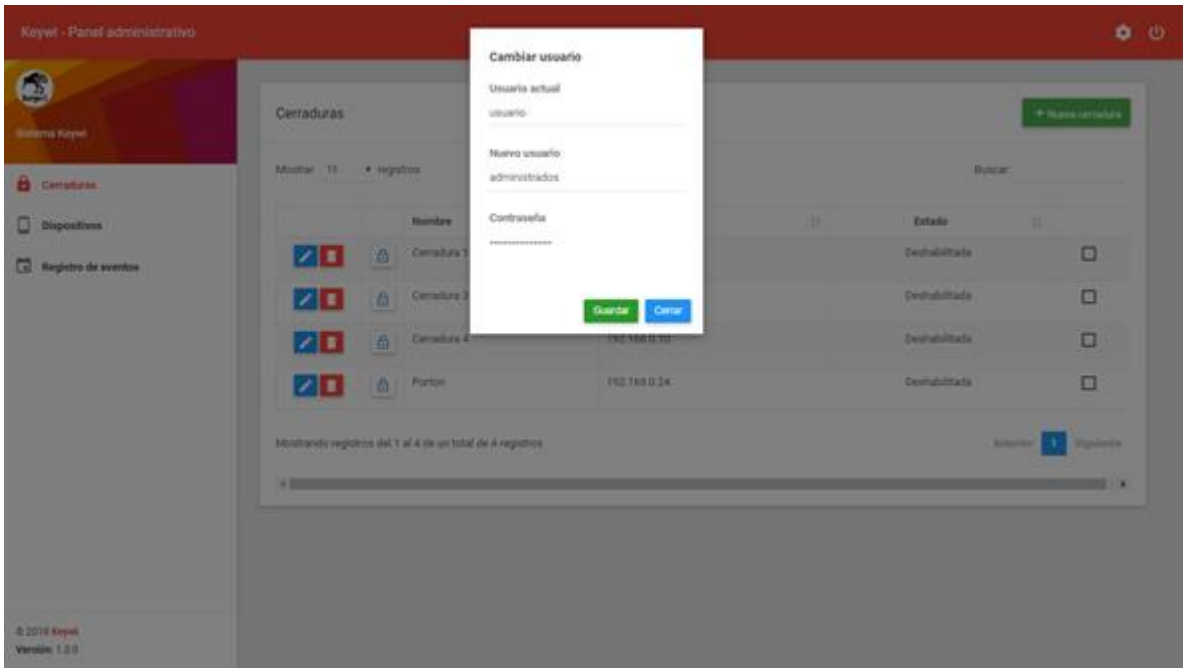


Figura 23) Interfaz gráfica panel administrativo – cambio de usuario

API REST - Gestión de cerraduras/dispositivos

El primer paso para implementar la gestión de las cerraduras y dispositivos contempló la construcción en el servidor de una API REST con la tecnología de Node.js, por las características y ventajas antes mencionadas, y de la cual pudiéramos consumir información desde una aplicación web, móvil u otro componente de software. Una API es la abreviatura de *Application Programming Interface*, o Interfaz de Programación de Aplicaciones. Podemos verla como contenedores de información que se envían entre sí distintas partes de una comunicación entre aplicaciones, y que consigue que los desarrolladores de componentes interactúen con los datos de la aplicación de un modo planificado y ordenado.

La arquitectura REST, abreviatura de *Representational State Transfer*, o Transferencia de Estado Representacional es una forma de diseñar una interfaz entre sistemas que utilice directamente el protocolo HTTP para compartir información entre un cliente (portátil, teléfono móvil, tablet, etc.) y un servidor, o la ejecución de operaciones sobre los datos.

Un escenario de petición a la API funciona de la siguiente manera:

1-Un cliente envía una petición HTTP a un servidor solicitando datos o ejecutar una tarea.

2-El servidor devuelve una respuesta HTTP con los datos o la respuesta de la ejecución.

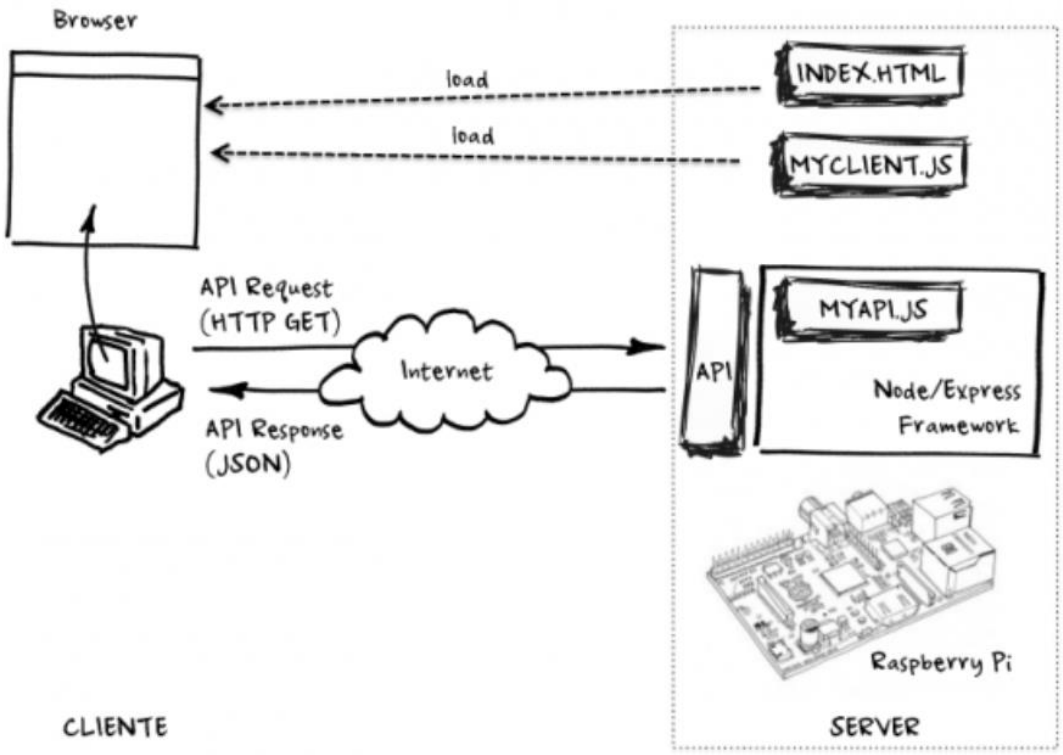


Figura 24) Interacción de componentes de la aplicación web

En la arquitectura planteada, el usuario accede a la información del sistema mediante la interfaz del panel administrativo y esta se comunica luego con la API REST desarrollada en

el servidor de cerraduras, la cual se encuentra en una aplicación distinta. Con este enfoque logramos separar la aplicación del servidor web correspondiente al panel administrativo de la API REST encargada de la gestión de datos y acciones sobre las cerraduras y dispositivos, aportando así un nivel mayor de seguridad. Conseguimos así que la aplicación del panel administrativo no tenga acceso a los datos almacenados en la base de datos si no es por medio de la API REST, la cual tiene métodos debidamente desarrollados y controlados para la obtención y manipulación de los datos del sistema de cerraduras mediante la autenticación del usuario administrador.

La autenticación en el servidor, almacenando la sesión era uno de los métodos más comunes hasta ahora. Pero para ello necesitábamos almacenar esa información en una base de datos.

Sin embargo, esto suponía una pérdida de escalabilidad en nuestra aplicación, ya que el servidor debe almacenar un registro por cada vez que el usuario se autentique en el sistema. Además, hacemos que el *backend* se encargue de ello y de esta manera si queremos desarrollar una aplicación móvil, necesitaríamos otro *backend* diferente, no pudiendo reutilizarlo.

Seguridad en la API

Una de las nuevas tendencias en cuanto al desarrollo web moderno, es la autenticación por medio de Tokens y que nuestro *backend* sea un API REST sin información de estado, *stateless*.

En este caso el usuario se autentica en nuestra aplicación con un par usuario/contraseña. A partir de entonces, cada petición HTTP que haga el usuario va acompañada de un *token* en la cabecera. JSON *Web Tokens* o simplemente JWT[3], fue la herramienta utilizada que nos permitió autenticarnos con el servidor mediante *tokens* de una forma simple y segura.

Los *tokens* son una cadena de alfanumérica, que es generada por el servidor y es enviada al cliente para autenticaciones futuras, evitando tener que enviar credenciales en cada invocación ya que este *token* no se almacena en el servidor, si no en el lado del cliente (por ejemplo, en *localStorage* o *sessionStorage*) y el API es el que se encarga de descifrar ese *token* y redirigir el flujo de la aplicación en un sentido u otro.

Como los tokens son almacenados en el lado del cliente, no hay información de estado y la aplicación se vuelve totalmente escalable. Podemos usar el mismo API para diferentes aplicaciones (*web* o *mobile*) solo debemos preocuparnos de enviar los datos en formato JSON y generar y descifrar tokens en la autenticación y posteriores peticiones HTTP a través de un *middleware* [15]

[3] https://jwt.io/

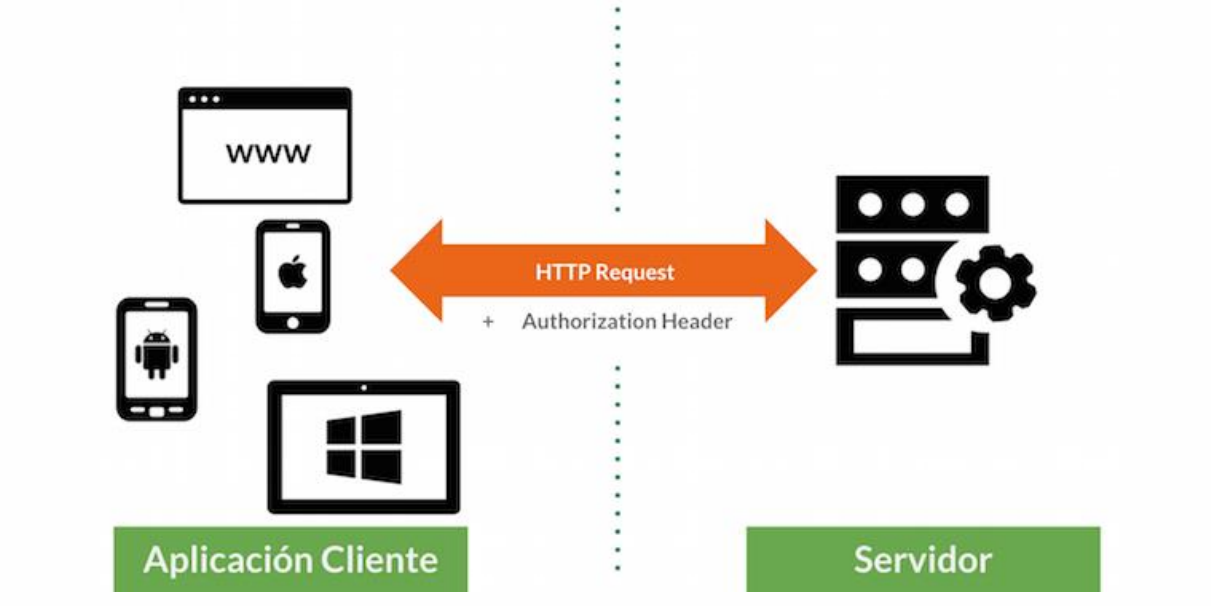


Figura 25) Petición HTTP cliente-servidor

También añade más seguridad al no utilizar cookies para almacenar la información del usuario, y así se logra evitar ataques CSRF (*Cross-Site Request Forgery*) que manipulen la sesión que se envía al *backend*. Además, se decidió hacer que el *token* expire después de un tiempo lo que le añade una capa extra de seguridad.

Enfoque de Diseño de la API

Además, para el desarrollo de este componente de software, debido a que alcanzaba una complejidad significativa, se decidió utilizar un patrón de diseño denominado DDD (por las siglas en inglés de *Domain Driven Design* o Diseño Orientado a Dominio) en el cual las clases del sistema se implementan en distintas capas de abstracción y se definen las interacciones entre las mismas a partir de contratos o interfaces [16]. Esto permite, en primer lugar, que la implementación de una capa sea agnóstica de cómo se implementa el resto, siempre y cuando respete la definición del contrato y da lugar a posibles mejoras a futuro sin necesidad de realizar mayores cambios a nivel infraestructura, simplemente cambiando la implementación de la interfaz que se utiliza. Encarar el diseño de esta manera permite:

- Enfocarse en el núcleo y la lógica del dominio.
- Basar diseños complejos en modelos de dominio.
- Poder colaborar más fácilmente con expertos del dominio, ya que el modelo de aplicación sólo resuelve problemas emergentes relacionados al dominio.

Para poder llevar a cabo este enfoque, se utilizó un patrón denominado Inyección de Dependencias (o *Dependency Injection* en inglés), que facilita la definición, mantenimiento y cambio de las clases que implementan las interfaces. Es una técnica donde se suministran los objetos a una clase en lugar de ser la propia clase la que cree dichos objetos. Esta

técnica está muy relacionada con un principio denominado Inversión del Control, en donde el programa no conoce cómo suceden las cosas, simplemente conoce qué sucede y cuándo, y alguien más, el inyector de dependencias en este caso, le entrega las instancias o referencias de los módulos con los que tiene que trabajar, reduciendo así el acoplamiento.

Servidor SocketIO

Internet se ha creado en gran parte a partir del llamado paradigma solicitud/respuesta de HTTP. Un cliente carga una página web, se cierra la conexión y no ocurre nada hasta que el usuario hace clic en un enlace o envía un formulario.

Hace ya algún tiempo que existen tecnologías que permiten al servidor enviar datos al cliente en el mismo momento que detecta que hay nuevos datos disponibles.

En la arquitectura planteada para el sistema de cerraduras resulta necesario que se puedan comunicar los datos disponibles desde el servidor de cerraduras hacia la aplicación web del panel administrativo, la cual los usuarios pueden utilizar accediendo a través de la intranet mediante un navegador.

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y *full-duplex* sobre un único socket TCP. Está diseñada para ser implementada en servidores web y navegadores. En estos, el cliente abre una conexión HTTP con el servidor, esta conexión se produce en tiempo real y se mantiene permanentemente abierta hasta que se cierre de manera explícita. Esto significa que cuando el servidor quiere enviar datos al servidor, el mensaje se traslada inmediatamente.

Como alternativas a esta solución encontramos las llamadas AJAX para conseguir un resultado parecido. AJAX es un acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML) y es una técnica de desarrollo web para crear aplicaciones interactivas [17]. Las llamadas AJAX consisten en el cliente que envía una petición al servidor, ahora en lugar de que el servidor responda con datos que puede que no tenga, esencialmente mantiene la conexión abierta hasta que los datos actualizados estén listos para ser enviados. El cliente entonces recibe esto y envía una solicitud.

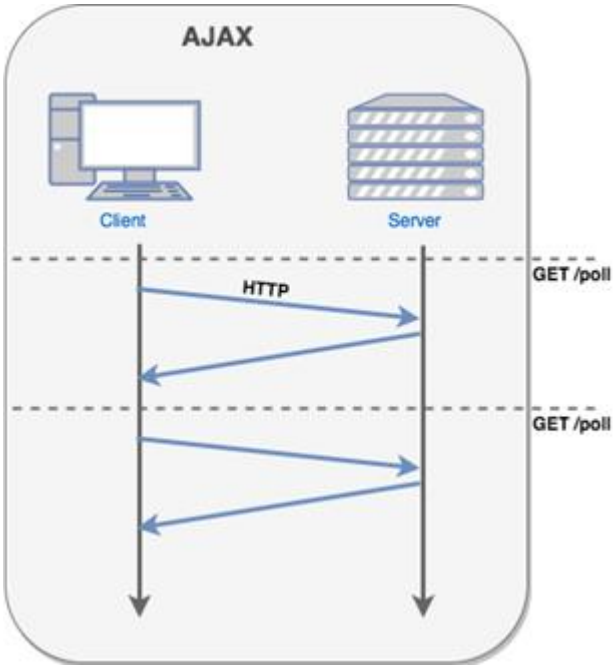


Figura 26) Petición cliente-servidor mediante AJAX

Sin embargo, esta no es realmente una pieza de elegante tecnología, además es también posible que para una petición el que se le acabe el tiempo, por lo tanto, una nueva conexión será requerida de cualquier forma.

La problemática de las llamadas AJAX al servidor radica en la sobrecarga de manera innecesaria del servidor para simular el tiempo real, debido a las innumerables peticiones se deben realizar para conseguir los resultados en un aparente tiempo real.

En cambio, la tecnología de *WebSockets*, al disponer de un socket abierto, el servidor puede enviar datos a todos los clientes conectados a ese socket, sin tener que estar constantemente procesando peticiones de AJAX. La ventaja en cuanto a rendimiento y escalabilidad es bastante evidente al utilizar *WebSockets*.

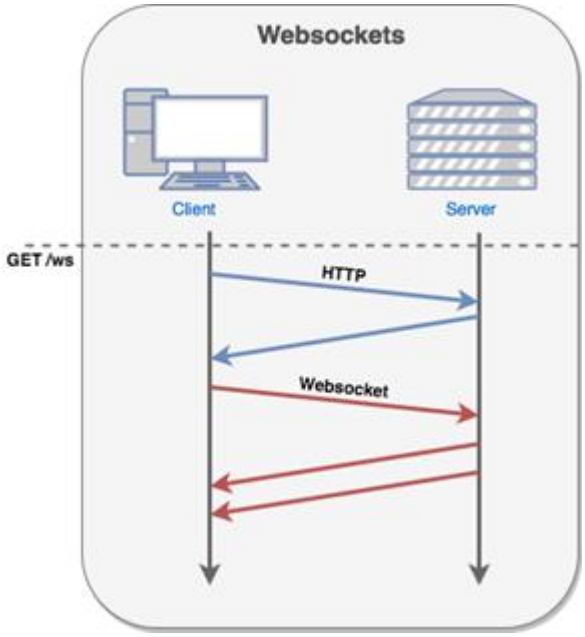


Figura 27) *Comunicación cliente-servidor mediante Websocket*

La solución planteada utiliza una biblioteca de JavaScript para aplicaciones web en tiempo real llamada Socket.IO.

Esta es una librería que permite manejar eventos en tiempo real mediante una conexión TCP y todo ello en *JavaScript*. Es realmente potente y podemos hacer todo tipo de aplicaciones en tiempo real ya que permite la comunicación bidireccional en tiempo real entre clientes web y servidores.

Socket.IO tiene dos partes fundamentales: una biblioteca del lado del cliente que se ejecuta en el navegador y una biblioteca del lado del servidor para Node.js.

Su utilización resultó un gran sustituto de AJAX como tecnología para obtener datos del servidor, ya que no tenemos que pedirlos, el servidor nos los enviará cuando haya nuevos mediante la emisión de mensajes que se propagan hasta el panel administrativo. De esta forma logramos funcionalidades como la sincronización de dispositivos y cerraduras, o bien la monitorización de las cerraduras en tiempo real desde el panel administrativo web.

Servidor de cerraduras (LAN)

Este servidor existe con la finalidad de generar una capa de abstracción que separe el control de las comunicaciones entre las cerraduras y el servidor de administración.

La existencia de éste permite la aislación del control del usuario y el manejo de dispositivos clientes, controlando que las comunicaciones se den de forma segura y únicamente entre clientes (cerraduras) y servidor.

- Servicios externos: El servidor de cerraduras se encuentra en ejecución en un hilo independiente directamente ejecutado sobre el sistema operativo. Éste es un servicio de red que trabaja con los protocolos TCP y UDP como transporte, atendiendo sobre los puertos TCP y UDP, utilizando el software Python como aplicación. Para poder comunicarse con el servidor, se encuentran definidos una serie de mensajes que corresponden a las funcionalidades que éste ofrece.
- Servicios internos: Éste servidor se divide en dos capas de abstracción, en primera instancia, el programa que corre como hilo principal, el cual es el encargado de prestar los **servicios externos** mencionados previamente, a su vez, éste es quién se encarga de analizar cada mensaje entrante desde la capa de transporte y ejecuta la acción correspondiente; esto nos lleva a la **segunda capa**, en la cual se encuentran los distintos programas o scripts que realizan las acciones particulares, éstos son considerados **servicios internos** ya que sólo pueden ser ejecutados desde el hilo principal. Éstos trabajan de forma autónoma en hilos paralelos, siendo interrumpida su ejecución únicamente en casos de necesidad como puede ser la realización de otra acción o solicitud de respuesta de un mensaje. Estos servicios internos están programados con lenguajes Python y Bash.

Servicios ofrecidos	Tipo	Protocolo
Actualización de datos de cerraduras El servidor de cerraduras será notificado cada vez que el usuario modifique algún permiso de acceso para enviarle a todas las cerraduras correspondientes la actualización sobre la información de acceso.	Externo	UDP
Sincronización de dispositivo Actúa como intermediario entre el panel de administración y la cerradura, interactuando a través de HTTP y TCP respectivamente.	Externo	TCP, HTTP
Notificación de cerraduras no registradas Escucha a los mensajes enviados por las cerraduras no registradas (UDP) y notifica al panel de administración para que el usuario las agregue (HTTP).	Interno, Externo	UDP, HTTP
Sincronización de cerraduras Se utiliza como interfaz entre la aplicación web y la cerradura para el proceso de incorporación de esta última al sistema. Comunicándose por HTTP y TCP respectivamente.	Externo	TCP, HTTP
Escucha de estado de cerraduras El servidor está constantemente escuchando en un puerto UDP para recibir el reporte de estado de cada una de las cerraduras, el cual es actualizado en la base de datos cuando se recibe	Externo	UDP

Tabla 2) Relación de servicios – tipo de conexión

Base de datos

La base de datos se compone por un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

En este caso se utilizó una base de datos relacional, mediante una colección de elementos organizados en un conjunto de tablas formalmente.

La interfaz entre la aplicación y la base de datos relacional es el lenguaje de consultas estructuradas (SQL). Los comandos de SQL son utilizados tanto para consultas interactivas para obtener información de la base de datos relacional y para la recopilación de datos para los informes de eventos.

Además de ser relativamente fáciles de crear y acceder, la utilización de una base de datos relacional nos presentó la importante ventaja de ser fácil de extender. Después de la creación original de una base de datos, una nueva categoría de datos se puede añadir sin necesidad de que todas las aplicaciones existentes sean modificadas.

Para la implementación de la base de datos se utilizó MariaDB. Este es un sistema de base de datos que proviene de MySQL, pero con licencia GPL (siglas en inglés de *General Public License*), por el fundador de MySQL y la comunidad de desarrolladores de software libre.

A continuación, se presenta al lector el diagrama de entidad relación que permite representar las entidades relevantes utilizadas en el sistema, así como sus interrelaciones y propiedades.

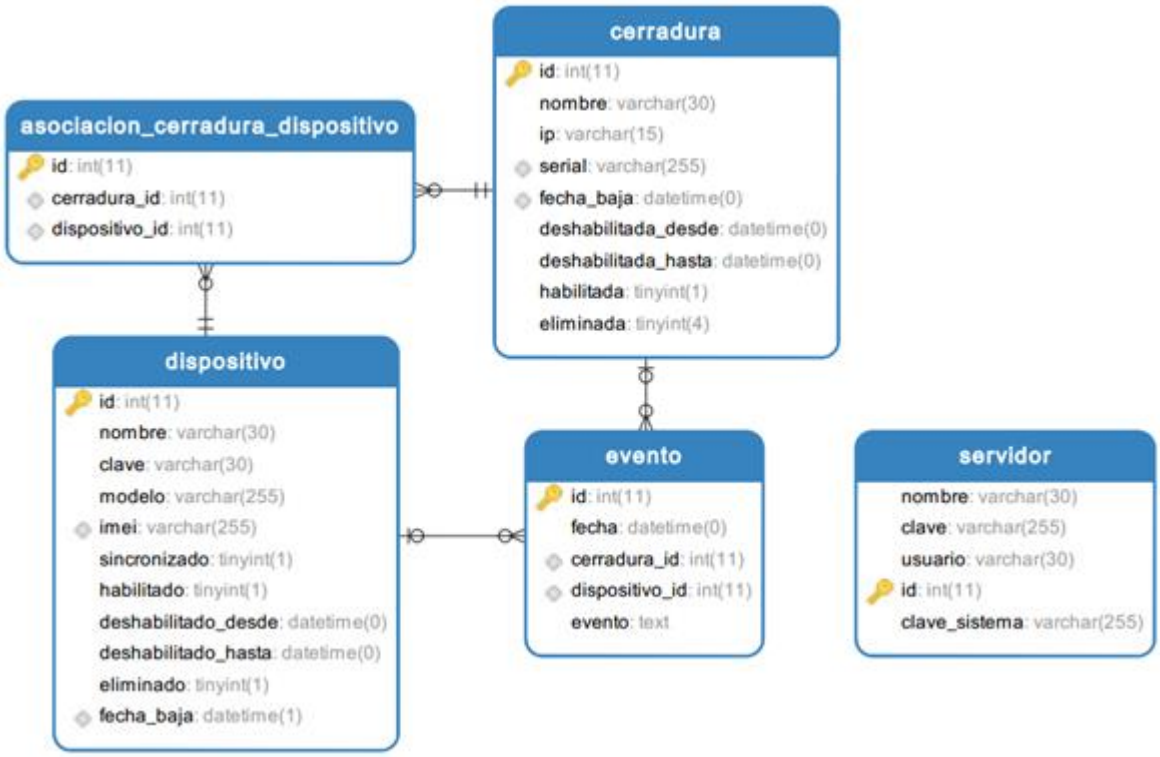


Figura 28) Diagrama Entidad – Relación del sistema

ii. Aplicación móvil

Conformando uno de los ejes centrales del proyecto, se desarrolló una aplicación móvil que tiene la capacidad de interactuar con la cerradura por medio de NFC. Además, esta pieza tiene una gran participación en la configuración inicial del sistema, no sólo para poder identificar y sincronizar los dispositivos en el sistema, sino que también es responsable de establecer configuraciones de red para la cerradura.

La aplicación se desarrolló para su funcionamiento en dispositivos con sistema operativo Android. Para la implementación se utilizó el lenguaje Kotlin, desarrollado por JetBrains, que corre sobre la máquina virtual de Java. Este lenguaje, si bien no tiene una sintaxis compatible con Java, está diseñado para interoperar con código Java y su biblioteca de clases. Esto fue verificado en el proyecto, ya que se realizaron algunos fragmentos en código Java que se acoplaron perfectamente al resto de la implementación.

La tarea principal de la aplicación es la comunicación NFC con la cerradura. Para esto, se realizó un servicio que se ejecuta en segundo plano, de manera que no haga falta tener la aplicación abierta para su funcionamiento, que detecte cuando el dispositivo se acerca al campo NFC de la cerradura.

El servicio funciona emulando a una tarjeta NFC, en lo que se denomina *Host-based card emulation* (HCE) [18], para poder establecer la comunicación con el lector NFC ubicado, en nuestro caso, en la cerradura. Sin embargo, en vez de asignar un valor estático para el dispositivo, el manejo de la recepción y envío de mensajes NFC es cedido a la aplicación, como se puede apreciar en la figura.

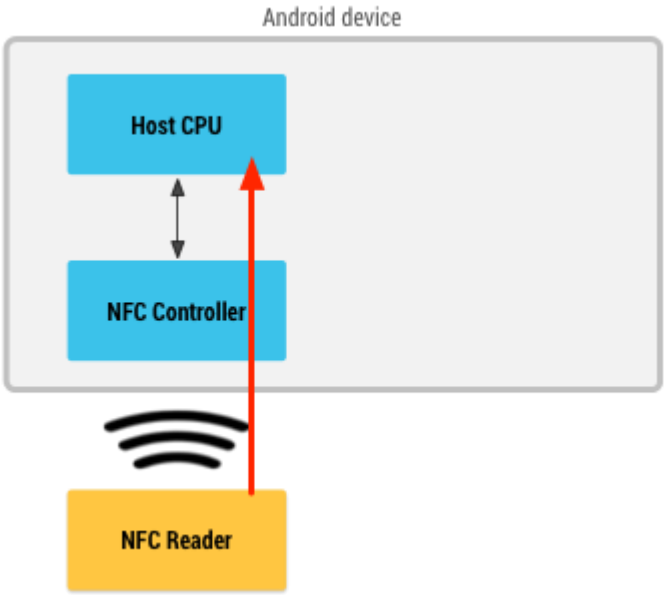


Figura 29) Controlador NFC en dispositivos Android

La transmisión NFC se realiza en conformidad con la especificación ISO/IEC 14443-4, y está implementada en el servicio HCE de Android. Para la utilización de este, y como se

especifica en ISO/IEC 7816-4, se realiza la recepción y envío de *Application Protocol Data Units* (APDUs).

Como HCE es un servicio de Android que está a disposición de toda aplicación que lo necesite, es indispensable la posibilidad de identificar unívocamente que el mensaje que se está recibiendo corresponde a una cerradura y debe ser respondido solamente por nuestra aplicación. De esta manera, la especificación ISO/IEC 7816-4 define un *Application ID* (AID) que consiste de hasta 16 bytes, con una restricción en los bits 8 a 5 (es decir, los más significativos) del primer byte según la siguiente tabla [19]:

Table 90 — Categories of application identifiers

Value	Category	Meaning
'0' to '9'	-	Reserved for backward compatibility with ISO/IEC 7812-1[3] (see annex D)
'A'	International	International registration of application providers according to ISO/IEC 7816-5[4]
'B', 'C'	-	Reserved for future use by ISO/IEC JTC 1/SC 17
'D'	National	National (ISO 3166-1[1]) registration of application providers according to ISO/IEC 7816-5[4]
'E'	Standard	Identification of a standard by an object identifier according to ISO/IEC 8825-1
'F'	Proprietary	No registration of application providers

Figura 30) Significado del valor del primer byte

De manera que el AID elegido deberá comenzar con F en el primer byte. Esta es una configuración tanto de la aplicación como del módulo NFC de la cerradura.

El AID es transmitido por la cerradura en un primer mensaje denominado SELECT, con su correspondiente cadena de bytes para identificarlo. Una vez que el dispositivo Android lee este mensaje e identifica la aplicación registrada para responder el AID correspondiente, la comunicación queda establecida entre el lector NFC y la aplicación seleccionada, hasta que el sistema detecte un nuevo mensaje SELECT. De esta manera, se puede hacer un envío y recepción de mensajes continuo, entre aplicación y lector, siempre y cuando el dispositivo se mantenga en el campo NFC del lector. Este procedimiento se realiza cada vez que se establece la comunicación y es comúnmente denominado el "*handshake*" de la comunicación.

La comunicación NFC con la cerradura se utiliza tanto para el común funcionamiento de esta, es decir, el intercambio de claves para la autorización del dispositivo, como para el alta de un nuevo dispositivo. Por lo tanto, y como la lógica y complejidad de cada una de estas operaciones es diferente, se decidió separar el procesamiento de estas en dos servicios distintos, con sus correspondientes AIDs para cada uno de ellos. Es importante aclarar que estos servicios se ejecutarán en simultáneo dentro de la aplicación, es decir, el dispositivo estará a la espera de cualquiera de los dos servicios y será capaz de responder a los mismos sin la intervención del usuario; sin embargo, del lado de la cerradura, la ejecución del servicio de autorización y de sincronización será mutuamente excluyente, es decir, sólo podrá realizarse uno de los procedimientos a la vez. Esto es así porque actor que inicia la comunicación es la cerradura y deberá iniciar la comunicación (a través del mensaje

SELECT) sí o sí con uno de los AID. Esto se verá con más detalle en la sección de la cerradura pertinente a las comunicaciones NFC.

Servicio de Sincronización

Si bien la sincronización del dispositivo inicia en el panel administrativo, consultando la opción de Agregar Dispositivo, también se deberá ingresar previamente a la opción correspondiente en la aplicación móvil para poder otorgar permisos a la misma para datos de reconocimiento del dispositivo (los cuales son sólo para la identificación interna de los dispositivos y no serán almacenados en el sistema). Esto se realiza mediante el botón denominado Sincronizar Dispositivo.

Una vez otorgado el permiso y comenzado el proceso de sincronización a partir del panel administrativo, simplemente se acerca el móvil a la cerradura escogida. La misma iniciará una comunicación NFC pidiéndole al dispositivo sus datos identificatorios. Una vez que el sistema dé de alta y otorgue un ID al dispositivo, la cerradura enviará, por el mismo medio, la ID tanto del dispositivo como del sistema de cerraduras. Posteriormente, la aplicación almacenará localmente estos datos (en una base de datos interna SQLite), y mostrará una notificación para avisarle al usuario que la acción se ha realizado con éxito. Si la conexión NFC fuera a romperse antes de que la aplicación pueda almacenar sus datos, el procedimiento fallará y se notificará de lo sucedido al usuario a través de una notificación.

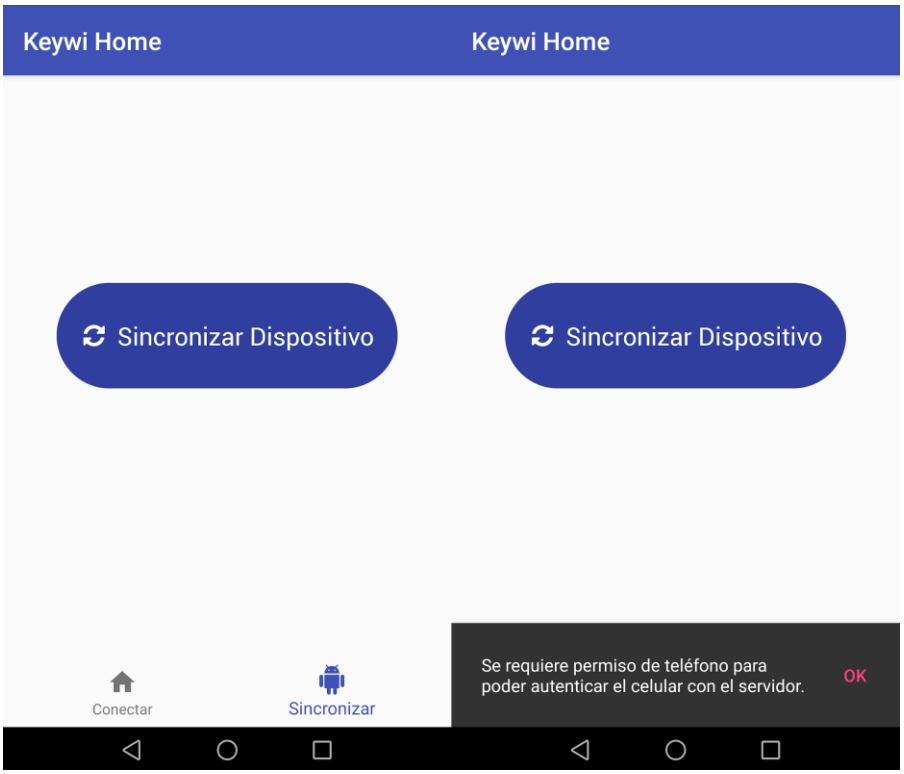


Figura 31) Interfaz gráfica aplicación móvil – Sincronización de dispositivo

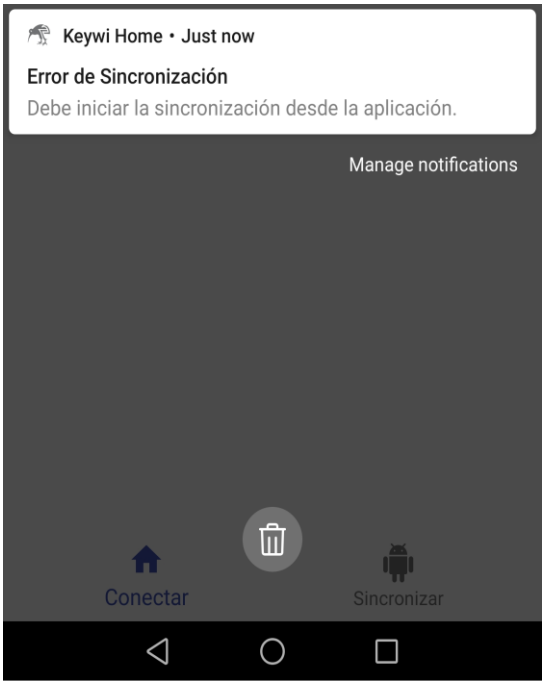


Figura 32) Interfaz gráfica aplicación móvil – mensaje de error

Servicio de Autorización

Cuando una cerradura ya está disponible para su normal funcionamiento, es decir, ya ha sido configurada en la red, fue dada de alta en el sistema y recibió datos sobre dispositivos permitidos, los dispositivos podrán acercarse e iniciar el procedimiento de autorización (siempre y cuando la cerradura no esté siendo utilizada en el momento para la sincronización de un nuevo dispositivo). Así, cuando un dispositivo se acerque, la cerradura enviará, una vez establecida la comunicación, la ID del sistema para que la aplicación pueda identificar su clave única para este sistema (almacenada previamente en el proceso de sincronización). Una vez transmitida su clave, la cerradura enviará un mensaje de éxito y habilitará el ingreso. En caso de que la clave no sea aceptada por la cerradura, la misma enviará un mensaje de error por el mismo medio para que la aplicación pueda notificar al usuario. Lo mismo ocurre si no se encuentran claves en el dispositivo para la clave de sistema recibida por la cerradura o si se rompiese la conexión en cualquier momento del intercambio.

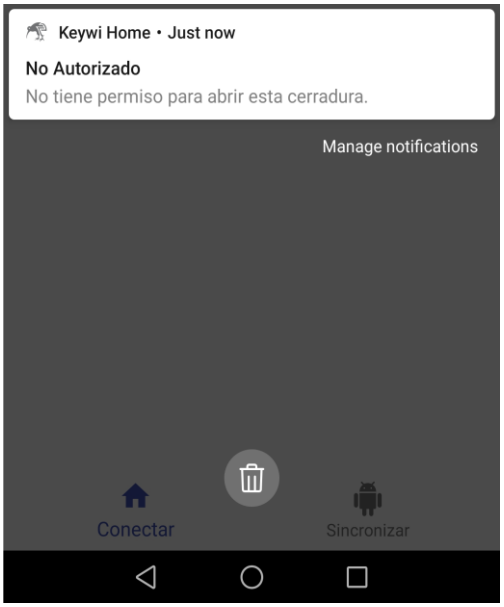


Figura 33) Interfaz gráfica aplicación móvil – Mensaje de dispositivo no autorizado

Por otro lado, y como se mencionó anteriormente, esta aplicación es responsable de la configuración inicial de las cerraduras. Esto es, la configuración de las credenciales de la red a la cual se deberá conectar la cerradura. Lo que se intentó imitar es el comportamiento de la configuración de dispositivos ya existentes en el mercado, como por ejemplo el de Google Chromecast, en el que, por medio de una aplicación móvil, se realizan las configuraciones de red del mismo. Así, el usuario deberá conectar su dispositivo a la red WiFi transmitida por la cerradura y presionar el botón para escanear las redes cercanas. De esta manera, se reciben las redes que son visibles en este momento por la cerradura y aparecerán en una lista para su selección. Una vez seleccionada la red e introducida la contraseña, se enviará la configuración a la cerradura la cual notificará por sí misma si el proceso se ha realizado con éxito (debido a que la aplicación perderá comunicación con la misma). A diferencia de la solución de Google, el usuario deberá introducir obligadamente la contraseña de la red, por más que ya esté presente esta configuración en el dispositivo, porque la adquisición de este tipo de credenciales por parte de aplicaciones Android de terceros no está permitida, por cuestiones de seguridad.

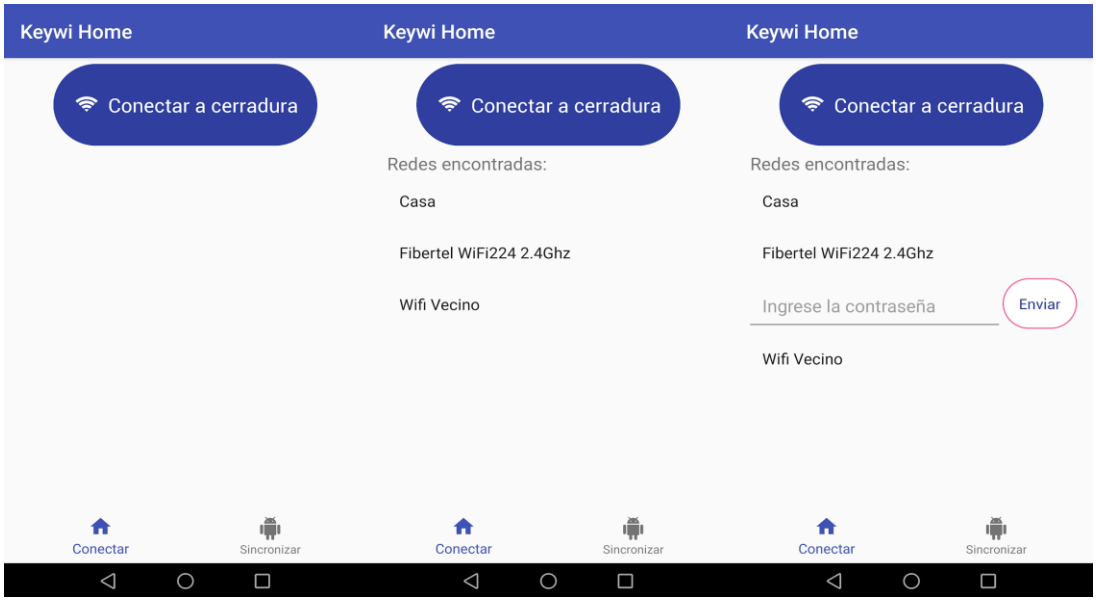


Figura 34) Interfaz gráfica aplicación móvil – conexión de cerradura a red WLAN

iii. Cerradura

Hardware

En lo que respecta a la cerradura, la misma se encuentra funcionando físicamente sobre una Raspberry Pi Zero, que es una computadora de placa reducida, de tamaño pequeño, pero con todas las funcionalidades necesarias para la implementación del proyecto, incluyendo conectividad WiFi incorporada. Para la implementación de este dispositivo se utiliza la misma solución de sistema operativo utilizada en el servidor (Raspbian).

Pensada también para realización de proyectos de Internet de las Cosas, la plaqueta tiene disponibles una serie de pines que pueden ser luego programados dentro de la misma, estos se denominan GPIO o entradas/salidas de propósito general (del inglés *general-purpose input/output*). Estos puertos son utilizados para comandar elementos electrónicos adicionales, que complementan físicamente el funcionamiento de la cerradura.

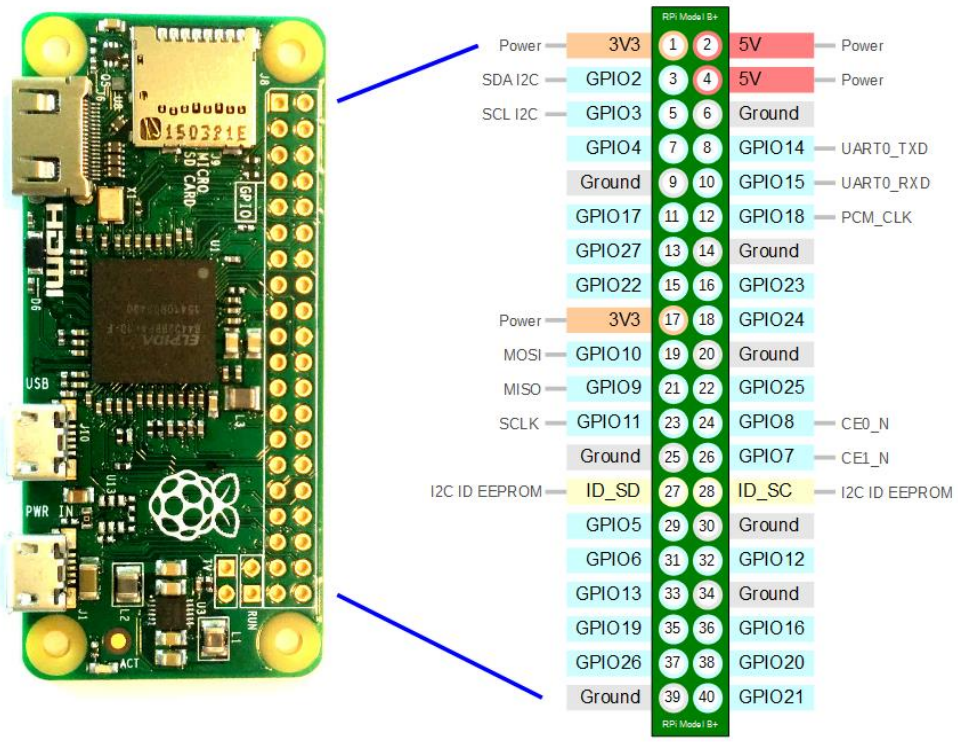


Figura 35) Pinout GPIO Rasbperry Pi Zero [20]

Motor paso a paso

Comandado lógicamente con las salidas GPIO de la Raspberry Pi, éste se utilizó para el movimiento del pestillo. Es accionado mediante las acciones “Abrir cerradura” y “Cerrar cerradura”.

LED RGB

Este se utilizó para indicar los diferentes estados de la cerradura. El uso del mismo será indicado en cada caso a medida que corresponda en el documento.

Sensor de proximidad:

Detecta el estado de la puerta, si la misma se encuentra cerrada o abierta, para determinar si es correcto o no realizar acciones como “Abrir cerradura” o “Cerrar cerradura” y así advertir al usuario si existe un inconveniente.

Zumbador

Mejor conocido por su denominación en inglés, *buzzer*, es un transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono y se utilizó para advertir al usuario de forma auditiva si la puerta se encuentra en estado abierta por un tiempo prolongado.

Pulsador

Es el botón que se utilizó para comandar físicamente la cerradura, empleado en funcionalidades como el reinicio de fábrica o la conexión a la red WLAN.

Tarjeta PN532 (Interfaz NFC):

El módulo *ITEAD ® PN532 NFC* es utilizado como interfaz NFC entre la cerradura y el dispositivo móvil, permite a la tarjeta trabajar sobre las interfaces I2C, UART y SPI. Como ésta conforma un elemento fundamental para el desarrollo de la cerradura, se explicará con más detalle a en su propio apartado a continuación.

El módulo NFC

Figura 36) Módulo de interfaz NFC para RaspberryPi

Por otro lado, si bien la plaqueta de la Raspberry Pi ya viene con muchos componentes incorporados, probados y funcionando, uno de los mayores motivos por la que la elegimos, incluyendo antenas de WiFi y bluetooth, no cuenta con un módulo de NFC. Por esta razón, tuvimos que investigar en el mercado para obtener una solución de terceros, genérica, que sea compatible con la Raspberry Pi. Como primer paso, consultamos en la web de uno de los mayores proveedores de componentes electrónicos de EE. UU. denominado Adafruit. Sin embargo, por cuestiones de precios de envío y disponibilidad, obtuvimos un módulo similar de otro proveedor, pero que era completamente compatible con la documentación brindada por Adafruit para el componente.

El componente puede ser utilizado a través de 3 interfaces [21]:

- **UART (*Universal Asynchronous Receiver/Transmitter*)**: Comunicación asincrónica simple entre dos dispositivos, no necesita direccionamiento, cada dispositivo utiliza su *clock* de manera independiente. Utiliza los pines 8 para transmitir y 10 para recibir, conectados a los pines de recepción y transmisión del módulo NFC.
- **SPI (*Serial Peripheral Interface*)**: Comunicación sincrónica simple pero cuyo protocolo de transferencia es propio de cada marca y fabricante, de manera que se debe seguir una guía específica para su utilización. Permite comunicación *full duplex* (es decir, en ambos sentidos al mismo tiempo) de alto ancho de banda. Utiliza 4 pines para la comunicación: el pin 23 para el *clock*, el 19 para transmisión, el 21 para recepción y el 24 para control de flujo.
- **I2C (*Inter-Integrated Circuit*)**: Comunicación sincrónica compleja de ancho de banda fijo, que permite la comunicación *half-duplex* entre múltiples maestros y múltiples esclavos. Necesita menos cables, utilizando sólo el pin 3 para transmisión de datos y el 5 para la señal de *clock*.

La interfaz elegida fue la I2C, debido a la menor cantidad de cables y a que, aunque su método de comunicación es más complejo, se contó con una solución ya implementada en la librería que se incorporó para manejo de NFC.

Finalmente, conectando los pines relacionados a alimentación de 5V, se integró sin problemas la plaqueta a la cerradura.

A continuación, un diagrama de la ubicación de los pines mencionados anteriormente en la Raspberry Pi, correspondiente a la cerradura y la conexión final con el módulo NFC [22].

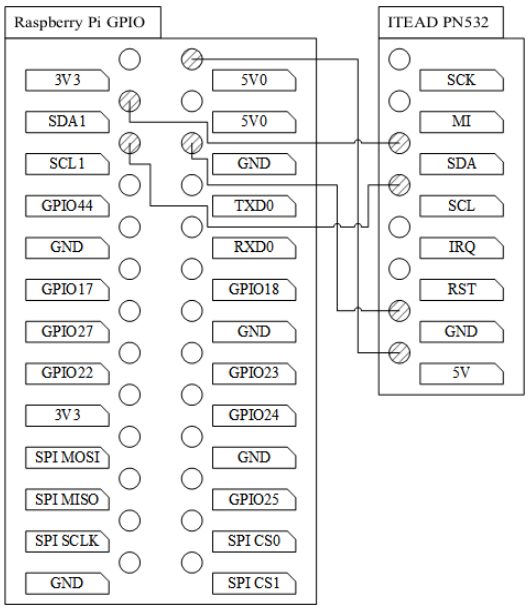


Figura 37) Conexión I2C mediante GPIO Raspberry Pi Zero con ITEAD PN532

Se debió configurar el módulo para su funcionamiento en modo I2C, así como también instalar un paquete denominado *i2c-tools* dentro del sistema operativo de la Raspberry Pi.

Por último, y como se mencionó anteriormente, para poder tener acceso programático al módulo NFC, es decir, desde nuestro programa de ejecución, se utilizó la librería denominada LibNFC, desarrollada bajo la licencia de GNU de software libre [23]. Para la utilización de esta, se necesita la modificación del archivo de configuración donde se debe incluir el nombre del módulo y la forma de conexión, en nuestro caso *PN532_I2c.*

El funcionamiento del intercambio de información a través de NFC fue explicado en la sección de la aplicación móvil. De este lado, además de incorporar el envío y recepción de mensajes al programa principal de la cerradura, se debió también definir y transmitir los APDUs (explicado en la sección 3 de este apartado) que la aplicación móvil deberá reconocer para su funcionamiento.

Conexión de los componentes

A continuación, podemos ver un diagrama de conexión de los dispositivos a la placa principal:

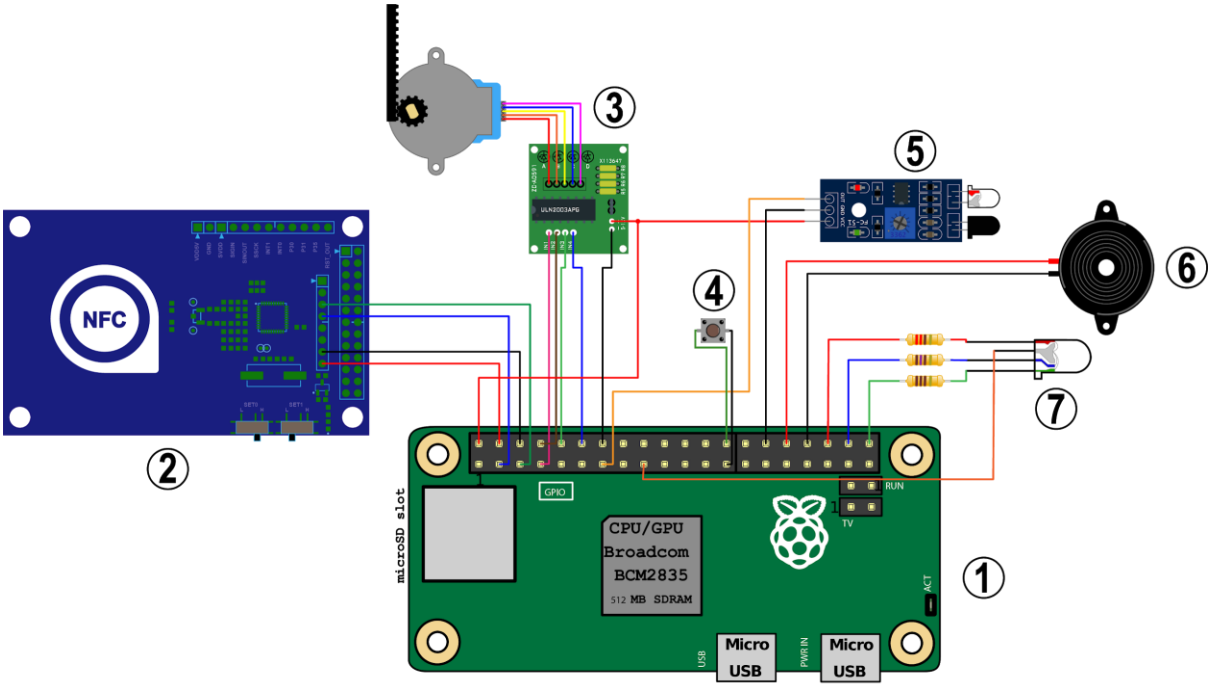


Figura 38) Diagrama de conexión electrónica componentes – Raspberry Pi

Referencias:

1) Raspberry Pi Zero (placa principal)

2) Tarjeta NFC, conectada en la interfaz i2C de la Raspberry (pines 3 y 5 para sda y sdb respectivamente) y alimentada con 5V (pin 4 para 5v y 6 para gnd).

3) Motor paso a paso, utiliza alimentación de 5V de la placa (pines 2 y 14 para tensión y gnd) y 4 pines GPIO en modo salida para control de rotación del motor (pines 7, 8, 10 y 12).

4) Pulsador, utiliza un pin GPIO como salida (pin 26) y un pin gnd para cerrar el circuito (ping 25).

5) Sensor de proximidad, alimentado con 5V, tomado del mismo pin de alimentación que el motor paso a paso y cerrando el circuito con el pin 30 de gnd, se utiliza el pin 13 en modo de entrada para recibir la lectura del sensor.

6) Buzzer, conectado en el pin 33 del GPIO en modo salida, cerrando el circuito con el pin 34 (gnd).

7) Led RGB, utiliza alimentación de 3,3V (pin 17), conectado en el ánodo del LED, y los pines 36, 38 y 40 del GPIO en modo entrada para los cátodos, cerrando los diodos rojo, verde y azul respectivamente.

Alimentación eléctrica

En lo que respecta al suministro eléctrico del dispositivo, se utiliza una batería de litio de 10.000mAh la cual mantiene energizada la placa principal con sus componentes, ésta se carga mediante un cargador inalámbrico colocado en el marco de la puerta, permitiendo mantener en carga constante el dispositivo.

En la siguiente imagen podemos visualizar un diagrama electrónico de la conexión.

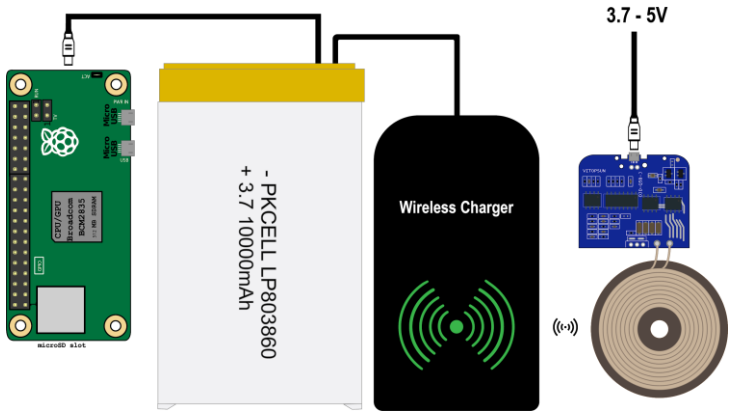


Figura 39) Diagrama de alimentación eléctrica de la placa principal

Implementación

En la siguiente imagen se puede observar la implementación de la conexión, para esto se utilizó una placa de pruebas, más conocido por su nombre en inglés *protoboard*. Este dispositivo es un elemento electrónico que consiste en un tablero con perforaciones interconectadas con un patrón donde se colocan los dispositivos electrónicos para armar circuitos.

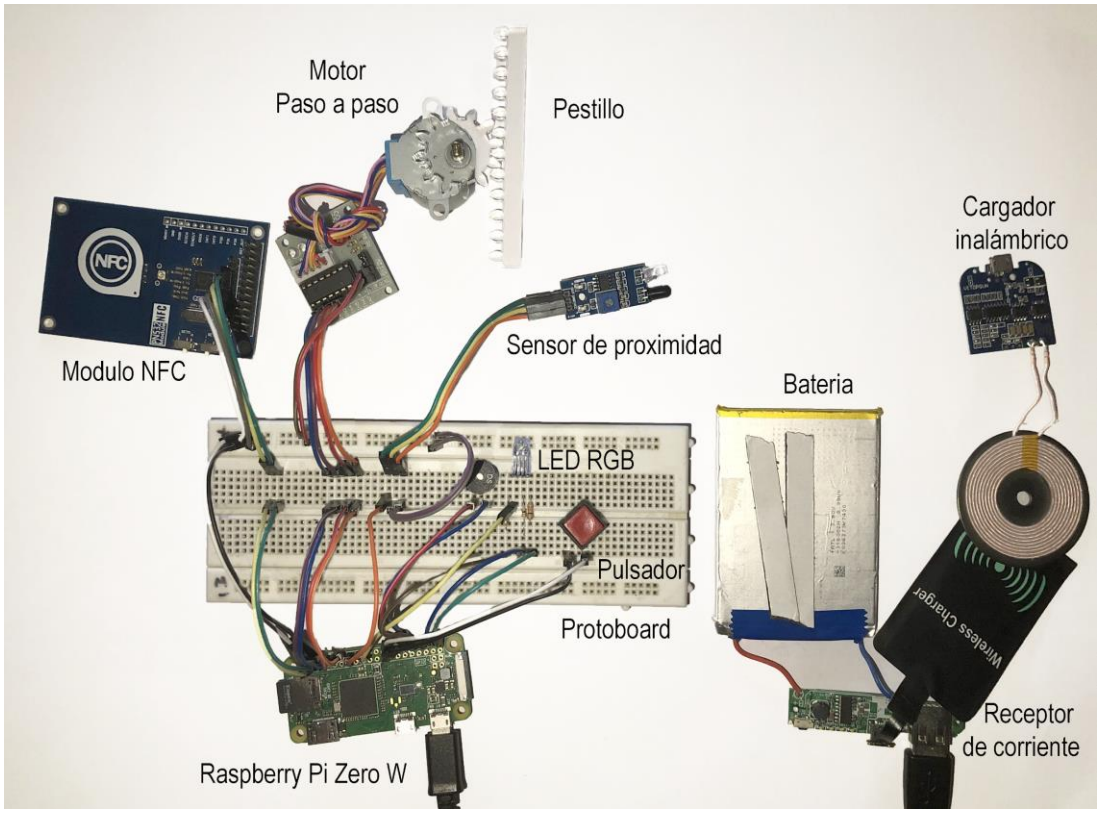


Figura 40) Foto de implementación del circuito completo de una cerradura

Software

Se dividió el funcionamiento de la cerradura en diferentes servicios que ofrece, dependiendo del estado de la configuración inicial del sistema o su normal funcionamiento.

Servicios de la Cerradura

Servicio de Conexión WiFi

Para configurar la cerradura por primera vez en el sistema, se diseñó un procedimiento de configuración en conjunto con la aplicación móvil.
En primer lugar, el usuario deberá presionar un botón físico de configuración en la cerradura ubicado en su parte inferior, el cual iniciará este servicio indicando en un LED (durante todo este proceso, el LED estará con una luz verde intermitente) que se inició la secuencia y activando a la cerradura en modo punto de acceso, de ésta forma, se podrá acceder a una red WiFi propia, a la cual se podrá conectar el dispositivo móvil con el que se llevará a cabo la sincronización. El SSID de la red es el mismo para todas las cerraduras y la contraseña se encuentra indicada en un manual de usuario ya que es única por dispositivo debido a motivos de seguridad.
Luego, el usuario deberá conectar el dispositivo e ingresar en el menú de configuración de WiFi dentro de la aplicación móvil, lo que hará que cerradura le transmita las redes que tiene disponibles para conectarse. Por último, el usuario deberá seleccionar una red de las listadas y colocar las credenciales correspondientes. Si éstas son las correctas, la cerradura dejará de emitir su SSID y detendrá el servicio de punto de acceso, para conectarse a la especificada por el usuario, indicando al usuario el éxito mediante el LED indicador con un color verde constante, de lo contrario, el LED se pondrá en color rojo intermitente durante 5 segundos y deberá iniciar nuevamente el procedimiento.

Servicio de Reporte de Cerradura No Registrada

Cuando el dispositivo detecta que está conectado a una WLAN, pero no está sincronizado con ningún servidor (es decir, no esté incorporada en un sistema KeyWi), se encuentra en un estado de “reporte”, es decir, utiliza el servicio de reporte de cerradura para indicar a los dispositivos que es una cerradura (mensaje que es recibido por el servidor al momento de incorporar una cerradura al sistema).
Este reporte consiste en un envío de un mensaje por protocolo UDP por difusión amplia (o *broadcast*) a un puerto conocido por el sistema en el cual se reporta en el cuerpo del paquete el nombre de la cerradura (código de serie). Este mensaje es enviado 3 veces por segundo, con el fin de que el servidor no deba estar recibiendo paquetes más de 1 segundo para reconocer todas las cerraduras de la red.

Una vez que la cerradura sea incorporada a un sistema, ésta detecta la situación y detiene el servicio.

Servicio de Sincronización de un Dispositivo

La cerradura también tiene un rol protagónico en la configuración inicial de un dispositivo en el sistema. Será el intermediario entre el panel de configuración y el nuevo dispositivo, por lo que la tarea de este servicio y la complejidad de la misma es mantener dos conexiones en simultáneo, una con el panel y otra con el dispositivo, y codificar los mensajes recibidos de un lado hacia el otro para la correcta interpretación de cada uno de estos.

Servicio de Apertura de Cerradura

La cerradura detectará cuando un dispositivo móvil se acerque al módulo NFC y comenzará el procedimiento de apertura de la cerradura. Para el cual la cerradura primero transmite al dispositivo el id del sistema, para que el dispositivo consulte su base de datos interna y obtenga la id que lo identifica en el sistema. Cuando la cerradura recibe esta id del dispositivo, consulta una pequeña base de datos que contiene los dispositivos autorizados para su apertura. Si el dispositivo se encuentra dentro de la lista de autorizados, se activará el mecanismo de la cerradura, de lo contrario, se le notificará al dispositivo. Además, éste servicio puede ser accionado mediante el panel web de administración.

Servicio de Actualización de Registros

Periódicamente, la cerradura intercambiará información con el panel de control para actualizar los dispositivos autorizados y para enviar sus registros de acceso. Este proceso consiste en la recepción de un paquete de forma periódica por parte del servidor, conteniendo en su cuerpo una lista de los dispositivos habilitados a la apertura de la cerradura. Luego, con esta información, el servicio se encarga realizar una actualización en la base de datos local.

Servicio de Reporte de Estado

Este servicio es utilizado en el sistema para que el servidor conozca el estado constante de la cerradura en la red. El servicio consiste en el análisis del estado de los dispositivos de red de la cerradura, además analiza el correcto funcionamiento de la interfaz NFC, esta información es almacenada en variables y enviada al servidor. El proceso se realiza en un bucle infinito con un intervalo de 3 segundos.

Servicio de Bloqueo de Cerradura

Si la cerradura recibe un mensaje del panel de control para que se bloquee, es decir, que no permita su accionamiento, este servicio se encargará de pausar el servicio correspondiente a la apertura de esta y de reanudarlo cuando se reciba el mensaje contrario.

Servicio de Reconexión Automática

La cerradura detectará cuando se haya desconectado de la red e intentará reconectarse con las credenciales almacenadas.

Servicio de Alarma por Puerta Abierta

La cerradura detectará, a través de su sensor de proximidad, cuando la puerta se encuentre abierta por un tiempo definido por el usuario desde el panel de control y hará sonar una alarma para alertar al usuario. Este servicio se levantará siempre después del servicio de apertura.

Lenguajes utilizados

Para la realización del software interno de la cerradura, se debió utilizar una combinación de lenguajes, debido a la variedad de componentes, servicios e interfaces que debían ser utilizadas para su funcionamiento.

Bash

Es un intérprete de consola y un lenguaje de comandos perteneciente al proyecto GNU de software libre. Este procesador de comandos, que por lo general se ejecuta directamente en una terminal de texto donde el usuario tipea comandos que ejecutan acciones, también puede ser ejecutado a partir de un archivo comúnmente denominado *script*. En nuestro caso, será la forma en que utilizaremos las funcionalidades de WiFi y manejo de la red dentro de la cerradura.

Python

Es un lenguaje de programación interpretado multiparadigma, ya que soporta orientación a objetos, programación imperativa y programación funcional. Un objetivo de diseño de Python es su facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++, pudiendo incluir Python en aplicaciones que necesitan una interfaz programable.
Esto último fue muy importante para nosotros porque tuvimos que incorporar una librería para el correcto funcionamiento e interacción con el módulo NFC que está programada en C, de manera que con una simple conversión de algoritmos pudo ser utilizada en Python sin ningún inconveniente.

El hilo de ejecución principal de la cerradura es ejecutado a través de un programa Python, el cual ejecuta, cuando sea necesario, los scripts correspondientes de Bash para la interacción con otros componentes de la cerradura. Esto se hizo así debido a que Python ya ofrece librerías para manejo de plaquetas programables para IoT y, más específicamente, cuenta con una librería para el manejo de pines de una Raspberry Pi.

Lenguaje C

Es un lenguaje de tipo de datos estáticos, débilmente tipificado, de medio nivel, ya que dispone de estructuras típicas de los lenguajes de alto nivel, pero a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

Finalmente, y como se mencionó anteriormente, la librería para la interacción con el componente de NFC adquirido para el funcionamiento físico de la cerradura está disponible en lenguaje C. De esta manera, se debió realizar el algoritmo de interacción y envío de mensajes por NFC inicialmente en lenguaje C para luego ser transcrito a Python, usando una librería específica que incorpora elementos y herramientas para el manejo de estructuras de menor nivel de las que ofrece por defecto Python. Una vez hecho esto, se incorporó la librería NFC como un módulo de extensión de Python para su funcionamiento desde el programa principal de la cerradura.

Base de datos de la cerradura

Para el almacenamiento de datos dentro de la cerradura, y como la información que debe guardar la cerradura en todo momento es de bajo volumen, más específicamente, sólo la identificación de dispositivos autorizados y un log temporal de accesos para luego ser enviado al panel central, se decidió utilizar simplemente un archivo como medio de almacenamiento. Sin embargo, si hubiéramos guardado esta información en un archivo de texto, cualquiera sea su codificación, sería muy engorrosa y extensa su lectura, escritura, *parsing* (o análisis) y codificación, de manera que se decidió utilizar una base de datos denominada SQLite.

SQLite es un sistema de gestión de bases de datos relacional, compatible con características de parámetros ACID, es decir, que permite la utilización de transacciones, que está contenida en una pequeña biblioteca escrita en C y cuyo proyecto pertenece al dominio público. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas

simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

5. Conclusión

El desarrollo de este proyecto nos ha aportado distintas experiencias en el ámbito personal, académico y profesional. Para nosotros el proyecto presentó un gran desafió que a la vez fue muy enriquecedor. Nos permitió conformar un equipo de trabajo donde cada uno tenía conocimientos más enfocados a diferentes tecnologías y áreas, como ser el desarrollo de software, comunicaciones e infraestructura o electrónica. El equipo que conformamos y la organización de tareas llevada a cabo permitió involucrarnos a todos en un proyecto que integrará todas estas áreas, permitiéndonos aprender y desarrollarnos en lo que era un ámbito desconocido en cuanto a la experiencia laboral de cada uno y así obtener como resultado la solución presentada.

Queremos destacar el conocimiento y prácticas que hemos obtenido a lo largo de la universidad que nos brindaron las herramientas para aprender y llevar adelante este proyecto, el cual nos permitió abordar un perfil no frecuente en nuestra carrera y el rápido aprendizaje y utilización de las nuevas tecnologías que van surgiendo con el paso del tiempo. Creemos que como resultado de nuestra formación académica estamos suficientemente capacitados para hacer frente a los avances de la tecnología para seguir ofreciendo productos y soluciones innovadoras.

En cuanto a la planificación inicial del proyecto creemos conveniente mencionar que, si bien la fecha de comienzo de este se pospuso por razones académicas y personales de cada uno de los integrantes del grupo, la duración de las etapas estimadas se mantuvo dentro de los plazos considerados para obtener un proyecto que permita alcanzar los objetivos generales y particulares que nos habíamos propuesto.

Al comienzo del proyecto nos resultó más difícil de lo planeado conocer y dominar las tecnologías elegidas para conseguir un ritmo de desarrollo constante. Luego con una leve demora en los tiempos pudimos comenzar a reutilizar componentes de código, lo que nos permitió ponernos al día y ganar tiempo al avance del proyecto.

Al tratarse de un desarrollo propio y no para un cliente externo, nos enfocamos en la prioridad de tener un producto funcional, aunque el proyecto se extendiera más allá de la planificación original, la cual de todos modos nos sirvió para ganar experiencia en la estimación de proyectos al embarcarnos con nuevas tecnologías.

Con este proyecto se buscó hacer un aporte a la industria de la domótica, mediante un producto innovador y competitivo con soluciones actuales, que permita mejorar el confort y otorgar seguridad a un edificio o vivienda, permitiendo no solamente la simplificación en la apertura de una puerta, sino también mantener una mejor administración y control más específico de los accesos.

Una de las motivaciones que nos guio durante el avance del proyecto para validar nuestro modelo de negocio fue la presentación y aprobación de este en la convocatoria de Becas Tics 2017 como instrumento de promoción y financiamiento por parte del FonSoft, la cual se detalla con mayor profundidad en el siguiente Anexo. Dicha evaluación del proyecto nos permitió ver y enfocar más a futuro al proyecto como la generación de un nuevo emprendimiento.

Finalmente, con la realización de este proyecto consideramos ya planteadas las bases de un sistema domótica enfocado en mejorar los aspectos de confort y seguridad en una edificación, mejorando los sistemas de cerraduras físicas y magnéticas comandadas por dispositivos pasivos

6. Referencias y Bibliografía:

[1] Agile Manifesto: http://agilemanifesto.org/iso/es/manifesto.html

[2] Kent Beck, Cynthia Andres: *Extreme Programming Explained: Embrace Change, 2nd Edition (The XP Series).* Addison-Wesley (November 26, 2004).

[3] ¿Qué es Arduino? Manual 30 proyectos con Arduino. https://www.programacionparacompartir.com/manual-30-proyectos-con-arduino/

[4] Historia de la Informática: Raspberry Pi. https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/

[5] José D. Sablón Brito: *Electroestimulación inalámbrica: Radio Frecuencia vs Bluetooth*. Madrid, España. 2017.

[6] Electronicsnotes: NFC Technology. https://www.electronics-notes.com/articles/connectivity/nfc-near-field-communication/technology.php

[7] DigitalGuide: ¿Qué es un servidor web y qué soluciones de software existen? https://www.ionos.mx/digitalguide/servidores/know-how/servidor-web-definicion-historia-y-programas/

[8] Node.js. https://nodejs.org/es/

[9] Esteban Borges: Servidor Base de Datos: ¿Qué es? Funciones, Tipos y Ejemplos. https://blog.infranetworking.com/servidor-base-de-datos/

[10] The Magpi Magazine: Raspberry Pi 3: Specs, benchmarks & testing. https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/

[11] The Magpi Magazine: Introducing Raspberry Pi Zero W. https://www.raspberrypi.org/magpi/pi-zero-w/

[12] Ayuda de Search Console: Proteger sitios web con el protocolo HTTPS. https://support.google.com/webmasters/answer/6073543?hl=es

[13] Microsoft Docs: Desarrollo de Software basado en Componentes. https://docs.microsoft.com/es-es/previous-versions/bb972268(v=msdn.10)

[14] Nicole Chapaval: Qué es Frontend y Backend. https://platzi.com/blog/que-es-frontend-y-backend/

[15] Oscar Blancarte: Autenticación con JSON Web Tokens. https://www.oscarblancarteblog.com/2017/06/08/autenticacion-con-json-web-tokens/

[16] Farlery, R.E.: *Managing and Leading Software Projects.* IEEE Computer Society, John Wiley & Sons, US, 2009

[17] Roger S. Pressman*. Ingeniería de Software, Un Enfoque Práctico*. Mc Graw-Hill. 7ma Edición, 2010.

[18] Canós, J., Letelier, P., Penadés, M. *Metodologías ágiles en el desarrollo de softwar*e. Proceedings VIII Jornadas de Ingeniería del software y Bases de Datos. Alicante, 2003

[19] Project Management Institute. *Guía de los fundamentos de la dirección de proyectos*. 3era. Edición. Newton Square, 2004.

[20] Don Wells. *Extreme Programming.* http://www.extremeprogramming.org/

[21] Sommerville, Ian. *Software Engineering*". 9na Edición. Pearson Educación, 2011

[22] Lea Karam: Diseño guiado por el dominio; beneficios clave. https://apiumhub.com/es/tech-blog-barcelona/diseno-guiado-por-el-dominio/

[23] David Walsh: WebSocket and Socket.IO. https://davidwalsh.name/websocket

[24] Android Developers: Host-based card emulation overview. https://developer.android.com/guide/topics/connectivity/nfc/hce

[25] ISO/IEC 7816-4:2013: Identification cards -- Integrated circuit cards -- Part 4: Organization, security and commands for interchange. https://www.iso.org/standard/54550.html

[26] StackExchange: GPIO Pinout Orientation RaspberyPi Zero W. https://raspberrypi.stackexchange.com/questions/83610/gpio-pinout-orientation-raspberypi-zero-w

[27] RF Wireless World: UART vs SPI vs I2C | Difference between UART,SPI and I2C. http://www.rfwireless-world.com/Terminology/UART-vs-SPI-vs-I2C.html

[28] Frederick Zhang: Connecting Waveshare 3.5″ Touchscreen and ITEAD PN532 NFC Module to Raspberry Pi. https://blog.onee3.org/2015/07/connecting-waveshare-3-5-touchscreen-and-itead-pn532-nfc-module-to-raspberry-pi/

[29] GNU Project: Bash Reference Manual. http://www.gnu.org/software/bash/manual/bash.html

[30] Wikipedia: C (lenguaje de programación). https://es.wikipedia.org/wiki/C_(lenguaje_de_programación)

Otros textos utilizados

[31] Yampier Medina Tarancón. Una taxonomía para la identificación de riesgos en
los proyectos de desarrollo de software de la Universidad de las Ciencias Informáticas. Universidad de las Ciencias Informáticas. La Habana, Cuba, 2012

[32] Boehm, B. *Tutorial: Software Risk Management*. Les Alamitos, CA, IEEE Computer Society, 1989.

[33] Sergio Fabian Ortiz Aguirre. *Near Field Communication*. Departamento de Electrónica e Informática de la Universidad Católica "Nuestra Señora de la Asunción", Paraguay, 2012.

[34] ISO/IEC. Standar 18092:2013 Information technology, Telecommunications and information exchange between systems, Near Field Communication, Interface and Protocol. 2013.

7. Anexo

a. Estudio de mercado - FONSOFT

BECAS TICs 2017
Ministerio de Ciencia, Tecnología e Innovación Productiva
Agencia Nacional de Promoción Científica y Tecnológica
Fondo Fiduciario de Promoción de la Industria del Software

El proyecto fue presentado en la convocatoria de Becas TICs 2017 del FONSOFT con código ***BECA FJPT SF018/17*** y cuya evaluación fue aprobada en la **Resolución N° 076/19,** pudiendo así validar el modelo de negocio presentado.

El impacto de los resultados del proyecto plantea la posibilidad de la sustitución de importaciones para lograr productos de mayor valor tecnológico a nivel nacional por sobre las cerraduras electrónicas extranjeras que se comercializan actualmente en el mercado. El proyecto planteó sustituir a las mismas con la oferta de un producto local más innovador y con fuertes aspectos de seguridad y control, logrando así una mejor relación precio-prestación. Por otro lado, se fortalece la industria nacional, lo que evita inconvenientes relacionados al proceso de importación de soluciones de cerraduras electrónicas desde el extranjero.

Como se mencionó previamente, el producto posee soluciones competidoras pero que no trabajan de forma interconectada, mucho menos con un sistema de administración centralizado, por lo tanto, este producto puede ser considerado innovador dada la posibilidad de incorporarlo en diversas empresas.

El modelo de negocios diagramado para la presentación del emprendimiento que surgió a partir del proyecto pretende ofrecer tres categorías del producto:
• Domiciliaria: Los dispositivos registrados podrán abrir todas las cerraduras y tendrá un número máximo de cerraduras y dispositivos posibles a administrar por central. Tendrá un registro básico de actividades de accionamiento de las cerraduras.
• Empresarial: A la categoría anterior se le agrega la posibilidad de ilimitadas cerraduras y dispositivos a administrar por central, mayor detalle en los reportes de actividades de acceso y la posibilidad de crear roles de usuarios para otorgar permisos específicos a distintas cerraduras del sistema en rangos de horarios configurables.
• Hotelera: Se le agregan configuraciones de tiempo de acceso para cada dispositivo registrado para poder configurar la utilización de la cerradura por parte de los huéspedes

para un rango de fechas especificado. Además, contará con un mayor nivel de categorización de las cerraduras administradas para una mejor gestión de estas. Se incorpora en la oferta de esta categoría un dispositivo dedicado para el emparejamiento y registro de los dispositivos móviles que utilizarán las cerraduras.

La versión inicial del sistema como se mencionó en el alcance del proyecto estuvo enfocada al desarrollo del sistema orientado a un ámbito hogareño la cual plantea menos requerimientos.

A continuación se listan los artículos solicitados para la puesta en marcha del emprendimiento al momento de la presentación a la convocatoria:

- Raspberry Pi + Fuente de alimentación + módulo NFC: En la etapa inicial se estableció adquirir 5 Raspberry Pi para Investigación y Desarrollo de la Solución planteada y las pruebas pertinentes
- Celular Android con NFC
- Producción de una cerradura con mecanismo electrónico (basado en promedio de precios de Cerraduras con mecanismo similares necesarios)
- Presupuesto destinado a publicidad, marketing y ventas
- Registro del dominio web de la marca
- Web Hosting:

En la etapa inicial de Investigación y Desarrollo se planteó producir 4 cerraduras a las que luego se integra el módulo con Raspberry Pi para su testeo y perfeccionamiento.
Con el 30% de presupuesto solicitado se planteó cubrir los gastos iniciales mencionados anteriormente y un excedente para cubrir posibles eventualidades.
Ya que el fuerte de la comercialización del producto se encuentra en la venta de una mayor cantidad de cerraduras frente a la cantidad de centrales de administración, se desestimó la ganancia de estas últimas y se distribuyó su costo, que solo considera una Raspberry Pi, entre la ganancia obtenida de las cerraduras.
En la etapa de producción, se estableció destinar el dinero restante de la solicitud de inversión para la fabricación de cerraduras destinadas a la comercialización.

La validación del modelo presentado para esta convocatoria se realizó con la consulta a profesionales con experiencia y contactos en la industria hotelera, quienes se presentaron como potenciales clientes y se obtuvo una retroalimentación positiva debido a la innovación y mejoras de seguridad en la gestión de accesos a habitaciones. Los profesionales consultados han tenido la oportunidad de viajar a otros países y observar e interactuar con

tecnologías similares mediante aplicaciones móviles y se logró observar durante el análisis que las empresas presentan tendencia a la utilización de cerraduras electrónicas para el control de acceso y mejoras de la seguridad, por lo que se presentan como potenciales clientes del producto ofrecido.

Por otro lado queremos destacar como validación del proyecto la creciente utilización de la tecnología NFC en aplicaciones móviles como en la billetera electrónica (e-wallet), la cual tiene gran aceptación por parte de los usuarios en diversos países del primer mundo.

b. Refinamiento de requerimientos definidos

En esta sección se presenta en mayor detalle el análisis de requerimientos y funcionalidades plasmado en historias de usuario propias de cada subsistema.

i. Historias de usuario del sistema administrativo

Historia 1: ABM Dispositivo Autorizado

El sistema debe permitir al administrador agregar, modificar y eliminar los dispositivos autorizados para la apertura de una cerradura. Para el alta de usuario se debe solicitar Nombre o Apodo que represente al dispositivo, tiempo de autorización (el tiempo que estará habilitado para la apertura de las cerraduras), cerraduras que se le permite abrir y por último en que cerradura se llevará a cabo la sincronización. Además, se permitirá definir un rango de horarios en el cual se le permitirá la operación del dispositivo, funcionalidad que será de baja prioridad. Estos datos serán todos obligatorios y podrán ser modificados por el administrador.

Historia 2: Sincronización de un dispositivo

Luego del alta de un dispositivo se debe sincronizar con el servidor, Para ello se debe acercar a la cerradura el dispositivo mientras la luz de sincronización parpadea, el servidor registrará una clave que identifique al dispositivo, y luego generará una clave única del sistema y se la enviará para que la registre.

Historia 3: Sincronización de claves con las cerraduras

Por cada cambio que se produzca en el registro de dispositivos de apertura, el servidor deberá sincronizar con todas las cerraduras los datos.

Historia 4: ABM Cerraduras

El administrador podrá realizar estas operaciones; seleccionando un botón de alta de cerradura, se mostrará en pantalla aquellas que estén en la red y no estén sincronizadas, cuando se selecciona una, esta tendrá un led que comenzará a destellar, y se le solicitará al usuario el ingreso de un nombre identificador para la cerradura. Una vez sincronizada el servidor le enviará los registros de dispositivos autorizados.

Historia 5: Generación de reportes

Se podrá visualizar un log de accesos que se podrá filtrar por cerradura, dispositivo y rango horario y podrá ser exportada o enviada por email.

Historia 6: Monitorización de las cerraduras

Se mostrará una lista de las cerraduras sincronizadas con su estado actual. Se notificará al administrador en caso de que una cerradura pierda la conexión o si presentara algún defecto.

Historia 7: Apertura de cerraduras vía web

Existirá la opción para poder abrir una cerradura de forma remota, desde el panel administrativo, cuando se seleccione una cerradura, se mostrará la opción para esta acción y se pedirá la confirmación.

Historia 8: Modificación de usuario

El administrador debe poder modificar su nombre de usuario y contraseña, en caso de tener la configuración por defecto del servidor, se deberá exigir el cambio de contraseña, y sugerir el cambio de nombre de usuario.

Historia 9: Nombre del sistema

El sistema deberá tener un nombre, por ejemplo “Casa Pepe”, el cual deberá ser ingresado la primera vez que se ingrese al sistema, y podrá ser modificado en cualquier momento por el administrador. Este nombre será solamente indicativo, además existirá un código único del sistema y no podrá ser visualizado ni modificado.

Historia 10: Bloqueo de cerradura

En el sistema se podrá seleccionar una cerradura (o varias) y se deberá poder seleccionar la opción “anular cerradura”, esta opción deberá bloquear la cerradura por un período determinado.

Historia 11: Suspensión de dispositivo

Cualquier dispositivo autorizado deberá poder ser suspendido desde un menú sin la necesidad de ser eliminado.

ii. Historias de usuario de la aplicación móvil

Historia 1: Estado del servicio

La aplicación no debe estar necesariamente visible para poder abrir una cerradura, es decir, simplemente se debe acercar el celular a la cerradura para abrirlo, sin tener que ejecutar un programa.

Historia 2: Almacenamiento de claves

El dispositivo debe tener guardado en una base de datos las claves que le envíe el servidor.

Historia 3: Mensaje de error

En caso de que el dispositivo no esté autorizado por el servidor para abrir una cerradura, se enviará una notificación al mismo para informar la situación.

Historia 4: Sincronización del dispositivo

Se deberá seleccionar una opción para el momento de sincronizar el dispositivo a un sistema, cuando esta acción es realizada, en ese momento se deberá aproximar el dispositivo a la estación de sincronización seleccionada por el sistema administrativo. Si no se concreta la operación en un período, se mostrará una notificación. De la misma manera, se notificará si existe algún inconveniente y no se pudo sincronizar.

Historia 5: Conexión de una cerradura a la red

Existe la opción de conectar una cerradura a la red para aquellas que no se encuentren en la LAN, para poder seleccionar esta opción se debe haber conectado previamente a la señal de Wi-Fi emitida por la cerradura. Una vez seleccionada la opción se deberán listar los SSID de las señales Wi-Fi que se encuentran disponibles, una vez seleccionada la opción deseada, se deberá ingresar la contraseña.

iii. Historias de usuario del sistema de la cerradura

Historia 1: Establecimiento de la conexión

Para poder conectar por primera vez la cerradura a la red interna, se deberá presionar el botón de “sincronización / reset” de la cerradura, luego el dispositivo deberá comenzar a emitir una señal Wi-Fi con un SSID identificativo de la cerradura, una vez un dispositivo móvil se conecte a ella, y realice el procedimiento mencionado en la *Historia 5* de la aplicación móvil, la cerradura se conectará a la red Wi-Fi y dejará de emitir su SSID. En el caso de no ser conectada, se dejará de emitir la red luego de un tiempo predefinido.
Estas credenciales serán almacenadas en la base de datos.

Historia 2: Restablecimiento de configuración

Al presionar el botón de “sincronización / reset” se deberá también eliminar toda la información referente al servidor de cerraduras y los dispositivos sincronizados.

Historia 3: Reconexión automática

Si la cerradura pierde la conexión en la LAN, deberá estar constantemente buscando la red y cuando ésta sea encontrada se re-conectará con las claves almacenadas.

Historia 4: Servicio de acceso

Se deberá escuchar por una comunicación NFC con un dispositivo. Mediante la misma se recibirá una clave de servidor y de dispositivo que se contrastará con las claves almacenadas en la cerradura. En caso de haber una coincidencia, se activará el mecanismo de la cerradura.

Historia 5: Cierre de puerta

La cerradura deberá activarse para el bloqueo de la puerta una vez que esta sea cerrada y pase un tiempo mínimo predefinido.

Historia 6: Alarma de puerta abierta

Si la cerradura fue activada, se activará un temporizador por un tiempo predefinido, o el tiempo definido por el usuario previamente, y una vez concluido este período, si la puerta no se encuentra cerrada, se activará una alarma indicativa de la situación.

Historia 7: Sincronización de datos de dispositivos

La cerradura deberá quedar esperando notificaciones del servidor sobre actualizaciones de los dispositivos sincronizados.

Historia 8: Registro de actividad

Cada dispositivo cerradura deberá llevar su propio log de actividad de cuándo es abierta y por qué dispositivo habilitado y sincronizará con el servidor central dichos registros.

Historia 9: Bloqueo de cerradura

La cerradura podrá ser bloqueada por el administrador, esto significa que, si se encuentra en este estado, no podrá ser accionada por el sistema, solamente se podrá abrir de forma física.

iv. Refinación de historias del sistema administrativo

Número: A001	Nombre: ABM Dispositivo Autorizado

Descripción:

Desde el panel principal de administración, una vez autenticado el usuario, se debe poder seleccionar la opción de agregar, modificar o eliminar los dispositivos autorizados.

El proceso de **alta** debe ser el siguiente:

1. Se requiere un nombre para el dispositivo.
2. Se requiere la selección de las cerraduras habilitadas para el dispositivo.
3. Se requiere la selección de una cerradura con la cual se emparejará el dispositivo.
4. Se solicita la confirmación del usuario.
5. Comienza el proceso de sincronización (Historia 002).
6. Se notifica el resultado al usuario.

a. Un mensaje de éxito en tal caso, y se retorna al menú principal.

b. Un mensaje de error si la sincronización no fue concretada y se muestran las opciones “Reintentar” y “Volver al menú principal”.

i. Si se selecciona “Reintentar” se vuelve al punto 5.

7. Comienza la *Historia 003*.

El proceso de **baja** es:

1. Se listan los dispositivos registrados.
2. Se selecciona el dispositivo a dar de baja.
3. Se requiere confirmación del usuario.
4. Se elimina de la base de datos y se muestra un mensaje de éxito de operación.
5. Se vuelve al menú principal.
6. Se comienza la *Historia 003*.

El proceso de **modificación** es:

1. Se listan los dispositivos registrados.
2. Se selecciona el dispositivo a modificar.
3. Se muestran los datos registrados del dispositivo con la opción de modificarlos:

a. Nombre del dispositivo.

b. Cerraduras donde está autorizado.

4. Se selecciona la opción “Aceptar”
5. Se solicita la confirmación de la operación.
6. Se registran los cambios.

7. Se muestra un mensaje de éxito. 8. Se comienza la *Historia 003*.
Observaciones: Si una cerradura seleccionada para el dispositivo se encuentra en estado “bloqueada” (Historia A010) se mostrará como mensaje de alerta al momento de aceptar el alta o modificación. Esto **no** afectará al curso normal del procedimiento, sólo será una advertencia.

Número: A002	Nombre: Sincronización de un dispositivo
Descripción:	
1. Se le envía a la cerradura seleccionada el comando para que inicie la secuencia de sincronización. 2. Se aguarda a que la cerradura notifique la conexión del dispositivo. 3. Se genera una clave única de dispositivo y se le envía a la cerradura. 4. Se aguarda a la recepción de datos de sincronización desde la cerradura. 5. Se registran los datos en la base de datos y se envía la confirmación de recepción a la cerradura.	
Observaciones: En los puntos 2 y 4 se aguarda durante un período (a definir), transcurrido el mismo si no se obtienen notificaciones se cancelará el proceso de sincronización y se notificará el error, volviendo al *punto 5* de la *Historia 001.*	

Número: A003	Nombre: Sincronización de claves con las cerraduras
Descripción:	
Tras efectuarse una acción de ABM Dispositivo, se debe enviar la información a todas las cerraduras: - Dispositivos autorizados para abrirla (independiente de cada cerradura). - Clave/s identificadora/s de los dispositivos (asociadas con cada uno). Para esto se recorre la lista de cerraduras emparejadas con el servidor.	

Observaciones: La información y claves de los dispositivos que serán solicitadas se especificarán en la instancia de evaluación y diseño de **seguridad** del proyecto.

Número: A004	Nombre: ABM Cerradura
Descripción:	
Desde el panel principal de administración, una vez autenticado el usuario, se debe poder seleccionar la opción de agregar, modificar o eliminar cerraduras. Para el **alta** de una nueva cerradura se llevará a cabo el siguiente procedimiento: 1. Se realiza un broadcast en la red para descubrir cerraduras conectadas. 2. Se listan las cerraduras conectadas a la red que aún no hayan sido registradas. 3. Se permite la selección de una de estas para su configuración. 4. Se envía una notificación a la cerradura seleccionada para iniciar su comando de sincronización. 5. Se espera la confirmación de la cerradura para continuar la sincronización. 6. Se requiere un nombre identificatorio para la cerradura. 7. Se solicita la confirmación del usuario. 8. Se registra la cerradura con su nombre e identificador de red en la base de datos. 9. Se envía a la cerradura la notificación de éxito y comienza el procedimiento descrito en la *Historia 003*. En caso de una **modificación**, se realizará lo siguiente: 1. Se listan las cerraduras registradas. 2. Se permite la selección de una de ellas para su modificación. 3. Se permite la edición de su nombre. 4. Se solicita la confirmación del usuario. 5. Se registran los cambios en la base de datos. 6. Se notifica al usuario que la operación se realizó con éxito. Si se trata de una **baja**, entonces: 1. Se listan las cerraduras registradas. 2. Se permite la selección de una de ellas para su eliminación. 3. Se solicita la confirmación del usuario. 4. Se envía un comando de restablecimiento de configuración a la cerradura. 5. Se espera a la recepción de la confirmación del comando. 6. Se elimina el registro correspondiente de la base de datos. 7. Se notifica al usuario que la operación se realizó con éxito.	

Observaciones:

Número: A005	Nombre: Generación de reportes
Descripción:	
En el menú principal de administración se contará con la funcionalidad de “Ver registros”, una vez ingresada en esta funcionalidad se mostrará una lista con los últimos 20 registros (lista por defecto) y luego se visualizarán filtros para: - Ampliar la cantidad de registros visualizados (con las opciones 5, 10, 20, 50 y 100). - Listar los accesos por cerradura. - Listar los accesos por dispositivo. - Listar los accesos para una fecha exacta. - Listar los accesos en un intervalo de fechas. La selección de filtros podrá ser múltiple, para que el usuario pueda combinarlos. Para que se pueda cargar el registro según el filtro que se seleccione, se mostrará un botón de “Crear reporte”. Este menú debe contar con la opción de “enviar el reporte” por correo electrónico, y se solicitará una dirección de correo para enviar. Habrá un botón de confirmación y uno de cancelación en este paso.	
Observaciones:	

Número: A006	Nombre: Monitorización de las cerraduras
Descripción:	
Existirá la funcionalidad de “Ver estado de cerraduras” que se mostrará en el menú principal, en donde se visualizará una lista de las cerraduras pertenecientes al sistema con un círculo de color al lado, que se verá verde si la cerradura es alcanzable por el servidor y rojo en caso contrario, también se podrá visualizar un mensaje de alerta si una cerradura se encuentra bloqueada (Historia A010). Además, si una cerradura pierde	

conectividad, luego de un período a determinar, se enviará un email informando la situación al administrador.
Observaciones:

Número: A007	Nombre: Apertura de cerraduras vía web
Descripción:	
En el menú principal, se verá la opción de “accionar cerradura”. Ingresado en esta funcionalidad se listará las cerraduras y cuando se seleccione una de ellas se visualizará la opción de “accionar”, una vez seleccionada esta opción, se pedirá una confirmación por parte del usuario, y en caso de ser aceptado, se enviará la señal a la cerradura para su accionamiento. Por último, se visualizará un mensaje de éxito (o de error en caso de surgir un problema).	
Observaciones: Si la cerradura se encuentra en estado “bloqueada” (Historia A010) no se podrá accionar la cerradura desde el sistema, por lo tanto, se deberá mostrar un mensaje de error indicando la situación y se sugerirá desbloquear la cerradura para que pueda ser accionada.	

Número: A008	Nombre: Modificación de usuario
Descripción:	
El en menú principal, se mostrará una opción para modificar las credenciales de acceso al sistema administrativo. Al acceder a esta pantalla, se podrá cambiar tanto el nombre de usuario como la contraseña de acceso, con su correspondiente campo de verificación. Antes de completar la operación se le pedirá la confirmación al usuario. Si el usuario está accediendo al sistema por primera vez, se mostrará como inicio esta pantalla requiriendo al usuario que cambie los valores otorgados por defecto.	
Observaciones:	

Número: A009	Nombre: Nombre del sistema

Descripción:
Desde el menú principal se debe mostrar una opción para poder cambiar el nombre del sistema, permitiendo el ingreso de cualquier cadena alfanumérica. Este nombre será simplemente indicativo para el usuario. La primera vez que se ingrese al sistema se sugerirá al usuario que cambie el nombre por defecto que será “KeyWi”.
Observaciones:

Número: A010	Nombre: Bloqueo de cerradura/s
Descripción:	
Existirá la opción en el menú principal de bloqueo de cerraduras, en el cual se listará las cerraduras, en la lista las cerraduras deberán tener un campo de selección donde se podrán elegir entre 1 y todas las cerraduras, y deberá haber un botón de selección de todas. Además, se podrá seleccionar un período en el cual las cerraduras permanecerán bloqueadas, o una opción de bloqueo por tiempo indeterminado. Una vez seleccionadas las cerraduras que se deseen bloquear y el período de bloqueo, el usuario pulsará un botón de “aceptar”, luego de esto se mostrará un mensaje donde se solicita la confirmación por parte del usuario y una vez confirmado se procederá marcará esa cerradura como “bloqueada” en la base de datos. El sistema deberá enviarle a la cerradura una señal de bloqueo. Se mostrará un mensaje de éxito y se preguntará si se quieren bloquear más cerraduras o volver al menú principal.	
Observaciones:	

Número: A011	Nombre: Suspensión de dispositivo
Descripción:	
En el menú principal se podrá acceder a la opción de “suspender un dispositivo”. En ella se listan los dispositivos autorizados y se verá un campo de “Tiempo de suspensión”, y un botón de “Siguiente”. Se deberá poder seleccionar 1 o más dispositivos, y se requerirá ingresar un tiempo de suspensión (deberá existir una opción “por tiempo indeterminado”). Una vez ingresados los datos y presionado el botón siguiente, se deberá solicitar confirmación nuevamente por parte del usuario y luego de esto se mostrará un mensaje de éxito y luego se procederá a marcar el dispositivo como “suspendido” en la base de datos. Se mostrará un mensaje de éxito y se preguntará si se quieren suspender más dispositivos o volver al menú principal.	
Observaciones:	

v. Refinación de historias de la aplicación móvil

Número: M001	Nombre: Estado del servicio
Descripción:	
La aplicación tendrá un servicio que se mantendrá en ejecución en segundo plano para escuchar cuando el dispositivo se acerque al emisor NFC. Al detectar esto, se comenzará el procedimiento de apertura de la cerradura: 1. La cerradura le envía su clave de sistema. 2. La aplicación busca en su base de datos el valor correspondiente asignado para la clave de ese sistema. 3. La aplicación envía este valor junto con una clave de identificación física. 4. Se espera la respuesta de la cerradura. 5. Se le muestra una notificación al usuario acerca del resultado, ya sea exitoso o no.	
Observaciones: Los datos intercambiados y los métodos de encriptación y transferencia serán definidos en la etapa de diseño de seguridad del sistema.	

Número: M002	Nombre: Almacenamiento de claves
Descripción:	
La aplicación mantendrá su propia base de datos donde almacenará las claves que le sean enviadas por los sistemas de administración de cerraduras con los que esté sincronizada. Las mismas se guardarán en una estructura de clave-valor donde la clave será la clave de identificación del sistema y el valor la clave que identifique al dispositivo en el susodicho.	
Observaciones: Queda pendiente la revisión de cuestiones pertinentes a la eliminación de las claves y cómo verificar que se siguen utilizando.	

Número: M003	Nombre: Mensaje de error
Descripción:	

Si el intento de apertura de cerradura es **rechazado** por el servidor, se mostrará un mensaje indicando que *el dispositivo no se encuentra autorizado a la apertura de esa cerradura.*
Observaciones: No es un mensaje de falla de apertura o de negociación con el servidor, es una falta de autorización.

Número: M004	Nombre: Sincronización del dispositivo
Descripción:	
En el menú principal de la aplicación, existirá la opción para sincronizar un sistema. Cuando se seleccione la opción, se deberá mostrar un mensaje indicando que se aproxime el dispositivo a la cerradura de sincronización (remarcando que en ella debe estar parpadeando un LED indicativo). Si no se aproxima el dispositivo en un lapso se deberá mostrar un mensaje de error indicando que no se realizó la operación, que intente nuevamente. Cuando se aproxime el dispositivo se comenzará el *intercambio de información* necesaria para el sistema. Si el proceso falla en cualquier momento del intercambio se mostrará un mensaje de error y se solicitará que comience nuevamente el procedimiento.	
Observaciones: Evaluar la opción de que, si se pierde la conexión NFC con la cerradura, detener el flujo principal, solicitar aproximar nuevamente el dispositivo, y luego de realizada esta acción, continuar con la tarea suspendida.	

Número: M005	Nombre: Conexión de una cerradura a la red
Descripción:	
En el menú principal se mostrará una opción de “Conectar cerradura a la red LAN”. Para esto el dispositivo debe estar conectado a la red propia de la cerradura, si no se encuentra conectado, al momento de seleccionar esta funcionalidad, se mostrará un mensaje indicando que “El dispositivo no se encuentra conectado a la red de una cerradura, debe conectarse a una y luego volver a seleccionar esta opción”. Ni bien se seleccione la opción, el dispositivo le solicitará a la cerradura la lista de redes	

disponibles y luego ingresará al menú de sincronización. Una vez entrado en el menú de sincronización, se mostrará un listado con los SSID que se recibieron de la cerradura, y cuando se seleccione uno de ellos se solicitará ingresar la clave de la red (esta opción deberá aparecer por defecto en forma oculta, mostrando el ingreso de cada caracter como un punto o un asterisco, y se podrá elegir una opción de “ver clave”). Cuando se ingresen los datos, se enviará la información a la cerradura, y se aguardará a la confirmación. En caso de ser exitosa la operación se mostrará un mensaje indicándolo. Si existe algún error, se informará qué clase de error es y se esperará nuevamente la selección de datos.
Observaciones:

vi. Refinación de historias del sistema de la cerradura

Número: C001	Nombre: Establecimiento de la conexión
Descripción:	
Una vez arrancado el sistema, se quedará a la espera de la señal de acción del botón de “sincronización / reset”. Al recibir este evento, se comenzará a transmitir una señal WiFi y esperará la recepción de la configuración desde la aplicación móvil, operación descrita en la *Historia M005.* Al lograr una conexión exitosa, se notifica a la aplicación móvil y se deja de transmitir su red propia. Si transcurren 120 segundos desde presionado el botón y no se han recibido datos de configuración de la red, se dejará de transmitir la red propia y el dispositivo volverá al mismo estado que al iniciarse.	
Observaciones:	

Número: C002	Nombre: Restablecimiento de la configuración
Descripción:	
Si se mantiene presionado el botón de restablecimiento de configuración durante 15 segundos, la configuración actual de la cerradura será borrada y se volverán a colocar los valores por defecto.	
Observaciones:	

Número: C003	Nombre: Reconexión automática
Descripción:	
Si la cerradura pierde la conexión de la LAN a la que se encuentra conectada, se deberá quedar buscando la señal de la misma hasta que la encuentre, y en caso de hacerlo conectarse automáticamente. Además, deberá indicar en un led de falla de conexión la situación.	

Observaciones:

Número: C004	Nombre: Servicio de acceso
Descripción:	
La cerradura quedará expectante de una señal NFC de un dispositivo móvil, cuando comience la transmisión de datos, solicitará al dispositivo las credenciales para el accionamiento del mecanismo de apertura, en caso de que sean válidos, se accionará y se abrirá la puerta.	
Observaciones:	

Número: C005	Nombre: Cierre de puerta
Descripción:	
Tras el accionamiento de apertura de la cerradura, descrito en la *Historia C004*, se esperará un tiempo predefinido y si se detecta que la puerta está cerrada, se accionará nuevamente el pestillo para trabarla. En caso de que se detecte que la puerta está abierta, se realizarán las acciones descritas en la *Historia C006*.	
Observaciones:	

Número: C006	Nombre: Alarma de puerta abierta
Descripción:	
En el caso de que en el transcurso de la historia C005 la puerta no sea cerrada en el período definido para dicho acto, comenzará a sonar una alarma para avisar al usuario dicha situación, en el momento que la puerta es cerrada, esta dejará de sonar y comenzará la historia C005	
Observaciones:	

Número: C007	Nombre: *Sincronización de datos de dispositivos*
Descripción:	
Deberá existir un servicio en un puerto a definir para que el servidor solicite la transferencia de nueva información de dispositivos autorizados. Cuando el servidor solicite la comunicación, se le solicitará al mismo la información con las credenciales del dispositivo y se actualizará la base de datos con la información enviada por el servidor.	
Observaciones:	

Número: C008	Nombre: *Registro de actividad*
Descripción:	
Se llevará un registro de actividad en una base de datos local con fecha, hora, su identificación de cerradura y la de los dispositivos involucrados. Las actividades a registrar con todas las relacionadas con las historias C004, C005 y C006.	
Observaciones:	

Número: C009	Nombre: *Bloqueo de cerradura*
Descripción:	
Cuando a la cerradura reciba la señal de bloqueo por parte del servidor, esta detendrá el servicio NFC de transferencia de datos y quedará en estado bloqueada, pudiendo ser solamente accionada de forma física. Quedará en este estado hasta que el servidor lo indique.	
Observaciones:	