Universidad San Jorge

Escuela de Arquitectura y Tecnología

Grado en Diseño y Desarrollo de Videojuegos

Proyecto Final

IoT Asistencial

Autor del proyecto: Jesús Lacarte Carazo

Director del proyecto: Jorge Echeverria

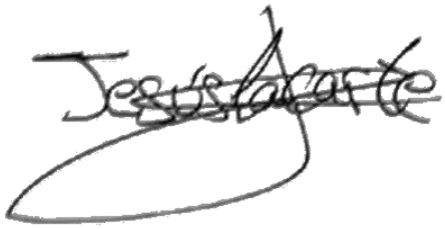
Zaragoza, 11 de septiembre de 2020

Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Diseño y Desarrollo de Videojuegos por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma: Fecha 11/09/2020



Dedicatoria y Agradecimiento

Primero me gustaría agradecer a los profesores que me han apoyado durante los años de la carrera, también quiero agradecer a la empresa Inycom por darme la oportunidad de realizar el trabajo de fin de grado en colaboración con ellos.

Y finalmente quiero agradecer a mis dos directores de proyecto, Jorge Echeverria director del proyecto por parte de la Universidad San Jorge, y Eduardo Morales director del proyecto por parte de Inycom, que me han guiado a lo largo de todo el proceso que ha sido la implementación y desarrollo de este proyecto.

Tabla de contenido

Índice de Figuras

Índice de tablas

Glosario

- **IoT:** Internet of Things
- **GPIO:** General Purpose Input/Output
- **API:** Application Programming Interfaces
- **IDE:** Integrated Development Environment
- **CRUD:** Create, Read, Update and Delete
- **Front-end:** Parte de software que interactua con el usuario.
- **Back-end:** Parte del software que procesa la entrada que proviene del front-end.
- **Endpoint:** Elemento que sirve para la comunicación del front-end con el back-end mediante. una URI.
- **GDPR:** General Data Protection Regulation

Resumen

Internet of Things (IoT) y el almacenamiento y control de la identidad digital de un usuario son dos de las tecnologías que se encuentran en la cima del mundo tecnológico actualmente. En este proyecto se han combinado ambas tecnologías para crear un producto de IoT Asistencial. Con este producto se busca facilitar la vida o algunas acciones a las personas que utilicen el producto que se ha desarrollado. El proyecto que se ha implementado es un sistema de una puerta inteligente utilizando la tecnología de identidad digital, utilizando la identidad digital en el proyecto se ha buscado crear una cerradura cuyo aspecto más importante sea la seguridad. En el mercado de las puertas inteligentes la primera característica que busca un usuario a la hora de adquirir una puerta inteligente es que sea segura, en este proyecto con la ayuda de la identidad digital se ha creado una puerta inteligente más segura que muchas de las puertas inteligentes que existen en el mercado actual.

Abstract

Internet of Things (IoT) and the storage and control of a user's digital identity are two of the technologies that are currently at the top of the technological world. In this project, both technologies have been combined to create a Healthcare IoT product. The objective of this product is to make life or some actions easier for the people who use the product that has been developed. The project that has been implemented is a smart door system using digital identity technology, using digital identity, the project has search to create a lock whose most important aspect is security. In the smart door market, the first characteristic that a user looks when acquiring a smart door is secure, in this project with the help of digital identity a smarter door has been created safer than many of the doors smart devices that exist in today's market.

1. Introducción

La definición de Internet of Things, IoT, es la agrupación e interconexión de diferentes dispositivos y objetos mediante una red, esta red puede ser privada o puede ser Internet. En los últimos años la tecnología IoT ha ido creciendo a una gran velocidad llegando a convertirse en una de las tecnologías más importantes que existen en la actualidad. Brendan O'Brien arquitecto jefe y cofundador de Aria Systems dijo "If you think that the internet has changed your life, think again. The Internet of Things is about to change it all over again!". En la Figura 1, se puede observar en la una aproximación de la evolución de la tecnología IoT en los últimos y los próximos años [1].

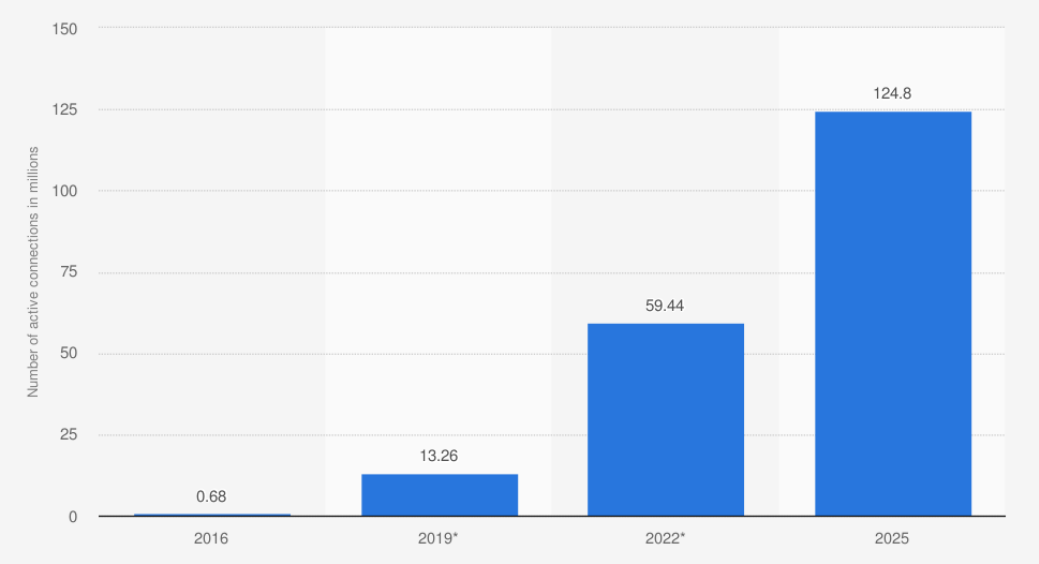


Figura 1: Evolución de la tecnología IoT (ETNO - Annual Economic Report 2019)

IoT permite que los objetos físicos puedan compartir y recopilar datos con una mínima intervención humana. Respecto al tipo de objetos o dispositivos que se utilizan pueden ser desde sensores, dispositivos mecánicos, hasta objetos cotidianos como electrodomésticos o ropa. Una de las aplicaciones de IoT más populares en la actualidad es la domótica, domótica es el conjunto de sistemas capaces de automatizar una vivienda o edificio, estos sistemas permiten una gestión eficiente de los diferentes aspectos que pueden preocupar a una persona, como por ejemplo aspectos energéticos, seguridad, accesibilidad… Un ejemplo de lo que puede lograr la domótica sería el control de diferentes aspectos del hogar como luces, calefacción, ventanas…, a distancia desde un dispositivo móvil.

La domótica es solo un ejemplo de las muchas aplicaciones que ofrece la tecnología IoT en la actualidad además de las aplicaciones que pueden surgir en el futuro ya que IoT es una de las tecnologías que más crecen en la actualidad.

Otro de los aspectos principales del proyecto es la identidad digital, el conjunto de datos, información y cualquier rastro de una persona en internet forman lo que se llama perfil o identidad digital de esa persona. La identidad digital puede sonar un poco peligroso puesto que es información de una persona que se encuentra en internet donde puede ser encontrada por otras personas, pero con una seguridad adecuada puede tener importantes usos.

En la actualidad donde la tecnología e internet forma parte de todo el mundo poder utilizar la identidad digital puede facilitar algunos aspectos, por ejemplo, si se utiliza una herramienta adecuada de identidad digital puedes firmar y autorizar diferentes transacciones, además puedes usar herramientas de identidad digital para añadir seguridad a diferentes operaciones.

Este proyecto se va a centrar en las soluciones de IoT de apertura y cierre de puertas y se va a desarrollar una solución que proporcione una seguridad mayor, para ello se va a utilizar la identidad digital. La seguridad es uno de los aspectos más importantes en muchos proyectos, pero en los que se encuentran relacionados con el control de cerraduras lo, es más. Con el uso de una aplicación de identidad digital se va a permitir que el usuario cree un perfil digital seguro y a partir de eso el usuario podrá utilizarlo para acceder a la cerradura.

El objetivo del proyecto es desarrollar una cerradura inteligente básica pero que se centre en proporcionar seguridad, además se busca que el acceso pueda ser regulado creando un sistema de invitados o personas autorizadas a acceder a la cerradura, pero con un acceso limitado y controlado por el propietario de la cerradura.

Con el desarrollo del proyecto que se comenta se puede lograr la introducción de una nueva tecnología en el ámbito de la seguridad en un mercado donde la seguridad es el principal objetivo. Mediante un correcto uso de la identidad digital se puede crear nuevas cerraduras inteligentes más seguras que las que están en el mercado actualmente, lo que es posible que produzca una evolución del mercado de puertas inteligentes hacía una nueva fase.

2. Antecedentes / Estado del Arte

IOT (Internet of Things) es un paradigma de internet que logra que muchos elementos puedan conectarse a internet, desde un robot, o un electrodoméstico, puertas…, logrando así una gran cantidad de aplicaciones posibles, como por ejemplo la domótica, que logra facilitar la vida del usuario permitiéndole un mayor control sobre elementos de su hogar.

IOT se encuentra en plena expansión y en estos últimos años ha conseguido un lugar en el mercado logrando posicionarse en los primeros lugares del mercado tecnológico lo que implica que existan una gran variedad de productos de IoT en el mercado actual.

En el mercado existen diferentes ejemplos de proyectos de puertas que utilizan IOT de una forma similar a como se va a mostrar en este proyecto y por ello se va a realizar una investigación de sus características y funcionamiento para poder implementar un producto que pueda competir con estos ejemplos en el mercado actual.

Los productos que se encuentran en la cima del mercado de puertas con tecnología IoT o puertas inteligentes, están desarrollados por grandes compañías las cuales cuentan con grandes equipos y por ello sus productos son mejores en varios aspectos a lo que el producto que se desarrolla, uno de los ejemplos más claros es que estos productos están centrados en ofrecer una gran variedad de modos de acceso además de que su aspecto es bastante atractivo para el cliente.

A pesar de que en aspectos como los comentados el producto que se va a desarrollar no puede competir con ellos, debido al tipo de proyecto, en este proyecto se busca competir con estos productos implementando una nueva funcionalidad en este tipo de productos, para ello se va a utilizar tecnología de identidad digital.

2.1. Ejemplos del mercado

A continuación, se va a mostrar algunos de los productos más utilizados en el mercado actual de puertas inteligentes.

2.1.1. Samsung Smart Doorlock

Samsung Smart Doorlock [2] es una puerta desarrollada por Samsung que utiliza la tecnología IOT para ofrecer diferentes funcionalidades. En 2018 Samsung ganó el premio IOT Innovation Award gracias a este proyecto.

Samsung Smart Doorlock ofrece la posibilidad de abrir la puerta con diferentes métodos:

- Aplicación de Smartphone
- Teclado táctil
- Lector de huella

Además de estas funcionalidades con la aplicación desarrollada para este producto puedes conceder el acceso a otras personas, puede ser acceso limitado, por ejemplo, un día en una franja horaria, y puedes ver en tiempo real el historial de accesos a tu casa o el lugar donde tengas instalado el producto que ofrece Samsung.

2.1.2. August Smart Lock Pro

Otra de las puertas inteligentes más vendidas en la actualidad es August Smart Lock Pro [3].

August Smart Lock Pro es una puerta desarrollada por la compañía August Home y utiliza la tecnología IOT para crear una puerta inteligente funcional.

August Smart Lock Pro ofrece un control remoto de la puerta, es posible cerrar o abrir la puerta desde cualquier lugar, siempre que tengas acceso a internet en tu smartphone, ofrece la posibilidad al cliente de saber el estado de la puerta en tiempo real y también proporciona al cliente la posibilidad de dar acceso a invitados en la franja horaria y día que él elija.

También ofrece la posibilidad de activar la función Auto Lock, que cierra la puerta a los 30 minutos de que una persona ha salido por ella, evitando que el cliente deje su casa abierta por un despiste.

Además, August Smart Lock Pro es compatible con tecnologías que utilizan comandos de voz como Amazon Alexa y Google Assistant.

2.1.3. Tesa Assa Abloy ENTR

Y como último ejemplo, ENTR de Tesa Assa Abloy [4], ENTR es una cerradura electrónica inteligente cuyo principal objetivo es proporcionar una cerradura segura y de fácil instalación.

ENTR ofrece los siguientes métodos de acceso:

- Aplicación de Smartphone
- Mando a distancia
- Teclado táctil
- Lector de huella

Este último ejemplo posee menos características que los anteriores, pero en cambio se centra en una fácil instalación.

2.2. Comparativa de mercado con el producto a desarrollar

Además de proyectos creados y comercializados por grandes empresas tecnológicas, como los nombrados anteriormente, en internet existen muchos ejemplos de proyectos de IoT, y entre todos esos proyectos también existen proyectos de puertas que utilizan IOT para su funcionamiento.

	Samsung Smart Doorlock (SHP-DR708)	*August Smart Lock Pro*	Tesa Assa Abloy ENTR	*IoT Proyect*
Alimentación	***8 AA alkaline 1.5 V batteries***	***4 AA alkaline 1.5 V batteries***	***1 Litio Ion 3.2 V***	***Conectada a la corriente + SAI***
Requerimientos Móvil	***Android V5.0/ iOS V10.0***	***Android V6.0/ iOS V11.0***	***Android V4.3/ iOS V8.0***	***Android V4.3/ iOS V9.0***
Modo de acceso	***Smartphone app, Huella, Teclado, Key de emergencia***	***Smartphone app, Smartphone Bluetooth***	***Smartphone app, Huella, Teclado, Mando a distancia***	***Smartphone app + Lector QR***

Tabla 1: Comparativa entre las cerraduras

Como se ha mostrado existen muchas puertas que utilizan IoT de forma similar al mostrado en este proyecto y que en varios aspectos son mejores que el proyecto a desarrollar, por eso se necesita algún elemento diferenciador, algo que es necesario para que el producto tenga éxito en el mercado. Ese distintivo es la identidad digital, la identidad digital es el conjunto de información que identifica a una persona en internet, el uso de identidad digital en el producto añade una mayor seguridad a la cerradura, uno de los aspectos más importantes de este mercado.

2.3. Identidad Digital

En el mundo actual donde el internet está en cualquier aspecto de la vida de una persona es necesario que cada persona tenga un control de su identidad o perfil digital, existen muchas aplicaciones que sirven para proteger la identidad digital de los usuarios y que les ofrece un uso seguro de ella para realizar diferentes operaciones como por ejemplo firma de documentos o autorización de operaciones.

Para el uso de identidad digital en el proyecto se va a utilizar la aplicación llamada Wallet Id [5] porque además de ofrecer un robusto sistema está desarrollado por la empresa con la que se realiza el trabajo de fin de grado lo que proporciona al desarrollador un mayor acceso a información sobre la aplicación y la posibilidad de apoyo en caso de la existencia de algún problema en el desarrollo.

2.3.1. Wallet Id

Wallet Id es una aplicación desarrollada por Inycom [6], que permite una identificación sencilla y segura, utilizando dispositivos móviles, para la autorización de operaciones y firma electrónica de documentos.

Además Wallet Id ofrece un almacenamiento seguro de la identidad o perfil digital de un usuario para luego su posterior uso para realizar operaciones como las nombradas anteriormente o para permitir identificar al usuario mediante su identidad digital, este último aspecto es el que se va a utilizar para identificar a la persona que quiera acceder a la cerradura, lo que va a proporcionar más seguridad que cualquiera de los métodos que implementan las otras puertas o cerraduras inteligentes que se ha estudiado en las secciones anteriores. Wallet Id ofrece utilizar su servicio mediante web o aplicación móvil, para este proyecto se va a utilizar la aplicación móvil Wallet Id TCV.

3. Objetivos

3.1. Objetivos del proyecto

En este proyecto se pueden vislumbrar cinco objetivos principales.

1) Dimensionar el proyecto a desarrollar:

   El primer paso es definir un proyecto sobre IoT Asistencial que se ajuste a las características del tipo de proyecto adecuado como un trabajo de final de grado.

2) Analizar y seleccionar los sensores y dispositivos más apropiados para la realización del proyecto:

   Una vez el proyecto se encuentra definido es necesario buscar los tipos de dispositivos necesarios para la realización de este y realizar un análisis del mercado actual para encontrar los dispositivos adecuados.

3) Estudiar e implantar el protocolo de comunicación más apropiado:

   Tras escoger los dispositivos adecuados es necesario estudiar e implementar el método de comunicación entre ellos que se va a desarrollar.

4) Desarrollar la plataforma software para el sistema:

   También es necesario desarrollar el software necesario para el funcionamiento de los dispositivos previamente seleccionados y el software encargado del correcto funcionamiento del proyecto definido.

5) Posibilidad de plantear un modelo analítico en función de los datos recopilados:

   Una vez el proyecto se encuentre finalizado existe la posibilidad de realizar un estudio analítico de los datos recopilados con el objetivo de estudiar la posibilidad de introducir en el mercado actual el producto desarrollado.

3.2. Motivaciones

En esta sección se va a explicar las motivaciones que han llevado al alumno a elegir el tema de IoT Asistencial como su proyecto de fin de grado. Se pueden observar tres motivaciones:

1. El tema para realizar es IoT asistencial, eso significa que parte del proyecto va a estar relacionado con el manejo de hardware, lo que me parece muy interesante además de tratarse de un aspecto de la informática que no ha sido un tema principal en el grado, por lo que encuentro el trabajo de fin de grado como una oportunidad para aprender aspectos nuevos relacionados con el hardware y todo lo relacionado con IoT.

2. El proyecto se realiza en colaboración con Inycom, empresa que proporciona servicios y soluciones tecnológicas, lo que me permite tener un primer acercamiento al mundo laboral de la informática. El hecho de realizar el trabajo de fin de grado de la carrera en colaboración con una empresa me permite empezar a conocer el mercado laboral al que entraré en un futuro cercano, además también me permite conocer cómo se realiza un proyecto en una empresa, es decir, durante el grado he aprendido a como realizar proyectos en equipos para prepárame para futuros proyectos que pueda realizar en una empresa, pero con el trabajo de fin de grado puedo echar un pequeño vistazo al funcionamiento de un trabajo en equipo fuera de la parte teórica que he aprendido durante el grado.

3. Y la última motivación surge una vez se ha definido totalmente el proyecto a realizar, la apertura de una cerradura mediante identidad digital define un nuevo uso a una aplicación que únicamente se ha utilizado en soluciones informáticas de software. Para el aspecto de identidad digital del proyecto se va a utilizar una aplicación desarrollada por Inycom llamada Wallet Id, el proyecto crea la oportunidad de la apertura de un nuevo mercado a la aplicación Wallet Id, es decir, Wallet Id no ha sido utilizada para la realización de proyecto con una gran parte de ellos relacionada con el hardware por lo que el proyecto sería el primero y puede ser la puerta a nuevos proyectos relacionados con identidad digital.

4. Metodología

4.1. SCRUM

La metodología utilizada en el proyecto es SCRUM, para ser más exactos es una modificación de la metodología SCRUM con el objetivo de adaptarla al tipo de proyecto que se realiza. Se puede obtener más información sobre la utilización de Scrum en su página web incluida en la bibliografía del proyecto [7].

Como se comenta en el párrafo anterior en este proyecto se aplica la metodología SCRUM pero con alguna modificación, la primera modificación sería la eliminación de los Daily SCRUM, en SCRUM estas reuniones se realizan con el objetivo de observar el progreso del proyecto y la realización de un pequeño esquema o guía de las tareas que se van a realizar en el día actual y que integrantes del equipo de desarrollo se va a encargar de cada aspecto de la progresión del proyecto en el día actual. Esta herramienta de SCRUM es innecesaria en este tipo de proyecto debido a que no existe un equipo de desarrollo, una única persona se encarga de realizar el desarrollo del proyecto, y además la envergadura del proyecto es menor que el tamaño del tipo de proyecto que se realizan mediante un equipo de desarrollo utilizando SCRUM. La eliminación de los Daily SCRUM no significa que no exista comunicación entre todos los integrantes del proyecto, se realizan reuniones semanales para observar el progreso del proyecto y para la resolución de cualquier problema o error que surja en el desarrollo, además de la existencia de una constante comunicación mediante el correo electrónico.

Otra de las modificaciones de SCRUM está relacionada con el hecho de que una única persona se encarga de la realización del proyecto, esta modificación consiste en la eliminación de roles, más exactamente es la absorción de varios roles por una misma persona, en SCRUM existen tres roles principales, el Product Owner, el SCRUM Master y el equipo de desarrollo, el rol de Product Owner es compartido por los directores del proyecto y por el alumno que realiza el proyecto. En cuanto a los otros dos roles son asumidos por el encargado de la realización del proyecto, el alumno adopta el rol de SCRUM Master, es el encargado de dirigir los Sprints y el trabajo realizado en ellos, además del rol del equipo de desarrollo, es la persona que realiza el completo desarrollo del proyecto. Antes de finalizar esta sección es necesario destacar un pequeño matiz, el alumno es la persona que adopta el rol de SCRUM Master pero a pesar de eso cuenta con el apoyo de los directores de proyecto para realizar esa tarea, los directores tienen más experiencia en desarrollo de proyectos por lo que su guía en ese aspecto del proyecto es muy útil.

4.2. Sprints

Cada uno de los sprints del proyecto está dividido en tres fases, la primera fase es en la que se decide que se va a realizar en el sprint, se identifica los problemas que se van a enfrentar/solucionar y cómo afrontar estos problemas. Esta fase consiste en una reunión entre el alumno y los directores del proyecto donde se solucionan todas las incógnitas sobre el sprint a realizar. La siguiente fase es la fase de desarrollo, en esta fase se realiza las tareas designadas para este sprint y de la forma que se han acordado en la primera reunión del sprint, además a lo largo de esta fase se realizan varias reuniones para realizar un seguimiento del desarrollo y solucionar posibles problemas que puedan surgir.

La fase final es la fase de testeo, en esta fase se comprueba que la solución desarrollada funciona de forma adecuada y correctamente, una vez superada esta fase se puede finalizar el sprint y comenzar el siguiente.

A continuación, se muestran los diferentes Sprints que forman el proyecto, el tamaño de los Sprints puede variar debido a diferentes factores como retrasos en algunas tareas o adelantos debidos a algún error en la planificación inicial.

4.2.1. Sprint 1

En esta primera fase del proyecto se realiza una investigación en el mercado actual buscando soluciones del mismo ámbito que el proyecto a realizar y se termina de desarrollar el proyecto logrando un factor que diferencie el proyecto de los que se encuentran en el mercado actual. Una vez se ha definido el proyecto se realiza un análisis y diseño inicial donde se crea la base de la arquitectura del proyecto. Para finalizar se realiza una fase de aprendizaje sobre las tecnologías que van a ser utilizadas en el proyecto.

▾ Sprint1	100%		Start	Due
Documentación	100%		Feb 3, 2020	Feb 23, 2020
Realización tutoriales Raspberry Pi	100%		Feb 5, 2020	Feb 17, 2020
Análisis Inicial	100%		Feb 15, 2020	Feb 19, 2020
Diseño Inicial	100%		Feb 20, 2020	Feb 23, 2020

Figura 2: Sprint 1

4.2.2. Sprint 2

En esta fase del proyecto comienza con la continuación del estudio de las tecnologías a utilizar en el proyecto, una vez finalizado ese estudio se realiza una toma de decisiones sobre las tecnologías que pueden resultar más eficaces para el proyecto. Una vez seleccionadas las tecnologías a utilizar se realiza la implementación de la parte mecánica del proyecto y la programación encargada de su control. Para finalizar es necesario testear la solución implementada y documentar el proceso.

Sprint2	100%		Start	Due
Estudio Tecnologías y Hardware	100%		Feb 24, 2020	Feb 27, 2020
Toma de decisiones(Elección basada en el estudio previo)	100%		Feb 28, 2020	Mar 1, 2020
Implementación Programa Encargado Parte Mecánica	100%		Mar 2, 2020	Mar 12, 2020
Testing Python Aplication	100%		Mar 13, 2020	Mar 15, 2020
Documentación	100%		Feb 24, 2020	Mar 15, 2020

Figura 3: Sprint 2

4.2.3. Sprint 3

Una vez se ha implementado la parte mecánica del proyecto el siguiente paso es realizar un estudio del sistema de identidad digital que se va a utilizar, este sprint consiste únicamente en estudio e investigación por lo que su tamaño es menor que los sprint anteriores.

Sprint3	100%		Start	Due
Estudio sistema de Identidad Digital actual(WalletID)	100%		Mar 16, 2020	Mar 22, 2020
Análisis	100%		Mar 23, 2020	Mar 26, 2020
Diseño	100%		Mar 27, 2020	Mar 29, 2020
Documentación	100%		Mar 16, 2020	Mar 29, 2020

Figura 4: Sprint 3

4.2.4. Sprint 4

Además de la tecnología de identidad digital es necesario implementar y configurar una base de datos donde se almacene toda la información necesaria además de una aplicación que permita la comunicación entre la base de datos y la parte mecánica.

Para finalizar el sprint se ha implementado una aplicación web para facilitar la comunicación entre el usuario y la base de datos.

Sprint4	100%		Start	Due
Implementación/Extensión/Modificación API	100%		March 30, 2020	April 19, 2020
Implementación Aplicación web	100%		April 10, 2020	April 19, 2020
Testing Identidad Digital Aplication	100%		April 20, 2020	April 26, 2020
Documentación	100%		March 30, 2020	April 26, 2020

Figura 5: Sprint 4

4.2.5. Sprint 5

Esta fase del proyecto comienza con el análisis de la integración de todos los módulos implementados, la parte mecánica, la identidad digital y la aplicación desarrollada, y posteriormente un rediseño de la arquitectura del sistema en caso de que sea necesario para facilitar la integración. Una vez es posible la integración es realizada y posteriormente testeada.

Sprint5	0%		Start	Due
Análisis de la integración	0%		Apr 27, 2020	Apr 30, 2020
Rediseño	0%		May 1, 2020	May 4, 2020
Integración de los dos sistemas	0%		May 5, 2020	May 19, 2020
Testing Integración de los sistemas	0%		May 20, 2020	Jun 24, 2020
Documentación	0%		Apr 27, 2020	May 24, 2020

Figura 6: Sprint 5

4.2.6. Sprint 6

Para finalizar se implementa y se testea una demostración donde se pueda observar todos los aspectos del proyecto, para acompañar la demostración es necesario la creación de material visual que facilite la demostración.

Sprint6	0%		Start	Due
Creación demo	0%		May 25, 2020	Jun 4, 2020
Testing demo	0%		Jun 5, 2020	Jun 11, 2020
Presentación(PowerPoint)	0%		Jun 12, 2020	Jun 14, 2020
Documentación	0%		May 25, 2020	Jun 14, 2020

Figura 7: Sprint 6

Además de lo comentado en las secciones anteriores sobre las fases del proyecto cada fase está acompañada por una detallada documentación de todo el proceso que se ha realizado.

4.3. Software

Para la realización del proyecto se han utilizado diferente software para la realización de los diferentes aspectos del proyecto como planificación de tareas, desarrollo de la aplicación, seguimiento del proyecto… las herramientas utilizadas son las siguientes:

4.3.1. Trello

Trello [8] es una herramienta de gestión de proyectos con interfaz web y aplicación móvil. Este software se ha utilizado para la planificación y organización de las diferentes tareas a realizar en el proyecto y para mantener un flujo de trabajo adecuado para la realización del proyecto de manera eficiente.

Otro de los motivos de la utilización de Trello es que proporciona una interfaz que facilita el método SCRUM y también facilita el seguimiento del estado de la implementación a todas las personas que forman parte del proyecto

4.3.2. Bitbucket

Bitbucket [9] es un servicio de alojamiento y colaboración basado en Git, Bitbucket ofrece un sistema de control de versiones de proyectos lo que facilita el desarrollo de cualquier producto de software.

Se ha utilizado Bitbucket para tener la posibilidad de tener un control de versiones de los diferentes programas que se han implementado en el proyecto.

4.3.3. Visual Studio Code

Visual Studio Code [10] es un editor de código desarrollado por Microsoft para Windows, Linux y macOs. Visual Studio Code ofrece soporte para diferentes lenguajes de programación además existen una gran cantidad de extensiones creadas para este editor lo que lo hace muy útil a la hora de realizar un proyecto en el cual la programación tiene una gran importancia.

Visual Studio Code es el entorno de desarrollo utilizado para la implementación de todo el código del proyecto a excepción del código implementado en la unidad de control de la parte mecánica del proyecto.

4.3.4. Thonny Python IDE

Thonny [11] es un entorno de desarrollo integrado para Python, es el entorno de desarrollo que se ha utilizado para implementar el código que controla la parte mecánica del proyecto.

4.3.5. Balsamiq Cloud

Balsamiq Cloud [12] es un software con interfaz web que proporciona herramientas para el diseño de interfaces para aplicaciones y páginas web

Balsamiq Cloud se ha utilizado para el diseño de todas interfaces que se han implementado en el proyecto.

5. Implementación

En esta sección de la memoria del proyecto se va a describir la implementación del sistema que se quiere realizar junto al estudio e investigación de todos los factores que sean necesarios analizar para el desarrollo del proyecto.

5.1. Diseño Inicial

Primero es necesario realizar un pequeño análisis del objetivo que se quiere lograr para poder realizar el diseño inicial de la arquitectura del sistema que se va a implementar. Este es un diseño inicial realizado con una primera vista del proyecto a implementar lo que implica que este diseño está abierto a cualquier cambio que sea necesario para la mejora del proyecto o la posible solución de errores que surjan a lo largo del desarrollo. El sistema se puede dividir en tres secciones principales, una unidad de control encargada de gestionar la cerradura, una aplicación móvil la cual será utilizada por el usuario para controlar la cerradura y un servidor el cual es capaz de invocar a la API del sistema de identidad digital.

El funcionamiento de la arquitectura del sistema se puede observar en la Figura 8 y es el siguiente, el usuario se identifica utilizando la aplicación móvil (1), la aplicación realiza una llamada al servidor para que invoque el servicio de identidad digital y compruebe la identidad del usuario (2), una vez comprobado, si la identidad es correcta el servidor se comunica con la unidad de control (3) la cual se comunica con la cerradura y la abre (4).

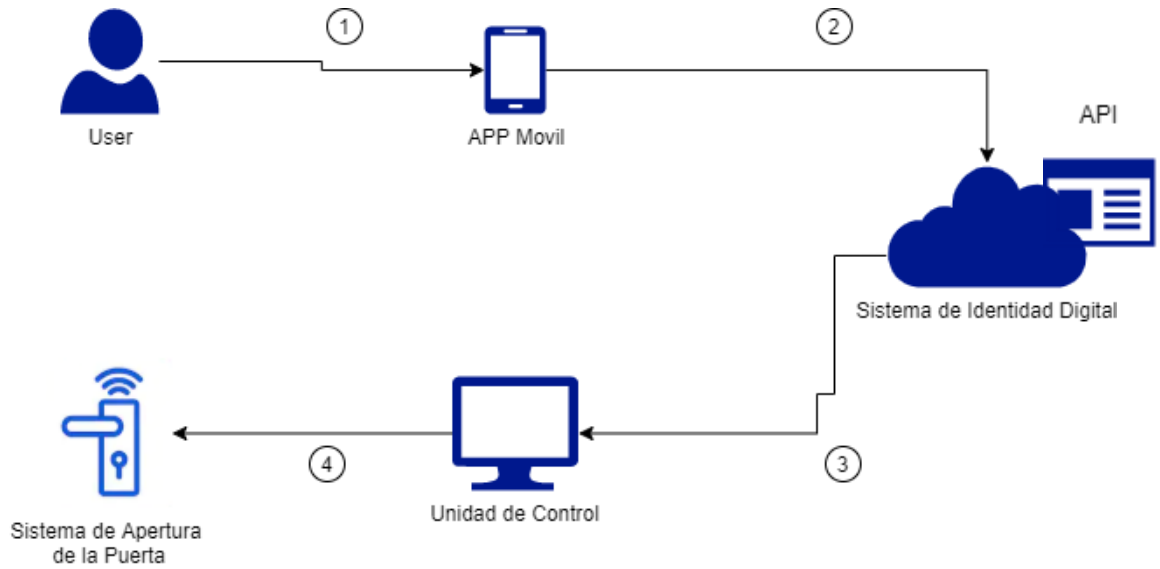


Figura 8: Diseño Inicial

5.2. Análisis Inicial

Una vez se ha diseñado la arquitectura básica necesaria para el proyecto es necesario realizar un análisis de cada sección que forma la arquitectura del sistema. Este análisis se realiza para conocer la forma de construir la arquitectura de cada sección y las herramientas y materiales que resulten más adecuados para el proyecto. Este análisis es una primera visión del proyecto por lo que en algunas secciones puede diferir bastante del estado final de la arquitectura del sistema.

Como comienzo del análisis se ha realizado una búsqueda de diferentes cerraduras en el mercado actual, la cerradura es el pilar del proyecto por lo que es necesario encontrar la adecuada. Con un primer sondeo al mercado se ha llegado a la conclusión de que existen dos posibles caminos diferentes que seguir a la hora de realizar el proyecto, el primero, utilizar una cerradura eléctrica básica la cual será controlada mediante señales enviadas desde la unidad de control a la cerradura, y el segundo camino, utilizar una cerradura eléctrica más avanzada que tenga su propio kit de desarrollo el cual permita crear un programa en la unidad de control que sea capaz de realizar operaciones básicas como la apertura y el cierre de la cerradura. Con los dos posibles caminos a seguir en mente se ha realizado una búsqueda más exhaustiva en el mercado y como resultado de la búsqueda se ha llegado a la conclusión de que la mejor opción es utilizar un cierre eléctrico básico [13] , como el que se puede observar en la Figura 9, por tres motivos principales, el montaje y el manejo de un cierre eléctrico básico es mucho más simple que las otras opciones, las cerraduras eléctricas con un kit de desarrollo propio encontradas son bastante más complejas, en ocasiones la documentación proporcionada es confusa y muy escasa, y el precio es mucho mayor que un cierre eléctrico básico, y para finalizar un cierre eléctrico básico se adapta mejor al tipo de proyecto que se realiza y si es necesario se puede realizar una actualización una vez el proyecto se encuentre avanzado.

Figura 9: Cierre Eléctrico Utilizado

Una vez se ha encontrado la cerradura adecuada para el proyecto es necesario encontrar la unidad de control encargada de controlar la cerradura, para ello la mejor opción es utilizar una placa de desarrollo que mediante la ejecución de un programa controle la apertura y el cierre de la cerradura. Las placas de desarrollo que son más utilizadas en la actualidad en un gran abanico de proyectos entre ellos proyectos y prototipos de IoT son Arduino y Raspberry Pi [14], además de estas dos opciones en el mercado actual existen muchas más opciones en cuanto a placas de desarrollo más complejas y con mayores prestaciones pero se han descartado principalmente por las siguientes razones, Arduino y Raspberry Pi ofrecen las herramientas necesarias para un proyecto de estas características, y además Arduino y Raspberry Pi siempre han estado presentes en muchos proyectos de IoT y en el ámbito educacional a lo largo de los años por lo que son consideradas como las perfectas opciones para un proyecto de estas características.

Una vez seleccionadas las dos posibles opciones para ocupar el puesto de unidad de control en la arquitectura del proyecto es necesario averiguar cual se adapta mejor a las necesidades actuales y futuras del proyecto.

La placa de desarrollo Arduino es un microcontrolador, es decir, Arduino no ejecuta un sistema operativo completo como otras placas de desarrollo, sino que simplemente ejecuta código tal como lo interpreta su firmware. Arduino solo puede ejecutar un programa de manera simultánea por lo que es ideal para proyectos donde interactúa con hardware como sensores y otros dispositivos.

Por otra parte, Raspberry Pi es un ordenador de placa reducida, a diferencia de Arduino, Raspberry Pi es como un mini ordenador lo que implica que además de poder ejecutar varios programas simultáneamente ofrece diferentes herramientas y la posibilidad de instalar librerías y software para interactuar con otros dispositivos y componentes. Raspberry Pi ofrece la posibilidad de crear un gran abanico de tipos de proyectos, desde servidores de archivos, interactuar con sensores, estaciones multimedia…, y otros muchos más los cuales se pueden conocer a través de los millones de proyectos que existen en la red utilizando Raspberry Pi como placa de desarrollo. En la Tabla 2 se puede observar una comparativa entre Raspberry Pi y Arduino.

	Arduino	*Raspberry Pi*
Definición	***Ordenador de placa reducida***	***Micro controlador***
Potencia de ejecución	***Puede ejecutar un único programa simultáneamente***	***Puede ejecutar múltiples programas simultáneamente***
Alimentación	***Pensado para funcionar con batería***	***Pensado para conectar a la corriente***
Lenguaje	***Arduino y C/C++***	***Una gran variedad disponible (Python, C, Java…)***
Precio	***Es barato***	***Más caro que Arduino***

Tabla 2: Comparativa entre Arduino y Raspberry Pi

Tras un análisis del funcionamiento de ambos dispositivos y sus características se ha llegado a la conclusión que la placa de desarrollo adecuada para la realización del proyecto es una Raspberry Pi [15], a pesar de que Arduino es la placa predilecta a la hora de interactuar con hardware Raspberry Pi no se queda atrás en ese ámbito además Raspberry Pi ofrece diversas herramientas que pueden facilitar el desarrollo, y como razón principal a la hora de descartar Arduino es la clara superioridad de Raspberry Pi en cuanto a potencia de procesamiento algo que puede resultar necesario dependiendo cómo evolucione el proyecto.

Actualmente en el mercado existen varios modelos de Raspberry Pi, siendo la Raspberry Pi 4 Model B el último modelo, para este proyecto se ha decidido trabajar con una Raspberry Pi 3 Model B la cual cumple con todos los requisitos requeridos para trabajar como unidad de control del proyecto.

Una vez se ha elegido el dispositivo que se va a utilizar como unidad de control el siguiente paso es elegir el lenguaje a utilizar. En Raspberry Pi existen varios lenguajes que pueden ser utilizados para programar, por ejemplo, Python, C, Java…, a la hora de elegir el lenguaje a utilizar se ha buscado uno que no fuese necesaria una gran carga de aprendizaje y que proporcionase las herramientas necesarias. No se quiere elegir un programa que no sea conocido por el desarrollador del proyecto porque se considera que ya existe bastante aprendizaje necesario el hecho de utilizar una Raspberry Pi por primera vez y utilizar un lenguaje conocido podría ayudar y hacer menos la carga de aprendizaje. Otro de los aspectos importantes es el hecho de que el proyecto necesita herramientas de diferentes aspectos tecnológicos, es decir, necesita controlar señales eléctricas, controlar cámaras e imágenes entre otras cosas, por eso es necesario un lenguaje para el cual existan una gran cantidad y variedad de librerías.

Tras estudiar las opciones que existen se ha elegido Python por las siguientes razones:

- Cumple con los requisitos que se han nombrado, es un programa conocido por el desarrollador, y Python es un lenguaje que posee una gran cantidad y variedad de librerías.
- Otra de las razones es la compatibilidad de Python con Raspberry Pi, Python es el programa por excelencia si se quiere programar en una Raspberry Pi, lo que facilita el desarrollo por la gran cantidad de proyectos que existen en la red.
- Y, por último, Python es un lenguaje que se encuentra muy activo. Python en la actualidad sigue siendo uno de los lenguajes más utilizados gracias a nuevas tecnologías como Machine Learning y Data Science, dos de las tecnologías que más importancia están teniendo en la actualidad, lo que implica que la comunidad de Python está muy activa y eso significa ejemplos de proyectos más actuales y librerías que siguen evolucionando y ofreciendo nuevas herramientas.

Otra parte importante en la arquitectura del sistema es el proceso de identificación que realiza el usuario para poder acceder a la cerradura, los componentes necesarios para realizar esa acción no se pueden definir todavía debido a que dependen del funcionamiento de Wallet Id, el sistema de identidad digital, por lo que a continuación se va a realizar un análisis del funcionamiento y las características que posee Wallet Id y en secciones posteriores creando un diseño más complejo de la arquitectura del proyecto se analizará la necesidad de desarrollar una API, una App móvil o una App web.

Y como último aspecto a analizar en este análisis inicial esta Wallet Id.

Wallet Id es una aplicación de identidad digital la cual se encarga de almacenar de forma segura la identidad o perfil digital para luego mediante su uso facilitar otras operaciones o acciones.

Wallet Id genera claves criptográficas únicas en el móvil del usuario lo que implica que cualquier transacción se realice de forma segura. Wallet Id funciona de la siguiente forma, primero se realiza una identificación en un entorno web o móvil, luego se pueden realizar varias operaciones, firma de documentos, autorización de operaciones

Wallet Id ofrece la posibilidad de realizar diversas acciones, firmas de documentos, autorización de operaciones e identificación digital, de forma segura mediante el uso de su aplicación la cual proporciona un sistema de autentificación que es necesario a la hora de realizar cualquier acción.

También ofrece dos aplicaciones móviles, Wallet Id y Wallet Id Tarjeta Ciudadana Virtual (TCV) [16], en este proyecto se va a utilizar la segunda, la cual ofrece una forma de identificarse mediante un QR dinámico que solo tiene validez durante varios segundos, esa forma de identificarse mediante un QR va a ser la que se va a utilizar a la hora de identificar a un usuario en el sistema que se va a implementar en este proyecto.

5.3. Implementación de la parte Mecánica

En esta sección de la memoria se va a realizar el estudio y la investigación necesaria para la implementación de la parte mecánica inicial del proyecto, además de la creación del circuito conectado al dispositivo y finalmente la implementación del programa encargado del control de la cerradura.

5.3.1. Estudio básico funcionamiento Raspberry Pi

Para comenzar con la parte mecánica del proyecto es necesario estudiar el funcionamiento del dispositivo que se va a utilizar, en este caso una Raspberry Pi 3, como se comenta en secciones anteriores una Raspberry Pi es un ordenador de placa reducida que permite la realización de proyectos a un bajo coste. A continuación, en la Figura 10 se puede observar el aspecto de una Raspberry Pi 3.

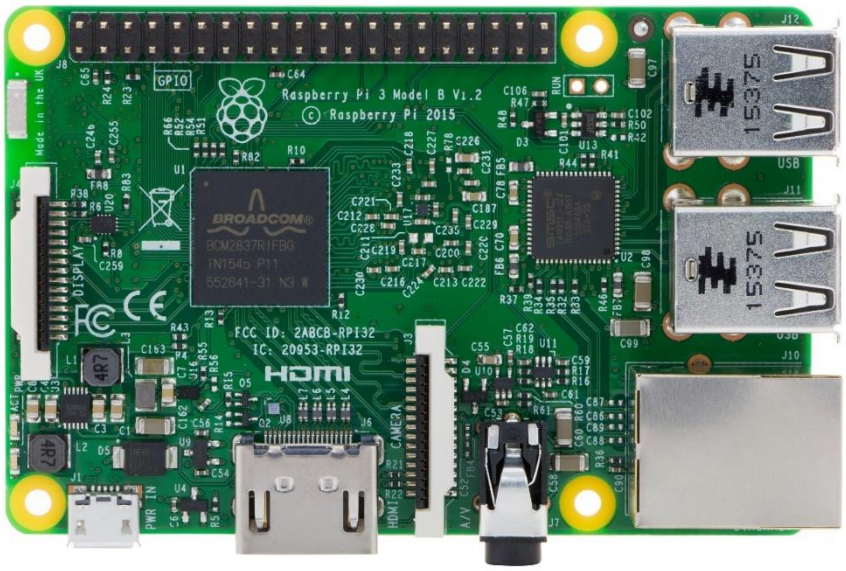


Figura 10: Raspberry Pi 3 Model B

Para la realización de este proyecto es necesario el estudio de los pines que ofrece el dispositivo para un correcto uso más adelante. Raspberry Pi ofrece un sistema de entrada y salida llamado General Purpose Input Output (GPIO), este sistema consiste en una serie de pines o conexiones que se pueden utilizar como entradas o salidas para la creación de diversos sistemas. Los pines GPIO tienen diferentes funciones y se pueden agrupar en ocho tipos diferentes.

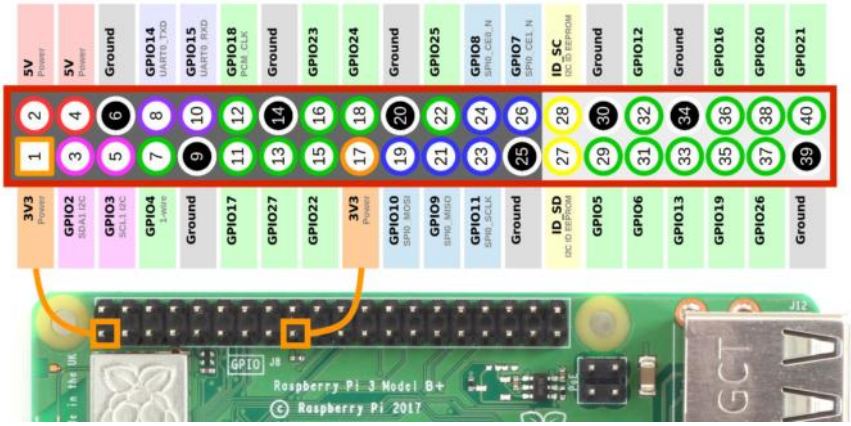


Figura 11: Esquema Pines Raspberry Pi 3

En la Figura 11 [17] se puede observar los diferentes pines que se encuentran en el dispositivo utilizado, estos se pueden agrupar de la siguiente manera utilizando los colores de la imagen:

- Naranja (2 pines): Alimentación a 3.3V
- Rojo (2 pines): Alimentación a 5V
- Gris (8 pines): Conexión a GND
- Verde + Azul + Rosa + Morado (26 pines): Entradas/Salidas de propósito general, y conexiones a dispositivos mediante diferentes protocolos.
- Amarillo (2 pines): Pines reservados

Una vez se conoce el funcionamiento de los diferentes pines proporcionados por Raspberry Pi es posible comenzar a implementar el circuito para controlar la cerradura mediante un programa ejecutado en la Raspberry Pi utilizando las diferentes entradas y salidas del sistema GPIO.

5.3.2. Implementación circuito y conexiones a Raspberry Pi

En esta sección del proyecto se va a mostrar el diseño y la implementación del circuito el cual conecta la cerradura con la Raspberry Pi permitiendo el control completo de la cerradura mediante un programa ejecutado en la Raspberry Pi. Primero se realiza el diseño para conocer los elementos necesarios para un adecuado funcionamiento del circuito.

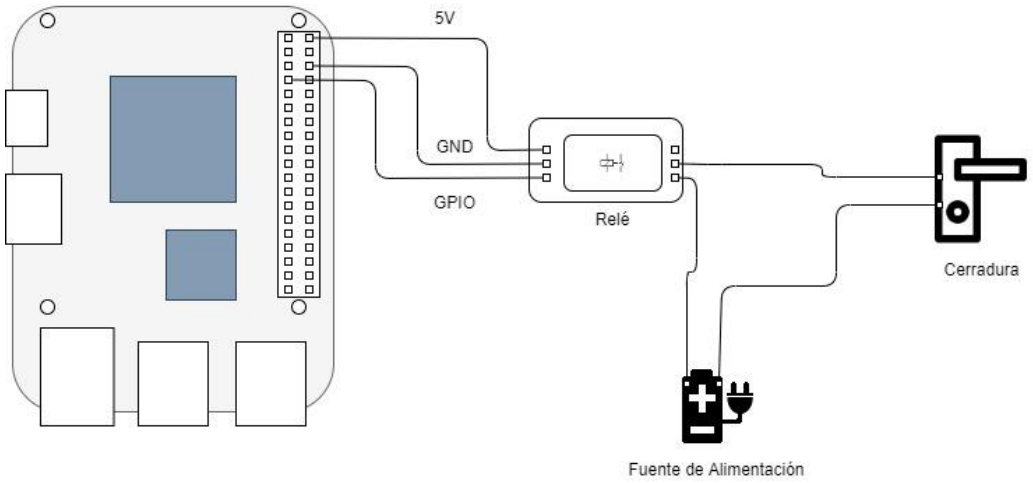


Figura 12: Diseño Circuito Control de Cerradura

Para la cerradura se ha elegido una opción básica, un cierre eléctrico que funciona con 12 voltios, en próximas etapas del proyecto si es necesario una cerradura más compleja se realizará el cambio, pero para esta fase inicial una opción básica como la elegida cumple todos los requisitos necesarios.

Una vez elegida la cerradura es necesario estudiar cómo controlarla mediante nuestra unidad de control, la Raspberry Pi solo proporciona 3 o 5 voltios por lo que es necesario el uso de una fuente de alimentación externa para accionar el cierre eléctrico. Además de la fuente de alimentación se necesita un relé para controlar los 12 voltios que proporciona la fuente de alimentación y accionar el cierre cuando es necesario. El relé funciona como un interruptor, cuando el relé es accionado cierra el circuito permitiendo el paso de los 12 voltios y encendiendo el cierre eléctrico.

La Raspberry Pi es la encargada de controlar el relé, para ello se utilizan los pines:

- Pin 2: 5 Voltios
- Pin 6: Conexión a tierra
- Pin 7: GPIO4, señal de control

La unidad de control envía una señal mediante el pin 7 al relé cuando se quiere abrir la cerradura, la señal enviada por este pin se controla mediante un programa en desarrollado en Python que se explica en la siguiente sección. Una vez diseñado el circuito se realiza la implementación de este utilizando los componentes más adecuados, conocidos gracias al diseño realizado.

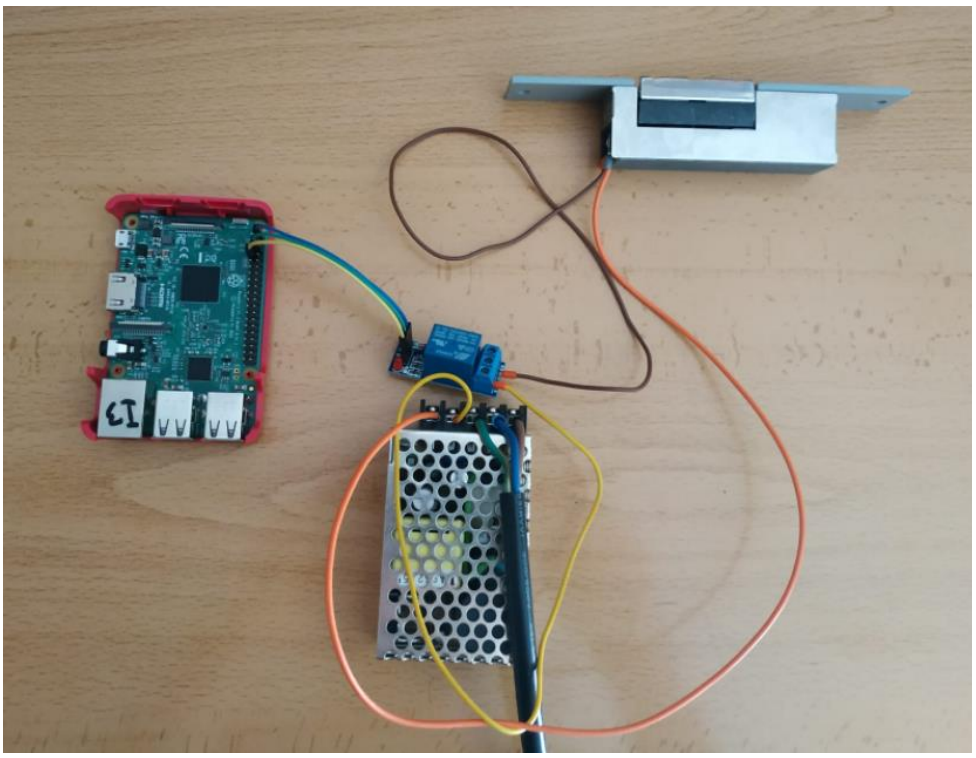


Figura 13: Circuito Control de Cerradura

5.3.3. Implementación programa Python

Una vez diseñado y montado el circuito es necesario la implementación del programa que se ejecute en la unidad de control para controlar la apertura y el cierre de la cerradura. Como se comenta en el análisis inicial el programa está desarrollado en Python, y para el control de los pines se utiliza RPI.GPIO [18], librería que proporciona un total control en los pines GPIO de una Raspberry Pi.

El programa implementado es muy básico, pero a su vez cumple el objetivo del aprendizaje necesario para el control de los diferentes pines que existen en la Raspberry Pi utilizada, y el objetivo de esta primera fase del proyecto, el control de la cerradura mediante la unidad de control. Posteriormente el proyecto evolucionará con la incorporación de más componentes al circuito y el aumento de la complejidad de las operaciones requeridas para lograr el objetivo final, simultáneamente a la evolución de proyecto el código ejecutado en la unidad de control también aumentará su complejidad. A continuación, en la Figura 14, se muestra la implementación del programa requerido en esta primera fase del proyecto.

```python
import RPi.GPIO as gpio
import time

gpio.setwarnings(False)

while True:
    input("Press key to open")
    #GPIO numbering using pines numbers
    gpio.setmode(gpio.BOARD)
    #Set pin 7 as output
    gpio.setup(7, gpio.OUT)
    #Set pin 7 value to high/1
    gpio.output(7, gpio.HIGH)
    time.sleep(2)

    input("Press key to close")
    #Set pin 7 value to high/0
    gpio.output(7, gpio.LOW)


    key = input("Press e to exit")
    if(key == "e"):
        gpio.cleanup()
        print("Exit")
        break
```

Figura 14: Programa de Control de la Cerradura

En el código que se muestra en la imagen previa se puede visualizar los diferentes pasos que son necesarios implementar.

- Primero es necesario la configuración de los pines mediante la librería RPIO.GPIO, se configura el modo de referirse a los pines, en este caso mediante la numeración de los pines de la Raspberry Pi, y el modo del pin a utilizar, en este caso se utiliza el pin 7 y su modo es output.
- Una vez se ha realizado la configuración se puede cambiar el valor de la salida del pin a valor HIGH o valor LOW. Si el valor del pin es HIGH se envía una señal al relé que se encuentra conectado al pin 7 lo que provoca que el cierre eléctrico se abra, si el valor es LOW se envía otra señal al relé lo que provoca que el cierre eléctrico no reciba el voltaje necesario y provoca el cierre de la cerradura.
- Para finalizar la implementación, si el usuario decide salir del programa se realiza un reinicio de la configuración realizada en los pines, utilizando la función *cleanup()* que proporciona la libraría RPi.GPIO, para un correcto funcionamiento en el siguiente uso de los pines de la unidad de control.

5.4. Análisis de la unión de la parte Mecánica con la tecnología de Identidad Digital

Una vez se ha implementado el programa con el cual se puede controlar la cerradura desde la unidad de control, la Raspberry PI, el siguiente paso es realizar la implementación de la API la cual con el apoyo de Wallet ID, aplicación que implementa el uso de identidad digital, proporcionará un sistema de autentificación el cual autorizará el acceso a cualquier persona que interaccione con la cerradura y tenga acceso autorizado.

Antes de implementar la aplicación es necesario analizar la unión del sistema ya implementado, el control de la cerradura desde la Raspberry Pi, y el nuevo sistema a implementar. Para ello se va a realizar un análisis mediante el diseño de varios casos de uso con el objetivo de encontrar el sistema óptimo.

5.4.1. Casos de Uso con QR en la cerradura

Esta opción como unión de la parte mecánica con la aplicación de identidad digital se basa en la lectura de un QR impreso en la cerradura con una aplicación desarrollada la cual se encarga de realizar todas las comunicaciones y finalmente informar a la unidad de control de si se autoriza el acceso.

5.4.1.1. Caso de Uso 1.1

En este primer caso se realiza la unión de la parte mecánica del proyecto únicamente con la API encargada de autentificar a los usuarios, en este primer caso de uso se ha dejado fuera la identidad digital para poder visualizar mejor la unión para que en los próximos casos se pueda introducir la unión con la aplicación de identidad digital eficazmente.

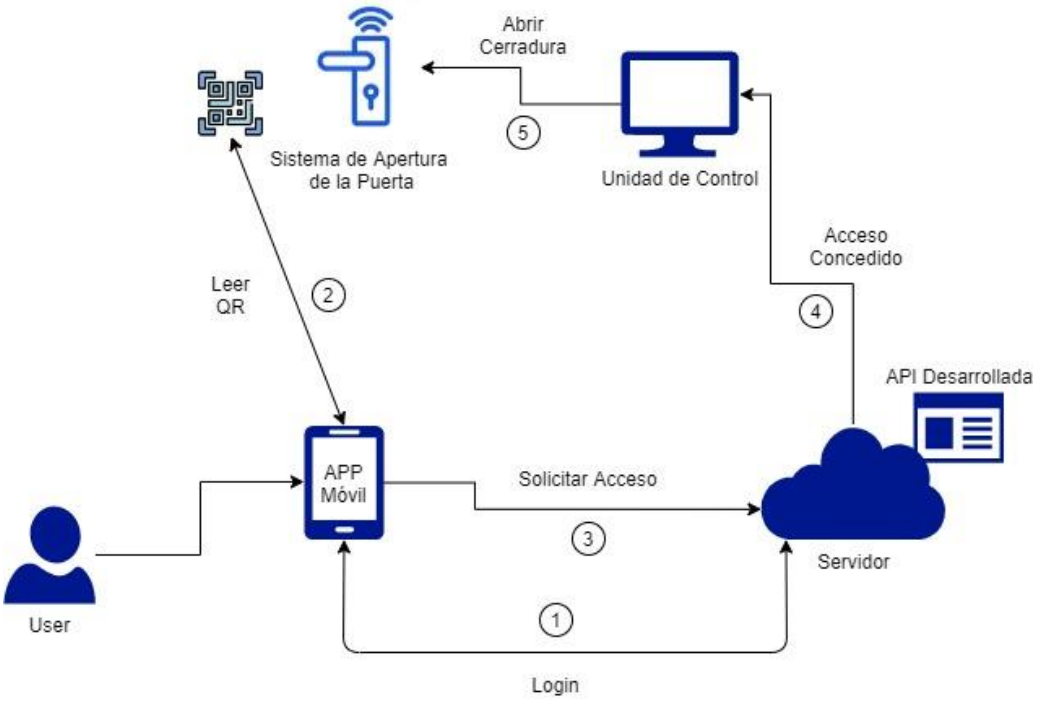


Figura 15: Caso de Uso 1.1

Escenario: Se necesita una aplicación móvil o web donde el usuario pueda identificarse para poder identificar al usuario que lee el QR, una vez leído se necesita una API alojada en un servidor la cual albergue una base de datos de los usuarios, sería necesario la implementación del CRUD de usuarios, de cerraduras y de permisos de accesos, además de un servicio de login, de autorización y de apertura.

Flujo:

1: El usuario se identifica en la aplicación, la aplicación lanza una llamada al servidor para comprobar su identificación y si es favorable le concede acceso a la aplicación.

2: El usuario utiliza el lector de la aplicación para leer el QR que se encuentra en la cerradura.

3: La aplicación se comunica con el servidor y solicita autorización de acceso a la cerradura.

4: El servidor comprueba los permisos del usuario con esa cerradura y si son correctos se comunica con la unidad de control, la Raspberry Pi, y le informa de que el acceso es correcto.

5: Finalmente la unidad de control se comunica con la cerradura y la abre.

5.4.1.1. Caso de Uso 1.2

Después de analizar la unión de la parte mecánica con la API desarrollada en el caso de uso anterior se puede introducir al sistema la aplicación de identidad digital. El sistema de identidad digital que se va a utilizar es una aplicación llamada Wallet ID que se encuentra en un servidor a la cual llamaremos para verificar la identidad de cada usuario que quiera abrir la cerradura.

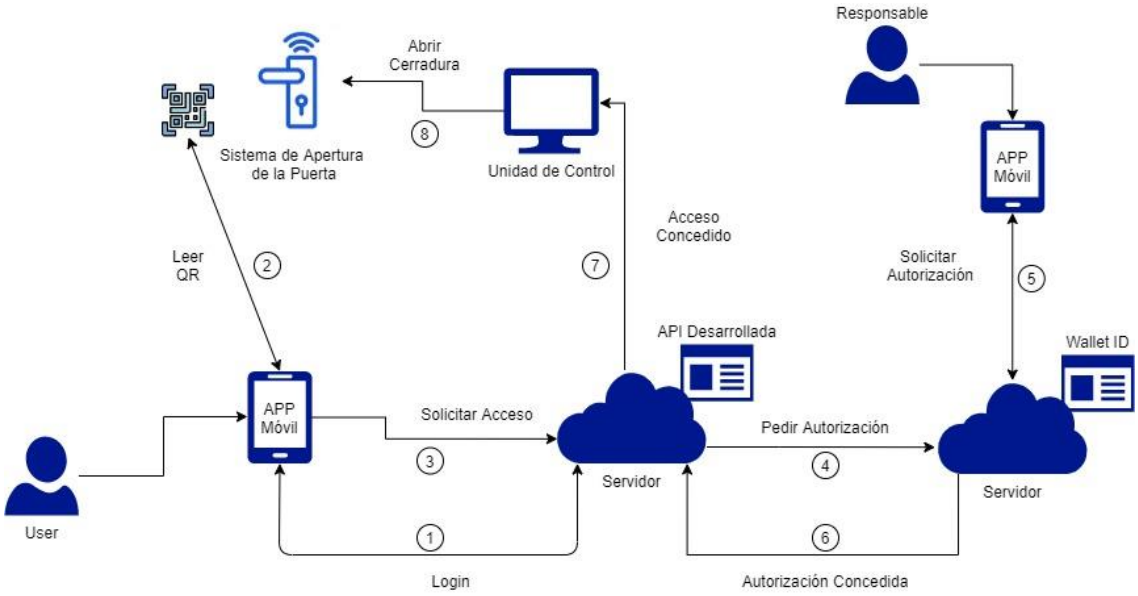


Figura 16: Caso de Uso 1.2

Escenario: Se necesita una aplicación móvil o web donde el usuario pueda identificarse para poder identificar al usuario que lee el QR, una vez leído se necesita una API alojada en un servidor la cual albergue una base de datos de los usuarios, sería necesario la implementación del CRUD de usuarios, de cerraduras y de permisos de accesos, además de un servicio de login, de autorización y de apertura.

Flujo:

1: El usuario se identifica en la aplicación, la aplicación lanza una llamada al servidor para comprobar su identificación y si es favorable le concede acceso a la aplicación.

2: El usuario utiliza el lector de la aplicación para leer el QR que se encuentra en la cerradura.

3: La aplicación se comunica con el servidor y solicita autorización de acceso a la cerradura.

4: El servidor manda una petición a Wallet ID para solicitar autorización al responsable.

5: Wallet ID manda una notificación mediante la aplicación móvil al responsable, en la mayoría de los casos el responsable y la persona que requiere acceso a la cerradura es la misma persona, pero existen casos en los que un usuario requiere acceso a una cerradura de la cual no es el responsable, esos usuarios son llamados invitados, son usuarios los cuales necesitan permiso del responsable de la cerradura para acceder.

6: Si el responsable autoriza el acceso Wallet ID comunica al servidor que se permite el acceso.

7: El servidor se comunica con la unidad de control, la Raspberry Pi, y le informa de que el acceso es correcto.

8: Finalmente la unidad de control se comunica con la cerradura y la abre.

5.4.2. Casos de Uso con lector de QR en la cerradura

Esta opción como unión de la parte mecánica con la aplicación de identidad digital se basa en la lectura de un QR generado en una aplicación móvil y leído por un lector que se encuentra en la cerradura. La aplicación móvil es una de las implementadas para utilizar la tecnología de Wallet ID, en este caso la aplicación móvil a utilizar es TCV Wallet ID la cual permite la identificación de un usuario con Tarjeta Ciudadana Virtual.

5.4.2.1. Caso de Uso 2

Este caso de uso se basa en la lectura de un QR y la posterior identificación del usuario mediante la aplicación de identidad digital.

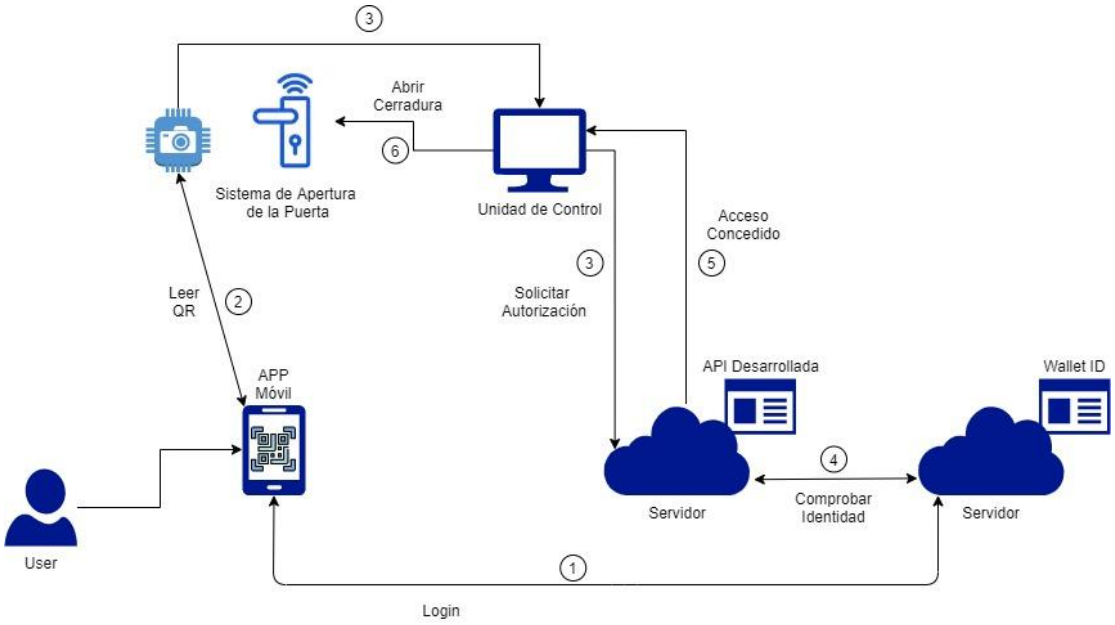


Figura 17: Caso de Uso 2

Escenario: Se utiliza la aplicación móvil Wallet ID TCV para generar el QR con el que identificar al usuario, además de eso se necesita una API alojada en un servidor la cual albergue una base de datos de los usuarios, sería necesario la implementación del CRUD de usuarios, de cerraduras y de permisos de accesos, además de un servicio de login, de autorización y de apertura. La función de la API es comprobar si el usuario identificado gracias a la tecnología de identidad digital tiene la autorización para acceder a la cerradura.

Flujo:

1: El usuario se identifica en la aplicación de Wallet ID TCV, la aplicación lanza una llamada al servidor para comprobar su identificación y si es favorable le concede acceso a la aplicación.

2: El usuario genera el código QR con la aplicación y el lector instalado en la cerradura lo escanea.

3: La unidad de control de la cerradura hace una llamada al servidor solicitando la autorización para desbloquear la cerradura.

4: El servidor se comunica con Wallet ID para identificar al usuario que intenta acceder.

5: El servidor comprueba si el usuario identificado previamente tiene autorización para acceder a la cerradura, el servidor comunica el resultado a la unidad de control.

6: Si el servidor le ha comunicado a la unidad de control que el usuario tiene autorización para entrar entonces la unidad de control desbloquea la cerradura.

5.4.3. Resultado análisis

Una vez se han definido los diferentes casos de uso es necesario realizar un análisis de cada uno de ellos para averiguar cuál es la arquitectura más adecuada para implementar en el proyecto.

	APP Móvil Necesarias	*Operación con QR*	*API Necesarias*
Caso 1.1	***APP Móvil a desarrollar***	***QR impreso en la cerradura***	***API a desarrollar***
Caso 1.2	***APP Móvil a desarrollar + APP Wallet Id***	***QR impreso en la cerradura***	***API a desarrollar + API Wallet Id***
Caso 2	***APP Wallet Id TCV***	***Lector QR en la cerradura***	***API a desarrollar + API Wallet Id***

Tabla 3: Comparación entre los diferentes casos de uso

Después de realizar el análisis de los diferentes casos de uso se ha decidido que la opción que implica la instalación de un lector de QR en la cerradura, caso de uso 2, es la óptima para el proyecto debido a que comparando con la otra opción que también usa identidad digital, caso 1.2, propone una solución más sencilla y proporciona el mismo resultado, a pesar de eso, tras analizar el caso de uso se han realizado unos pequeños cambios para aumentar su eficacia.

Caso de Uso 2.1:

Este caso de uso es muy similar al caso de uso 2, la diferencia es que ya no es el servidor que aloja la API desarrollada el que se comunica con Wallet ID para comprobar la identificación del usuario, es la unidad de control la que lleva a cabo esta operación.

Esta variación se ha realizado para reducir la carga de trabajo en el servidor y para evitar llamadas innecesarias al servidor, como por ejemplo en el caso de que la identificación del usuario sea errónea no es necesario comunicarse con el servidor puesto que no se necesita comprobar ningún permiso, por lo que la unidad de control se puede encargar de finalizar la tarea y rechazar el acceso al usuario.

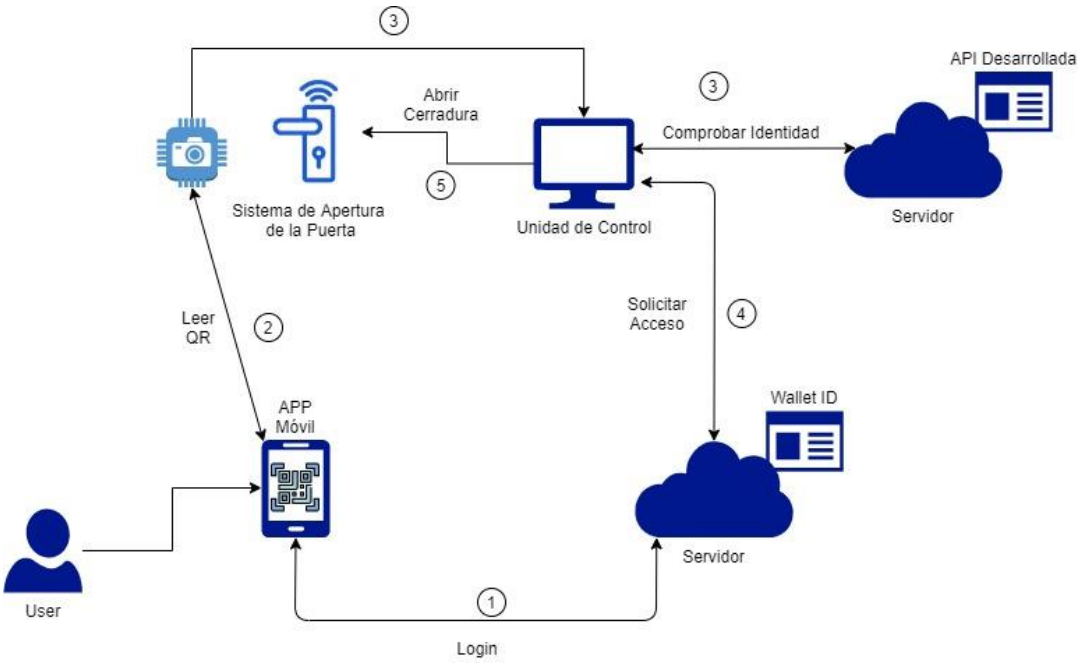


Figura 18: Caso de Uso 2.1

Escenario: El escenario es igual que en el caso de uso 2, no es necesario realizar ninguna variación en él.

Flujo:

1: El usuario se identifica en la aplicación de Wallet ID TCV, la aplicación lanza una llamada al servidor para comprobar su identificación y si es favorable le concede acceso a la aplicación.

2: El usuario genera el código QR con la aplicación y el lector instalado en la cerradura lo escanea.

3: La unidad de control de la cerradura se comunica con Wallet ID para identificar al usuario que intenta acceder.

4: Si la identificación es correcta, la unidad de control se comunica con el servidor solicitando una autorización de acceso para ese usuario identificado. El servidor comprueba si el usuario tiene autorización para acceder a la cerradura, el servidor comunica el resultado a la unidad de control.

5: Si el servidor le ha comunicado a la unidad de control que el usuario tiene autorización para entrar entonces la unidad de control desbloquea la cerradura.

5.5. Implementación del Diseño Final del Proyecto

5.5.1. Implementación del Lector QR

Para la implementación del diseño final es necesario desarrollar un lector de códigos QR usando la unidad de control, para ello se va a realizar un programa en Python el cual sea capaz de leer un código QR y almacenar la información que se encuentra en el código. Primero es necesario instalar y configurar una cámara en la Raspberry Pi, la unidad de control del sistema, la cámara que sea escogido es una cámara básica integrada en la Raspberry Pi 3.

Una vez la cámara ha sido instalada y correctamente configurada es necesario buscar una librería que permita leer y manejar códigos QR en Python, para ello se ha escogido la librería OpenCV [19], concretamente una de las dos interfaces proporcionadas por OpenCV, cv2. OpenCV es una librería multiplataforma que se encuentra centrada en el procesamiento de imágenes. OpenCV ofrece una gran cantidad de herramientas como capturar video, detección de objetos y caras…

Entre todas las herramientas que ofrece OpenCV también se encuentra un detector de QR que es que se va a utilizar en este proyecto. En la Figura 19 que se encuentra a continuación se muestra el código implementado para la detección de código QR.

```
import cv2

# Detección de la cámara de la Raspberry Pi
cap = cv2.VideoCapture(-1)

detector = cv2.QRCodeDetector()#detector de QR

#Bucle de captura de video
while True:
    _, img = cap.read()#video capturado de la cámara
    data, bbox, _ = detector.detectAndDecode(img)#Detección y descodificación QR

    #Interfaz detección QR y valor QR
    if(bbox is not None):
        for i in range(len(bbox)):
            cv2.line(img, tuple(bbox[i][0]), tuple(bbox[(i+1) % len(bbox)][0]), color=(255,
                     0, 255), thickness=2)
        cv2.putText(img, data, (int(bbox[0][0][0]), int(bbox[0][0][1]) - 10), cv2.FONT_HERSHEY_SIMPLEX,
                    0.5, (0, 255, 0), 2)
        if data:
            print("data found: ", data)
    #Mostrar video capturado
    cv2.imshow("code detector", img)
    if(cv2.waitKey(1) == ord("q")):
        break
#Reset cámara
cap.release()
cv2.destroyAllWindows()
```

Figura 19: Código detección QR

En la Figura 19 se puede observar cómo una vez se inicializa la cámara y el detector de QR se captura el video capturado por la cámara y si se detecta un QR, si la variable bbox y data no son null, se extrae la información que contenga. Además de la detección de QR se ha implementado una interfaz básica para facilitar el testeo de la implementación, el programa muestra en una nueva ventana el video capturado por la cámara de la Raspberry Pi y además añade una interfaz a la detección del QR, resalta el QR y muestra la información que contiene el código. En la Figura 20 se puede observar cómo se detecta un código QR que se ha generado para testear el sistema, el código contiene la palabra prueba como información y en la interfaz se puede observar cómo se resalta el código QR y aparece la información que contiene, en este caso un string que contiene la palabra prueba.

Figura 20: Interfaz de detección de código QR

5.5.2. Apertura de cerradura mediante lectura de código QR

Una vez que se ha desarrollado la lectura de QR mediante el uso de una cámara integrada en la unidad de control es siguiente paso en el desarrollo del proyecto es la unión de los dos módulos que se encuentra desarrollados, el módulo de apertura de la cerradura y el que se comenta en la sección anterior, el módulo de lectura de un código QR. Primero es necesario desarrollar el nuevo circuito, partiendo del circuito previamente implementado, en la sección 5.3.2. El circuito del que se parte es un circuito que mediante él envió de señales de control al relé desde los pines de la Raspberry Pi se encarga de la apertura y cierre de la cerradura, partiendo de ese circuito es necesario la instalación de una cámara para obtener el circuito necesario para esta nueva sección del proyecto.

A continuación, en la Figura 21, se muestra el resultado final de la implementación del circuito, el cual, si no se detectan problemas en el diseño del proyecto, es el circuito final.

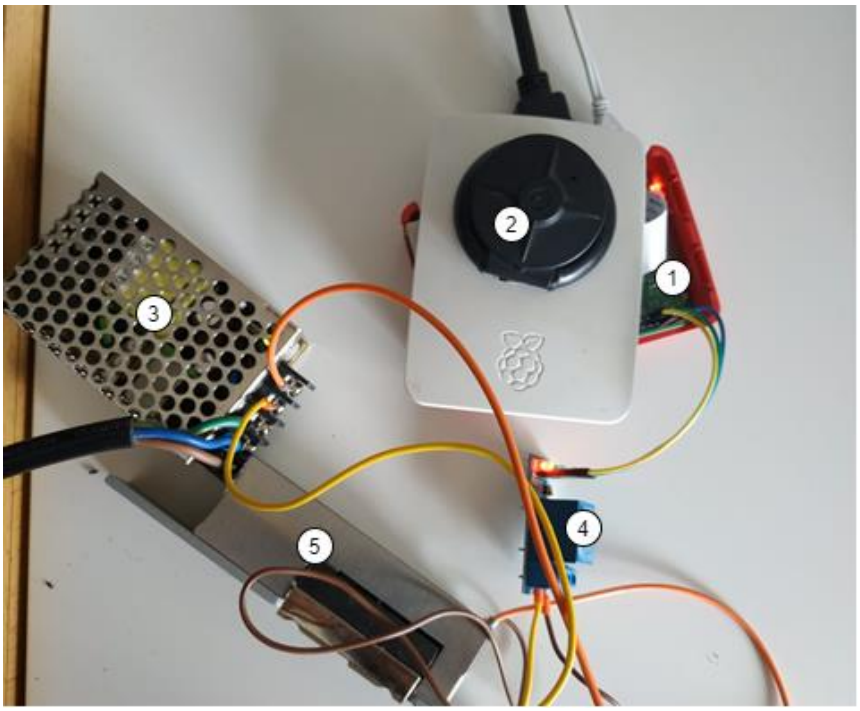


Figura 21: Circuito final del proyecto

En la Figura 21 se puede observar los diferentes componentes que forman el circuito, como unidad de control se utiliza una Raspberry Pi (1), en la cual se ha instalado una cámara para la lectura de códigos QR (2), además de la unidad de control también hay una fuente de alimentación (3) y un relé para controlar el circuito (4), finalmente también se puede observar el cierre eléctrico (5). Para esta fase se va a implementar la unión de los dos módulos comentados previamente formando la base de lo que va a resultar el programa final, el cual mediante su ejecución en la unidad de control será capaz de controlar la apertura de la cerradura.

En esta unión se ha simplificado el módulo de apertura de la cerradura:

- Se ha eliminado el control de la apertura mediante teclado debido a que su única función era de testeo, emulando el funcionamiento de un cierre mecánico como los que se pueden encontrar en muchos bloques de pisos, es decir la cerradura se abrirá y se mantendrá abierta durante un periodo corto de tiempo y luego se cerrará.

- El módulo de la lectura de QR se ha mantenido casi en su totalidad, aunque en el programa final se eliminará la interfaz cuya única función es añadir un apartado visual para facilitar el testeo.

```python
import cv2
import RPi.GPIO as gpio
import time

gpio.setwarnings(False)

#Función abrir 5 segundos la cerradura
def lockFunction():
    gpio.setmode(gpio.BOARD)
    gpio.setup(7, gpio.OUT)
    gpio.output(7, gpio.HIGH)

    time.sleep(5)
    gpio.cleanup()

#Función principal
def cameraFunction():
    #Configurar cámara y detector de QR
    cap = cv2.VideoCapture(-1)
    detector = cv2.QRCodeDetector()

    while True:
        _, img = cap.read()
        data, bbox, _ = detector.detectAndDecode(img)
        #Interfaz del programa
        if(bbox is not None):
            for i in range(len(bbox)):
                cv2.line(img, tuple(bbox[i][0]), tuple(bbox[(i+1) % len(bbox)][0]), color=(255,
                         0, 255), thickness=2)
            cv2.putText(img, data, (int(bbox[0][0][0]), int(bbox[0][0][1]) - 10), cv2.FONT_HERSHEY_SIMPLEX,
                        0.5, (0, 255, 0), 2)
            if data:
                #Comprobar valor del QR
                if (data == "Hello World"):
                    lockFunction()
                    cap.release()#Reiniciar cámara
                    cameraFunction()#Reiniciar función
        cv2.imshow("code detector", img)
        if(cv2.waitKey(1) == ord("q")):
            break
    cap.release()
    cv2.destroyAllWindows()

#Ejecutar función
cameraFunction()
```

Figura 22: Código de apertura de cerradura mediante detección de un código QR

En la Figura 22 se puede observar cómo se han creado dos funciones, la primera *lockFunction()* encargada de abrir la cerradura durante 5 segundos y luego cerrarla, y la segunda función *cameraFunction()* encargada de ejecutar todo el código del programa. Dentro de la función principal podemos observar que es bastante similar al programa mostrado en la sección 5.5.1, la diferencia es que en lugar de mostrar por consola el valor del código QR se ejecuta la función de apertura de la cerradura si el valor coincide con el valor indicado, en este caso *Hello World*.

A continuación, en la Figura 23 se puede observar un diagrama con el funcionamiento del programa implementado.

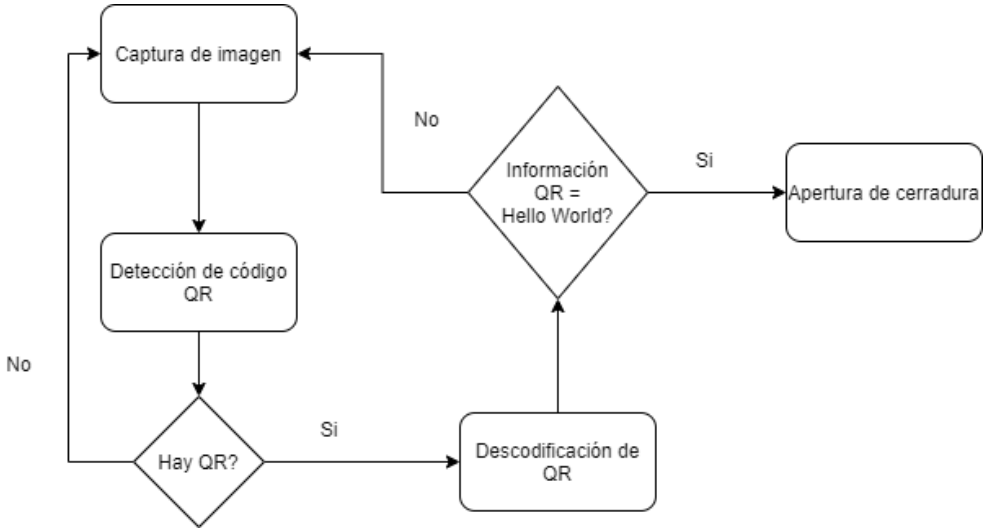


Figura 23: Diagrama con el funcionamiento de la apertura de la cerradura mediante un QR

5.5.3. Investigación previa a la implementación de la aplicación y la base de datos

Una vez se ha implementado la lectura de códigos QR por parte de la unidad de control, el siguiente paso es la implementación de una API donde registrar las diferentes cerraduras y donde comprobar la identidad del usuario que quiere acceder a la cerradura. La identidad del usuario se obtiene mediante una llamada a la aplicación de identidad digital mediante la lectura del QR que ofrece la aplicación de Tarjeta Ciudadana de Wallet Id, el proceso de comunicarse con la Wallet Id se va a realizar posteriormente a la construcción de la API y la base de datos debido a que primero es necesario poseer la infraestructura donde comprobar la identidad proporcionada por Wallet Id.

Para comenzar la implementación de esta fase del proyecto primero es necesario decidir las herramientas y lenguajes que se van a utilizar, es necesario decidir el tipo de base de datos a utilizar y el lenguaje con el que se va a implementar la aplicación. Primero la base de datos, es necesario decidir si la base de datos que se va a utilizar va a ser SQL O NoSQL, para ello es necesario realizar una sencilla investigación para conocer cual tipo se adapta mejor al proyecto, para ello se va a hacer una comparación de los dos tipos y posteriormente mediante una conclusión se va a decidir el tipo más adecuado.

La Tabla 4 refleja una comparativa entre las dos tecnologías de bases de datos.

Características	SQL	NoSQL
Almacenamiento	Tablas	Documentos: JSON, Tablas dinámicas…
Esquema	Rígido	Flexible
Escalabilidad	Vertical	Horizontal
Consultas	Joins requeridos, consultas más lentas	Joins requeridos, consultas más rápidas

Tabla 4: Bases de datos SQL vs NoSQL

Después de realizar el estudio y poder observar las características de ambos tipos de bases de datos se ha decidido que se va a utilizar una base de datos NoSQL por su esquema flexible gracias al cual la base de datos es dinámica y puede evolucionar a lo largo de la implementación, además de esta razón también ha sido elegida por sus consultas más rápidas y su tipo de almacenamiento, la posibilidad de trabajar con documentos JSON como almacenamiento facilita diferentes operaciones de la implementación [20]. Una vez se ha elegido el tipo de base de datos que se va a utilizar hay que seleccionar la base de datos que se va a utilizar, se ha decidido que se va a utilizar MongoDB por su gran flexibilidad y su almacenamiento de estructura similar a un JSON, además también se ha elegido por su afinidad con la herramienta utilizada para la implementación de la aplicación, la cual se explica a continuación [21]. La siguiente fase después de seleccionar la base de datos que se va a utilizar es seleccionar el lenguaje y herramientas que se van a utilizar para realizar la aplicación, encargada de ser el enlace de la base de datos con el resto de los sistemas que forman el proyecto.

Para la implementación de la aplicación se ha decidido utilizar Node.js. Node.js es un ambiente de ejecución de JavaScript que utiliza un modelo de entrada y salida sin bloqueos controlado por eventos lo que lo hace ligero y eficiente frente a otras aplicaciones que se ejecutan en tiempo real en los dispositivos. Una de las principales características es la creación de aplicaciones de red rápidas gracias a que es capaz de manejar una gran cantidad de conexiones simultáneamente manteniendo un alto nivel de rendimiento. Además de eso Node.js proporciona la posibilidad de realizar el front-end, back-end y aplicaciones móviles con un mismo lenguaje. Y para finalizar otra de las razones de que se haya elegido Node.js es porque cuenta con Node Package Manager (NPM) [22], que es una herramienta de gestión de paquetes que permite instalar módulos nuevos a un proyecto para facilitar la implementación.

Después de seleccionar Node.js como ambiente de ejecución tenemos que elegir el lenguaje con el cual implementar la aplicación, más concretamente el back-end. Para ello existe dos posibilidades, JavaScript y TypeScript. TypeScript es un lenguaje construido sobre JavaScript el cual añade nuevas funcionalidades a JavaScript siendo la principal una sintaxis más clara para la programación orientada a objetos. A la hora de elegir el lenguaje hay que tener en cuenta las dimensiones de la aplicación a implementar, y aunque TypeScript añade muchas funcionalidades que pueden resultar útiles en la implementación de una aplicación también añade complejidad por lo que debido a las dimensiones del proyecto se ha decidido que JavaScript es el lenguaje por utilizar en el desarrollo de la aplicación. Una vez se ha elegido las herramientas y lenguajes a utilizar se puede empezar el desarrollo del back-end de la aplicación, el front-end se desarrollará posteriormente de la implementación del back-end y también se realizará una investigación previa para conocer las mejores herramientas para desarrollar una aplicación web y una aplicación móvil si se implementa, esta última forma parte de los posibles elementos extras que añadir al proyecto.

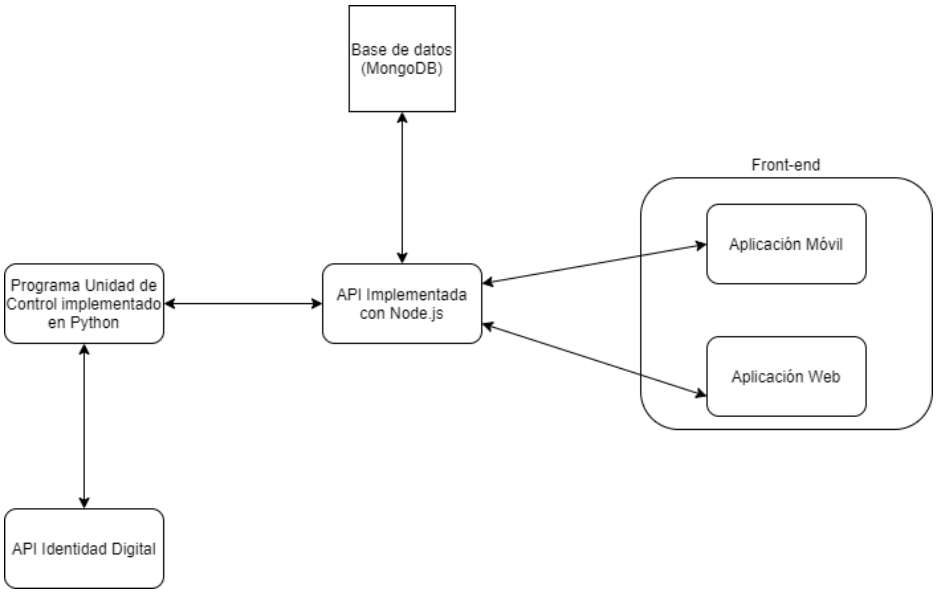


Figura 24: Arquitectura módulos del sistema

En la Figura 24 se muestra la arquitectura del proyecto, se puede observar los módulos que forman el proyecto, los actuales y los futuros, y las herramientas y lenguajes con las que han sido implementados, en el caso del Front-end están por definir.

5.5.4. Implementación de la aplicación y la base de datos

Con las herramientas seleccionadas es el momento de comenzar la implementación de la aplicación, para ello se ha utilizado Visual Studio Code como entorno de desarrollo.

La primera fase de la implementación consiste en la instalación de todos los módulos para luego posteriormente crear la estructura de la aplicación. Los módulos que se han instalado y utilizado se encuentran en el anexo del proyecto, pero a continuación se va a nombrar los más importantes y su uso en el desarrollo de la aplicación. Primero de todo es necesario instalar el módulo llamado NPM el cual se nombra en la sección anterior de la memoria, este módulo permite una fácil instalación de módulos en nuestro proyecto además de otras funciones que facilitan el desarrollo. A continuación, se instalan el resto de los módulos, tres de los más importantes son los siguientes: express, mongoose y passport.

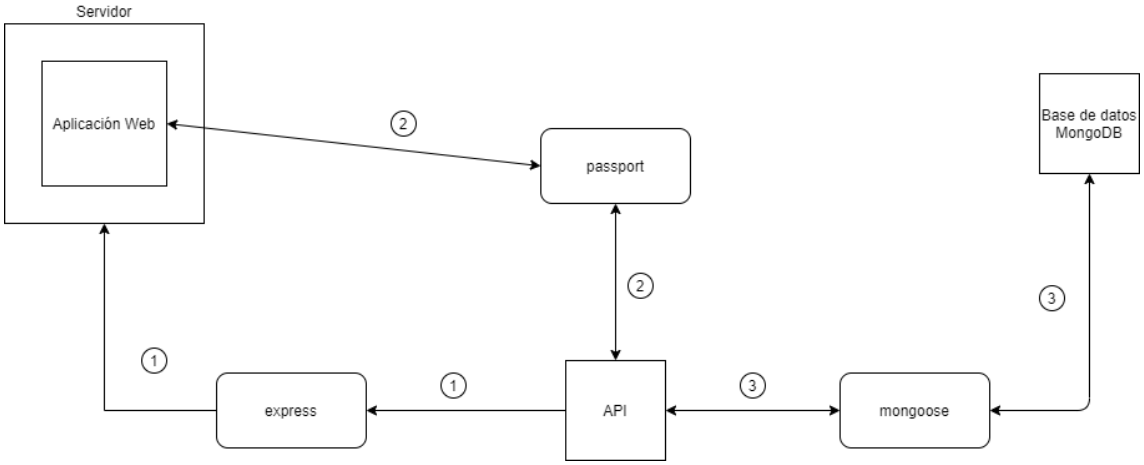


Figura 25: Funcionamiento librerías principales de la aplicación

En la Figura 25 se puede observar cómo interactúan los módulos nombrados con la aplicación, el módulo de express es el framework encargado de construir e iniciar el servidor el cual es la base de la aplicación (1), mongoose es el módulo encargado de la comunicación con la base de datos, en este caso con MongoDB (2), mongoose proporciona una solución sencilla basada en esquemas para modelar los datos de la aplicación que van a ser almacenados en la base de datos y además mongoose proporciona una gran variedad de herramientas con las que facilitar la manera de interactuar con la base de datos (validación, creación de consultas…). Para finalizar otro de los módulos más importantes es passport, passport es un módulo de autentificación el cual proporciona diversos tipos de autentificación con los cuales proteger tus aplicaciones (3).

Una vez se han instalado todos los módulos el siguiente paso es la creación y configuración de la estructura del proyecto, se realiza todas las configuraciones necesarias para la comunicación con la base de datos, se realiza las configuraciones básicas de mongo como la dirección de la base de datos y otras variables de entorno de MongoDB. También se inicia el servidor utilizando el módulo express, se configura las rutas de algunos archivos que van a ser utilizados por varios módulos de la aplicación y se inicializan los diferentes middlewares que se van a utilizar. La próxima fase de la implementación es la creación e inicialización de las colecciones de la base de datos. Primero es necesario estudiar las colecciones que son necesarias para la implementación de la aplicación para eso se ha realizado un esquema con las posibles colecciones y los atributos necesarios. En la Figura 26 que se encuentra a continuación se muestra un esquema de las colecciones que se van a implementar.

Figura 26: Esquema colecciones base de datos

En la Figura 26 se puede observar 3 tablas diferentes, cada una de esas tablas es un esquema de las colecciones que se han implementado en la base de datos. En las tablas se puede observar dos columnas diferentes, en la derecha están las variables, y el tipo de cada una, que tiene cada colección, y en la izquierda se puede observar dos siglas RQ Y UC, que significan requerido y único respectivamente. La columna de la izquierda está destinada a los tipos de variable que existen en MongoDB, en la implementación de la aplicación se han utilizado dos tipos, requerido y único, requerido hace referencia a variables que son obligatorias introducir para la creación de un objeto en una colección, y único hace referencia a las variables que no pueden tener el mismo valor en dos o más objetos diferentes de una misma colección.

A continuación, se va a comentar con detalle cada una de las colecciones que se han implementado en la base de datos, se ha implementado una colección donde poder almacenar las diferentes cerraduras, en cada cerradura se guarda una variable de tipo String y requerida con el nombre de la cerradura que ha introducido el usuario para identificar la cerradura, otra variable de tipo String y opcional con una descripción de la cerradura en el caso de que el usuario

quiera añadir una pequeña descripción a la hora de crear la cerradura. Las dos últimas variables son de tipo String, la primera la id del usuario propietario de la cerradura para poder relacionar las cerraduras con los usuarios que las han creado, esta variable es requerida, y para finalizar una variable requerida y única donde almacenar la dirección MAC de cada cerradura. La dirección MAC funciona como un identificador con el cual identificar la cerradura cuando se realizan llamadas a la base de datos desde la unidad de control.

Las dos colecciones restantes son usuarios e invitados, ambos son tipos de usuarios que utilizan la cerradura para acceder a un lugar, los usuarios son los propietarios de la cerradura y pueden acceder en cualquier momento, los invitados tienen el acceso limitado, el acceso de los invitados es controlado por los propietarios de la cerradura, son los usuarios los que registran a los invitados y tienen control total sobre ellos, pueden eliminarlos y pueden limitar su acceso mediante una variable. La colección de usuarios tiene cuatro variables de tipo String, la primera es una variable requerida donde almacenar el nombre del usuario, la segunda variable es única y requerida y almacena el email del usuario, el email funciona como un identificador del usuario y junto a la contraseña son las dos variables necesarias para que el usuario se consiga acceder a la aplicación correctamente mediante el método de login. Las otras variables almacenadas son el teléfono y la contraseña, la primera es igual que el email única y requerida y en una posible actualización de la aplicación podría compartir con el email su función en el acceso de la aplicación. Por último, está la contraseña, es una variable requerida pero su almacenamiento es algo más complejo que el resto de las variables.

La contraseña de un usuario en todas las aplicaciones es una información que debe tratarse con mucha seguridad para evitar que se produzcan perdida de información y el usuario se vea perjudicado. Para la seguridad de la contraseña del usuario se ha decidido encriptar la contraseña, es decir, en la base de datos no se va a almacenar la cadena de caracteres que el usuario indique como su contraseña, sino que se va a utilizar alguna herramienta que encripte esa cadena de caracteres introducida por el usuario y así si un atacante consigue acceder a la base de datos y si logra acceder a la contraseña del usuario esta se encuentre protegida para que el atacante no pueda utilizar para dañar al usuario. Para encriptar la contraseña del usuario se ha utilizado una función criptográfica llamada hash, hash es un algoritmo matemático que convierte cualquier bloque de datos en una cadena de caracteres, en el Anexo II – Librerías utilizadas en la aplicación se encuentra explicado en detalle la función hash. Para utilizar la función hash se ha utilizado una librería que proporciona esa función entre otros métodos relacionados con el ámbito de la criptografía, la librería se llama bcrypt [23]. En la imagen a continuación se puede observar cómo

se han creado dos métodos para la seguridad de la contraseña del usuario. El primer método mediante el uso de la función *hash()* implementada por la librería bcrypt se encripta la contraseña del usuario, en el segundo método se utiliza el método *compare()* de bcrypt para comprobar si una cadena de caracteres es igual a otra cadena de caracteres encriptada, esta función se utiliza en la aplicación en el método login para comprobar si la contraseña introducida por el usuario para acceder a la aplicación es igual a la que se encuentra encriptada en la base de datos.

```
UserSchema.methods.encryptPassword = async password => {
    const salt = await bcrypt.genSalt(10);
    return await bcrypt.hash(password,salt);
};

UserSchema.methods.matchPassword = async function(password) {
    return await bcrypt.compare(password,this.password)
};
```

Figura 27: Código de encriptación y desencriptación

Y por último la colección donde se almacenan los invitados creados por los usuarios. Las variables que se almacenan en esta colección son similares a las que se almacenan en la colección de usuarios debido a que las dos colecciones almacenan los datos de una persona que intenta acceder a la cerradura. En la colección de invitados se almacenan las variables nombre, email y teléfono exactamente igual que en la colección de usuarios. Las diferencias entre las dos colecciones son tres:

- Primero los invitados no necesitan almacenar una contraseña puesto esta es utilizada para acceder a la aplicación web y los invitados no necesitan acceder a ella.
- Segundo es necesario almacenar la cerradura a la cual se les permite el acceso, para ello se almacena una variable de tipo requerido con la identificación de la cerradura.
- Y por último la principal diferencia entre usuarios e invitados es que el acceso de los invitados puede estar limitado, es decir, es posible limitar su acceso a una franja horaria del día, para ello se almacena una variable de tipo Array opcional donde guardar la franja de acceso del invitado.

Todas las colecciones se han creado mediante la implementación de un modelo de datos de cada colección utilizando la librería mongoose la cual facilita las operaciones de manejo de datos con la base de datos. Una vez se ha creado las colecciones en la base de datos y los modelos en la aplicación se puede comenzar a trabajar con los datos almacenados en la base de datos. Para ello se ha creado un archivo para cada colección llamado controlador en el cual se va a

implementar métodos los cuales interactúan con la base de datos. En cada controlador se realizan diferentes métodos dependiendo de los datos que se necesiten de cada colección, pero hay cuatro métodos que comparten los tres controladores y son las llamadas funciones CRUD. CRUD son los métodos básicos que hay que desarrollar para cada colección, tabla o cualquier tipo de objeto que se almacene en una base de datos, CRUD está formado por cuatro funciones las cuales le dan nombre, Create, Read, Update y Delete, en español las funciones son, Crear, Leer, Actualizar y Borrar.

La función Create almacena un nuevo objeto en la base de datos mediante la función *save()* proporcionada por la base de datos, concretamente por la librería mongoose que funciona como un enlace entre la aplicación y MongoDB. La función Read busca y obtiene los objetos de la base de datos que cumplan los valores de búsqueda, para ello se utiliza la función *find()* donde se puede añadir parámetros para la búsqueda como el valor de una variable del objeto u objetos que se desea obtener. Además de la función *find()* existen variantes las cuales vienen con algunas de esas variables de búsqueda integradas, como por ejemplo *findById().* Y por último las funciones Update y Remove, las dos funciones necesitan la id del objeto con el cual se va a realizar la operación, Update actualiza el objeto al cual pertenece la id indicada y Remove elimina el objeto al cual pertenece la id indicada. Las funciones proporcionadas por la base de datos son, *findByIdAndUpdate()* y *findByIdAndRemove().*

Una vez se ha implementado las funciones que forman CRUD para cada una de las colecciones se implementa las funciones necesarias para las operaciones a realizar con cada uno de los objetos almacenados en cada una de las colecciones. En el controlador del usuario se ha implementado, además de las funciones CRUD, una función de búsqueda por un email y por una dirección MAC indicada, esta función es necesaria porque cuando un usuario quiere acceder a la cerradura se realiza una llamada a la aplicación indicando el email del usuario y la dirección MAC de la cerradura que quiere acceder y se comprueba si tiene permiso.

En el controlador del invitado se ha implementado, además de las funciones CRUD, una función de búsqueda por un email y por una dirección MAC indicada al igual que en el controlador del usuario, el uso de esta función es el mismo que en el controlador del usuario. Y para finalizar, en el controlador de la cerradura se ha implementado las funciones CRUD, no se ha implementado una función de búsqueda, en cambio sí que se realiza una búsqueda por dirección MAC de cerradura en los controladores de invitado y usuario. Se ha tomado esta decisión porque tanto la colección de los usuarios como la colección de invitados no almacena la dirección MAC de la

cerradura a la que pueden acceder, para ello es necesario una búsqueda que obtenga una cerradura mediante una dirección MAC indicada y una vez se ha obtenido la cerradura, los controladores tienen acceso a toda la información necesaria para comprobar si una persona tiene acceso a una cerradura.

```
usersCtrl.findByEmailAndMacAddress = async (req, res) => {
    const lock = await Lock.findOne({macAddress: req.query.macAddress}).lean();
    if(lock)
    {
        const user = await User.findOne({_id: lock.user, email: req.query.email}).lean();
        if(user)
        {
            res.send(user);
        }
        else{
            res.sendStatus(404);
        }
    }
    else{
        res.sendStatus(404);
    }
```

Figura 28: Función de búsqueda de usuario mediante una dirección MAC y un email

En la Figura 28 se puede observar cómo funciona el método implementado en el controlador del usuario que es similar al controlador del invitado, primero es necesario obtener el objeto de la cerradura que realiza la llamada, una vez que se obtiene la cerradura se realiza la segunda búsqueda, aquí es donde se diferencian el usuario y el invitado, el usuario realiza una búsqueda utilizando el email proporcionado por la unidad de control y el id del usuario propietario de la cerradura comprobando así si la persona que intenta acceder es el propietario de la cerradura que intenta abrir. El método del controlador del invitado funciona diferente, realiza una búsqueda utilizando el email proporcionado por la unidad de control y el id de la cerradura obtenida comprobando así si la persona que intenta acceder es un invitado con acceso a esa cerradura.

Además de las funciones de búsqueda en los controladores también se han implementado funciones para la renderización de apartados de la aplicación web, pero al tratarse del apartado de front-end de la aplicación se explicarán más adelante en la sección 5.5.9. Una vez se han implementado los diferentes métodos necesarios en los controladores es necesario hacer que sean accesibles por módulos ajenos a la aplicación, para ello se crean los Endpoints. Los Endpoints son las herramientas de comunicación que utilizan sistemas ajenos a la aplicación para acceder a los diferentes recursos de la API, las APIs funcionan mediante requests y responses, peticiones y respuestas, el lugar al que una aplicación web o un servidor web lanza una petición esperando una respuesta se llama Endpoint.

Para crear los diferentes Endpoints necesarios en la aplicación se han creado tres archivos de rutas, uno para cada colección, en estos archivos se crean los diferentes Endpoints y cada uno utiliza una de los métodos previamente definidos e implementados en los controladores. Los Endpoints se crean utilizando el módulo Router de la librería express y cada Endpoints se configura para que utilice uno de los diferentes métodos de petición de HTTP (GET, POST, PUT, DELETE) dependiendo de la operación que quiera realizar, si es lectura se utilizará el método GET, si es escritura en la base de datos el método POST, si se quiere actualizar un elemento de la base de datos se utiliza el método PUT, y finalmente si se quiere eliminar un elemento se utiliza el método DELETE, toda esta información se refleja en la Figura 29.

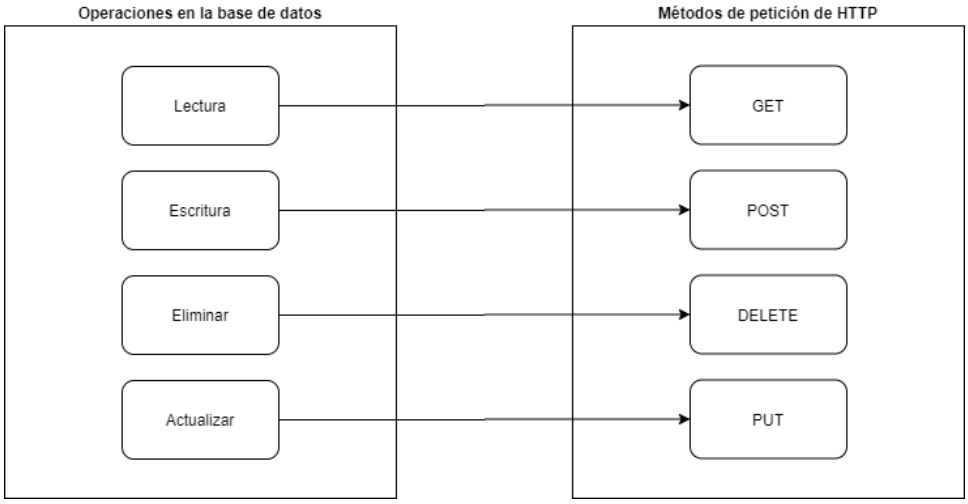


Figura 29: Operaciones en la base de datos y sus respectivas peticiones HTTP

5.5.5. Uso de la aplicación de identidad digital desde la unidad de control

Una vez se ha implementado el programa con el que se controla la apertura y el cierre de la cerradura desde la unidad de control y la API donde almacenar los usuarios, invitados y las cerraduras, el siguiente paso es utilizar la identidad digital para abrir la cerradura, para ello se necesita utilizar la aplicación Wallet Id TCV, aplicación desarrollada por Inycom la cual permite crear una Tarjeta Ciudadana Virtual con la que acceder a diversos servicios mediante la identidad digital. Wallet Id TCV proporciona un generador de códigos QR el cual muestra un código con el cual al leerlo obtener información del propietario de la tarjeta ciudadana virtual, el sistema a implementar consiste en la lectura del código QR mediante la cámara instalada en la cerradura, una vez se ha obtenido información del usuario se utiliza esa información para conocer si el usuario tiene permitido el acceso a la cerradura, para ello se realizará una llamada a la API desarrollada mediante uno de los Endpoints creados el cual devolverá un código de respuesta HTTP 200 si el usuario puede acceder y entonces la unidad de control abrirá la cerradura, o devolverá un código de respuesta HTTP 400 si el usuario no puede acceder por lo que la unidad

de control mantendrá cerrado el cierre eléctrico. En la Figura 30 se muestra un esquema del funcionamiento del sistema que se ha comentado.

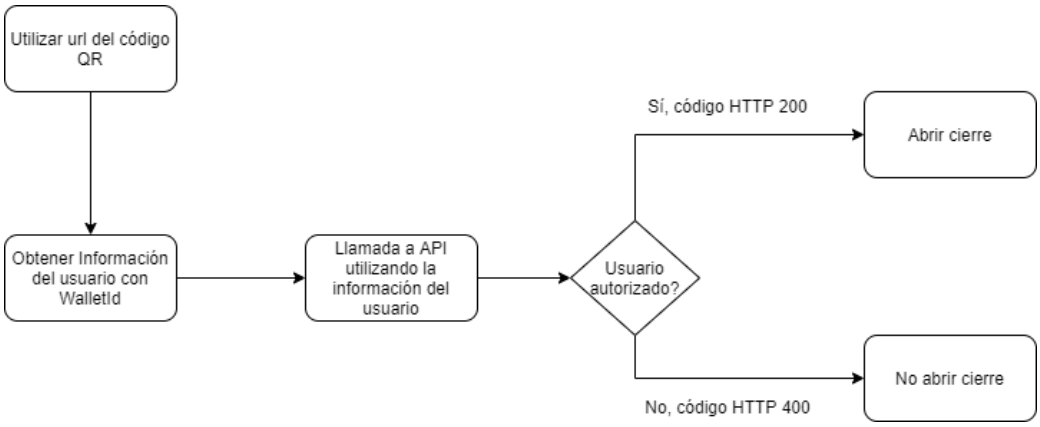


Figura 30: Funcionamiento del programa python tras detectar QR

Con la arquitectura del sistema completo diseñada y desarrollada se comienza con la implementación del sistema final, más que una implementación es la unión de los diversos módulos implementados en secciones anteriores, cuando se comienza la implementación se ha descubierto un error, concretamente un módulo que no cumple las características necesarias, este error se encuentra en el módulo de lectura de códigos QR.

La lectura de QR ha funcionado correctamente en todas las pruebas que se han realizado pero estas pruebas se han realizado utilizando una aplicación que genera código QR, estos códigos generados son de mayor tamaño que el código QR que se genera en la aplicación de Wallet Id TCV, cuando se intenta leer el código QR de la aplicación utilizando el módulo de lectura de códigos QR implementado la cámara no consigue detectar el código QR. La razón de este comportamiento es debido a la resolución a la que trabaja la cámara que se utiliza, es una cámara bastante sencilla por lo que la resolución de captura de video no es la suficiente para poder detectar códigos QR de ese tamaño. Una vez identificado el problema es el momento de pensar una solución, la solución más sencilla sería utilizar una cámara de mayor calidad que cuente con una mayor resolución a la hora de capturar video, pero se ha decidido que antes se van a estudiar otras soluciones.

La primera solución que se ha probado ha sido la utilización de dos funciones de la librería OpenCV para mejorar la calidad de la imagen capturada, *resize()* y *convertScaleAbs()* . OpenCV es la librería que se utiliza en el módulo de lectura de QR y al tratarse de una librería cuya principal función es procesado de imágenes cuenta con varias funciones de procesado de imágenes. Primero se ha probado redimensionando la imagen obtenida utilizando el método *resize()* que

proporciona la librería pero el resultado ha sido el mismo, debido a que la función *resize()* no realiza una operación de zoom, simplemente cambia el tamaño de la imagen pero a su vez se empeora la resolución de esta. Debido a no obtener una solución al problema que se ha presentado se ha buscado otra solución. Posteriormente se intentó cambiar el contraste de la imagen intentando mejorar la detección del código QR utilizando la función *convertScaleAbs()* de la librería como se puede observar en la Figura 31, pero el resultado tampoco ha sido el que se necesita.

```
alpha = 1.5 # Contrast control (1.0-3.0)
beta = 0 # Brightness control (0-100)

adjustedImage = cv2.convertScaleAbs(img, alpha=alpha, beta=beta)
```

Figura 31: Código de cambio de contraste de un video

La segunda solución que se ha pensado ha sido utilizar otra librería para la detección del QR, se ha pensado seguir utilizando OpenCV para capturar video con la cámara de la Raspberry Pi, pero utilizar otra librería para capturar y decodificar el código QR. OpenCV ofrece un método de detección de código QR, pero al no ser una librería centrada únicamente en la detección de código QR se ha pensado que quizás una librería centrada únicamente en la detección de códigos QR pueda ofrecer una mejor detección. Se ha investigado las librerías Python de lectura de QR que son más utilizadas y se han seleccionado dos de ellas, pyzbar [24] y qrtools [25]. Ambas librerías no son solo las más utilizadas, sino que más bien son casi las únicas utilizadas, junto a OpenCV, en proyectos donde se necesita lectura de códigos QR. Primero se han probado ambas librerías con una imagen capturada por la cámara donde se puede ver el código QR que genera la aplicación Wallet Id TCV y han podido detectarlo perfectamente por lo que el siguiente paso es integrarlas una de las librerías junto a OpenCV en el módulo de lectura de QR, cuando se ha intentado la integración ha sucedido otro problema con la librería qrtools, esta librería no se encuentra activa para Python 3, solo funciona para Python 2, y se ha tenido que descartar porque OpenCV solo funciona en Python 3 y OpenCV es la librería que se encarga de capturar video por lo que es indispensable para la solución por lo que se ha decidido descartar qrtools y utilizar pyzbar que si funciona en Pyhton 3.

La integración de pyzbar se realiza correctamente, se deja de utilizar el módulo de detección de códigos QR que implementa OpenCV y se utiliza el detector que proporciona pyzbar como se puede ver en la Figura 32.

```
#Detector de QR de OpenCV
#data, bbox, _ = detector.detectAndDecode(img)

#Detector de QR de pyzbar
data = decode(image)
if data:
    print(data[0].data.decode())
```

Figura 32: Cambio de detector de QR en el código de la unidad de control

A pesar del cambio de detector de códigos QR el resultado sigue siendo el mismo, el código QR de la aplicación de Wallet Id TCV es demasiado pequeño para que se detecte, por ello se ha implementado otra forma de detección de QR, se ha tenido que rehacer por completo el módulo de detección de QR. Para la nueva implementación se va a realizar capturas de imagen cada dos segundos en vez de capturar video, porque la resolución a la que captura video la cámara utilizada es menor que la resolución a la hora de capturar imágenes y se sabe que desde una imagen pyzbar sí que puede detectar un código QR del tamaño como el que proporciona la aplicación porque, como se ha comentado, cuando se han probado las librerías han sido probadas utilizando imágenes y han funcionado correctamente. El nuevo módulo de lectura de código QR funciona de la siguiente manera, se realizan capturas de imagen cada dos segundos y estas capturas se guardan, la imagen que se almacena se va sobrescribiendo por lo que no existe posibilidad de un problema de espacio al almacenar muchas imágenes, y una vez guardada la imagen se utiliza el detector de códigos implementado por la librería pyzbar para obtener la información almacenada en el código QR.

```
while True:
    camera.capture('exampleTakingQR.jpg')

    d = decode(Image.open("exampleTakingQR.jpg"))
    if d:
        data = d[0].data.decode()
        print(data)

    time.sleep(2)
    if(cv2.waitKey(1) == ord("q")):
        break
```

Figura 33: Programa de lectura de códigos QR

5.5.6. Llamada a aplicación de identidad digital Wallet Id

Una vez se ha conseguido detectar el código QR de la aplicación móvil Wallet Id TCV es el momento de hacer la llamada a la aplicación de identidad digital. Para ello es necesario obtener

una parte de la url que se obtiene al leer el código QR de la aplicación móvil. No se puede utilizar la url que se obtiene cuando se descodifica el código QR porque esa url llama a una web que a su vez llama a otros servicios en segundo plano, si realizamos una petición con la url del código obtenemos un error y no la información del usuario que es lo se espera, para poder obtener esa información es necesario realizar directamente una petición a uno de los servicios que llama la web en segundo plano, para ello necesitamos la id del usuario. La id del usuario es una de las variables que se encuentran en la url del código QR por ello debemos dividir la url hasta obtener únicamente un string con la id del usuario.

Una vez se ha obtenido la id del usuario se puede realizar la petición al servicio correcto y como respuesta se obtiene la información del usuario, entre esa información el email del usuario, el cual se va a utilizar para identificar al usuario en nuestro sistema. A continuación, en la Figura 34 se puede observar la respuesta de la aplicación de Wallet Id a la petición, la url del servicio y la petición no se muestran en la memoria del proyecto debido a que es un servicio privado y se utiliza una autorización privada pertenecientes a la empresa que ha desarrollado la aplicación de identidad digital.


```
"User": {
    "accountId": "f65790ad-af05-42dd-85d0-5aec81cf617e",
    "cardEnabled": true,
    "credentialLevel": 1,
    "email": "jesuslacartee@gmail.com",
    "enabled": true,
    "id": "699c947f-c3ff-4d51-9901-573fce57b89c",
    "identity": {
        "created": "2020-06-25T07:20:56Z",
        "documentExpiration": "2021-06-25T07:19:46Z",
        "documentType": "",
        "fullname": "Jesus Lacarte",
        "identifier": "1R",
        "validated": true,
        "validatedBy": "25169334C",
        "validatedByUserId": "c2b519bb-abee-4a97-953e-a259d9b107b1",
        "validationMode": "PRESENTIAL",
        "validationSignatureId": ""
    },
    "name": "jesus.lacarte",
    "phone": "34600111222",
```


Figura 34: Respuesta petición aplicación identidad digital

5.5.7. Implementación del programa final de la unidad de control

Para la implementación final del programa que se ejecuta en la unidad de control y controla la apertura y el cierre de la cerradura se tiene que unificar el módulo de apertura de la cerradura, el módulo de lectura de códigos QR que se ha vuelto a implementar en la sección anterior y un pequeño módulo de peticiones HTTP que se va a desarrollar a continuación.

Con los módulos que se encuentran actualmente implementados se puede leer un código QR y abrir la cerradura, pero también es necesario comprobar la identidad de la persona que quiere acceder, para ello es necesaria realizar peticiones a la aplicación de identidad digital Wallet Id y a la API que se ha desarrollado donde se almacena los usuarios y las cerraduras del sistema. Para realizar las peticiones a las aplicaciones se ha utilizado la librería de Python llamada requests [26], la cual permite realizar peticiones HTTP y obtener las respuestas a estas peticiones y la información que contienen. Una vez se realizan las peticiones a los Endpoints adecuados de las aplicaciones se puede unificar todos los módulos para crear el programa final.

La implementación final funciona de la siguiente manera, se mantiene capturando imágenes cada dos segundos, si en una de las imágenes el detector de código QR detecta un código obtiene la información del código y realiza una petición a la aplicación de identidad digital utilizando como url de la petición la información obtenida de código QR, si la identidad del usuario que quiere acceder es correcta wallet id envía una respuesta a la unidad de control con información del usuario. De toda la información sobre el usuario que envía la aplicación de Wallet Id la que se va a utilizar es el email del usuario, el email se utiliza en la aplicación desarrollada como un identificador de los usuarios y los invitados. Una vez se ha obtenido el email de la persona que quiere acceder se realiza una petición a un Endpoint de la aplicación que almacena los usuarios, este Endpoint necesita otro dato más además del email del usuario, necesita un identificado de la cerradura a la que se quiere acceder. Como identificador se ha decidido utilizar la dirección MAC de la unidad de control, para obtenerla se ha utilizado una librería llamada getmac. Con la dirección MAC y el email del usuario que quiere acceder se realiza una comprobación de credenciales de forma que se puede observar en la Figura 35.

```python
def checkCredentials(data):
    urlUser = urlBasicUser + data
    response = requests.get(urlUser)
    if response.status_code == 200:
        dataGuest = response.json()
        if dataGuest != "":
            return True
        else:
            return False
    else:
        urlGuest = urlBasicGuest + data
        response = requests.get(urlGuest)
        if response.status_code == 200:
            dataGuest = response.json()
            if dataGuest != "":
                return True
            else:
                return False
        else:
            return False
```

Figura 35: Comprobación de credenciales

Para comprobar las credenciales del usuario a acceder primero se realiza una llamada al Endpoint que realiza la búsqueda de un usuario utilizando un email y una dirección MAC, si se encuentra el usuario se devuelve un código de respuesta HTTP 200 y la función *checkCredentials()* devuelve una variable booleana de valor True, en el caso de que el usuario no se encuentre, se realiza otra petición a la API, pero en este caso se realiza una búsqueda en la colección de invitados, si la búsqueda es exitosa la aplicación devuelve un código de respuesta HTTP 200 y la función *checkCredentials()* devuelve una variable booleana de valor True, en el caso de que la persona que quiere acceder no se encuentre significa que no tiene las credenciales necesarias para acceder por lo que la función *checkCredentials()* devuelve una variable booleana de valor False. El orden en que se comprueba las credenciales del usuario es intencionado, es decir se busca primero si la persona que quiere acceder a la cerradura es el propietario y luego se comprueba si es un invitado, esto se debe porque la mayoría de las veces la persona que quiere acceder es la propietaria de la cerradura por lo que con este orden se evita tener que realizar dos peticiones a la aplicación la mayoría de las veces.

Una vez se han comprobado las credenciales del usuario y son correctos, es decir la función *checkCredentials()* devuelve una variable booleana de valor True, se procede a la apertura del cierre eléctrico utilizando el módulo de apertura de cerradura implementado en la sección 5.3.3.

En la Figura 36 podemos observar un diagrama de secuencia con el funcionamiento del sistema.

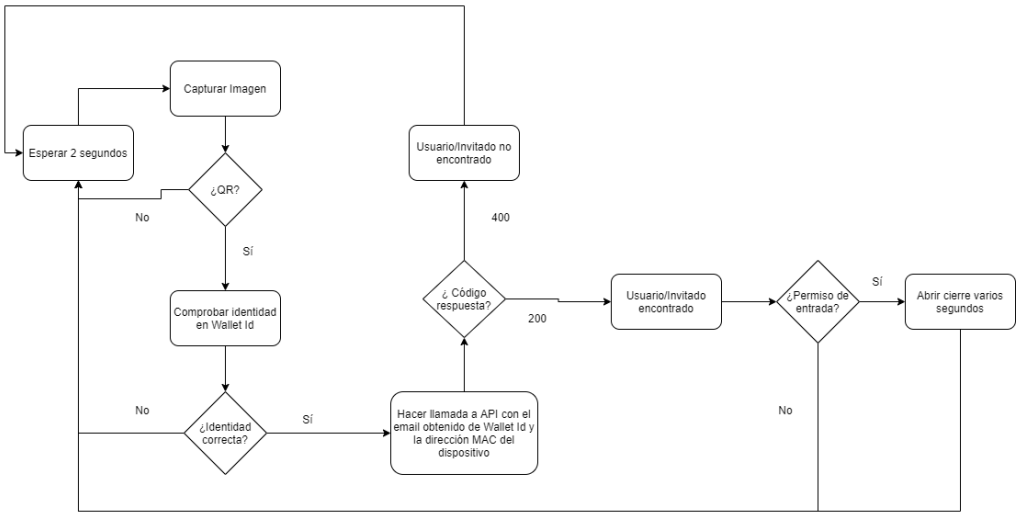


Figura 36: Diagrama Secuencia Programa Final Unidad de Control

5.5.8. Implementación de funcionalidad de acceso temporal para invitados

Una vez se ha implementado el programa final de la unidad de control se va a añadir una funcionalidad extra la cual se ha comentado en algunas secciones del proyecto, esta funcionalidad es la posibilidad de hacer que el acceso de un invitado se ha limitado por tiempo, es decir la posibilidad de crear un rango de tiempo donde el invitado pueda acceder y fuera de ese rango no pueda acceder. Para ello se utiliza una variable previamente creada en la base de datos en la colección de invitados llamada rangeAccess, esta variable es un array donde se va a almacenar dos strings, los cuales indicarán el comienzo y el final del rango de acceso.

Con esa funcionalidad ya implementada en la base de datos es necesario implementar una función en el programa de unidad de control donde se compruebe si el invitado pueda acceder. Para realizar operaciones con fechas y horas se ha utilizado la librería datetime de Python, primero se ha creado una nueva función llamada *checkRangeAccess()* la cual es llamada en el caso de que el usuario que quiera acceder sea un invitado. Dentro de esta función primero se obtiene la fecha y hora actual, para ello se utiliza la función *today()* que implementa la librería datetime. Una vez se ha obtenido la fecha y hora actual es necesario convertir el rango de acceso del invitado a un formato adecuado para poder realizar las comparaciones correctamente. El rango de acceso del usuario se encuentra almacenado en dos strings dentro de un array, primero es necesario dividir cada uno de los strings en los diferentes datos que se encuentran en ellos (año, mes, día y hora) para una correcta implementación en la página web se encuentra indicado el correcto formato que se tiene que usar para introducir las fechas a la base de datos. Una vez obtenidos los

diferentes elementos es necesario transformarlos a enteros porque la librería datetime no funciona con variables del tipo string. Una vez se encuentran todos los datos divididos y con el tipo de variable correcto se puede crear las dos variables donde se almacena la fecha de entrada y la fecha de salida que marcan el rango de tiempo cuando el invitado puede acceder, cuando sean creado las variables se realiza una sencilla comparación conociendo si el usuario puede acceder o no y abriendo la cerradura en el caso afirmativo.

```
def checkRangeAccess(rangeAccess):
    s1 = rangeAccess[0].split('/')
    s2 = rangeAccess[1].split('/')

    time = datetime.today()
    time1 = datetime(int(s1[0]), int(check0FirstLetter(s1[1])),
     int(check0FirstLetter(s1[2])), int(check0FirstLetter(s1[3])))
    time2 = datetime(int(s2[0]), int(check0FirstLetter(s2[1])),
     int(check0FirstLetter(s2[2])), int(check0FirstLetter(s2[3])))


    if time > time1:
        if time < time2:
            return True
        else:
            return False
    else:
        return False
```

Figura 37: Función de comprobación de acceso de invitados

En la Figura 37 se puede observar el funcionamiento que se ha explicado además de una función que no ha sido explicada, esta función es *check0FirstLetter()* la cual se encarga de eliminar los ceros que se encuentren a la izquierda en los meses, días y horas porque en el caso de dejar una fecha con esos ceros se produce un error.

5.5.9. Implementación de la aplicación web

Una vez se ha implementado el sistema completo, mediante el cual un usuario puede acceder a la cerradura utilizando la aplicación de Wallet Id TCV, el siguiente paso es desarrollar una aplicación web donde el usuario propietario de la cerradura pueda configurar los aspectos necesarios. Cuando se comenta, la aplicación web es para el usuario propietario de la cerradura, como se nombra en secciones anteriores existen dos tipos de usuarios, usuarios propietarios y usuarios invitados, los usuarios invitados no tienen acceso a la aplicación web, son solo los propietarios los que pueden configurar las cerraduras y todos los aspectos relacionados con ellas. En la aplicación web el usuario puede realizar todo tipo de operaciones relacionadas con las cerraduras, puede registrar cerraduras, eliminar cerraduras, editar cerraduras y además puede

gestionar los invitados de las diferentes cerraduras. Además de estas operaciones el usuario puede editar sus datos al igual que en cualquier aplicación web donde un usuario necesita registrarse. Para implementar la aplicación web se ha utilizado el módulo express-handlebars, este módulo permite el uso del motor de plantillas llamado Handlebars en un servidor creado con express. Handlebars [27] es un sistema de plantillas en Javascript basado en el lenguaje Mustache Templates. Handlebars permite escribir bloques de código escritos en HTML en los cuales puedes añadir datos que provengan de un JSON, con Handlbars te evitas usar otras librerías para añadir información, que provenga del back-end de tu proyecto, en un código HTML, Handlebars proporciona una manera de sencilla de comunicación entre el back-end y el front-end de un proyecto. Además de Handlebars se ha utilizado otra herramienta para la implementación de la aplicación web, esa herramienta es Bootstrap 4 [28]. Bootstrap es un framework construido en HTML, CSS y JavaScript que facilita el desarrollo de aplicaciones web y móvil. Una de las principales ventajas de usar Bootstrap es su flexibilidad a la hora de adaptarse a los diferentes tamaños de pantallas en los que se muestra la aplicación desarrollada.

Para comenzar el desarrollo de la aplicación web, primero hay que configurar el módulo de express-handlebars en el archivo llamado server de la API, en ese archivo se inicializa y configura todos los middlewares utilizados entre otras cosas. Además de las configuraciones básicas del módulo express-handlebars se ha utilizado una herramienta llamada helper que ofrece Handlebars. Los helpers son funciones que Handlebars permite crear al desarrollador donde implementar nuevas funcionalidades que usar más tarde en el código, en este caso se han creado dos funciones relacionadas con operaciones con strings. Handlebars proporciona diferentes operadores de bucles, pero por ejemplo si se quiere usar el operador if para comparar si dos strings son iguales no se puede, por eso se ha creado dos funciones de ayuda, los helpers, que implementan esas operaciones que se necesitan, en este caso las operaciones que se han creado sirven para saber si dos strings son iguales o si por lo contrario son diferentes. En la Figura 38 se puede observar cómo se ha implementado los dos helpers.

```
//Create  Handlebars Helpers
helpers: {
    ifEquals: function(arg1, arg2, options) {
      if (arg1 == arg2) { return options.fn(this); }
      return options.inverse(this);
    },
    ifNotEquals: function(arg1, arg2, options) {
      if (arg1 != arg2) { return options.fn(this); }
      return options.inverse(this);
    }
}
```

Figura 38: Helpers Handlebars

Una vez se ha configurado el módulo de express-handlebars se puede comenzar la implementación de la aplicación web, para ello primero sea realizado un diseño de todas las vistas que se van a desarrollar en la página web. Para crear los diseños sea utilizado una herramienta online que permite crear los diferentes mock ups o vistas de una aplicación web o móvil, esta herramienta es Balsamiq Cloud. A la hora de diseñar las diferentes vistas de la aplicación web se ha buscado implementar un diseño simple y que sea intuitivo para el usuario. Las vistas que se han diseñado se encuentran en el Anexo IV – Mock Ups Front-end.

Posteriormente se crea un archivo de tipo handlebars, con la extensión de archivo. hbs, para cada vista que se ha diseñado. Se ha creado diferentes vistas, pero se pueden dividir en tres tipos, la primera es la más básica y son las vistas en las que simplemente se muestra texto, como por ejemplo vistas donde se proporciona información al usuario o las vistas que se usan como vistas iniciales de la aplicación. El segundo tipo son las vistas donde se muestra información, pero esta vez no es simplemente texto, esta información que se muestra son objetos que vienen de una búsqueda en la base de datos, en este tipo pueden entrar las paginas donde se muestran las cerraduras que tiene un usuario o los invitados de una cerradura. Y por último están las vistas donde es el usuario el que introduce información, para ello se han utilizado tag form de HTML, con estos formularios el usuario puede introducir datos nuevos a la base de datos, nuevos invitados, nuevas cerraduras…, o también puede editar datos que se encuentren en la base de datos. Una vez se encuentran todas las vistas creadas es necesario crear algo con lo que el usuario pueda acceder a ellas, para ello se crean nuevos Endpoints en la API. Estos nuevos Endpoints que se crean ya no solo sirven para llamar a funciones que realizan búsquedas en la base de datos, algunos de estos nuevos Endpoint tienen como única función la de renderizar vistas de la web. La comunicación entre el back-end y el front-end se realiza mediante los diferentes controladores creados, en la sección en la que se explica el funcionamiento de la API se nombra que la utilidad de los controladores es implementar funciones que realizan operaciones en la base de datos, pero esa no es su única función. Los controladores además de operar con la base de datos son los encargados de renderizar las diferentes vistas que se han creado, además algunas de esas vistas necesitan datos de la base de datos por lo que en los operadores se crean tres tipos diferentes de funciones, un tipo son las que se dedican únicamente a operar con la base de datos, otro tipo es el que su única función es renderizar vistas de a la aplicación web, y el último tipo es una mezcla de los dos anteriores, se realizan una búsqueda en la base de datos y el resultado se envía a una vista la cual se renderiza.

Anteriormente se comenta como se realizan búsquedas en la base de datos y los resultados de las búsquedas se utilizan en la aplicación web, pero también existe el caso contrario, es decir en

vez del flujo de datos sea desde la base de datos hasta la aplicación web puede darse los casos contrarios donde el usuario quiera introducir, editar o eliminar información en la base de datos. En estos casos se utiliza formularios HTML con un método HTTP, en estos formularios solo se puede utilizar el método POST lo que supone un problema cuando un usuario quiere editar o eliminar información en la base de datos, para solucionar este problema se ha instalado una librería llamada method-override, esta librería permite modificar funciones y métodos para darles otro uso, en este caso se sobrescribe el método POST para que funcione como los métodos DELETE o PUT. En la Figura 39 a continuación se puede observar cómo se realiza la modificación al método POST, se añade un input oculto con el método que se quiere utilizar para lograr utilizar los métodos PUT y DELETE en lugares donde el cliente no los soporta.

```
<form action="/locks/delete/{{_id}}?_method=DELETE" method="POST">
    <input type="hidden" name="_method" value="DELETE">
    <button type="submit" class="btn btn-danger btn-block btn-sm">delete</button>
</form>
```

Figura 39: Uso de método PUT en la aplicación web

Una vez se ha implementado todas las funciones necesarias se puede finalizar el front-end, el conjunto de vistas y su navegación sería la que refleja la Figura 40.

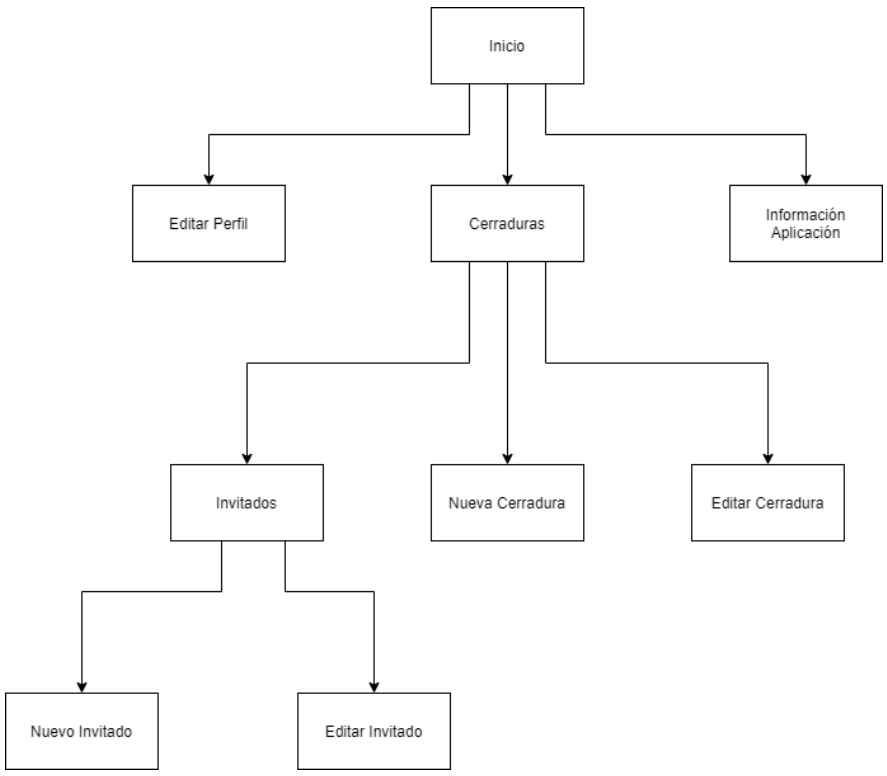


Figura 40: Mapa de navegación de la aplicación web

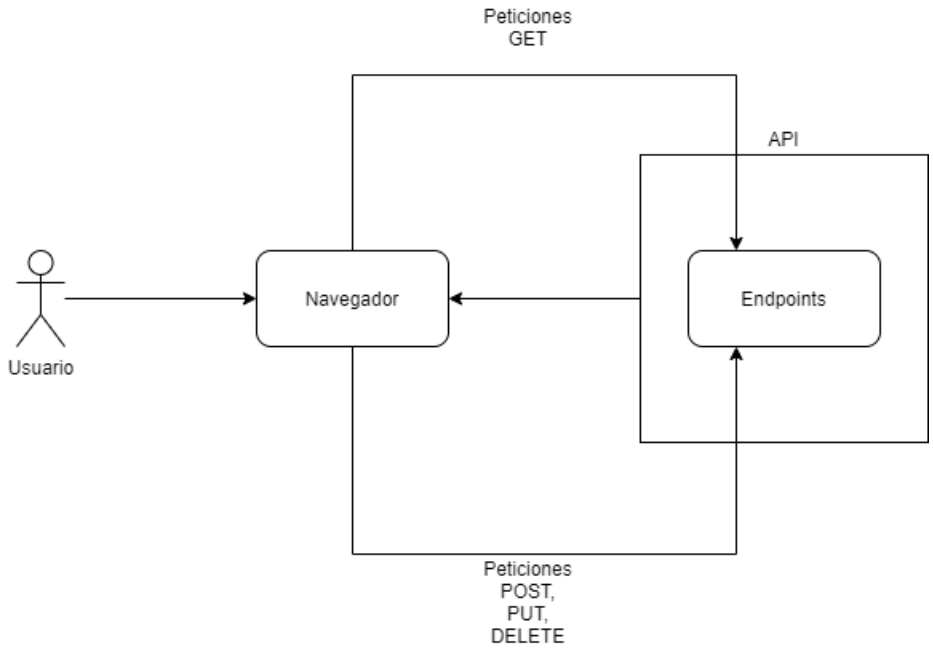


Figura 41: Arquitectura aplicación web

En la Figura 41 se puede observar el funcionamiento básico de la comunicación entre el front-end y el back-end del proyecto, el usuario mediante su navegador puede acceder a la aplicación utilizando una url, lo que es una petición GET, si existe un Endpoint con esa url la API devuelve al usuario, utilizando el navegador, información. Otra forma de comunicación es cuando es el usuario el que envía información a la base de datos o realiza operaciones en ella, para ello se utilizan formularios o botones los cuales lanzan peticiones POST, PUT o DELETE y si existen los Endpoint se realiza una operación.

5.5.10. Implementación de la aplicación móvil

Como finalización del proyecto se ha comenzado la implementación de una aplicación móvil, para ello se ha utilizado Ionic [29] como framework y como lenguaje se ha utilizado Angular y el lenguaje que proporciona Ionic, básicamente son lenguajes web como HTML, CSS…, los cuales Ionic transforma para una mejor implementación.

La aplicación desarrollada ofrece una serie de herramientas de lectura de la base de datos, para realizar esas lecturas se han implementado varios servicios en Angular donde se realizan peticiones a los Endpoints creados en la API.

La implementación de la aplicación móvil es el comienzo de un nuevo módulo que se va a desarrollar en el futuro, este futuro del proyecto se comenta en la sección Resultados, como inicio de esa futura aplicación se ha implementado una aplicación básica de lectura de la base de datos para facilitar la instalación del producto creado. El producto que se ha creado requiere una instalación y por eso se ha creado una aplicación móvil con la cual la persona encargada de la instalación pueda revisar el proceso de instalación y registro de la cerradura.

Como la aplicación se encuentra en desarrollo no se ha generado una aplicación móvil concretamente, la aplicación se encuentra en una primera fase por lo que el proyecto no ha sido convertido en una app móvil, es decir no se pueda instalar en un dispositivo Android o IOS hasta que se genere una apk, en el caso de Android.

6. Estudio Económico

En esta sección se recopila el coste económico de la realización del proyecto, para calcular este coste se debe tener en cuenta tres tipos de costes, el coste de recursos humanos, el coste de los recursos materiales utilizados en el proyecto y el coste de las licencias de los diferentes software y herramientas que se han utilizado para la implementación del proyecto. Primero, el coste de recursos humanos, para obtener la cifra de este coste es necesario recopilar el total de horas que se han utilizado para el desarrollo del proyecto, en el cómputo total de horas se recopila las horas utilizadas en la creación de la memoria, las horas de desarrollo de la aplicación y cualquier programa necesario para la implementación del proyecto, las horas invertidas en aprendizaje de conceptos para su uso en el proyecto y las horas invertidas en el montaje del circuito desarrollado como se refleja en la Tabla 5:

Tareas	Tiempo (Horas)
Implementación del circuito de la cerradura	25
Desarrollo del programa de la unidad de control	55
Desarrollo de la API/ Desarrollo de la aplicación web y aplicación móvil	100
Memoria/Informe del proyecto	90
Investigación/Aprendizaje	40
Total	310

Tabla 5: Desglose de horas del proyecto

Una vez se ha calculado el cómputo total de horas invertidas en el proyecto es necesario multiplicar las por el salario de la persona encargada de realizarlas, en este caso un programador junior, para obtener el salario de un programador junior se ha realizado una investigación del salario de un programador de esta experiencia en España en 2020 y se ha hecho una media.

Los datos del salario se han obtenido de una web de búsqueda de empleo [30] y los meses se han contado con 160 horas trabajadas.

	Tiempo (Meses)	Salario	Total
Coste Recursos Humanos	1.9	1565 €	2973 €

Tabla 6: Coste recursos humanos

Una vez se han calculado los costes de recursos humanos se va a realizar el cálculo de los costes materiales que en este proyecto ocupan una parte importante debido a que el hardware es una de las partes principales de este proyecto. Los costes materiales se pueden dividir en dos grupos, el coste de los dispositivos con los que se ha realizado el desarrollo de las aplicaciones, otros programas necesarios y la investigación requerida para el proyecto, y el otro grupo es el coste de los componentes del circuito implementado. Primero, el coste de los dispositivos utilizados en el desarrollo del proyecto, para la realización de este proyecto se ha utilizado un ordenador portátil ASUS X550VXK, con una memoria RAM de 8GB, un procesador Intel® Core™ i7-7700HQ CPU @ 2.80GHz, 2808 MHz y un disco duro de 1TB, este dispositivo está valorado en 900 €. Además del portátil se ha utilizado un dispositivo móvil valorado en 170 €, dos Raspberry Pi 3 Model B y varios periféricos para el desarrollo de la programación necesaria en la Raspberry Pi.

El coste de portátil, el dispositivo móvil y el de algún periférico se ha calculado dependiendo la media de vida que tienen estos dispositivos para calcular el coste por los meses en los que se ha desarrollado el proyecto y no utilizar el coste total del dispositivo. El número de meses que ha costado la realización del proyecto se ha calculado con los datos de la Tabla 5 y utilizando 160 horas como el número de horas trabajadas por mes.

En la Tabla 7 se refleja el coste los dispositivos utilizados.

Dispositivo	Coste Total	Tiempo de vida	Tiempo de uso (Meses)	Coste
Ordenador Portátil	900 €	5 años	1.9	28.5 €
Dispositivo Móvil	170 €	3 años	1.9	9 €
Monitor	100 €	10 años	1.9	1.5 €

Tabla 7: Costes por tiempo de vida del hardware

El coste del conjunto de los dispositivos utilizados es:

Dispositivos Utilizados	Coste (€)
Portátil ASUS X550VXK	28.5 €
Dispositivo Móvil	9 €
Raspberry Pi 3 Model B (2)	80 €
Periféricos	50 €
Total	167.5 €

Tabla 8: Coste dispositivos utilizados

El coste de los componentes utilizados en el circuito de control de la cerradura es el siguiente:

Componentes circuito	Coste (€)
Cerradura	15 €
Cámara Raspberry Pi	30 €
Fuente de Alimentación	13 €
Cableado y otros	15 €
Total	73 €

Tabla 9: Coste componentes del circuito

Una vez calculado el coste de los dispositivos utilizados y los componentes del circuito podemos observar el total del coste de recursos materiales:

	Coste (€)
Dispositivos utilizados para la implementación e investigación	167.5 €
Componentes circuito control de la cerradura	73 €
Total	240.5 €

Tabla 10: Coste Recursos Materiales

Y para finalizar con el aspecto económico del proyecto está el coste de licencias de software, en este caso al tratarse de un proyecto de esta envergadura y ser realizado por una única persona no ha existido la necesidad de adquirir la licencia de ningún software, el software utilizado es gratuito, o tiene versión gratuita que proporciona las herramientas necesarias para la implementación del proyecto, o su periodo de prueba ha sido suficiente para la realización de las tareas requeridas. Una vez calculado todos los costes podemos observar el coste total que ha requerido la implementación del proyecto:

	Coste (€)
Recursos Humanos	2973 €
Recursos Materiales	240.5 €
Total	3213.5 €

Tabla 11: Coste total del proyecto

En la Figura 42 se refleja el total del coste del proyecto y los diferentes tipos de coste.

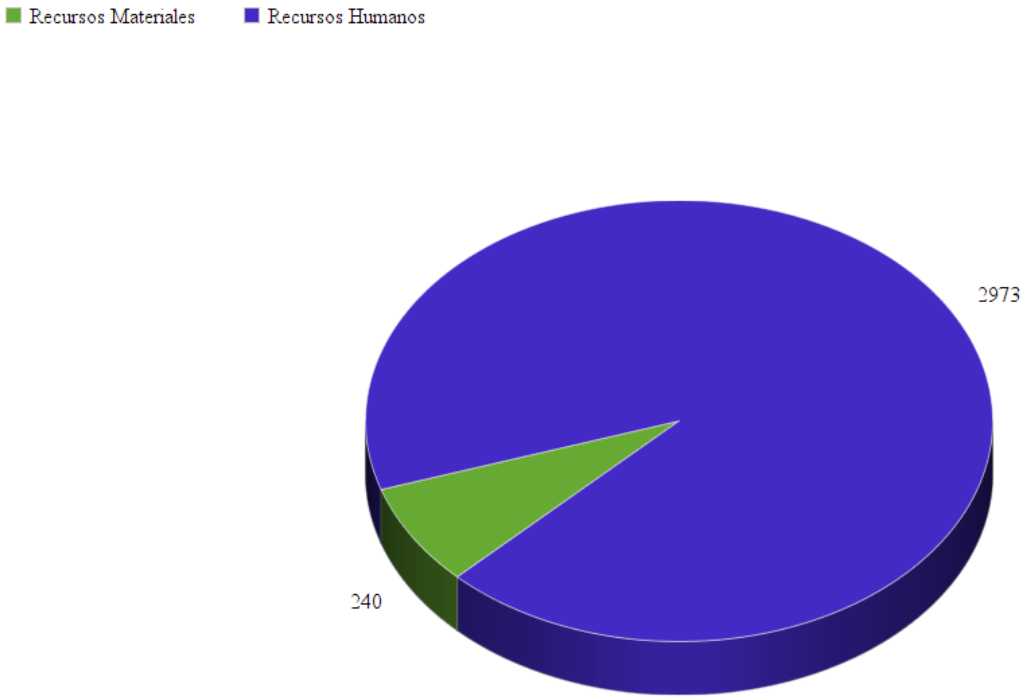


Figura 42: Gráfico coste total proyecto

7. Resultados

En esta sección se va a mostrar los productos que se han obtenido al realizar el trabajo de fin de grado además de cualquier deviación o cambio que se ha realizado en la planificación inicial del proyecto.

7.1. Objetivos cumplidos

En esta sección se va a comentar los objetivos cumplidos y como se han completado.

Los objetivos cumplidos son los siguientes:

- Dimensionar el proyecto a desarrollar: El primero objetivo que se ha completado ha sido dimensionar el proyecto, el desarrollo a comenzado con una idea bastante básica y al comienzo se ha cogido esa idea básica y se le ha dado forma logrando el diseño de un proyecto completo a desarrollar.
- Analizar y seleccionar los sensores y dispositivos más apropiados para la realización del proyecto: Cuando el proyecto estaba totalmente definido se ha estudiado que dispositivos y sensores se iban a utilizar, en este caso tras un estudio se ha decidido utilizar una Raspberry Pi y más tarde cuando se ha decidido la utilización de código QR se ha elegido una cámara como sensor para detectar el código QR.
- Estudiar e implantar el protocolo de comunicación más apropiado: Una vez se ha definido los dispositivos a utilizar se ha estudiado la manera de comunicar todos los módulos que se iban a desarrollar, para ello se ha utilizado la Raspberry Pi como unidad de control y desde ese dispositivo se realizan peticiones HTTP al resto de módulos implementados.
- Desarrollar la plataforma software para el sistema: Finalmente en este objetivo se ha implementado el programa que encargado de control la cerradura y comunicarse con el resto de los módulos, este programa se ha implementado en Python. También se ha implementado una API donde almacenar y manejar todos los datos necesarios para el sistema.

7.2. Productos Obtenidos

Primero en cuanto a los productos que se han obtenido al realizar este trabajo de fin de grado se ha obtenido un producto, pero se podría considerar que el producto está formado por dos subproductos, una cerradura eléctrica con un lector de código QR cuyo cierre y apertura se controla mediante identidad digital, y una aplicación web la cual puede ser utilizada por el usuario para gestionar las cerraduras.

A pesar de que se ha comentado que hay dos subproductos, existen otros módulos que se han desarrollado o se han utilizado, se ha desarrollado una API nueva y se ha utilizado otra API de identidad digital, estos módulos no han sido contabilizados como productos puesto que a pesar de ser esenciales en el desarrollo de proyecto no es algo que obtiene el usuario y pueda manejar, el usuario hace uso de esos módulos pero sin darse cuenta, son la cerradura y la aplicación web con lo que el usuario puede interactuar y por eso se han considerado los dos productos del proyecto.

Otro apartado importante es la razón por la que se habla de dos productos siendo los dos partes de un único proyecto. La razón es porque si se quisiera con algo de trabajo la cerradura eléctrica, la unidad de control y el código que se ejecuta en ella se podría convertir en un producto por si solo sin la necesidad del resto de módulos desarrollados, con esto me refiero a la posibilidad de convertir la cerradura eléctrica en una cerradura que se abra mediante un código QR determinado sin utilizar la identidad digital. La apertura mediante un único código QR a pesar de ser menos seguro que el proyecto creado es una posibilidad que podría acceder al mercado actual y por eso la cerradura se podría considerar un producto y el resto del proyecto otro, pero aun así eso solo es una posibilidad que podría surgir en el futuro por lo que ahora mismo la unión de todos los módulos serían el producto que se ha desarrollado. La arquitectura del producto final está formada por cinco módulos, dos de ellos son módulos que han sido implementados, por la empresa con la que se realiza el proyecto, y se utilizan algunas de sus herramientas para el proyecto, y otros tres módulos nuevos que han sido implementados.

Los módulos que forman la arquitectura del sistema final son los siguientes:

- API de Identidad Digital
- APP Móvil Wallet Id TCV
- Unidad de control (Raspberry Pi que controla la cerradura mediante código escrito en Python)
- Aplicación Web
- API desarrollada + Base de Datos (MongoDB)

A continuación, en la Figura 43 se muestra la arquitectura que tiene el proyecto en su fase final con todos sus módulos y como interactúan entre ellos.

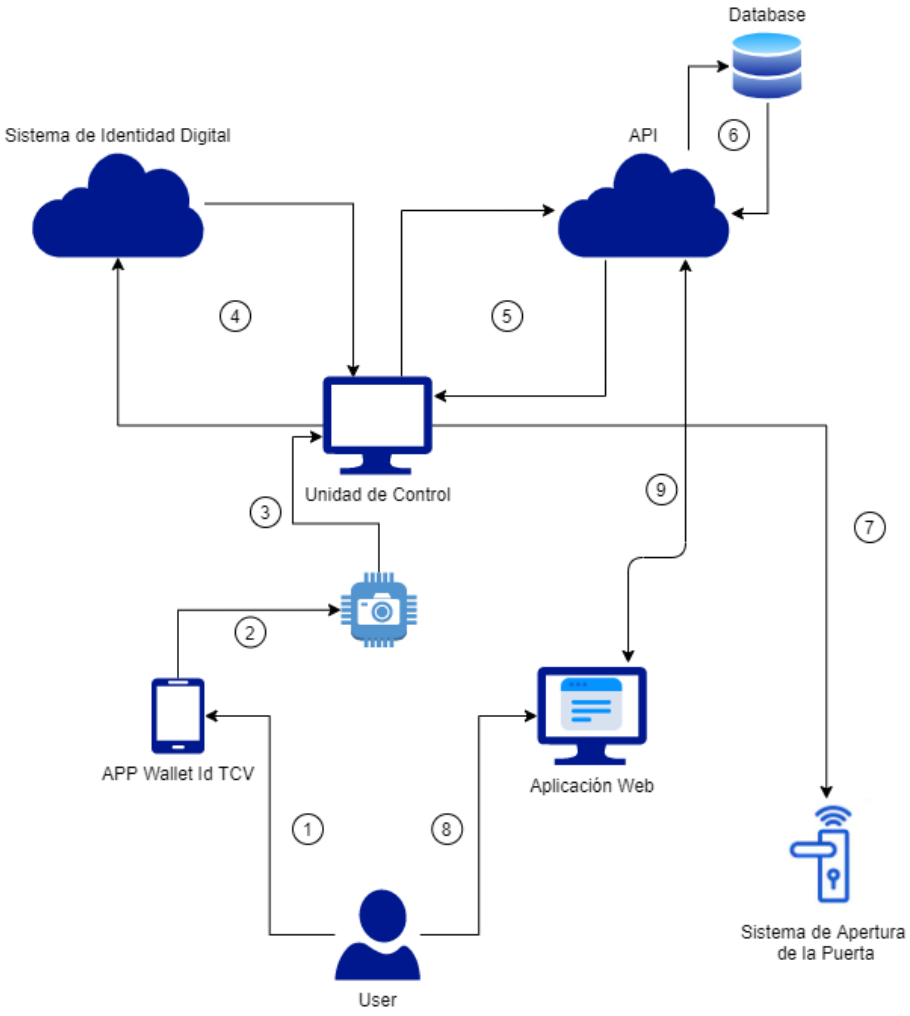


Figura 43: Arquitectura final del sistema

La Figura 43 refleja los pasos que se realiza en el sistema cuando un usuario interactúa con él, el usuario tiene dos formas de interactuar, puede intentar acceder a una cerradura, o puede utilizar la aplicación web desarrollada para interactuar con el sistema.

Los pasos que se realizan en el sistema cuando un usuario quiere acceder a la cerradura son los siguientes:

- Mediante los pasos 1, 2 y 3 el usuario utiliza la aplicación Wallet Id TCV para interactuar con la cerradura que se ha creado pidiendo acceso:

1. El usuario utiliza la aplicación Wallet Id TCV para obtener un código QR que mostrar al lector que se encuentra en la cerradura.

2. El lector detecta el código QR y envía la información obtenida a la unidad de control.
3. La unidad de control utiliza la información obtenida para preparar las peticiones a las diferentes aplicaciones.

- Una vez el usuario se ha comunicado con la cerradura, esta mediante los pasos 4, 5, 6 comprueba la identidad y autorización del usuario comunicándose con la API de Wallet Id y la API implementada:

4. Se produce una comunicación entre la unidad de control y el sistema de identidad digital obteniendo la identidad del usuario que quiere acceder si se trata de un código QR correcto.
5. La unidad de control se comunica con la API desarrollada para comprobar si el acceso es autorizado.
6. Para comprobar el acceso la API debe comprobar la identidad que le ha proporcionado la unidad de control en la base de datos.

- Si la identificación que el usuario ha utilizado con la cerradura el sistema abre la cerradura permitiendo al usuario acceder (7)

Además de la cerradura el usuario puede acceder a la aplicación web desarrollada y configurar diversos aspectos del sistema, esas operaciones se pueden observar en los pasos 8 y 9:

8. El usuario accede a la aplicación web desarrollada.
9. Con las herramientas que proporciona la aplicación web el usuario puede realizar operaciones en la base de datos, el usuario puede registrar, editar y eliminar cerraduras, además el usuario es capaz de configurar invitados que tengan acceso a las cerraduras que ha registrado.

7.3. Desviación en la metodología y en la planificación inicial

A continuación, se va a comentar las desviación y cambios que se han realizado durante el desarrollo del proyecto en la metodología y la planificación inicial.

Durante el desarrollo del trabajo de fin de grado se ha seguido la metodología SCRUM modificada que se ha establecido al comienzo del proyecto. El único posible cambio ha sido en las reuniones semanales, en ocasiones por falta de compatibilidad de horarios entre los directores del proyecto y el alumno a cargo del proyecto se han producido variaciones en las reuniones semanales, si el contenido de la reunión no era muy complejo o importante en ocasiones se han sustituido por simple comunicación mediante el correo electrónico, sin embargo si el contenido de la reunión obligaba a una comunicación más completa que el correo electrónico y existía incompatibilidad de horarios se han retrasados alguna reunión convirtiendo algunas reuniones semanales en reuniones cada dos semanas.

Otro cambio que si se ha realizado en la metodología ha sido el cambio del tamaño de algún sprint, pero ha sido a causa de una desviación en la planificación inicial y que se va a comentar a continuación.

Como se ha comentado ha existido una desviación de la planificación inicial, en el inicio del proyecto se planifico la entrega de este para Junio de 2020, pero a causa de varios problemas surgidos se ha tenido que retrasar la entrega a Septiembre de 2020 lo que supuso un cambio en la planificación inicial, básicamente se redujo la carga de trabajo y se aumentó el tamaño de los primeros sprints, otra complicación que surgió fue la aparición de otro proyecto a realizar por el alumno, el alumno ha tenido que realizar dos proyectos de características similares en cuanto a complejidad simultáneamente lo que se ha visto reflejado en el tamaño de los sprints, el alumno encargado no disponía del tiempo necesario para completar los sprints en el tiempo indicado al comienzo del proyecto por lo que el tamaño de los sprints fue aumentado, para una mejor visualización se puede observar dos diagramas de Gantt en el Anexo V – Diagramas de Gantt, un diagrama es el de la planificación inicial y el otro es el de la planificación final del proyecto. Los diagramas se han incluido en el anexo para una mejor visualización.

8. Conclusiones

Gracias a la realización del proyecto se ha podido confirmar que Internet of Things es uno de los temas principales en el ámbito de la informática en la actualidad, con la investigación que se ha realizado para el proyecto se ha podido observar que existen muchos proyectos de IoT que se encuentran activos. Se ha comprobado que IoT es un ámbito en el que la cantidad de tipos de proyectos y productos desarrollados se incrementa muy rápidamente, ya sea gracias a la implementación de un nuevo software que interaccione con objetos tecnológicos, o como se ha hecho en el proyecto con la creación de una nueva utilidad a una herramienta software ya desarrollada.

En el ámbito de la informática donde, en la actualidad este ámbito abarca casi la totalidad de los temas con millones de proyectos diferentes, es igual de importante el desarrollo de nuevos proyectos como la reutilización de proyectos ya desarrollados para en otro ámbito diferente. En el proyecto se combina un proyecto de identidad digital con la apertura de cerraduras, lo que añade un nuevo aspecto que incremente su valor, en este caso la identidad digital añade un factor muy importante que es un incremento de la seguridad, lo que especialmente en el ámbito de cierre y control de cerraduras y puertas es uno de los aspectos más importantes.

Otro aspecto que desarrollar en este apartado de la memoria son los objetivos que se marcaron al inicio del proyecto, y si han sido cumplidos todos y si no es el caso estudiar el motivo de ello.

En cuanto a los objetivos que se crearon al inicio del desarrollo del proyecto, se han completado todos excepto el último, la posibilidad de plantear un modelo analítico en función de los datos recopilados. Este objetivo no ha sido cumplido debido a una decisión que se ha tomado por el correcto futuro del proyecto, en lugar de realizar un estudio sobre la introducción del producto en el mercado se ha decidido que es más beneficioso centrarse en mejorar algunos aspectos del producto antes de estudiar su introducción al mercado actual.

Los aspectos que se consideran necesarios mejorar son la mejor accesibilidad del cliente al producto, en lugar de realizar el estudio de la introducción en el mercado se ha elegido mejorar la aplicación web creada. La aplicación web creada es una página web básica pero totalmente funcional para las características del proyecto, pero aun así se ha decidido que es mejor entregar al usuario una aplicación web más intuitiva y visualmente más atractiva, para que el usuario se sienta más cómodo al utilizarla.

Además de la mejora en la aplicación web se ha comenzado el desarrollo de una aplicación móvil, en un inicio la aplicación móvil es para facilitar la instalación del dispositivo al encargado de instalar los dispositivos, es decir sería el instalador además de que obtendría el rol de administrador de la base de datos, gracias a la aplicación móvil se podría hacer un rápido registro de la cerradura y una comprobación básica de su funcionamiento si el cliente lo requiere.

Una vez se ha comentado el grado de cumplimiento de los objetivos iniciales es el momento de comentar el futuro del proyecto, posibles mejoras del proyecto y futuras ampliaciones. El futuro del proyecto comenzaría con un testeo inicial de todos los módulos desarrollados para comprobar su funcionamiento una vez más para poder construir nuevos módulos sobre una base sólida. Una vez se tiene la base sólida sobre la que trabajar es necesario estudiar si alguno de los módulos que forman el proyecto pueden ser mejorados y en este caso son dos, la lectura de códigos QR y la aplicación móvil.

En cuanto a la aplicación móvil como se ha comentado en los párrafos anteriores, es un módulo extra que se ha realizado, en estos momentos es una aplicación móvil que facilita la instalación al trabajador encargado de la instalación de los dispositivos pero en un futuro se convertiría en una aplicación móvil que los clientes puedan utilizar como la aplicación web y además se crearía un sistema de roles, un rol cliente que ofrecería lo que se ha comentado, las herramientas que ofrece la aplicación web, y un rol administrador el cual mantendría las funcionalidades actuales de la aplicación móvil, facilitar la instalación a los trabajadores que instalan los dispositivos.

Una vez se perfeccione la aplicación móvil el otro modulo a mejorar es la lectura de QR, en el proyecto ha habido problemas con la lectura de un código QR en un video por las características de la cámara utilizada por lo que el próximo paso debe ser la utilización de una cámara de mejor calidad y la vuelta a la lectura del código QR mediante video, aunque la lectura que se ha implementado funciona siempre funciona mejor mediante la captura de video.

Otro de los aspectos a mejorar en un futuro está relacionado con la obtención de la fecha y hora por parte de la Raspberry Pi. Una Raspberry Pi es un dispositivo cuya placa base no cuenta con una pila para suministrar energía de respaldo, estas pilas se encargan de proporcionar energía al Real Time Clock que es el encargado de almacenar los ajustes de fecha y hora en el dispositivo. Por este motivo se puede desincronizar la hora del dispositivo y producir un error en el sistema,

por eso en un futuro se implementará un sistema que mediante la conexión con un servidor NTP sincronice la fecha y hora evitando posibles errores.

Antes de realizar el estudio previo al lanzamiento del producto es necesario revisar el Reglamento General de Protección de Datos (GDPR) para conocer cómo manejar la información del usuario, y en el caso de incumplir algún apartado del reglamento se buscaría una solución. Cuando se termine de mejorar los módulos existentes llegaría el momento de realizar el objetivo que no se realizó durante el desarrollo del proyecto, la posibilidad de plantear un modelo analítico en función de los datos recopilados, aunque antes de eso sería necesario la creación de una carcasa o algo donde se pueda instalar el circuito, para que se encuentre protegido y no se vea afectado el atractivo del producto por cables y partes del circuito a la vista del cliente.

En el momento que se haya acabado todas las mejoras y el estudio de la introducción del producto al mercado el próximo paso es la introducción del producto al mercado actual, pero antes del lanzamiento del producto es necesario buscar un servicio web al cual subir la API desarrollada, durante el desarrollo del proyecto debido a las características de este el servidor con la API y la base de datos ha sido local, en el caso de lanzar el producto al mercado es necesario buscar un servicio web donde se ejecute la API y la base de datos. Una vez se ha lanzado el producto al mercado es necesario ofrecer un mantenimiento a los clientes en caso de avería y realizar una actualización de vez en cuando ofreciendo algunas mejoras a los clientes.

Después de lanzar el producto al mercado se puede buscar nuevas aplicaciones las cuales solamente necesiten algún pequeño cambio en el código. El producto actual está desarrollado para la puerta de una casa, pero también se puede desarrollar por ejemplo para el control de acceso de las salas de un edificio ofreciendo un sistema mejor a la hora de asignar invitados a los diferentes cierres o también se podría aplicar a las cerraduras de coches ofreciendo al usuario abrir el coche con su móvil de una forma segura. También si se implementa un sistema mejor de horarios de acceso se puede utilizar para reservar salas de un edificio, por ejemplo, se podría utilizar para reservar y acceder diferentes salas en un edificio de educación como una universidad.

Y para finalizar la realización del proyecto me ha aportado una gran experiencia donde he aprendido nuevas tecnologías que me van a ser muy útiles en el futuro, además al tratarse de un trabajo de fin de grado realizado en colaboración con una empresa me ha aportado un pequeño avance de lo que va a ser el mundo laboral.

9. Bibliografía

[1] "The State of Digital Communications 2019." [Online]. Available: https://etno.eu/downloads/reports/thestateofdigitalcommunications2019finalpages.pdf.

[2] "Samsung Smart Doorlock." https://www.samsungsds.com/global/en/solutions/off/cddl/smart_doorlock.html.

[3] "August Smart Lock Pro." https://august.com/products/august-smart-lock-pro-connect.

[4] "ENTR de Tesa Assa Abloy." https://www.tesa.es/es/site/tesa/producto/cerraduras-inteligentes/entr/unidad-principal-entr/.

[5] "Wallet Id." https://www.wallet-id.com/.

[6] "Inycom." https://www.inycom.es/.

[7] "Web Oficial SCRUM." https://www.scrum.org/.

[8] "Trello." https://trello.com/es.

[9] "BitBucket." https://bitbucket.org/.

[10] "Visual Studio Code." https://code.visualstudio.com/.

[11] "Thonny Python IDE." https://thonny.org/.

[12] "Balsamiq Cloud," [Online]. Available: https://balsamiq.cloud/#.

[13] "Cerradura utilizada para el proyecto." https://www.amazon.es/Jis-M265743-electrico-automatico-frontal/dp/B00MDJ2PW2/ref=sr_1_2?__mk_es_ES=ÅMÅŽÕÑ&dchild=1&keywords=cierre+electrico+jis+832&qid=1597656209&sr=8-2.

[14] "Raspberry Pi y Arduino en proyectos IoT." https://www.digiteum.com/comparing-arduino-raspberry-pi-iot.

[15] "Raspberry Pi Web Oficial." https://www.raspberrypi.org/.

[16] "Wallet Id TCV." https://apps.apple.com/es/app/tcv-wallet-id/id1490996695.

[17] "Raspberry Pi Pines." https://www.raspberrypi.org/%0Ahttps://www.raspberrypi-spy.co.uk/wp-content/uploads/2012/06/Raspberry-Pi-GPIO-Header-with-Photo-702x336.png%0A.

[18] "RPi.GPIO." https://pypi.org/project/RPi.GPIO/.

[19] "OpenCV." https://pypi.org/project/opencv-python/.

[20] "Ventajas e incovenientes de NoSQL y de SQL." https://www.mongodb.com/nosql-explained/nosql-vs-sql.

[21] "Características MongoDB." https://www.mongodb.com/compare/mongodb-mysql.

[22] "NPM." https://www.npmjs.com/.

[23] "Bcrypt." https://www.npmjs.com/package/bcrypt.

[24] "Pyzbar." https://pypi.org/project/pyzbar/.

[25] "Qrtools." https://pypi.org/project/qrtools/.

[26] "Requests." https://requests.readthedocs.io/es/latest/.

[27] "Handlebars." https://handlebarsjs.com/.

[28] "Bootstrap." https://getbootstrap.com/.

[29] "Ionic." https://ionicframework.com/.

[30] "Salario Programador Jr España 2020." https://es.indeed.com/salaries/programador-junior-Salaries?period=monthly.

Anexo I – Propuesta del proyecto

Nombre alumno: Jesús Lacarte Carazo

Titulación: Graduado en Ingeniería Informática/Graduado en Diseño y Desarrollo de Videojuegos

Curso académico: 2019-2020

1. TÍTULO DEL PROYECTO

IoT asistencial

2. DESCRIPCIÓN Y JUSTIFICACIÓN DEL TEMA A TRATAR

El proyecto consiste en el estudio, análisis y desarrollo de un sistema basado en IoT que nos asista en situaciones cotidianas mediante el uso de sensorística específica, permitiéndonos recopilar datos. Adicionalmente y dependiendo de la situación sobre la que se centre el proyecto se ofrecerá la posibilidad de plantear uno o varios modelos analíticos en función de los datos recogidos. La situación en la que actuar se definirá al comienzo del proyecto, de forma que el alumno participe desde el principio en el diseño del mismo.

3. OBJETIVOS DEL PROYECTO

Los objetivos del proyecto son:

i) Definir el proyecto a desarrollar.

ii) Analizar y seleccionar los sensores y dispositivos más apropiados para la realización del mismo.

iii) Estudiar e implantar el protocolo de comunicación más apropiado.

iv) Desarrollar la plataforma software para el sistema.

v) Posibilidad de plantear un modelo analítico en función de los datos recopilados.

4. METODOLOGÍA

La metodología se establecerá en las primeras fases del proyecto

5. PLANIFICACIÓN DE TAREAS

Las tareas quedan predefinidas de manera global en los objetivos. Serán fijadas de forma concreta durante el desarrollo del proyecto

6. OBSERVACIONES ADICIONALES

Para que la asignación del proyecto sea definitiva, el alumno deberá contar con la aprobación por parte de la empresa Inycom tras una entrevista personal.

En el supuesto de no ser asignada la propuesta al alumno, se le asignará desde la coordinación de la asignatura un tutor para desarrollar una nueva propuesta.

Anexo II – Librerías utilizadas en la aplicación

1. Express:

   Express es un módulo que proporciona un conjunto de funciones y herramientas para desarrollar aplicaciones web y móvil, es el módulo más adecuado cuando se quiere montar un servidor utilizando Node.js y se quiere configurar middlewares para responder solicitudes HTTP y realizar otras operaciones relacionadas con el método HTTP y la URL.

2. connect-flash:

   Connect-flash es un módulo que permite almacenar mensajes para que sean desplegados al usuario en diferentes situaciones, por ejemplo, cuando un usuario intenta acceder a la página con una contraseña incorrecta se muestra un mensaje informando al usuario del error.

3. Bcryptjs:

   Bcrypt es una librería que proporciona herramientas y algoritmos de encriptación, en el proyecto se ha utilizado para proteger las contraseñas del usuario mediante el algoritmo hash que bcrypt proporciona.

4. express-handlebars:

   Handlebars es uno de los motores de plantillas más utilizado en la actualidad, en el proyecto se ha utilizado para desarrollar la sección de front-end de la API. Handlebars proporciona un lenguaje similar a HTML, pero añade herramientas con las cuales se facilita la comunicación entre el back-end y el front-end.

5. express-session:

   Express-session es un módulo que permite crear sesiones donde almacenar información para facilitar algunas operaciones en el front-end, por ejemplo, una vez un usuario ha accedido a la aplicación web se almacena información sobre él en la sesión para facilitar algunas operaciones en las diferentes vistas de la aplicación web.

6. method-override:

   Method-override es un módulo que permite utilizar los métodos PUT y DELETE de HTTP en lugares donde el lado del cliente de la aplicación no puede.

7. Mongoose:
   Mongoose es un módulo que permite crear un esquema para los datos almacenados en la base de datos MongoDB, gracias a eso las operaciones con esos datos se vuelven más sencillas.

8. Passport:
   Passport es un módulo de autentificación que permite autentificar peticiones.

9. Nodemon:
   Nodemon es una herramienta para el desarrollo de proyectos, este módulo facilita el desarrollo del proyecto evitando que el programador reinicie el servidor cada vez que se realiza un cambio, nodemon reinicia el servidor automáticamente cada vez que se realiza un cambio.

Anexo III – Algoritmo Hash y Salt

En el proyecto el algoritmo Hash para proporcionar seguridad a la hora de almacenar la contraseña del usuario.

```
UserSchema.methods.encryptPassword = async password => {
    const salt = await bcrypt.genSalt(10);
    return await bcrypt.hash(password,salt);
};
```

Figura 44: Código Encriptación Contraseña

Hash es una función criptográfica que mediante un algoritmo matemático transforma cualquier bloque de datos en una serie de caracteres con una longitud fija, además de utilizar el algoritmo hash se añade un valor llamado salt para añadir seguridad. El uso de salt sirve para dar un extra de seguridad a los sistemas que utilizan el hash de la información para mantenerla segura, su funcionamiento consiste en introducir una cadena de caracteres junto con la información que va a ser encriptada por el algoritmo hash, haciendo que presente un hash diferente en función de los usuarios incluso si repiten información. Si lo aplicamos al proyecto, las contraseñas que se encuentran almacenadas en la base de datos serán diferentes, aunque dos usuarios utilicen la misma contraseña. A continuación, se muestra un ejemplo de lo que sucede cuando dos usuarios repiten la misma contraseña:

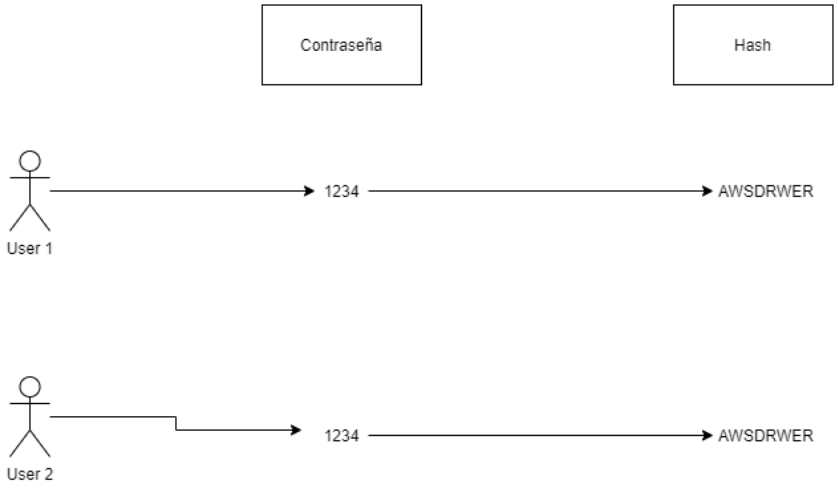


Figura 45: Explicación algoritmo Hash

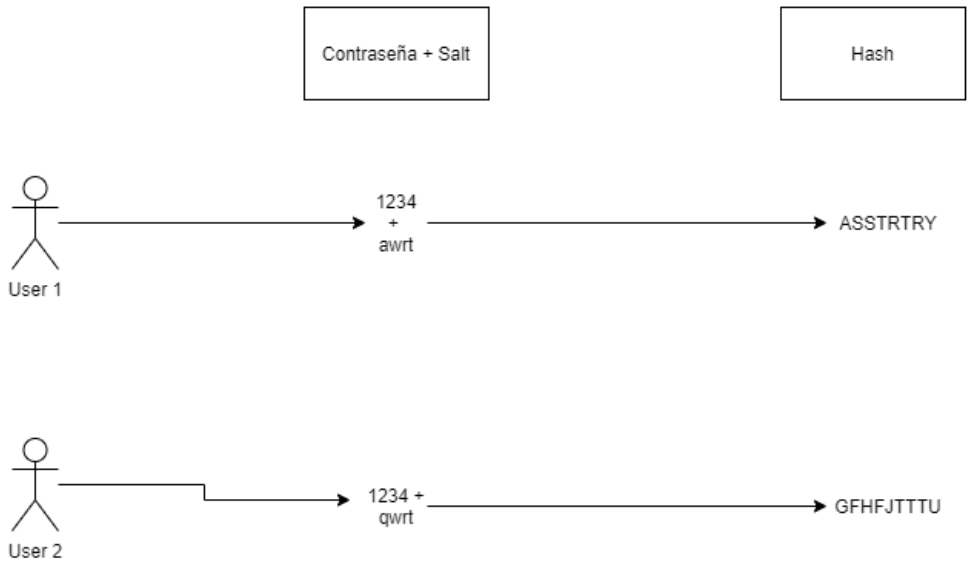


Figura 46: Explicación algoritmo Hash + Salt

Como se puede observar si dos usuarios utilizan la misma contraseña, en la base de datos se almacenaría lo mismo (el resultado de la función hash), lo que significa que, si las contraseñas fueran extraídas de la base de datos, serían más fáciles de extraer por un atacante que consiga acceder a la base de datos. Sin embargo, con el método salt, cada contraseña que quieras extraer tiene que ser probada con la salt del usuario específico, haciendo que obtener todas las contraseñas sea mucho más caro en términos de computación.

Anexo IV – Mock Ups Front-end

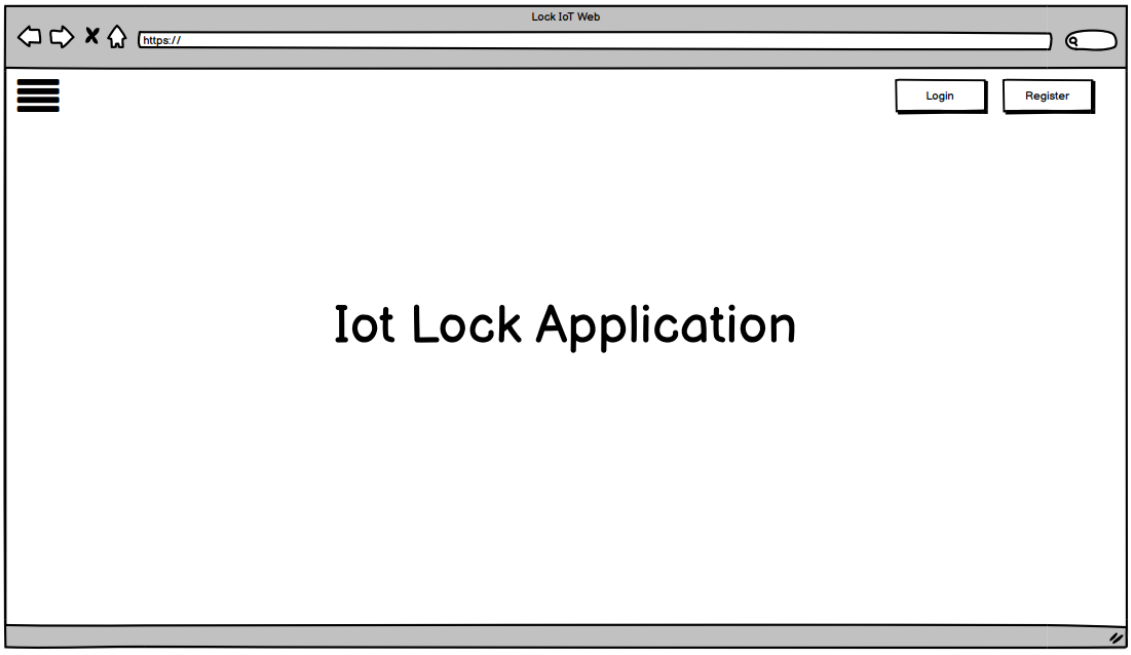


Figura 47: Mock Up web página inicio

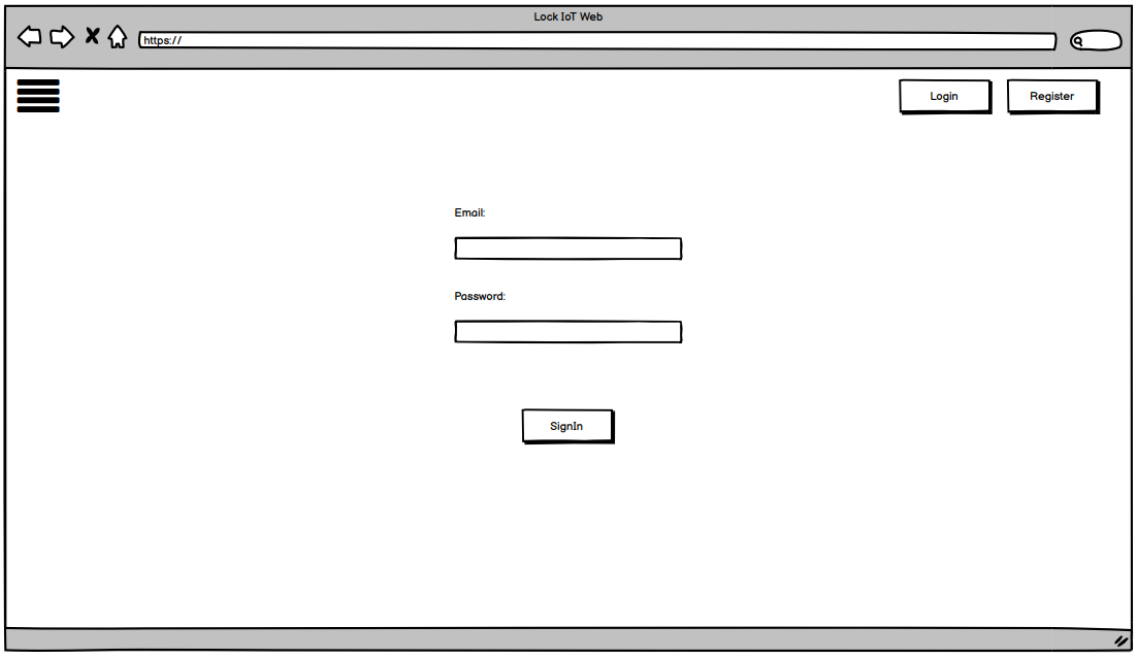


Figura 48: Mock Up web página login

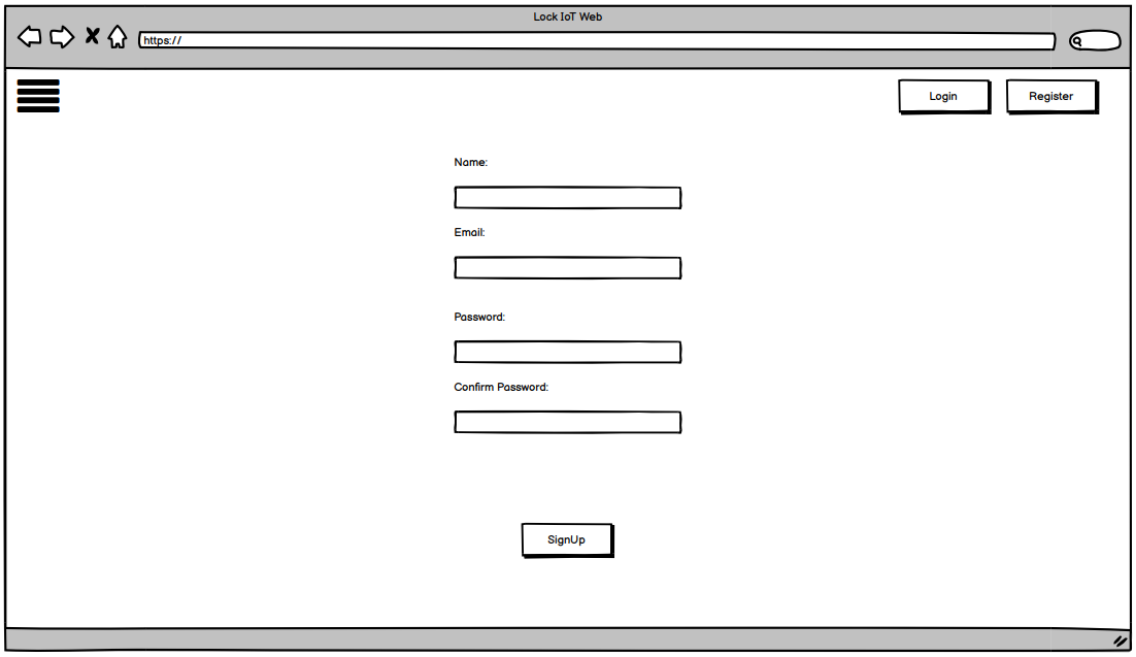


Figura 49: Mock Up web página registro

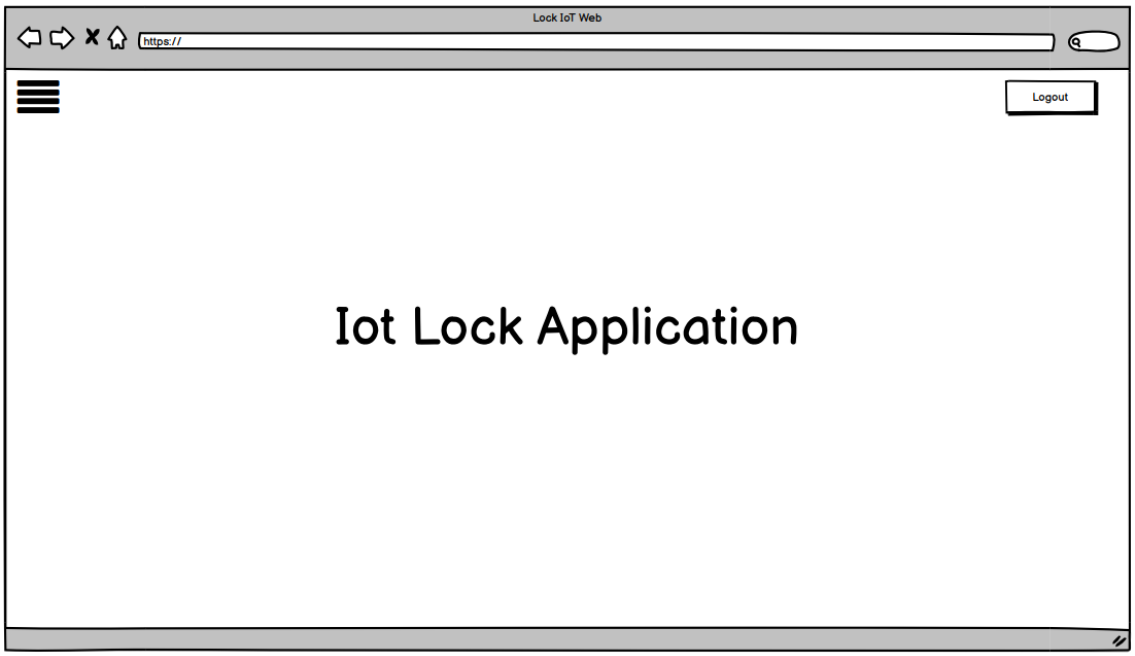


Figura 50: Mock Up web página Inicio con usuario logueado

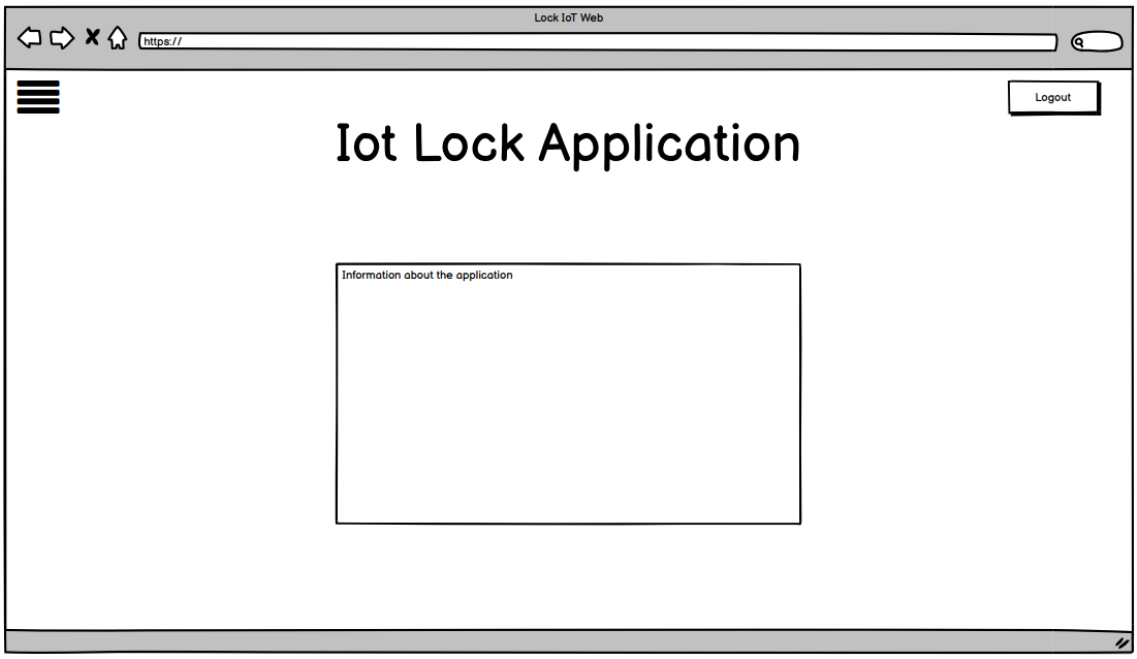


Figura 51: Mock Up web página información aplicación

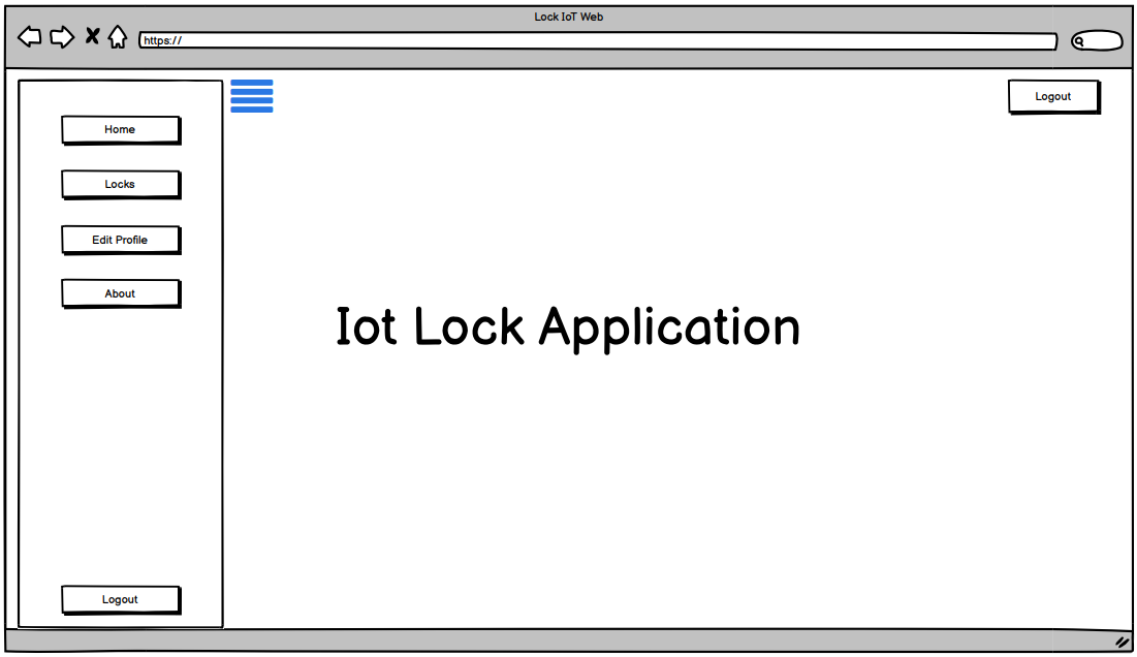


Figura 52: Mock Up web página Inicio con menu en lado

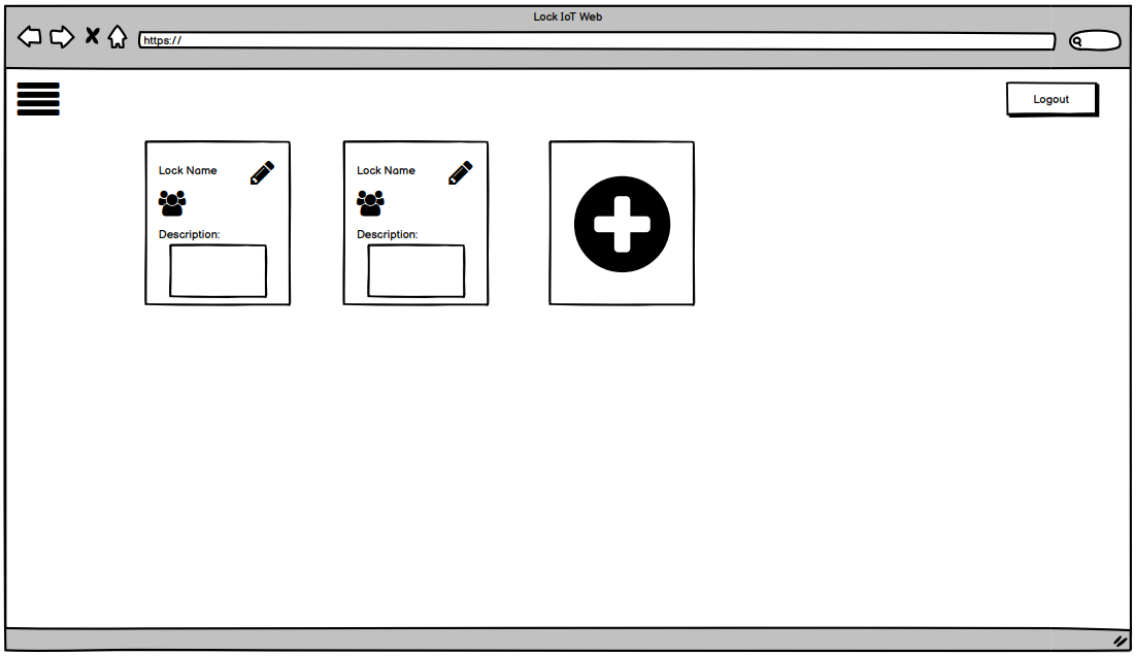


Figura 53: Mock Up web página cerraduras

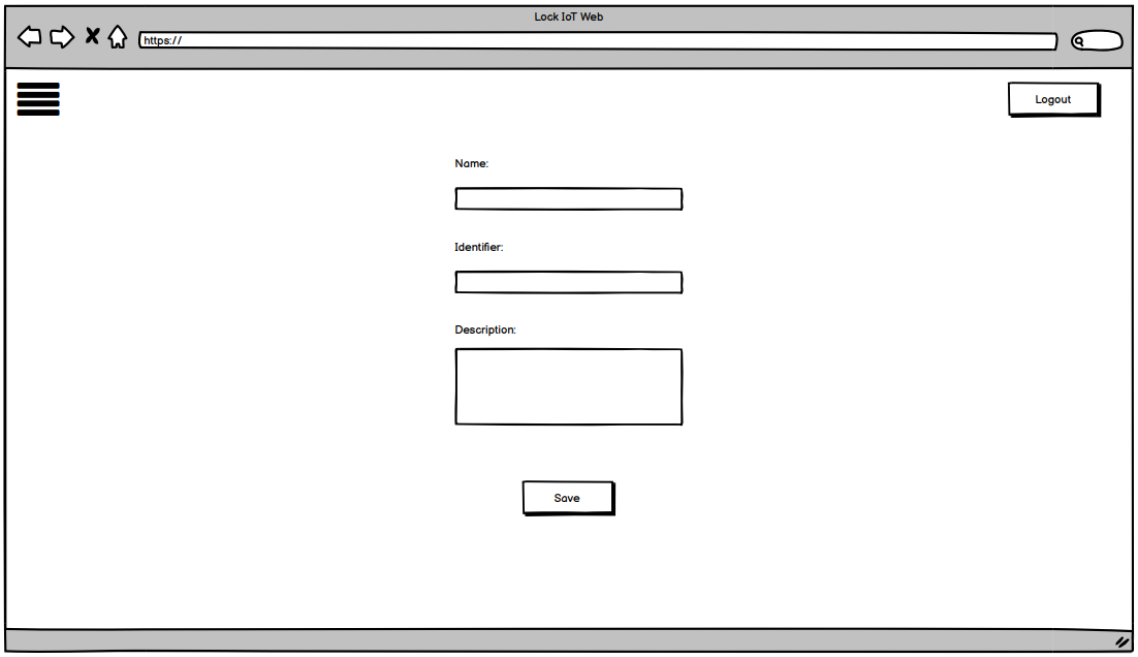


Figura 54: Mock Up web página registro cerradura

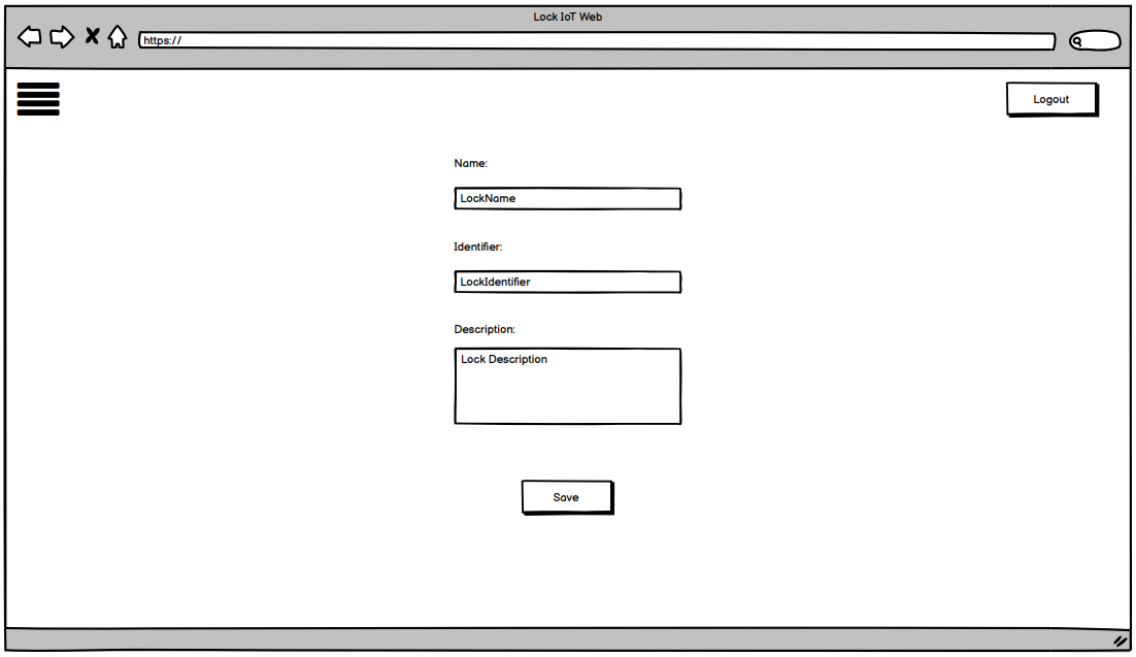


Figura 55: Mock Up web página editar cerradura

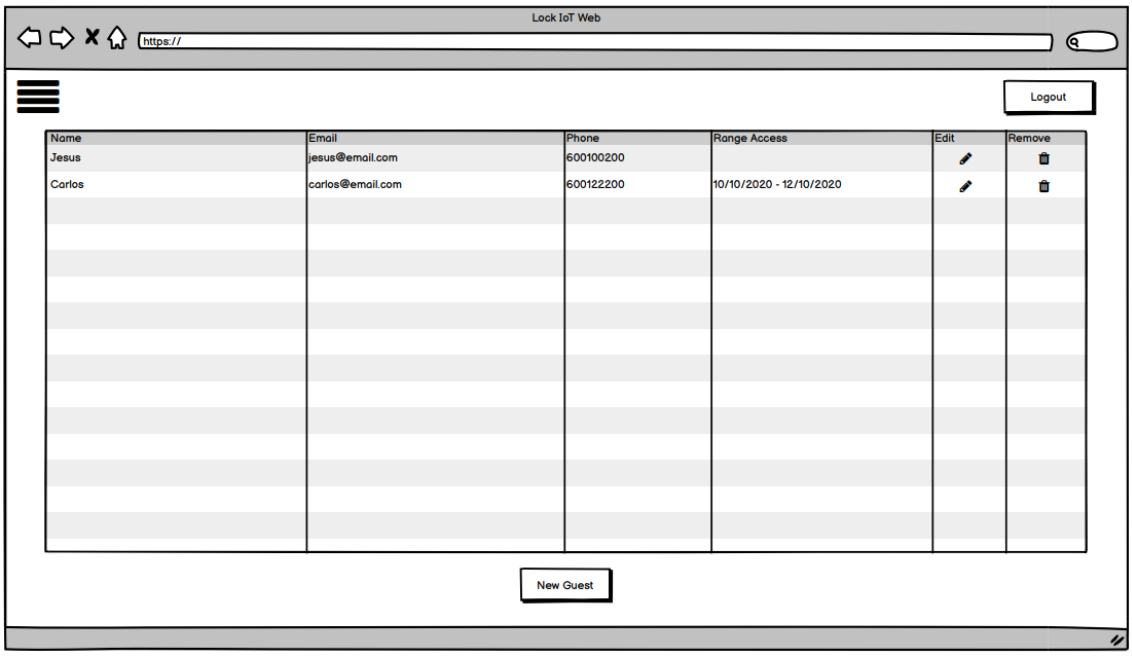


Figura 56: Mock Up web página invitados

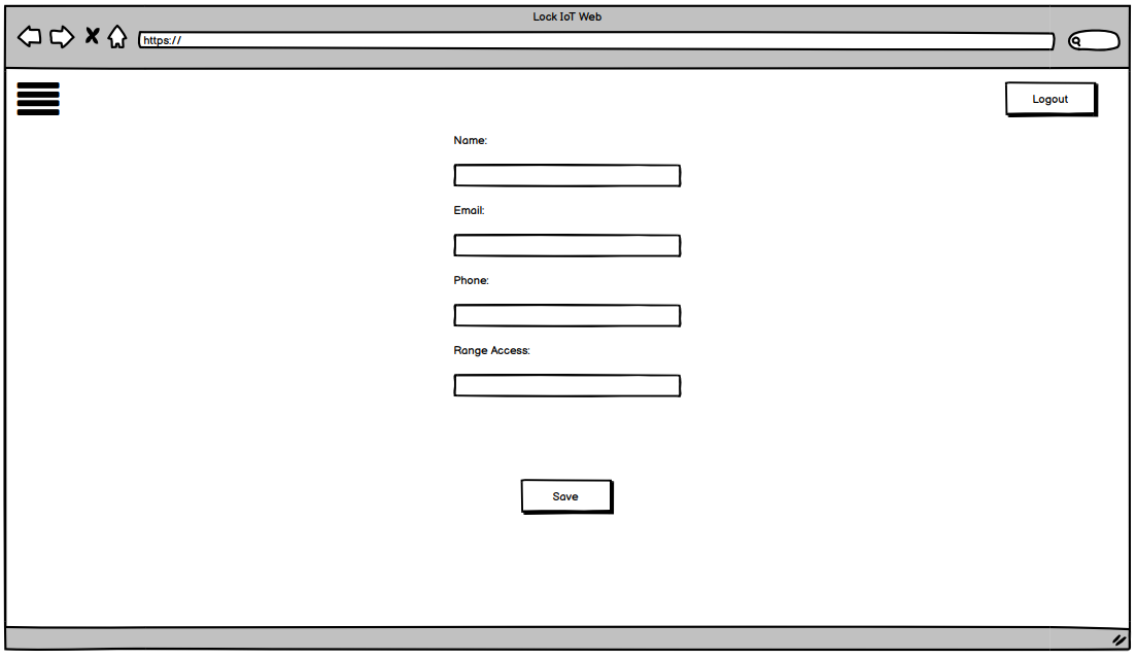


Figura 57: Mock Up web página registro invitado

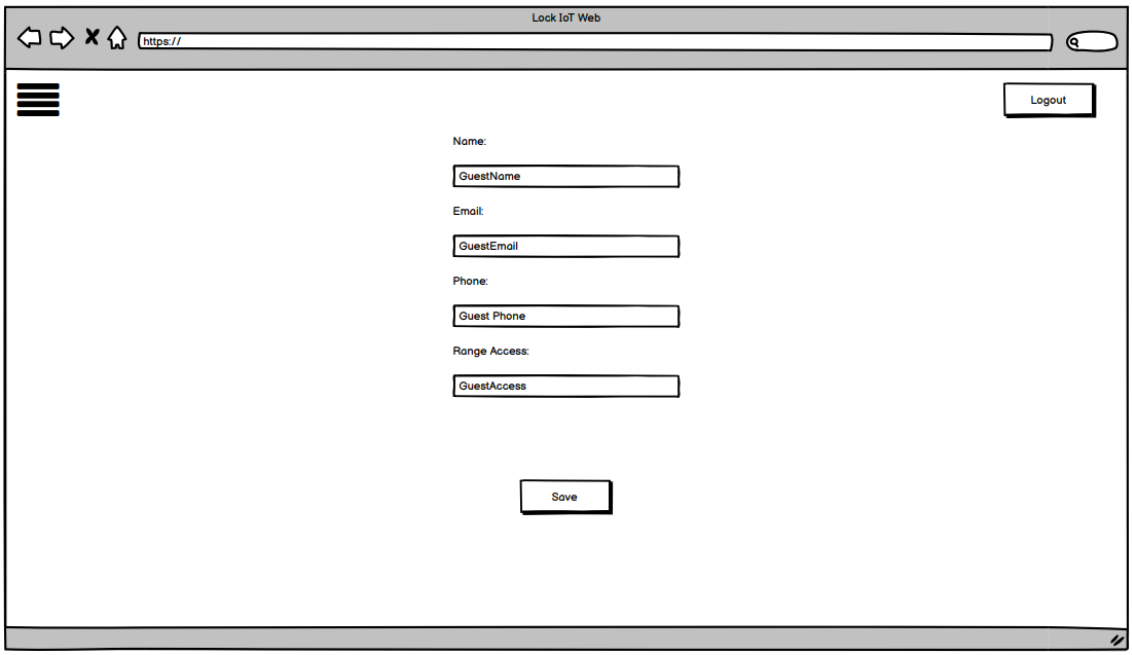


Figura 58: Mock Up web página editar invitado

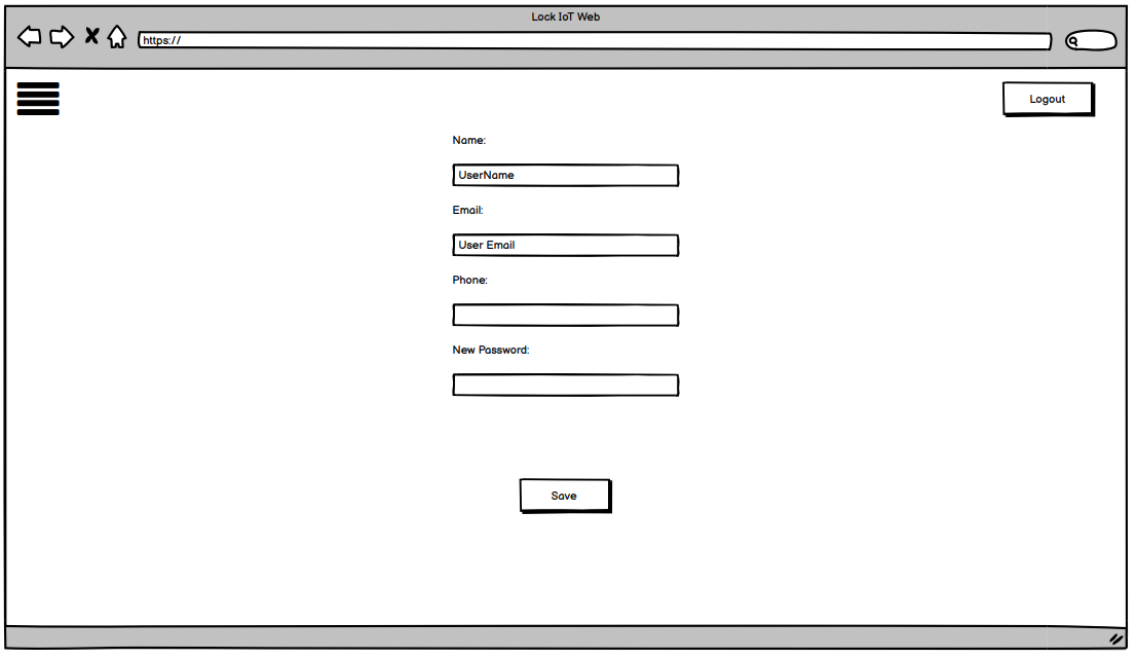


Figura 59: Mock Up web página editar usuario

Anexo V – Diagramas de Gantt

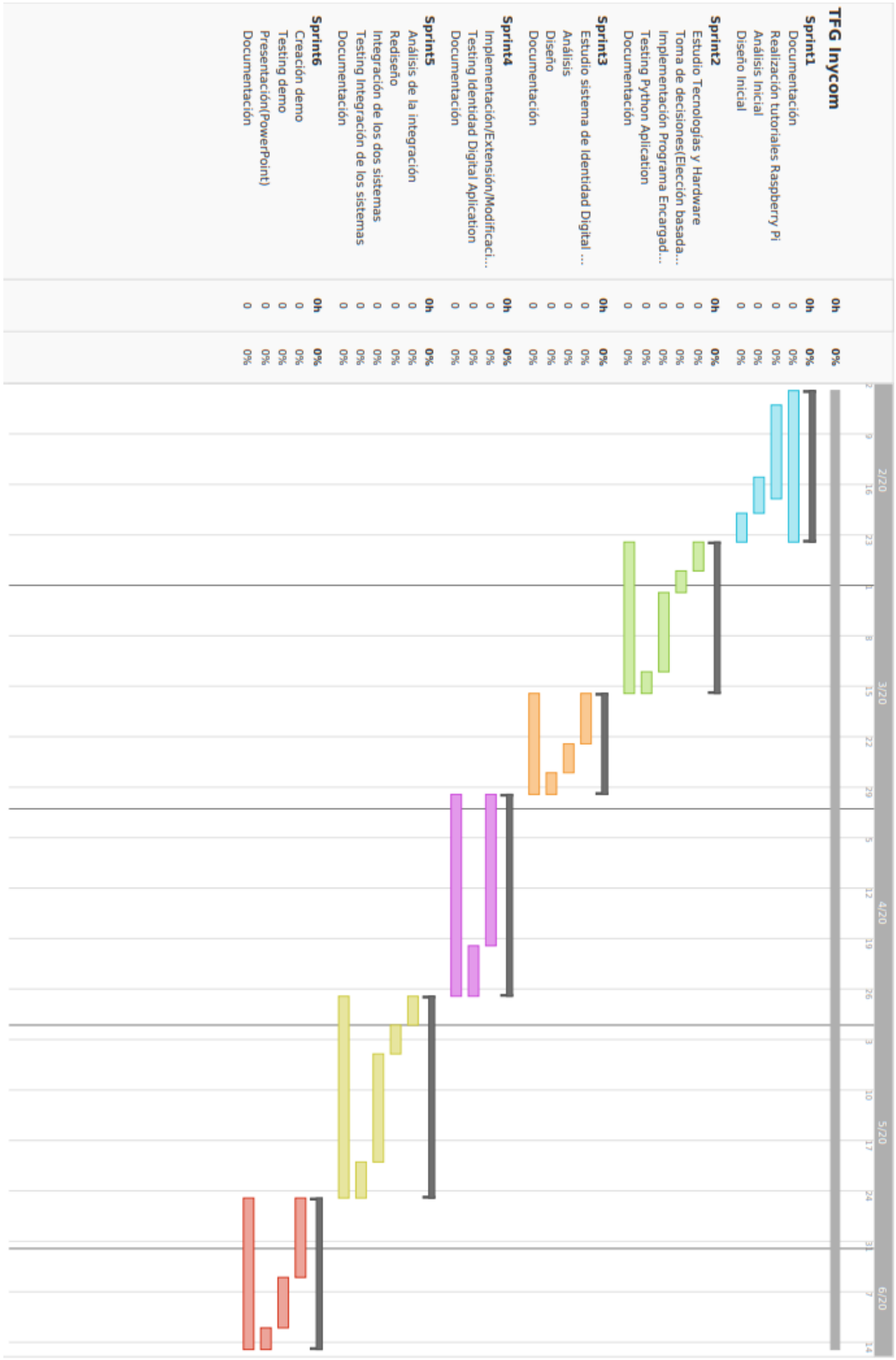


Figura 60: Diagrama de Gantt inicial

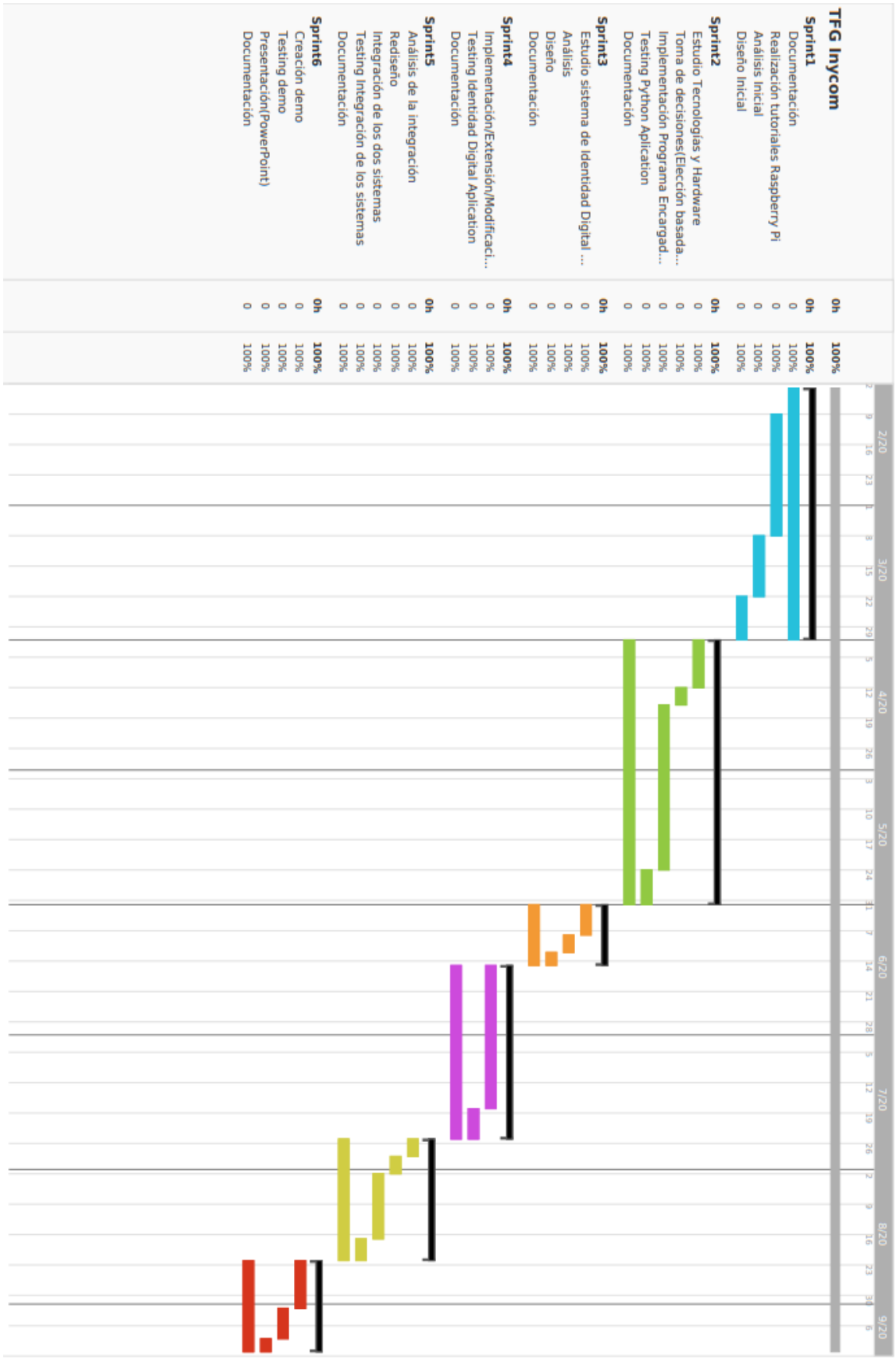


Figura 61: Diagrama de Gantt final